UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Ph.D. Thesis

Optimization of fluid bed dryer energy consumption for pharmaceutical drug processes through machine learning and cloud computing technologies

Author: Roberto Barriga Rodríguez

Directed by:

Houcine Hassan Mohamed, Universitat Politècnica Valencia

Supervised by:

Miquel Romero Obon, Industrias Farmacéuticas Almirall

July 2023

Abstract

High energy costs, the constant regulatory measures applied by administrations to maintain low healthcare costs, and the changes in healthcare regulations introduced in recent years have all significantly impacted the pharmaceutical and healthcare industry. The industry 4.0 paradigm encompasses changes in the traditional production model of the pharmaceutical industry with the inclusion of technologies beyond traditional automation. The primary goal is to achieve more cost-efficient drugs through the optimal incorporation of technologies such as advanced analytics. The manufacturing process of the pharmaceutical industry has different stages (mixing, drying, compacting, coating, packaging, etc..), and one of the most energy-expensive stages is the drying process. This process aims to extract the liquid content, such as water, by injecting warm and dry air into the system. This drying procedure time usually is predetermined and depends on the volume and the kind of units of a pharmaceutical product that must be dehydrated. On the other hand, the preheating phase can vary depending on various parameters, such as the operator's experience. It is, therefore, safe to assume that optimization of this process through advanced analytics is possible and can have a significant cost-reducing effect on the whole manufacturing process. Due to the high cost of the machinery involved in the drug production process, it is common practice in the pharmaceutical industry to try to maximize the useful life of these machines, which are not equipped with the latest sensors. Thus, a machine learning model using advanced analytics platforms, such as cloud computing, can be implemented to analyze potential energy consumption savings. This thesis is focused on improving the energy consumption in the preheating process of a fluid bed dryer by defining and implementing an IIOT (Industrial Internet of Things) Cloud computing platform. This architecture will host and run a machine learning algorithm based on Catboost modeling to predict when the optimum time is reached to stop the process, reduce its duration, and consequently its energy consumption. Experimental results show that it is possible to reduce the preheating process by 45% of its time duration, consequently reducing energy consumption by up to 2.8 MWh per year.

Resumen

Los altos costes energéticos, las constantes medidas regulatorias aplicadas por las administraciones para mantener bajos los costes sanitarios, así como los cambios en la normativa sanitaria que se han introducido en los últimos años, han tenido un impacto significativo en la industria farmacéutica y sanitaria. El paradigma Industria 4.0 engloba cambios en el modelo productivo tradicional de la industria farmacéutica con la inclusión de tecnologías que van más allá de la automatización tradicional. El objetivo principal es lograr medicamentos más rentables mediante la incorporación óptima de tecnologías como la analítica avanzada. El proceso de fabricación de las industrias farmacéuticas tiene diferentes etapas (mezclado, secado, compactado, recubrimiento, envasado, etc.) donde una de las etapas más costosas energéticamente es el proceso de secado. El objetivo durante este proceso es extraer el contenido de líquidos como el agua mediante la inyección de aire caliente y seco en el sistema. Este tiempo de secado normalmente está predeterminado y depende del volumen y el tipo de unidades de producto farmacéutico que se deben deshidratar. Por otro lado, la fase de precalentamiento puede variar dependiendo de varios parámetros como la experiencia del operador. Por lo tanto, es posible asumir que una optimización de este proceso a través de analítica avanzada es posible y puede tener un efecto significativo en la reducción de costes en todo el proceso de fabricación. Debido al alto coste de la maquinaria involucrada en el proceso de producción de medicamentos, es una práctica común en la industria farmacéutica tratar de maximizar la vida útil de estas máquinas que no están equipados con los últimos sensores. Así pues, es posible implementar un modelo de aprendizaje automático que utilice plataformas de analítica avanzada, como la computación en la nube, para analizar los posibles ahorros en el consumo de energía. Esta tesis está enfocada en mejorar el consumo de energía en el proceso de precalentamiento de un secador de lecho fluido, mediante la definición e implementación de una plataforma de computación en la nube IIOT (Industrial Internet of Things)-Cloud, para alojar y ejecutar un algoritmo de aprendizaje automático basado en el modelo Catboost, para predecir cuándo es el momento óptimo para detener el proceso y reducir su duración y, en consecuencia, su consumo energético. Los resultados experimentales muestran que es posible reducir el proceso de precalentamiento en un 45% de su duración en tiempo y, en consecuencia, reducir el consumo de energía hasta 2.8 MWh por año.

Resum

Els elevats costos energètics, les constants mesures reguladores aplicades per les administracions per mantenir uns costos assistencials baixos, així com els canvis en la normativa sanitària que s'han introduït en els darrers anys, han tingut un impacte important en el sector farmacèutic i sanitari. El paradigma de la indústria 4.0 engloba els canvis en el model de producció tradicional de la indústria farmacèutica amb la inclusió de tecnologies que van més enllà de l'automatització tradicional. L'objectiu principal és aconseguir fàrmacs més rendibles mitjançant la incorporació òptima de tecnologies com l'analítica avançada. El procés de fabricació de les indústries farmacèutiques té diferents etapes (mescla, assecat, compactació, recobriment, envasat, etc.) on una de les etapes més costoses energèticament és el procés d'assecat. L'objectiu d'aquest procés és extreure el contingut de líquids com l'aigua injectant aire calent i sec al sistema. Aquest temps de procediment d'assecat normalment està predeterminat i depèn del volum i del tipus d'unitats de producte farmacèutic que cal deshidratar. D'altra banda, la fase de preescalfament pot variar en funció de diversos paràmetres com l'experiència de l'operador. Per tant, podem assumir que una optimització d'aquest procés mitjançant analítiques avançades és possible i pot tenir un efecte significatiu de reducció de costos en tot el procés de fabricació. A causa de l'elevat cost de la maquinària implicada en el procés de producció de fàrmacs, és una pràctica habitual a la indústria farmacèutica intentar maximitzar la vida útil d'aquestes màquines que no estan equipats amb els darrers sensors. Així, es pot implementar un model d'aprenentatge automàtic que utilitza plataformes de analítiques avançades com la computació en núvol, per analitzar l'estalvi potencial del consum d'energia. Aquesta tesis està enfocada a millorar el consum d'energia en el procés de preescalfament d'un assecador de llit fluid, mitjançant la definició i implementació d'una plataforma IIOT (Industrial Internet of Things)-Cloud computing, per allotjar i executar un algorisme d'aprenentatge automàtic basat en el modelatge Catboost, per predir quan és el moment òptim per aturar el procés i reduir-ne la durada, i en conseqüència el seu consum energètic. Els resultats de l'experiment mostren que és possible reduir el procés de preescalfament en un 45% de la seva durada en temps i, en conseqüència, reduir el consum d'energia fins a 2.8 MWh anuals.

Acknowledgments

This work is dedicated in the first place to my wife and my son for their patience during these last years. To my parents for their unconditional support throughout my life in the different projects on which I have embarked, and to my thesis supervisors Houcine and Miquel for their support and motivation during the execution of this thesis.

Contents

1 Introduction

1.1. Motivation

The manufacturing process of medicines is divided into several phases: weighing, granulation, drying, sieving, mixing, compression, and packaging. Fluid bed drying technology is commonly used in pharmaceutical manufacturing due to the high efficiency of drying granules obtained by wet granulation. The biggest challenge when using a fluid bed dryer is to reduce the massive amount of time and energy the machine takes to complete the process. The drying process consists of three phases: (i) preheating the

machine without introducing any type of product, (ii) drying the product, and (iii) cooling the machine for product cooling. Cost is associated with all three phases, i.e., time consumed by machines and the energy required to heat and send the air. Moreover, the cost is also associated with the number of operators handling the machine.

The economic situation, the constant measures applied by the administrations to contain healthcare costs, and the changes in healthcare regulations that have been introduced in recent years have a significant impact on the rise of the production costs of pharmaceuticals. For this reason and due to the high cost of fluid bed dryers and the rest of the machinery involved in the production process of medicines, an attempt is made to extend the useful life of these machines by maximum. The industry 4.0 paradigm encompasses changes in the traditional production model of the pharmaceutical industry with the inclusion of technologies that go beyond traditional automation [Arden et al., 2021]. The primary goal is to achieve more cost-efficient drugs through the optimal incorporation of technologies such as advanced analytics [Chi et al., 2009].

In the pharmaceutical sector, fluid bed dryers are frequently used to reduce the water content of medicinal powders and their granules. The emulsification of feed materials is the basis of the equipment's operation. In a fluidization procedure, heated air is injected with high pressure with the help of a perforated bed of moistened solid particles. The humid particles are raised from the bottom of the tank and stopped in the fluidized-state airflow. Regular interaction between both wet solids and hot gasses is used to transfer heat. The dryer vapors carry the vaporized water away. Exhaust gasses are sometimes completely reprocessed to conserve energy [Arun, 2015].

Notably, most fluid bed dryers in production plants are not equipped with sensors that indicate when the machine has reached the optimum temperature for any of the three phases (preheating, drying, and cooling). They are usually performed in a deterministic way. That is, fixed times are used for each phase of the process, and these times are controlled by the operator who manages the machine. Also, during the drying process, the operator stops the machine after a specific time to obtain a sample of the product and to measure the humidity to check whether any of the critical parameters of the machine should be adjusted (inlet air temperature or airflow).

Due to the high cost of the machinery involved in the drug production process, it is common practice in the pharmaceutical industry to try to maximize the useful life of these machines. Therefore, old fluidized bed dryers are not equipped with the latest sensors.

It is observed that this situation is frequent in the pharmaceutical industry (that machines are amortized in the long term or reused). Thus, a model implemented in equipment with older sensors can imply actual savings for a company.

This thesis has been developed and implemented at Almirall [ALM], a manufacturing plant facility in Sant Andreu de la Barca, Barcelona, Spain (Figure 1).

Figure 1. Almirall manufacturing facilities at Sant Andre de la Barca, Barcelona, Spain (www.almirall.com)

Almirall is a leading medical dermatology-focused global pharmaceutical company that partners with healthcare professionals, applying Science to provide medical solutions to patients & future generations. Almirall is focused on medical dermatology addresses sustainable, granular unmet needs in well-defined patient and indication sub-groups. The company, founded over 75 years ago and with headquarters in Barcelona, is listed on the Spanish Stock Exchange (ticker: ALM) and was part of the IBEX35, IBerian IndEX Spain's principal stock exchange during the years 2020 and 2022. Almirall provides medical solutions and a product portfolio marketed through 13 affiliates, operating in 21 countries in Europe and the US. Almirall has agreements with strategic partners in over 70 countries on the five continents that contributes to its global business model [ALM]. Almirall has 1785 employees, with total revenues of 814.5 € million in the fiscal year 2020), and it has three research and development facilities and three manufacturing sites in Spain and Germany. Almirall facilities are structured to optimize the sustainable use of resources, with a particular focus on energy efficiency. Almirall is taking concrete steps

to reduce the dependency on electricity and gas and reduce our consumption. As an example, since 2016, Almirall has been moving to solar power with the installation of photovoltaic panels at our chemical plant in Sant Celoni, Spain (300 kW) and their pharmaceutical plant in Sant Andreu de la Barca, Spain (800 kW). Almirall has been able to reduce its energy consumption to 8%, from 2017 to 2020 as seen in Table 1.

Evolution of energy consumption, 2018-2020

Energy consumption (MWh)	2018	2019	2020
Natural Gas	22,509	21,741	21,315
Company electricity	28,615	27,142	25,859
Renewable energy Self-produced	402	1,505	1,524

Table 1. Almirall evolution energy consumption (www.almirall.com)

Almirall is interested in using artificial intelligence or machine learning predictive models to improve its energy consumption in its manufacturing process; what is the objective of this work. This thesis has been developed in the most important manufacturing site of Almirall manufacturing facilities, located at Sant Andreu de la Barca, Barcelona, Spain. Figure 2 shows some of the products produced in Sant Andreu de la Barca manufacturing plant.

Figure 2. Almirall products

Several types of machinery produce these products, such as mixers, dryers, compacters, coaters, and packaging lines. This thesis is focused on improving the energy

consumption in the drying process by defining and implementing an IIOT (Industrial Internet of Things) -Cloud computing platform to host and run a machine learning algorithm based on Catboost modeling to predict when the optimum time is to stop the process and reducing its duration, and consequently its energy consumption. After connecting an actual fluid bed dryer to our architecture, it is demonstrated that it can be saved up to 2.8 MWh per year just for one of the processes from the fluid bed dryer, the preheating. The architecture defined in this thesis could also be used for other manufacturing machineries involved in the tablet manufacturing process, such as mixers, compacters, coaters, and packaging lines, by adapting the prediction model presented to a more suitable algorithm, based on the data available from the sensors. The energy savings could be up to 800 MWh per year, considering the number of manufacturing equipment in the different Almirall production plants.

1.2. Objectives and Contributions

The primary objective of the present research is to propose a machine-learning model that can reduce the time needed for the preheating and drying phases, therefore, reducing overall energy consumption. Furthermore, since the experiments were performed on the Fielder Aeromatic MP 6/8 (FAMP68), an older machine, the methodology used to develop the model can be implemented in a wide range of equipment that does not possess state-of-the-art sensor technology.

To obtain the data from the Fluid bed dryers, 56 sensors that measure inlet/outlet air temperature, airflow, and other outputs were used. The data was collected by a PLC (Programmable Logic Controller), stored in SCADA (Supervisory Control and Data Acquisition), and then uploaded to the Azure cloud to develop the model. We examined all data collected during this process to find information that can assist us in optimizing a preheating stage. After evaluating a set of ML algorithms, the Catboost algorithm was selected to develop the model for reducing energy consumption during the preheating and drying phases. The investigation and evaluation of the trial findings led to identifying the optimal configuration.

Using our model, we were able to reduce the preheating time on average by 45 minutes. Regarding energy consumption, we can save 13.95 kWh per batch of 150 kg of a drug (API- Active Product ingredient) during the preheating phase. Considering a production of 200 batches per year, we save an average of 2.8 MWh during the preheating phase. It is important to note that the experiments were performed in an actual pharmaceutical plant of a multinational company in Barcelona- Almirall.

The thesis is divided into several parts. Firstly, an overview of industry 4.0, the opportunities arising in producing solid drugs, and the use of machine learning techniques in the industry. Secondly, fluid bed dryer historical data will be analyzed to identify critical variables and patterns using preprocessing advanced analytics techniques. Thirdly, different machine learning algorithms will be evaluated using the collected data to select the most accurate one. Finally, an IIOT-Cloud computing architecture will be presented and implemented, showing the results regarding energy savings from analyzing fluid bed dryer data in real-time. Besides, some potential future work will be mentioned.

In summary, the main achievements presented in this thesis are:

- A review of state of art on fluid bed dryer energy consumption and utilization of machine learning algorithms to improve manufacturing production process.

- The proposal of applying EDA (exploration data analysis) methodology to analyze and optimize a large-scale drug production process, such as the preheating drying process for solid drugs (pharmaceutical granules) through a fluid bed dryer.

- The study and adaptation of the machine learning algorithm Catboost for predicting the optimum preheating time based on fluid bed dryer air inlet-outlet temperature differences in actual equipment.

- The proposal and implementation in an actual manufacturing plant located in Barcelona of an IIOT (Industrial Internet of Things) and Edge – Cloud Computing architecture based on Microsoft Azure to ingest, store and process fluid bed dryer data and manage our Catboost prediction model implementation.

- The results demonstrate how implementing the Catboost machine learning algorithm in Microsoft Azure architecture and using real-time data from the fluid bed dryer, an average energy consumption saving is around 2.8 MWh per year for 200 batches of a drug product.

1.3. Structure of the thesis

This thesis has 6 chapters:

- **Chapter 1 Introduction:** The introduction, motivation, and structure of the thesis.

- **Chapter 2 Related Work and theoretical framework**: Presents state-of-the-art and a review of the most relevant literature on the different topics addressed in this thesis, including an introduction to the pharmaceutical manufacturing process, a brief description of fluid bed dryer operations, and an introduction to machine learning, including some state of art related work in the field of applying machine learning to reduce the energy consumption in the manufacturing industry.

- **Chapter 3 Data capture and exploratory data analysis (EDA):** This chapter provides a practical example of how to use EDA (exploration data analysis) methodology to analyze and optimize a large-scale drug production process, such as the preheating drying process for solid drugs (pharmaceutical granules) through a fluid bed dryer.

- **Chapter 4 Machine Learning model development:** Presents the selection and development of a data model algorithm to predict the optimum time to stop the fluid bed dryer preheating process.

- **Chapter 5: Fluid Bed Dryer Cloud-IIOT architecture:** Provides the definition and implementation of an IIOT and Edge – Cloud computing platform connected through OPC server technology to a fluid bed dryer in real-time and presents a calculation of how much time and energy we can reduce if we provide to the fluid bed dryer operators a prediction that indicates when the optimum moment is to stop the preheating process.

- **Chapter 6 Conclusions and future work:** Presents a summary of our solution's main conclusions and potential future implementations.

2 Related Work and theoretical framework

The primary objective of this thesis is to study how machine learning and advanced analytics techniques can improve pharmaceutical production processes by reducing their energy consumption, with a particular focus on fluid bed dryer processes for solid pharmaceutical dosages, also known as tablets.

A review of the most relevant literature on these topics is presented. First, we introduce the pharmaceutical manufacturing process, focusing on the fluid bed drying process and briefly explaining how a fluid bed dryer works. Next, we explore the current paradigm of Industry 4.0 and how it is being tackled by the pharmaceutical industry, creating a new sub-concept called Pharma 4.0. We will comment on the digital twin technology, as it is one of the main enablers for Pharma 4.0. We will review their relationship with ICH, the International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use. This organization brings together the regulatory authorities and the pharmaceutical industry to discuss scientific and technical aspects of pharmaceuticals and develop guidelines. Finally, we present an introduction to machine learning, including some state-of-the-artwork in applying machine learning to reduce energy consumption in the manufacturing industry.

2.1. Pharmaceutical manufacturing process

Large-scale manufacture of medicines requires advanced technologies that allow all the parameters of the production process to be controlled. In addition, this type of medicine requires that the active ingredients be worked on in a closed circuit and by highly qualified professionals. To work in optimal conditions, professionals have personal protection equipment such as autonomous breathing systems and specially adapted divers. In summary, the pharmaceutical manufacturing process can be summarized in 8 steps [Burggraeve et al., 2013]:

- **Weigh-in.** First, the active ingredients and excipients necessary for manufacturing a batch of medicines are divided and weighed with extreme precision, including the quantities indicated in the formula of each specialty, also called the recipe.

- **Granulation.** In this phase, the active ingredient and the excipients are mixed with a solution to form the wet granulate. To achieve a perfect mix, the equipment has two agitators that can rotate at a speed of more than 200 revolutions per minute.

- **Drying**. During the drying process, this solution is extracted to obtain a granulate with the appropriate moisture content. Drying is done with hot, filtered air, for which up to 800 cubic meters are used every hour. This is the process that we will be focused on in this thesis.

- **Screening.** This phase's objective is always to obtain the appropriate granule size for each drug.

- **Mixed.** The necessary excipients for its compression are added to the granules and then mixed until they are perfectly homogeneous. In this process, two specific parameters are controlled for each specialty: the tank's speed and turning time.

- **Compression.** The granulate obtained after mixing is subjected to pressure to obtain tablets. In this phase, one hundred percent of the tablets produced are controlled in real-time, and statistical control of their weight, hardness, and dimension is also carried out. The team with which it works has a speed of 8,000 tablets per hour.

- **Coating.** Sometimes, tablets are coated with a polymer film, which is applied using a spray gun system. This coating constitutes a barrier between the tablet and the environment. This is the most delicate part of the process, and its functions can be to isolate it from light, modulate its release at the intestinal or gastric level, or simply give the tablet the desired color.

- **Packaging.** Once manufactured, the medicines go through the packaging process, placed in a blister to protect them from the environment. Together with the prospectus, they are packaged in their corresponding individual cases.

2.1.1. Fluid bed dryer operations

In this thesis, we will focus on the drying process, using fluid bed drying machines as they are widely used in the pharmaceutical industry to dry the granules of future pharmaceutical tablets. In this process, the drying is carried out through the transmission of hot air and the extraction of the humidity of the product by diffusion and forced convection. During this process, the granulate is fluidized by hot air and dehumidified so that the conversion of mass and energy takes place [Aghbashlo et al., 2014].

In more detail, the fluid bed dryer operates on the principle of material-fluidization. The fluidization procedure requires forcing heated air or gasses across the bed of hard particles. Through crevices between these particles, these gasses or airflow will rise. Some upward dragging factors on particles rise as velocity accelerates until equaling a gravitational pull beneath. As a result, the bed is hydrated, and particles hang in it. Following is an overview of the main fluidization steps [PHARMA]:

- **Load fluidized bed dryer:** Substances can be pulled from the high-shear mixing chamber by the feeding tube, and a new batch of the wet granular is added to the products chamber using negative-pressure pumping.

- **Air Acquisition (air inlet):** The control panel turns on the blower unit. The airflow is taken consistently from the Air Handled unit and into a tower through the lower plenum after the blower is turned on.

- **Fluidization:** The fluidization phenomenon works in five different stages: streamlined fluidization, pressurized fluidization, turbulent fluidization, and initial fluidization. Inlet air is blown up through a static power bed. As the velocity of air increases, so does space among powdered particles till the material has become reprimanded in the bed.

- **Dryness:** Until the final limit is reached, the drying process is divided into three phases (at the endpoint, the solid particle's moisture level is equal to or less than 1 percent)

- **Preheating:** In the hot and dry airflow stream, wet materials are suspended. As energy moves through the body (traditional heating), humidity on the particle's surface evaporates, and the evaporation rate gradually rises as particles absorb additional heat. Although moisture loss while preheating is minimal, the overall temperature of the bed slowly rises.

- **Shake the filtering bag:** The blowers take and release airflow from the Fluidized-bed-dryer on a constant basis. Fines, or very minute particulates, may be present in the airstream. The particles are captured in filter bag pores, but this causes the dust layer to build up, which jams filter-bags and causes the pressure-drop. Mechanically shaking is the most effective approach to eliminate a dust layer, and it is carried out by the pneumatic cylinder at predetermined intervals. We have several filtration chambers, particularly 2, and shaking alternates between them.

- **Emptying of dry substances:** The evacuation of the dried materials from the fluidized bed dryer is referred to as discharge. It can be performed manually by releasing a product vessel and rolling it to the next procedure on its cart. Conversely, vacuum transporting can be done by attaching product containers to the tube and employing the vacuum transfer system to provide minus pressure for the suction. Grinding is the next step following drying.

The fluid bed drying machine has three critical parameters that characterize the efficiency of the drying process and, therefore, can influence the product's final quality. These parameters are temperature, humidity, and airflow, as presented in [Mujumda, 2012]. In theory, a higher temperature and flow rate of the inlet air to the machine implies a shorter drying time. However, each of these three parameters must be configured correctly depending on the product type to avoid quality problems and deterioration of the final product obtained after drying. It is important to note that the inlet air temperature should not exceed the critical temperature of the product to be dried so as not to jeopardize its quality or pharmaceutical properties.

2.1.2. Improvements in the drying process

A fluid bed dryer can eliminate unwanted humidity from a variety of substances, and its basic working principle is highly accurate and focuses on dehydrating materials while affecting their material characteristics. Pharmaceuticals, chemicals, food processing, fertilizer, and the dairy sectors are just a few businesses that use fluid bed dryers. For this reason, this technology adapts well to many processes in industries, taking into account physical and technical particularities in each one [Haron et al., 2017]. Different mathematical models have been studied in each industrial sector, almost always assisted by Computational Fluid Mechanics (CFD) tools. These are applied above all to the design of the geometry of the machine's distributor and the fluidized bed itself, contrasting later with the measurements of the experimental hydrodynamics. CFD introduces, for the calculation, simplifications such as the assumption of isothermal, non-reactive flow, and no mass transfer between the solid and the air, which result in a slight (although significant) deviation from the experimental results, which is situated between 7% and 15%. Among the achievements of this approach, it is worth mentioning the correct prediction of the temperature distribution of the particles in a pseudo-2D geometry of the bed, as well as the transfer of heat to the granular phase. It has also been successfully applied to calculating a gas-solid flow in a circulating bed, revealing the existence of convective flows (upward and downward) of the solid particles themselves.

In the fluid-bed-dryer process for powdered drying, airflow is drawn in from outside by the fan powered by the electric motor within the dryer. The airflow is warmed as it moves through the dryer's heating system. The wet substances deposited on shaking pierced metal beds are then forced to pass through this heated drying air. This airflow is injected at the proper speed and temperature to condense the bed, enabling every particle to have close relations with air. The granules in the bed are transported gradually all along the height of the dryer while the bed vibrates or rattles. The heated air absorbs all the moisture in a particle that flows into a dust regenerator system and is recovered for use in the process. As particles flow over the bed, airspeed and temperature may be regulated, allowing very wet viscous substances to be evaporated efficiently with a fluid bed dryer. Experimental investigations have revealed some relationships between process variables in fluidized bed drying. For example, a higher temperature increases the rate of moisture diffusion and, thus, the drying rate. A decrease in bed load positively affects diffusivity as well. The supernatant discarded is subsequently supplied into the

cooling area, where the heat of the material is reduced to a necessary level by cold air. The entire procedure is handled by an automatic system, which speeds up the process. As a result, with the assistance of the fluid-bed-dryer, material or powdered drying can be accomplished effectively. As a result, with some help of the fluid-bed-dryer, item or powdered dehydration can be accomplished effectively.

Some progress in fluidized bed technology incorporates a second heat source in the process, giving rise to a fluidized bed-assisted dryer. Said heat source may consist of a microwave oven, a solar collector, and an infrared emitter. Higher moisture reduction rates are obtained in all cases than in traditional fluid bed dryers.

One of the most recent drying methods and an intelligent way to dry medication is to employ a dependable sensor for detecting and supervising different quality parameters of materials in line, allowing it to function and detect errors or inadequacies in dryer procedures. At the same time, it customizes the tools, techniques, mineral wealth, and practices that contribute to power conservation and environmental sustainability, allowing it to regulate medicine dryer-operating conditions domestically to generate high-value medicine [Su et al., 2015]. Moisturized content, color, form, flavor, odor, or dry conditions, including suction, movement, heat, and dampness in the drying process, are all monitoring quality metrics that provide meaningful intelligence on drying system performance.

As a result, intelligent drying methods include not merely dryers, but smart and advanced sensors, translators, and the control-systems that help increase product quality and power efficiency by changing operating parameters associated with the material drying. Intelligent drying technology should be developed with an understanding of the product to be dehydrated. Beneficial ingredients of the medicine must be subjected to rigorous quality restrictions, such as local drying conditions, an effect of predetermined quality attributes, and operational parameters, in order to ensure the product's quality while using as little energy as possible and having an as little environmental impact as possible. It is worth highlighting which food's freshness should be checked throughout the procedure. To ensure the quality of finished goods, the medicine unit working should be precisely regulated and managed throughout production utilizing cutting-edge instrumentation. [Chalortham et al., 2008]. As a result, making advanced drying technology for pharmaceutical dryers necessitates a thorough understanding of the process and meticulous refinement of best operating procedures. Even though the lofty

goal is unachievable in pharmaceutical industrial-dryers real soon, correctly engineered efficient drying technology can assist in boosting productivity and output dramatically.

For more than twenty years, [Allison et al., 2015] and [Byrn et al., 2014] have advocated for developing advanced or cognitive dryers. Medicine drying is a high-energy process that has a significant impact on product quality and has a negative impact on the environment due to chemical emissions. It will be essential to make a long-term production by incorporating the most recent advancements in associated technologies, such as modern computer hardware and software and process control. It is possible to construct smart dryers thanks to recent improvements in numerical techniques, sophisticated sensors for the real-time measurements of the variables of requirement in automated dryer management, and robust control techniques. Advancements in computer technology, materials science, sensors technology, and online detection technology, as well as a better knowledge of underlying transport phenomena in medicine drying, have made it possible to achieve this goal.

2.1.3. Psychometrics and fluid bed dryer

Psychometrics is a crucial technique used to understand better how a fluid bed dryer works, and it is used in this thesis to model the preheating and drying process. Psychometrics is an area of physics dealing with the properties and processes typically of moist air (the gas phase of H2O), which can be broadened to cover mixtures of the gas of one substance and the condensation vapor of a second substance [Gatley, 2004].

In the literature, the psychrometric model has been the basis of research in data exploration and modeling for complex systems. For example, [Schoen, 2005], in a meteorological context, developed a new model of the THI (Temperature-Humidity Index), which represented a simplification of the current NWS (National Weather Service) model (3 parameters vs. 16 for the NWS model). In [Kayihan, 1985], a Monte-Carlo simulation model was developed to predict the drying behavior of lumber in batch kilns. The drying rates were approximated by a novel combination of high and low moisture asymptotic rates, which provide a simple correction procedure to compensate for the temperature and humidity variations. In [Mittal et al., 2003], an artificial neural network (ANN)-based psychrometric chart was used for real-time calculations of the air properties required in drying agricultural and food materials and ventilation of farm buildings. Two ANN were developed to predict psychrometric parameters. In [Simões, 2019], mathematical models were developed for the psychrometric chart. The aim was to

identify and model dynamic mathematical relationships between psychrometric properties. Theoretical and empirical models were compared, the latter using a two-layer neural network as a transfer function for the relative humidity of the air.

2.1.4. Energy consumption advances in fluid bed dryer machines

The biggest challenge when using a fluid bed dryer is to reduce the enormous amount of time and energy the machine takes to complete the process. Following the electric energy crises of the 1970s [Lifset, 2014], electricity consumption became a topic of discussion. Furthermore, it has been established that global electric energy use is quickly expanding [Boyd, 2013], specifically in the pharmaceutical industry, which is a growing field nowadays. As a result, every pharmaceutical company seeks to utilize as little electric energy as possible in many sectors, such as manufacturing fields, packing industrial processes, and transportation to different hospitals or medical stores [Thomas, 2006]. Because power energy originates from three sources: coal and oil fuel, solar energy, and nuclear power energy, keeping track of these forms of energy use in various areas takes much effort. Nevertheless, in doing so, we can forecast the quantity of electric energy utilized in various medicine manufacturing processes and attempt to devise strategies that are tailored to a specific use and domain.

Predicting electricity utilization is very important for decision-making and policymakers for all the pharmaceutical industry energy-taking machines. We can conceive of improvements to pharmaceutical manufacturing processes or works to lessen the quantity of electric energy consumed if we understand how much electric energy will be utilized. Predicting future electric energy utilization in the pharmaceutical manufacturing industries, both in the short-term and long-term, would enable us to understand where we can save energy in the pharmaceutical manufacturing process and how we can reduce the current consumed energy. Moreover, types of electric energy are most often used and attempt to change the trend, as has happened in recent years with coal and oil and now with solar energy. Various elements, including such processing time of medicine, weather, and climate, affect the quantity of electric energy utilized in different companies and manufacturing steps of medicine. With many variables, estimating energy usage is a problematic manufacturing task [Mujumdar, 2014].

Machine learning models are currently employed in various fields since they are beneficial. Machine learning operates similarly to the function that nicely maps the input data to the output. Machine-learning models can give high-accuracy predictions for energy usage in the pharmaceutical process or the heating process in the manufacturing process. As a result, pharmaceutical companies can use them to enact energy-saving initiatives in different manufacturing domains. For example, machine learning algorithms can forecast how much electric energy is utilized in a dryer machine in manufacturing [Aghbashlo et al., 2012]. They can also be used to forecast the future-energy consumption, such as power or organic gas [Ghasemi-Varnamkhasti et al., 2014]. This will be presented in more detail in the following chapters.

