Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Federica Ferrari

Data Analytics for Hydrogen Safety: Prediction of Liquid Hydrogen Release Characteristics

Master's thesis in Reliability, Availability, Maintainability, and Safety
Supervisor: Nicola Paltrinieri
Co-supervisor: Federico Ustolin
February 2022

NTNU
Norwegian University of
Science and Technology

Federica Ferrari

Data Analytics for Hydrogen Safety: Prediction of Liquid Hydrogen Release Characteristics

Master's thesis in Reliability, Availability, Maintainability, and Safety
Supervisor: Nicola Paltrinieri
Co-supervisor: Federico Ustolin
February 2022

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

Hydrogen can be adopted as a clean alternative to hydrocarbons fuels in the marine sector. Liquid hydrogen (LH_2) is an efficient solution to transport and store large amounts of hydrogen, thus it is suitable for the maritime field. Additional safety knowledge is required since this is a new application and emerging risk might arise. Recently, a series of LH_2 large-scale release tests was carried out in an outdoor facility as well as in a closed room to simulate spills during a bunkering operation and inside the ship's tank connection space, respectively. The extremely low boiling point of hydrogen (-253°C) can cause condensation or even solidification of air components, thus enrich with oxygen the flammable mixture. This can represent a safety concern in case of ignition of the flammable mixture of LH_2 and solid oxygen, since it was demonstrated that the resulting fire may transition to detonation. In this study, the abovementioned LH_2 release experiments were analysed by using an advanced machine learning approach. The aim of this study was to provide critical insights on the oxygen condensation and solidification during an LH_2 accidental spill and to evaluate whether the hydrogen concentration within the gas cloud formed due to the LH_2 evaporation would reach the lower flammability limit. In particular, a model was developed to predict the possibility and the location of the oxygen phase change and of the hydrogen concentration above the lower flammability limit depending on the operative conditions during the bunkering operation (e.g. LH_2 flow rate). The model demonstrated accurate and reliable predicting capabilities. The outcomes of the model can be exploited to select effective safety barriers and adopt the most appropriate safety measures in case of liquid hydrogen leakage.

Table of contents

List of figures

List of tables

Chapter 1
Introduction

1.1 Background

Nowadays hydrogen is gaining increasing importance, as it is considered a promising source of energy (Aaneby, Gjesdal and Voie, 2021). In fact, it represents one of the best alternatives to fossil fuels and guarantees a "clean, secure and affordable energy future" (Birol, 2019). Hydrogen is a light, non-toxic and abundant substance, characterised by a wide flammability rage (4-75 vol% in air) and an extremely low ignition energy (0.019 mJ), (Aaneby, Gjesdal and Voie, 2021). Its gravimetric energy density is considerable: for given mass, hydrogen contains about 2.5 times more energy than natural gas (Verfondern et al., 2021). However, hydrogen gas density at atmospheric conditions is very low (0.0883 kg/m^3 (NIST, 2022)), which means that in order to reduce its volume and increase its storage capacity it must be compressed or liquefied. Liquid hydrogen is indeed one of the most efficient solutions for storage and transportation of hydrogen, being more practical when it comes to bunkering and handling of the fuel (Aaneby, Gjesdal and Voie, 2021). The most significant drawback of liquid hydrogen is the high amount of energy required by the liquefaction process (Verfondern et al., 2021); since hydrogen gas is a supercritical fluid, it cannot be liquefied by compression at a normal ambient temperature, therefore the liquefaction is performed cryogenically, obtaining a cryogenic fluid stored virtually at atmospheric pressure and at a temperature of -253°C (NIST, 2022).

Many hazards connected to the storage and utilisation of liquid hydrogen can be identified, and they are analysed in safety and risk assessments. The main issue is associated with leakage, which leads to the formation of cryogenic pools on the ground, the propagation of flammable clouds in the environment as well as major damages on humans and infrastructures in case of ignition (Verfondern et al., 2021). Moreover, the low storage temperature of liquid hydrogen may cause condensation or solidification of air components, such as nitrogen and oxygen, which might clog ventilation masts or other pipes (Aaneby, Gjesdal and Voie, 2021). In addition, the liquefaction or solidification of oxygen is of major concern, since it could enrich flammable mixtures, having a higher boiling point than nitrogen, and therefore it would increase the likelihood of detonation in case of ignition (Hooker, Willoughby and Royle, 2011). Many

tests have been performed to investigate these hydrogen safety aspects, both on a laboratory scale and on a larger scale, such as those performed by the Norwegian Defence Research Establishment (FFI) in the UK. Those experiments have provided useful data for the elaboration and improvement of numerical models employed in risk analyses (Aaneby, Gjesdal and Voie, 2021).

1.2 Objectives

The aim of this thesis is to provide a data driven model which is able to support the engineer in the decision-making process in critical situations of safety concern.
The method described in this work is based on the analysis of liquid hydrogen leakage data to build a model capable of performing several tasks (e.g. prediction, classification, etc.). Such data consist in measurements returned by sensors in experimental studies carried out by the Norwegian Defence Research Establishment (FFI) concerning hydrogen release scenarios. The tests reproduce possible leakage events in bunkering operations.
First, a detailed analysis of the leakage scenarios has been performed in order to identify the main problems connected to the handling and storage of an ultra-low boiling point cryogenic fluid. Then, after accurate research to identify the most suitable methods, Machine Learning models have been developed and tested on their ability to predict the occurring of those issues, such as the formation of liquid or solid oxygen and hydrogen gas concentration. In other words, the main objectives of this thesis are:

- the analysis of experimental studies to identify different classes related to safety issues;
- the training of three Machine Learning models: Linear, Deep and Wide&Deep;
- the evaluation of the capability of those models to predict the different classes.

1.3 Approach

All the analyses described in the present work have been performed on databases built on data collected during the experimental studies conducted by the Norwegian Defence Research Establishment.
Firstly, the data collected during such experimental tests involving liquid hydrogen leakage have been studied and the main problems connected to a cryogenic liquid spill have been

evaluated. The databases have been built on the basis of the pointed-out problems: some mitigation measures in case of a release scenario have been analysed and a response time necessary to such measures to take action has been considered to develop the databases.
Finally, these datasets have been used to train and evaluate three different machine learning models: the Linear Model, the Deep Model and the Wide&Deep Model.
The analyses have been performed using python as a programming language. Both PyCharm 2021.2 and Google Collaboratory have been used.
The method presented in this thesis has been developed with the purpose of predicting the condensation or solidification of air oxygen and the formation of a flammable atmosphere: using this method to predict other metrics may not lead to the same results.

1.4 Outlines

This work includes six chapters and two appendices. Chapter 2 describes the theoretical background of the present work, and it is divided into four sections. In the first section, the experimental studies conducted by the FFI are described in detail. In the second section, the main hazards connected to liquid hydrogen accidental releases are analysed. In the third section, some mitigation measures to be activated in case of a cryogenic liquid spill scenario are presented. The fourth section introduces Machine Learning and dives deeper into the models utilised in this work. Chapter 3 focuses on the different databases built upon the experimental studies' data. Furthermore, the analyses performed during this thesis work are described in detail. Specifically, the final section of the chapter focuses on the Machine Learning simulations. In Chapter 4, the results obtained from the analyses described in Chapter 3 are presented. The results are then discussed and evaluated in depth in Chapter 5. Moreover, the limitations of the methods are here highlighted and some suggestions about possible improvements are provided. In the final chapter (i.e. Chapter 6), the findings are summarized and some recommendations for further works are presented. Finally, Appendix A includes the code used for the Machine Learning simulations while Appendix B shows some extracts of the analysed databases.

Chapter 2
Theoretical background

2.1 Experimental studies

The Norwegian Defence Research Establishment (FFI) has performed a series of experimental tests with the objective of understanding the liquid hydrogen behaviour to facilitate its introduction as fuel for ships. The release tests were carried out by varying the flowrate and duration to simulate realistic accidental spills for maritime applications (Aaneby, Gjesdal and Voie, 2021).

Two different kinds of tests have been performed:

- Outdoor leakage studies;
- Closed room and ventilation mast studies.

2.1.1 Outdoor leakage studies

The outdoor leakage tests consisted in the release of liquid hydrogen on the ground on a pad above which many sensors and thermocouples were placed. Ambient conditions were recorded in each test; the measurements included wind speed and direction, humidity and ambient pressure. The wind speed and direction were measured through sensors installed in a mast near the pad. In each test, pad temperature, field temperature and gas concentration, were recorded. Temperature measurements were conducted using thermocouples. The hydrogen concentration in air was measured by oxygen sensors whose measurements were translated to hydrogen concentrations based on oxygen depletion.

A total of 48 thermocouples were placed on or in the test pad for the outdoor leakage experiments, as shown in *Figure 1*, and more than 40 thermocouples were used for field temperature measurements (see *Figure 3*).

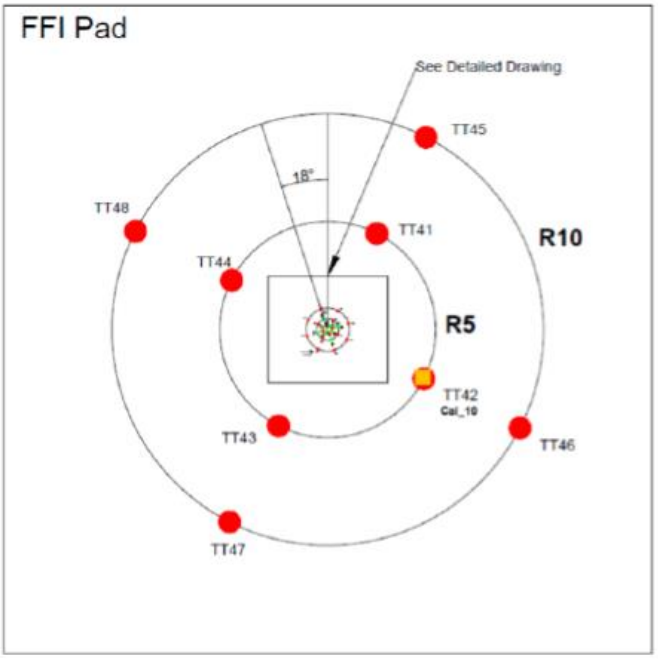


Figure 1. Outdoor leakage studies: placement of thermocouples to measure pad temperature and calorimeters to measure heat flux (ignited tests only) (Aaneby, Gjesdal and Voie, 2021).

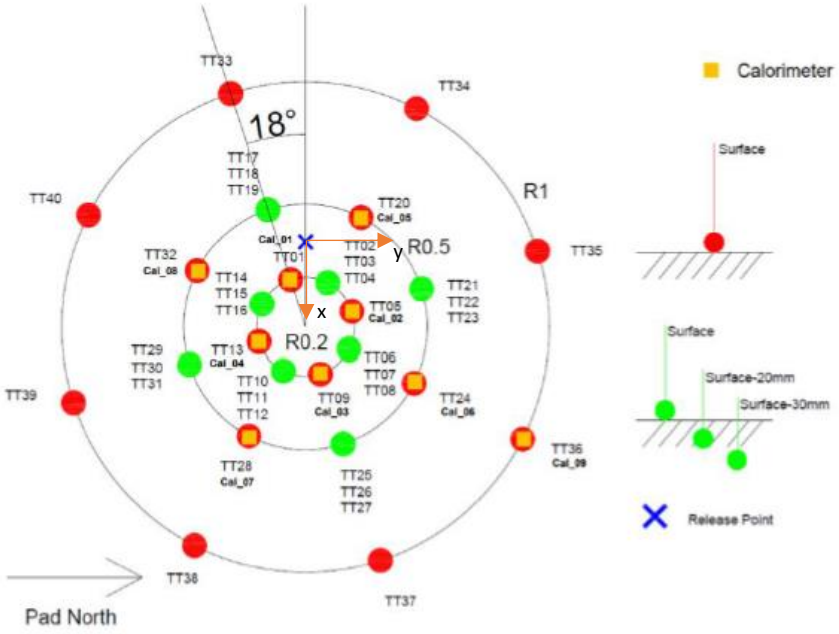


Figure 2. Details around the release point. The red dots indicate the locations of the surface measurements; the green dots indicate the locations of the measurements below the concrete surface. The blue cross indicates the release point of LH_2 (Aaneby, Gjesdal and Voie, 2021).

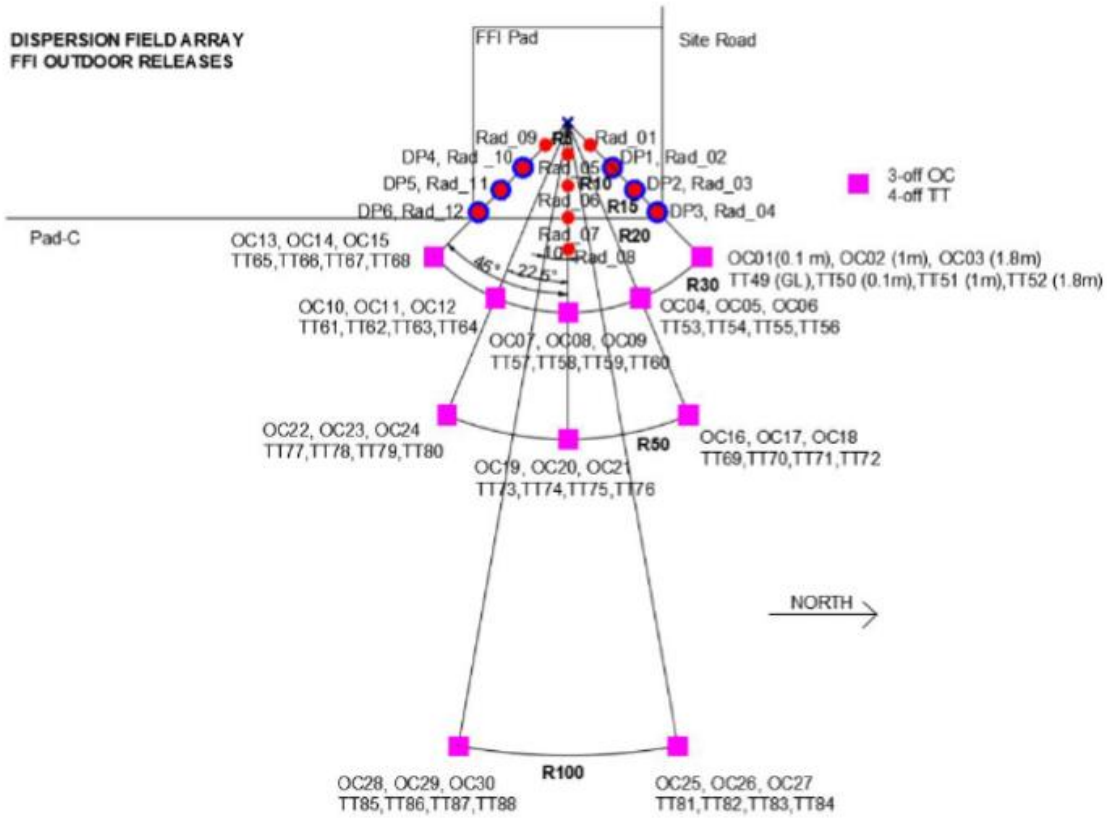


Figure 3. Outdoor leakage studies: *initial instruments' locations for measurements of field temperature, gas concentration, thermal radiation and field overpressure (pink squares: oxygen sensors and thermocouples for field temperature measurements; red dots: radiometers; blue circles: pressure sensors). The blue cross indicates the release point of LH_2 (Aaneby, Gjesdal and Voie, 2021).*

A total number of seven tests were performed. These tests aimed to simulate liquid hydrogen spills from bunkering operations, as depicted in *Figure 4*. The liquid hydrogen release flowrates were varied up to 50 kg/min – which reproduce real accidental release rates – and two intermodal containers were placed close to the release point to simulate obstacles.

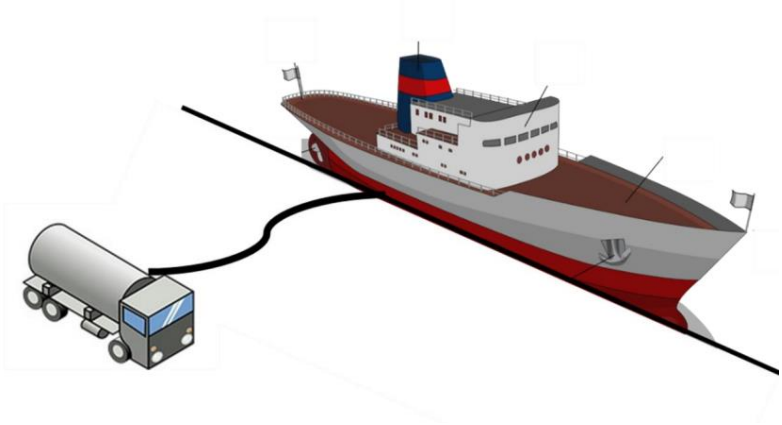


Figure 4. Illustration of a truck to ship bunkering (Aaneby, Gjesdal and Voie, 2021)

As stated by Aaneby, Gjesdal and Voie (2021), this study aimed to (i) provide information about the formation, propagation and duration of a cryogenic liquid pool, (ii) evaluate the gas cloud generated by such leakage and (iii) describe the cloud behaviour in case of ignition as a simple burning, a deflagration or a detonation event.

As emerged from the tests, the formation of the liquid pool on the ground depended on the orientation of the release hose (vertical downwards or horizontal) and it only extended up to 0.5 m from the release point. The release of a cryogenic fluid may lead to condensation and freezing of air components on the ground due to the ultra-low boiling point of hydrogen - 20 K (Perry and Green, 2008). These phenomena are particularly critical since they might enhance the risk of explosion of the flammable mixture in case of ignition. Moreover, the concentration of hydrogen within the gas cloud generated by the partial vaporization of the released liquid hydrogen exceeded the lower flammability limit (LFL) within 50 m from the release point. In none of the tests a spontaneous ignition was observed.

2.1.2 Closed room and ventilation mast studies

The closed room and ventilation mast studies consisted in liquid hydrogen release on the bottom of a shipping container to simulate the spill of the cryogenic fluid in the technical room (tank connection space, TCS) connected to the storage tank (see *Figure 6*).

Ambient conditions were recorded in each test; the measurements included wind speed and direction, humidity and ambient pressure. The wind speed and direction were measured through sensors installed in a mast near the pad. Temperature and gas concentration were measured in the TCS and in the field.

In the enclosed room studies setup, 15 thermocouples were placed on the TCS floor, 10 thermocouples were placed in the TCS and three more inside the ventilation mast, as shown in *Figure 5*.

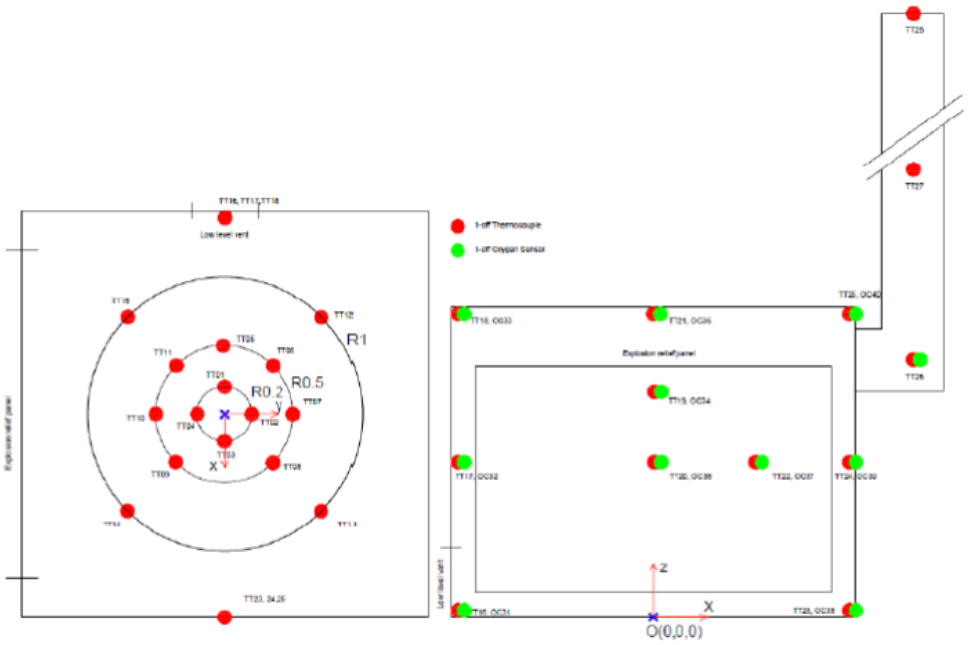


Figure 5. Ventilation mast studies: locations of thermocouples (red dots) and oxygen sensors (green dots). Floor of the TCS to the left (top view). TCS connected to ventilation mast to the right. Blue cross indicates the release point of LH_2 (Aaneby, Gjesdal and Voie, 2021).

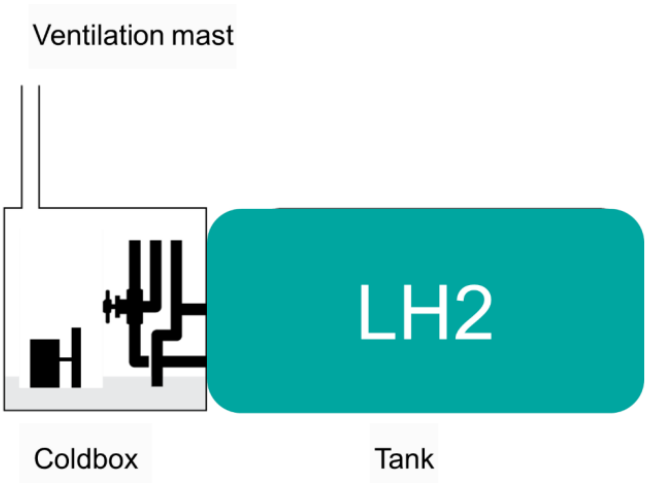


Figure 6. Illustration of a tank connection space (TCS) also called coldbox, connected to a ventilation mast (Aaneby, Gjesdal and Voie, 2021)

A total number of eight tests were performed. In one test the enclosed room was first purged with nitrogen in order to prevent the formation of flammable mixtures that can lead to explosions in case of ignition. The main objectives of these tests were the following:

- providing information about the hydrogen concentration inside the TCS;

- detecting eventual clogging of the ventilation mast;
- observing pressure build-up in the TCS due to the evaporation of liquid hydrogen.

As outlined in the report, the concentration of hydrogen in the room reached 100% within 30 s after the release had started. Moreover, since the room was not completely tight, no pressure build-up was observed due to liquid hydrogen vaporization.

Furthermore, the condensation or solidification of air components inside the ventilation mast did not result in its clogging. Finally, evidence of condensed or frozen air components on the floor's surface in the room remained for a long period, whereas the liquid pool on the ground rapidly evaporated.

2.2 Liquid hydrogen release hazards

A liquid hydrogen release scenario is analysed in this section.

The cryogenic liquid initially flashes to gas when released onto the ground due to the large temperature difference between the liquid and the ground surface. Then, the surface cools down enough to allow the formation of a pool of liquid hydrogen within few minutes. The cloud of gaseous hydrogen is visible due to the condensation of water vapour contained in air during the release. Moreover, its dispersion highly depends on the wind speed and direction.

Therefore, the pool formed on the ground is composed of liquid hydrogen and condensed air. It is also possible to observe the formation of a solid deposit which might be a mixture of solid oxygen and nitrogen (Royle and Willoughby, 2014). *Figure 7* shows how the growth of a solid deposit might be formed.

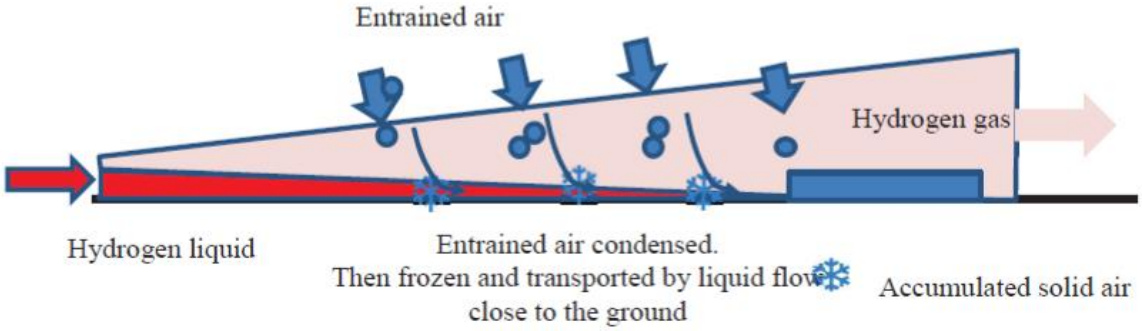


Figure 7. Schematic representation of solid air accumulation during a liquid hydrogen spill scenario (Atkinson, 2021).

Air condenses and freezes on the pool's surface and right in front of it, the solid deposit eventually impedes the liquid to flow further on the ground. Not all the droplets of condensed air fall on the liquid pool. Many of them fall outside the pool due to the restriction of the liquid extent caused by the solid deposit. In this way, condensed air – whose components have higher boiling points than the hydrogen one – accumulates and freezes on the previously deposited material forming a larger amount of solid deposit, at a temperature higher than the hydrogen boiling point. Only the quantity of solid material which is right in front of the liquid pool reaches temperatures close to -253°C, being in contact with LH_2. Therefore, only a proportion of the solid deposit forms a detonable mixture with liquid hydrogen (Atkinson, 2021).

The main problem connected to the condensation and freezing of air components on the ground is related to the behaviour of the flammable mixture in case of ignition: a condensed phase explosion might occur. Condensed phase explosions can have harmful consequences to both buildings and people (Davies, 1993), which manifest as:

- shock wave: the explosion event is accompanied by a blast wave that instantly increases the atmospheric air pressure to a peak value which can cause great damage to buildings and people. Then this overpressure phase is followed by a phase of negative pressure, as shown in *Figure 8*, which can still cause moderate damage.

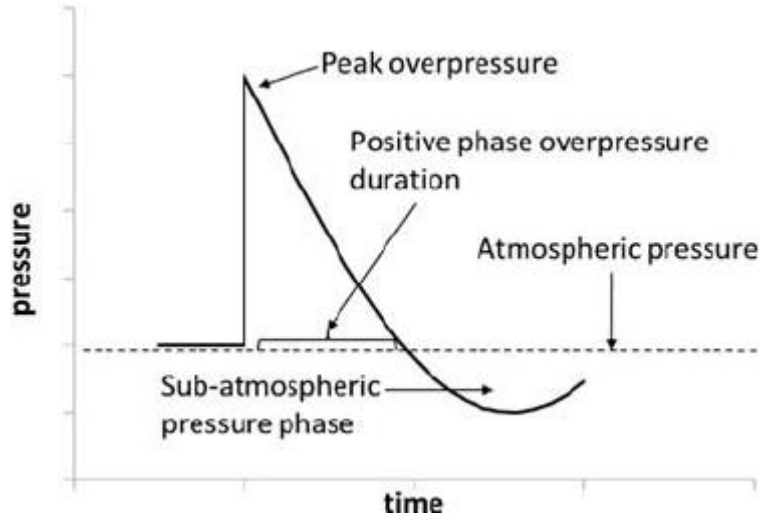


Figure 8. Pressure trend over time after an explosive event (Ramasamy et al., 2013).

Generally, an overpressure of 0.70 bar is considered as resulting in total demolition of constructions. Human lungs are generally highly resistant to overpressure, while the main death cause in case of explosion is due to impact with hard surfaces.

- Fragment generation: as a result of explosion fragments can be either generated from the initial tank or container involved in the explosion – primary missiles – or from objects close to it – secondary missiles. The energy delivered to them from the blast wave can transform the fragments into missiles characterised by high velocity and penetration effects.

- Thermal radiation: typically caused by fireballs growth. These fires consist in immediate ignition of the aerosol cloud above the pool generated after the rupture of the tank containing the liquid hydrogen under pressure and can cause damage to humans by direct contact with the flame or by radiation. Secondary fires can be generated by radiation exposure.

The condensed phase explosion may or may not occur if a liquid hydrogen – condensed air mixture is ignited depending on some conditions. Many experimental tests were performed in the past in order to investigate the behaviour of the flammable mixture composed by liquid hydrogen and oxygen. Those tests established that in case of ignition of a liquid hydrogen – pure solid oxygen mixture, a rapid deflagration to detonation transition occurs and can still be observed if the solid oxygen is diluted with nitrogen to 50 % wt/wt (Atkinson, 2021). For higher nitrogen contents the mixture burns without exploding. This means that if air condenses on the surface of a cryogenic liquid spill, the resulting flammable mixture may or may not lead to a condensed phase explosion if ignited. The outcome depends on the composition of the frozen air and the extent to which it has been enriched with oxygen: oxygen has higher melting and boiling temperatures than nitrogen, therefore it may condense faster than nitrogen, leading to oxygen enrichment in the solidified deposit (Aziz, 2021).

More recent studies about large hydrogen spills scenarios onto concrete pads have shown significant condensed phase explosion following the initial ignition (secondary explosion), revealing that some flow conditions such as hydrogen to air ratio and wind conditions lead to oxygen enrichment and facilitate the deflagration to detonation transition (Atkinson, 2021). These studies revealed that ambient air is fully condensed if the liquid hydrogen to ambient air ratio exceeds a value of 7, as shown in *Figure 9*. In this case there is no oxygen enrichment.

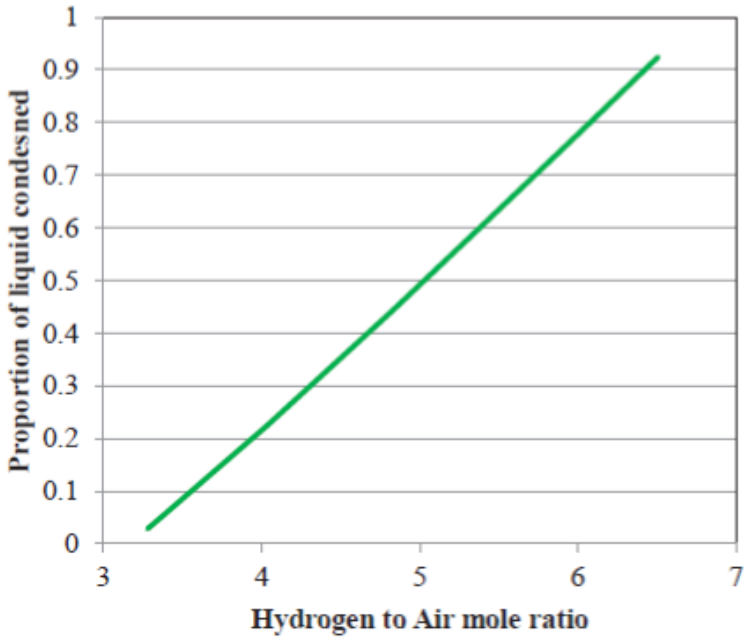


Figure 9. Equilibrium state of liquid hydrogen and air mixed in different proportions – Molar proportion of condensed air (Atkinson, 2021).

If more air is present the ratio falls. For a hydrogen to air ratio of 3 or less, no condensation of air is observed, it all remains entrained in the gas phase. If the hydrogen to air ratio is between 3 and 7, varying degrees of oxygen enrichment can be observed (see *Figure 10*). The first liquid (i.e. condensed air) formed as more hydrogen is mixed with air has an oxygen molar fraction of around 60%, reached at a temperature of around 72K (see *Figure 10* and *Figure 11*).

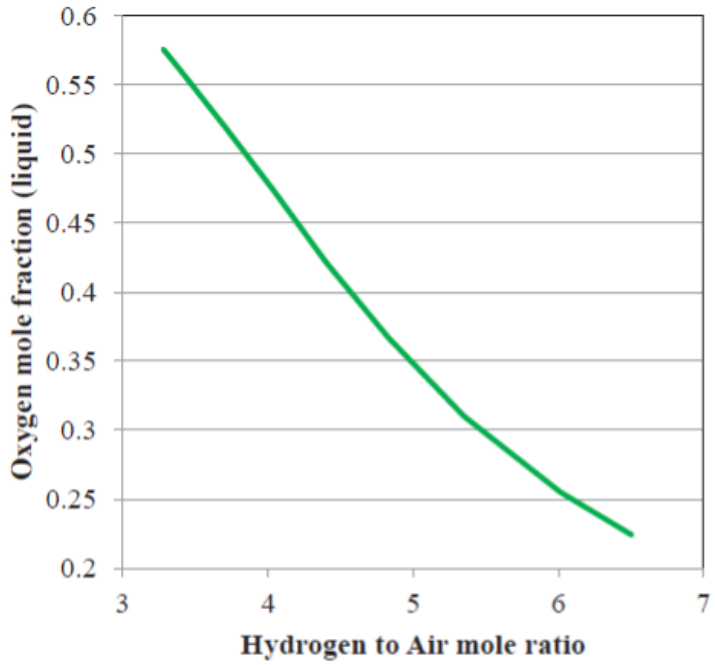


Figure 10. Equilibrium state of liquid hydrogen and air mixed in different proportions - Oxygen mole fraction in the liquid phase (Atkinson, 2021).

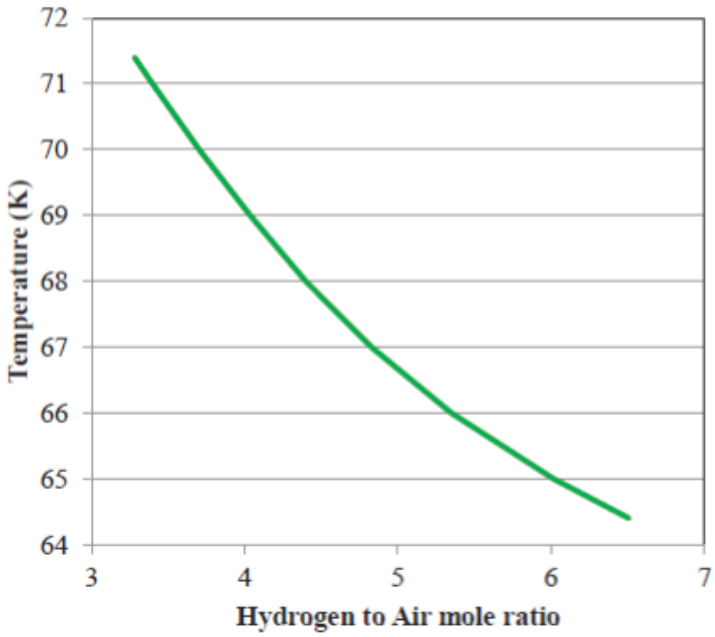


Figure 11. Equilibrium state of liquid hydrogen and air mixed in different proportions – Final Temperature (Atkinson, 2021).

Another hazard connected to the leakage of liquid hydrogen is related to the vapour cloud which is formed immediately after the release. This cloud may be characterised by the concentration of gaseous hydrogen in air. If this concentration reaches the lower explosivity limit, a

flammable atmosphere, which could cause a simple fire or an explosion if ignited, arises. When a leakage occurs, part of the cryogenic liquid flash vaporises forming a flammable aerosol (Liu et al., 2019). If it is immediately ignited a jet fire continuously fed by the leakage is originated. On the other hand, if an immediate ignition is not present, a cloud is formed and it spreads as a plume from the release point along the wind direction, diluting with air. If the dilution is fast enough, the cloud disperses safely, whereas, if the cloud finds a delayed ignition (i.e. 1÷5 minutes after the release began) a fire or an explosion are generated:

- vapour cloud fire: fire with no explosive effects;
- vapour cloud explosion: fire with explosive effects. It occurs when the flame front accelerates to a velocity higher than 40 m/s in presence of partial confinement (Thomas, Eastwood and Goodrich, 2015).

Since the consequences of an explosion event are extremely severe – due to the above-mentioned aftermaths – predicting whether or not the concentration of hydrogen in the cloud will reach the lower flammability limit is crucial (Pritchard and Rattigan, 2010).

2.3 Mitigation measures

Once the release has occurred, some safety procedures should be automatically activated. Typically, sensors and detectors can be used to detect hydrogen leakages. These sensors should incorporate automatic shutoff to limit the amount of liquid hydrogen released and activate alarms to warn the operators. A good practice is to set the detector's set point at 1% hydrogen by volume in air, which corresponds to 25% of the lower flammability limit (LFL), (Hydrogen Tools, 2022). Sensor and detectors might also be designed so that they could activate mitigation tools, such as sprinklers, water curtains or release inert gases in case of hydrogen leakage detection, as well as ventilation.

Many tests have been performed in order to guarantee the possibility to utilise such tools as mitigation measures to control the flow or accumulation of liquid hydrogen and the dispersion of hydrogen gas, as described by Atkinson (2020).

Ventilation is one of the most important mitigation measures since it ensures that the flammable gas is prevented from accumulating and it can either be natural – due to pressure differences between indoor and outdoor spaces, or due to the wind blowing in outdoor spaces – or mechanical – achieved using fans (Spoelstra, 2020).

Sprinklers and water curtains are water systems typically used in the firefighting framework, since they can be automatically activated by high temperature.

Automatic sprinklers are heat-sensitive devices designed to activate at certain temperatures to release a stream of water, and to distribute it in a specified pattern. In normal conditions, water is entrained by a cap or valve held tightly against the nozzle. Then, it is sprayed out by the releasing mechanism in case of activation. The most commonly used release mechanisms include fusible links, glass bulbs, and chemical pellets. Sprinklers are made for installation in an upright (SSU), pendent (SSP), or sidewall position (Spoelstra, 2020).

A water curtain is defined as a line of closely spaced sprinklers. Water spray curtains are used for the dispersion of hazardous vapor clouds through different mechanisms (Rana and Mannan, 2010 and Rana, Guo and Mannan, 2010):

- mechanical effects of creating a barrier to the passage of a gas cloud and imparting momentum to it – the imparted momentum can disperse the vapour cloud upwards, downwards or sideways depending on the nozzle orientation;
- dilution of the gas cloud by air entrained by the water spray;
- thermal effects between the gas cloud, water droplets, and entrained air – it can play an important role in dissipating cold clouds by enhancing their buoyancy;
- absorption of gas in water droplets with or without chemical reaction.

The main problem when using water system is related to the potential generation of a rapid phase transition (RPT), which might occur when a cryogenic liquid vaporises violently upon encountering water (Verfondern, 2021). It is not a chemical explosion since it is not provoked by a combustion or other chemical reactions. Instead, RPT events are vapor or physical explosions. However, they are powerful enough to determine major damage. The experimental studies performed by the European project PRESLHY (Verfondern, 2021) proved that such explosion does not occur when water droplets get in contact with liquid hydrogen, hence both sprinklers and water curtains are effective in mitigating liquid hydrogen release's consequences. However, the contact with water enhances the rate of vaporisation and if the cloud is ignited it could lead to a larger fireball (Verfondern, 2021). These tools are fast acting, a typical response time is about 3 minutes after the activation temperature has been reached (Wade et al., 2007).

Inert gases, such as nitrogen, argon, helium, carbon dioxide can be introduced inside an enclosed room where hydrogen has been released in order to reduce the concentration of oxygen

in air. Since an explosion needs a combustible, an oxidizer (oxygen) and an ignition to occur, if oxygen concentration is lower than the flammability limit because of the presence of an inert gas, the explosion cannot occur, as represented schematically in *Figure 12*, where point C represents the minimum concentration of oxygen below which the mixture cannot burn if ignited (Baldissin, 2015).

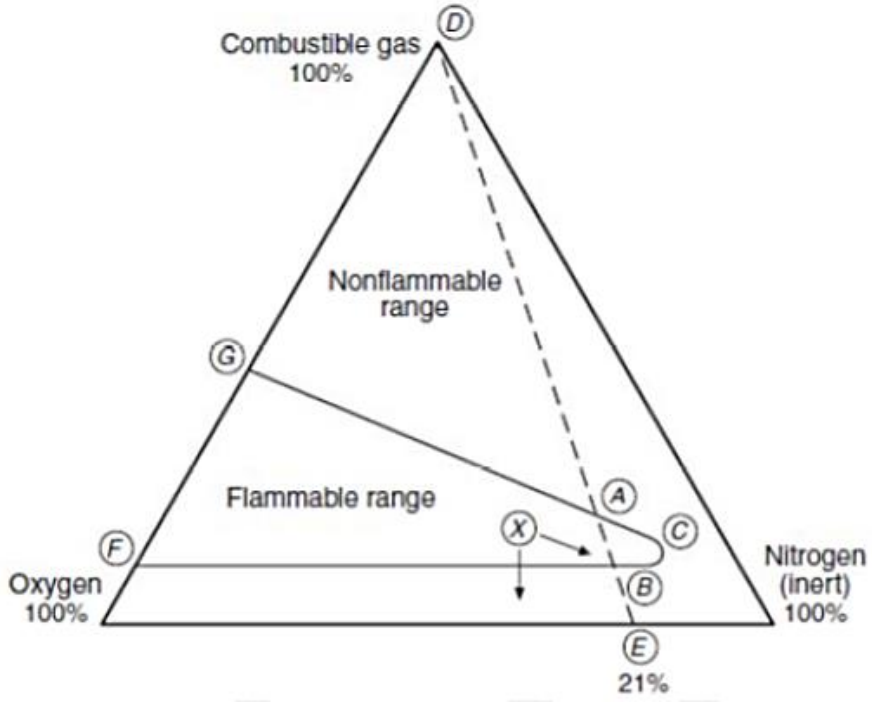


Figure 12. Explosivity triangular diagram related to a generic combustible gas (Baldissin, 2015).

The lower concentration of oxygen also guarantees to avoid its solidification which could bring to a severe condensed phase explosion in case of oxygen enrichment of the flammable mixture, as described in section 2.2. The main drawback of this approach is that the room must be immediately evacuated before entering the inert gas, since oxygen depletion is responsible of asphyxia, which is lethal for human being; therefore, the response time increases (Arrieta et al., 2009).

All of these mitigation measures are time-sensitive: they typically have a response time of a few minutes, but the sooner they are activated, the more effective their action is. Therefore, the prediction of a liquid hydrogen release's consequences is crucial to establish and quickly activate the most appropriate mitigation measure. Machine Learning techniques allow to build models able to carry out these kinds of predictions.

2.4 Machine Learning approach

Machine learning is a field of Artificial Intelligence (AI) (Brink, Richards and Fetherolf, 2016). Artificial intelligence can be described as "the effort to automate intellectual tasks normally performed by humans" (Chollet, 2017).

The history of AI began in 1950, when AI was simply a predefined set of rules that applied to an input dataset would return some output. Nowadays artificial intelligence has evolved into a much more complex field involving many different techniques such as Machine Learning, Neural Networks (NN) etc. (see *Figure 13*).

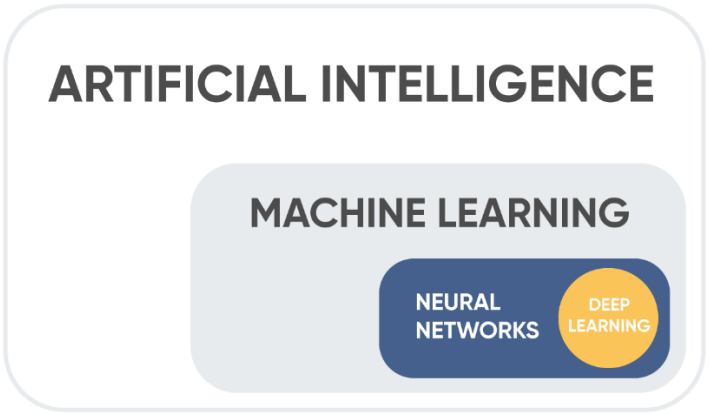


Figure 13. Artificial Intelligence's constituent parts (ActiveWizards, 2019).

The family of AI known as machine learning (ML) has become an important tool for gaining information from large data sets (Ben-David and Shalev-Shwartz, 2014). The term refers to "the automated detection of meaningful patterns in data" (Ben-David, and Shalev-Shwartz, 2014), which allows a machine to acquire knowledge from the past. In other words, whereas AI requires human to define a set of rules, ML is able to figure out the rules itself knowing the input data and what the output should be like.

Some of ML applications are here listed and many more are discovered every day (JavaTPoint, 2021):

- image recognition;
- risk management;
- traffic prediction;
- product recommendations;
- self-driving cars;

- spam detection;
- medical diagnosis;

There are three main categories of machine learning (ActiveWizards, 2019):

- supervised learning: the model is fed with both input and output data to perform a task, such as Classification (for categories or classes prediction) or Regression (to predict a continuous outcome);
- unsupervised learning: the model is fed only with the input data;
- reinforcement learning: the system is rewarded for correct outcomes and penalised for wrong ones developing an algorithm with the highest reward and the lowest penalty.

Since the aim of this thesis is to determine whether or not a certain hydrogen release will cause air component's liquefaction or solidification or if the resulting cloud will be or will be not characterised by a concentration of hydrogen higher than the LFL, Classification only has been used. In the next section, a more detailed description of the Classification task is given.

2.4.1 Supervised Learning

Before diving deeper into Classification's general aspects, it is necessary to introduce some definitions (Mohri, Rostamizadeh and Talwalkar, 2018), and in order to make them clearer a simple example is used. The Titanic's passengers are taken into consideration, and the task of the learner is to guess whether a certain passenger will survive or not (TensorFlow, 2022a).

- Example: instance of data or item used for learning. In the Titanic example these items correspond to the collection of passengers.
- Features: vector of attributes associated to an instance of data. In the example above some relevant features may be the age, the sex, the class, etc. of the passenger. The values of the features can either be numerical (e.g. 24, as the passenger age), categorical (e.g. Female, as the passenger sex) or Boolean (Nilsson, 2005). If the task is to predict the liquefaction or freezing of air components due to liquid hydrogen leakage, some meaningful features, for instance, might be the release rate and orientation, as well as the pressure inside the tank that leaks and the wind characteristics.

- Labels: values or categories assigned to an example that have to be predicted by the model. In the Titanic example the label is ‘survived’, or ‘not survived’.
- Train dataset: set of examples used to train the model (e.g. set of passengers), including corresponding labels.
- Test dataset: set of examples used to validate the model and evaluate its performance. The algorithm predicts the labels associated to these examples and then they are compared to the actual labels to evaluate the performance.

The previously described example is a supervised learning binary classification problem, which is the one used in this thesis. It might be reduced to the following question: given an example x from a dataset X, which value between 0 (associated to the non-happening of a certain event) and 1 (associated to the happening of a certain event) the corresponding label will assume (Smola, and Vishwanathan, 2008)? In order to answer this question, the model has to be trained and tested. First, relevant features must be associated to each example of the train dataset. Second, these features are used to train the model. The aim of the training step is to find a function that correlates the labels to the features (Brink, Richards and Fetherholf, 2016). This function is also called the model of the ML algorithm (TensorFlow, 2021a).
Finally, the test dataset, which the model has not seen yet, is used to predict the labels. The output of the evaluating phase is a probability vector: the first value is the probability of the label being 0 and the second value is the probability of the label being 1. These probabilities are then compared to a threshold value (0.5 by default) to determine the predicted labels that will be compared to the actual labels to evaluate the performance of the model by calculating a loss function. Such function can either be represented by a mean absolute error, a mean squared error, etc.

2.4.2 TensorFlow

In the past decades, machine learning techniques have led to advances in many different fields, especially due to the availability of new ML models and software platforms for building such models (Abadi et al., 2016).
TensorFlow is an open-source programme for machine learning designed by Google that allows to easily build models no matter what programming language is used (TensorFlow, 2022b).
Some features of TensorFlow might be summarised as follows:

- it is able to easily define, optimise and calculate mathematical expression using multi-dimensional arrays, also known as tensors;
- it includes a coding support of deep neural networks and machine learning approaches;
- it uses GPU computing.

TensorFlow offers a variety of machine learning and deep learning algorithms and includes a consistent number of ML libraries. The basic data structure used in TensorFlow language is the tensor, which is characterised by a rank, that is the number of dimensions of the tensor, a shape, that is the number of rows and columns, and the data type of a tensor's elements. Tensors are the connecting edges of the data flow graphs used to execute numerical computation (Tutorials Point, 2018).

Generally speaking, the aim of this work is to build three different models (Linear, Deep and a combination of these two – Wide&Deep) using TensorFlow's estimators, which, given inputs and a number of parameters, return the necessary operations to perform training, evaluation or predictions (TensorFlow, 2021b).

2.4.3 Models

Many models are available to perform a Classification task based upon algorithm such as Random Forest, Decision Trees, etc. (Shin, 2020). In this thesis, a Linear model, a Deep Neural Network and a hybrid Wide&Deep model are used in the framework of the python library Tensorflow.

2.4.3.1 The Linear Model

A linear Classifier in Machine Learning is a method that "makes classification decision based on the value of a linear combination of characteristics of an object" (Guru99, 2021). Therefore, these classifiers separate data using a line, a plane or a hyperplane - which represents a plane in more than 2 dimensions - (Machine Learning Notebook, 2021), as depicted in *Figure 14*.

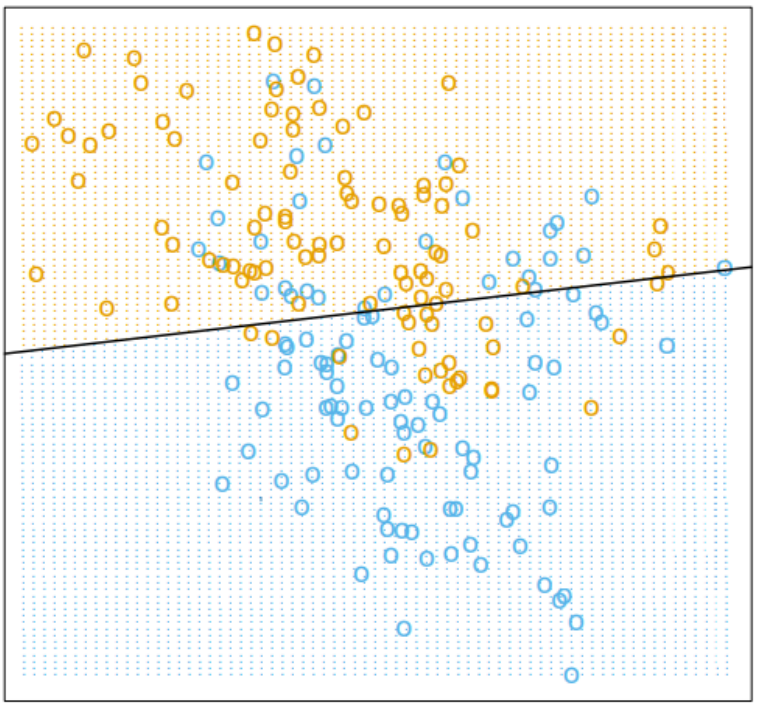


Figure 14. Binary classification example: the classes are coded as a binary variable (blue=0, orange=1). The line represents the decision boundary defined as $X^T W = 0.5$ where X is the inputs vector and W is the weights vector (Hastie, Friedman and Tibshirani, 2009).

Three major algorithms are synthetically explored in this section, but only the first one has been used to demonstrate the approach:

- Perceptron: it consists in a transfer function and an activation function. The transfer function, takes the features vector as input, and its output is then transferred to the activation function, as in Equation (1):

$$Y = w_o + \sum_{j=1}^{p} x_j w_j = f(w, x) \qquad (1)$$

  Equation (1) can be written as an inner product: $Y = X^T W$ (Hastie, Friedman, and Tibshirani, 2009).

  Where Y represents the output (vector of predicted labels), X the vector of inputs (it is usually a matrix containing the features associated to each object), w_o is the intercept or bias and W is the vector of weights. Y is then defined as a linear function of all the features x_j. The weights represent the direction of the correlation between features and labels: a positive correlation rises the probability of the positive class whereas a negative correlation increases the probability of the negative class (Guru99, 2021). The activation function acts like a threshold in this case, as shown in Equation (2).

$$output = \begin{cases} 1, f(w,x) \geq 0 \\ 0, f(w,x) < 0 \end{cases} \quad (2)$$

- Logistic regression: it is based upon a sigmoid function defined as follows

$$h(z) = \frac{1}{1+e^{-z}} \quad (3)$$

The sigmoid function in Equation (3) represents the activation function, it receives a weighted linear combination of features as input ($X^T W$) and gives a number between 0 and 1 as output, which is the probability of observing a certain label.

- Support vector machine: given some linearly separable data many hyperplanes can act as a separation boundary, SVM chooses the optimal one, as represented in *Figure 15*.

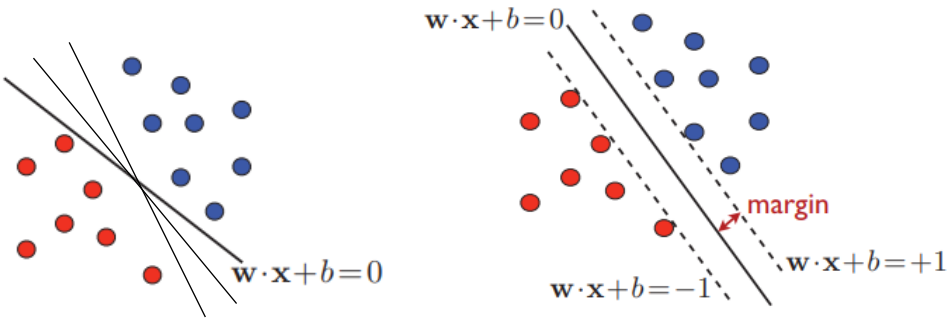


Figure 15. On the left: possible decision boundary hyperplanes. On the right: optimal decision boundary hyperplane (Mohri, Rostamizadeh and Talwalkar, 2018).

The optimal hyperplane is the one that has the maximum margin, which is the maximum distance from the data points of each class.

The linear model is generally fast, easy to interpret and suitable for analysing large sets of features as well as reliable (Hastie, Friedman, and Tibshirani, 2009). Moreover, inter-features relations' effect on the output can be taken into consideration by combining different features into a single column, obtaining the so-called Crossed Features (TensorFlow.org, 2020c).
The main drawback of linear models is that they cannot describe nonlinear relations between features nor even express the effect of previously unseen combinations of features (Cheng et al., 2016).

2.4.3.2 The Deep Neural Network

Neural networks are nonlinear statistical models. They can be represented by a network diagram as depicted in *Figure 16*.

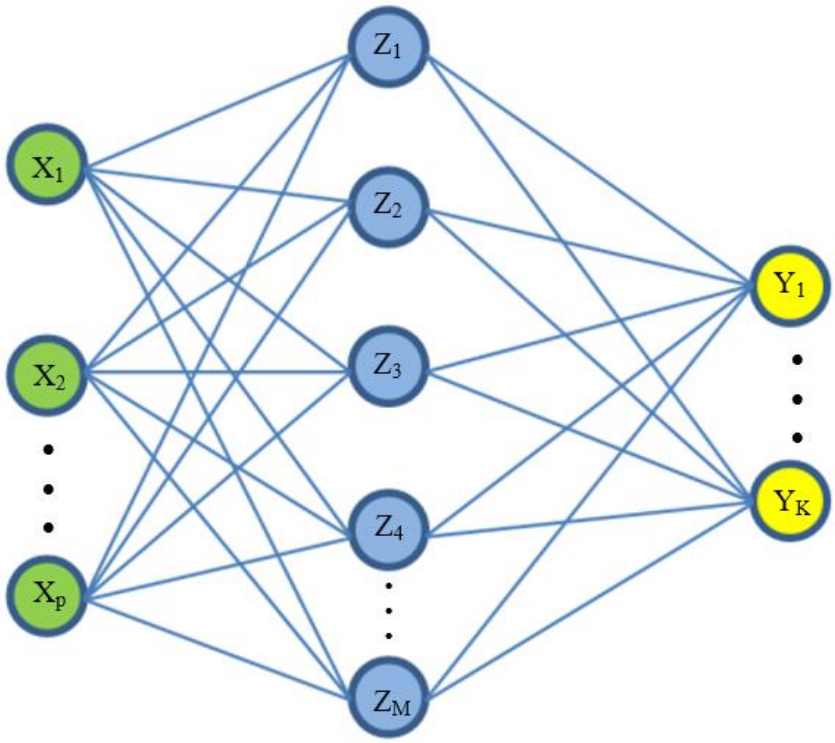


Figure 16. Neural Network diagram, adapted from Danilov and Karpov (2018).