2.2. Industry 4.0 in the pharmaceutical industry

Industry 4.0 is the fourth industrial transformation that combines different fast-growing technologies, for example, internet-of-things (IOT), intelligent systems (AI), autonomous robotics, and sophisticated computation, to alter the production environment drastically, also called digital twins. Related, independent, and self-organizing manufacturing industries are the characteristics of Industry 4.0. To achieve Industrial Revolution 4.0 for medicines and overcome the conservatism of conventional pharmaceutical manufacturing architecture, procedures, and regulations, the latest ways of planning will be necessary. Whereas required to implementation of many advanced advance technologies and mass production methodologies required to facilitate Industry 4.0 may be difficult, it may be meaningful because they offer the potential for maximum output, maximum manufacturing protection, increased quality, improved value, maximum agility, great flexibility, and minimum wastage with high efficiency [Ezell, 2016], [Buvailo, 2018] and [Tilley, 2017].

2.2.1. Industrial revolution in pharmaceutical manufacturing

Industry 1.0: If Industry 4.0 is the future, then Industry 1.0 is the contemporary pharmaceutical industry. Herbals or organic remedies have been used as medications since the dawn of human civilization. The way materials are handled and prepared for

medicinal uses has changed dramatically in the last three centuries. Mechanical processes of the botanical, minerals, and animal-derived substances progressed from ordinary hand-operated instruments to commercial-scale equipment capable of crushing, milling, blending, and pressing more significant amounts of medications in Industry 1.0. [Anderson, 2005]. Independent pharmaceuticals and the chemicals business [Sonnedecker et al., 1976], [Daemmrich et al., 2005] were two sources of larger-scale medication manufacturing using non-electric power-driven equipment in the 19th century. This shift from small-scale to large-scale medication manufacturing drove the formation of the pharmaceutical industry in the nineteenth century, which experienced phenomenal expansion over the previous century. Nevertheless, a few early machineries from the first industrialization, including pneumatic grinders and tablet presses, are also routinely employed nowadays [Barriga and Hassan, 2019].

Industry 2.0: Electrical and earlier electronics equipment and manufacturing processes having preset controllers which combined total mechanization and procedure management gave manufacturers the capacity to specify fundamental processing parameters enabling the second-generation industry revolution. This established itself in pharmaceutical manufacturing companies as electronics machine-based smashing, grinding, mixing, and tablet pressing, enabling larger-scale productions and more significantly, better procedure and quality control. On the other hand, process controls were typically limited to predetermined and static configurations that only permitted for vigilance systems and passively controlled measures. Sophisticated pills press which can consistently generate hundreds of thousands of tablets per minute [Nashet al., 2003], are examples of Industrial revolution 2.0 innovations. Consequently, most of the modern pharmaceuticals manufacturing businesses might be said to continually be operating under Industry 2.0 framework [Lorenz et al., 2018].

Industry 3.0: The developments and affordability of computers and their communication techniques, including network computing, world wide web, and intra-wireless transmission, permitted the industrialization of revolution. These innovations allow for greater automated procedures and equipment, enabling principles like the continuous-manufacturing and active control in pharmaceutical companies' manufacturing. Human-computer interactions facilitated the development of more complex control techniques and improved product and process quality. Remote sensing and monitoring eliminated a need for human operators on factory floors and allowed for improving tracking of production factors and KPIs.

Several businesses have already transitioned to Industry 3.0, while the pharmaceutical industries are still in the early stages. Continuous manufacturing, for illustration, is a method that continually transmits components created throughout every process stage to the next phase for more processing; it has been extensively implemented in various businesses. The pharmaceuticals sector needs to be more active in implementing continuous production for a variety of purposes [Lee et al., 2015]. As a result, unlike other companies, the pharmaceutical business has failed to consistently reach the six-sigma production capability (For example, 3.5 errors per thousand possibilities) [Yu et al., 2017]

The third industrial revolution brought improved process analytic technology (PAT) to pharmaceutical manufacturing, intending to provide processes and products quality information in nearly real-time. The Models-based or Qualities by the Design (QbD) procedures, which strive to regulate desired product quality characteristics inside predetermined quality criteria, were also upgraded in Industry 3.0. Nevertheless, to realize the full possibilities of the PAT and the QbD, extra technological improvements are required to gain profound processes knowledge and real-time predictive analysis, allowing for further widespread, meaningful release testing by high grades of products quality-assurance – particularly for the biotechnology products. Considering the quality difficulties accounting for over two-thirds of medicine constraints, it is evident that more work is needed to strengthen process control and reliability [FDA, 2019]. Regardless, Industry 3.0 allows for a much better grasp of acquiring, analyzing, and safeguarding enormous quantities of information in the pharmaceutical production process.

Industry 4.0: The 4th industrial revolution combines sophisticated manufacturing technology to create interconnected, autonomous, and self-organizing production systems which work without human intervention. The knowledge gathered in automatic and digital environments of Industry 3.0 paves the way for the general shift to Industry 4.0 in pharmaceutical production. Unlike Industrial 3.0, which saw significant advances in specific applications and instruments, Industry 4.0 promised enhancements in complete manufacturing infrastructures and applications. The productivity of data can be examined by various algorithms and uses for simple vital operations and the business considerations which directly affect the manufacturing outputs in such an atmosphere [Fuhr et al., 2014].

The journey from simple data collection to digital maturity consists of converting raw data collected during manufacturing processes to knowledge obtained via data analytics tools. This "intelligence" is just what drives the self-optimizing, judgmental, autonomous

movements, and controller design in the autonomous designs and the cyber-physical machinery (for example, computers with processes operated by the computer algorithms) [Guilfoyle, 2018].

The emergence of Industry 4.0 compels us to imagine how the completely digital and autonomous manufacturing environment might appear alike and how it might affect pharmaceutical procedures and laws [Leurent et al., 2018] [Moore, 2018]. As a result of digitization, automation, and real-time data aggregation, the latest operational paradigms will emerge in pharmaceutics, allowing for more significant than the six-sigma reliability for both tiny and big molecule therapeutic goods. The COVID-19 global medical crisis has brought attention to the need for production technology that can adapt to shifting demands and reduces reliance on human involvement. In the face of difficulties that prevent individuals from working in conjunction with everyone else, automation and robotics-based procedures may be essential. For a manufacturer, the consequence is a well-controlled, hyper-connected, digitized environment and pharmaceuticals value chain [Markarian, 2016]. Figure 3 shows a summary of the fourth industrial revolution commented.

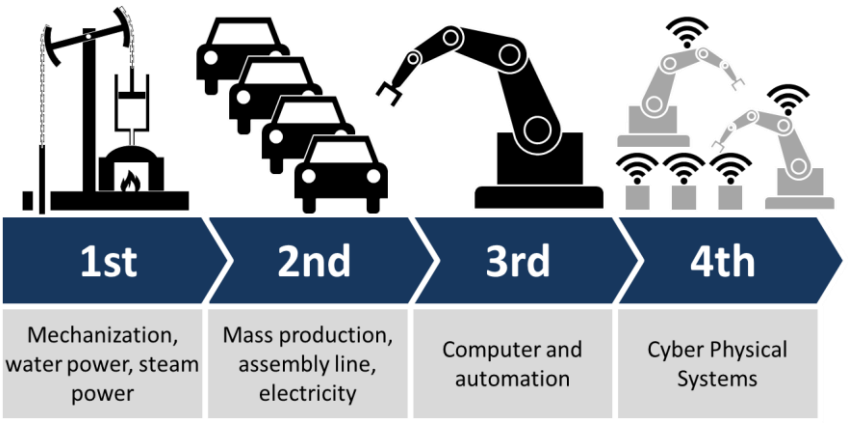

Figure 3. By Christoph Roser at AllAboutLean.com under the free CC-BY-SA 4.0 license.

2.2.2. From Industry 4.0 to Pharma 4.0

The ICH (International Council for Harmonization of Technical Requirements for Pharmaceuticals for Human Use) guidelines established the pharmaceutical industry 4.0 as the new paradigm where new production processes, equipment, facility design,

logistics, and operational concepts are integrated. Early collaboration of all departments in the pharmaceutical industry (quality assurance, quality control, process development, manufacturing operations, engineering, automation, and information technology) is required to design robust, flexible structures that operate at the level of quality required by a changing market with a strong demand such as the pharmaceutical.

According to [Herwig et al., 2017], the current objective is to achieve a control strategy based on the global or integral vision of the process. For this, these authors identify a new need in the industry: design towards data integration. This encompasses process maps, data process maps, and data flows one step ahead of the current flow charts or process charts used in business today. The new design tools for data integration will entail implementing and controlling each change, physical or operational while ensuring the inclusion of key human factors such as knowledge and experience stored by the company's staff and workers' critical thinking. Therefore, prior knowledge must be present in one form or another.

Likewise, the strategy based on the global vision will require the transversal confluence of all the organization's departments that, combined with data science and information technologies, will give rise to the goal of data integration. Integrating all computerized systems is critical to achieving this goal regarding data terminals and concepts such as PAT technologies and RTRT (real-time release testing) to achieve continuous manufacturing [Lourenço et al., 2012]. Companies that have adopted this philosophy have established a single significant data source for the entire company, using Big Data infrastructures, which they also use for real-time data acquisition and decision-making support.

Regarding the adaptation of industry 4.0 to Pharma 4.0, the following characteristics must be implemented:

- **Work instructions for manufacturing.** The master operations record will continue to be used, but to enable flexible execution, flexible strategies and predictive maintenance are implemented. ICHs are promoting this type of planning.

- **Quality and compliance with specifications.** Modern Pharma 4.0 establishes quality assurance through transversal management of business resources, connecting organizational tasks and functions.

- **Execution.** To ensure economic efficiency, data must be evaluated, analyzed, and used for process optimization to a level of complex treatment that only big data can provide. Moreover, generating results at such a speed that the downtime and effort of optimization can be reduced.

- **Integration Plug & Produce.** The characteristics of these new enormous data flows between equipment and cloud data centers will allow equipment and machines that were already working to adapt to the new way of functioning, like a digital twin, while facilitating the connection of new era equipment, making the Plug & Produce concept a reality. In the future, the flexibility of the industries will make this type of interconnectivity possible, reducing costs and minimizing the changes to be made in the production lines.

In this thesis, we have used a digital twin approach by integrating real-time data from a fluid bed dryer to a machine learning algorithm, creating a digital model that can predict when the process needs to be stopped by the operator, with the consequent energy saving.

2.2.3. Digital Twin technology and Industry 4.0

A Digital Twin is a virtual representation that serves as the real-time digital counterpart of a physical process [Barricelli et al., 2019] [Boschert et al., 2016]. Digital twins are the outcome of continuous improvement in the creation of product design and engineering activities. Digital Twins can be considered the current leading edge in the evolution of design and simulation tools: from handmade product drawings and engineering specifications to computer-aided drafting/computer-aided design (CAD) to model-based systems engineering (MBSE). The digital twin of a physical process depends on the digital thread—the lowest level design and specification for a digital twin—and in order to preserve accuracy, the "twin" depends on the digital thread. With applications like real-time system monitoring and control using Process Analytical Technology (PAT), continuous data acquisition from equipment, intermediate, and final products, and continuous global modeling and data analysis platforms [Lourenço et al., 2012], digital twins are playing an increasingly significant role in pharmaceutical and biopharmaceutical manufacturing systems [Chen et al., 2020], [Cheng et al., 2020]. Moreover, Digital twins have recently been identified as a critical approach in Industry

4.0 [Lourenço et al., 2012], and the European Union is financing a diversity of projects to develop digital twins in different fields, such as SPIRE industries and biomedical applications, among others. In [Fornasiero et al., 2021], a survey was recently conducted on the degree of implementation of Artificial Intelligence and Big Data systems in the "process industries" in Europe. Machine learning and predictive maintenance were found to be key fields and the most popular to be implemented in process industries, as commented by [Park et al., 2018] and [Ali et al., 2021]. Also, Cyber-Physical Systems were found to be an important framework being adopted by industries, especially for proactive or predictive maintenance solutions and tools [Shcherbakov et al., 2020]. Other solutions involving Big Data, user data management, and data processing methodologies are shown in [Shafqat et al., 2020] and by using digital twins in [Burggraeve et al., 2013]. It is also noteworthy to mention the use of data visualization for value analysis [Colombo et al., 2020]. Data-driven models trained using machine learning algorithms have recently been developed, as commented in [Liu, 2022], and also for specialized industrial processes requiring high precision. In this sense, micro-pull winding and laser ablation processes have been modeled and simulated to find the optimum control parameters for a required production specification [Wasiak et al., 2017] [Nettleton et al., 2016].

Figure 4 shows an overall depiction of the digital twin concept we have implemented in this thesis, with the physical system on the left, including the fluid bed dryer, the data-driven simulator on the right, and the SCADA/Cloud interface in the center [Barriga ISPE, 2022].

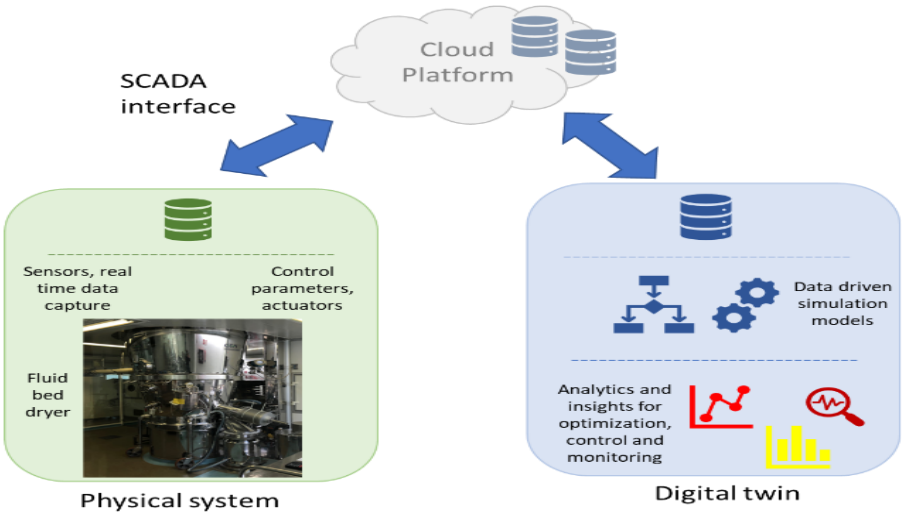

Figure 4. Schematic representation of the physical process and digital twin

Data mining and artificial intelligence techniques reviewed in the literature have been implemented in different applications including drying processes in the pharmaceutical industry [Peterson et al., 2019], [Keskes et al., 2020] and [Petrović et al., 2011]. In this thesis, different AI techniques have been evaluated to reduce pharmaceutical processes' high costs in terms of energy consumption.

2.3. Machine Learning applied to manufacturing

The usage of machine learning in the manufacturing industry represents many benefits and advantages in terms of efficiency and improvements. Machine learning in manufacturing has a number of immediate advantages, including improving operational efficiency, lowering energy or raw material costs, reducing maintenance costs, reducing inventory levels, improving quality control on production lines, and reducing waste or improving safety, among other benefits [Barriga and Hassan, 2021]. In this chapter, we will briefly explain an introduction to the machine learning process approach, types of machine learning algorithms, and some related work papers reviewed related to energy reduction in manufacturing.

2.3.1. Introduction to machine learning

Machine learning, also known as artificial intelligence, allows computers to acquire knowledge and progressively improve tasks' performance and data analysis. It presents an exciting way of generating learning based on the information patterns extracted from the data analyzed. By taking the data's behavior, we can create predictive models designed for decision-making with a considerably high efficiency. The machine learning process can be broken down into seven major steps:

Objective definition: The main purpose of this step is to understand the problem to be solved. For example, in our thesis, the main problem to be solved is to improve the pharmaceutical manufacturing drying process by reducing the energy consumption of a fluid bed dryer.

Data collection: This step aims to collect reliable data so that the machine learning model can find the correct patterns. The quality of the data used for the algorithm will determine how accurate the model is. If data is incorrect, the model will return wrong predictions. Good data contains few missing and repeated values and a good representation of the various scenarios to be analyzed. In this thesis, we will use data directly extracted from a fluid bed dryer collected by their sensors.

Data preparation: Data preparation is one of the most effortful phases of machine learning. It includes data cleaning tasks like detecting and fixing, when possible, incomplete data sets or normalizing data to put the same format or scale. For example, in our thesis, we detected that some sensors were generating data related to the mixing process and not impacting the drying process, so we decided to eliminate it.

Data understanding: Understanding the problem is as important as understanding the data we have available. EDA, exploratory analysis of data [Cox, 2017], is the technique used by data scientists to make analysis, graphs, correlations, and descriptive statistics to understand better what story the data is telling us. It also helps to estimate if our data is sufficient and relevant to build a model. This step usually requires more time and effort in building a machine learning project.

Model building: In this step, we will select the machine learning model/algorithm that fits our objective and data set. The machine learning algorithm will automatically learn to obtain the relevant results with the historical data we have prepared, also called the training data set. There are various machine learning algorithms: predictive, classification, linear regression, clustering,
and Deep Learning, among many other variants. The next section will define the most common models/algorithms/techniques used.

2.3.2. Machine learning algorithms

Machine learning algorithms are classified into several that are described in Figure 5 [Ghori et al., 2020].

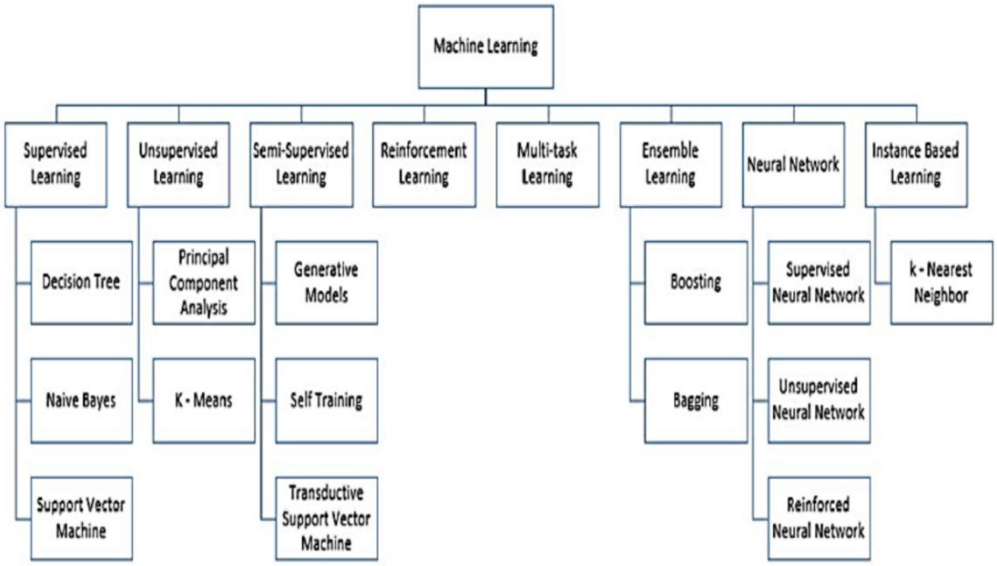

Figure 5. Machine learning and its types. [Ghori et al., 2020]

Supervised Machine Learning: Supervised machine learning algorithms require external assistance to carry out their operation. The input dataset is split into training and test sets, whereby the training dataset has an output variable that needs to be classified. [Kotsiantis et al., 2007] Reported that all types of algorithms learn input data configuration from the training set and apply it to the test set for classification or forecast. The three most common supervised machine learning programs are Naïve Bayes [Lowd et al., 2005], Decision tree, and Support vector machine.

Unsupervised Machine Learning: This machine learning type is employed to draw valuable insights from datasets with unlabeled input data [Meyer, 2004]. The machine-learning algorithm uses the formerly learned features to detect the data class. This technique is mainly used for feature reduction and clustering. K-mean clustering and Principal Component Analysis (PCA) are the two key unsupervised learning algorithms.

Semi-Supervised Machine Learning: This technique combines the strength of supervised and unsupervised. It is successful in both data mining and machine learning, where unlabeled data is available, and getting these data labeled is a tedious process [Zhang et al., 2003]. Many categories of semi-supervised learning algorithms, such as self-training, generative model, and transudative support vector machines, are described by [Zhu, 2005].

Reinforcement Learning: This type of machine learning makes decisions based on which exact actions to take to ensure a more positive outcome. This learning is carried out without prior knowledge until given a particular scenario. Reinforcement learning mainly depends on two conditions: delayed outcome and trial and error search [Zhu et al., 2009].

Multi-Task Machine Learning: Multitasking learning is a type of machine learning that helps other learners to perform excellently better. Applying multistage learning to a task retains task procedures, how it solved the problem, or how a certain insight was deduced. The algorithm follows these steps to solve similar problems. This learning system can also be termed an inductive transfer learning mechanism [Sutton, 1992] [Dey, 2016].

Ensemble Learning: This is a combination of various individual learners to form one learner. It was disclosed by [Chaudhary, 2019] that a combination of several learners is way better at doing a particular task than an individual learner. The two most popular ensemble learning activities are boosting and bagging.

Instance-Based Learning: In this type of machine learning, the algorithm learns a certain type of arrangement. This same arrangement is then applied to newly fed input data. This type of learning waits for the test set to be available and then acts on it together with the training set. The bigger the size of the data, the higher the complexity of the algorithm. One common example of instance-based learning is K-Nearest Neighbor [Opitz, 1999].

Neural Network Learning: Neural Network learning is derived from the biological theory of neurons. It is also called the Artificial Neural Network (ANN). Neurons have a cell-like structure in the brain. The neural network imitates the working principle of a human brain. ANN can be used in data mining, expert systems, medical, fuzzy logic, business, weather forecast, aviation, and computer science. Some notable ANN advantages are real-time operation, adaptive learning, pattern recognition, self-organization, amongst others.

2.3.3. Machine learning applied to energy reduction in the manufacturing industry

Several studies demonstrate the high applicability of machine learning techniques in the pharmaceutical industry. [Aksu et al., 2012], performed a systematic study of the application of ANN to developing and formulating pharmaceutical products in *Quality by design approach for tablet formulations using artificial intelligence techniques*. Using historical data, they could infer detailed information on the interactions between the formulation and the specifications of various drugs. In the conclusions of their essay, they assured the efficiency of neural networks and genetic algorithms for the optimization of formulations, reducing energy consumption.

Our thesis focuses on applying artificial intelligence algorithms to improve methods and processes in the manufacturing industry, particularly in drying processes. We have found stimulating studies such as the one published by [Ugur et al., 2008] in which they present a simplified physical model of the drying phenomenon of solid particles and approach the solution through genetic algorithms. For this, the authors establish the predictive control model, in which air temperature and air humidity are taken as control variables, while are taken as prediction variables those of the moisture content and the quality (a substance that disappears while the process continues and that, in our case, would be the active ingredient of the drug). They found that the training process converged reasonably and that the obtained drying times improved those obtained in the laboratory. Although it incorporates classical physical parameters, this modeling is less ambitious than the one we implemented since we study the evolution of a more complex system formed not by a simple particle (whose behavior is described by the starting equations). In our case, the system to be studied is made up of N solid particles that move randomly through the fluidized bed dryer, traversed by a hot air current that moves between the particles. Although the substance to be dried in this study to which we refer differs from the one we handle in our thesis, and the physical process is significantly less complex, it gives us an idea of which methods of machine learning can optimize highly complex problems that would otherwise be difficult to tackle.

Another research reference was carried out by professors [Nazghelichi et al., 2011], from the Faculty of Agricultural Engineering of the University of Tehran. They focused on the energetic aspects of the fluidized bed drying process. They also had a machine in which different tests were carried out for their experiment, and they trained the model against

experimental data. However, their results cannot be extrapolated to our work because it is a different machine, and the tests are carried out with different materials. Despite the differences, this research brings two interesting novelties: they tried to measure the influence of many variables in their study. Thus, they changed the temperature of the inlet air (50ºC, 60ºC and 70ºC) in the thickness of the bed (30, 60 and 90 mm) and the drying time. In total, 518 tests were performed, of which 259 were used for training and 259 for validation.

2.4. Introduction to Cloud Computing and IIOT

2.4.1. Cloud Computing IaaS, SaaS and PaaS

Cloud computing offers the possibility to consume computing services such as servers, storage, databases, networking, software, analytics, and intelligence over the Internet to offer flexible resources and economies of scale. Rather than owning a computing infrastructure or data centers, this can be rented to access applications, storage, or infrastructure from a cloud service provider. Utilizing cloud computing services has the advantage of avoiding the upfront costs and complexity of purchasing and maintaining one's own IT infrastructure in favor of paying for it only as it is utilized. Cloud computing service companies can gain enormous economies of scale by offering the same services to a wide range of consumers. Manufacturing plants can use a cloud computing solution to handle the 'big data' associated with manufacturing operations and complex computational capacities in a secure, protected environment. In addition, when different systems are running on the cloud, they can be synced to communicate automatically. Cloud computing can be broken down into several different elements, focusing on different parts of the technology stack and different use cases such as IaaS (Infrastructure as a Service), SaaS (Software as a Service) and PaaS (Platform as a Service). Each cloud service model (IaaS, SaaS and PaaS) gives a range of control, which corresponds to a range of responsibility as shown in Figure 6. A SaaS system is completely managed by the service provider, and some configurations can be changed. IaaS gives complete control because the infrastructure is rented, not owned. PaaS solutions allow the service provider handles everything else except the application and data.

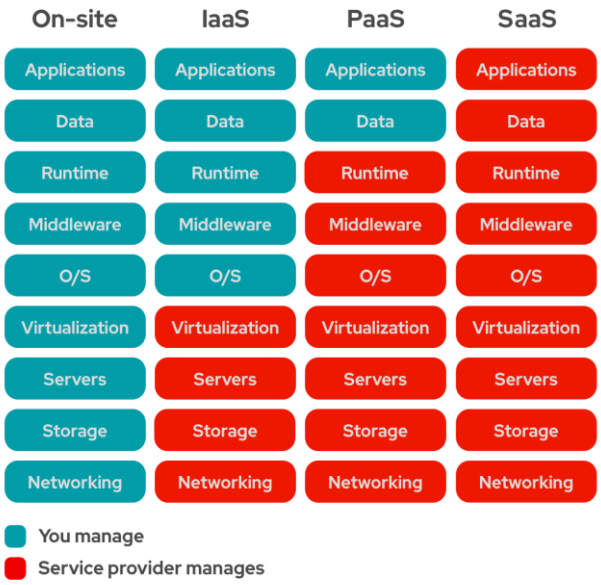

Figure 6. Service models in cloud computing. IaaS, PaaS, SaaS

As the term "infrastructure" suggests, IaaS architecture contains all the physical data centers to support your application and your servers on virtual machines to provide a virtual data center [DNS] and [EGINN]. This allows outsourcing all network and computing needs to a cloud architecture. This includes operating systems, databases, development tools, and other middleware or applications, which can enable running essential operations like building web applications, hosting websites, developing, and testing new environments, and running high-performance computing like machine learning algorithms. IaaS gives virtualized resources such as servers, disks, networks, and IP addresses, but they are not responsible for administering the operating system, data, applications, middleware, and runtimes. IaaS allows the freedom to buy only the compute capability required and scales those resources up or down as necessary. Besides, there are other advantages of using IaaS, such as a range of hardware configurations with pre-configured operating systems like Linux or Windows are available from cloud service providers. It also enables dynamic scaling – add capacity during peak times, scale down as needed and eliminate the need for large investments. The "pay as you use" pricing models offered by cloud service providers allow to only pay for the resources utilized, lowering costs. Billing stops when a virtual machine stops. As opposed to standard flat/fixed prices, they only charge for the actual usage in this situation, which results in significant cost savings. Provisioning and deploying resources are very simple and global infrastructure with edge locations is available worldwide.

There are some disadvantages of IaaS. Given that the service provider controls the infrastructure, outages in the infrastructure can affect the customer infrastructure. Because IaaS users have limited access to the cloud service provider's infrastructure, troubleshooting is more challenging. Besides, if peak usage is high, monthly costs may be much higher than expected, and the provider may share infrastructure across multiple clients. This adds to the security risk when working in a highly regulated industry such as the pharmaceutical industry.

PaaS (Platform as a Service) offers a platform for developing and deploying applications. The technical stack required for application development is available on the cloud, which requires no download or local installation. With PaaS, developers can concentrate on creating their apps rather than worrying about infrastructure, storage, software upgrades, or operating systems. As a result, programmers may create, launch, and manage their own apps quickly and easily without having to construct and maintain the infrastructure or platform that is often required for the process. PaaS apps adopt certain cloud features, making them scalable and highly available. PaaS can also be beneficial when needed to create and deploy applications quickly or if it is needed to streamline workflows when multiple developers are working on the same development project.

A web-based software deployment approach called software as a Service (SaaS) makes the software accessible through a web browser. The main benefit is that it is not relevant where the software is hosted, which operating system it uses, or which language it is written in. The SaaS software is made accessible from any device with an internet connection. Capital expenses of purchasing servers or software while using SaaS are avoided. You only need to connect to the SaaS application using a console dashboard or API because the service provider is shielding you from software maintenance. Microsoft Office 365, Intuit, Salesforce CRM, Zoom, ZoomInfo, Dropbox, Google Apps, and many more products geared toward end users are typical examples. These applications run on the cloud and need not be downloaded to a local device. Webmail such as Outlook, Gmail, Yahoo, etc., is one of the earliest forms of SaaS [EGINN].

2.4.2. Edge computing

Edge computing refers to computing done at the location closest to a system's data source where information is coming from or going. Edge architecture allows processing to occur more quickly by reducing latency. Applications and programs running at the

edge can work quicker and more efficiently than a cloud computing architecture, resulting in a better user experience and improved overall performance. For example, if you're a manufacturing company, your "edge" infrastructure can be located close to the production lines where row material is loaded, for collecting and managing information. Your manufacturing cloud computing solution could be located miles away, housing the main datacenter, but the edge is where the app-processing action is. Edge computing in manufacturing uses sensors, communication, and data processing technologies to interconnect many components [REDHAT]. For instance, the data generated from sensors from production lines located on the shop floor needs to be uploaded to the cloud computing layer, and the routing strategy will directly affect the delay performance. Edge computing refers to a new computing model that analyzes and processes a portion of data using the computing, storage, and network resources distributed on the paths between data sources and the cloud computing center. Edge computing uses devices with sufficient computing power to implement local preprocessing of source data [Qiu et al., 2020].

Factors	Cloud Computing	Edge Computing
Computing architecture	Centralized	Distributed
Server node location	Edge network	Data center
Transmission bandwidth load	High	Low
Energy consumption	High	Low
Data processing	Slow	Fast
Latency	High	Low
Real time	Weak	Strong
Security	Low	High
Reliability	High	Low
Computing resources	Unlimited	Limited
Computing cost	High	Low
User experience	Weak	Strong

Figure 7. The comparison of cloud computing and edge computing. [Sun et al.,2020]

Figure 7 shows the comparison between cloud computing and edge computing. Edge computing is an extension of the concept of cloud computing, which cannot completely replace cloud computing. The relationship between edge computing and cloud computing is collaborative and complementary. The edge ends can analyze and process a large number of real-time data quickly, but most of the data is not only used once. Even after the edge-end processing ends, it needs to be collected from the edge end to the cloud. The mining and analysis of massive data, the storage of key data and the linkage

of multiple edge nodes all need to rely on the cloud, and the virtualization resources and management of the edge also need to be completed by the cloud. When edge computing and cloud computing work closely together, they can achieve different demand scenarios, thus maximizing the application value of edge computing and cloud computing [Sun et al., 2020l]. Figure 8 shows a basic cloud computing and edge computing architecture for manufacturing. The factory network contains production lines sending/receiving data to/from edge computing infrastructure located in on-premise servers at the same physical location. On the other hand, edge computing is connected to cloud computing infrastructure located outside the manufacturing facilities.

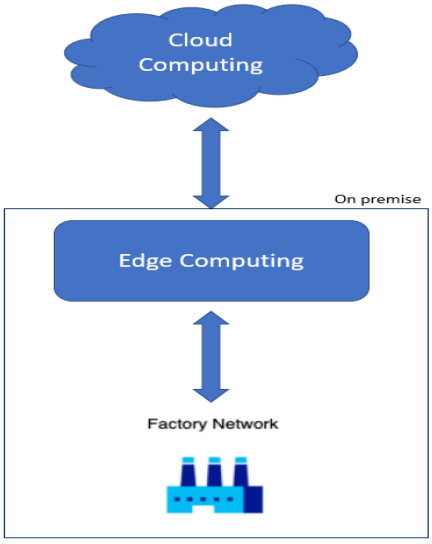

Figure 8. Cloud computing and Edge computing architecture.