For K-class classification, there are K units on the right, which represent the probabilities of each class. For a binary classification method as the one analysed in this thesis K=2.

X_1, …, X_p are the inputs, which are linearly combined to obtain the derived features Z_1,..Z_M. These units are known as hidden units, and they constitute a hidden layer. A neural network may consist of several hidden layers, in that case we talk about deep neural networks (Kriegeskorte and Golan, 2019). Generally speaking, it is better to have too many hidden units rather than too few, otherwise the model might not be flexible enough to catch nonlinearities in the data. Choosing a good number of hidden layers is led by experience and background knowledge (Hastie, Friedman, and Tibshirani, 2009).

The output is then modelled as a function of the linear combination of such derived features:

$$Z_m = \sigma(\alpha_{om} + a_m^T X),\ m = 1, \dots, M \quad (4)$$

$$T_k = \beta_{ok} + \beta_k^T Z,\ k = 1, \dots, K \quad (5)$$

$$f_k(X) = g_k(T),\ k = 1, \dots, K \quad (6)$$

where Z = (Z_1,... ,Z_M) is the vector of the derived features, T = (T_1,T2,... ,T_K) is the vector of the linear combinations of the derived features and:

- α_{0m} = bias;
- α_m = vector of model parameter*s*;
- σ = activation function;
- β_{0k} = bias;
- β_k = vector of model parameters;
- Y_k = a label;
- g_k = the output function.

The activation function σ(v) is usually chosen as a sigmoid (Hastie, Friedman, and Tibshirani, 2009). The output function allows to finally convert the vector of outputs T.

The output function initially utilised was the identity function $g_k(T) = T_k$, mainly applied to regression tasks, which nowadays has been replaced by the softmax function in Equation (7):

$$g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}} \quad (7)$$

The neural network parameters, or weights, are tuned during the training phase using either the squared error or the cross-entropy deviance (Hastie, Friedman, and Tibshirani, 2009):

$$R(\theta) = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} log f_k(x_i) \quad (8)$$

The approach is to minimize R by gradient descent, through a back-propagation algorithm: first the activation states are computed upward for each unit; then, the predicted outputs are compared to the expected ones and the deviance is calculated (see Equation (8)). This allows to calculate the sensitivity of the cost evaluating how much the deviance would change by varying the activation.
Finally, each weight is adjusted in the direction that minimises the error (Kriegeskorte and Golan, 2019).

As a general consideration, Deep Neural Networks, in opposition to linear models, are able to capture even nonlinear relations between features, can generalise better and can produce decision boundaries of any shape, as one can notice in *Figure 17* (Hastie, Friedman, and

Tibshirani, 2009). On the other hand, though, they require a higher computational effort and are more sensitive to the quality of the input dataset (Brink, Richards and Fetherolf, 2016).

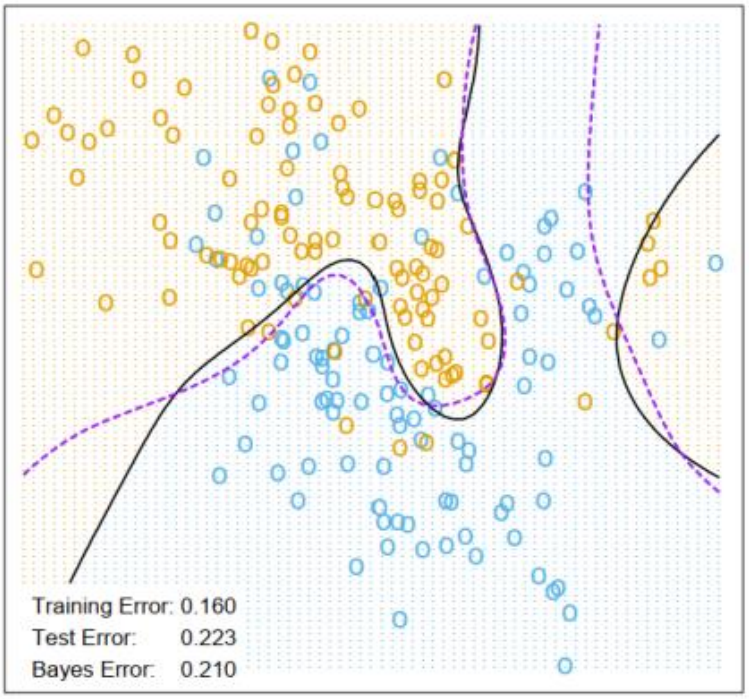


Figure 17. A neural network example: representation of two decision boundaries (Hastie, Friedman, and Tibshirani, 2009).

2.4.3.3 The Wide&Deep Model

The Wide&Deep Learning was developed by Google for recommender systems.

It was developed to overcome the previously described models' flaws. The main feature of wide and deep learning is the integration of a wide linear model and a deep neural network, in order to achieve both their respective powers of memorisation and generalisation (see *Figure 18*).

The wide learning part consists in a generalised linear model responsible for memorisation, that "emphasises on frequent co-occurrences of the features in the past" (Bastani, Asgari and Namavari, 2019).

The Deep Neural Network (DNN) part consists of a nonlinear statistical model responsible for generalisation, which exploits new inter-relations between features that have never or rarely occurred in the past. In these cases, in fact, wide learning would fail as there are no data to train the model. On the other hand, deep learning can generalise to previously unseen interactions.

The resulting model is able to achieve both tasks by integrating the two approaches – memorisation and generalisation – being able to account for unseen features interactions, but

still avoiding the over-generalisation typical of deep learning. The training of wide and deep parts is carried out jointly by backpropagation (Bastani, Asgari and Namavari, 2019).

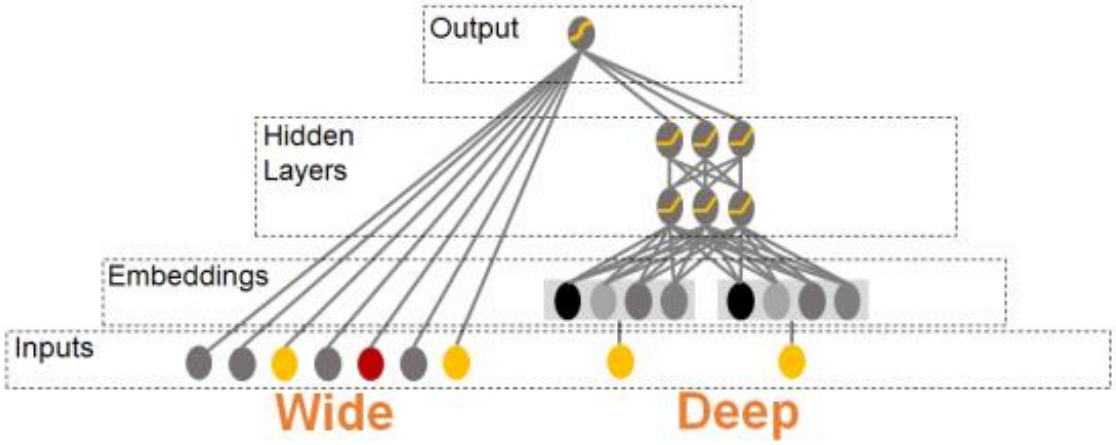


Figure 18. Illustration of the Wide&Deep model (Bastani, Asgari and Namavari, 2019).

2.4.4 Performance metrics of machine learning models

Performance metrics are vital in evaluating a model's performance. Among the wide number of performance metrics available, only those which better evaluate the specific model's performance must be selected. In binary classification problems, the confusion matrix depicts the total number of predictions dividing them in four possible outcomes (see *Figure 19*):

- True Positive (TP): both the real label and the predicted label of a sample are positive (= 1);
- True Negative (TN): both the real label and the predicted label of a sample are negative (= 0);
- False Positive (FP): the real label is negative and the predicted label is positive;
- False Negative (FN): the real label is positive and the predicted label is negative (Jiao and Du, 2016).

Most of the classifier's performance metrics are obtained combining these four values.

		Predicted Label	
		0	1
Real Label	0	TN	FP
	1	FN	TP

Figure 19. Confusion matrix. TP = true posisitve, FP = false positive, FN = false negative, TN = true negative

Based on the definitions above three basic performance metrics can be obtained as follows:

- $Accuracy = \frac{TN+TP}{TN+FN+TP+FP}$ which expresses the fraction of predictions correctly performed by the model (Google, 2020a);
- $Precision = \frac{TP}{TP+FP}$ which expresses the fraction of correct positive predictions;
- $Recall = \frac{TP}{FN+TP}$ which expresses the fraction of real positive labels correctly predicted, (Google, 2020b).

A classification algorithm uses a parameter known as decision threshold to decide which class a given sample belongs to. The default threshold is 0.5, but this parameter can be changed in case of class imbalance in the initial dataset, and this leads to a modification of the values inside the confusion matrix. More specifically, by decreasing the decision threshold the number of FP increases whereas the number of FN decreases (Seliya and Hulse, 2009). Therefore, precision and recall vary as functions of the decision threshold and for given values of the threshold, a precision-recall curve can be obtained.

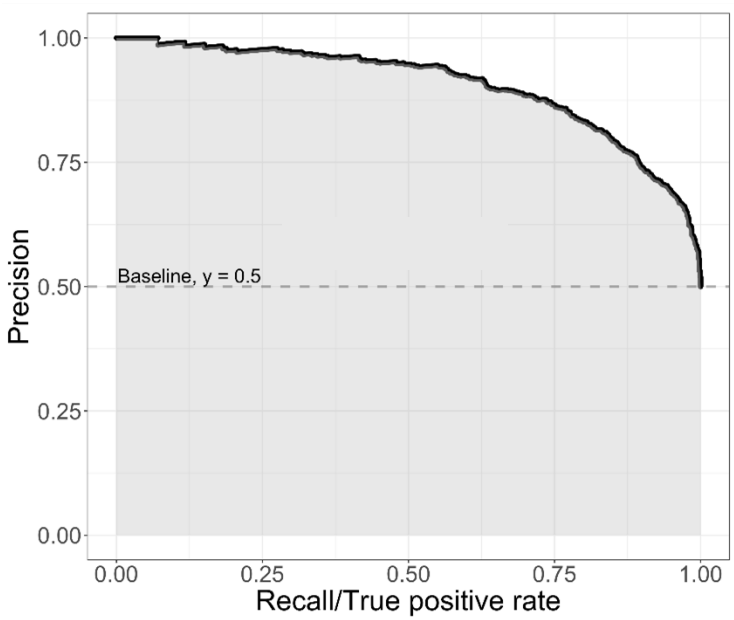


Figure 20. Example of Precision-Recall curve obtained for different threshold values (GitHub,2020a).

The Precision-Recall curve introduces another important performance metric which is the Area Under the Precision-Recall Curve (AUCpr) (see *Figure 20*). This is represented by a single value ranging from 0 to 1; the higher the AUCpr the better the classifier's performance. (Seliva and Hulse, 2009).

Finally, another significant metric is the F-measure, that can be calculated as a function of Precision and Recall:

$$F - measure = \frac{(1+\beta^2)\cdot Precision\cdot Recall}{\beta^2\cdot Precision+Recall} \quad \text{(Chinchor, 1992)} \qquad (9)$$

Where $\beta = 1$ if precision and recall are of equal weight, and the F-measure in this case would be their harmonic mean, $\beta = 1.5$ if recall's optimisation is more important than precision's optimisation, and $\beta = 0.5$ if recall is half as important as precision (Chinchor, 1992). The maximum value of the F-measure corresponds to the best threshold value, that optimises precision or recall or both the parameters.

Chapter 3
Databases and simulations

The databases used in this work have been obtained from data collected during experimental tests performed in England by the FFI (Aaneby, Gjesdal and Voie, 2021), as described in Paragraph 2.1. The analyses performed in the thesis are here presented and described in detail. Specifically, the step-by-step procedure followed to build the Machine Learning models is outlined.

3.1 Data used

Three different databases have been developed:

- First database: outdoor leakage studies – condensation or freezing of air components prediction;
- Second database: outdoor leakage studies – hydrogen concentration within the gas cloud prediction;
- Third database: closed room studies, also referred to as indoor leakage studies for simplicity – condensation or freezing of air components prediction.

All the databases consist of temperature and hydrogen concentration values measured by several thermocouples and oxygen sensors (to determine the concentration of hydrogen in air by oxygen depletion) respectively, placed nearby the liquid hydrogen release. Each row of the database represents the temperature or concentration value given by each element in a certain timestamp. Each column represents a piece of information characterising the release.
The common attributes between the three databases are listed in *Table 1*.

Table 1. Common features between the three databases.

Feature	*Description*
Timestamp	Time elapsed after the beginning of the release test
Ambient pressure	Ambient pressure measured by a sensor

Relative humidity	Relative humidity measured by a sensor
Release rate	Rate that characterises the release scenario
Release orientation	Direction of the release hose, can either be horizontal or vertical downwards
Internal pressure	Liquid hydrogen tank internal pressure measured by four sensors
Internal temperature	Liquid hydrogen tank internal temperature measured by four thermocouples
Temperature values	Temperature values measured by the thermocouples
Thermocouples positions	Spatial coordinates of each thermocouple

Additional features for the outdoor leakage studies databases:

Table 2. Outdoor leakage studies databases' additional features

Feature	*Description*
Wind direction	Wind direction detected at 10 m and 5 m above the ground
Wind speed	Wind speed detected at 10 m and 5 m above the ground

Additional features for the enclosed room studies database:

Table 3. Indoor studies database's additional features.

Feature	*Description*
Purge	In some of the experimental studies the container is either purged with air or nitrogen
Sealing	The container is characterised by two openings: the low-level vent (connection point between the TCS and the hydrogen tank) and the opening connecting the ventilation mast to the container. The experimental tests might be conducted with all of the openings sealed, none of them sealed or only the low-level vent sealed

Not all the thermocouples used in the experimental studies are considered in the databases. As discussed in Paragraph 1.2, the main objective of this thesis is to build a model able to predict the condensation or solidification of air oxygen due to the accidental release of a cryogenic fluid in the atmosphere. Since many thermocouples are placed considerably far away from the release point, it is highly improbable that such thermocouples would reveal the presence of liquid or solid oxygen, so they have been excluded from the database. All the thermocouples placed over 30 m away from the release point have been excluded from the first database used to predict the condensation or freezing of air components. Moreover, the thermocouples placed inside the test pad were not considered in the development of the database as well, since no air can be found under the cement pad's surface.

In order to evaluate whether the concentration of hydrogen in the gas cloud reaches the lower flammability limit or not, all the oxygen sensors placed at 30, 50 and 100 m from the release point in the wind direction, as represented in *Figure 3*, have been considered in the second database.

For the third database, only the 25 thermocouples placed inside the TCS were considered to build it in order to evaluate the formation of liquid or solid oxygen inside the TCS and to perform a comparison with the outdoor case.

Finally, as mentioned before, the databases have been developed by considering a row for each temperature or concentration value measured by every single thermocouple or sensor for each instant of time, for the entire duration of the experimental test. This has been repeated for every test and all the data collected for each of them have been merged.

A simplified representation of the first outdoor leakage studies database is reported in *Table 4* to provide an example. For a better understanding of the databases' structures see the tables in Appendix B.

Table 4. Simplified representation of the First database (outdoor leakage studies).

Timestamp (s)	*Ambient pressure (mbar)*	*...*	*Temperature values (°C)*	
t_0	P		TT01	
...	P		...	*Test 1*
t_0	P		TT48	...
...	...		...	

t_f	P		TT01	
...	P		...	
t_f	P		TT48	
t_0	P		TT01	
...	P		...	
t_0	P		TT48	...
...	...		...	*Test 7*
t_f	P		TT01	
...	P		...	
t_f	P		TT48	

3.1.1 Labels

One or more labels have been associated to each timestamp based on the value measured by the thermocouple or sensor. The label must refer to a future event in order to allow a prediction using the previously described models.

For the first and third databases two labels have been investigated: for each position on the pad, liquid oxygen or solid oxygen deposition may or may not be observed. The label is 1 if the value measured by the specific thermocouple after 200 s is lower than the boiling point – to predict the condensation of air oxygen – or the melting point – to predict the solidification of air oxygen – of pure oxygen. Only in one case this method cannot be applied: in one of the experimental tests of the indoor studies the enclosed room was previously purged with nitrogen, therefore no oxygen was present inside the room once the release had started. For this specific situation a label of 0 has been associated to each thermocouple for every timestamp.

For the second database only one label has been considered: for each position on the pad a concentration of hydrogen in air higher than the lower flammability limit may or may not be observed after 200 s.

Both the oxygen boiling and melting point have been considered at an average pressure of 0.96 bar, and they are equal to T_b = -183.5 °C and T_m = -218.79 °C, respectively (Perry and Green, 2008).

The lower flammability limit of hydrogen considered at atmospheric pressure is 4% vol. (Hochgraf, 2009).

In this work, a response time of 200 seconds has been considered to predict the condensation or freezing of air components or the formation of a flammable atmosphere.

The models will then be able to carry out a prediction of the label corresponding to a certain thermocouple after 200 s. Since in this thesis hydrogen sensors activated systems have been considered as mitigation measures for liquid hydrogen leakage's consequences, the 200 s interval has been selected on the basis of water systems' response time after a hydrogen release.

3.1.2 Database analysis

The first database contains five experimental tests with a total of 48 thermocouples each. The second database contains four experimental tests with a total of 30 concentration sensors each. The third database contains five experimental tests with a total of 25 thermocouples, all of them placed inside the container.

Not all the experimental tests have been considered in the databases since some of them were not long enough to allow a consistent prediction or were conducted in different wind conditions with respect to the others.

3.1.2.1 Outdoor leakage studies databases

In order to better analyse the different databases, the histograms representing the features' distributions are here discussed. The abscissa axis gives the values assumed by the feature into consideration, whereas the ordinate axis represents the number of repetitions of that value in the database.

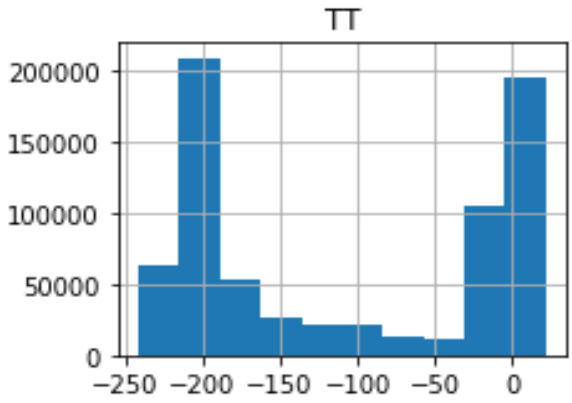


Figure 21. Distribution of the temperature values measured by the thermocouples (TT) considered in the first database (outdoor case).

As shown in *Figure 21* the temperature values measured by the different thermocouples vary between 0°C and -250°C, with the larger number of repetitions being at the extremes of the distribution.

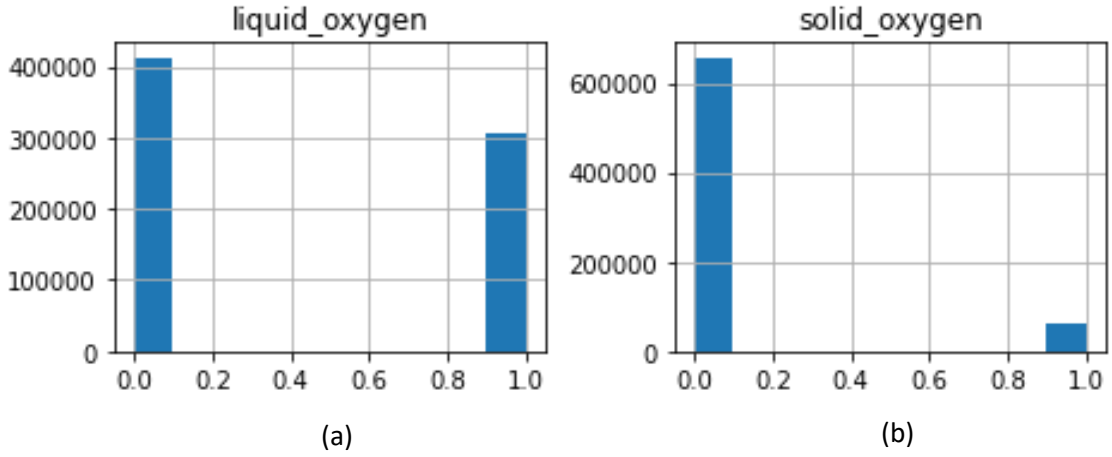


Figure 22. Distribution of the label values considered in the first database (outdoor case) for the labels (a) liquid oxygen formation and (b) solid oxygen formation.

Figure 22 represents the labels taken into consideration for the first database: a value of 1 is assigned to those thermocouples which will measure a temperature lower than the boiling or melting point of oxygen after 200 s, whereas a value of 0 is assigned to those thermocouples which will measure a temperature value higher than the boiling or melting point of oxygen. As expected, liquid oxygen is formed on the ground many more times than solid oxygen, since the boiling point, being higher, is more easily reached than the melting point.

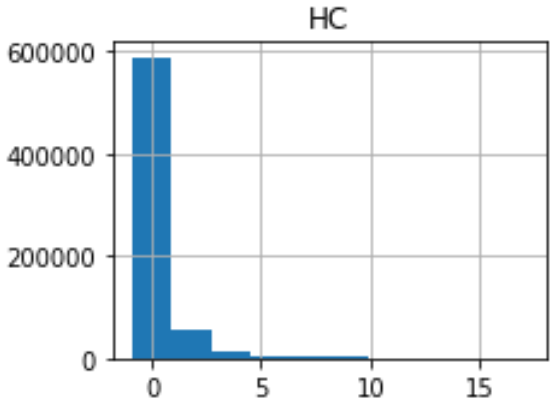


Figure 23. Distribution of the hydrogen concentration (HC) values measured by the oxygen sensors in the second database.

The hydrogen concentration (HC) values measured by the different oxygen sensors vary between 0% and around 10%, with the larger number of repetitions being at the lower extreme of the distribution, as displayed in *Figure 23*.

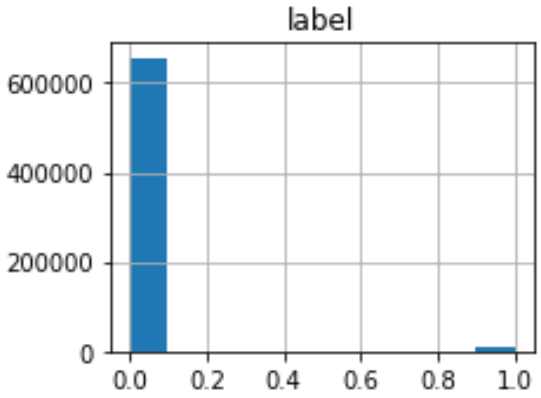


Figure 24. Distribution of the label values considered in the second database (outdoor case).

The chart in *Figure 24* reveals that the second database is strongly imbalanced, being the negative label detected many more times than the positive one, which indicates that experimentally the concentration of hydrogen in air after the release rarely reaches the lower flammability limit of 4% that guarantees the fire to occur in case of ignition of the flammable mixture.

3.1.2.2 Enclosed room and ventilation mast studies database

Figure 25 highlights that the temperature values measured by the different thermocouples vary between 0°C and -250°C, with the larger number of repetitions being at the lower extreme of the distribution, which then smooths towards 0°C.

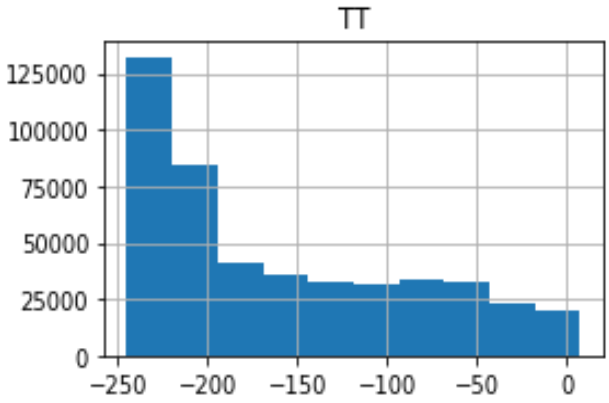


Figure 25. Distribution of the temperature values measured by the thermocouples (TT) considered in the third database (indoor case).

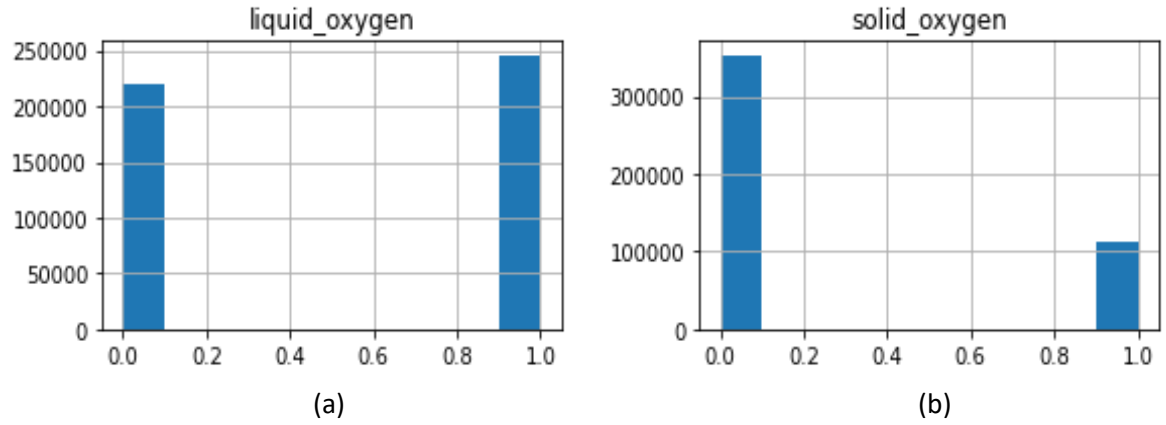


Figure 26. Distribution of the label values considered in the third database (indoor case) for the labels (a) liquid oxygen formation and (b) solid oxygen formation

Figure 26 represents the labels taken into consideration for the third database: a value of 1 is assigned to those thermocouples which will measure a temperature lower than the boiling or melting point of oxygen after 200 s, whereas a value of 0 is assigned to those thermocouples which will measure a temperature value higher than the boiling or melting point of oxygen. It can be stated that liquid oxygen is formed on the ground many more times than solid oxygen. Moreover, differently from what happens for the outdoor case, the temperature inside the container is low enough to allow a significant condensation of oxygen, so that the probability to find it in the liquid state is higher than the one to find it in a gaseous phase, as depicted in the first chart of *Figure 26*.

3.2 Procedure description

The first step in developing a machine learning algorithm is to build a database containing both features and labels (see Section 2.4.1). The selection of the most relevant features relies mainly on experience. Next, the database is split into two parts; the first one is used to train the model and the second one is used to evaluate it. Later, the model is selected (e.g. Linear, Deep or Wide&Deep), and the features are converted to fit the model's needs. Finally, the model is trained and evaluated. In this thesis Python has been used to code and build the models.

A more detailed description of these steps is presented in the following.

3.2.1 Database creation

Features and labels must be organised to form a database – a matrix – to prepare the data to feed the models with. The database is built in such a way that the features are reported in the columns of the database itself and the labels are stored in the last column. Each row of the database represents a list of the hydrogen release features for each sensor or thermocouple and the related label. The general structure of a hypothetic database is displayed in *Table 5*.

Table 5. Machine Learning Database: general structure.

Feature 1	*Feature 2*	*…*	*Feature n*	*Label*
t_0			TT0	1
...			...	…
t_f			TTf	0

The databases have been developed by considering a row for each temperature or concentration value measured by every single thermocouple or sensor for each instant of time, for the entire duration of the experimental test. This has been repeated for every test and all the data collected for each of them have been merged.