2.4.3. Industrial Internet of Things (IIOT)

Industrial internet of things (IIOT) is the use of smart sensors to improve manufacturing and industrial processes. IIOT uses the power of cloud computing and edge computing to perform Utilizing real-time analytics to benefit from the data that "dumb machines" have been producing in industrial settings for years. Connected sensors enable manufacturing plants to quickly pick up on inefficiencies and problems and save time and money while supporting business intelligence efforts. The IIOT has enormous potential for improving supply chain efficiency, traceability, sustainable and green manufacturing methods, and quality control. In an industrial setting, IIOT is key for improving processes such as predictive maintenance, enhanced field service, energy

management and asset tracking. IIOT is a network of devices or sensors connected to a system that monitor, collect, exchange, and analyze data. These sensors transmit information directly to the data communications infrastructure, edge computing or cloud computing, or both, where it is transformed into useful knowledge about how a specific piece of equipment is working. This data can be utilized to improve manufacturing procedures and for predictive maintenance.

There are some differences between IOT and IIOT. Cloud platforms, sensors, networking, machine-to-machine communications, and data analytics are just a few of the technology that both use but they are used for different purposes. Applications for the Internet of Things (IOT) link devices in a variety of industries, including agriculture, healthcare, consumer products, utilities, and government and urban areas. Smart appliances, fitness trackers, and other IOT applications typically don't cause emergencies if something goes wrong. On the other hand, IIOT applications link machines and gadgets. mainly in manufacturing industries. IIOT implementations can lead to high-risk situations as a result of system failures and downtime. IIOT applications are likewise more focused on increasing productivity. versus the user-centric nature of IOT applications.

2.4.4. Communications gateway OPC UA

Open platform communications unified architecture OPC UA is a standard that guarantees the open connectivity, interoperability, security, and dependability of cloud and edge computing systems as well as industrial automation equipment. OPC UA is widely recognized as the key communication and data modeling technology for Industry 4.0 projects connecting manufacturing production lines with software capabilities. The OPC UA standard is driven by the OPC Foundation, a non-profit organization to facilitate multi-vendor, multi-platform, secure, and reliable interoperability. Manufacturing automation consists of different controllers and devices from different providers or vendors with different protocols. These controllers and devices are essential to communicate with management systems (Enterprise Resource Planning, Manufacturing Execution Systems, etc.). OPC UA, therefore, creates an environment for accessing real-time plant floor data from these vendors. It also offers "plug and play" connectivity from proprietary devices and acts as an interface between various data sources such as PLCs (Programable Logic Controllers) and field devices, sensors and actuators; applications

such as the SCADA (Supervisory Control And Data Acquisition system). This technology is implemented in server/client pairs.

On the one hand, the hardware communication protocol used by a PLC is converted by a software program called the OPC server. In comparison, client software is any program that needs to connect to the hardware. On the other hand, the client uses the server to get data or send commands to the hardware. OPC is valuable because it is an open standard, which results in cheaper prices for producers and more options for consumers. To enable communication between their devices and any OPC client, hardware manufacturers just need to supply a single OPC server. Software vendors need only OPC client capabilities in their products, and they instantly become compatible with thousands of hardware devices. Ultimately, users can choose any client software they need, safe in the knowledge that it will communicate seamlessly with their OPC-enabled hardware, and vice versa [MSC].

2.5. Microsoft Azure Cloud Computing platform

Several cloud computing platforms are available in the market, like Amazon Web Services, IBM Cloud, Google Cloud or Microsoft Azure. [Muhammed et al., 2020] compared the big three, using the constraints of hubs, analytics, and security. The study also recommends which IOT cloud platform vendor is ideal.

MS Azure is the platform that the company where the fluid bed dryer is allocated is using as a cloud computing corporate solution. Microsoft's cloud services and resources can be accessed and managed using Azure [AZURE], a framework for cloud computing and online portal. These resources and services may keep and modify our data based on our needs. To use these tools and services, we need an operational internet connection and the ability to log in to the Azure site [Gundu et al., 2020]. MS Azure was launched on February 1, 2010, longer ago than its largest opponent, Amazon Web Services. It's free to sign up and follows a pay-per-use model, which means that you pay only for the services you use. Besides, Azure supports a number of programming languages, including Java, Node js, C# and Phyton, the language we use for our machine learning algorithm. Around 200 services are available on Azure, categorized into 18 different groups. Computing, containers, networking, storing, the Internet of Things, analytics,

mobile, artificial intelligence migration, machine learning, databases, security, developer tools, media identification, management tools, and web services are just a few. Following is a list of some of the key Azure services we will use for our work.

2.5.1. MS Azure Databricks

The Apache Spark implementation on Azure is called Databricks. Large data workloads may be processed utilizing fully managed Spark clusters, particularly helpful for data engineering, exploration, and machine learning-based data visualization. The team behind Apache Spark and Microsoft collaborated to create Databricks. It offers a unified platform for big machine learning and data processing for analytical, engineering, and data science teams. The Apache-Spark environment provided by Databricks is quick and efficient and enables large-scale data processing for batch and streaming applications. Databricks is one of the most prominent platforms you can use to deal with big data and perform collaborative tasks in the Data Science field. We will use Databricks to store the dataset in the data lake and create pipelines to integrate the data sources with the platform. We will use AI/ML module to analyze the data and make predictions after training models. Figure 9 shows a summary of the different components of MS Azure Databricks.

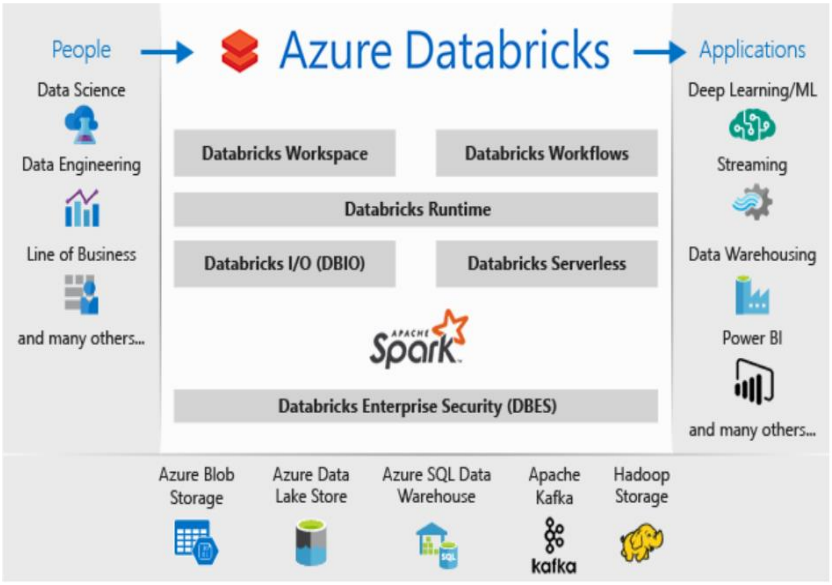

Figure 9. MS Azure Databricks components. Source Microsoft.com

Data scientists, Data engineers, and machine learning engineers may work in an interactive environment due to Databricks engineering and data science. Sometimes,

Databricks Data Science & Engineering is referred to as Workspace. It is an Apache Spark-based analytics platform. Apache Spark cluster technologies and capabilities are fully open-source and included in Databricks Data Science & Engineering. A managed service for experiment tracking, feature development, model training and maintenance, feature and model serving is comprised in the integrated end-to-end machine learning platform called Databricks. The development of a well-suited collection for ML is automated using Databricks machine learning. The most renowned machine learning libraries, including PyTorch, Keras, TensorFlow, and XGBoost, are contained in Databricks Runtime machine learning clusters.

Databricks machine learning allows us to:

- Develop models manually or using AutoML. In our thesis, we will develop the model manually.
- Training variables and models are monitored by using MLflow tracking experiments.
- Produce feature tables used for model training and inference.
- Utilize Model Registry to share, oversee, and provide models.

Databricks machine learning is a crucial service that will enable us to register over the workspace and dataset and then train our machine learning models on our dataset.

2.5.2. MS Machine Learning

A cloud-based service for building and administering machine learning solutions is called Azure Machine Learning (Azure ML). It is intended to assist data scientists and machine learning experts in utilizing their current modelling and data processing abilities. Assist them in scalability, workload distribution, and cloud deployment as well. Classes are available in the Azure ML SDK for Python that we may use to interact with Azure ML in our Azure subscriptions. It enables us to build, manage, deploy, test, or keep an eye on machine learning models in a scalable cloud setting. Many open-source Python packages, including TensorFlow, Matplotlib, and scikit-learn, are supported by it. This module enables us to build, test, deploy, manage, or monitor our ML models. Models will create, train, and deploy by using a few different tools from Azure machine learning [AZURE]:

- **Azure Machine Learning for Visual Studio Code Extension:** It is an add-on for free that enables resource management, process modelling for deployments and training in Visual Studio Code.

- **Jupyter Notebooks:** We may generate and transfer documents with live code, graphics, narrative prose, and mathematics with this open-source web application.

- **Azure Machine Learning Studio:** You can design, develop, and train machine learning models in this workspace.

- **Model Registry:** It is a machine learning service where the model is stored once trained. A model registry is responsible for keeping records of the models being built and trained. The versions and names of the models can be used to identify them. The registry service records each new model registered with a name used as a previous version. The model's name is left unaltered while the version number is raised. Additional metadata tags can be added during the model registration, which helps in easy searching.

- **Image Registry:** It keeps a record of the pictures that the models produced. When creating an image that is stored by an image registry, more metadata tags are added. To discover the image, you can use these tags as a search term.

Before we start collecting and processing our data, we need a Workspace where we can perform all the operations. The most organized level of machine learning solution is represented by a Workspace. It contains a list of all the computation targets utilized during model training. It keeps a record of each training session's metrics, results, and snapshots. The optimum training model for the project can be chosen with the use of these data. Through the workspace, the model is registered. Azure ML service workflow is a three-step process that includes:

- **Prepare Data:** The process of collecting and processing data from datasets and datastores is the initial stage in the creation of a machine learning model. Some examples of supported Azure storage services that can be listed as datastores are: Azure Data Lake, Azure SQL Database, Databricks File System and Azure Blob Container.

- **Experiment:** The following stage is to create, test, and train the model after the data has been registered and stored in the dataset.

- **Model:** It is a part of code that receives input and outputs the specified results. The steps involved in creating a ML model include choosing an algorithm, obtaining data, and fine-tuning hyperparameters. A trained model inherits what it learned from the training process due to the cyclical nature of training. Executing in Azure Machine Learning produces the model.

- **Compute Targets:** The host service deployments or training scripts are run on a system or a group of machines. A compute target might alternatively be a local computer or a distant computing resource. The compute resources used for compute targets are attached to the workspace.

- **Deployment:** Once the model is trained and tested, it is stored in the model registry and then deployed in web service or IOT modules.

2.5.3. MS Azure Data Factory

Data is moved and transformed between different data repositories and compute resources using Azure Data Factory. Data-driven workflows, also known as pipelines, can be planned, and created to ingest data from various data repositories. With data flows or computing services like Azure Synapse Analytics, Azure Databricks, Azure HDInsight, and Azure SQL Database, you can create intricate ETL processes that graphically change data. Data Factory's function is to extract data from one or more data sources and transform it into a structure that can be processed. It may be necessary to remove noise from the data sources because they may present data differently.

Data can be converted in a format that the other services in the warehousing solution can use to handle it. You may create the data copy, ingestion, and transformation workflows using the various parts of Azure Data Factory, by creating pipelines to carry out one or more actions. Afterward, the associated services linked to the data sources or services can be used. You can also add triggers to an existing pipeline to have it run automatically at predetermined intervals or in response to certain occurrences. In Figure 10, we can observe the different components that will be explained in more detail [AZURE].

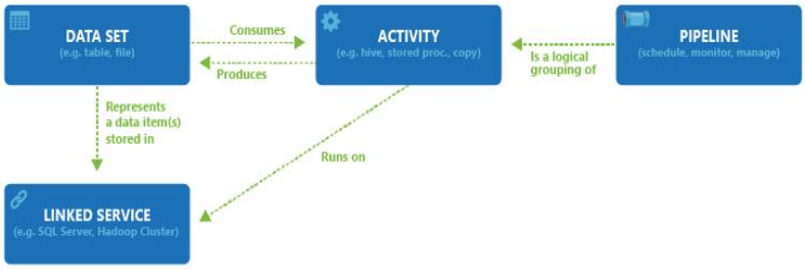

Figure 10. Azure Data Factory components. source Microsoft.com

- **Pipeline:** A pipeline is a logical collection of tasks that carry out a single task. There could be one or even more pipelines in Data Factory. A pipeline can be activated physically or by a trigger. The operations in a pipeline can run independently in parallel or sequentially when chained together.

- **Activity:** Activities consume and/or produce Datasets. Activities in a pipeline reflect a specific processing step or activity. Activities in a pipeline specify the actions to be taken concerning the data. Activities typically assist data transformation, control, and movement. Activities can be carried out sequentially or concurrently.

- **Dataset:** Dataset is the representation of the data. Datasets serve as a representation of data structures in data repositories. The data to input or output for actions are represented by datasets (data source or sink).

- **Linked Services:** The information a linked service contains varies according to the resource. The linked service defines the link to the source of data. It indicates the location of the data. Similar to connection strings that specify the connection details required for Data Factory to join to external resources, Linked Services define the connection information (source or destination). A linked service specifies a target data storage or a compute service.

The connection between Linked Services and Activity is made possible by the Integration Runtime. Data Factory uses Integration Runtime as its computing environment (infrastructure), where the activities operate on or are dispatched. When a pipeline should be executed is decided by triggers. A pipeline can be run on a wall-clock

schedule, at regular intervals, or during an event. When a pipeline execution requires to begin, triggers serve as the processing unit that makes that call. Data engineers can visually create a data transformation logic using these unique activities rather than writing code. A visual editor can alter data in several phases without writing any more code than data expressions. For scaled-out processing using Spark, they are carried in the ADF pipeline on the Azure Databricks cluster (managed Spark cluster). ADF manages all code translation and data flow execution. It can handle many data easily.

In summary, for our project, we will use Data Factory to create Pipelines for data transfer. We will set the schedule of pipelines according to our requirements daily or weekly to run the job. Data factory will also be used to visualize and monitor the pipelines and set up security alerts.

2.5.4. MS Azure Data Lake

Azure Data Lake Storage offers a highly scalable and secure data lake for high-performance analytics applications. The Azure Lake Data Store is occasionally used to refer to Azure Lake Data Storage. It offers a single storage platform that can utilize to connect their data and is intended to do away with data silos. With tiered storage and policy control, the storage can aid in cost optimization. Data of any shape, size, and speed with the aid of Azure Data Lake by data scientists, developers and analysts, which provides all the tools and services required. It is beneficial to carry out various processing and analytical tasks across platforms and in different languages. Using batch, streaming, and interaction analytics, it simplifies and speeds up storing and absorbing data. There are several advantages of Azure Data Lake because it is hosted in the cloud, it is very versatile and scalable and enables streamlined data storage for any business requirements. Processing enormous amounts of data simultaneously enables speedy access to insights. As shown in Figure 11, Data Lake holds all types of information, including binary, chat, people, sensor, log, and XML data. There is no file or data size restriction. Allows for extremely high analytics workloads for thorough analysis. It supports storage data with no schema. In our project, Azure data lake makes storing the data in any shape easy. Our data is dynamic in type and comes from different sources (historical and real-time fluid bed dryer data), so Data Lake is suitable for storing them. It also allows training machine learning models and scaling them according to future needs.

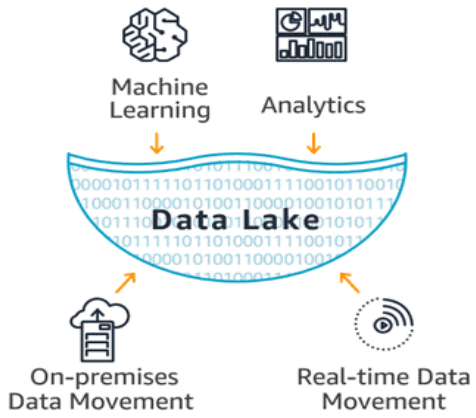

Figure 11. Data Lake source Microsoft.com

2.5.5. MS Azure IOT Hub

A networking of physical objects, such as furniture, vehicles, appliances, and other items, that are linked together and share data is known as the Internet of Things (IOT). These objects are embedded with electronics, sensors, actuators, software, and connectivity. Microsoft's Internet of Things cloud connector is called Azure IOT Hub. With the help of this managed cloud service, millions of IOT devices might safely and reliably communicate with a back end of a system. Azure IOT hub permits two-way communication between IOT applications and managed devices. With this cloud-to-device communication, you may not only receive the data from your devices but also communicate with them by sending commands and policies. The way Azure IOT hub differs from other options is that it also offers the infrastructure needed to connect, authenticate, and manage the connected devices as shown in Figure 12.

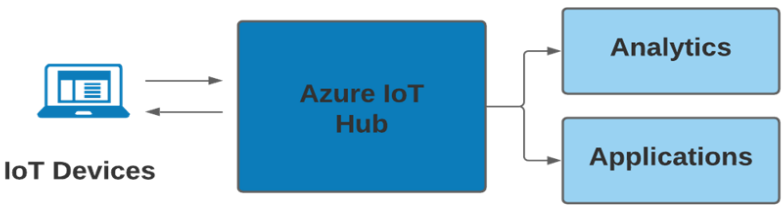

Figure 12. MS Azure architecture for IOT

Full-featured and scalable IOT systems are possible with Azure IOT Hub. Virtually any device can be connected using Azure IOT Hub, which can scale to hundreds of

thousands of devices. It is possible to record and watch events such as the formation, failure, and connections of devices. For simple device connecting, Azure IOT Hub offers:

- Device libraries for the most popular programming languages and platforms for simple device interfacing.

- Hyperscale connectivity between devices and clouds that is secure and offers a variety of possibilities.

- Storage of meta-data and individual device state information.

In our project, we use Azure IOT to take/send data from/to the fluid bed dryer to our MS Azure architecture, which will be explained in more detail in the next chapter.

2.6. Conclusions

To recap the content of this chapter, we have presented the state of art and a review of the most relevant literature of the different topics addressed in this thesis, including an introduction to the pharmaceutical manufacturing process, a brief description of fluid bed dryer operations, the current paradigm Industry 4.0 and the new sub concept called Pharma 4.0, digital twin technology and an introduction to machine learning, including some state of art related work in the field of applying machine learning to reduce the energy consumption in the manufacturing industry.

Besides, we have included an introduction to cloud computing concepts, industrial internet of things, Edge computing, and an overview about Microsoft Azure platform main components or modules that have been used to implement this thesis.

This thesis offers an innovative proposal on improving energy consumption on fluid bed dryer operations in the pharmaceutical industry, proposing combining different techniques including in the related work, for data extraction, data preprocessing, data modeling, and real time cloud computing connection to the machinery to predict and help operators to know what the optimum moment is to stop the process and save time and costs.

We proposed a methodology that covers end-to-end implementation, from data capture to cloud computing prediction in real time. This methodology can be easily implemented to reduce energy consumption in fluid bed dryers in pharmaceutical manufacturing or other industries, such as food, dairy, metallurgical, or chemicals. This methodology could also be extended to other process operations such as mixing, compacting or coating in the pharmaceutical industry.

3 Data capture and exploratory data analysis (EDA)

This chapter proposes a methodology to show how the data is captured, pre-processed and analyzed from a fluid bed dryer [Barriga et. al. 2023]. The fluid bed drying machine that will be used in this thesis is the Fielder Aeromatic MP 6/8 (FAMP68) located in a pharmaceutical manufacturing plant in Barcelona (Spain), which is shown in Figure 13. This machine has 56 sensors governed by SCADA (Supervisory Control And Data Acquisition), through which the operators monitor and configure the basic parameters of the machine such as the inlet air temperature or the air flow.

The fluid bed drying machine has three critical parameters that characterize the efficiency of the drying process and therefore can influence the final quality of the product. These parameters are: temperature, humidity and air flow. In theory, a higher temperature and flow rate of the inlet air to the machine implies a shorter drying time.

However, each of these three parameters must be configured correctly depending on the type of product, to avoid quality problems and deterioration of the final product obtained after the drying process. It is important to note that the inlet air temperature should not exceed the critical temperature of the product to be dried so as not to jeopardize its quality or pharmaceutical properties.

Figure 13. Fluid bed dryer model Fielder Aeromatic MP 6/8

This process is monitored by the operator through SCADA, which records the increase in outlet air temperature as the product is being dried, taking into account that the outlet air temperature is almost the same to the inlet air when the product has been dried and water has been completely removed from the granulate. At this point it is critical to stop the operation, since lengthening it more than necessary could put the quality of the product at risk, as well as consume more time and energy than necessary for the process, with as consequence an increase in costs for the process.

The fluid bed drying machine it is not equipped with sensors that indicate when the machine has reached the optimum temperature for the different drying phases

(preheating, drying and cooling), so usually, fixed times are used for the drying phases by the human operators. However, for the preheating phase, the time can vary depending on the experience of the operator with the machine.

3.1. Proposed methodology

This section describes the proposed methodology to capture, pre-process and analyze the data of the fluid bed dryer. This process aims to evaluate if there is room for improvement in the finalizing time of the current processes that the machine performs a) preheating, b) drying, or c) cooling.

Figure 14 shows the general process that will be followed for the data capture and exploration analysis.

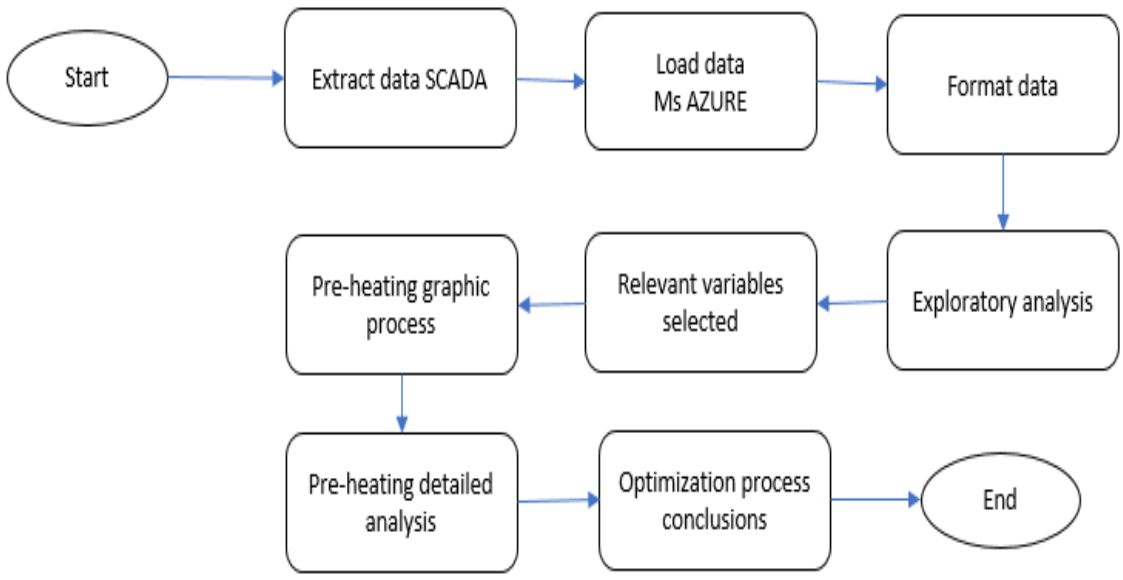

Figure 14. Proposed methodology for pre-heating analysis

- **Extract data SCADA:** First step is to extract the data from the fluid bed dryer. As commented before, the fluid bed dryer is equipped with some sensors connected to a SCADA, so we can easily extract from the SCADA an historic of a year and a half data in a CSV format that includes sensor values for each minute.

- **Load data MS Azure:** Once the data is extracted to a CSV file, it will be uploaded to a Microsoft Azure Databricks platform to perform next steps.

- **Format data:** Phyton libraries are used to format/clean the data. The practice of correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. Even if results and algorithms appear to be correct, they are unreliable if the data is inaccurate.

- **Exploratory analysis:** With the aid of summary statistics and graphical representations, exploratory data analysis refers to the crucial process of conducting first investigations on data in order to find patterns, uncover anomalies, test hypotheses, and verify assumptions.

- **Relevant variables selected:** During this step, based on the exploratory analysis, the most relevant variables of the preheating process are identified thanks to the analysis of how the fluid bed dryer works.

- **Pre-heating graphic process:** Statistical graphic visualization tools are used to be able to analyze in detail the behavior of each batch in the fluid bed dryer and detect behavior patterns.

- **Pre-heating detailed analysis:** In this step, the batches will be analyzed in detail to detect possible time savings in the preheating process.

- **Optimization process conclusions:** Finally, the conclusions of the analysis and the potential savings in time and therefore in energy, will be shared and verified with the experts of the fluid bed dryer.

As shown in Figure 15, the data will be collected in a matrix D with *m* columns and *n* rows, where n corresponds to the time measured minute by minute and m to the 56 sensors of the machine. We define the variables that measure the airflow, fan motor and phase as Cφ, Mσ, F~, respectively. The variables used to select data by days and batches are also defined as Q_D and Q_L, respectively. PS will indicate the drying process and *t* the time.

1. *Define a matrix D [m, n]*
2. *Assign data extracted from SCADA to D*
3. *Upload D in the cloud environment (Azure)*
4. *Eliminate values with variability equal to zero in D*
5. *Generate charts from D*
6. *Select key variables D~(TA_e TA_s $C\phi$ $M\sigma$ F~)*
7. *Select key rows D~ ~ (Q_D Q_L)*
8. *Generate charts from D~ ~ for PS*
9. *Distribution t (D~ ~, PS)*
10. *Distribution TA_D (D~ ~, PS)*

Figure 15. Algorithm of the proposed methodology for pre-heating analysis

3.2. Data Capture

The first step of the process is to capture data from the different sources as shown in Figure 14. In our case, we have the FAMP68 machine that currently operates in a real pharmaceutical plant of a multinational company in Barcelona. This machine typically processes between one or two batches of pharmaceutical drug granules per day, each batch of product contains approximately 150Kg of drug that has been mixed previously with 25Kg of alcohol and 10Kg of another excipient before being introduced into the fluid bed dryer. The machine has 56 sensors that measure inlet / outlet air temperature, air flow in m3/h, motor rotation speed, and air pressure, among others. Each sensor collects data minute by minute. We have 2 years' worth of data, which is equivalent to more than 700,000 readings of each of the 56 signals. The data is collected by a PLC (Programmable Logic Controller) and stored in SCADA (Supervisory Control And Data Acquisition).

Table 2 shows the fluid bed dryer sensors including a description for each signal, the minimum and maximum value and their units of measure.

Item	TagName (Symbol)	Description	Units	Min	Max	PMA	TSG	CIP
1	FS3_GEA_EIS1200_ME	Impeller power [Kw]	Kw	0	30	X		
2	FS3_GEA_EOP_GP	Current EOP at GP	None	0	1000	X		
3	FS3_GEA_EOP_MP	Current EOP in MP	None	0	1000		X	
4	FS3_GEA_FIC1217_ME	Liquid Flow in GP [cl/min]	cl/min	0	833	X		
5	FS3_GEA_FIC1217_XS	Liquid flow setpoint in GP [cl/min]	cl/min	0	833	X		
6	FS3_GEA_FIC200_ME	Air flow [m3/h]	m3/h	0	4500		X	
7	FS3_GEA_FIC200_XS	Air flow setpoint [m3/h]	m3/h	0	4500		X	
8	FS3_GEA_FIC701_ME	Spray liquid flow in PM [cl/min]	cl/min	0	667		X	
9	FS3_GEA_FIC701_XS	Setpoint liquid spray flow in MP [cl/min]	cl/min	0	667		X	
10	FS3_GEA_LI940_ME	Cleaning water tank level [L]	L	0	500			X
11	FS3_GEA_MIS213_ME	Inlet air humidity [g/Kg]	g/Kg	0	25		X	
12	FS3_GEA_NFGP	No. Current phase in execution in GP	None	0	1000	X		
13	FS3_GEA_NFMP	No. Current phase in execution in PM	None	0	1000	X		
14	FS3_GEA_NFW	No. Current cleaning phase in execution	None	0	1000			X
15	FS3_GEA_NW	No. Current cleaning running	None	0	1000			X
16	FS3_GEA_PDIA111_1ME	Product Pressure [Pa]	Pa	-14000	10000		X	
17	FS3_GEA_PDIA111_ME	Air inlet pressure [Pa]	Pa	-10000	10000		X	
18	FS3_GEA_PDIA112_ME	Air Outlet Pressure [Pa]	Pa	-10000	10000		X	
19	FS3_GEA_PIA740_ME	Liquid pressure spray 1 [mbar]	mbar	0	4000		X	
20	FS3_GEA_PIA741_ME	Liquid pressure spray 2 [mbar]	mbar	0	4000		X	
21	FS3_GEA_PIA742_ME	Liquid pressure spray 3 [mbar]	mbar	0	4000		X	
22	FS3_GEA_PIA743_ME	Liquid pressure spray 4 [mbar]	mbar	0	4000		X	
23	FS3_GEA_PIA744_ME	Liquid pressure spray 5 [mbar]	mbar	0	4000		X	
24	FS3_GEA_PIA745_ME	Liquid pressure spray 6 [mbar]	mbar	0	4000		X	
25	FS3_GEA_PIA746_ME	Liquid pressure spray 7 [mbar]	mbar	0	4000		X	
26	FS3_GEA_PIA747_ME	Liquid pressure spray 8 [mbar]	mbar	0	4000		X	
27	FS3_GEA_PIC702_ME	Spray air pressure [mbar]	mbar	0	10000		X	
28	FS3_GEA_QI917_ME	Cleaning water conductivity [mS]	mS	0	200			X
29	FS3_GEA_SIC1200_ME	Impeller speed [rpm]	%	0	170	X		
30	FS3_GEA_SIC1200_XS	Impeller speed setpoint [rpm]	rpm	0	170	X		
31	FS3_GEA_TFGPM	Current phase elapsed time in GP [min]	min	0	1000	X		
32	FS3_GEA_TFGPS	Current phase elapsed time in GP [s]	s	0	1000	X		
33	FS3_GEA_TFMPM	Current phase elapsed time in MP [min]	min	0	1000		X	
34	FS3_GEA_TFMPS	Current phase elapsed time in MP [s]	s	0	1000		X	
35	FS3_GEA_TFWM	Current cleaning step elapsed time [min]	min	0	1000			X
36	FS3_GEA_TFWS	Current cleaning step elapsed time [s]	s	0	1000			X
37	FS3_GEA_TI115_ME	Product temperature in MP [°C]	°C	0	120		X	
38	FS3_GEA_TI214_ME	Inlet air temperature [°C]	°C	-30	70		X	
39	FS3_GEA_TIA242_ME	Preheating steam temperature [°C]	°C	0	200		X	
40	FS3_GEA_TIA312_ME	Outlet air temperature [°C]	°C	0	120		X	
41	FS3_GEA_TIA918_ME	Cleaning water temperature [°C]	°C	0	100			X
42	FS3_GEA_TIC1223_ME	Product temperature in GP [°C]	°C	0	120	X		
43	FS3_GEA_TIC1223_XS	Product temperature set point in GP [°C]	Pa	0	120	X		
44	FS3_GEA_TIC201_ME	Dryer inlet temperature [°C]	°C	0	120		X	
45	FS3_GEA_TIC225_ME	Preheat temperature [°C]	°C	-30	50		X	
46	FS3_GEA_TIC225_XS	Preheating temperature set point [°C]	°C	0	120		X	
47	FS3_GEA_TIC231_ME	Cooling temperature [°C]	°C	0	120		X	
48	FS3_GEA_TIC231_XS	Cooling temperature set point [°C]	°C	0	120		X	
49	FS3_GEA_TIC711_ME	Liquid temperature [°C]	°C	0	120	X		
50	FS3_GEA_TIC711_XS	Liquid temperature set point [°C]	°C	0	120	X		
51	FS3_GEA_TIC914_ME	Cleaning preheating temperature [°C]	°C	0	120			X
52	FS3_GEA_TIC914_XS	setpoint temp. cleaning preheating [°C]	°C	0	120			X
53	FS3_GEA_TOWS	Time Opr. clean running [s]	s	0	1000			X
54	FS3_GEA_TSPWS	Actual cleaning step setpoint time [s]	s	0	1000			X
55	FS3_GEA_TTFWM	Current cleaning phase elapsed time [min]	min	0	1000			X
56	FS3_GEA_TTFWS	Current cleaning phase elapsed time [s]	s	0	1000			X

Table 2. Fluid Bed Dryer sensors

Some of these sensors are involved in different processes, such as granulation (column PMA), drying (column TSG) or cleaning (column CIP). For the exploration phase, we will select the sensors involved just in the drying process (column TSG), but as we will explain in chapter 4, for the data modeling, we will select all of them, simulating a real situation where we were not able to differentiate which sensor belongs to which phase.