Selecting the most meaningful features often requires a trial-and-error approach. Only the most meaningful features that have been used in the final simulations are now presented.

First and second databases' features:

- the timestamp when the thermocouple or sensor measures a value of temperature or concentration;
- ambient pressure measured during the experimental test;
- release rate;
- relative humidity measured during the experimental test (RH%);
- release orientation;
- hydrogen tank internal pressure measured by the sensor P01;

- hydrogen tank internal pressure measured by the sensor P02;
- hydrogen tank internal pressure measured by the sensor P03;
- hydrogen tank internal pressure measured by the sensor P04;
- hydrogen tank internal temperature measured by the sensor PT_01;
- hydrogen tank internal temperature measured by the sensor PT_02;
- hydrogen tank internal temperature measured by the sensor PT_03;
- hydrogen tank internal temperature measured by the sensor PT_04;
- wind direction measured by a sensor placed at 10 m from the ground (Wind_Direction_High);
- wind direction measured by a sensor placed at 5 m from the ground (Wind_Direction_Low);
- wind speed measured by a sensor placed at 10 m from the ground (Wind_Speed_High);
- wind speed measured by a sensor placed at 5 m from the ground (Wind_Speed_Low);
- temperature or concentration values measured by the thermocouples and sensors (TT or HC);
- spatial coordinates of the instrumentation (x, y, z).

Third database's features:

- the timestamp when the thermocouple or sensor measures a value of temperature or concentration;
- ambient pressure measured during the experimental test;
- release rate;
- relative humidity measured during the experimental test (RH%);
- type of purge preformed (air or nitrogen);
- type of sealing (enclosed room partially or completely sealed);

- hydrogen tank internal pressure measured by the sensor P01;
- hydrogen tank internal pressure measured by the sensor P02;
- hydrogen tank internal pressure measured by the sensor P03;
- hydrogen tank internal pressure measured by the sensor P04;
- hydrogen tank internal temperature measured by the sensor PT_01;
- hydrogen tank internal temperature measured by the sensor PT_02;
- hydrogen tank internal temperature measured by the sensor PT_03;
- hydrogen tank internal temperature measured by the sensor PT_04;
- temperature values measured by the thermocouples (TT);
- spatial coordinates of the instrumentation (x, y, z).

The last column of *Table 5* represents the labels; a label can be either zero or one. If the thermocouple or sensor will measure the presence of liquid or solid oxygen or a concentration of hydrogen higher than the LFL within 200 s, the label is "1". Otherwise, the label is "0".

3.2.2 Data pre-processing

Nowadays it is a good practice to pre-process the data using a normalization method in order to avoid the higher impact of some features over others only due to their different scale. The normalization method used in this work is the MinMaxScaler and has been selected among others as it is the most utilised in various fields of application with good results. Therefore, all the variables are rescaled to be in the range [0,1] through the following approach:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$ (Pedregosa *et al.*, 2021a) *(10)*

3.2.3 Training and evaluation datasets

The database is split in two parts. The first part will be used for the training phase, the second part will be used to evaluate the performance (i.e. the prediction capability) of the trained models. The training database comprises the 75% of the original database. The remaining part

constitutes the evaluation database. In the original database, the data (i.e. rows) are displayed in chronological order and this may pose problems when splitting the database: for instance, it may happen that most of the positive labels "occurred" in the first part of the database (i.e. the training part) and very few positive labels would be included in the evaluation dataset. Machine Learning algorithms prefer well-distributed data, therefore before splitting the original database, the rows must be randomly shuffled to avoid poor data distribution.

Finally, the last column of the evaluation databases (i.e. the labels) must be removed and stored in a separate variable. Since the aim of the evaluation phase is to predict the label and then compare it to the actual one, the model must not have access to the true label during the evaluation phase.

3.2.4 Further elaboration of the database

In some cases, when the databases are highly imbalanced (i.e. the number of positive labels is much lower than the number of negative labels) it is necessary to furtherly elaborate the initial database.

Over-sampling and under-sampling techniques are applied in specific cases in order to improve class balance. In this thesis the SMOTE (Synthetic Minority Over-sampling Technique) technique has been utilised, since it is one of the most commonly used in various fields of application with good results. "SMOTE works by selecting examples that are close in the feature space, drawing a line between the examples in the feature space and drawing a new sample at a point along that line" (Brownlee, 2021).

In other words, a generic example from the minority class is selected and a certain number (usually 5) of the nearest neighbours are identified. Then, one neighbour is chosen and a synthetic example is created in a point between the two examples in the feature space through a convex combination of such examples, as shown in *Figure 27*.

This algorithm can create as many synthetic examples for the minority class as required to balance the dataset.

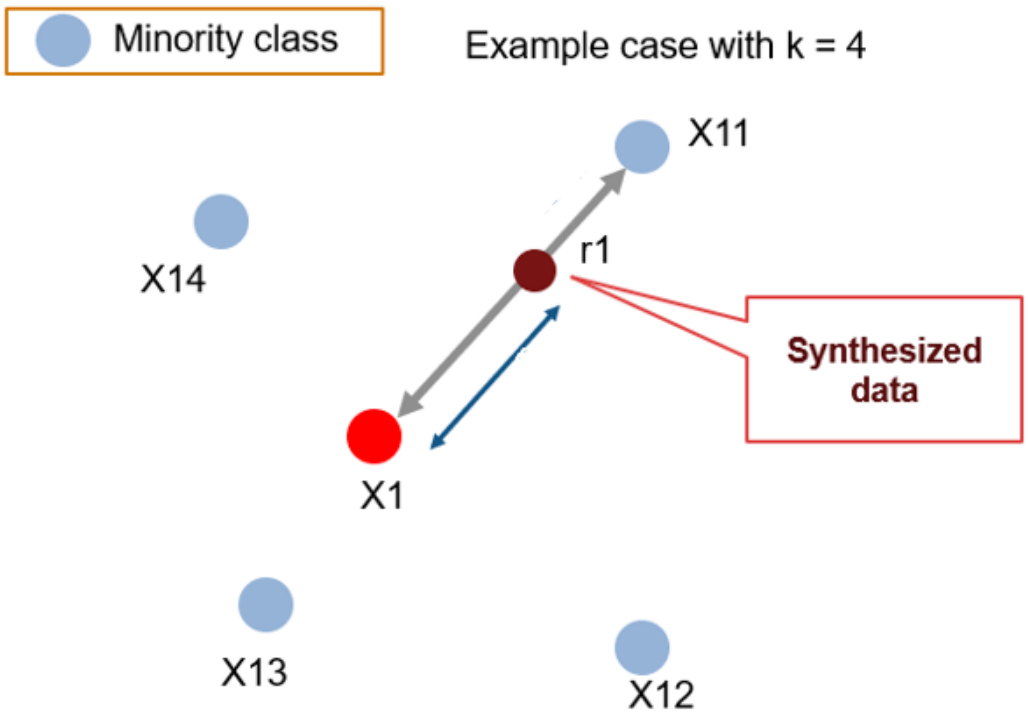


Figure 27. Graphic representation of SMOTE algorithm, adapted from GitHub (2020b).

Over-sampling or under-sampling techniques are usually applied to train datasets only, as the evaluation dataset is a collection of previously unseen real data whose labels' values must be predicted by the model to test its performances.

3.2.5 Model selection and features conversion

In this work, only supervised learning has been used. Three different models have been trained and evaluated (i.e. Linear, Deep, Wide&Deep) using the same databases and the same features. Still, the features need to be converted based on the selected model, as performed in the code in Appendix A.

Table 6 summarises the features (e.g. numerical, categorical and crossed) used to train each model on the different databases:

Table 6. Features summary.

	Linear	*Deep*	*Wide&Deep*
Numeric	Timestamp Ambient_pressure	Timestamp Ambient_pressure	Timestamp Ambient_pressure

	Release_rate RH% P01, …, P04 PT_01, …, PT_04 Wind_Direction Wind_speed TT, HC x, y, z	Release_rate RH% P01, …, P04 PT_01, …, PT_04 Wind_Direction Wind_speed TT, HC x, y, z	Release_rate RH% P01, …, P04 PT_01, …, PT_04 Wind_Direction Wind_speed TT, HC x, y, z
Categorical	Release_orientation Purge Sealing	Release_orientation Purge Sealing	Release_orientation Purge Sealing
Crossed	x × y × z TT (or HC) × x × y × z Release_rate × P01 × P02 × P03 × P04	-	x × y × z TT (or HC) × x × y × z Release_rate × P01 × P02 × P03 × P04

Where categorical features are non-numerical features (e.g. strings) that can influence the prediction's results, whereas crossed features combine different features in one column as described in Paragraph 2.4.3. Crossed features have been selected on the basis of physical correlation between the different variables: the release rate depends on tank's internal pressure, whereas the thermocouples and sensors are placed in a certain location identified by coordinates.

Finally, the Deep model and the deep part of the Wide&Deep model have three hidden layers. The first hidden layer has 1024 hidden units, the second 512 and the third 256.

3.2.6 Training phase

The models are fed with the training dataset. During this phase, the weights of the models are optimised to provide an accurate mapping from the input (the features) to the output (the labels). All the models have been trained for the same number of steps.

3.2.7 Evaluation phase

The trained models are now evaluated on their ability to predict the correct labels. The evaluation database is fed into the model. As a result, the model provides the predicted label's probability (as discussed in section 2.4.1). The threshold used to evaluate the predicted label is set by default to 0.5. By comparing the predicted labels with the real labels, the software calculates and displays the performance metrics described in Section 2.4.4. Although useful, these metrics are limited to the default threshold, that can be changed in order to optimise one of the performance metrics to obtain a more meaningful prediction for each specific case.

All of the previously described steps have been repeated and applied to a new version of the databases. In order to build a model that could be used in emergency situations of liquid hydrogen accidental release, the features that are not likely to be obtained quickly have been removed from the original databases. In a chemical plant ambient pressure, humidity, wind direction and speed are continuously monitored, moreover in case of accidental releases the release rate and orientation can be easily obtained. Some coordinates can be also fed into the model in order to predict the condensation or solidification of air oxygen or the formation of a flammable atmosphere in certain locations in the field. The only feature which is not likely to be easily measured in a real hydrogen spill scenario is the temperature or the hydrogen concentration in the field, since usually there are no thermocouples or sensors permanently placed in the field. Therefore, the column storing the values measured by such thermocouples or sensors has been removed from the databases and a new model has been built.

Chapter 4
Results

Several simulations have been performed, firstly using raw data and secondly pre-processing the data using the MinMaxScaler normalization method.

4.1 TensorFlow simulations

This section focuses on the results obtained from the three previously introduced models trained and evaluated on the complete databases (i.e. the databases containing the temperature or concentration values column).

4.1.1 Outdoor leakage studies

The main results of the simulations performed on the outdoor leakage studies data are here reported. All the performance metrics presented in this chapter have been obtained for a decision threshold default value of 0.5.

4.1.1.1 First database: label "liquid oxygen formation"

- *Results without data normalization*

Table 7. Performance metrics resulting from the evaluation of the three models trained over the raw outdoor leakage studies database for the label "liquid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.929	0.934	0.895	0.981
Deep	0.968	0.958	0.966	0.995
Wide&Deep	0.971	0.961	0.971	0.996

The confusion matrices resulting from the three different models are displayed in *Figure 28*.

		Predicted Label	
		0	1
Real Label	0	98449	4820
	1	8030	68437

(a)

		Predicted Label	
		0	1
Real Label	0	100027	3242
	1	2579	73888

(b)

		Predicted Label	
		0	1
Real Label	0	100289	2980
	1	2209	74258

(c)

Figure 28. Confusion matrices obtained for the label "liquid oxygen" without having performed data pre-processing on the first database by emplying the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 29*, *Figure 30* and *Figure 31*.

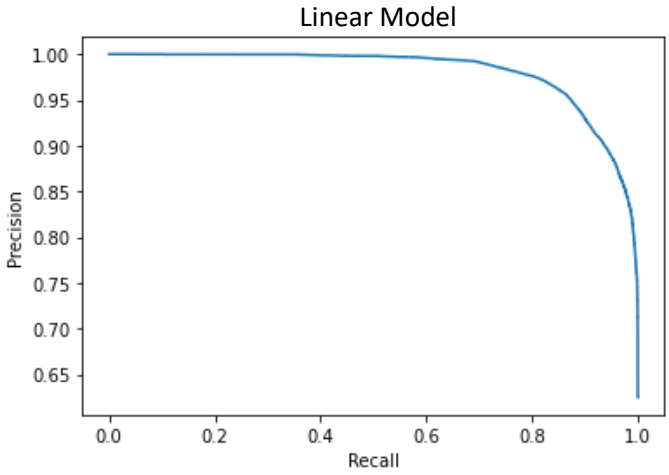


Figure 29. Precision-recall curve of the Linear Model (label: liquid oxygen; first database without data normalization)

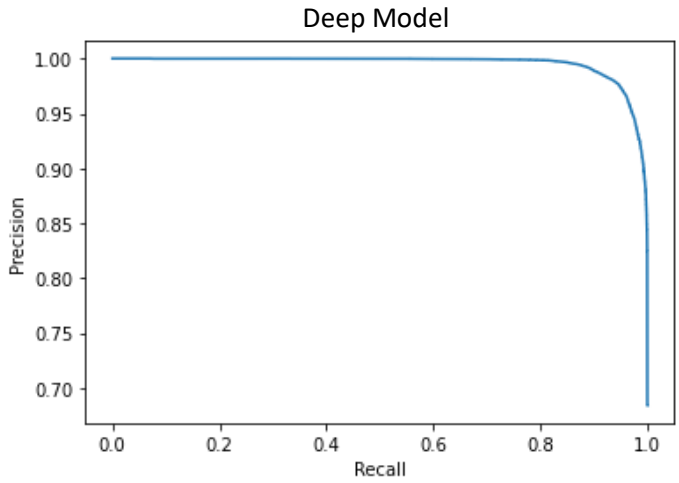


Figure 30. Precision-recall curve of the Deep Model (label: liquid oxygen; first database without data normalization)

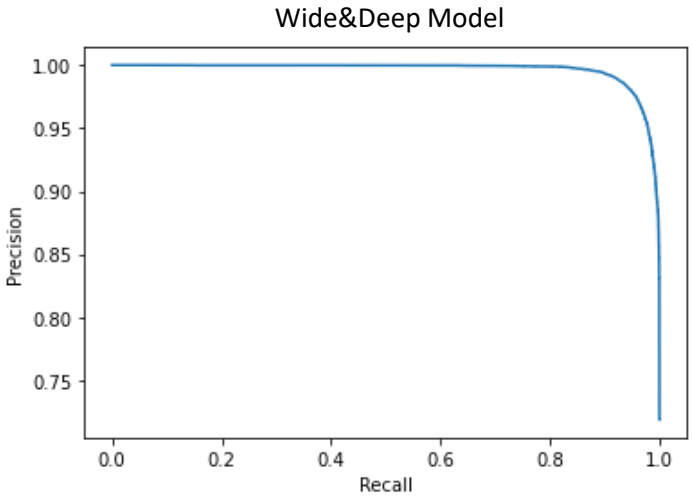


Figure 31. Precision-recall curve of the Wide&Deep Model (label; liquid oxygen; first database without data normalization)

By observing the metrics in *Table 7*, it is possible to conclude that the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics, thus they perform better than the Linear model. The confusion matrices displayed in *Figure 28* highlight that the linear model is weaker when it comes to predicting the label "1": the number of False Negatives (bottom left of the confusion matrices) is higher than the number of False Positives (top right of the confusion matrices). The opposite behaviour can be observed for the Deep and Wide&Deep models.

- *Results with data normalization*

Table 8. Performance metrics resulting from the evaluation of the three models trained over the normalized outdoor leakage studies database for the label "liquid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.902	0.848	0.936	0.949
Deep	0.909	0.872	0.922	0.962
Wide&Deep	0.909	0.871	0.923	0.962

The confusion matrices resulting from the three different models are displayed in *Figure 32*.

		Predicted Label	
		0	1
Real Label	0	90457	12812
	1	4861	71606

(a)

		Predicted Label	
		0	1
Real Label	0	92898	10371
	1	5948	70519

(b)

		Predicted Label	
		0	1
Real Label	0	92856	10413
	1	5906	70561

(c)

Figure 32. Confusion matrices obtained for the label "liquid oxygen" having performed data pre-processing on the first database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are reported in *Figure 33*, *Figure 34* and *Figure 35*.

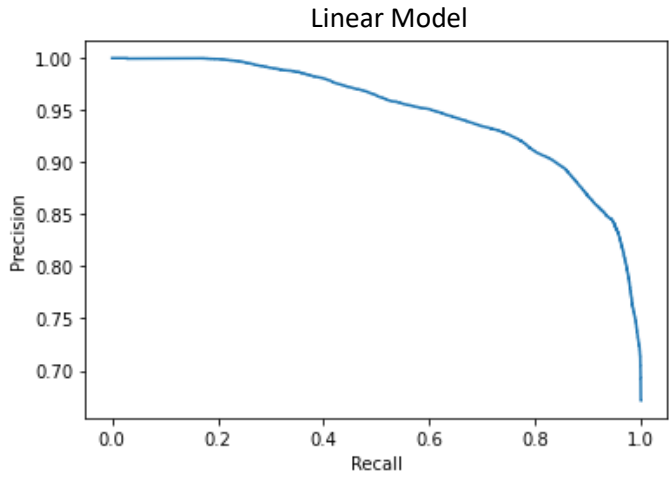


Figure 33. Precision-recall curve of the Linear Model (label: liquid oxygen; first database with data normalization)

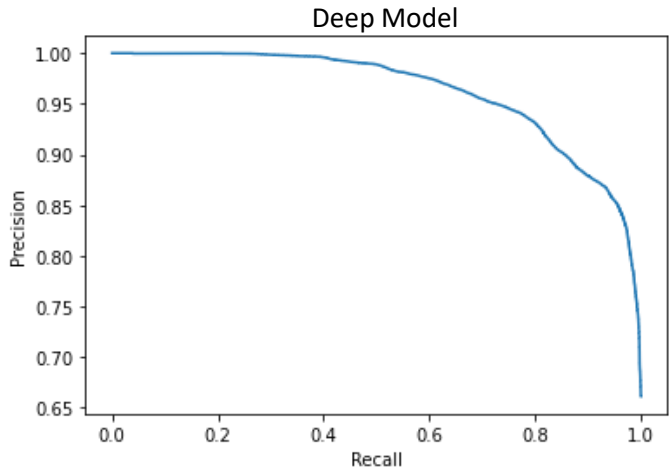


Figure 34. Precision-recall curve of the Deep Model (label: liquid oxygen; first database with data normalization)

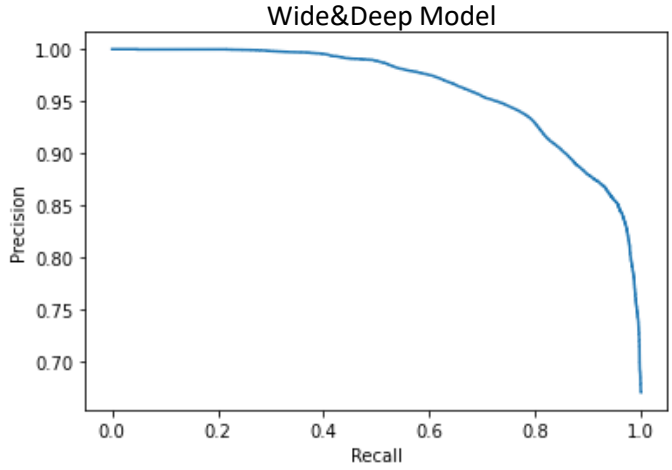


Figure 35. Precision-recall curve of the Wide&Deep Model (label: liquid oxygen; first database with data normalization)

Once again, by observing the metrics in *Table 8*, it is possible to conclude that the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics. The confusion matrices displayed in *Figure 32* highlight that the models are weaker when it comes to predicting the label "0"; the number of False Negatives (bottom left of the confusion matrices) is always lower than the number of False Positives (top right of the confusion matrices).

4.1.1.2 First database: label "solid oxygen formation"

- *Results without data normalization*

Table 9. Performance metrics resulting from the evaluation of the three models trained over the raw outdoor leakage studies database for the label "solid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.977	0.817	0.935	0.956
Deep	0.994	0.983	0.942	0.994
Wide&Deep	0.994	0.982	0.942	0.993

The confusion matrices resulting from the three different models are displayed in *Figure 36*.

		Predicted Label	
		0	1
Real Label	0	161478	3154
	1	982	14122

(a)

		Predicted Label	
		0	1
Real Label	0	164383	249
	1	875	14229

(b)

		Predicted Label	
		0	1
Real Label	0	164377	255
	1	872	14232

(c)

Figure 36. Confusion matrices obtained for the label "solid oxygen" without having performed data pre-processing on the first database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 37*, *Figure 38* and *Figure 39*.

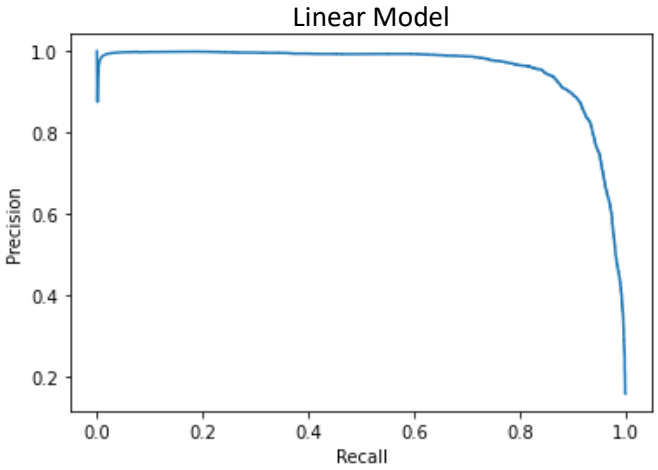


Figure 37. Precision-recall curve of the Linear Model (label: solid oxygen; first database without data normalization)

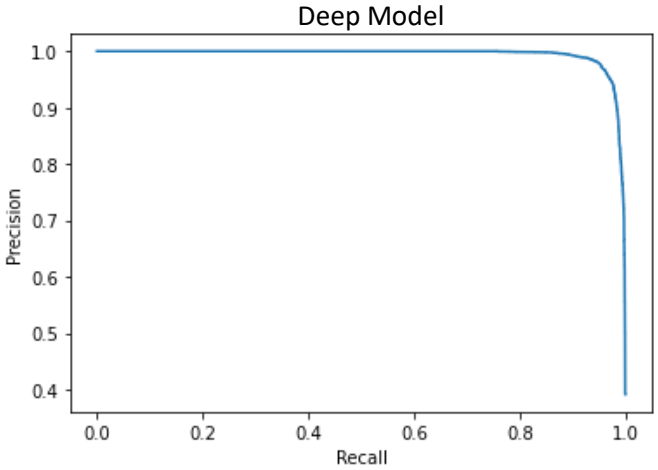


Figure 38. Precision-recall curve of the Deep Model (label: solid oxygen; first database without data normalization)

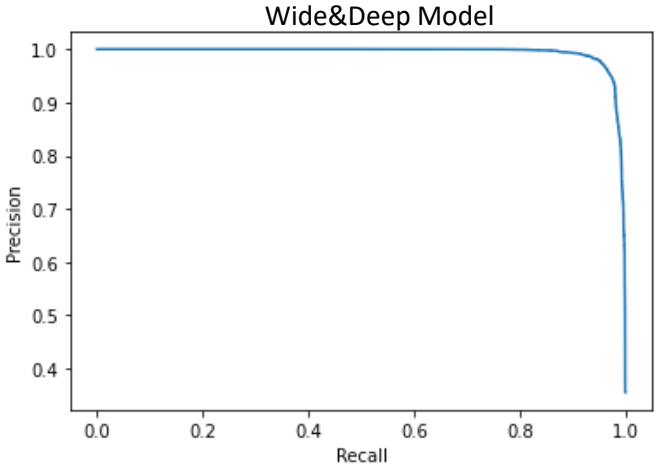


Figure 39. Precision-recall curve of the Wide&Deep Model (label: solid oxygen; first database without data normalization)

By observing the metrics in *Table 9*, it is possible to conclude that, similarly to the previous case, when predicting the label "solid oxygen formation" the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics. In opposition to what has emerged from the results obtained from the prediction of oxygen's condensation, the confusion matrices displayed in *Figure 36* highlight that the Deep and Wide&Deep models are weaker when it comes to predicting the label "1"; the number of False Negatives (bottom left of the confusion matrices) is higher than the number of False Positives (top right of the confusion matrices). The opposite behaviour can be observed for the Linear model.

- *Results with data normalization*

Table 10. Performance metrics resulting from the evaluation of the three models trained over the normalized outdoor leakage studies database for the label "solid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.957	0.830	0.613	0.807
Deep	0.967	0.873	0.706	0.878
Wide&Deep	0.966	0.868	0.700	0.874

The confusion matrices resulting from the three different models are displayed in *Figure 40*.

		Predicted Label	
		0	1
Real Label	0	162736	1896
	1	5845	9259

(a)

		Predicted Label	
		0	1
Real Label	0	163084	1548
	1	4435	10669

(b)

		Predicted Label	
		0	1
Real Label	0	163021	1611
	1	4528	10576

(c)

Figure 40. Confusion matrices obtained for the label "solid oxygen" having performed data pre-processing on the first database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 41*, *Figure 42* and *Figure 43*.

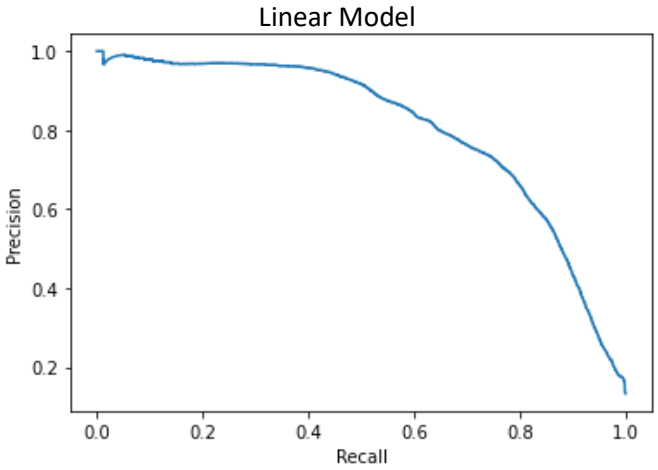


Figure 41. Precision-recall curve of the Linear Model (label: solid oxygen; first database with data normalization)

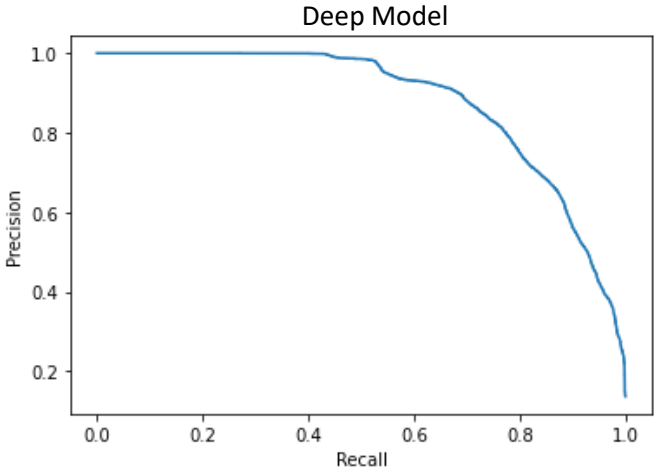


Figure 42. Precision-recall curve of the Deep Model (label: solid oxygen; first database with data normalization)

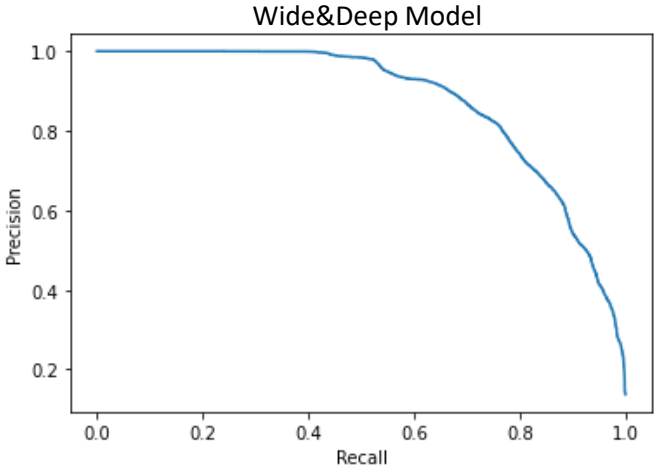


Figure 43. Precision-recall curve of the Wide&Deep Model (label: solid oxygen; first database with data normalization)

The performance metrics in *Table 10* show that the Linear model tends to perform worse than the Deep and Wide&Deep models. The confusion matrices displayed in *Figure 40* highlight that all the models are weaker when it comes to predicting the label "1"; the number of False Negatives (bottom left of the confusion matrices) is always higher than the number of False Positives (top right of the confusion matrices).

4.1.1.3 Second database: label "hydrogen concentration above the LEL"

- *Results without data normalization*

Table 11. Performance metrics resulting from the evaluation of the three models trained over the raw outdoor leakage studies database for the label "hydrogen concentration above the LFL"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.983	0.406	0.624	0.370
Deep	0.988	0.584	0.254	0.322
Wide&Deep	0.984	0.397	0.433	0.382

The confusion matrices resulting from the three different models are displayed in *Figure 44.*

		Predicted Label	
		0	1
Real Label	0	162109	1989
	1	822	1362

(a)

		Predicted Label	
		0	1
Real Label	0	163704	394
	1	1630	554

(b)

		Predicted Label	
		0	1
Real Label	0	162716	1382
	1	1220	964

(c)

Figure 44. Confusion matrices obtained for the label "hydrogen concentration above the LFL" without having performed data pre-processing on the second database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 45*, *Figure 46* and *Figure 47*.

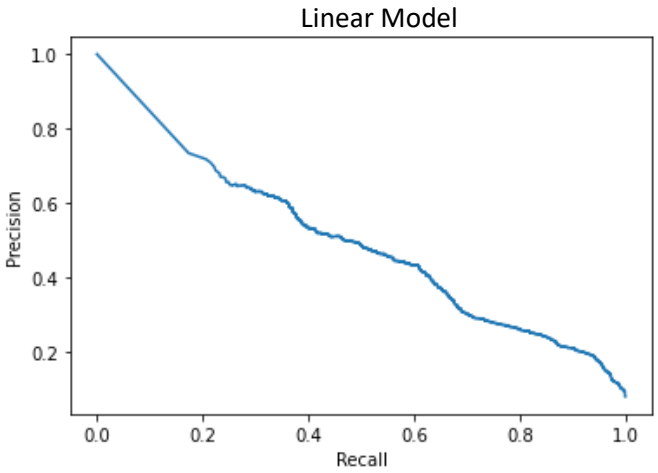


Figure 45. Precision-recall curve of the Linear Model (label: hydrogen concentration above the LFL; second database without data normalization)

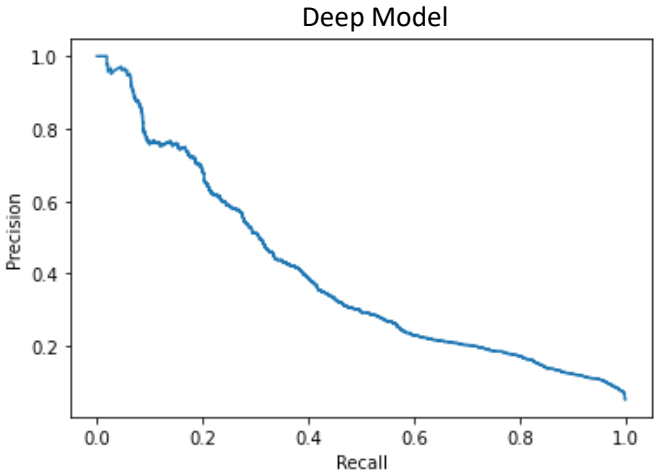


Figure 46. Precision-recall curve of the Deep Model (label: "hydrogen concentration above the LFL; second database without data normalization)

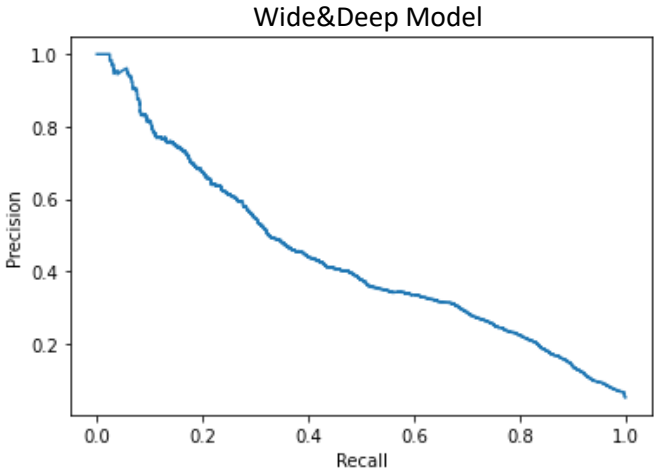


Figure 47. Precision-recall curve of the Wide&Deep Model (label: hydrogen concentration above the LFL; second database without data normalization)

As it can be observed by the precision-recall curves and by the metrics displayed in *Table 11*, all of the models are characterised by high accuracy and low precision and recall, with an area under the curve relatively small. The Wide&Deep model seems to give the best performance, as it produces the larger area under the precision-recall curve. This means that there is an allowance for improvement performing a recall optimisation.

- *Results with data normalization*

Table 12. Performance metrics resulting from the evaluation of the three models trained over the normalized outdoor leakage studies database for the label "hydrogen concentration above the LFL"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.988	0.649	0.184	0.366
Deep	0.988	0.646	0.203	0.400
Wide&Deep	0.988	0.642	0.205	0.411

The confusion matrices resulting from the three different models are displayed in *Figure 48*.

		Predicted Label	
		0	1
Real Label	0	163881	217
	1	1783	401

(a)

		Predicted Label	
		0	1
Real Label	0	163855	243
	1	1741	443

(b)

		Predicted Label	
		0	1
Real Label	0	163848	250
	1	1736	448

(c)

Figure 48. Confusion matrices obtained for the label "hydrogen concentration above the LFL" having performed data pre-processing on the second database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are displayed in *Figure 49*, *Figure 50* and *Figure 51*.

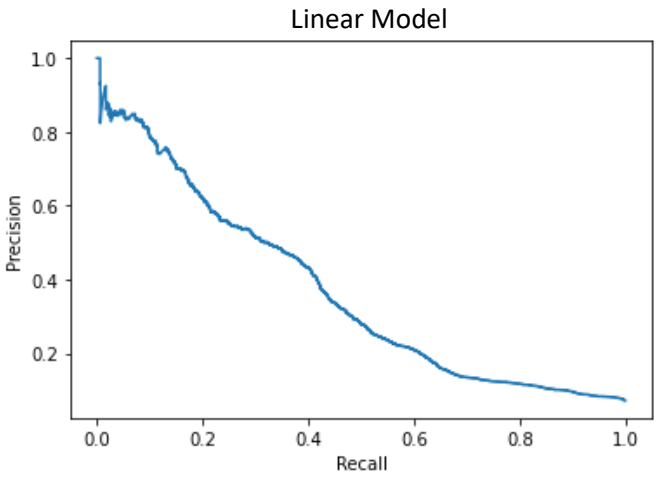


Figure 49. Precision-recall curve of the Linear Model (label: hydrogen concentration above the LFL; second database with data normalization)

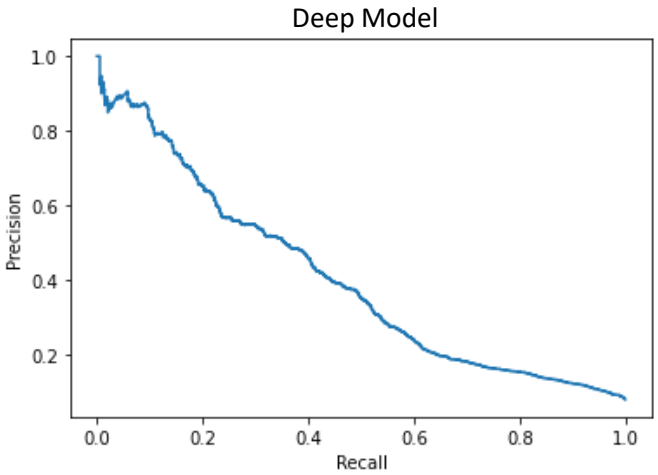


Figure 50. Precision-recall curve of the Deep Model (label: hydrogen concentration above the LFL; second database with data normalization)

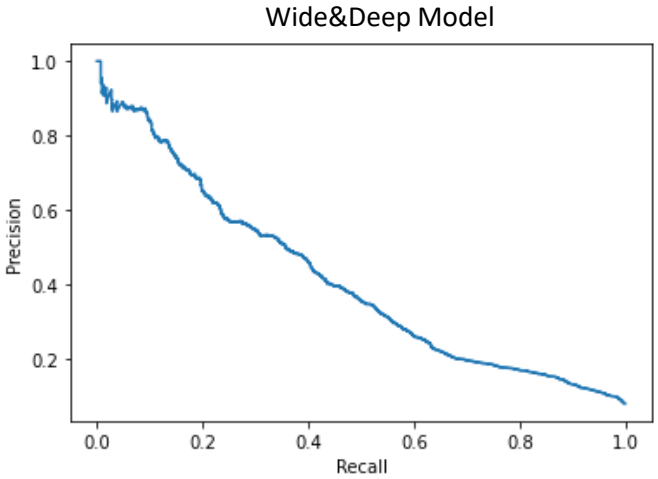


Figure 51. Precision-recall curve of the Wide&Deep Model (label: hydrogen concentration above the LFL; second database with data normalization)

As it can be observed by the precision-recall curves and by the metrics displayed in *Table 12*, all of the models are characterised by high accuracy and low precision and recall, with an area under the curve relatively small. Accordingly to what has been seen for the raw database, even after pre-processing the input data the Wide&Deep model seems to give the best performance, as it produces the larger area under the precision-recall curve.

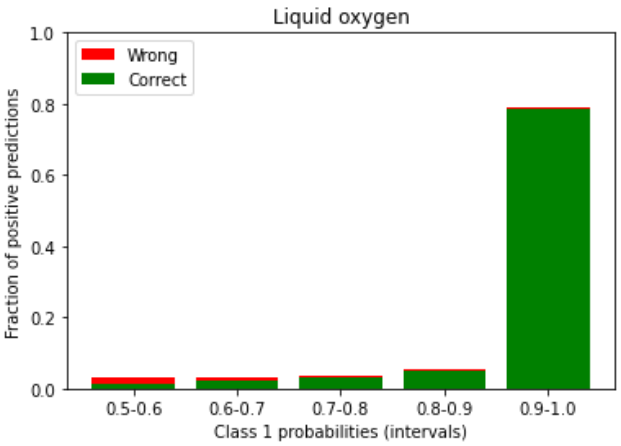
4.1.1.4 Insights on the Wide&Deep Model's results

Being one of the best performing models given the high values of recall and area under the precision-recall curve, the Wide&Deep one has been further investigated in its predictive capability. A chart displaying the true positive rate (green bar) and the false positive rate (red bar) for each probability category has been produced for all the analysed labels.

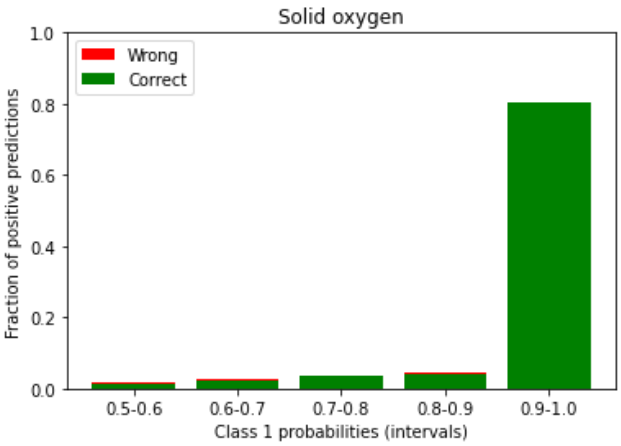
- *Results without data normalization*

As it can be stated by comparing the bar charts in *Figure 52* the model predicts the condensation or solidification of oxygen on the ground with a high confidence, being most of the positive predictions made within the highest probability range. A higher number of wrong predictions are made when it comes to evaluating whether or not the hydrogen concentration in air will reach the LFL within 200 s.

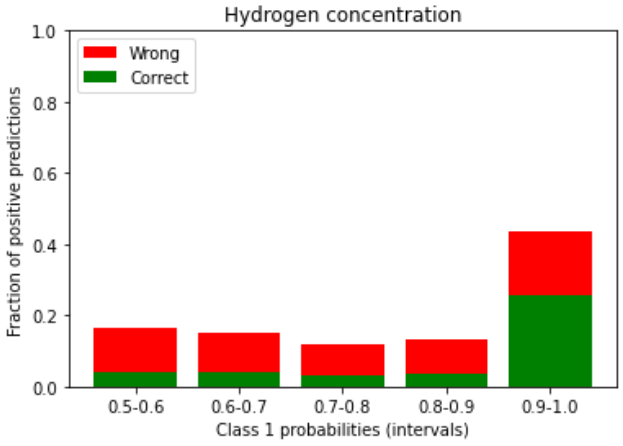
Figure 52 (c), as well as the metrics and curves displayed in Section 4.1.1.3, shows that the model cannot make accurate predictions about values of hydrogen concentration in air. The obtained metrics are characterised by high accuracy and low precision and recall.



(a)



(b)



(c)

Figure 52. Fraction of positive predictions for each probability class (Wide&Deep model without data normalization) for the labels (a) liquid oxygen formation, (b) solid oxygen formation and (c) hydrogen concentration above the LFL

To improve the results of the hydrogen gas concentration's prediction a recall optimisation has been performed. The recall optimisation is carried out by finding the best threshold value that maximises the F-measure (or F-score), as described in Section 2.4.4. β has been taken equal to 1.5.

The following results are obtained:

- best threshold = 0.066;
- F-score = 0.476;
- Precision = 0.293;
- Recall = 0.657.

Figure 53 and *Figure 54* show the trend of the F-score and the Precision and Recall curves as functions of the threshold respectively, whereas *Figure 55* shows the fraction of positive predictions for each probability class, having considered the best threshold value instead of the default value of 0.5 to evaluate the positive predictions.

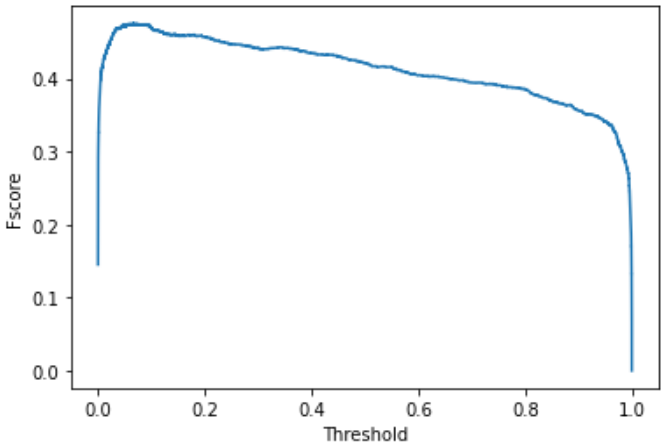


Figure 53. F-score over threshold curve.

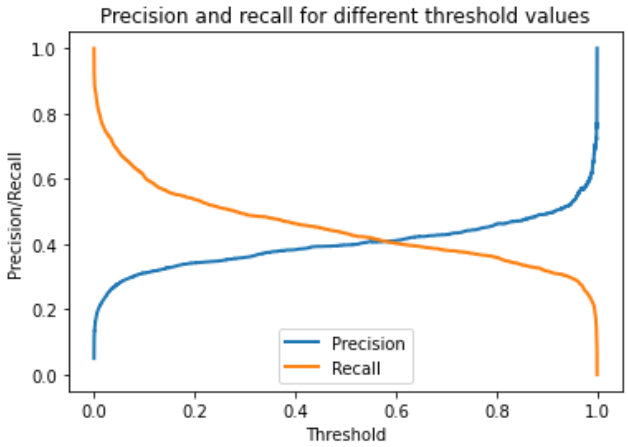


Figure 54. Precision/Recall over threshold.

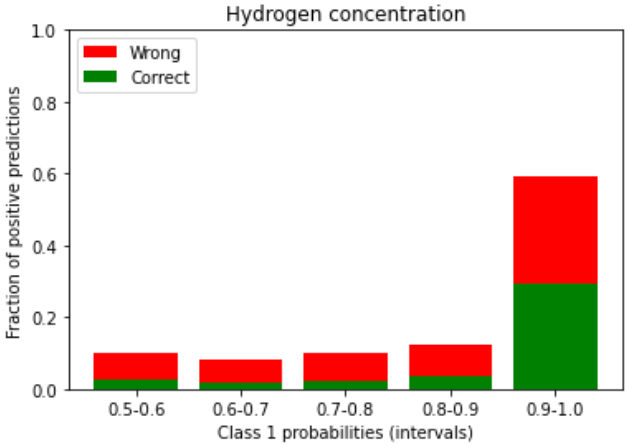
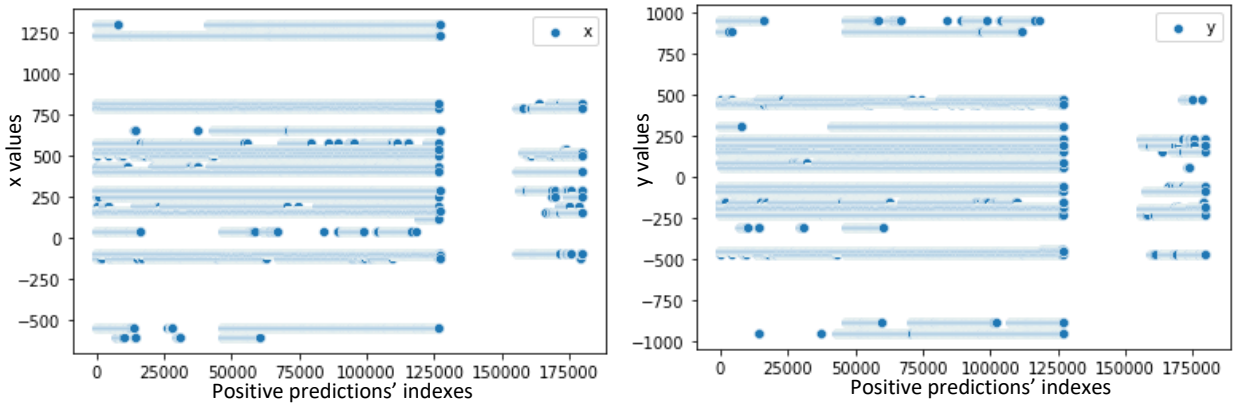


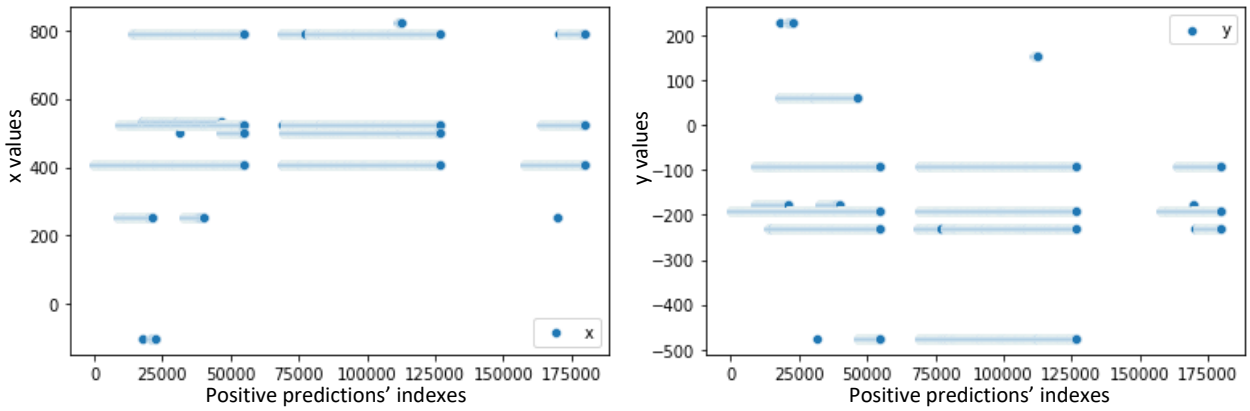
Figure 55. Fraction of positive predictions for each probability class obtained considering the best threshold, label: hydrogen concentration above the LFL (Wide&Deep model without data normalization).

The charts below represent the spatial extension of the liquid (see *Figure 56*) or solid (see *Figure 57*) oxygen deposit on the pad. These charts have been obtained by plotting the values of the x and y coordinates from the release point corresponding to a positive prediction.



(a) (b)

Figure 56. Spatial coordinates of the positive "liquid oxygen" predictions of the Wide&Deep model on the (a) x and (b) y axis.



(a) (b)

Figure 57. Spatial coordinates of the positive "solid oxygen" predictions of the Wide&Deep model on the (a) x and (b) y axis

- *Results with data normalization*

The Wide&Deep model trained on normalized data shows the same behaviour as the one seen for the model trained on raw data. *Figure 58* (a) and *Figure 58* (b) highlight that the model is quite good at predicting the condensation or solidification of air components on the ground, given the high true positive rate for the highest probability range. On the other hand, it seems to be rather weak in predicting whether or not the concentration of hydrogen will reach the LFL within 200 s.

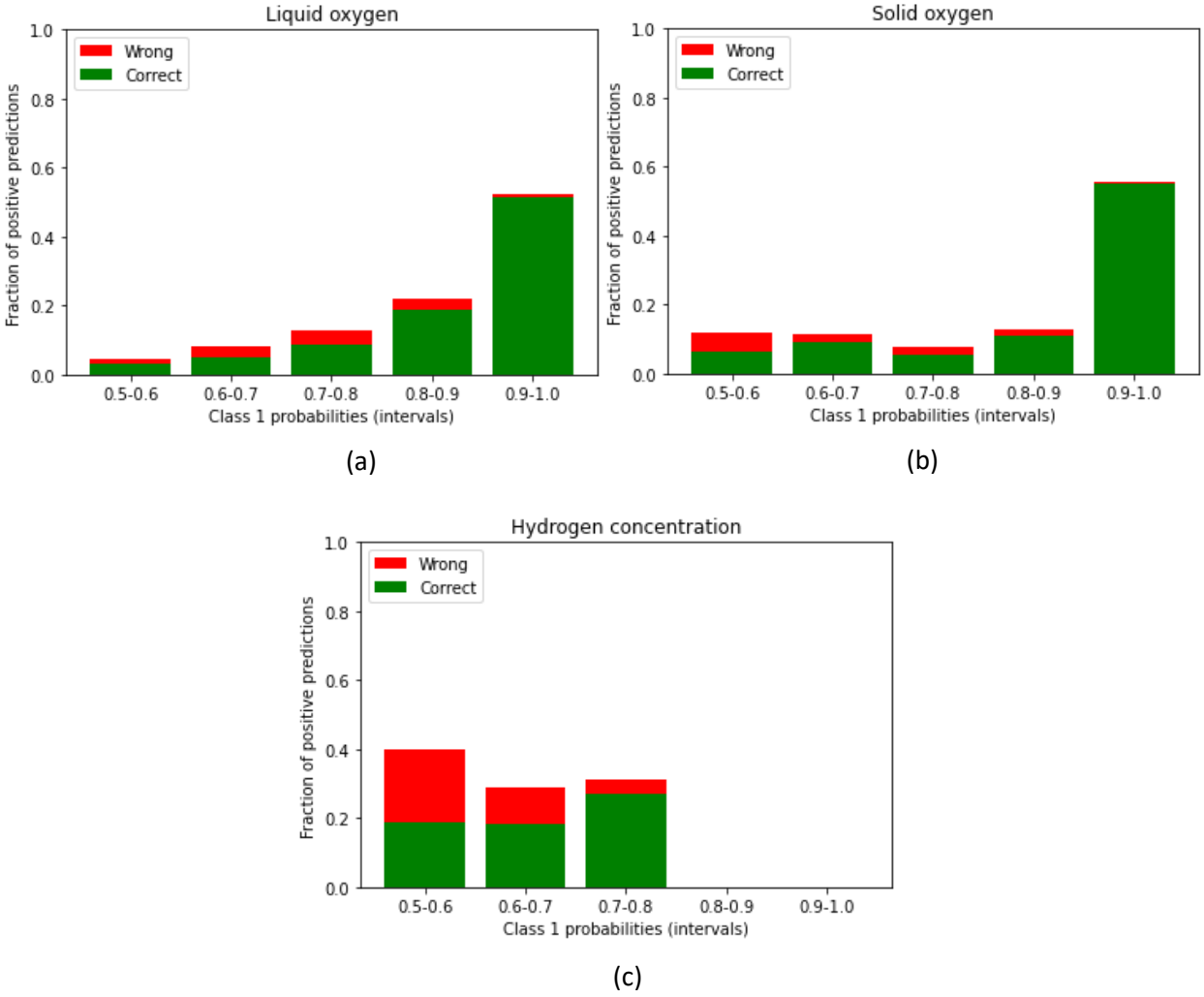


Figure 58. Fraction of positive predictions for each probability class (Wide&Deep model with data normalization) for the labels (a) liquid oxygen formation, (b) solid oxygen formation and (c) hydrogen concentration above the LFL

In order to improve the model performance when it comes to predicting whether or not the concentration of hydrogen in air will be higher than the lower flammability limit the SMOTE (Synthetic Minority Over-sampling Technique) technique has been utilised, since the initial database is extremely imbalanced (only 1.25% of positive labels over the entire database) .

The SMOTE algorithm has been applied to the train dataset only, as the evaluation dataset is a collection of previously unseen real data whose labels' values must be predicted by the model to test its performances. Since the database contains both numeric and categorical features, a specific SMOTE technique has been implemented: SMOTE-NC; Unlike SMOTE, SMOTE-NC is used for datasets containing numerical and categorical features, but it cannot work with only categorical features (Imbalanced Learn, 2022).

The results obtained by the Wide&Deep model trained on the over-sampled dataset are here presented:

Table 13. Results of the Wide&Deep model trained on the oversampled train dataset for the label "hydrogen concentration above the LFL".

Accuracy	*Precision*	*Recall*	*AUC_pr*
0.421	0.022	1.0	0.040

		Predicted Label	
		0	1
Real Label	0	67884	96214
	1	0	2184

Figure 59. Confusion matrix of the Wide&Deep model trained on the oversampled train dataset for the label "hydrogen concentration above the LFL".

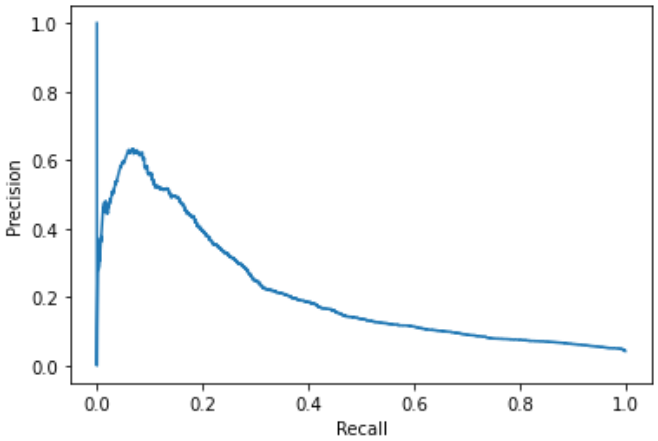


Figure 60. Precision-recall curve of the Wide&Deep model trained on the database where SMOTE has been applied.

Figure 59 shows that the model predicts a high number of False Positives and a null value of False Negatives, which means it is weaker when it comes to predicting the label "0".