Figure 16 shows the SCADA that is used by the operators to interact with the machine (start / stop controller, inlet air temperature indicator, inlet air flow indicator, etc.). The data from SCADA has been exported into a table composed of more than 700.000 rows and 56 columns.

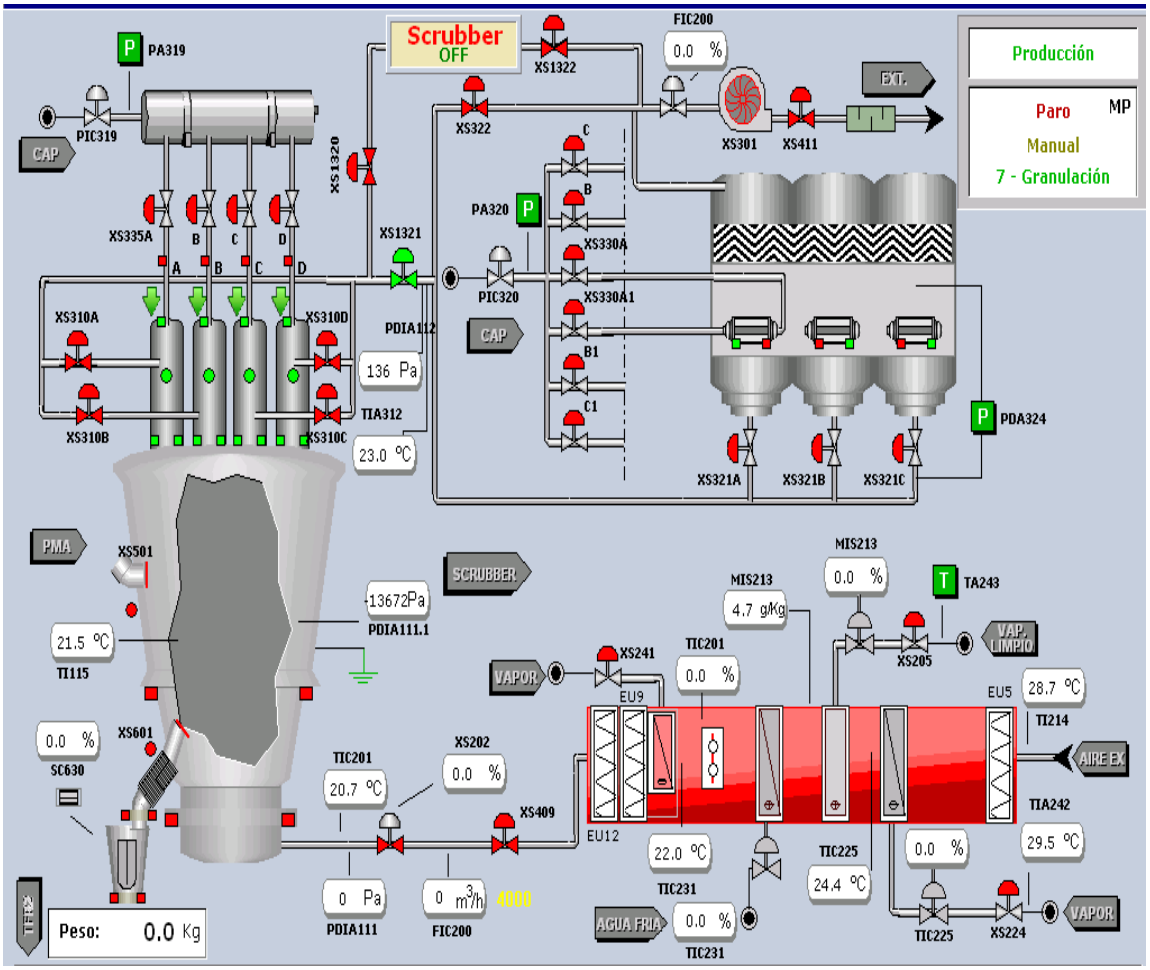

Figure 16. Fluid bed dryer SCADA

On the SCADA screen, we can see the status of the station in detail, including the values of the sensors and valves, as for example temperature or pressure, and in the upper right it shows the state of the fluid bed dryer, what process it is carrying out and what state each of them is in (granulating, drying or cleaning). For example, when steam is

added to the fluid bed dryer to control the humidity of the air that is introduced into the dryer, if the humidity is very low, more steam is added to increase it. The air that is introduced into the dryer, allow us to control both the temperature and its humidity. The pressure of the dryer is indicative of the clogging of the filters, if there is a big difference between the internal pressure and the output pressure, it means that we have dirty filters, and you need to clean them. The SCADA records and monitors the operating status of the fluid bed dryer in its operating modes and states and the duration of these and the registers of the analog parameters involved. The system must acquire, display and record the following analog variables of the air conditioner associated with the fluid bed dryer process:

- Air inlet temperature (ºC)
- Preheating temperature (ºC)
- Cooling temperature (ºC)
- Industrial steam temperature (ºC)
- Inlet air humidity (ºC)

3.3. Exploratory Data Analysis

The collected data is loaded to a cloud computing platform to be processed. Due to the high volume of data (more than 3GB of data), we have selected the Azure platform and its advanced analytics module Databricks using Python for data analysis. Before we begin the exploration analysis, we must first format and clean our dataset. To ensure that the dataset has a valid format for the exploration, we will use some Phyton functions, such as “normalize” or var_zero_remove”. Although the majority of the columns are already using the same format, we will use functions such as the function shown in Figure 17. The function normalize permits to ensure consistency in the dataset's format. In this case, sensor FS3_GEA_NFMP and FS3_GEA_NFMP are normalized with the function. These sensors indicate the fluid bed dryer number of phase (preheating, drying or cooling).

```
def normalize(df):
    dfn = (df - df.mean())/df.std()
    dfn.FS3_GEA_NFMP = df.FS3_GEA_NFMP
    dfn.FS3_GEA_NFGP = df.FS3_GEA_NFGP
    dfn['date'] = df.index.date
    return dfn.fillna(0)
```

Figure 17. Sensor data normalization function

After normalizing the data, it is observed that the data produced by some sensors (columns) do not vary over time. So, we proceed to delete these columns to simplify the data set and to focus on the data from sensors that can help optimize the preheating process of the machine. Overall, we find 11 sensors that do not provide any relevant information about the preheating or drying process as they remain constant over the year and a half period and can therefore be eliminated from the dataset.

3.3.1. Psychometric model

A psychometric model to select the most relevant sensors/variables for exploratory analysis will be presented in this subsection. The psychrometric chart (see Figure 18) is a useful and easy to use tool for determining moist air psychrometric properties and visualizing the changes of properties in a sequence of psychrometric processes.

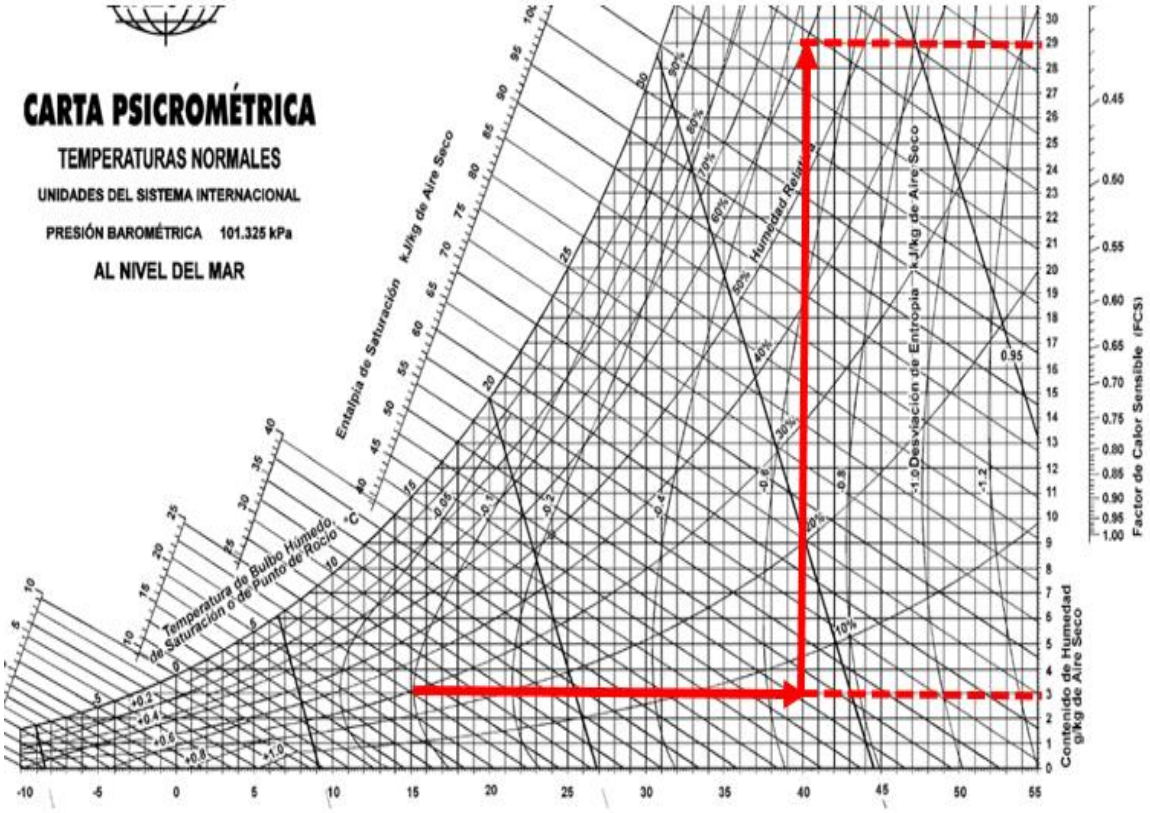

Figure 18. Psychrometric Chart including the example commented in red

The vapor pressure is the outermost curve, which would mark the water-vapor change of state. As we can see, the pressure of change of state increases with increasing temperature, therefore, if we carry out a process with constant humidity, we have that the partial pressure of the water in the mixture increases with increasing temperature. For this reason, the constant RH curves are increasing. In our fluidized bed drying process, we are carrying out a constant pressure process. This causes the air entering the chamber to travel along a horizontal line starting from an initial point at the entrance to the chamber.

The movement of the point, represented in Figure 18, is going to be the horizontal line to the left that starts from the initial point because, by absorbing the water from the granules, the humidity of the air is going to increase. The air can continue to absorb water as long as this line does not reach the Dew Point, at which time the air will be saturated with moisture. Fluidized bed dryers are designed so that the inlet air is sufficiently hot and dry so that the length of this dew point line is long enough to absorb

all the moisture in the granules. According to the psychrometric chart, the psychrometric ratio is defined formally as the ratio of the heat transfer coefficient to the product of mass transfer coefficient and humid heat at a wetted surface. It is quantified using the following equation, where r = psychrometric ratio, h_c = convective heat transfer coefficient (Wm-2K-1), k_y = convective mass transfer coefficient (kg m-2 s-1) and c_z= humid heat (J kg-1K-1).

$$r = \frac{h_c}{k_y c_z}$$

Based on the psychometric diagram [Barriga and Romero, 2022], noticed lines colored in red, we can determine the quantity of heated air necessary to evaporate a given quantity of water. For example, an air temperature of 15ºC and a humidity of 30% can import up to 26g of water per kg of dry air if this has been boiled at 50ºC previously and considered exit at 60% relative humidity. The comparison of the theoretical value of evaporable water and the real one is a measurement of the effectiveness of the assessment, typical of the conditions of each recipient and each product named for this equipment. These effectiveness values can be set and fixed in such a way that the measure of evolved over time to support the preventive maintenance actions. Taking into account the psychrometric model and how the fluid bed machine works, we have selected four sensors for the exploration analysis:

- **Fan motor:** The signal shows when the fluid bed dryer is on or off.

- **Air flow:** The signal indicates the air flow (quantity in m3 / h) that enters the fluid bed dryer. This is configured by the machine operator. This sensor helps us to identify if the fluid bed dryer is preheating or drying, as both processes need air to be completed

- **Inlet air temperature:** The signal indicates the temperature at which the air enters the fluid bed dryer and is also set by the operator at the beginning of the process.

- **Outlet air temperature:** The signal indicates the temperature at which the air leaves the fluid bed dryer.

In Table 3, it can be seen the information for 4 selected sensors (power impeller, liquid flow, ai flow, and inlet air temperature) from the 56 sensors of the fluid bed dryer shown in Table 2. Fluid Bed Dryer sensors. The information presented in the rows of Table 3 corresponds to the number of sensor readings (count), the average value of each sensor (mean), the standard deviation of each sensor (std), as well as the maximum and minimum values, and the limit of each of the quartiles for each sensor. This preliminary information allows us to know the average values of the variables and discard batches that have been processed and do not approximate the average parameters, since they would correspond to batches that have had, for example, a problem during the drying process where an unexpected issue has occurred.

	Power impeller [Kw]	Air flow [m3/h]	Inlet air temperature [ºC]	Outlet air temperature [ºC]
count	1441,000000	1441,000000	1441,000000	1441,000000
mean	0,208952	1040,490632	23,295212	27,635045
std	0,944463	1366,454053	6,842901	7,364493
min	0,000000	-58,000000	10,700000	22,100000
25%	0,000000	-55,000000	16,800000	22,900000
50%	0,000000	-55,000000	27,700000	23,400000
75%	0,100000	2471,000000	29,500000	30,800000
max	11,600000	4042,000000	31,600000	47,900000

Table 3. Example of signals used for the experiment

Following the principle of the psychrometric process, once the fluid bed dryer is running and the hot air inlet process begins, we have to take into account the heat absorbed by the machine to reach preheating temperature. This means that we can rely on the sensor that indicates the temperature of the outlet air of the machine to know how much heat the fluid bed dryer is absorbing. By subtracting the air inlet and outlet temperatures, we can detect when the machine is not capable of absorbing more heat and therefore the inlet air temperature will be similar to the outlet air temperature. To better understand process behavior, we will consider the temperature differences of the air inlet and outlet of the machine, as we have commented previously, a variable that we will define in equation 1.

$$TA_D = TA_s - TA_e \qquad (1)$$

where TA_s is the outlet air temperature, TA_e the inlet air temperature and TA_D is the temperature difference.

3.3.2. Sensor Exploratory Data Analysis

Once the sensors have been selected, as next step, we will choose random days, to observe the behavior of the machine signals when carrying out the preheating, drying and cooling process each time a batch of pharmaceutical product is processed. The main objective of this exploration is to identify trends and better understand fluid bed dryer processes to identify improvement opportunities. Figure 19 illustrates graphically the behavior of the signals on different days, which indicates a full day operation of the fluid bed dryer. Figure 19 also shows on the *x-axis* the elapsed time for one day fluid bed dryer operation (1440 minutes in total corresponding to 24 hours) and on the *y-axis* the difference in temperature of the machine's inlet and outlet air. Blue dots indicate the preheating process, orange dots the drying process, and green dots the cooling process.

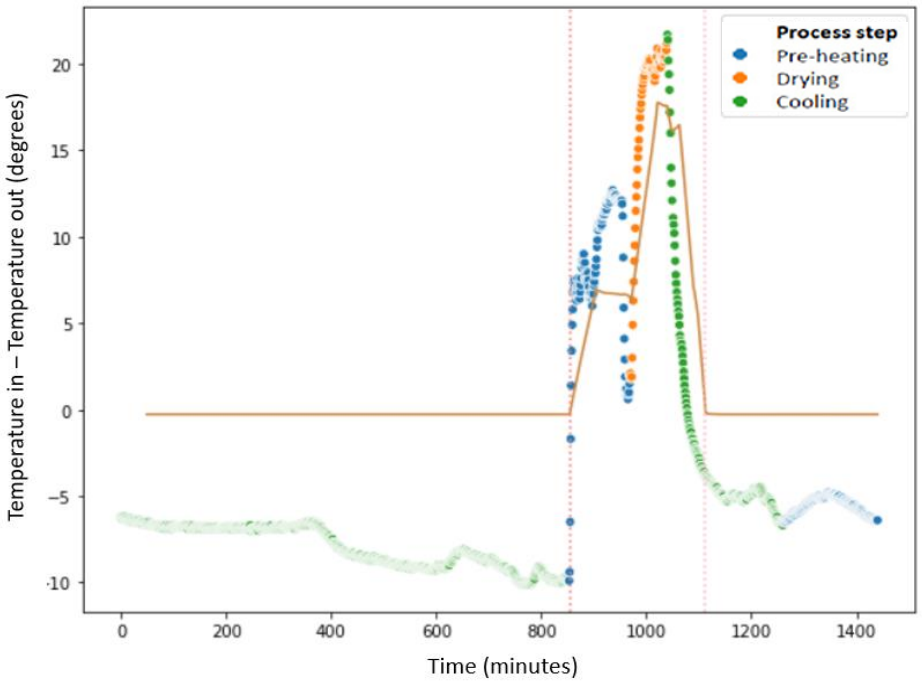

Figure 19. Plot of phases of the drying process.

Table 4 includes the name of the four sensors commented in Table 3, and a new sensor that will identify in which phase the fluid bed dryer is operating.

Abbreviation	Sensor	Average	Max	Min
IF	Phase Indicator (1,2,3)	N/A	3	1
TAE	Temperature inlet Air	23,2	52,8	0
TAS	Temperature outlet Air	27,1	47,9	0
CAE	Inlet Air Flow	1040	4042	0
MOT	Fan Motor	19,8	151	0

Table 4. Signals used for the experiment

The IF signal (phase indicator) can take the values 1, 2 or 3, depending on the phase the fluid bed dryer is in. Phase 1 corresponds to the preheating phase, where the machine needs to warm up through the hot air inlet to be able to start the drying process (which corresponds to the value 2). When the IF signal (phase indicator) acquires the value 3, it means that the drying process has concluded (phase 2), and therefore the machine must be cooled with air inlet to a lower temperature for the environmental conditioning that will avoid condensation when cooling. The TAE (inlet air temperature) signal corresponds to the degrees to which the air enters the machine for any of the three phases (1: preheat the machine, 2: dry the product, 3: cool the machine). The TAS (outlet air temperature) signal corresponds to the temperature in degrees of the air coming out of the machine. The CAE signal (inlet air flow) indicates the volume of air per unit of time supplied by the machine's fan, and finally the MOT (fan motor) signal is used to know when the machine is activated in any of the three phases (when the fan motor starts). In Table 4, we can see the different signals, as well as their mean, maximum and minimum values for a random sample of signals. Note that it is expected to see null values for the minimums of the inlet and outlet temperatures.

Finding how many product batches are dried in the fluid bed dryer each day is the first task that has been performed for the data exploratory analysis. A random sample of signals is taken using only those in which we have the fan motor running (MOT> 1). In

Figure 20, the x-axis indicates the number of minutes elapsed in a day 1400 minutes, and the y-axis corresponds to the inlet air temperature difference (TAE) and output (TAS) of the machine. Each point corresponds to a phase of the IF signal (phase indicator). The value 1 corresponds to the preheating phase (blue), value 2 to the drying phase (orange), and the value 3 to the cooling phase (green) of the machine.

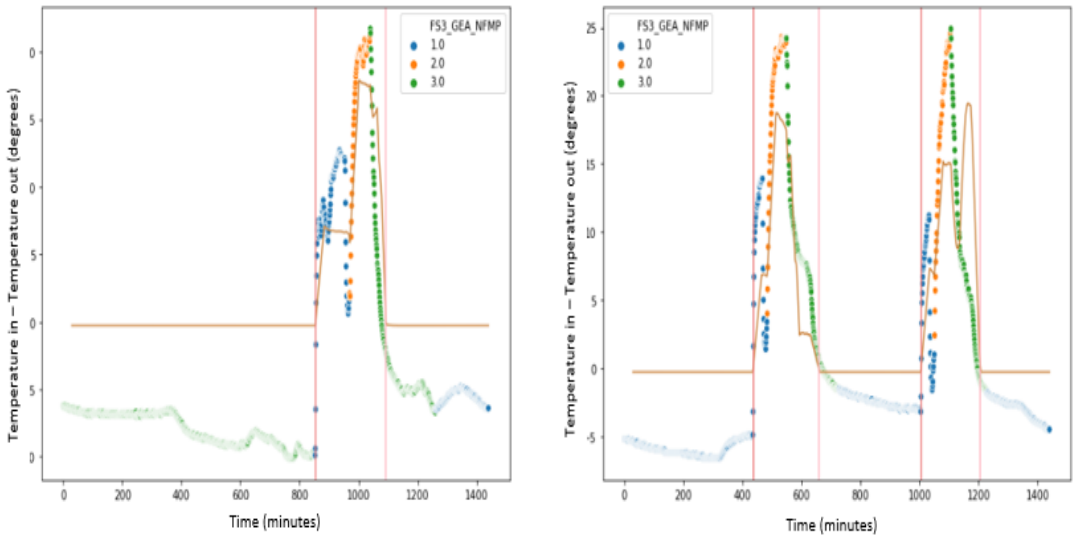

Figure 20. Batches produced per day. Left one batch. Right two batches.

It has been also analyzed how much time (in minutes) it takes on average to complete the three drying phases: preheating, drying and cooling, as shown in

Figure 20. It is identified that the fluid bed dryer requires approximately 350 minutes (or 6 hours), to dry a batch of product. Figure 21 shows an example of 8 different days taken randomly where we can observe that some days the fluid bed dryer processes one batch, and other days two batches, with an average of around 350 minutes per batch. We can observe, how figure of date 02-12-2019 there are two batches that are processed and if we look at the blue dots, we will see that the preheating process lasts much longer in the two batches, compared to the duration of the preheating process, for example, on 07-10-2018, where we see that the blue dots are much smaller and the temperature difference, y-axis, does not exceed 10 degrees. We can also observe that the duration of the drying process, orange dots, is more or less homogeneous, it lasts approximately the same for all days and all batches (x-axis), as well as the temperature differences are approximately similar (y axis). Another relevant example of excessive duration of the heating process would be the figure of the date 04-10-2018, where it can be observed that the pre-heating process, blue dots, lasts approximately 150 minutes (x-axis). If we compare the duration of this process with other days, for example on 07-03-2018, it can clearly be observed that there are no criteria to define the optimum number of minutes that the fluid bed dryer needs to be properly preheated before starting the drying process.

As a conclusion, we can observe that the duration of the preheating process seems to be variable. To preheat the fluid bed dryer, some batches take longer time preheating the machine than others, with the consequent unnecessary consumption of energy [Barriga MAD, 2019].

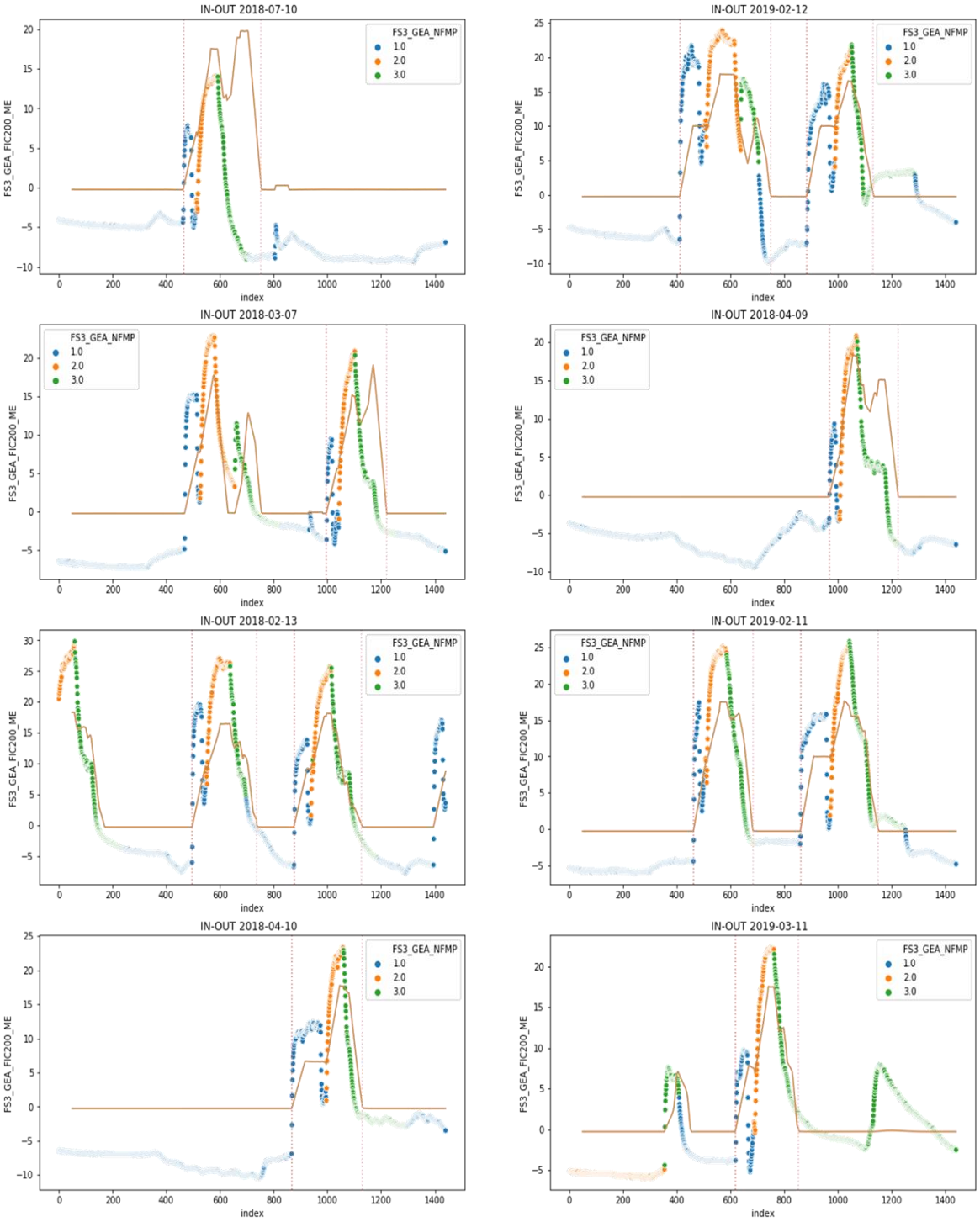

Figure 21. Example of 8 different days of batch drying. Above each figure is plotted the date of the batch.

3.3.3. Preheating phase analysis

The preheating phase is denoted by the blue dots in Figure 22 is the center of our attention once the average drying time of a batch, which includes its three phases (preheating the machine, drying the product, and cooling the machine), is determined. This time period lasts roughly 6 hours. The goal is to know how much time it takes to heat up the dryer before starting the drying process. Since we are going to focus on the preheating phase, we will select the data that meets the condition IF = 1 (preheat phase) and CAU> 0 (airflow greater than zero), and we will choose a day to identify the duration in minutes of the preheating phase.

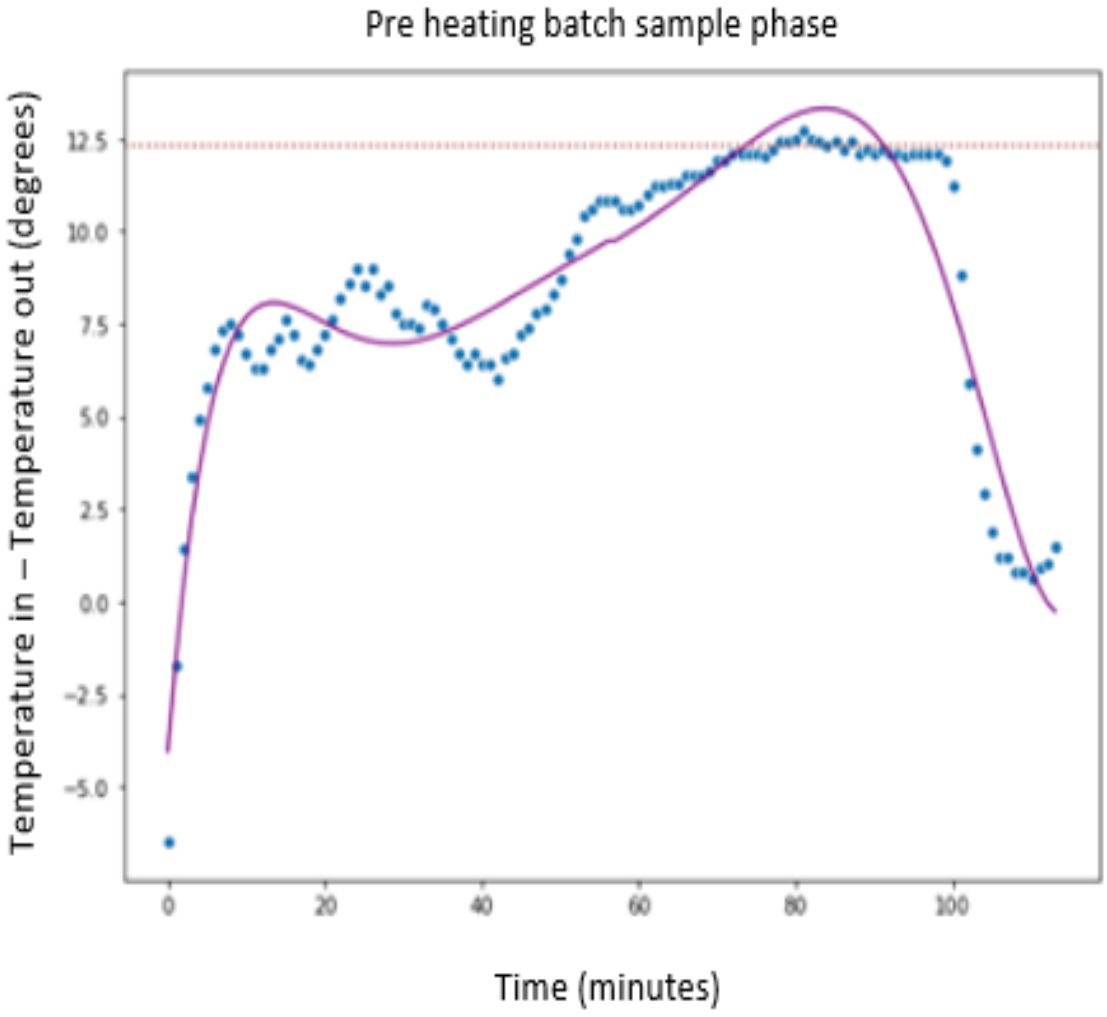

Figure 22. Preheating phase analysis of 1 batch of product.

In Figure 22, the blue dots represent the difference between the inlet (TAE) and outlet (TAS) temperatures. The x-axis corresponds to the time in minutes of the preheating phase IF = 1. We have used a smooth function, Savitzky–Golay filter, that is a digital filter that can be applied to a set of digital data points for the purpose of smoothing the data, that is, to increase the precision of the data without distorting the signal tendency. The purple horizontal line represents the maximum value predicted by the smoothing function, and the horizontal line dotted in red represents the maximum value.

The data consists of a set of points *(x_j, y_j), j = 1...n*, where x_j is an independent variable and y_j is an observed value. The data is processed with a set of *m* convolution coefficients C_i, expressed in equation 2:

$$Y_j = \sum_{i=\frac{1-m}{2}}^{\frac{m-1}{2}} C_i y_{j+i} \ , \ \frac{m+1}{2} \leq j \leq n - \frac{m-1}{2} \qquad (2)$$

where Y_j is a smoothed data point corresponding to observed value y_j.

By studying the data from a day of processing of a batch of product from Figure 22, we can observe that the machine uses for preheating more than 100 minutes (x-axis), at which point the curve begins to descend. At this point, the next phase starts where the granulated product is loaded into the machine to begin drying. It can also be observed that the maximum difference between the air inlet and outlet temperatures in both cases is between 12 and 15 degrees (y-axis).

Therefore, it can be deduced that hot air is being introduced into the machine for a longer period than necessary (since the temperature differences between the inlet and outlet air remain stable). Thus, the fluid bed dryer is keeping the process longer than the necessary time and during which energy is being consumed and wasted.

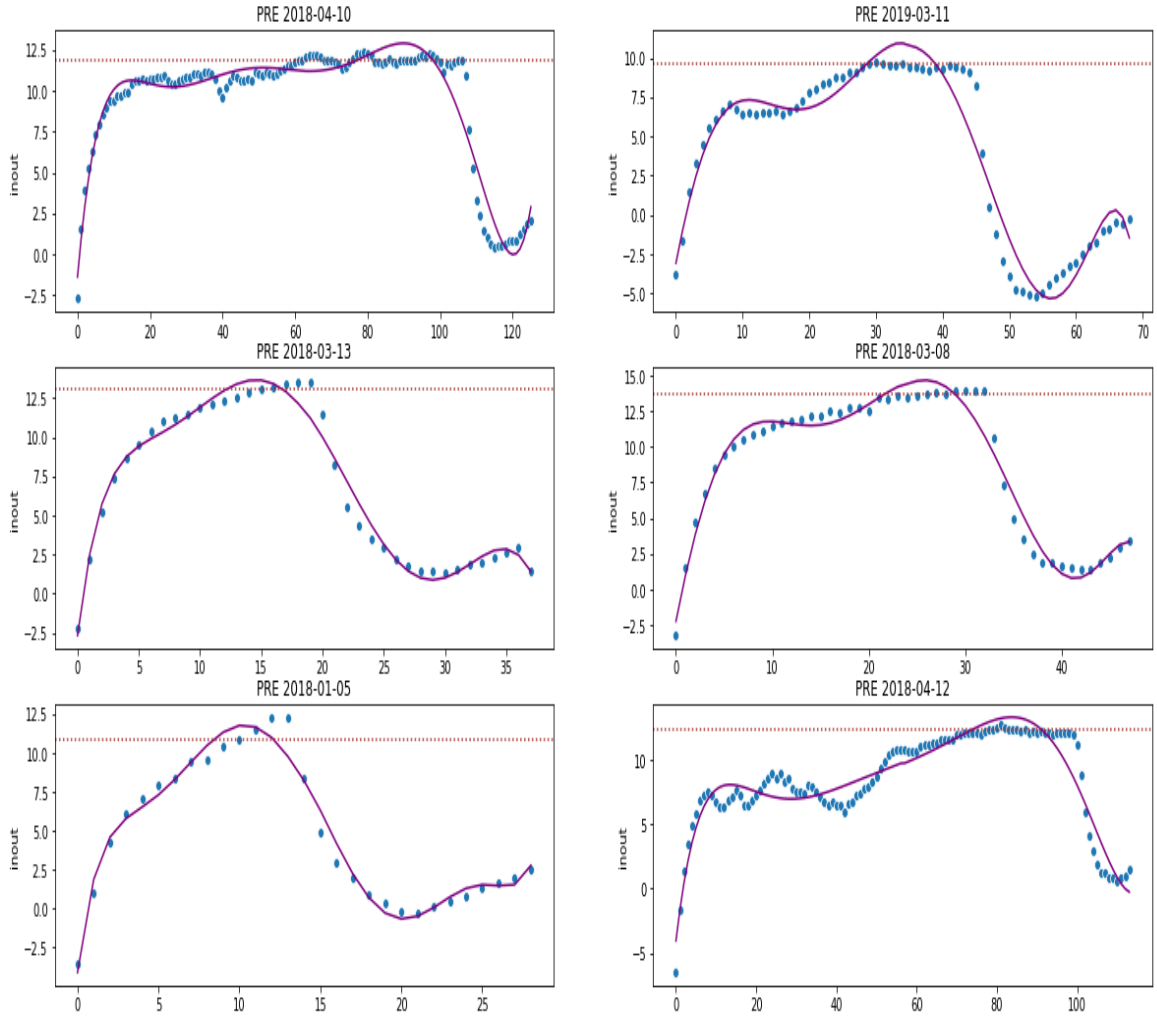

Figure 23. Preheating phase analysis of several days. X axis minutes, Y axis in-out temperature difference.

Figure 23 shows data for 6 random days. We can observe the same behavior for most of the days related to the inlet-outlet temperature differences. Values are over 10-12.5 degrees, however, there are some days that the duration of the process is less than 100 minutes, which could indicate that the operator stopped the preheating operation before due to for example that the fluid bed dryer already was preheated from a previous process. For instance, if we compare the days 04-10-2018 and 03-11-2019, it can be observed how the minutes duration is almost twice (x-axis). In the first case, 04-10-2018, the duration of the preheating process is around 100 minutes, which is when the blue dotted line begins to descend on the y axis, which indicates the difference in temperature between the air entering and the air leaving. It can be observed in both graphs that from minute 30, the temperature difference between the air that comes out and the one that

enters is stable and constant around 10-12 degrees, for which the fluid bed dryer is already in its optimal state of preheating, and it is ready for the drying process.

3.3.4. Evaluation of preheating phase for historical production

After key trends have been identified in selected sample days, the next step is to evaluate the preheating phase of all the 200 product batches available in the 700,000 signals. One year and a half fluid bed dryer sensor's data will be processed using histograms and box plot tools by analyzing the distribution of preheating times to determine if any outliers or patterns exist for the 200 product batches data.

A histogram is a graphical representation of the distribution of a set of numerical data. It is an estimate of the probability distribution of a continuous variable. The data is divided into a set of intervals (or "bins"), and the height of each bar represents the number of observations that fall within that interval. Histograms are used to visualize the distribution of data and to identify patterns and trends.

A box plot is a graphical representation of numerical data that provides information about the distribution of the data, including median, quartiles, and outliers. The box in the plot represents the interquartile range (IQR), which is the range of the middle 50% of the data. The line in the middle of the box represents the median, and the top and bottom of the box represent the first and third quartiles, respectively. The whiskers extending from the box show the range of the data, excluding outliers, which are plotted as individual points outside the whiskers. Box plots are useful for quickly visualizing the distribution of a dataset and identifying outliers.

3.3.4.1 Inlet – outlet air temperature analysis

First, we will create a histogram and a box plot to visualize the 200 batches inlet – outlet air temperature difference. To create a histogram, the range of sensor data is divided into intervals, or bins, of equal size (5 points each). The bins represent the inlet-outlet air temperature difference. Then, we count the number of batches that fall into each bin and

represent this count with a bar. The height of each bar represents the frequency of batches in that bin.

Second, we will create a box plot to verify the results. The box plot shows that the majority of the batches are between 12 and 15 inlet-outlet air temperature difference, with the median score being 12.5. There are no outliers in the sensor dataset related to inlet-outlet air temperature difference.

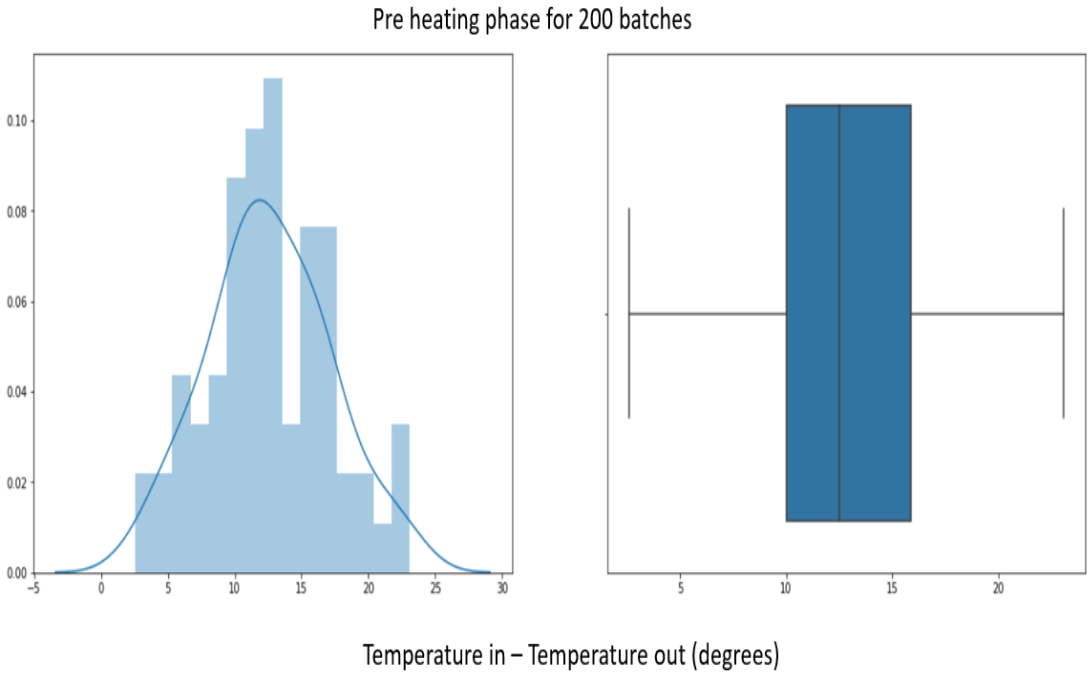

Figure 24. Temperature distribution during preheating for 200 batches of product: Histogram (left), Box plot (right)

Figure 24 shows the temperature distribution during preheating for 200 batches of product. We can see how the inlet and outlet temperature difference is distributed. This difference indicates when the fluid bed dryer is in its optimum preheating state, as it is not able to absorb more hot air. It can be observed that the median is around 12 degrees which is similar to the median of the previous commented sample figures.

3.3.4.2 Preheating duration analysis

Next, same steps will be followed to visualize the 200 batches data but instead of focusing on the inlet-outlet air temperature difference, we will focus on the preheating duration in minutes. Initially, we will use histograms and box plots to analyze time

duration in minutes of the 200 batches. The histogram will be created by dividing the sensor data range into 100 point or bins, which corresponds to preheating minutes duration, counting the number of batches in each bin, and showing the frequency with bar heights. The box plot will provide a visual representation of the majority of the batches being processed. Figure 25 shows the duration in minutes of the preheating phase for all the batches analyzed. The preheating duration varies mainly between 50.1 and 180.3 minutes, with the median being around 99.7 minutes.

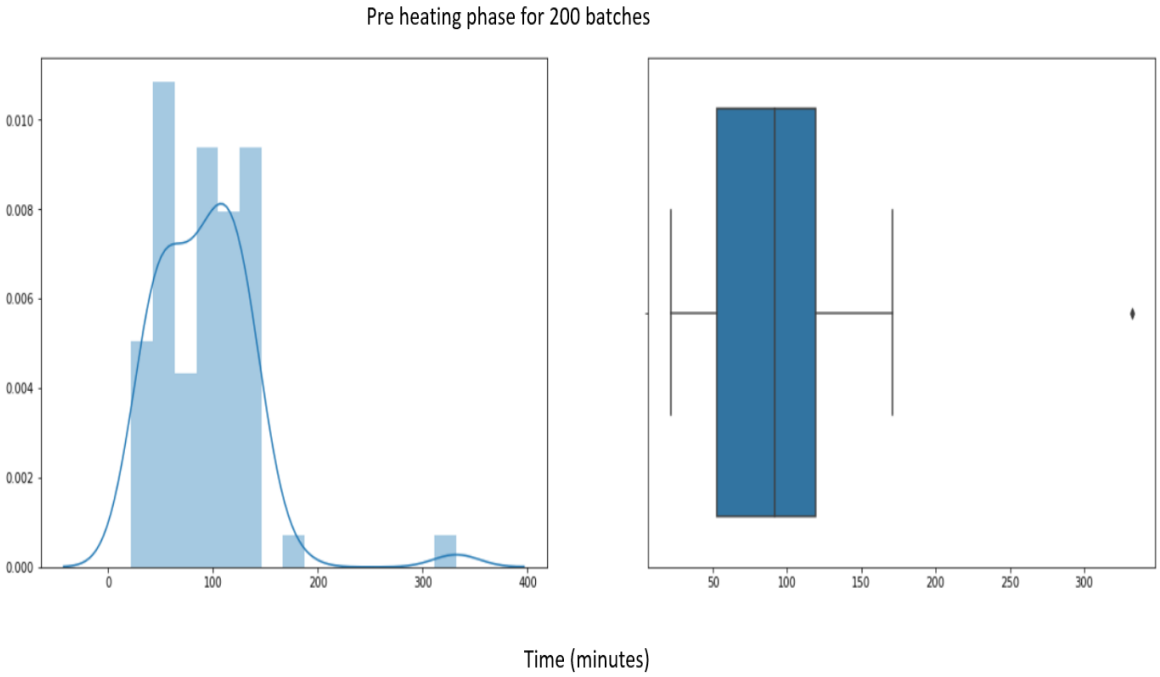

Figure 25. Distribution of preheating completion times for 200 product batches: Histogram (left), Box plot (right)

In Figure 26, it can be observed for the 200 batches analyzed, how many minutes on average the fluid bed dryer is used to perform the preheating process. Each line indicates for each individual batch the time taken to complete the preheating process in the fluid bed dryer. This time variability depends on when the operator has started and finished the preheating process. Since it is a manual process due to the age of the machine, the machine is kept for preheating less than 50.1 minutes, whereas other times, the machine is kept preheating for up to 180.3 minutes. The fluid bed dryer is initially set up with hot air inlet at 45 degrees and airflow 2000 m3/h. However, the fluid bed dryer doesn't have any sensor notifying when the machine is warm enough to introduce the drug product and start the drying process. Red dotted line in the graph indicates the average that is around 99.7 minutes duration to complete the preheating process. As summary, this indicates again the opportunity to harmonize the preheating process by establishing an

optimum preheating time, and potentially, to be able to reduce the preheating process time, and consequently reducing the fluid bed dryer energy consumption.

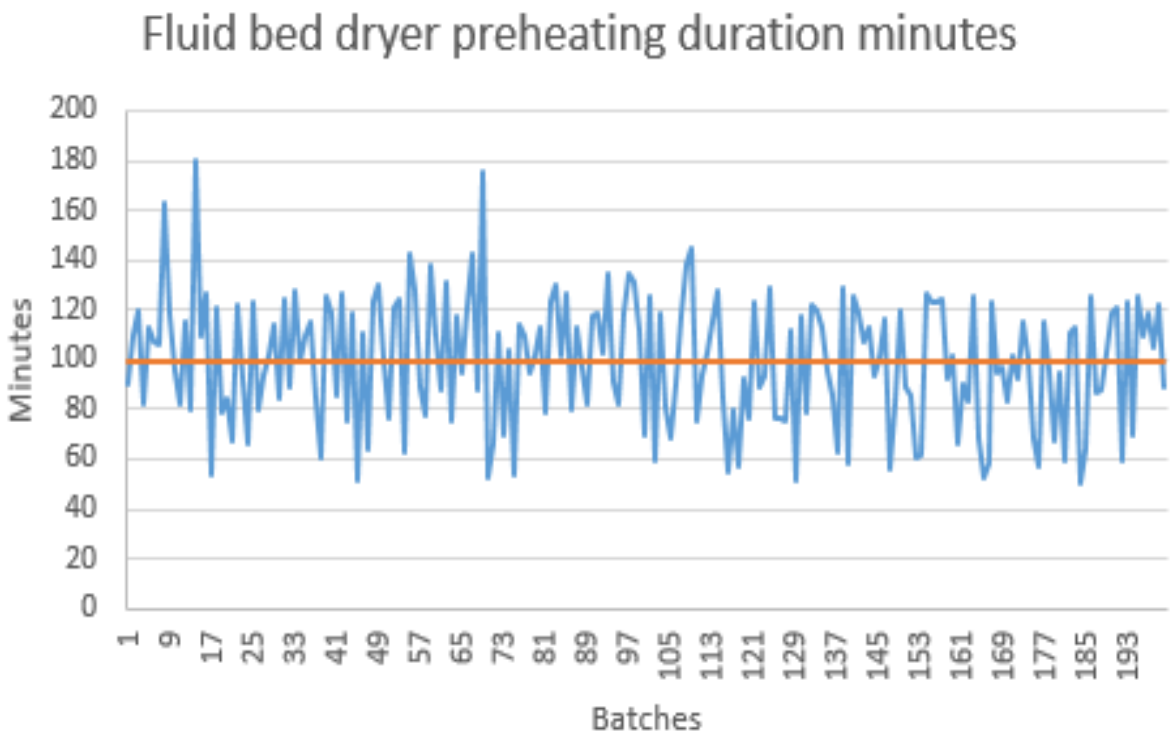

Figure 26. Fluid bed dryer preheating duration minutes.

3.3.4.3 Energy savings analysis

Figure 27 shows the variability of analyzing the energy consumption used to complete the preheating process for each batch in the fluid bed dryer. The energy consumption EC_b is calculated using the equation 3.

$$EC_b = Batch_t * \text{Cpm} \quad (3)$$

where $Batch_t$ is the time consumed by the fluid bed dryer for preheating the batch, and Cpm corresponds to the fluid bed dryer energy consumption per minute. The fluid bed dryer currently consumes 18.5 kWh during the preheating process, this means that for each minute it consumes 0.31 kWh (18.5 kWh / 60 minutes = 0.31kWh). If the preheating process may take between 50.1 and 180.3 minutes, therefore the fluid bed dryer consumes between 15.5 kWh and 55.8 kWh for preheating the machine to dry one batch of drug product.

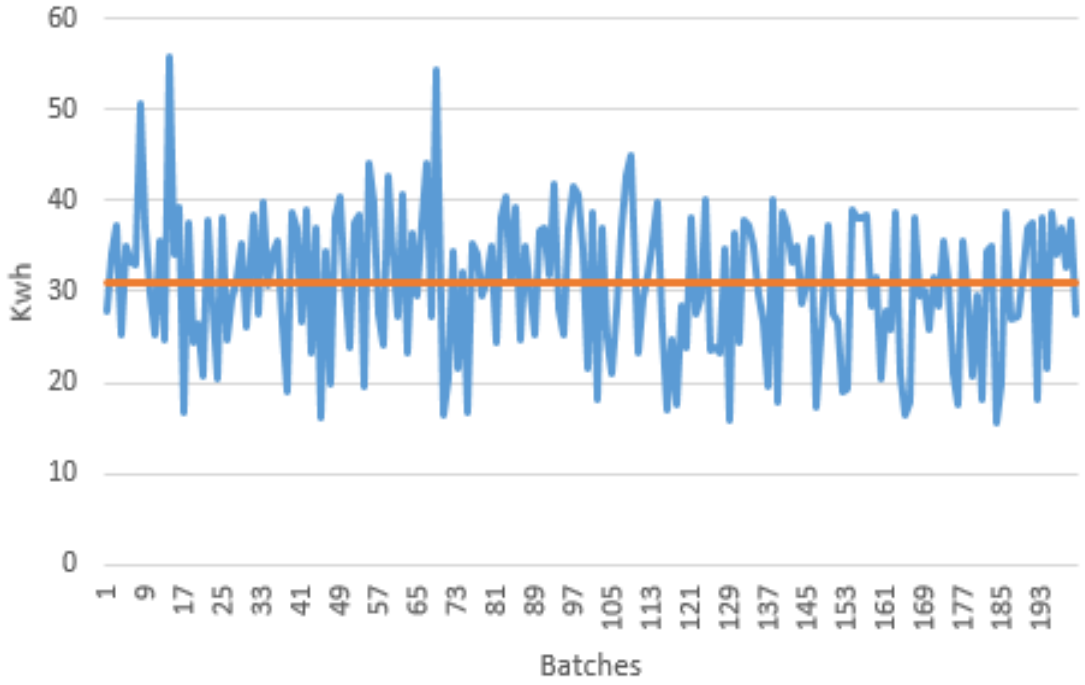

Figure 27. Fluid bed dryer preheating energy consumption (kWh)

It can be observed that some batches need 55.8 kWh, however, other batches need less than 15.5 kWh, which means in some cases around 72.2% less energy consumption for some batches. The dotted red line indicates the average consumption for the 200 batches, around 30.9 kWh. This indicates important potential energy savings if the preheating process in the fluid bed dryer is optimized. To calculate the potential energy savings of the fluid bed dryer during the preheating process for each batch, a machine learning model will be implemented, in the next chapter, to predict when the right time is to stop the process, and therefore, consume just the energy needed for preheating the fluid bed dryer.

3.4. Conclusions

This chapter has proposed an adapted exploration data analysis methodology to analyze and optimize a large-scale drug production process, such as the preheating drying process for pharmaceutical granules solid drugs through a fluid bed dryer.

It has been shown that once the 12.5º temperature difference between the inlet and outlet air is reached, the fluid bed dryer is at the correct temperature to proceed with the drying of the product. Therefore, from this point on, it is not necessary to continue

preheating, since this temperature is optimal for drying, going beyond this point implies a waste of time and energy consumption.

As a conclusion drawn from exploratory data analysis of the signals, it can be stated that the preheating phase lasts longer than necessary. Some batches need less than 50.1 minutes to complete the preheating process, however, there are batches that take up to 180.3 minutes. In terms of energy consumption, it means that for some batches the fluid bed dryer consumes 15.5 kWh, and for others is 55.8 kWh, which could represent savings, in some cases, of 72.2% of energy.

In the next chapter, we will develop a data model using machine learning algorithms to predict when the optimum time is to stop the fluid bed dryer for the preheating process, and we will calculate how much time and energy we are able to save.

4 Machine Learning model development

4.1. Proposed methodology

In this chapter, a machine learning model to reduce energy consumption of the fluid bed dryer preheating process, is proposed [Barriga et al., 2022]. The overall approach for data modeling and simulating follows a pipeline as illustrated in Figure 28 from left to right. First, a business needs and objective have to be defined. Checking and exploring the data steps have been detailed in chapter 3. In this chapter we will focus on creating and evaluating the results of the data model to predict the optimization of the preheating process. This method can in practice become a cyclic process iterating back from the results evaluation phase to the data obtaining phase, or even back to re-evaluate the business need. Next, we will explain briefly the steps:

Figure 28. Overall procedure for data analysis and modeling

- **Define business problem:** The initial phase of the machine learning workflow involves defining the business problem. The duration of this step varies, ranging from several days to a few weeks, depending on the complexity of the problem and its specific application. During this stage, data scientists collaborate with subject matter experts (SMEs) to gain a comprehensive understanding of the problem. This involves conducting interviews with key stakeholders, gathering pertinent information, and establishing overall project goals. In the case at hand, our objective is to minimize energy consumption in the fluid bed dryer.

- **Get the data:** Once the understanding of the problem is achieved, it is about getting the information identified and available for solving the business problem. In our case, we will use the data obtained from the fluid bed dryer directly.

- **Explore the data (EDA).** The next step in the process is data exploration (EDA), which involves analyzing the raw data. The primary objective of EDA is to delve into the data, evaluate its quality, identify any missing values, examine feature distributions, assess correlations, and so on.

- **Create the model:** Model creation encompasses various tasks, including dividing the data into training and testing sets, handling missing values, training multiple models, fine-tuning hyperparameters, consolidating models, evaluating performance metrics, and ultimately selecting the optimal model for deployment to forecast our target variable. In our specific scenario, we aim to predict the duration required for the preheating process in order to minimize energy consumption.

This chapter is devoted to the definition of the most suitable model to improve the energy consumption during the preheating process. Figure 29 shows the proposed approach to select and fine tune the machine learning model, consisting of: data selection, ML algorithm definition, hyper-parameter tuning, and finally the deployment of the predictive data model.

- **Select key variables:** In machine learning, variables or features are used to build models that can predict outcomes or classify data. Selecting key variables involves identifying which features are most relevant to the problem at hand and excluding those that are not. This can help to reduce the complexity of the model and improve its accuracy.

- **Benchmark algorithms:** Benchmarking involves testing and comparing the performance of different algorithms on a given task. This is useful to determine which algorithm is best suited for the problem at hand and provide a baseline for evaluating the performance of other algorithms.

- **Tune hyperparameters:** Hyperparameters are parameters that are set before training a machine learning model and can significantly impact its performance. Tuning involves adjusting these hyperparameters to optimize the model's

performance. This is typically done using techniques such as grid search, random search, or Bayesian optimization. The goal is to find the hyperparameters that produce the best results on a validation set.

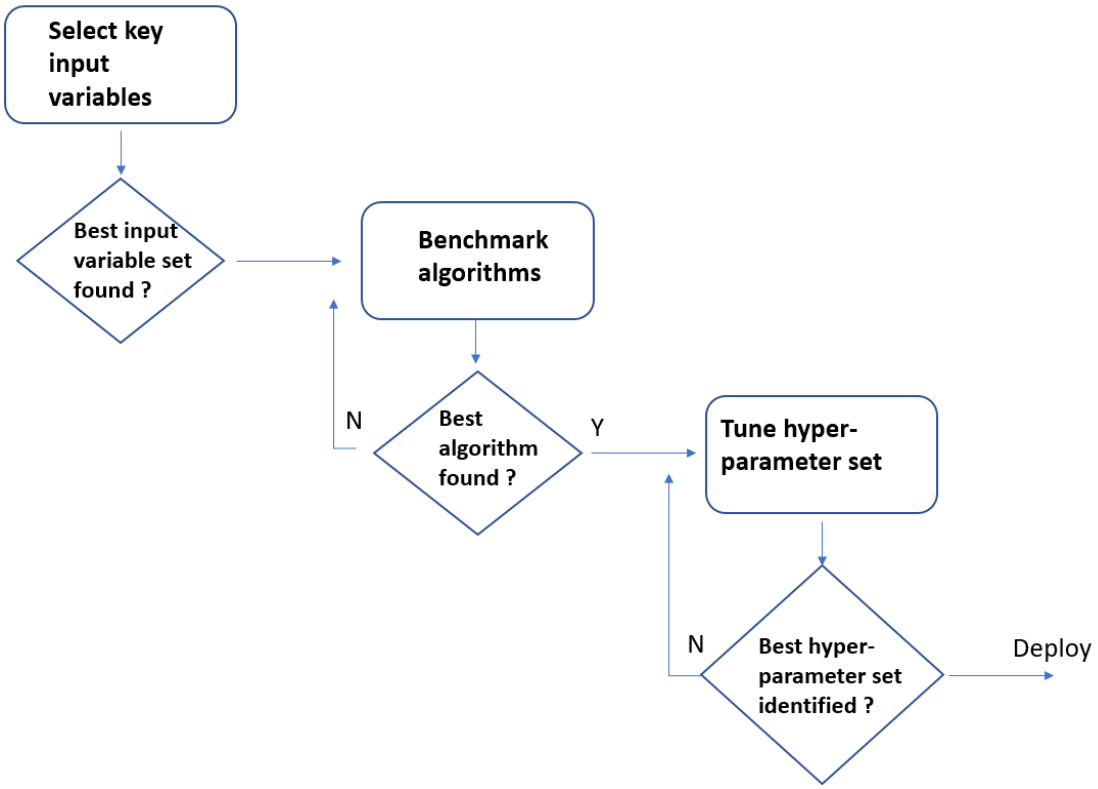

Figure 29. Proposed approach for data model

4.2. Data selection

One of the main challenges is that the fluid bed dryer does not incorporate sensors that indicate when the air inside is at the optimum temperature to finish the drying process due to the age of the machine. Besides, depending on the type of product to be treated, the drying time and the temperature, airflow, or humidity conditions vary. The operator manually analyzes the different parameters of the machine according to the formula of a corresponding product, as well as defines the time of the preheating, drying, and cooling processes.

From a data modeling point of view, the problem has several interesting features:

- The number of potential inputs is very high because multiple sensors are considered (56 sensors).

- The number of production batches is large, more than 200 batches of dried product, but the machine does not record the beginning or end of the drying process. Hence, the deduction is performed based on the temperature differences and air inlet and outlet of the machine.

- The objective is that the estimated model is interpretable to provide information on the sources of variability of the air inlet and outlet temperature difference curves. In this way, the estimation of the time required for the drying process can be performed.

In conceptual terms, a function model *f (i)* will be built to estimate the drying time through the data of a matrix *X* that contains the data extracted from the fluid bed dryer. The information available in the automatic learning models is used to predict the estimated drying time for each batch so that the operators can know when the optimal instant is to stop the machine's drying operation, with the consequent energy saving. The expected output from our model will be the remaining time that the fluid bed dryer needs to complete the drying process (based on the inlet-outlet temperature differences). It was applied data preprocessing techniques to the input dataset to reduce the unwanted information for the further analysis such as missing values.

The first step to select the most suitable model, is to split the data set into training and testing data. This technique is used for evaluating the performance of a machine learning algorithm. It can be used for classification or regression problems and can be used for any supervised learning algorithm. The process consists of taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset [ML].

The objective of splitting the dataset into train and test, is to estimate the performance of the machine learning model on new data that will be capture directly from the fluid bed dryer. Namely, to fit it on available data with known inputs and outputs, then make predictions on new examples in the future where we do not have the expected output or target values. The train-test procedure is appropriate when there is a sufficiently large dataset available, what means that there is enough data to split the dataset into train and test datasets and each of the train and test datasets are suitable representations of the

problem domain. A suitable representation of the problem domain means that there are enough records to cover all common cases and most uncommon cases in the domain.

```
In [12]: X = df[df.qcut_len>1].drop(columns=['inout_minutes'])
         y = df[df.qcut_len>1].inout_minutes
         X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=X[["FS3_GEA_BATCHN","qcut_minutes"]])
```

Figure 30. Splitting data set Code into training and testing datasets

Figure 30 indicates the used code to split our data set into training and testing. Our target value, the value which we are trying to predict, is *inout_minutes* which is time to optimal time when the fluid bed dryer should be stopped based on the in-out air temperature. The function takes the data, drops the target and saves it into *X*, which is our feature matrix, and it takes the data and only saves the target, which is *inout_minutes*, into the variable which we are trying to predict. Finally, it makes the split on these two data *X* and *Y*. The result is that we have *X_train, Y_train, X_test, Y_test*, where 80% of the data set is included in training, and 20% of the data will be used for testing. The *train_test_split()* makes a random split. Sometimes, rarely not always, it happens that while making this split the distribution of values of train and test changes. For instance, if the overall data has y which has the following distributions:

Y > 10 = 50% values
Y <= 10 = remaining 50% values

But after splitting, let us say by chance all the 20% test values come from *Y <= 10*. This result is not adequate because now the distribution of both train and test comes from different values. To solve these situations, we will use the function stratify:

stratify=X[["FS3_GEA_BATCHN","qcut_minutes"]

The Python function *"stratify*" is used in the context of splitting a dataset into training and testing sets, while ensuring that the proportion of a certain combination of variables remains the same in both sets. The function is using the columns *"FS3_GEA_BATCHN"* and *"qcut_minutes"* from the data *X* to create strata. Strata refers to subgroups or partitions of a population that share similar characteristics or attributes. The function will then assign the observations in *X* to different strata based on these two variables and split the data into training and testing sets, such that each set has a proportional representation of the different strata.

The column *FS3_GEA_BATCHN* is the batch identifier which means to perform a split where the distribution of batch identifiers for both train and test are the same. The second column qcut_*minutes* contains numerical values representing time intervals in minutes. The column has been created using the function *"qcut"*, which is used to bin a numerical variable into discrete intervals or quantiles. The data has been divided into equal-sized time intervals that contain an equal number of observations.