4.1.2 Indoor leakage studies

The main results of the simulations performed on the indoor leakage studies data are here reported. All the performance metrics presented in this chapter have been obtained for a decision threshold default value of 0.5.

4.1.2.1 Third database: label "liquid oxygen formation"

- *Results without data normalization*

Table 14. Performance metrics resulting from the evaluation of the three models trained over the raw indoor leakage studies database for the label "liquid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.961	0.941	0.987	0.994
Deep	0.988	0.987	0.990	0.998
Wide&Deep	0.988	0.989	0.988	0.998

The confusion matrices resulting from the three different models are displayed in *Figure 61*.

		Predicted Label	
		0	1
Real Label	0	51295	3797
	1	786	60684

(a)

		Predicted Label	
		0	1
Real Label	0	54305	787
	1	626	60844

(b)

		Predicted Label	
		0	1
Real Label	0	54411	681
	1	723	60747

(c)

Figure 61. Confusion matrices obtained for the label "liquid oxygen" without having performed data pre-processing on the third database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are displayed in *Figure 62*, *Figure 63* and *Figure 64*.

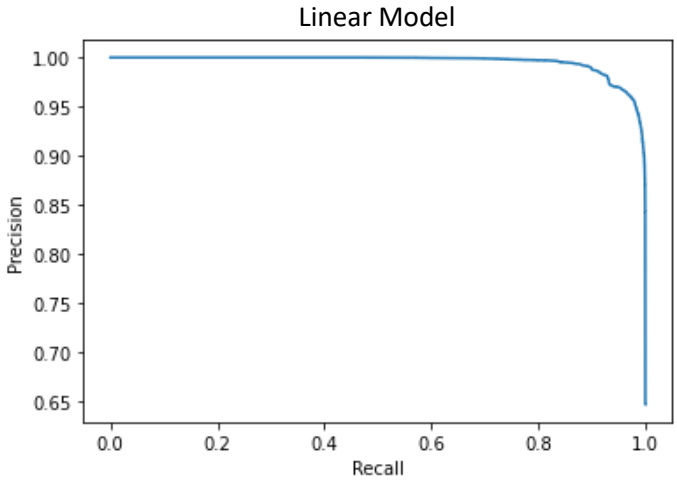


Figure 62. Precision-recall curve of the Linear Model (label: liquid oxygen; third database without data normalization)

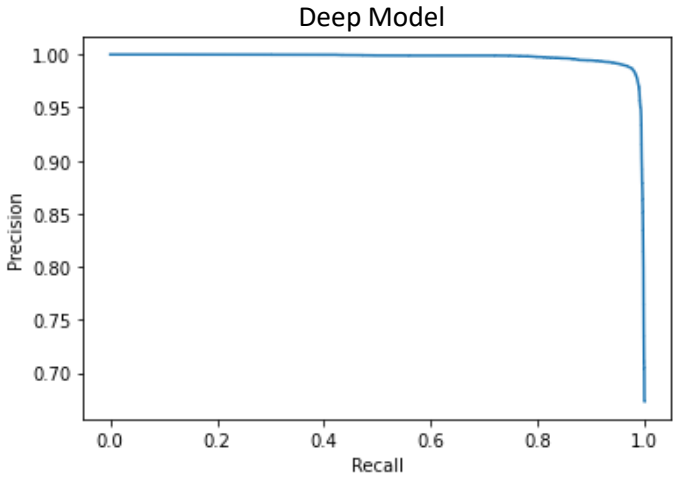


Figure 63. Precision-recall curve of the Deep Model (label: liquid oxygen; third database without data normalization)

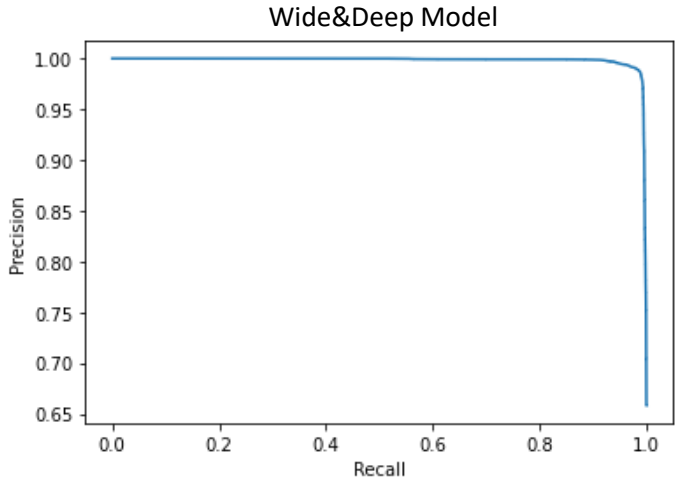


Figure 64. Precision-recall curve of the Wide&Deep Model (label: liquid oxygen; third database without data normalization)

As previously seen for the outdoor studies, even when training the model over the indoor tests' results the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics. The confusion matrices displayed in *Figure 61* highlight that the Linear model is significantly weaker when it comes to predicting the label "0", being the number of False Negatives (bottom left of the confusion matrices) lower than the number of False Positives (top right of the confusion matrices). The Wide&Deep model gives the opposite behaviour.

- *Results with data normalization*

Table 15. Performance metrics resulting from the evaluation of the three models trained over the normalized indoor leakage studies database for the label "liquid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.939	0.926	0.962	0.989
Deep	0.987	0.984	0.991	0.998
Wide&Deep	0.986	0.984	0.990	0.998

The confusion matrices resulting from the three different models are displayed in *Figure 65*.

		Predicted Label	
		0	1
Real Label	0	50350	4742
	1	2365	59105

(a)

		Predicted Label	
		0	1
Real Label	0	54130	962
	1	544	60926

(b)

		Predicted Label	
		0	1
Real Label	0	54084	1008
	1	585	60885

(c)

Figure 65. Confusion matrices obtained for the label "liquid oxygen" having performed data pre-processing on the third database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 66*, *Figure 67* and *Figure 68*.

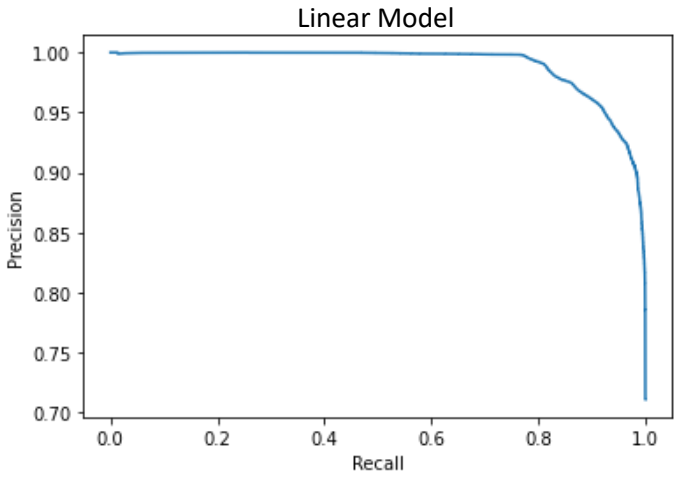


Figure 66. Precision-recall curve of the Linear Model (label: liquid oxygen; third database with data normalization)

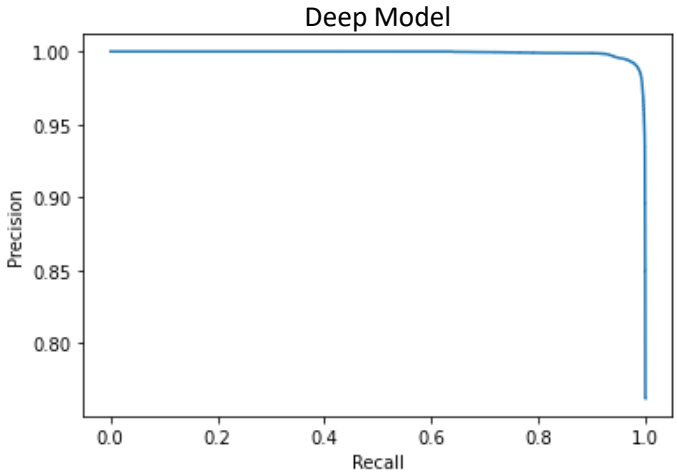


Figure 67. Precision-recall curve of the Deep Model (label: liquid oxygen; third database with data normalization)

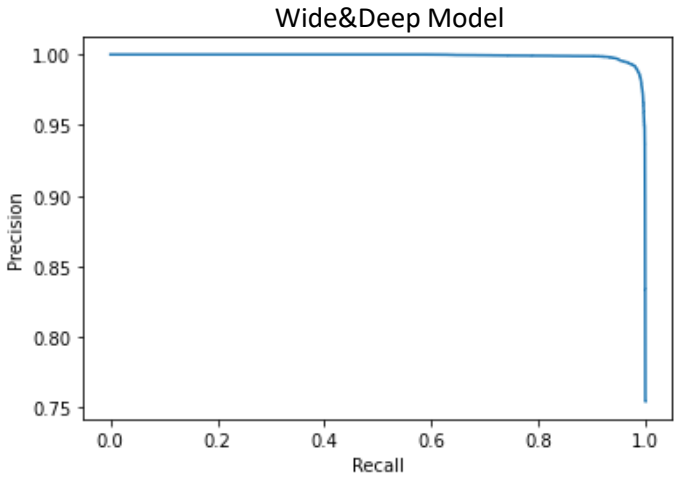


Figure 68. Precision-recall curve of the Wide&Deep Model (label: liquid oxygen; third database with data normalization)

When training the models over the normalized indoor tests' results the Linear model is the worst performing one. Once again, the confusion matrices displayed in *Figure 65* highlight that all the models are weaker when it comes to predicting the label "0"; the number of False Negatives (bottom left of the confusion matrices) is always lower than the number of False Positives (top right of the confusion matrices).

4.1.2.2 Third database: label “solid oxygen formation”

- *Results without data normalization*

Table 16. Performance metrics resulting from the evaluation of the three models trained over the raw indoor leakage studies database for the label “solid oxygen formation”

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.971	0.954	0.925	0.980
Deep	0.993	0.982	0.987	0.998
Wide&Deep	0.991	0.975	0.987	0.998

The confusion matrices resulting from the three different models are displayed in *Figure 69*.

		Predicted Label	
		0	1
Real Label	0	87305	1255
	1	2106	25896

(a)

		Predicted Label	
		0	1
Real Label	0	88063	497
	1	365	27637

(b)

		Predicted Label	
		0	1
Real Label	0	87848	712
	1	362	27640

(c)

Figure 69. Confusion matrices obtained for the label “solid oxygen” without having performed data pre-processing on the third database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained by varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 70*, *Figure 71* and *Figure 72*.

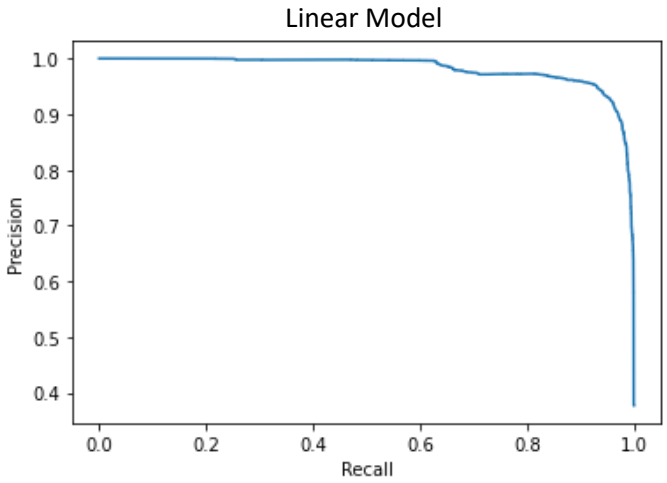


Figure 70. Precision-recall curve of the Linear Model (label: solid oxygen; third database without data normalization)

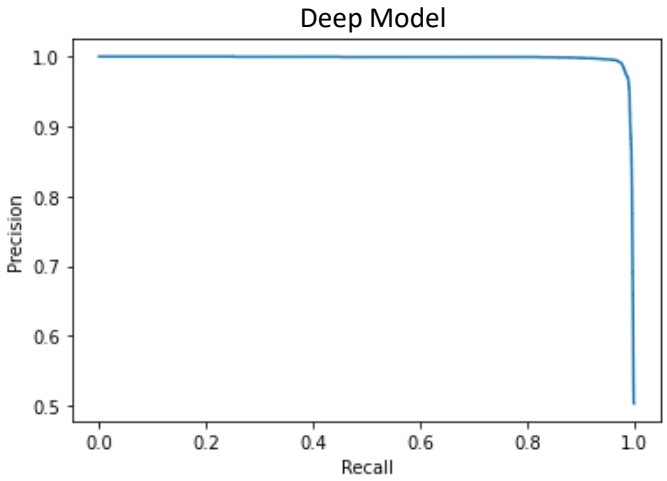


Figure 71. Precision-recall curve of the Deep Model (label: solid oxygen; third database without data normalization)

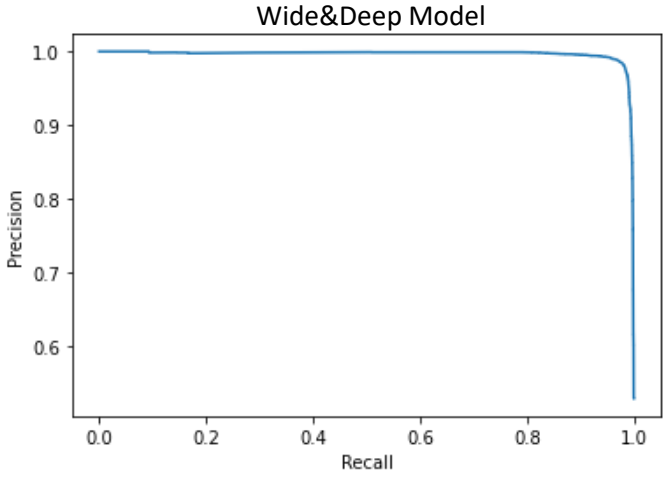


Figure 72. Precision-recall curve of the Wide&Deep Model (label: solid oxygen; third database without data normalization)

By comparing the figures in *Table 16* it is evident that the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics. The confusion matrices displayed in *Figure 69* highlight that the models with a deep approach are weaker when it comes to predicting the label "0"; the number of False Negatives (bottom left of the confusion matrices) is lower than the number of False Positives (top right of the confusion matrices).

- *Results with data normalization*

Table 17. Performance metrics resulting from the evaluation of the three models trained over the normalized indoor leakage studies database for the label "solid oxygen formation"

	Accuracy	*Precision*	*Recall*	*AUC_pr*
Linear	0.941	0.853	0.912	0.931
Deep	0.983	0.960	0.968	0.996
Wide&Deep	0.982	0.960	0.967	0.996

The confusion matrices resulting from the three different models are displayed in *Figure 73*.

		Predicted Label	
		0	1
Real Label	0	84156	4404
	1	2477	25525

(a)

		Predicted Label	
		0	1
Real Label	0	87433	1127
	1	907	27095

(b)

		Predicted Label	
		0	1
Real Label	0	87443	1117
	1	934	27068

(c)

Figure 73. Confusion matrices obtained for the label "solid oxygen" having performed data pre-processing on the third database by employing the (a) Linear Model, (b) Deep Model and (c) Wide&Deep Model.

The precision-recall curves obtained varying the threshold between 0 and 1 associated to the different models are depicted in *Figure 74*, *Figure 75* and *Figure 76*.

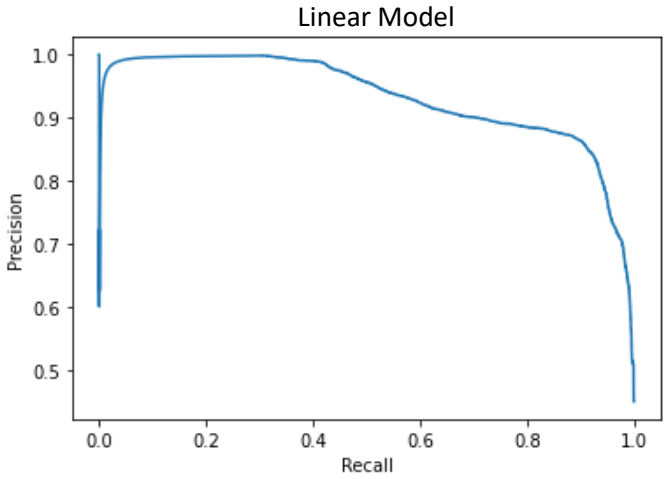


Figure 74. Precision-recall curve of the Linear Model (label: solid oxygen; third database with data normalization)

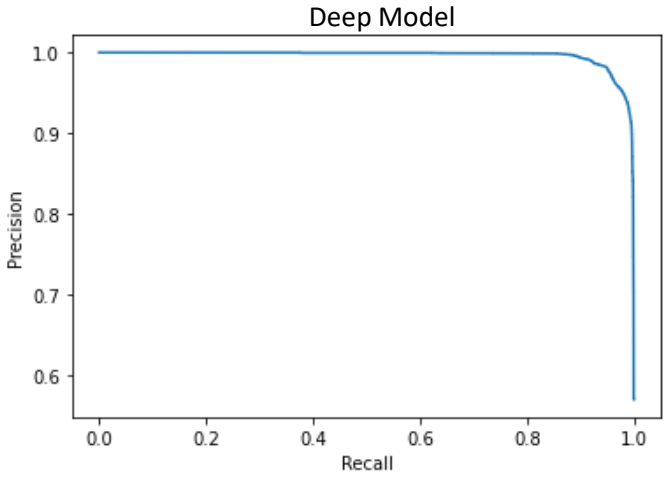


Figure 75. Precision-recall curve of the Deep Model (label: solid oxygen; third database with data normalization)

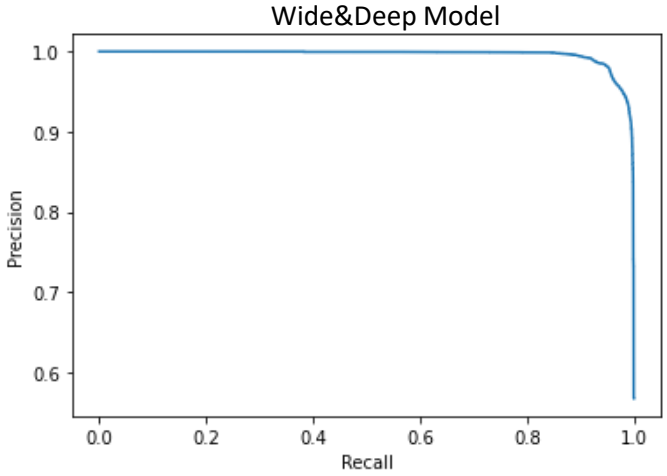


Figure 76. Precision-recall curve of the Wide&Deep Model (label: solid oxygen; third database with data normalization)

By comparing the figures in *Table 17* it is possible to conclude that the Linear model produces smaller metrics, whereas the Deep model and the Wide&Deep model produce similar larger metrics. The confusion matrices displayed in *Figure 73* highlight that all the models are weaker when it comes to predicting the label "0"; the number of False Negatives (bottom left of the confusion matrices) is always lower than the number of False Positives (top right of the confusion matrices).

4.1.2.3 Insights on the Wide&Deep model's results

Being one of the best performing models given the high values of recall and area under the precision-recall curve, the Wide&Deep one has been further investigated in its predictive capability: for all the analysed labels a chart displaying the true positive rate (green bar) and the false positive rate (red bar) for each probability category has been produced.

- *Results without data normalization*

Figure 77 shows the fraction of positive predictions for each probability range for the different labels. The model seems to be good at predicting the condensation or solidification of air components given that most of the positive predictions are made with the highest probability.

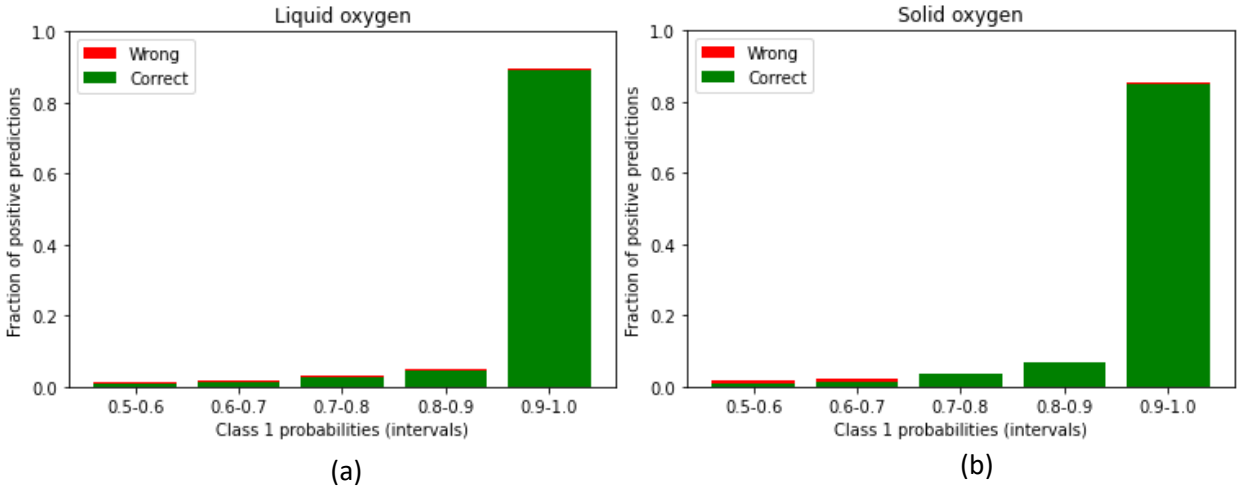
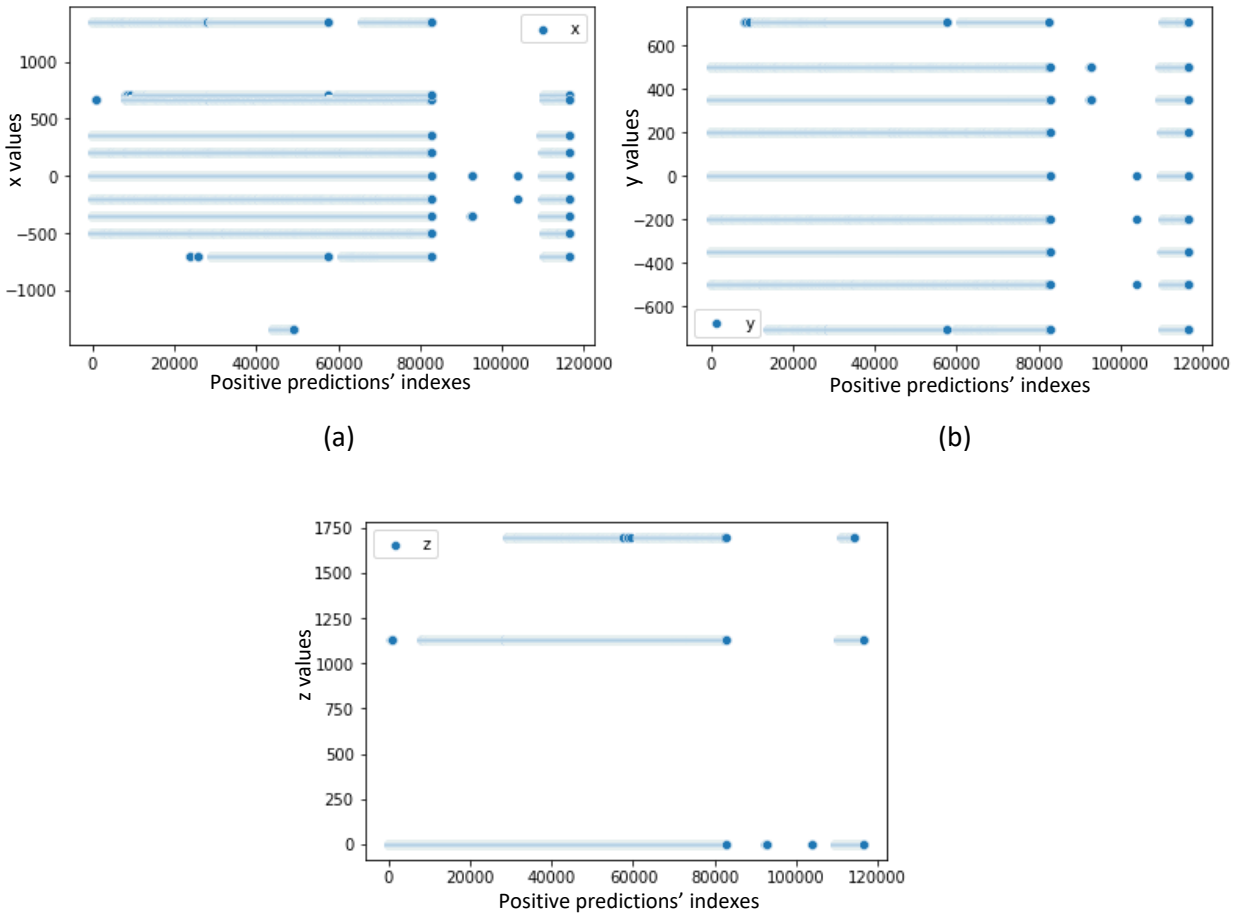


Figure 77. Fraction of positive predictions for each probability class (Wide&Deep model without data normalization) for the labels (a) liquid oxygen formation and (b) solid oxygen formation.

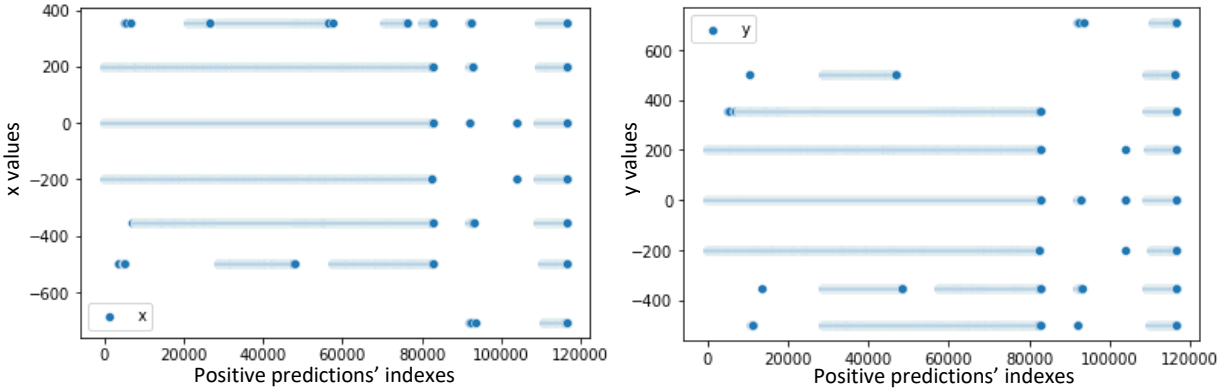
The charts below represent the spatial extension of the liquid (see *Figure 78*) or solid (see *Figure 79*) oxygen deposit on the pad. These charts have been obtained by plotting the values of the x and y coordinates corresponding to a positive prediction.



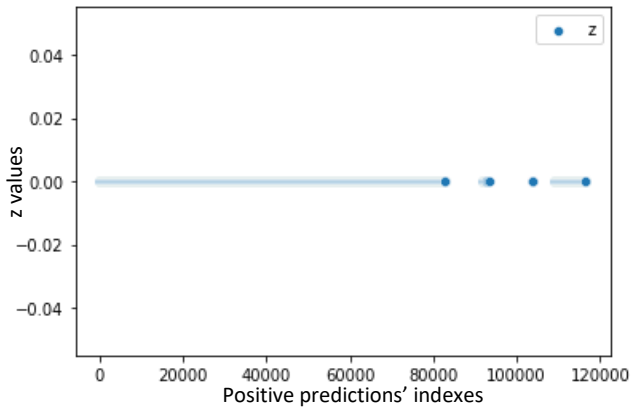
(a) (b)

(c)

Figure 78. Spatial coordinates of the positive "liquid oxygen" predictions of the Wide&Deep model on the (a) x, (b) y and (c) z axis.