4.3. Machine learning algorithms benchmarking

Once we have split our data set into training and testing data, the next step is to find the most suitable ML algorithm to adapt to the problem of the fluid bed dryer. To this end, a list of ML models that will be benchmarked, is proposed in Table 5.

ID	Name	Reference
lr	Linear Regression	sklearn.linear_model._base.LinearRegression
lasso	Lasso Regression	sklearn.linear_model._coordinate_descent.Lasso
ridge	Ridge Regression	sklearn.linear_model._ridge.Ridge
en	Elastic Net	sklearn.linear_model._coordinate_descent.Elast...
lar	Least Angle Regression	sklearn.linear_model._least_angle.Lars
llar	Lasso Least Angle Regression	sklearn.linear_model._least_angle.LassoLars
omp	Orthogonal Matching Pursuit	sklearn.linear_model._omp.OrthogonalMatchingPu...
br	Bayesian Ridge	sklearn.linear_model._bayes.BayesianRidge
ard	Automatic Relevance Determination	sklearn.linear_model._bayes.ARDRegression
par	Passive Aggressive Regressor	sklearn.linear_model._passive_aggressive.Passi...
ransac	Random Sample Consensus	sklearn.linear_model._ransac.RANSACRegressor
tr	TheilSen Regressor	sklearn.linear_model._theil_sen.TheilSenRegressor
huber	Huber Regressor	sklearn.linear_model._huber.HuberRegressor
kr	Kernel Ridge	sklearn.kernel_ridge.KernelRidge
svm	Support Vector Regression	sklearn.svm._classes.SVR
knn	K Neighbors Regressor	sklearn.neighbors._regression.KNeighborsRegressor
dt	Decision Tree Regressor	sklearn.tree._classes.DecisionTreeRegressor
rf	Random Forest Regressor	sklearn.ensemble._forest.RandomForestRegressor
et	Extra Trees Regressor	sklearn.ensemble._forest.ExtraTreesRegressor
ada	AdaBoost Regressor	sklearn.ensemble._weight_boosting.AdaBoostRegr...
gbr	Gradient Boosting Regressor	sklearn.ensemble._gb.GradientBoostingRegressor
mlp	MLP Regressor	pycaret.internal.tunable.TunableMLPRegressor
xgboost	Extreme Gradient Boosting	xgboost.sklearn.XGBRegressor
lightgbm	Light Gradient Boosting Machine	lightgbm.sklearn.LGBMRegressor
catboost	CatBoost Regressor	catboost.core.CatBoostRegressor

Table 5. List of Machine learning algorithms

Firstly, some of the top state of the art ML algorithms shown in Table 5 will be briefly described: Random Forests, Lasso Regression, k-Nearest Neighbors and Ridge Regression in more detail, and a brief explanation for the rest of the models. A special focus will be performed in the explanation of Catboost algorithm because it will be the solution adopted in the present thesis.

- Random forests [Breiman, 2001] are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. Using a random selection of features to split each node yields error rates that are equal to other state of the art algorithms, but with the advantage that they are more robust with respect to noise.

- Lasso regression [Tibshirani, 1996] is based on estimation for linear models. LASSO (Least Absolute Shrinkage and Selection Operator) minimizes the residual sum of squares subject to the sum of the absolute value of the coefficients being less than a constant. LASSO solves two problems with standard regression techniques: (i) overfitting to outliers and (ii) overestimation of model performance based on variability.

- The k-nearest neighbors (KNN) algorithm [Fix et al., 1989] is a simple, supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but it has a performance drawback of becoming significantly slower as the size of the data grows. The KNN algorithm groups data records based on their" closeness" to each other, which is calculated by an appropriate distance metric. When independent variables in a multiple-regression model are highly correlated, ridge regression is a technique for predicting their coefficients.

- Ridge regression was developed as a possible solution to the imprecision of least square estimators when linear regression models have some multicollinear (highly correlated) independent variables. It creates a "ridge regression estimator" (RR), which provides a more precise ridge parameters estimate, as its variance and mean square estimator are often smaller than the least square estimators previously derived.

- Linear Regression is a linear model that seeks to find the relationship between a target variable and one or more predictor variables.

- Elastic Net is a linear model that combines the L1 (Lasso) and L2 (Ridge) penalties to balance between sparsity and smoothness in the solution.

- Least Angle Regression is a linear model that seeks to identify the most important predictors and add them to the model one at a time.

- Lasso Least Angle Regression is a combination of Lasso Regression and Least Angle Regression.

- Orthogonal Matching Pursuit is a linear model that selects a subset of predictors that are most correlated with the target variable.

- Bayesian Ridge is a linear model that applies Bayesian methods to the regression problem to estimate the posterior distribution of the regression coefficients.

- Automatic Relevance Determination is a linear model that applies Bayesian methods to the regression problem to estimate the relevance of each input variable to the target variable.

- Passive Aggressive Regressor is a linear model that updates the regression coefficients in an online manner to handle streaming data.

- Decision Tree Regressor is a decision tree that models the relationship between the target variable and input variables using a tree-like structure.

- Extra Trees Regressor is an extension of Random Forest that further randomizes the splitting process to increase diversity among the trees.

- Support Vector Machine (SVM) is a non-linear model that maps the input variables to a higher-dimensional feature space and finds a linear boundary between the classes.

- Gradient Boosting Regressor is a decision tree ensemble method that sequentially fits new trees to the residual errors of the previous trees to improve accuracy.

- Extreme Gradient Boosting: An extension of Gradient Boosting that incorporates additional regularization to reduce overfitting.

- Light Gradient Boosting Machine is a highly optimized implementation of Gradient Boosting that uses histogram-based algorithms to speed up the computation.

- AdaBoost Regressor is an ensemble method that fits a sequence of weak learners to the training set, with each new learner focusing on the misclassified observations of the previous learners.

- Extra Boost Regressor is an extension of AdaBoost that adds a random component to the weight updates of the training set.

4.4. Catboost algorithm

Catboost Regression is a recent and reportedly powerful new machine learning based algorithm with numerous advantages [Prokhorenkova et al., 2017]. In general, machine learning algorithms are applied to identify complex patterns in large volumes of data to predict future behaviors. Catboost is a technique for decision trees that uses gradient boosting. For regression and classification problems, gradient boosting is a machine learning technique that generates a prediction model in the form of a group of "weak prediction models," often decision trees [Liu, 2022]. The overall idea is to apply a steepest descent step to a minimization problem (functional gradient descent). A gradient boosting procedure iteratively builds a sequence of approximations Ft: Rm→R, t = 0,1, … in a greedy fashion. Thus, Ft is obtained from the previous approximation Ft-1 in an additive manner: Ft = Ft-1+αht, where α is a step size and function ht: Rm→R, known as a base predictor, is chosen from a family of functions H such that it minimizes the expected loss ht. Catboost is an implementation of gradient boosting using binary decision trees as the function h(x), which is defined as

$$h(x) = \sum_{j=1}^{J} b_j l_{\{x \in R_j\}}$$

where Rj are the disjoint regions corresponding to the leaves of the tree and $b_j l_{\{x \in R_j\}}$ is the jth binary variable corresponding to attribute x.

One key innovation of the Catboost implementation is that it can process mixed data types together to build a model. That is categorical (converts to numbers) and numerical inputs. Other strong points are (i) how its default hyper-parameters require very little tuning - they work well for the majority of data scenarios and (ii) auto-correction for overfitting.

Application of Catboost to the data: In order to mitigate the increase of the model size and the memory consumption, the following measures were taken through the assignment of the meta-parameters:

- RAM limit - limit value to restrict memory usage
- Set max_ctr_complexity to 1 or 2. Default values is 4.
- Model_size_reg assigned a bigger value to penalize heavy combinations.

Memory usage is indeed the major limitation of Catboost currently. Catboost demands that all data be immediately accessible in memory for quick random sampling, unlike stochastic gradient and neural network models. This can be mitigated using for batch training the following configuration:

Random subspace method is the percentage of features to use at each split selection, when features are selected repeatedly at random. Another consideration is to introduce random subspaces along rows, with a similar approach to a rolling-tree generator, with the following steps:

1) Read N initial rows from pool
2) Generate M trees
3) Discard first k<N rows, read next k rows from pool, return to step 2.

where N is user defined, and M and k are deducted from the total number of rows and total number of iterations, respectively This could also be applied to other techniques such as random forests, gradient-boosting trees. Another important issue is the sensitivity of Catboost to hyper-parameters and the importance of hyper-parameter tuning. This can also be dependent on the Big Data environment, such as the Apache

Spark distributed framework [Markarian, 2018]. The hyper-parameter tuning details are given later. For use with very large datasets, the Catboost model can be fit to a representative sample using the Catboost Python API, then applied to the larger dataset using Spark or Hadoop [Nettleton et al., 2018] with Catboost's Java API.

4.5. Evaluation of ML algorithms

To perform the evaluation and selection of the best fit algorithm for the fluid bed dryer process we used the Phyton libraries [Barriga, 2021]. The same dataset has been injected in the different algorithms. The dataset contains 18 months data coming from the 56 sensors of the fluid bed dryer and the values represent the average of 10-fold cross validation (partitioning of the data set into 10 parts, 9 for train and one for test, then rotating 10 times to obtain different combinations of partitions). The results of the most relevant algorithm's evaluation are shown in Table 6.

Model	**MAE**	**MSE**	**RMSE**	**R2**
Catboost Regressor	8.1453	130.1740	11.2289	0.7079
Extra Trees Regressor	8.6712	142.9386	11.8076	0.6779
Extreme Gradient Boosting	9.1954	166.0124	12.7120	0.6246
Light Gradient Boosting Machine	9.5825	171.4744	12.9023	0.6138
Gradient Boosting Regressor	10.4752	189.7955	13.5794	0.5689
Random Forest Regressor	10.8158	205.1762	14.1438	0.5397
K Neighbors Regressor	13.0786	280.3520	16.5922	0.3520
AdaBoost Regressor	14.0844	290.1670	16.9228	0.3361
Decision Tree Regressor	12.8016	361.9191	18.5659	0.1620
Lasso Regression	16.0411	388.7161	19.5774	0.1198
Elastic Net	16.3309	392.0491	19.6636	0.1135
Orthogonal Matching Pursuit	16.0035	393.0683	19.6999	0.1045
Bayesian Ridge	16.4593	403.1359	19.9437	0.0854

Table 6. Benchmarking results of different machine learning and statistical techniques on the dataset

Regarding the assessment, four contrasting evaluation metrics were calculated for each test. The metric is one single value which provide the performance of the model. By itself,

it has no worth but when there are two metric values, both can be compared. So, metrics are useful only when they are available for many algorithms to be compared.

In regression, the basic objective is to predict the observations as closely to the true values as possible. Let us say I have one model which predicts one observation as 14 while the true is 13. We can just take the difference between these two and say metric = 14 - 13 = 1, now, let us say we have another model which predicts for the same observations, 13.7, here, the same metric will be 0.7. Hence, the second model wins because it has low error.

The four metrics that analyzed are: MAE (Mean Absolute Error), MSE (Mean Squared Error), RMSE (Root Mean Squared Error) and R2.

- MAE (Mean Absolute Error) is a metric used to evaluate the average difference between the predicted and actual values. It is calculated as the mean of the absolute differences between each predicted and actual value. It is used to measure the accuracy of regression models.

- MSE (Mean Squared Error) is another metric to evaluate the performance of regression models. It measures the average of the squared differences between the predicted and actual values. It penalizes large errors more heavily than small ones.

- RMSE (Root Mean Squared Error) is the square root of MSE. It provides the same unit of measurement as the dependent variable and is a more interpretable metric. It also penalizes large errors more heavily than small ones.

- R2 (R-squared) is a metric used to evaluate the goodness of fit of a regression model. It represents the proportion of the variance in the dependent variable that is explained by the independent variables. It ranges from 0 to 1, with 1 indicating a perfect fit and 0 indicating no linear relationship between the dependent and independent variables. The R2 value has the advantage that it is scale free, a negative value indicates the model is worse than predicting the average.

Based on Table 6, the Catboost Regressor has the lowest MAE of 8.1453 and the lowest RMSE of 11.2289, indicating that it has the best predictive accuracy compared to the other models. It also has the highest R2 value of 0.7079, indicating that it can explain about 70.79% of the variance in the target variable. The Extra Trees Regressor has the second-best performance, with slightly higher MAE and RMSE values than the Catboost model, and an R2value of 0.6779. The Extreme Gradient Boosting, Light Gradient Boosting Machine, and Gradient Boosting Regressor models have higher MAE, MSE, and RMSE values and lower R2values than the Catboost and Extra Trees Regressor models, indicating that they may not perform as well on this specific dataset, same with the rest of the models. To select the best metric for the Catboost algorithm, it is considered the nature of the problem and the evaluation criteria. To measure the proportion of variance in the target variable that can be explained by the model, R2 is the most suitable metric. MAE has been discarded because focus on minimizing the average absolute difference between predicted and actual values, and MSE or RMSE penalize larger errors more than smaller errors.

Let us focus in R2 parameter which is actually quite an important metric but should not be looked at alone. R2 is the percentage of variance described by the model. So, if R2 is 0.7, it means that our current model is able to explain the 70% variation in the value which we are trying to predict and leaving the 30% to random causes. It should not be looked at alone because sometimes it may happen that although R2 is large but other error metrics are also large. Also, R2 can simply increase by adding redundant features which does not help the model. Ideal values of errors should be 0 while R2 should be 1.

4.6. Catboost adaptation to the fluid bed dyer

4.6.1. Catboost model configuration

In this section, the adaptation of the Catboost ML algorithm to the fluid bed dryer process and the evaluation of the quality of the model using a set of observational data of the machine, is performed. The variable to predict the performance is the time remaining to finish the drying process based on temperature differences of inlet and outlet air. The data set that will be used is the same data set used in chapter 3 for the exploratory data analysis. The fluid bed dryer data from the 56 sensors that measure inlet / outlet air

temperature, air flow in m3/h, motor rotation speed, and air pressure. Each sensor collects data minute by minute. We have one year and a half worth of data, which is equivalent to more than 700,000 readings. Due to the high volume of data (more than 3GB of data), we have selected the Azure platform and its advanced analytics module Databricks using Python for data analysis of the Catboost model.

Firstly, Catboost model was run using default parameter settings, giving the results shown in Table 7. It is applied to the 10-fold cross validation. The 0 to 9 are just interval partitions of the train data to evaluate the models. So, it can be seen as fitting 10 different Catboost models on some subset of train data and getting results and storing them, then the final result will be the mean of all these 10 runs. These 10 runs in ML terminology are known as “cross Validations”. Cross validation is a technique where training data is further splitted into two parts. One part of data is used to actually train the model and another part is used to calculate the cross-validation score. But it is performed multiple times [Barriga BER, 2019]. For example, in the experiment, it was done 10 times which means that at each time, some parts of training data were used to train and while the remaining part was used as cross validation data and the score shown in the Table 7 is actually a cross validation score.

Table 7 shows the results of a 10-fold cross-validation of the Catboost model, with metrics including mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), and R-squared (R2). The table shows the results for each of the 10 iterations, as well as the mean and standard deviation (SD) of the metrics across all iterations. The R2 column in this cross-validation table shows the coefficient of determination, which is a measure of how well the model fits the data. The R2 values range from 0 to 1, where a value of 1 indicates a perfect fit and a value of 0 indicates that the model does not explain any of the variability in the data. In this table, the R2 values range from 0.5315 to 0.7829, with a mean R2 value of 0.7079 and a standard deviation of 0.0805. These values suggest that the model has a good fit with the data, with some variability in performance across the different folds of the cross-validation.

	MAE	MSE	RMSE	R2
0	65.538	837.852	91.534	0.7603
1	93.913	1.714.283	130.931	0.6881
2	109.114	2.508.916	158.396	0.5883
3	87.544	1.540.916	124.134	0.5315
4	75.330	908.290	95.304	0.7005
5	83.647	1.165.782	107.971	0.7444
6	60.740	849.854	92.188	0.7602
7	82.784	1.279.620	113.120	0.7415
8	86.285	1.321.672	114.964	0.7829
9	69.633	890.218	94.351	0.7816
Mean	81.453	1.301.740	112.289	0.7079
SD	13.590	494.731	20.212	0.0805

Table 7. Benchmarking of Catboost with 10-fold cross validation

4.6.2. Analysis of Catboost model

In this subsection, the analysis of the results using residuals, prediction error and learning curve graphs, is performed. A residual plot is a graph that displays the residuals on the vertical axis and the independent variable on the horizontal axis.

The difference between the observed value of the dependent variable (y) and the predicted value (ŷ) is called the residual (e). If the points on a residual plot are randomly scattered around the horizontal axis, a linear regression model is appropriate for the data; otherwise, a nonlinear model is more appropriate. The ideal graph should be a graph where all points are on a horizontal line around zero i.e Residuals should be small.

For our case, we observe in Figure 31, on the x-axis, the predicted value that is computed by our model. On the x-axis, we plot the residuals. On the right of the Figure 32, we plot the histogram of the residuals. These values are centered around zero (an ideal case), but sometimes we also have situations where some values are far away from the right and left. Besides, it can be seen, in green, that the distribution of most of the residuals for the test data, fall between -10 and 10, which means that the selected model is correct.

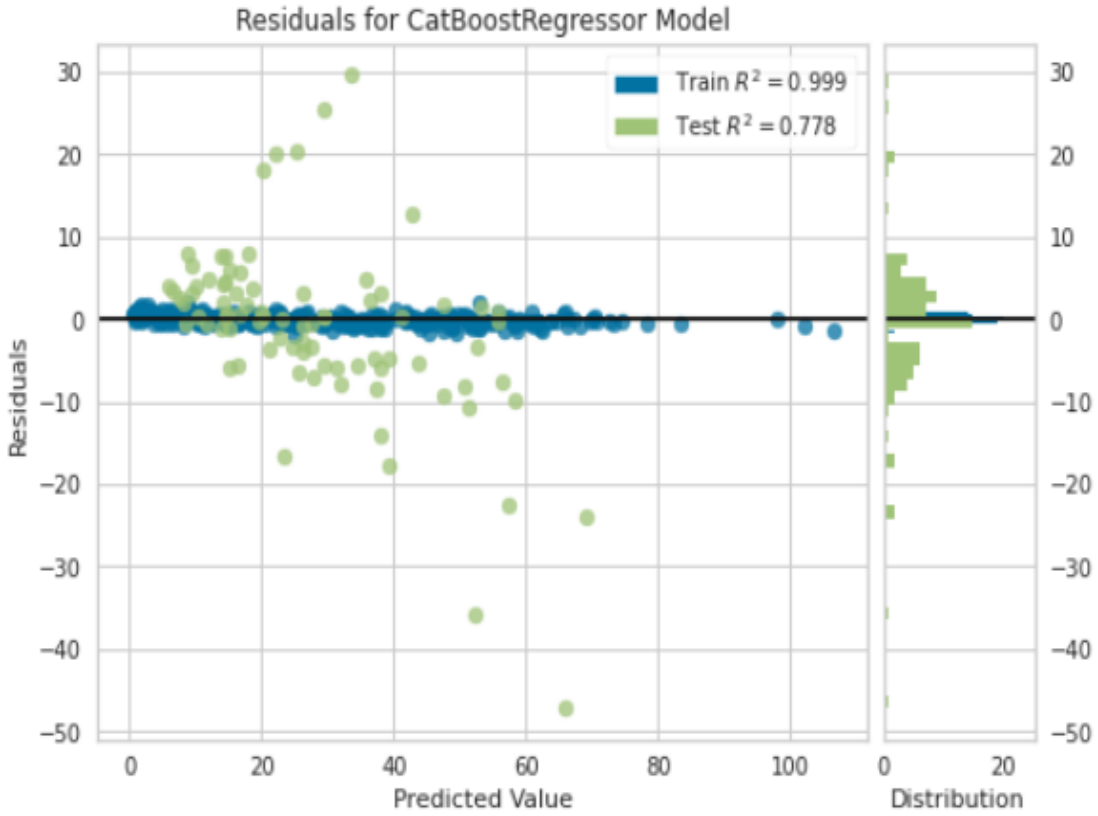

Figure 31. Distribution of residuals for CatboostRegressor model

Comparing the distribution of residuals for CatboostRegressor with a LinearRegression model (see Figure 32), it can be observed that the linear regression model is just too good to be true. We can see the graph of predicted vs actual values that they are in perfect line. This situation indicates that overfitting could arise. Overfitting is a simple phenomenon where the model performs best in a training set while it performs poorly on the test data.

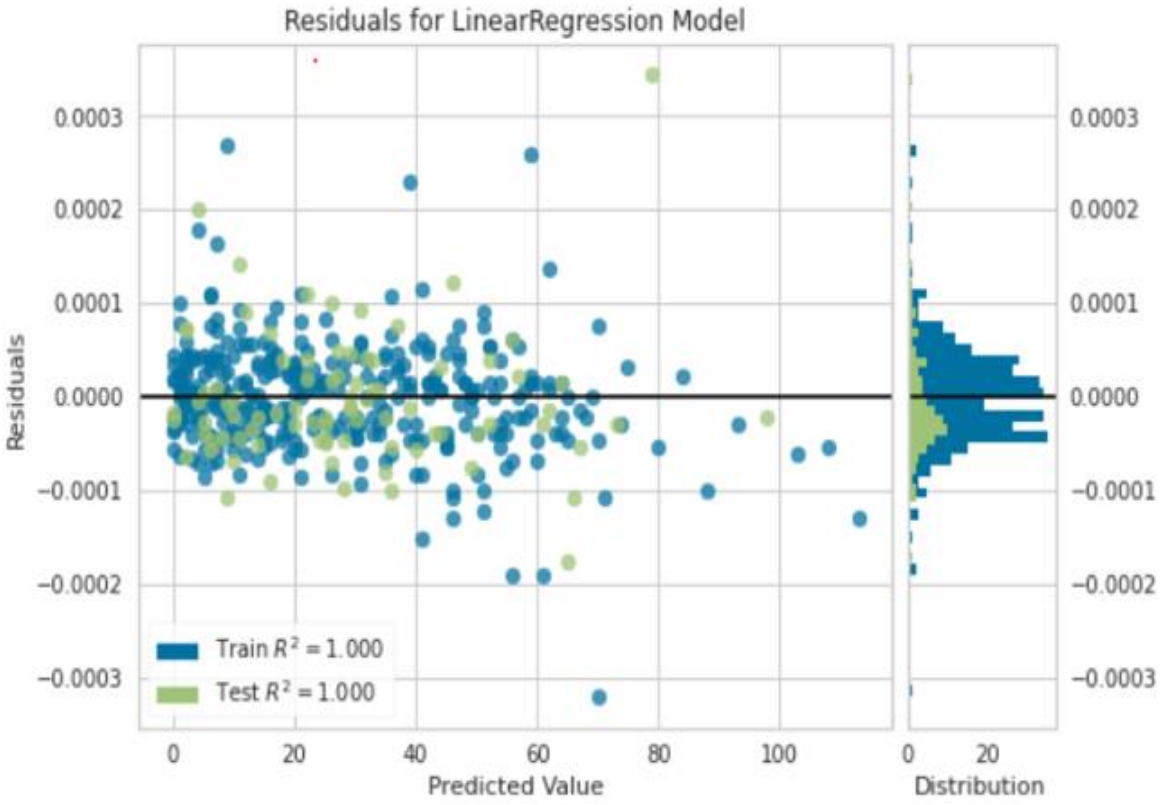

Figure 32. Distribution of residuals for LinearRegressor model

The second parameter used to evaluate our model is the prediction error. Prediction error is a method to check what is wrong with a machine learning model. Typically, when we build a regression model, we are concerned with the error metric that describes how well it fits the data. In our case, we use the root mean square error parameter (MAE). This parameter informs us that the mean squared error has an approximate value X, this is a very precise description of the error. Let us say the actual *inout_minutes* (temp) are 60 and against that our model has predicted 70.8, then we will plot these two pairs on x and y axis respectively. Ideally, since both predicted and actual values should be equal, this plot should form a straight line. The further away this plot is from the diagonal line, the worse the model is. Figure 33 shows the prediction error of the Catboost model. An R-squared (R2) value of 0.778 for a CatBoostRegressor model indicates that the model explains approximately 77.8% of the variance in the target variable. In other words, the model is able to capture and predict about 78% of the total variation in the data. The remaining 22% of the variation in the target variable is not captured by the model and is considered to be the prediction error.

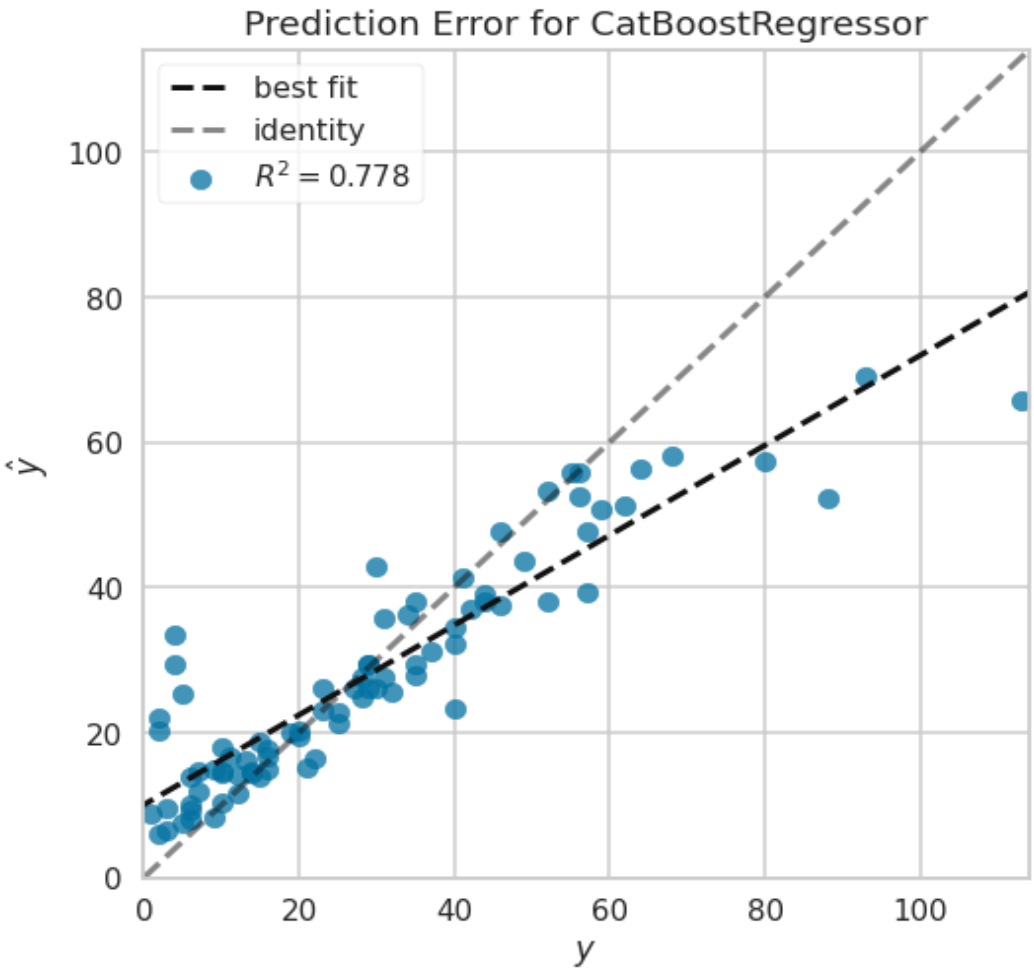

Figure 33. Distribution of prediction error for CatboostRegressor model

Another used technique is the learning curve. This technique represents a graph relating the number of iterations and score on a training and testing set. It is a measure to find out the optimal number of iterations required to get a good score. When the model training is performed, the model starts with the first iteration and then we get a certain score. Let us say with the first iteration we got R2 of 0.1 and with the second we got 0.12 and this way we will keep on training until the model starts to have a flattened score, which essentially means that the score is no more increasing with the increase of number of iterations. So, the learning curve plots the number of iterations and so on x-axis, and corresponding scores on y-axis. This is important to avoid the problem of overfitting and at the same time saving the computational resources.

Figure 34 explains the learning curve for the Catboost model. It can be seen how the test precision behaves (y axis) with respect to an increasing number of training instances (x axis), which reaches a maximum of 0.7 for 300 instances, what means that as the model

is taking more data, the learning curve is improving, and the prediction will be more accurate.

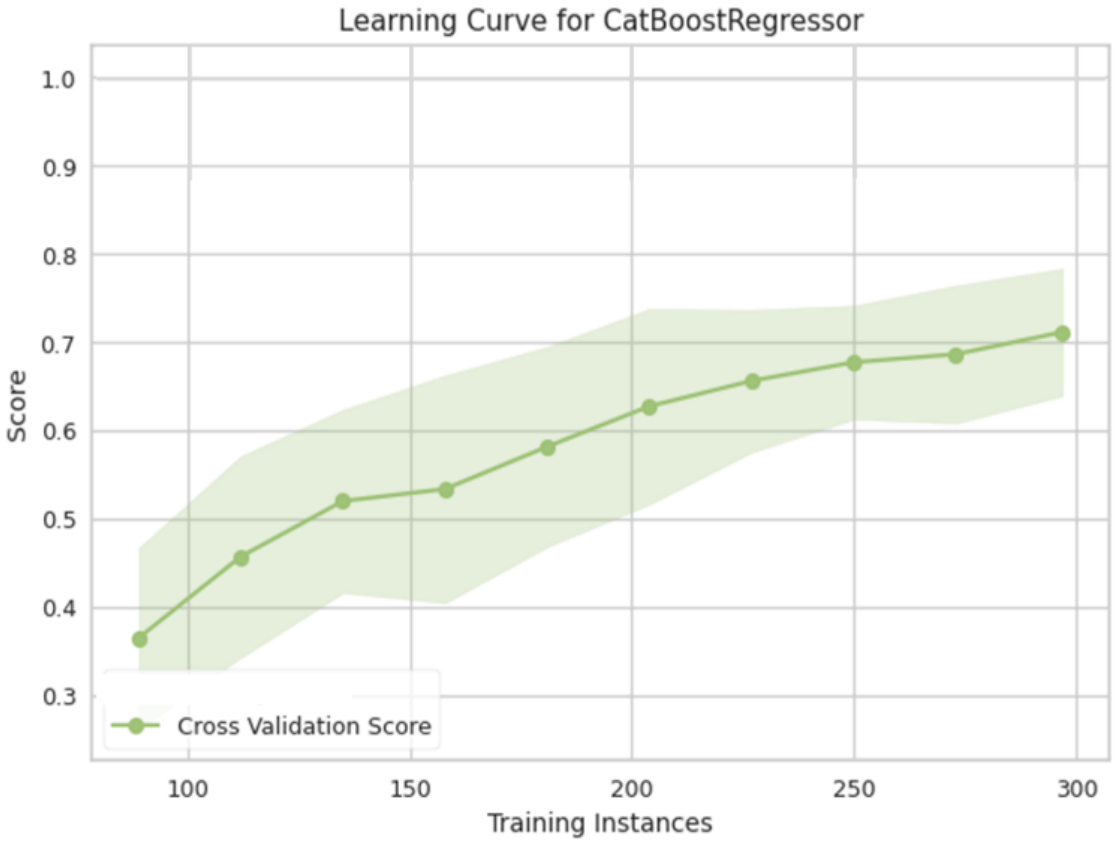

Figure 34. Distribution of learning curve for CastBoostRegressor

The shap plot objective, Figure 35, is used to measure the feature importance of the Catboost model. On the right, we have been given the scale of importance. The blue indicates low while red indicates high importance. A feature which has more reds than blues will be considered very important for the model. Each feature has some blues and some reds. More red points in the feature indicate its importance in predicting the target which is *inout_minutes* temperature.

If we need the exact measure of which feature is most important in terms of numerical values, we need to run a different function to get shap numerical values. Let us try to understand one of the features, say min15_mean_inout. There appears to be more blues, more on the positive side. It means this feature has a more positive side response on the target what is the prediction of the pending time to finish the fluid bed dryer preheating process.

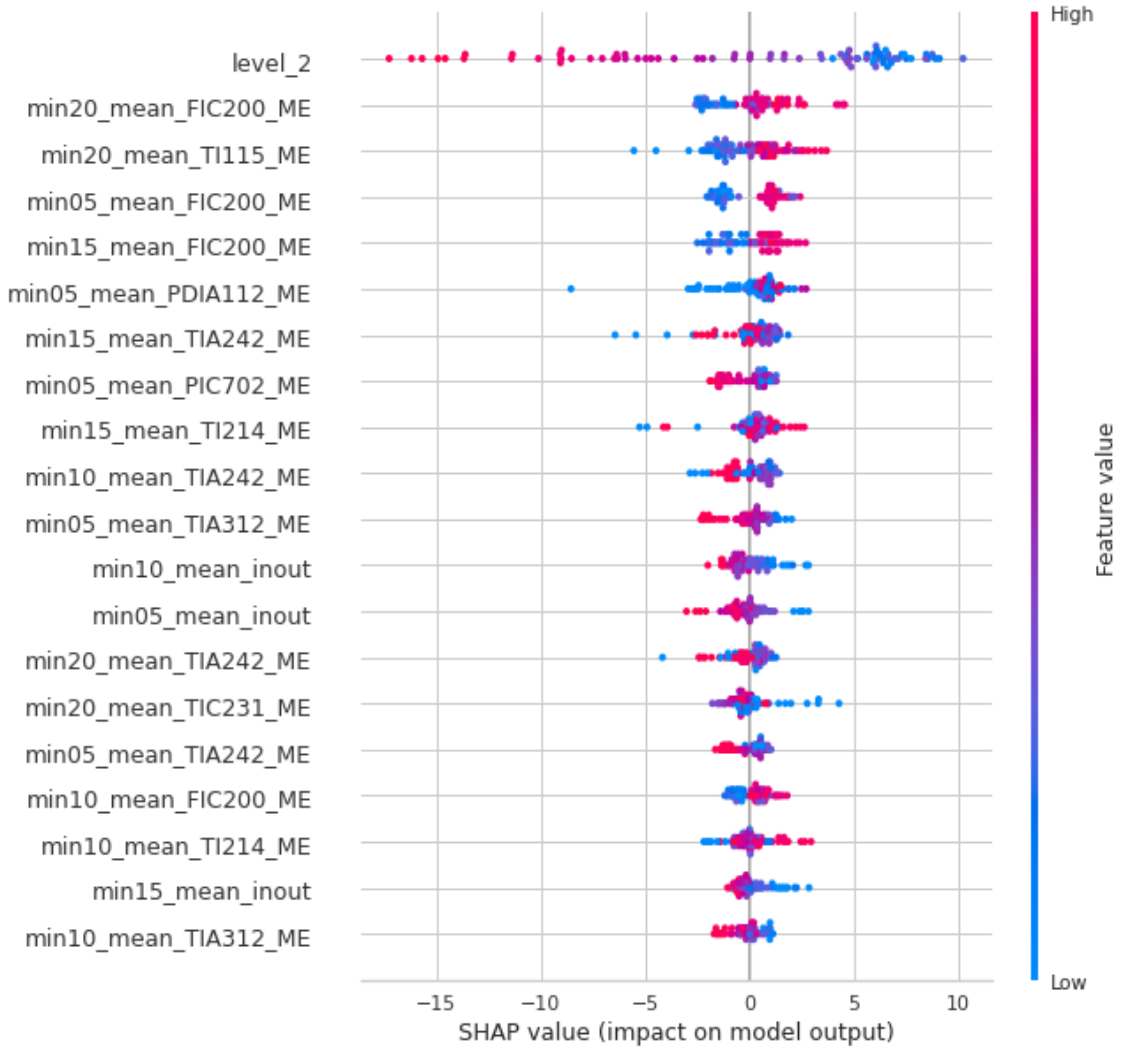

Figure 35. Shap plot

Next step is to apply hyper-parameter tuning to check if there is a better set of optimal hyperparameters for improving the learning Catboost algorithm accuracy. For this purpose, it is implemented *hyperoptfitter* algorithm shown in Figure 36.

hyperoptfitter

```
In [7]: from catboost import CatBoostRegressor
        from src.models.hyperopt_fitter import HyperoptFitterMinutes

In [26]: dp = df[df.inout_minutes<70]
         X = dp.drop(columns=['inout_minutes'])
         y = dp.inout_minutes

In [27]: hpf = HyperoptFitterMinutes(X, y, "r_squared")
         hpf.fit(max_evals=80)

         100%|██████████| 80/80 [37:13<00:00, 27.92s/trial, best loss: -0.6527323624318435]

Out[27]: HyperoptBestResult(metric=Metric(name='r_squared', value=0.6527323624318435), model=Pipeline(memory=None,
                 steps=[('scaler',
                         StandardScaler(copy=True, with_mean=True, with_std=True)),
                        ('fillNA',
                         SimpleImputer(add_indicator=False, copy=True,
                                       fill_value='mean', missing_values=nan,
                                       strategy='mean', verbose=0)),
                        ('varZero', VarianceThreshold(threshold=0.0)),
                        ('clf',
                         <catboost.core.CatBoostRegressor object at 0x7fb2b3f31190>)],
                 verbose=False), parameters={'border_count': 179, 'iterations': 8893, 'l2_leaf_reg': 2, 'learning_rate': 0.004568720596
         627703, 'min_data_in_leaf': 14.0, 'random_strength': 2.6145016765190943, 'type': 'cb'})

In [11]: model = hpf.train_best_model()
```

Figure 36. Hyperoptfitter algorithm

This code imports the CatBoostRegressor class from the CatBoost library and a custom-defined HyperoptFitterMinutes class from the src.models.hyperopt_fitter module. It then loads a dataset from a *pandas* data frame, filters out rows where the "*inout_minutes*" column is less than 70, and separates the features (X) from the target variable (y). The HyperoptFitterMinutes object is initialized with X, y, and the name of the metric to optimize (r_squared). The fit method of the HyperoptFitterMinutes object is called with a maximum number of evaluations to run (80). This uses Bayesian optimization to find the best hyperparameters for the CatBoostRegressor model, and the best result is printed, including the metric value and the hyperparameters used. Finally, the best model is trained using the hyperparameters found by the optimization process and saved in the "model" variable.

In summary, this code will take an *eval_metric* (R2 in this case) and try to find out that combination of hyperparameters which gives us maximum R2. Some of the important hyperparameters for Catboost algorithm are:

- Number of trees
- Number of interactions
- Max_leaves
- Learning Rate
- Tree Depth
- Border Count
- Tree growing policy

Some of these hyperparameters are explained below:

- **Number of trees**: This hyperparameter decides how many trees we train in a cat boost algorithm since it is a tree-based algorithm which fits many decisions tree and uses each one to boost the previous model's results.

- **Max Depth:** We know that the Catboost model is just a collection of trees boosted together. We know that a decision tree is formed by asking a series of questions and keeping splitting the data.? How many questions should I ask for a decision tree to split my data? This is defined by the maximum depth of each tree.

- **Samples required to make a split**: We looked at the *max_depth* in the previous point. Now, consider a situation where we keep asking 10 questions and we are in a case where the data has been splitted so much that only 1 data. In that case, our leaf node has 1 data point and we will be predicting based on that one point only. One sample cannot be a representative of a group. Hence, we should have a way to specify a minimum number of samples required to make a split. If the number of samples go below that, we stop splitting that node and that one becomes a leaf node.

- **Learning Rate:** It is a measure of how quickly the learning happens. If you fix a large value for this, then with few iterations only, we can converge faster. But it should be tweaked very carefully because the minima can be overshot by a large learning rate.

HYPERMARAMETER	*Value*
'sampling_frequency'	'PerTree'
'leaf_estimation_method'	'Newton'
'grow_policy'	'SymmetricTree'
'penalties_coefficient'	1
'boosting_type'	'Plain'
'model_shrink_mode'	'Constant'
'feature_border_type'	'GreedyLogSum'
'bayesian_matrix_reg'	0.10000000149011612
'l2_leaf_reg'	3
'rsm'	1
'boost_from_average'	True
'model_size_reg'	0.5
'depth'	6
'posterior_sampling'	False
'border_count'	254
'sparse_features_conflict_fraction'	0
'leaf_estimation_backtracking'	'AnyImprovement'
'best_model_min_trees'	1
'model_shrink_rate'	0
'min_data_in_leaf'	1
'loss_function'	'RMSE'
'learning_rate'	0.032058000564575195
'score_function'	'Cosine'
'leaf_estimation_iterations'	1
'bootstrap_type'	'MVS'
'max_leaves'	64.
'learning_rate'	.032058000564575195
'score_function'	'Cosine'
'leaf_estimation_iterations'	1
'bootstrap_type'	'MVS'
'max_leaves'	64.

Table 8. HYPER-PARAMETER TUNNING RESULTS

A brief explanation of the results of Table 8 is provided below:

- sampling_frequency: This hyperparameter specifies the sampling frequency for features when building decision trees. It can take on different values depending on the specific implementation of the Catboost algorithm.

- leaf_estimation_method: This hyperparameter specifies the method used to estimate the values of leaf nodes in decision trees. In the table, the value 'Newton' indicates that the algorithm uses a Newton-Raphson method to estimate the leaf values.

- grow_policy: This hyperparameter controls the strategy used to grow decision trees. The value 'SymmetricTree' indicates that the algorithm grows symmetric decision trees, where each leaf node has the same depth.

- penalties_coefficient: This hyperparameter is a penalty coefficient used in the algorithm's objective function. In the table, the value of 1 indicates that there is no penalty.

- boosting_type: This hyperparameter specifies the type of boosting used in the Catboost algorithm. In the table, the value 'Plain' indicates that the algorithm uses standard Catboosting.

- feature_border_type: This hyperparameter specifies the method used to split data along the feature borders. The value 'GreedyLogSum' indicates that the algorithm uses a greedy algorithm to find the optimal split point.

- bayesian_matrix_reg: This hyperparameter is a regularization term used to control the complexity of the model. In the table, the value of 0.1 indicates a moderate level of regularization.

- l2_leaf_reg: This hyperparameter is another regularization term that controls the L2 regularization applied to the weights of the decision trees.

- rsm: This hyperparameter is the "feature fraction" parameter, which controls the fraction of features that are randomly selected for each tree.

- model_size_reg: This hyperparameter is a regularization term that controls the size of the trees in the ensemble.

- posterior_sampling: This hyperparameter specifies whether or not to perform posterior sampling during training.

- border_count: This hyperparameter controls the number of splits to consider when finding the best split point along each feature.

- sparse_features_conflict_fraction: This hyperparameter is used in sparse data to determine the minimum overlap between categories before considering them distinct.

- leaf_estimation_backtracking: This hyperparameter specifies the type of backtracking algorithm used to optimize the leaf values during training.

- best_model_min_trees: This hyperparameter specifies the minimum number of trees in the ensemble to consider the model as "best".

- model_shrink_rate: This hyperparameter specifies the shrinkage rate for the ensemble.

- min_data_in_leaf: This hyperparameter specifies the minimum number of samples required to form a leaf node in the decision tree.

- loss_function: This hyperparameter specifies the loss function used during training. In the table, the value 'RMSE' suggests that the algorithm minimizes the root mean squared error.

- score_function: This hyperparameter specifies the score function used to evaluate the model during training.

- leaf_estimation_iterations: This hyperparameter specifies the maximum number of iterations for optimizing the leaf values in each tree.

- bootstrap_type: This hyperparameter specifies the type of bootstrap sampling used during training.

- learning_rate: This hyperparameter controls the step size taken during gradient descent updates to the model parameters.

In Figure 37, we compare the residuals and prediction for error graph of Catboost versus Catboost with *hyperopt* function. It can be observed that values are not improving as the Test R2 metric is reduced from 0.778 Catboost, to 0.632 if we apply *hyperopt* function. In the left plot, the training performance is good which is clearly visible from tightly packed residuals in the left plot. In the right plot, residuals are not tightly packed around zero. The testing performance is also better in the left plot as very few points are away from zero horizontal line than in the right plot.

Catboost

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	6.5538	83.7852	9.1534	0.7603	0.4030	0.4096
1	9.3913	171.4283	13.0931	0.6881	0.8586	1.1642
2	10.9114	250.8916	15.8396	0.5883	0.8546	2.0374
3	8.7544	154.0916	12.4134	0.5315	0.7260	0.5391
4	7.5330	90.8290	9.5304	0.7005	0.8917	1.5181
5	8.3647	116.5782	10.7971	0.7444	0.9220	2.1194
6	6.0740	84.9854	9.2188	0.7602	0.6275	0.9314
7	8.2784	127.9620	11.3120	0.7415	0.8221	1.2937
8	8.6285	132.1672	11.4964	0.7829	0.9820	0.6970
9	6.9633	89.0218	9.4351	0.7816	0.6580	0.6514
Mean	8.1453	130.1740	11.2289	0.7079	0.7745	1.1361
SD	1.3590	49.4731	2.0212	0.0805	0.1643	0.5749

Catboost after hyperopt

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	8.6387	127.3073	11.2831	0.7741	0.7246	1.2390
1	11.2509	200.7010	14.1669	0.5577	1.0115	1.8988
2	8.7180	131.0235	11.4465	0.7660	0.5808	0.9162
3	8.3733	131.9308	11.4861	0.6688	0.5658	0.7325
4	9.7173	164.8617	12.8398	0.4265	0.8392	1.3466
5	9.4494	152.3616	12.3435	0.6593	0.8829	1.2092
6	10.0365	193.7914	13.9209	0.6251	0.8987	2.0876
7	10.1556	175.6947	13.2550	0.6715	0.8554	1.2000
8	8.5323	112.6965	10.6159	0.7297	0.5738	0.8558
9	12.7494	316.3308	17.7857	0.5504	1.1203	1.4280
Mean	9.7622	170.6699	12.9143	0.6429	0.8053	1.2914
SD	1.3139	56.0014	1.9723	0.1023	0.1811	0.4105

Figure 37. Catboost vs Catboost after hyperopt function results

If we compare the residuals and prediction for error graph, Figure 38, we can also observe that values are not improving as the Test R2 metric is reduced from 0.778 Catboost, to 0.632 if we apply *hyperopt* function. In the left plot, the training performance is way good which is clearly visible from tightly packed residuals in the left plot. In the right plot, residuals are not tightly packed around zero. The testing performance is also better in the left plot as very few points are away from zero horizontal line than in the right plot. So, overall, the left plot looks superior than the right one which is also clear from the R2 itself.

It is not uncommon to see a decrease in performance (in terms of metric like R2) after applying hyperparameter tuning. Hyperparameter tuning has limitations to how much improvement can be achieved by tuning hyperparameters. In our case this happens for following reasons:

- Overfitting: When tuning hyperparameters, there is a risk of overfitting the model to the training data. This means that the model is optimized to perform well on the training data, but does not generalize well to new data. This can lead to a decrease in performance on the validation or test data.

- Randomness: The performance of machine learning models can be affected by random fluctuations in the data or in the modeling process. It is possible that a particular combination of hyperparameters that performed well on one dataset may not perform as well on another dataset due to these random fluctuations.

- Complexity: In some cases, the best hyperparameters may lead to a more complex model than the default hyperparameters. This can lead to a decrease in performance on the validation or test data.

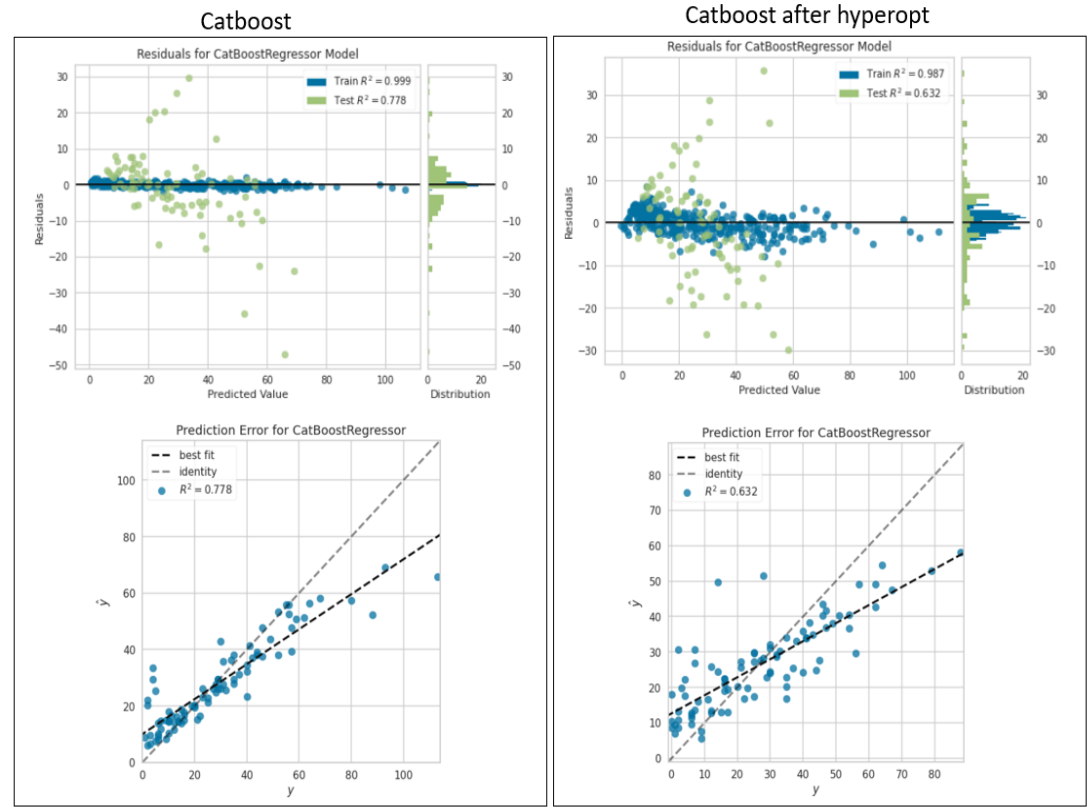

Figure 38. Catboost vs Catboost after hyperopt for residuals and prediction error

4.7. Evaluation of the model with fluid bed dryer data

In this section, Catboost model will be executed using the historical data obtained from the activity of the fluid bed dryer process in the production plant. The data set corresponds to 200 batches of medical products processed during one year and a half. The objective is to analyze the prediction of the model in terms of reduction of the preheating time of the machine and of the energy consumption savings of the fluid bed dryer for the preheating process. To this end, the evaluated parameters are: inlet air temperature difference, the preheating time prediction and energy consumption prediction. In this subsection, the Catboost model prediction regarding the duration and power savings of the preheating phase will be compared with the real results of duration and energy consumption.

4.7.1. Preheating temperature analysis

Figure 39 contains the analysis for the 200 batches from the historical information extracted from the fluid bed dryer using the Catboost model. The Y-axis indicates the difference in inlet air temperature and outlet, and the X-axis the time in minutes. The continuous red line is the average time used to preheat the fluid bed dryer before introducing the drug product for drying. The red dotted line is the Catboost prediction that indicates a potential saving of the preheating time. Approximately, on average, Catboost predicts that at 14.8 degrees, the air differences are stabilized. Therefore, from this instant on the machine is consuming unnecessarily energy since the fluid bed dryer is at the optimum point. As the temperature difference between the inlet and outlet air is very low, which means that the fluid bed dryer is warm enough, so it cannot absorb air and the air difference between inlet and outlet is stable.

.

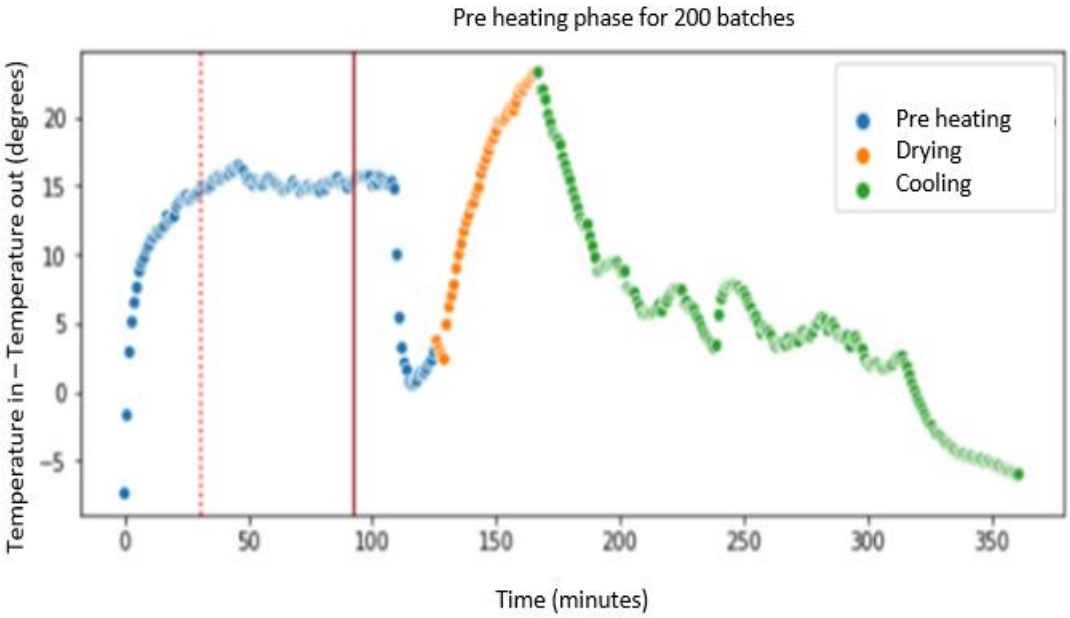

Figure 39. Average time of 200 test batches. Red continuous line is real average preheating and red dotted line is predicted value.

4.7.2. Time duration analysis

In Figure 40, it can be seen in blue color the real duration of the preheating process per month from the historical dataset. This duration is measured in minutes and represents the average of the time spent by the process for the whole month. This measure has been performed for the 200 batches evaluated during 18 months.

The results show that the preheating process duration varies from one month to another and fall in between 88.5 and 110.6 minutes, depending on when the optimal temperature difference in-out is reached. The average duration of the 200 batches during the 18 months is around 99.7 minutes. This key information will allow us to calculate the real consumption of the preheating process.

Figure 40 shows also the Catboost prediction duration of the preheating process. It can be observed how for the 200 batches, during 18 months evaluation, the predicted time is always lower than the real time. The reduction of the predicted time is significant, ranging this decrease from 34.7 minutes (39.2% time reduction) in the month of Dec'18 to 66.0 minutes (59.68.2% reduction) in the month of Oct'18.

The optimal time predicted by the algorithm corresponds on average per month between 42.5 and 59.5 minutes, with an average of 49.4 minutes The average predicted time reduction is 50.3 minutes. Therefore, duration of the process can be reduced on average by 50.45%.

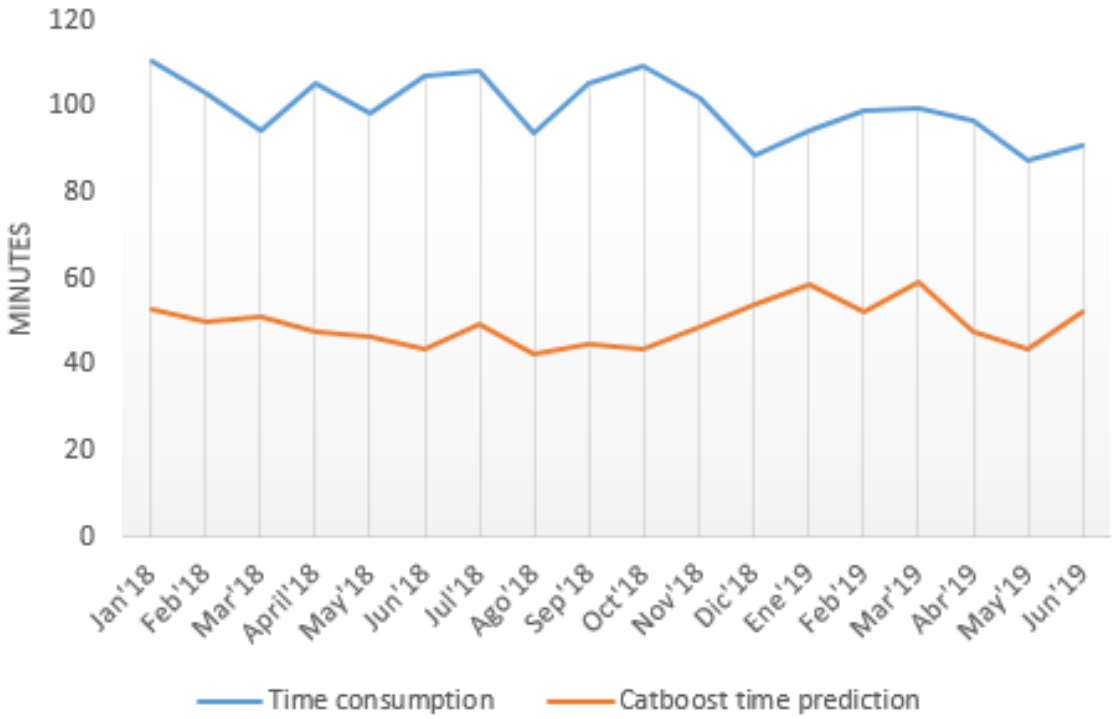

Figure 40. Time duration for preheating process comparing real duration with Catboost prediction

4.7.3. Energy saving analysis

In Figure 41, it can be seen in blue color the real energy consumption of the preheating process per month from the historical dataset. This energy consumption is measured in kWh and represents the average of the energy spent by the process for the whole month per batch. This measure has been performed for the 200 batches evaluated during 18 months.

The results show that the real preheating process energy consumption varies from one month to another and fall in between 27.1 kWh and 34.3 kWh per batch every month,

depending on when the optimal temperature difference in-out is reached. The average energy consumption of the 200 batches is 30.9 kWh per batch.

Figure 41 shows also the Catboost prediction energy consumption of the preheating process. It can be observed how for the 200 batches, during 18 months evaluation, the predicted energy consumption is always lower than the real energy consumption. The reduction of the predicted energy consumption is significant, ranging this decrease from 10.8 kWh (39.8% energy reduction) in the month of Dec'18 to 20.5 kWh (59.76% energy reduction) in the month of Oct'18.

The optimal energy consumption predicted by the algorithm corresponds on average per batch between 13.2 kWh and 18.4 kWh.The average predicted energy reduction is 15.6 kWh. Consequently, the reduction of energy consumption predicted by the algorithm, to complete the prehearing process, represents 50.48% less energy.

The total energy saving is calculated using the equation 4.

$$ES_t = Nbatches * ES_b \quad (4)$$

Being $Nbatches$ the number of batches and ES_b the energy saved per batch.

Based on Figure 40, there is a potential saving of 50.3 minutes per batch each time the fluid bed dryer is preheated. This means a saving of around 15.6 kWh per batch (50.3 minutes x 0.31 kWh). If the machine processes approximately 200 batches per year, based on the current estimation, then the annual potential energy savings could be approximately 3.120 kWh applying equation 4.

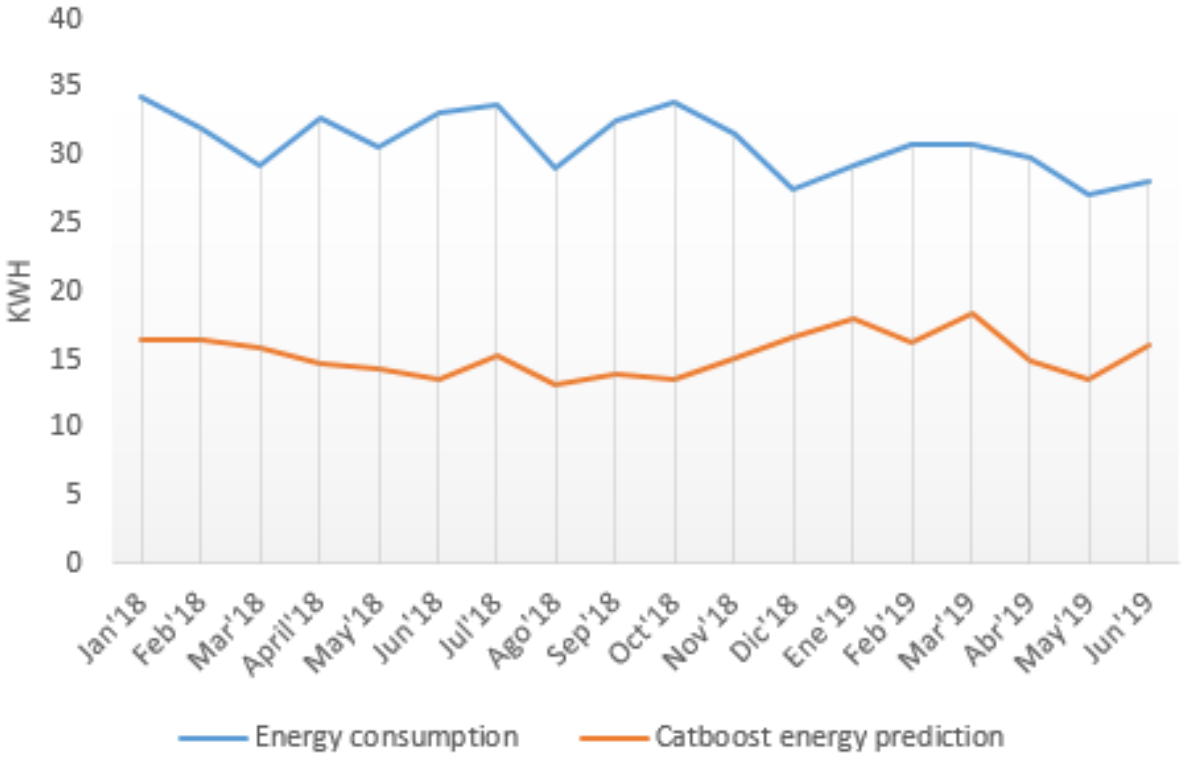

Figure 41. Energy consumption for preheating process comparing real energy with Catboost prediction

4.8. Conclusions

As summary, in this chapter we have selected the most suitable model for the fluid bed dryer prediction process based on the current data set obtained from the activity of the fluid bed dryer process in the production plant. First, several models, including Catboost, elastic net, random forest or linear regression, have been compared. We have selected Catboost because it provides the lowest error at the same time highest R2 as it has been described in previous sections.

Catboost with default hyperparameters have been evaluated in more detail as it was the best among all the algorithms, with highest R2 and lowest errors. Linear Regression model has been also evaluated but results It has been disregarded because of overfitting. Finally, we have evaluated Catboost with bayesian hyperparameter tuning but it did not produce better results.

Once the model has been selected, it has been adapted to the fluid bed dryer process. The analysis of the historical data set, 200 batches from 18 months of production has been performed. It has been shown that the model is able to predict on average a reduction of 50.45% of the preheating process duration and up to 59.68% in some cases. Likewise, the energy consumption of the fluid bed dryer for the preheating process could be reduced on average by 50.48% and up to 59.76%, what results on average in around 3.120 kWh energy consumption savings per year.

5 Fluid Bed Dryer Cloud-IIOT architecture

This chapter has been removed as it contains confidential information about the architecture and the infrastructure implemented.

6 Conclusions and Future Work

In this chapter, we summarize the main conclusions that have been achieved. Also, we include the publications and contributions derived from the research that has been carried out during this thesis. Finally, future lines of work that could be extended from the conclusions and contributions are described.

6.1. Conclusions

The motivation of this work was broad as we wanted to define an end-to-end IOT – Cloud computing definition and implementation architecture to reduce energy consumption for a fluid bed dryer machine located in a pharmaceutical manufacturing plant in Barcelona (Spain). This thesis adopts an innovative approach to improve the pharmaceutical drying process, identifying start and end of the preheating phase, by continuously monitoring critical parameters of the manufacturing equipment.