(a) (b)

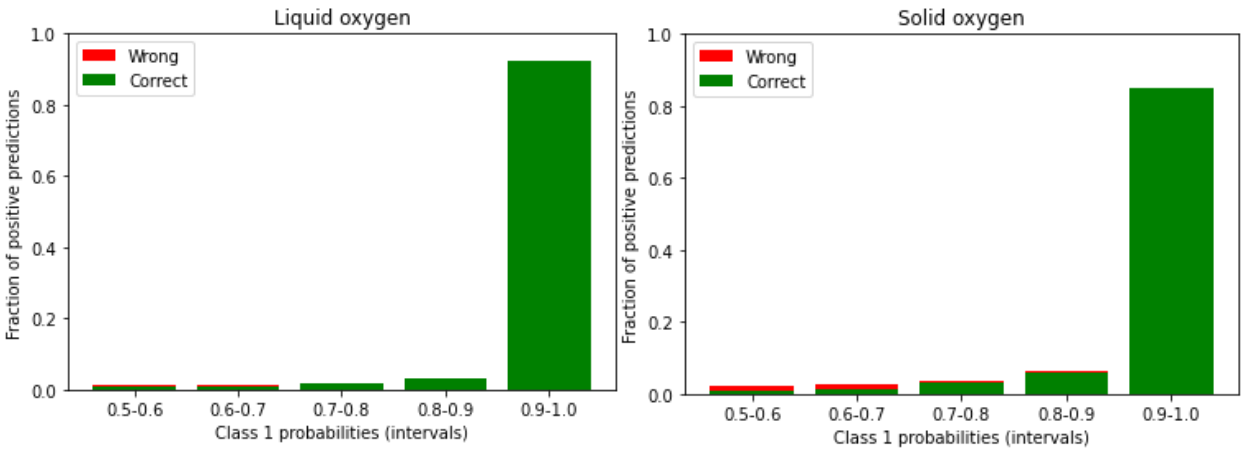


(c)

Figure 79. Spatial coordinates of the positive "solid oxygen" predictions of the Wide&Deep model on the (a) x, (b) y and (c) z axis.

- *Results with data normalization*

The Wide&Deep model trained on normalized data shows the same behaviour as the one seen for the model trained on raw data. *Figure 80* highlights that the model is extremely good at predicting the condensation or solidification of air components on the ground, given the high true positive rate for the highest probability range.



(a) (b)

Figure 80. Fraction of positive predictions for each probability class (Wide&Deep model with data normalization) for the labels (a) liquid oxygen formation and (b) solid oxygen formation

4.1.3 TensorFlow simulations: emergency situations

This section describes some of the results obtained by building the models on the databases where the temperature (or concentration) values column has been removed, in order to simulate the data that can easily be collected in a real emergency situation, as mentioned in Section 3.2. To provide some examples only the results coming from the application of the Wide&Deep model on raw data (i.e. no data pre-processing has been performed) considering the labels "liquid oxygen formation" and "solid oxygen formation" are presented in *Table 18.*

Table 18. Emergency situations models' results

	Outdoor tests	*Indoor tests*
Label: liquid oxygen formation	Accuracy = 0.955 Precision = 0.963 Recall = 0.930 AUC_pr = 0.990 TP = 71092 FP = 2723 TN = 100546 FN = 5375	Accuracy = 0.988 Precision = 0.988 Recall = 0.989 AUC_pr = 0.998 TP = 60779 FP = 759 TN = 54333 FN = 691
Label: solid oxygen formation	Accuracy = 0.990 Precision = 0.926 Recall = 0.954 AUC_pr = 0.981 TP = 14416 FP = 1149	Accuracy = 0.989 Precision = 0.966 Recall = 0.988 AUC_pr = 0.997 TP = 27665 FP = 968

	TN = 163483 FN = 688	TN = 87592 FN = 337

The precision-recall curves obtained for both the labels in the outdoor scenario show a trend similar to the one displayed in *Figure 81*.

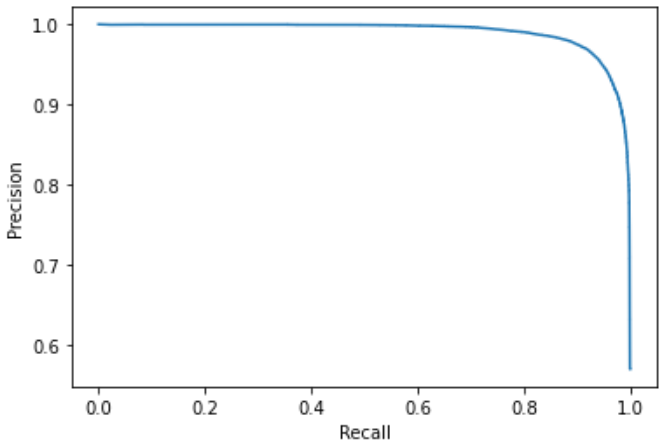


Figure 81. Precision-recall curve of the Wide&Deep model (outdoor hydrogen leakage scenario, emergency situations)

Whereas, when training the model on the indoor tests' results for predicting both the labels condensation and solidification of air oxygen, the following precision-recall curve is provided:

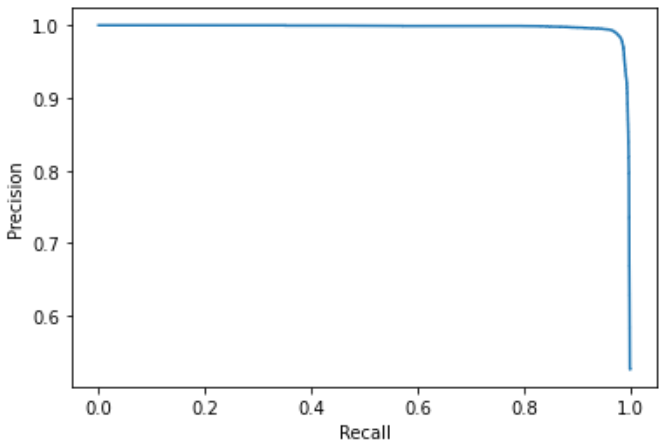
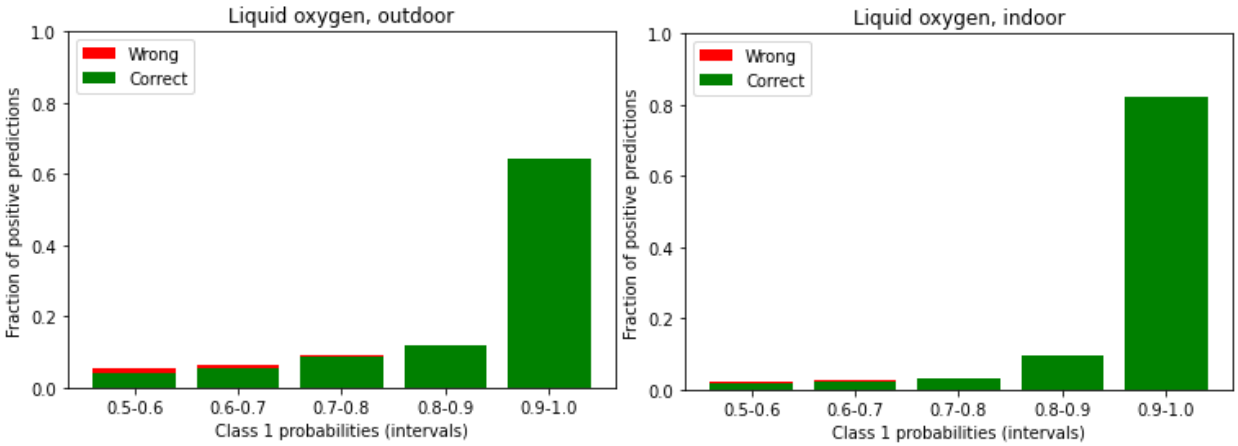


Figure 82. Precision-recall curve of theWide&Deep model (indoor hydrogen leakage scenario, emergency situations)

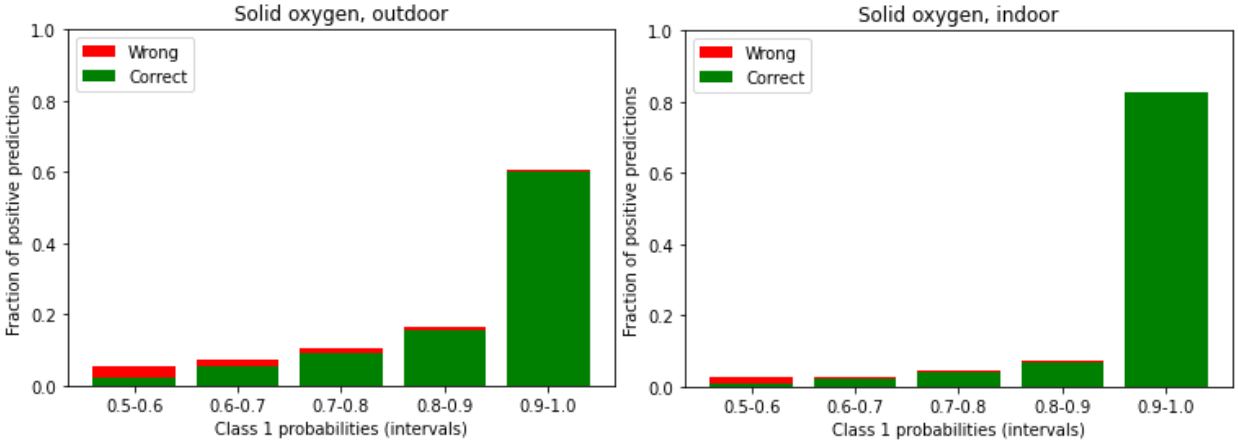
The bar charts representing the fraction of positive predictions for each probability class have been produced also in this case. As shown in *Figure 83* and *Figure 84*, the built model is quite confident in predicting the solidification or condensation of oxygen since most of the correct

positive predictions are made within the highest probability range. Moreover, the fraction of positive predictions is always higher in the “indoor” scenario.



(a) (b)

Figure 83. Fraction of positive predictions for each probability class, label: Liquid oxygen formation (Wide&Deep model trained over the raw databases where the temperature column has been removed) for the (a) outdoor and (b) indoor cases



(a) (b)

Figure 84. Fraction of positive predictions for each probability class, label: Solid oxygen formation (Wide&Deep model trained over the raw databases where the temperature column has been removed) for the (a) outdoor and (b) indoor cases.

Chapter 5
Further elaboration and discussion of the results

5.1 TensorFlow simulations

In this chapter the results obtained from the different simulations performed are analysed in depth.

5.1.1 Outdoor leakage studies

The results acquired after having trained and evaluated the three models – Linear, Deep and Wide&Deep – show that a deep approach is fundamental in order to obtain good performances. This happens both without data normalization and with data pre-processing. The Deep and Wide&Deep models have comparable performances, whereas the Linear model seems to perform slightly worse. This is crucial since the Deep model is more sensitive to the quality of the data, and being the database built upon sensors' measurements it is characterised by a high quality and detailed dataset, therefore the deep approach gives higher metrics.

As highlighted by the charts displayed in *Figure 52* (a) and *Figure 52* (b) in the previous chapter, the built model is extremely good at predicting the condensation or solidification of air components on the ground, given the high true positive rate for the highest probability range.
In order to evaluate the magnitude of the oxygen phase change phenomenon, *Figure 85* has been obtained by tracing a curve around the predicted extension along the x and y axis of the liquid or solid oxygen deposit, which is displayed in the charts in *Figure 56* and *Figure 57*.

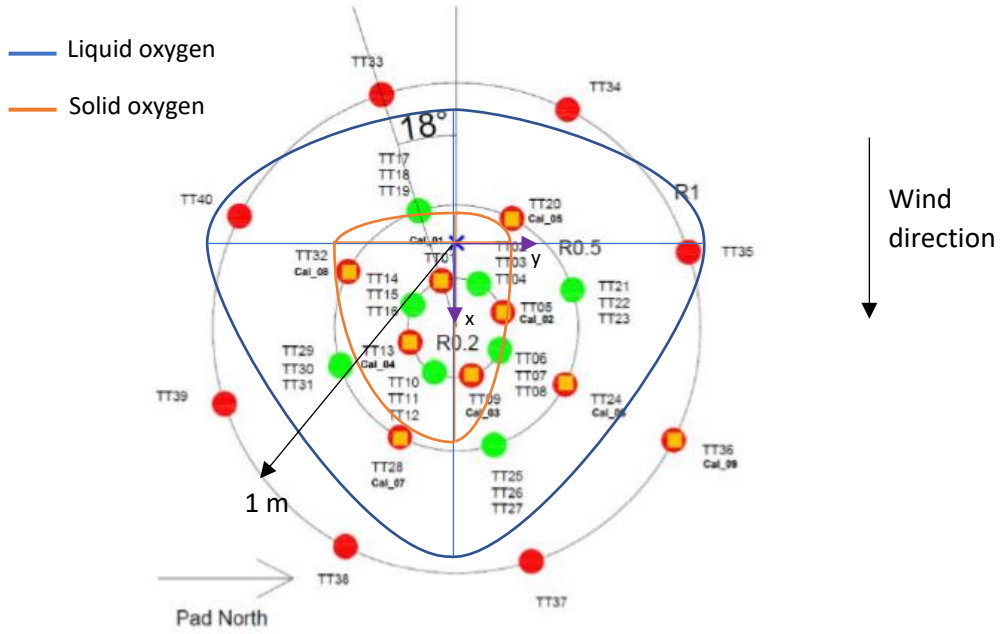


Figure 85. Predicted spatial extension of the liquid and solid oxygen on the test pad for outdoor leakage experiments (top view)

As depicted in *Figure 85*, the condensation or solidification of air oxygen on the ground is predicted to occur only within 1 m from the release point, which is experimentally confirmed by the studies carried out by the FFI (Aaneby, Gjesdal and Voie, 2021). The extension of the liquid oxygen area is larger than the solid oxygen one and it is elongated in the wind direction.

From the results obtained in Section 4.1.1.3, it is clear that the built model is extremely weak in predicting whether or not the concentration of hydrogen in air will overcome the LFL. The results are characterised by high accuracy and false positive rate, and low precision and recall, which is what is usually obtained when dealing with imbalanced dataset. To overcome these poor performances two approaches have been investigated:

- recall optimisation by maximizing the F-score;
- application of an over-sampling technique on the train dataset.

The results obtained after the recall optimisation are here discussed.

Maximising the F-score usually results in a lower threshold that leads to a higher recall, being the number of false negatives reduced. This is crucial in cases like the one examined here, where the hazardous event (hydrogen concentration higher than the LFL) is rare but with serious consequences. As depicted in *Figure 55* by varying the threshold the model performances improve, the model is more confident in predicting the positive class since most positive

predictions have a probability higher than 90%. A comparison between the fraction of positive predictions' distribution before and after optimising the recall can be seen in *Figure 86.*

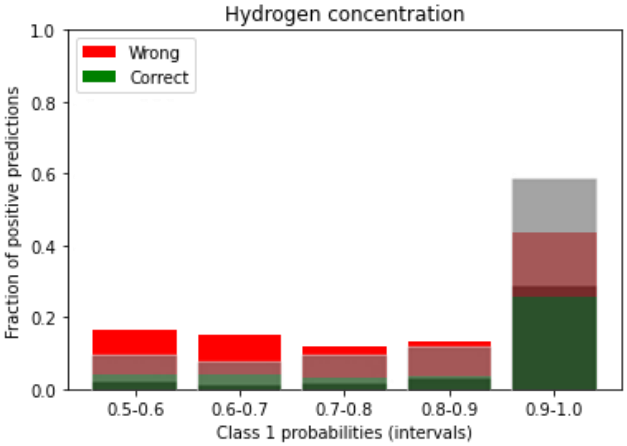


Figure 86. Fraction of positive predictions' distribution for each probability class; label: hydrogen concentration above the LFL. The red and green bars are obtained for a threshold value of 0.5; the grey bars are obtained for the best threshold that optimises the recall.

The main problem still remains the high false positive rate; the model might produce an excessively conservative prediction: if it predicts the hydrogen concentration to be above the lower flammability limit within 200 s even if this will not occur, safety measures will be activated anyway. Safety procedures typically include shutdowns of plants and evacuation of personnel, which determines a significant economic damage.

In order to better comprehend whether the model's prediction is reliable or too conservative, a visual representation of the positive predictions' distribution has been produced and it is displayed in *Figure 87*.

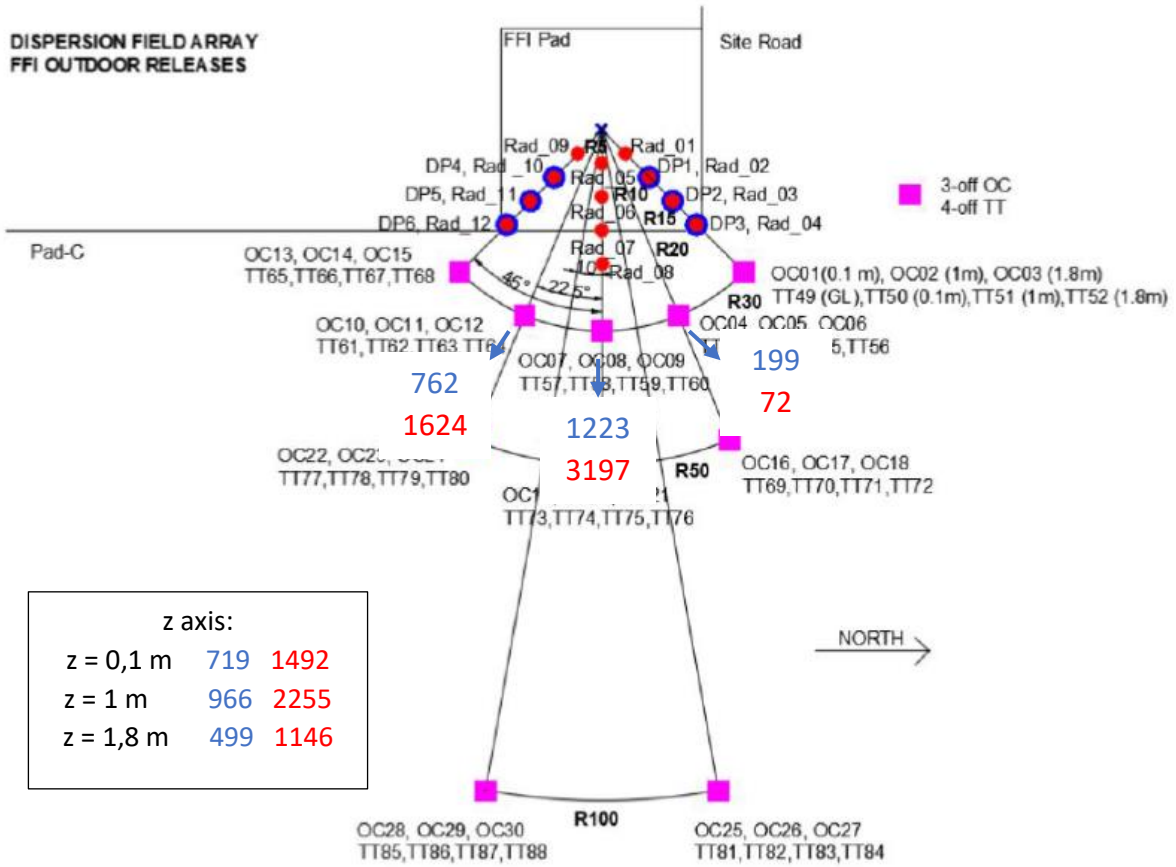


Figure 87. Real (blue) and predicted (red) positive labels' distribution

Figure 87 represents the distribution of the positive labels in the field: the blue number is the number of the real positive labels for each location in the field, whereas the red one is the number of the positive predictions for each location. The predicted extension of the gas cloud with a concentration of hydrogen higher than the LFL does not reach 50 m from the release point, as stated in the report by Aaneby, Gjesdal and Voie (2021). Having said that, it is evident that the model tends to overestimate the number of positive labels, according to the high false positive rate, with only one exception: for x = 27616 mm and y = 11481 mm (spatial coordinates from the release point) the model underestimates the number of positive labels. This might be due to the fact that, as mentioned in the report (Aaneby, Gjesdal and Voie, 2021), the wind blows from the south-west promoting the dispersion in that direction.

The same procedure has been carried out also considering polar coordinates instead of cartesian coordinates:

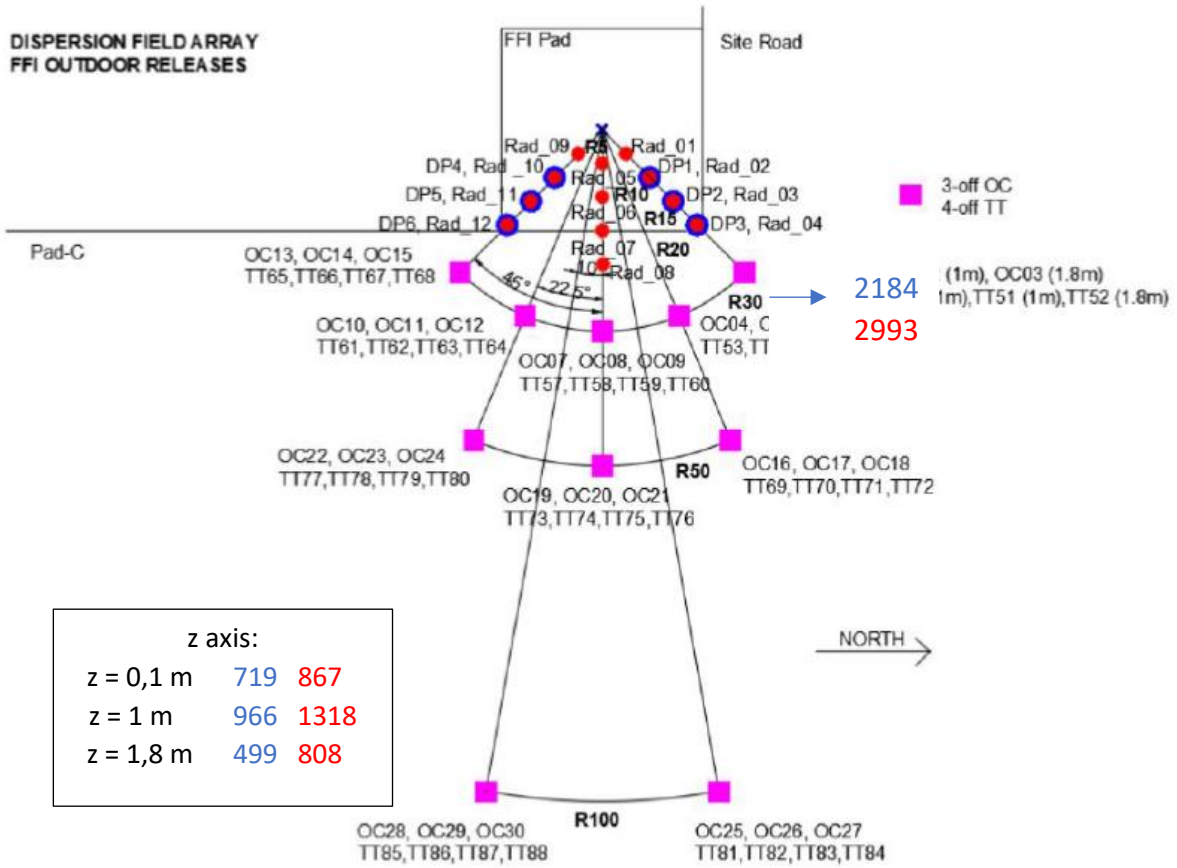


Figure 88. Real (blue) and predicted (red) positive labels' distribution considering the radial coordinate.

As previously stated, even considering a radial distance from the release point the concentration of hydrogen is predicted to be higher than the LFL only within 50 m. However, *Figure 88* shows that the number of positive predictions is closer to the number of real positive labels when the radial distance is considered, which means that the model should be fed with the radial coordinate instead of cartesian coordinates.

To conclude, despite the high false positive rate, the model is quite good at predicting the distribution of the positive labels: the concentration of hydrogen in air is predicted to overcome the lower flammability limit within 50 m from the release point, accordingly to what happened experimentally. Therefore, on one hand the model tends to overestimate the frequency by which such hydrogen concentration is above the LFL and this guarantees a conservative approach, which is crucial when dealing with safety aspects; on the other hand, further research must be carried out in order to improve and optimise the model's prediction.

As previously mentioned, another method to improve the model's performance is the application of an over-sampling technique on the train dataset. The results obtained after having

trained and tested the model over the over-sampled database presented in the previous section are here discussed.

Figure 59 shows that the model predicts a high number of false positives and a null value of false negatives, which means it is weaker when it comes to predicting the label "0".

The high false positive rate might be caused by the fact that the number of positive examples used to create new synthetic samples is extremely low and all the features associated to both the labels "0" and "1" are similar to one another. Therefore, the synthetically created data are not reliable. In fact, when analysing the labels' distribution in each features' pair both for the raw database, *Figure 89*, and the oversampled database, *Figure 90*, it is glaring that the minority class's synthetic data fall in the region of the majority class, which makes it difficult for the model to reliably predict the label "0". SMOTE, indeed, "is sometimes problematic particularly in very skewed dataset as it blindly generalizes the regions of the minority class without regard to the majority class instances, this depending on the sparseness of the minority instances in the dataset may lead to an increase in class overlapping" (Sowah et al., 2016).

The results cannot be further improved by evaluating the best threshold, since such threshold would end up being extremely close to 1 in order to minimise the number of false positives, which is unacceptable.

A good approach in this case, given the few positive samples, could be to under-sample the initial database, so to reduce the class imbalance by randomly deleting some data from the majority class. This technique is tendentially avoided because it might determine a loss of relevant information.