To fulfill these objectives, in chapter 2 it was presented a review of the most relevant literature of the different topics addressed in this thesis, including an introduction to the pharmaceutical manufacturing process, a brief description of fluid bed dryer operations, the current paradigm Industry 4.0 and Pharma 4.0., digital twin technology and an introduction to machine learning, including some state of art related work in the field of applying machine learning to reduce the energy consumption in the manufacturing industry

In chapter 3, fluid bed dryer historical data has been analyzed to identify critical variables and patterns using preprocessing advance analytics techniques. As conclusion, it can be stated that the preheating phase lasts longer than necessary. Some batches need less than 50.1 minutes to complete the preheating process, however, there are batches that take up to 180.3 minutes. In terms of energy consumption, it means that for some batches the fluid bed dryer consumes 15.5 kWh, and for others is 55.8 kWh, which could represent savings, in some cases, of 72.2% of energy.

In chapter 4, it was selected the most suitable model for the fluid bed dryer prediction process based on the current data set obtained from the activity of the fluid bed dryer process in the production plant. It was selected Catboost model because it provides the lowest error at the same time highest R2. Once the model was selected, it was adapted to the fluid bed dryer process. The analysis of the historical data set, 200 batches from 18 months of production shown that the model is able to predict on average a reduction of 50.45% of the preheating process duration and up to 59.68% in some cases. Likewise, the energy consumption of the fluid bed dryer for the preheating process could be reduced on average by 50.48% and up to 59.76%, what results on average in around 3.120 kWh energy consumption savings per year.

Finally, in chapter 5, an IIOT-Cloud computing architecture has been implemented, presenting the results in terms of energy savings from analyzing fluid bed dryer data in real time. According to the evaluated results, after three months of analyzing the fluid bed dryer data with our IIOT – Cloud computing architecture, the proposed machine learning Catboost model exhibits good performance and is capable of reducing 45 minutes of drying time per batch, which implies an energy saving for each batch of 13.95 kWh, corresponding to an estimated annual energy saving of approximately 2.8 MWh.

As it will be explained in future work, this architecture can be used to reduce energy consumption for other fluid bed processes such as drying, and in the future by other type of machinery, such us mixers, compactors, or coaters, by defining new machine learning models adapted to the new processes.

6.2. Publications and contributions.

During the development of this thesis, some papers were published in scientist journals. Besides we presented our work in different conferences, with the aim to share our research and obtain valuable feedback from other colleagues. In addition, we collaborated indifferent research projects. Finally, our work was awarded with the annual prize from Actualidad Economica magazine, El Mundo newspaper, as one of the best ideas of 2021 in the category of Industry 4.0.

6.2.1. Research projects

The present thesis has been developed in the framework of 3 research projects. These projects are directly related to energy savings in fluid bed dryers and the use of advanced analytics or machine learning for the optimization of production processes, which are main topics of the thesis. The projects are briefly described below, as well as their duration and the collaborating entities.

- “INCOGNITO: Towards the smart, green and self-organized Cognitive Plant of the future for the European process industries”. Objective: Definition of the use case 'Optimization of manufacturing processes through Data Analytics' in the proposal for the R+D+i for the DT-SPIRE-06-2019 contest within the innovation framework of the European Commission H2020. Duration 2 years between 2018 and 2019. Sponsored by IRIS Technology and Almirall SA.

- “IOT Analytics chemical plant”. Objective: To implement a proof of concept using IOT and machine learning technologies, to improve cleaning process from the chemical reactors of a manufacturing chemical plant, by reducing the consumption of reagents and energy. Duration 1 year during 2019. Sponsored by Almirall SA.

- “Dryer IOT Analytics”: Objective: To connect the IOT sensors from Fielder Aeromatic dryer to a Machine Learning algorithm platform (Azure) in real time either a model deployed in Azure Container Instances or an edge device (OPC Server). Duration 6 months between 2020 and 2021. Sponsored by Almirall SA.

6.2.2. Publications in scientific journals

- Barriga, R, Hassan, H, Romero, M, Nettleton, D. “Advanced data modeling for industrial drying machine energy optimization”, The Journal of Supercomputing, volume 78, pages 16820–16840, 2022.

- Barriga, R. Zahn, M. Blumenthal, R. Zamora, D and Romero, M. “Artificial Intelligence Used to Optimize Fluid Bed Drying”, International Society for Pharmaceutical Engineering, ISSN 0273-8139, Spain, 2022

- Barriga, R. Hassan, H. Romero, M and Nettleton, D. “Energy consumption optimization of drying machines in pharmaceutical process control”, Sensors Journal, 23 (8), 3994, 2023

- Barriga, R. Hassan, H. Romero, M and Nettleton, D. “Cloud computing-IIOT architecture for reducing fluid bed drier energy consumption”, Cluster Computing, 2023 (Under Review).

6.2.3. Conferences

- Barriga, R. “Big data in the pharmaceutical Industry 4.0”, Conference of Industry of things World, Berlin, Germany, September 2019

- Barriga, R. “Industrial big data o analitica avanzada de datos, claves para la toma de decisiones y mejorar el proceso productivo”, Conferencia de Fabrica Inteligente, IKN, Madrid, Spain, June 2019

- Barriga, R. “Industria 4.0”, Conferencia de Convergencia IT/OT Industria 4.0, Altran, Barcelona, Spain, June 2019

- Barriga, R and Romero, M. “Digitalization & Advance Analytics in Pharmaceutical Operations”, Conference of Artificial intelligence applied to pharmaceutical, International Society for Pharmaceutical Engineering, Madrid, Spain, November 2020

- Barriga, R. “Machine Learning and Artificial Intelligence for production optimization”, Conference of Digital Transformation Masterclass, Zigurat, Barcelona, Spain, April, 2021

6.2.4. Awards

Actualidad Economica magazine of El Mundo newspaper selected, as one of the ‘100 best Ideas of the year) in 2021, within the Industry 4.0 category, our “IOT Dryer Analytics” project, which has been the frameworkof the present thesis. The award ceremony has

been organized in Madrid by El Mundo newspaper. Miquel Romero (center of the picture), co-tutor of this thesis and me (right side of the picture) attended the ceremony.

6.3. Future work

The main areas of future research and developments that has been noticed during the development of this thesis are described in the following subsections.

6.3.1. Drying process implementation

One area with potential benefits, consists of using the current fluid bed dryer cloud computing infrastructure, for analyzing other processes beyond the current preheating process analyzed in this thesis, as for example, the optimization of the fluid bed dryer drying process. In chapter 3 we explain that the fluid bed dryer for drying a drug product includes 3 phases, the first phase is the preheating, which is the central objective of this thesis, the second phase with a shorter duration is drying, where the product to be dried is introduced inside the machine, and the third phase is the cooling.

Taking the one year and a half historical data from the fluid bed dryer hosted in our Azure architecture for the three processes (preheating, drying and cooling), we did some preliminary analysis of the drying process, using our Catboost model explained in chapter 4, and we discovered some interesting insights of reducing also the drying

process. This potential saving would be applied to the processing time and the energy consumed, therefore, in addition to reducing the 350 minutes that the entire process lasts (from the time we start the preheating of the machine, until the end of the cooling) approximately between 20-25 minutes per batch, the saving of the energy could be achieved.

In order to use any software that implies a change in the current drying process, like for example using a machine learning algorithm to predict when the drying process is finished, the system must be validated, as it has an impact on the activities of production of medicines or medical devices and, therefore, that may affect the quality of the final product, the safety of the patient and the integrity of the data generated. The validation of computer systems is a requirement that companies in the pharmaceutical sector must complete in order to comply with applicable regulations and obtain the necessary authorizations and certifications. It involves conducting a review process to validate that the computer system adheres to its specifications and possesses the capability to perform its designated task in accordance with relevant regulations and the intended use by the end user who is subject to regulation. In order to validate the infrastructure implemented in the development of this thesis, it could be necessary between 6 months and 1 year of work with the quality department.

6.3.2. Drug product end-to-end architecture

Another area for future work, it would be the use of the architecture developed in this thesis for the optimization of another type of process besides drug drying. As explained in previous chapters of this thesis, the production process of a drug product in the pharmaceutical industry, consists of different phases. The API (Active Product Ingredient), is the component of a drug or that has biological activity, and as first step it has to be mixed with other types of excipients or components in a mixer, then it goes through the drying process in a dryer, a fluid bed dryer in our case. Once the drug product is dried, then it is compacted in a compactor to give it the shape of a tablet, and finally the coating process is carried out to enhance the surface properties for corrosion and wear protection in a coater machine, and finally the drug product is moved to a packaging line to be included in a blister and a final consumer packaging box. As future work, it could be possible to connect all this type of machinery that are part of the drug manufacturing process (mixer, dryers, compacters, coaters, and packaging lines), directly to our IOT – Cloud computing solution as shown in Figure 42. We will have to

connect each PLC directly to our OPC Server, and then configure our MS Azure Edge Computing and Databricks modules, to be able to process the data coming from the sensors. We will have first to preprocess the data as explained in chapter 3, in order to identify the most relevant variable of the process, and to analyze each process to detect patterns and potential areas of optimization. After this, we will have to develop a new machine learning algorithm as shown in chapter 4, that is able to improve the performance of the process. In summary, most of the defined architecture in this thesis could be used not only to improve the drying process, but also for the complete end-to-end improvement of the drug manufacturing process.

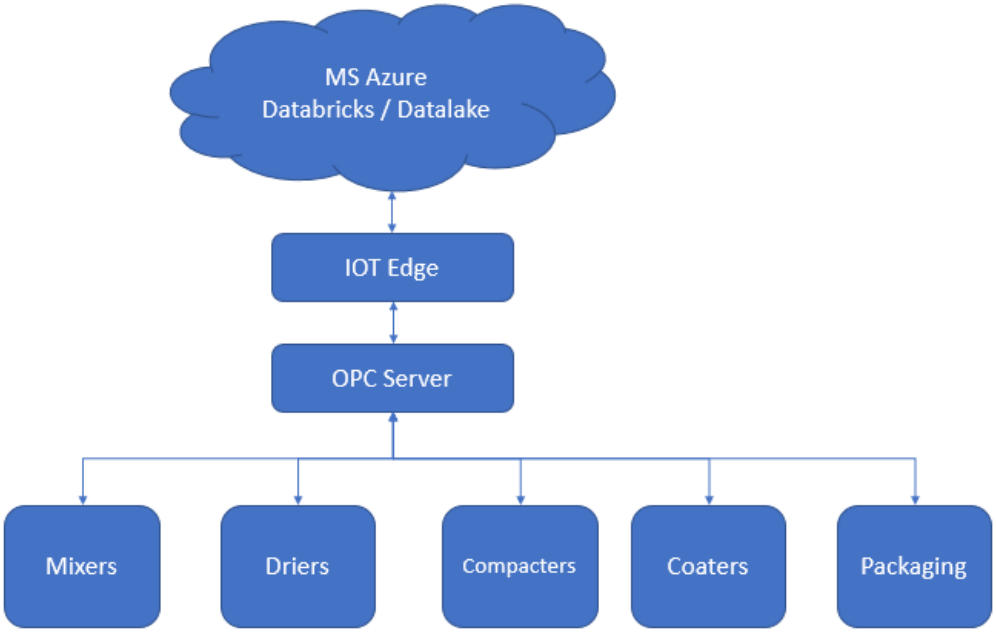

Figure 42. Architecture proposal for machine learning end to end drug manufacturing process.

6.3.3. Signal alert device

Finally, another potential benefit would be the implementation of a signal alert device in the fluid bed dryer. Once the Catboost model was connected in real time to the fluid bed dryer in order to save energy consumption, it is detected that in order to make easier for the operators to stop the preheating process, once our algorithm informs them in the SCADA screen explained in chapter 5, about the pending time to finish the preheating process, it would be interesting to provide the SCADA with a signal device. Signaling

devices, including tower lights, beacons and audible alarms, provide operators with alerts, allowing them to react to problems or manual actions more quickly.

List of Figures

List of Tables

Bibliography

[Aghbashlo et al., 2012] Aghbashlo, M.; Mobli, H.; Rafiee, S.; Madadlou, A. The use of artificial neural network to predict exergetic performance of spray drying process: A preliminary study. Drying Technology, 88, 32–43, 2012.

[Aghbashlo et al., 2014] M.; Sotudeh-Gharebagh, R.; Zarghami, R.; Mujumdar, A.S.; Mostoufi, N. Measurement techniques to monitor and control fluidization quality in fluidized bed dryers: A review. Drying Technology 32(9), 1005–1051, 2014.

[Aksu et al., 2012] Aksu B, Matas MD, Cevher E, Özsoy Y, Güneri T, York P. Quality by design approach for tablet formulations containing spray coated ramipril by using artificial intelligence techniques. International Journal of Drug Delivery. Jan 1;4(1):59, 2012

[Ali et al., 2021] Ali, Z.H., Ali, H.A. Towards sustainable smart IOT applications architectural elements and design: opportunities, challenges, and open directions. J Supercomput 77, 5668–5725, 2021.

[Allison et al., 2015] Allison G, Cain YT, Cooney C, Garcia T, Bizjak TG, Holte O. Regulatory and quality considerations for continuous manufacturing. May 20–21, 2014 continuous manufacturing symposium. J Pharm Sci.

[ALM] www.almirall.com, last accessed 2023.

[Anderson, 2005]. Anderson, S. Making Medicines: A Brief History of Pharmacy andPharmaceuticals. Pharmaceutical Press, 2005

[Arden et al., 2021] Arden NS, Fisher AC, Tyner K, Lawrence XY, Lee SL, Kopcha M. Industry 4.0 for pharmaceutical manufacturing: preparing for the smart factories of the future. Int J Pharm 602:120554, 2021

[Arun, 2015] Arun S. Handbook of Industrial Drying. CRC Press, Taylor and Francis Group, New York, USA, 2015

[AZURE] https://azure.microsoft.com/, last accessed 2023.

[Barricelli et al., 2019] Barricelli BR, Casiraghi E, Fogli D. A survey on digital twin: definitions, characteristics, applications, and design implications. IEEE Access 7:167653–167671, 2019.

[Barriga and Hassan, 2019] Barriga, R and Hassan, H. “IoT and Industry 4.0 – Manufacturing of the Future”, Zigurat, Innovation & Technology, The digital transformation blog, Spain, August 2019

[Barriga and Hassan, 2021] Barriga, R and Hassan, H. “Advanced Analytics to Optimize Manufacturing Operations”, Zigurat, Innovation & Technology, The digital transformation blog, Spain, March 2021

[Barriga BCN, 2019] Barriga, R. “Industria 4.0”, Conferencia de Convergencia IT/OT Industria 4.0, Altran, Barcelona, Spain, June 2019

[Barriga BER, 2019] Barriga, R. “Big data in the pharmaceutical Industry 4.0”, Conference of Industry of things World, Berlin, Germany, September 2019

[Barriga MAD, 2019] Barriga, R. “Industrial big data o analitica avanzada de datos, claves para la toma de decisiones y mejorar el proceso productivo”, Conferencia de Fabrica Inteligente, IKN, Madrid, Spain, June 2019

[Barriga and Romero, 2020] Barriga, R and Romero, M. “Digitalization & Advance Analytics in Pharmaceutical Operations”, Conference of Artificial intelligence applied to pharmaceutical, International Society for Pharmaceutical Engineering, Madrid, Spain, November 2020

[Barriga, 2021] Barriga, R. “Machine Learning and Artificial Intelligence for production optimization”, Conference of Digital Transformation Masterclass, Zigurat, Barcelona, Spain, April, 2021

[Barriga ISPE, 2022] Barriga, R. Zahn, M. Blumenthal, R. Zamora, D and Romero, M. “Artificial Intelligence Used to Optimize Fluid Bed Drying”, International Society for Pharmaceutical Engineering, ISSN 0273-8139, Spain, 2022

[Barriga et al., 2022] Barriga, R, Hassan, H, Romero, M, Nettleton, D. “Advanced data modeling for industrial drying machine energy optimization”, The Journal of Supercomputing, volume 78, pages 16820–16840, 2022.

[Barriga et. al., 2023] Barriga, R. Hassan, H. Romero, M and Nettleton, D. “Energy consumption optimization of drying machines in pharmaceutical process control”, Sensors Journal, 23 (8), 3994, 2023

[Barriga CLR, 2023] Barriga, R. Hassan, H. Romero, M and Nettleton, D. “Cloud computing-IIOT architecture for reducing fluid bed drier energy consumption”, Cluster Computing, 2023 (Under Review).

[Boschert et al., 2016] Boschert S, Rosen R. Digital twin—the simulation aspect. Mechatronic futures. Springer, Cham, pp 59–74, 2016

[Boyd, 2013] Boyd, GA. Development of a Performance-based Industrial Energy Efficiency Indicator for Pharmaceutical Manufacturing Plants, Duke University, USA, 2013

[Breiman, 2001] Breiman, L. Random forests. Machine learning, 45(1), 5-32, 2001.

[Burggraeve et al., 2013] Burggraeve, A., Monteyne, T., Vervaet, C., Remon, J. P., De Beer, T. Process analytical tools for monitoring, understanding, and control of pharmaceutical fluidized bed granulation: A review, European Journal of Pharmaceutics and Biopharmaceutics, Volume 83, Issue 1, pp. 2-15, 2013.

[Burggraeve et al., 2013] Burggraeve, A.; Monteyne, T.; Vervaet, C.; Remon, J.P.; De Beer, T. Process analytical tools for monitoring, understanding, and control of pharmaceutical fluidized bed granulation: A review. European Journal of Pharmaceutics and Biopharmaceutics, 83(1), 2–15, 2013

[Buvailo, 2018] Buvailo, A. The Why, How and When of AI in the Pharmaceutical Industry. Forbes, 2018

[Byrn et al., 2014] Byrn S, Futran M, Thomas H, Jayjock E, Maron N, Meyer RF. Achieving continuous manufacturing for final dosage formation: challenges and how to

meet them. Continuous Manufacturing Sympos. J Pharm Sci. Epub 2014. Mar;104(3):792-802. May, 2014

[Chalortham et al., 2008] Chalortham N, Pesawat P, Buranarach M, Sunidhi T. Ontology development forpharmaceutical tablet production expert system. In2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. May 14 (Vol. 1, pp. 205-208). IEEE, 2008

[Chaudhary, 2019] Chaudhary, D., and Vasuja, E. R. A Review on Various Algorithms used inMachine Learning. International Journal of Scientific Research in ComputerScience, Engineering, and Information Technology, 5(2), pp. 915-920, 2019

[Chen et al., 2020] Chen Y, Yang O, Sampat C, Bhalode P, Ramachandran R, Ierapetritou M. Digital twins in pharmaceutical and biopharmaceutical manufacturing: a literature review. Processes 8(9):1088, 2020

[Cheng et al., 2020] Cheng, D., Zhang, J., Hu, Z., Xu, S., & Fang, X. A digital twin-driven approach for on-line controlling quality of marine diesel engine critical parts. International Journal of Precision Engineering and Manufacturing, 21(10), 2020

[Chi et al., 2009] Chi, H. M., Moskowitz, H., Ersoy, O. K., Altinkemer, K., Gavin, P. F., Huff, B. E., & Olsen, B. A. Machine learning and genetic algorithms in pharmaceutical development and manufacturing processes. Decision Support Systems, 48(1), 69–80, 2009. DOI: 10.1016/j.dss.2009.06.010

[Colombo et al., 2020] Colombo, E. F., Shougarian, N., Sinha, K., Cascini, G., & de Weck, O. L. Value analysis for customizable modular product platforms: Theory and case study. Research in Engineering Design, 31(1), 2020

[Cox, 2017] Cox, V. Exploratory data analysis. In Translating Statistics to Make Decisions (pp. 47-74). Apress, Berkeley, CA, USA, 2017

[Daemmrich et al., 2005] Daemmrich, A., Bowden, M. Emergence of pharmaceutical science and industry: 1870–1930. Chem. Eng. News. 83, 2005

[Dey, 2016] Dey, A. Machine Learning Algorithms: A Review. International Journal of Computer Science and Information Technologies, Vol. 7, No. 3, pp. 1174-1179, 2016

[DNS] https://www.dnsstuff.com/, last accessed 2023.

[EGINN] https://www.eginnovations.com/, last accessed 2023.

[Ezell, 2016] Ezell, S.A. Policymaker's Guide to Smart Manufacturing. Information Technology & Innovation Foundation, Corpus ID: 190504769, 2016

[FDA, 2019] FDA. CDER Conversation: Assuring Drug Quality Around the Globe. 2019. https://www.fda.gov/drugs/news-events-human-drugs/cder-conversation-assuring-drug-uality-around-globe.

[Fix et al., 1989] Fix, E., & Hodges, J. L. Discriminatory analysis. Nonparametric discrimination : Consistency properties. International Statistical Review/Revue Internationale de Statistique, 57(3), 238-247, 1989.

[Fornasiero et al., 2021] Fornasiero R, Nettleton D, Kiebler L, Martinez de Yuso A, De Marco CE. AI and BD in process industry: a literature review with an operational perspective. Advances in production management systems, 5–9 Sept 2021, Nantes, France, 2021

[Fuhr et al., 2014] Fuhr, T., Gonce, A., Positano, L., Rutten, P., Tepis, V. From Measuring Failure to Building Quality Robustness in Pharma. In: McKinsey & Company, Chicago, IL, USA, 2014

[Gatley, 2004] Gatley, D. P. Psychrometric chart celebrates 100th anniversary. Ashrae Journal, 46(11),16, 2004

[Ghasemi-Varnamkhasti et al., 2014] Ghasemi-Varnamkhasti, M.; Aghbashlo, M. Electronic nose and electronic mucosa as innovative instruments for real-time monitoring of food dryers. Trends in Food Science & Technology, 2014

[Ghori et al., 2020] Ghori KM, Abbasi RA, Awais M, Imran M, Ullah A, Szathmary L. Performance analysis of different types of machine learning classifiers for non-technical loss detection. IEEE Access, vol. 8, pp. 16033-16048, 2020

[Guilfoyle, 2018] Guilfoyle, P. Pharma 4.0: Industry 4.0 Applied to Pharmaceutical Manufacturing.Pharmaceutical Process World, Northwest Analytics, 2018

[Gundu et al., 2020] Gundu, S. R., Panem, C. A., & Thimmapuram, A. The Dynamic Computational Model and the New Era of Cloud Computation Using Microsoft Azure. *SN Computer Science*, *1*(5), 1-7, 2020

[Haron et al., 2017] Haron, N & Zakaria, P. Recent advances in fluidized bed drying. IOP Conference Series: Materials Science and Engineering, 2017

[Herwig et al., 2017] Herwig, C., Wolbeling, C., and Zimmer, T.A holistic approach to production control from industry 4.0 to pharma 4.0. Pharmaceutical Engineering. 37. 44-49, 2017

[INFLUX] https://www.influxdata.com/, last accessed 2023.

[Kayihan, 1985] Kayihan, F. Stochastic modeling of lumber drying in the batch kilns. In Drying' 85 (pp.368-375). Springer, Berlin, Germany, 1985

[Keskes et al., 2020] Keskes, Sonia & Hanini, Salah & Hentabli, Mohamed & Laidi, Maamar. Artificial Intelligence and Mathematical Modelling of the Drying Kinetics of Pharmaceutical Powders, 2020

[Kotsiantis et al., 2007] Kotsiantis, S. B., Zaharakis, I., and Pintelas, P. Supervised machine learning: Areview of classification techniques. Emerging artificial intelligence applications in computer engineering, 160(1), pp. 3-24, 2007

[Lee et al., 2015] Lee, S.L., O'Connor, T.F., Yang, X. Modernizing pharmaceutical manufacturing: from batch to continuous production. J. Pharmaceut. Innov. 10 (3), 191–199, 2015

[Leurent et al., 2018] Leurent, H., deBoer, E. The Next Economic Growth Engine Scaling FourthIndustrial Revolution Technologies in Production http://www3.weforum.org/docs/WEF_Technology_and_Innovation_The_Next_Economic_Growth_Engine.pdf, 2018.

[Lifset, 2014] Lifset, R. D.A new understanding of the American energy crisis of the 1970s. Historical Social Research/Historische Sozialforschung, 22-42, 2014.

[Liu, 2022] Liu, Z. Using neural network to establish manufacture production performance forecasting in IOT environment. J Supercomputing, 2022,

[Lorenz et al., 2018] Lorenz Binggeli, H.H., Woelbeling, Christian, Zimmer, Thomas. Pharma 4.0 Hype or Reality? Pharmaceutical Engineering, 2018.

[Lourenço et al., 2012] Lourenço V, Lochmann D, Reich G, Menezes J, Herdling T, Schewitz J. A quality by design study applied to an industrial pharmaceutical fluid bed granulation. Eur J Pharm Biopharm 81(2):438–447, 2012

[Lowd et al., 2005] Lowd, D., and Domingos, P. Naive Bayes models for probability estimation. In Proceedings of the 22nd international conference on Machine learning (pp. 529-536), 2005

[Markarian, 2016] Markarian, J. The Internet of Things for Pharmaceutical Manufacturing. PharmTech, 2016

[Markarian, 2018] Markarian, J. Modernizing pharma manufacturing. Pharmaceutical Technology, 42(4): 20-25, 2018

[Meyer, 2004] Meyer, D. Support vector machines: The interface to libsvm in package e1071, 2004.

[Mittal et al., 2003] Mittal, G. S., & Zhang, J. Artificial neural network-based psychrometric predictor. Biosystems Engineering, 85(3), 283-289, 2003

[ML] https://machinelearningmastery.com/, last accessed 2023.

[Moore, 2018] Moore, M. What is Industry 4.0? Everything you need to know. TechRadarpro, 2018

[MSC] https://www.microsoft.com/, last accessed 2023.

[Muhammed et al., 2020] Muhammed, Sand Ucu. "Comparison of the IOT Platform Vendors, Microsoft Azure, Amazon Web Services, and Google Cloud, from Users' Perspectives," *2020 8th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 1-4, 2020

[Mujumda, 2012] Mujumdar, A.S. Editorial: The role of drying technology in sustainable R&D and innovation. Drying Technology 30(13), 1387–1387, 2012

[Mujumdar, 2014] Mujumdar, A.S. Research and development in drying: Recent trends and future prospects. Drying Technology 22(1–2), 1–26, 2014

[Nash et al., 2003] Nash, Robert A., Wachter, Alfred H. Pharmaceutical process validation. pages 93-121. 2003

[Nazghelichi et al., 2011] Nazghelichi, T.; Aghbashlo, M.; Kianmehr, M.H.; Omid., M. Prediction of energy and exergy of carrot cubes in a fluidized bed dryer by artificial neural networks. Drying Technology, 29(3), 295–307, 2011

[Nettleton et al., 2016] Nettleton, D. F., Bugnicourt, E., Wasiak, C., & Rosales, A. (2016). Towards automatic calibration of in-line machine processes. In Proc. 18th International Conference on Industrial Engineering and Manufacturing (ICIEMPM), London, U.K., 2016

[Nettleton et al., 2018] Nettleton, D. F., Wasiak, C., Dorissen, J., Gillen, D., Tretyak, A., Bugnicourt, E., Rosales, A. Data Modeling and Calibration of In-Line Pultrusion and Laser Ablation Machine Processes, International Conference on Advanced Data Mining and Applications (ICADMA), Barcelona, Spain, 2018

[Nettleton et al., 2018] Nettleton, D. F., Wasiak, C., Dorissen, J., Gillen, D., Tretyak, A., Bugnicourt, E., Rosales, A. Data Modeling and Calibration of In-Line Pultrusion and Laser Ablation Machine Processes, International Conference on Advanced Data Mining and Applications (ICADMA), Barcelona, Spain, 2018

[Opitz, 1999] Opitz, D., and Maclin, R. Popular ensemble methods: An empirical study. Journalof artificial intelligence research, 11, pp. 169-198, 1999

[Park et al., 2018] Park, J., Ferguson, M., & Law, K. H. Data driven analytics (machine learning) for system characterization, diagnostics and control optimization, 2018, doi:10.1007/978-3-319-91635-4_2

[Peterson et al., 2019] Peterson, J. J., Snee, R. D., McAllister, P. R., Schoeld, T. L., and Carella, A. J. Statistics in pharmaceutical development and manufacturing. Journal of Quality Technology, 41(2):111-134, 2019

[Petrović et al., 2011] Petrović, Jelena & Chansanroj, Krisanin & Meier, Brigitte & Ibrić, Svetlana & Betz, Gabriele. Analysis of fluidized bed granulation process using conventional and novel modeling techniques. European journal of pharmaceutical sciences: official journal of the European Federation for Pharmaceutical Sciences. 44. 227-34, 2011

[Prokhorenkova et al., 2017] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A. CatBoost: unbiased boosting with categorical features, 2017

[PHARMA] https://pharmaceuticalupdates.com, last accessed 2023.

[Qiu et al., 2020] Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., & Wu, D. O. Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys & Tutorials*, *22*(4), 2462-2488, 2020

[REDHAT] https://www.redhat.com/, last accessed 2023.

[Schoen, 2005] Schoen, C. A new empirical model of the temperature humidity index. Journal of applied meteorology, 44(9), 1413-1420, 2005

[Shafqat et al., 2020] Shafqat, S., Kishwer, S., Rasool, R.U. et al. Big data analytics enhanced healthcare systems: a review. J Supercomputing 76, 1754–1799, 2020.

[Shcherbakov et al., 2020] Shcherbakov, Maxim V., Artem V. Glotov, and Sergey V. Cheremisinov. "Proactive and predictive maintenance of cyber-physical systems." Cyber-Physical Systems: Advances in Design & Modelling. Springer, Cham, 2020. 263-278, 2020

[Simões, 2019] Simões Aparício, Maria Madalena.Modeling the variation of psychrometric properties of air in depth. Master of Science Thesis dissertation, Instituto Superior Técnico, Lisboa, Portugal, 2019

[Sonnedecker et al., 1976] Sonnedecker, G., Urdang, G. Kremers and Urdang's history of pharmacy. Lippincott, pages 221-240, 1976

[STRIIM] https://www.striim.com/, last accessed 2023.

[Su et al., 2015] Su Y, Zhang M, Mujumdar AS. Recent developments in smart drying technology. Drying Technology.Feb 17;33(3):260-76, 2015

[Sun et al.,2020] Sun, L., Jiang, X., Ren, H., & Guo, Y. Edge-cloud computing and artificial intelligence in internet of medical things: architecture, technology and application. *IEEE Access*, *8*, 101079-101092, 2020

[Sutton, 1992] Sutton, R. S. “Introduction: The Challenge of Reinforcement Learning”, MachineLearning, 8, Page 225-227, Kluwer Academic Publishers, Boston, USA, 1992

[Thomas, 2006] Thomas, P. “Will Pharma Wear the Energy Star,” Pharma Manufacturing, March 2006

[Tibshirani, 1996] Tibshirani, R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1),267-288, 1996.

[Tilley, 2017] Tilley, J. Automation, Robotics, and the Factory of the Future, McKinsey & Company, Operations, September 2017

[Ugur et al., 2008] Ugur & becerkli, Yasar & Turker, Mustafa. (2008). Dynamic Neural-Network-Based Model-Predictive Control of an Industrial Baker's Yeast Drying Process. Neural Networks, IEEE Transactions on. 19. 1231 – 1242, 2008

[Wasiak et al., 2017] Wasiak, C., Nettleton, D., Janssen, H., & Brecher, C. Quantification of Micro-Pullwinding Process as Basis of Data Mining Algorithms for Predictive Process Model. In 21st Int. Conf. on Composite Materials (ICCM), Xi’an, China, 2017

[Yu et al., 2017] Yu, L.X., Kopcha, M. The future of pharmaceutical quality and the path to getthere. Int. J. Pharm. 528 (1–2), 354–359, 2017

[Zhang et al., 2003] Zhang, D., and Nunamaker, J. F. Powering e-learning in the new millennium: anoverview of e-learning and enabling technology. Information systems frontiers, 5(2), pp. 207-218, 2003

[Zhu et al., 2009] Zhu, X., and Goldberg, A. B. Introduction to semi-supervised learning. Synthesislectures on artificial intelligence and machine learning, 3(1), pp. 1-130, 2009

[Zhu, 2005] Zhu, X. (2005). Semi-Supervised Learning Literature Survey, University of Wisconsin, WI, USA, 2005