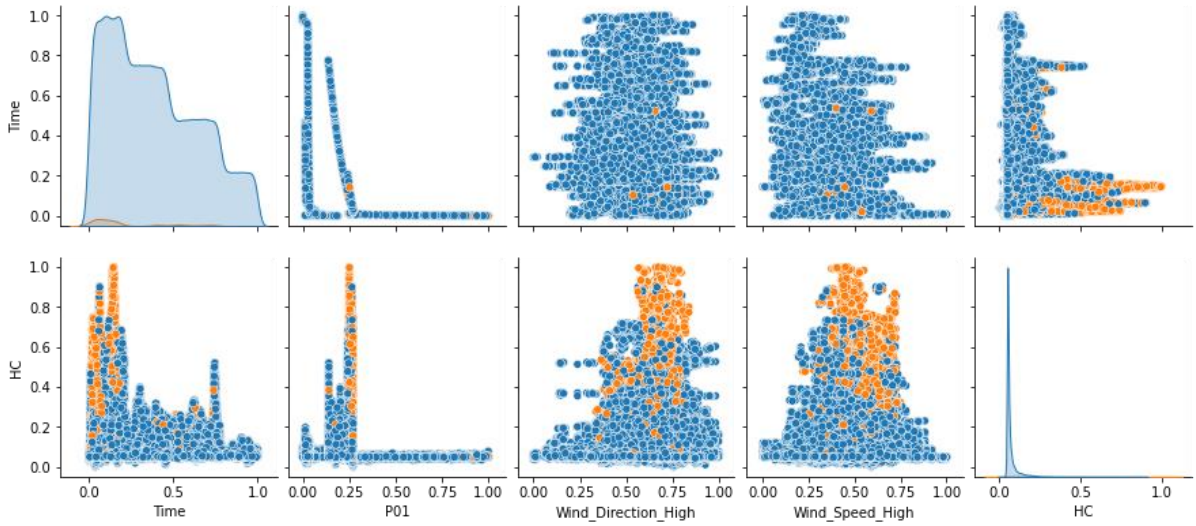


Figure 89. Raw database's labels distribution in the features plane

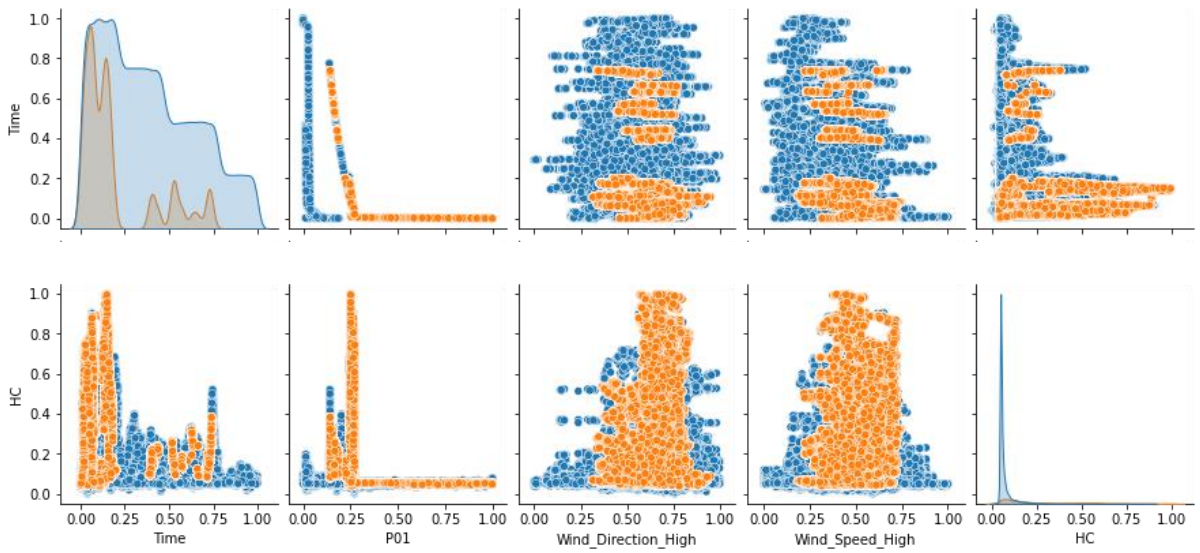


Figure 90. Oversampled database's labels distribution in the features plane

Another aspect to take into consideration is that, contrary to what one would expect, the model trained on normalized data gives much worse results than those obtained from the model trained and tested over raw data, as shown in the charts displayed in Chapter 4. This might be due to the normalization method chosen. The Min-max algorithm scales a variable in the training samples in the interval of [0, 1]. It is useful for preserving the relationships among the feature columns, unlike other normalization methods which are based on mean and standard deviations. However, when the testing samples fall outside of the training data range, the scaled values will be out of the interval [0, 1], and that may pose problems in some applications; moreover, it is very sensitive to the presence of outliers (Cao, Stojkovic and Obradovic, 2016). In fact, the original values in the normal range are squeezed into a narrow range after the scaling, as represented in *Figure 91*. Outliers do not necessarily need to be the result of measurement errors, but may also represent completely valid instances. So, if the dataset contains outliers it is going to be biased and results are systematically prejudiced (Singh and Singh, 2020, Pedregosa *et al.*, 2021b and Loukas, 2020).

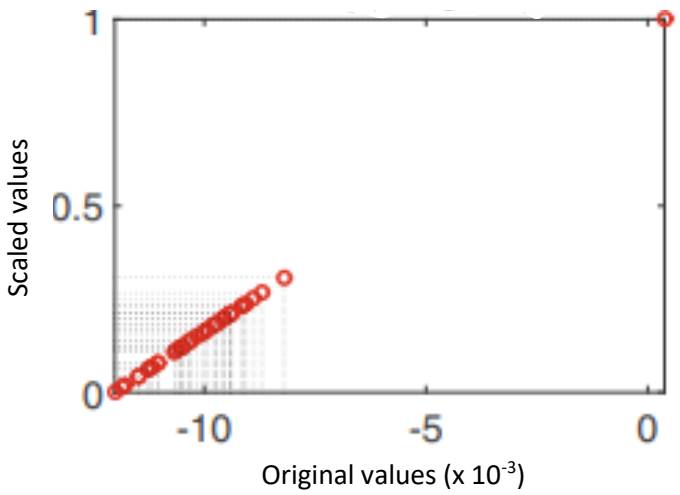


Figure 91. Example of how the presence of outliers would affect the data distribution, adapted from Cao, Stojkovic and Obradovic (2016).

To evaluate if this is the case, analogous charts to the one displayed in *Figure 91* have been produced for each feature. Some of the resulting graphs are displayed below.

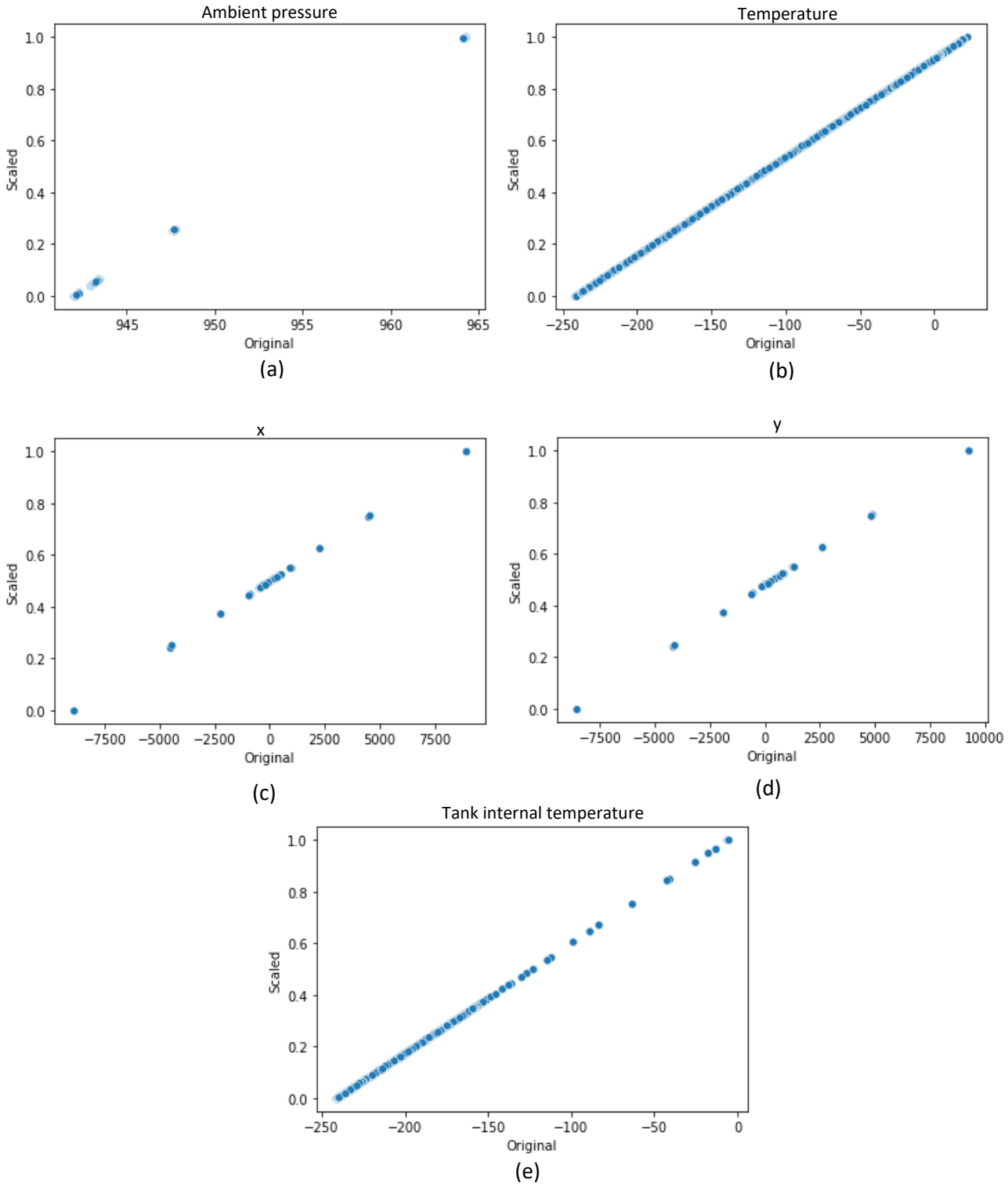


Figure 92. Outliers effect on data normalization for the feature columns (a) ambient pressure, (b) temperature, (c) x axis, (d) y axis and (e) tank internal temperature.

Figure 92 shows the trend of the scaled features as function of the raw features themselves. Some of the features contain outliers, such as the ambient pressure, the spatial coordinates and the hydrogen tank's internal temperature. This is proved by the fact that the inliers are squeezed into a narrow range, whereas the distribution of the temperature measured by the thermocouples seems to be rather homogeneous. These charts highlight that the normalization method chosen might not be the most proper one for this peculiar case, because of the presence of outliers in some of the features' values, therefore, further research should be performed in order to find the best normalization technique that would improve the model's performances.

5.1.2 Indoor leakage studies

As discussed in the previous section, even when considering the indoor tests' data the results obtained after having trained and evaluated the three models show that a deep approach is crucial to get the best performances. This can be observed when no normalization is applied to raw data, but it is even more evident when performing data pre-processing. The Deep and Wide &Deep models indeed have comparable performances, whereas the Linear model seems to perform slightly worse. Once again, this could be traced back to the high quality of the data, being the deep learning approach more sensitive to this parameter.

The model shows similar predictive capacities for both the condensation and solidification of oxygen on the ground.

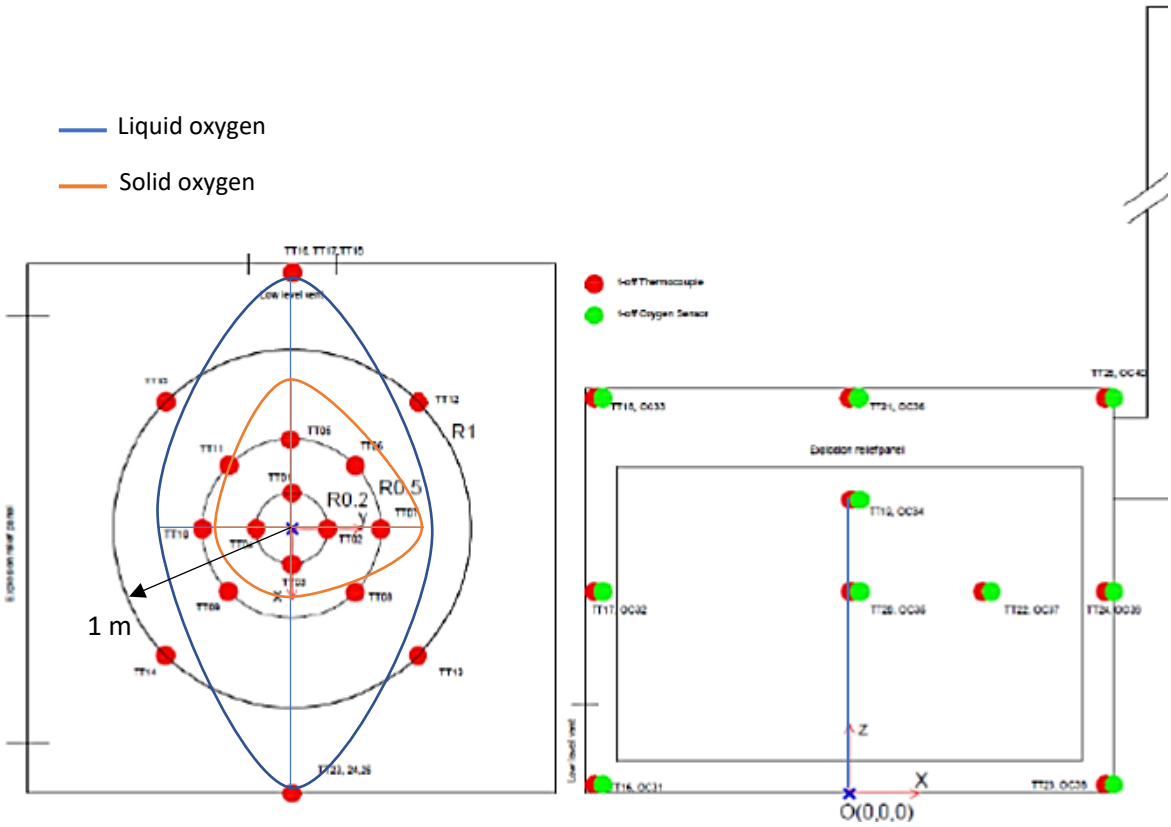


Figure 93. Predicted spatial extension of the oxygen phase change on the ground and inside the tank connection space.

In order to evaluate the magnitude of the oxygen phase change phenomenon, *Figure 93* has been obtained by tracing a curve around the predicted extension over the x and y axis of the liquid or solid oxygen deposit, which is displayed in the charts in *Figure 78* and *Figure 79*.

As depicted in *Figure 93* the condensation of air oxygen on the ground is predicted to occur within about 1.5 m from the release point, which is experimentally confirmed by the studies carried out by the FFI (Aaneby, Gjesdal and Voie, 2021). The extension of the liquid oxygen area is way larger than the solid oxygen one, and droplets of oxygen can be also found in the area above the ground in the tank connection space, whereas solid oxygen only settles on the concrete pad, where the lowest temperature can be measured due to the presence of liquid hydrogen.

An important aspect that must be underlined is that the results obtained by applying the model on normalized data are similar to those collected without pre-processing the database, despite what has been stated in the previous section for the outdoor studies. A possible explanation to this behaviour might be that the ambient pressure values and the spatial coordinates of the thermocouples show a more homogeneous distribution if compared to the outdoor case (see

Figure 94), and since the MinMaxScaler normalization technique is sensible to outliers, a more homogeneous distribution provides better predictive performances.

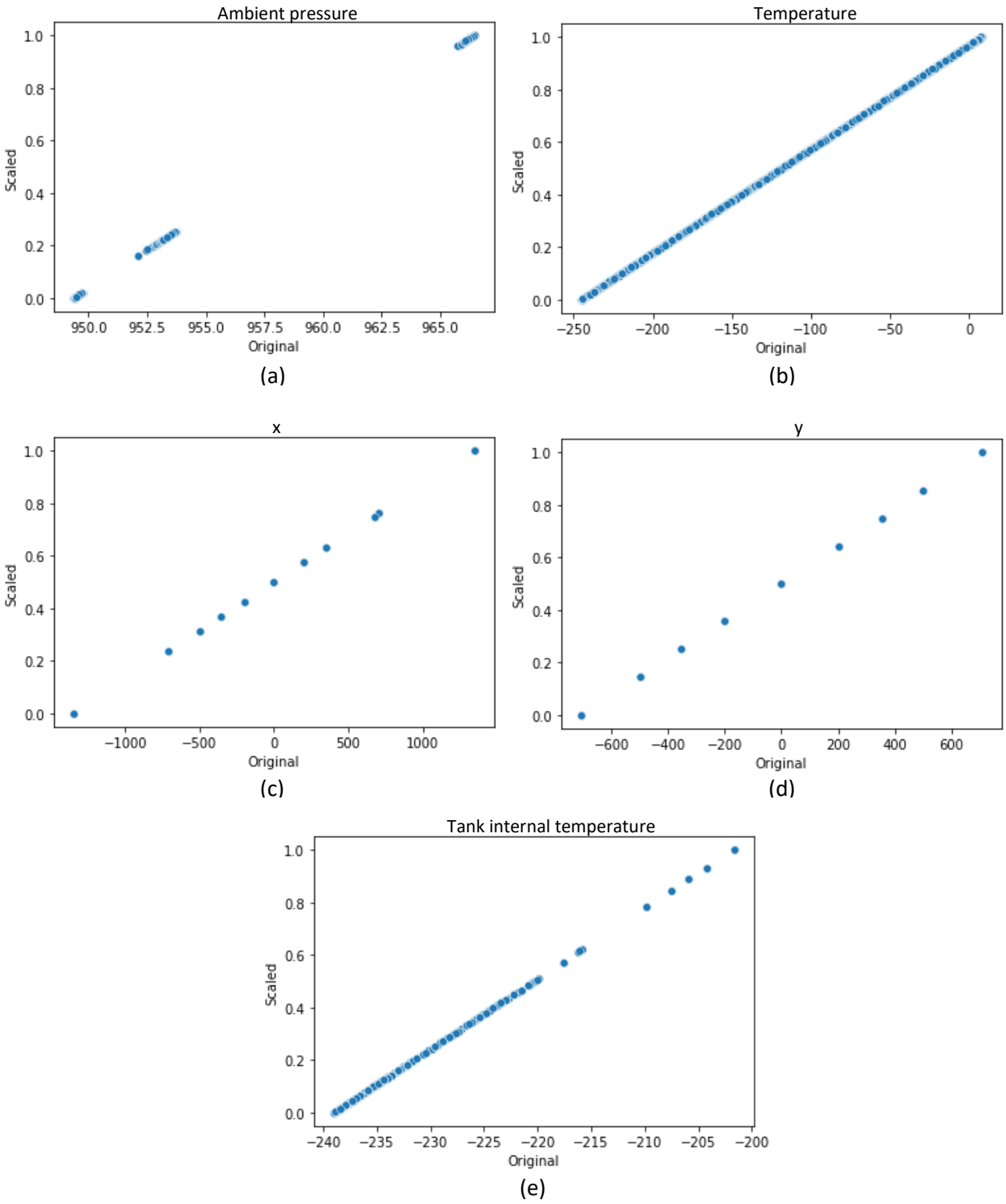


Figure 94.Outliers effect on data normalization for the feature columns (a) ambient pressure, (b) temperature, (c) x axis, (d) y axis and (e) tank internal temperature.

5.1.3 Comparison

Once the "outdoor" and the "indoor" results have been separately analysed, they can be compared to one another in order to evaluate possible similarities or major differences. First, when comparing the "outdoor" charts in *Figure 52* (a) and *Figure 52* (b) with the "indoor" ones in *Figure 77* (a) and *Figure 77* (b) it is immediately clear that the model has a higher confidence in predicting the condensation or freezing of air components in an enclosed room rather than in an open space. This is due to the fact that the condensation or freezing of air components in case of cryogenic spills is much more likely to occur if such release is confined within a closed room.

Second, when considering the predicted spatial extension of the liquid or solid deposit, this is much higher on the ground of the tank connection space then on the outdoor pad, as pointed out by comparing *Figure 85* and *Figure 93.* In fact, inside a container the temperature decreases much faster in case of hydrogen leakage determining air components to condense shortly after the release.

Lastly, air oxygen is predicted to condense only on the ground in the outdoor studies, whereas *Figure 78* (c) shows the predicted presence of liquid oxygen even above the ground, in the tank connection space's atmosphere, since it also reaches low temperatures being heat exchange with external air impeded by the container.

5.2 TensorFlow simulations: emergency situations

The following results have been obtained by training and testing the Wide&Deep model over a database where the temperature measurements column has been removed, which has been done in order to simulate a real hydrogen leak scenario. First, it is necessary to compare the charts in Section 4.1.3 with those represented in *Figure 52* (a), *Figure 52* (b) and in *Figure 77* (a), *Figure 77* (b). By removing the thermocouples measurements from the inputs, the model predicts positive labels with a lower degree of certainty, being the positive predictions characterised by lower probabilities. This is also confirmed by the higher number of False Positives shown in *Table 18*.

Second, the built model is quite confident in predicting the solidification or condensation of oxygen since most of the correct positive prediction are made within the highest probability range. Though, the model is better at predicting the condensation of air oxygen rather than its solidification, being the fraction of wrong positive predictions higher in the bar chart in

Figure 84. Moreover, when comparing the "outdoor" chart with the "indoor" one it is immediately clear that the model has a higher confidence in predicting the condensation or freezing of air components in an enclosed room rather than in an open space. This is due to the fact that the condensation or freezing of air components in case of cryogenic spills is much more likely to occur if such release is confined within an enclosed room.

As it can be observed in *Figure 95* the spatial extension of liquid oxygen is higher than the solid oxygen one. Moreover, the extension of the oxygen deposit's surface, either liquid or solid, tends to be more significant in the indoor leakage scenario.

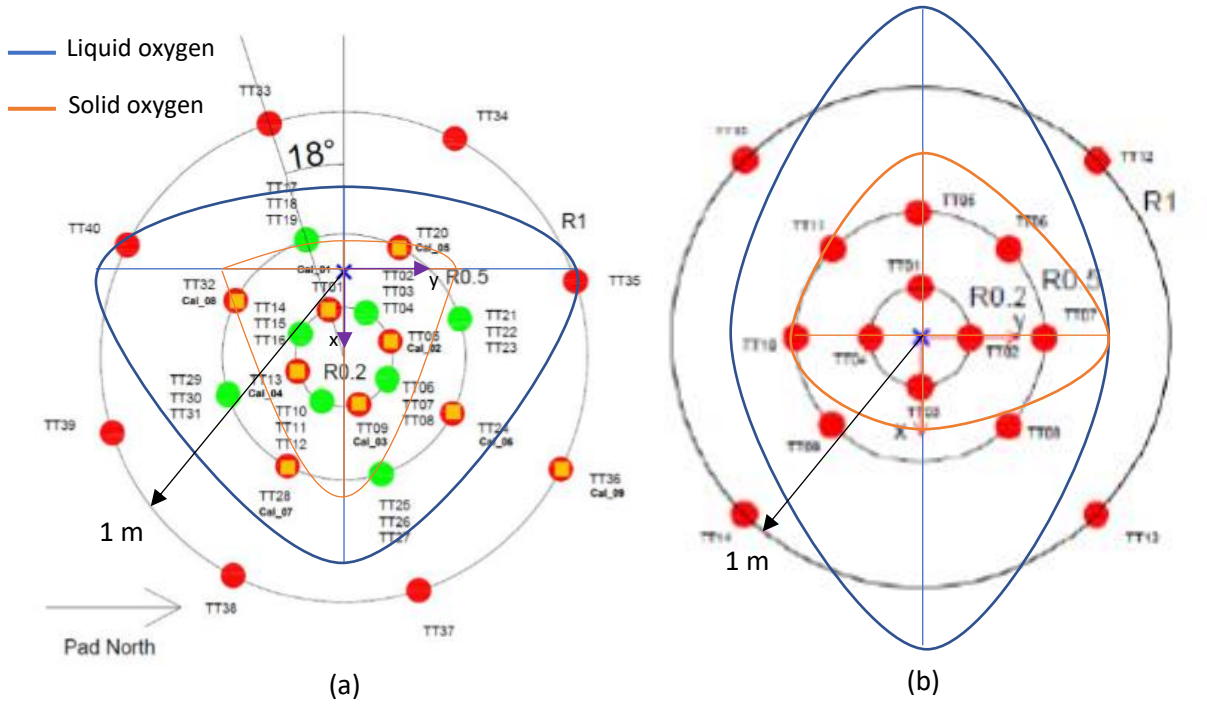


Figure 95. Comparison between the indoor and outdoor spatial extension of liquefied and frozen air oxygen in case of accidental hydrogen release for the (a) outdoor and (b) indoor leakage cases.

The predicted spatial extensions are quite similar to those obtained when considering the thermocouples' measurements in the input database, this means that the model trained and tested over data that might be easily collected in emergency situations is quite reliable in predicting the risk area around the cryogenic liquid release point.

5.3 Safety procedures in case of hydrogen leakage

In case of a real liquid hydrogen accidental release the proposed method can be utilised to predict if, where and to what extent the oxygen in the air condenses or freezes on the ground

and the spatial extension of the vapour cloud where the concentration of hydrogen is higher than the lower flammability limit, helping the decision-making process related to the safety actions that must be taken.

First, these safety actions usually consist in restricting people not wearing protective equipment from the spill area: liquid hydrogen will condense humidity water in the atmosphere, producing a vapour cloud and the zone of flammability may extend beyond this cloud (Linde, 2021 and Pohanish, 2008). Then all the ignition sources must be safely removed. A shut down procedure must be implemented: firstly, it is necessary to stop the leak by closing isolation valves upon gas detection; secondly, the hydrogen system needs to be shut down too (DNV, 2021).

Now, if the model, fed with release rate, orientation, ambient conditions and coordinates of the points we are interested in evaluating, predicts the condensation or freezing of oxygen in such points, this could lead to the risk of a condensed phase explosion. Moreover, the gaseous flammable mixture air-hydrogen if ignited could cause a jet fire or even an explosion in some conditions. Therefore, proper safety procedures must be followed. In particular, safety distances can be calculated so that personnel can evacuate in a safe area, far away from the incident source. “Separation distances, or alternatively referred to as safety distances, are the minimum separation between a hazard source and an object (human, equipment or environment), which will mitigate the effect of a likely foreseeable incident and prevent a minor incident escalating into a larger incident.” (Pohanish, 2008).

These safety distances can be evaluated on the basis of threshold values of the physical effects of fires and explosions: the operator has to reach a distance corresponding to a value of thermal radiation or overpressure sufficiently low not to cause any damage.

Table 19. Thermal radiation threshold values.

	Thermal radiation (kW/m^2)	
With protective equipment	4 ÷ 5	(Crocker and Napier, 1986)
Without protective equipment	1.5	(Atallah and Allan, 1971)

Table 20. Peak overpressure threshold values.

	Peak overpressure (kPa)	
Eardrums rupture	35	(Malhotra, Carson, and Mcfadden, 2017)
Lungs lethal damage	210	(Malhotra, Carson, and Mcfadden, 2017)

The explosive energy associated to a condensed phase explosion originated by an outdoor liquid hydrogen release of 0.14 kg/s is similar to a blast of about 2 kg TNT (Atkinson, 2021).
In order to evaluate the safety distance in case of explosion it is necessary to determine the scaled distance z (see Equation (11)) first, using the peak overpressure diagram in *Figure 96* (TNO, 2005).

$$z = \frac{r}{(m_{TNT,eq})^{1/3}} \qquad (11)$$

In Equation (11) r is the distance between the blast's epicentre and the target.

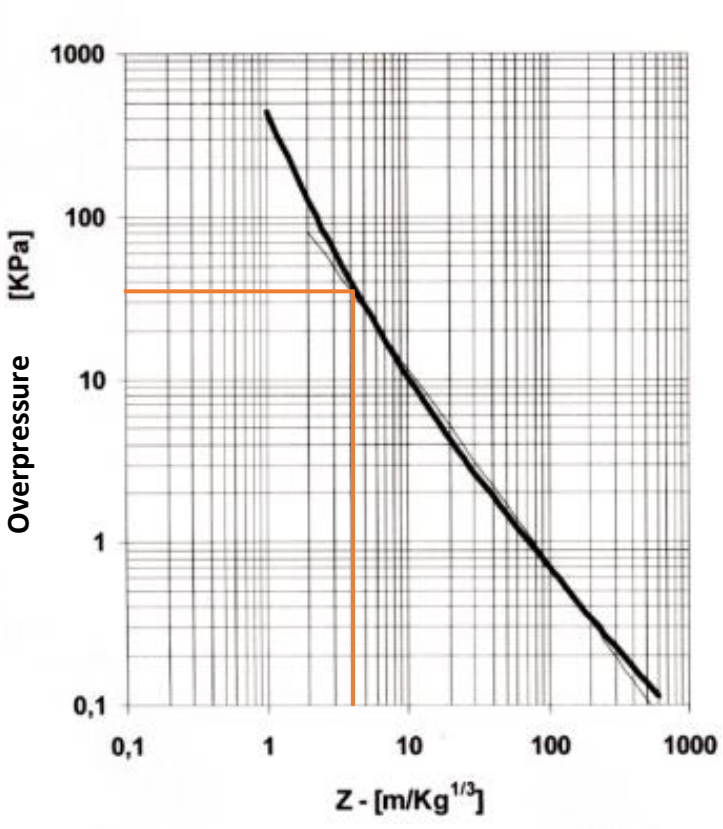


Figure 96. TNT peak overpressure diagram, adapted from Genova, Ripani and Silvestrini (n.a.).

Considering a peak overpressure of 35 kPa, at which humans experience minor damages, the corresponding value of z (scaled distance) is about 4 $\frac{m}{kg^{1/3}}$, and considering a $m_{TNT,eq} = 2\ kg$ (Atkinson, 2021) the distance between the explosion's epicentre and the target (human in this case) will be about 5 m, which is expected to increase as the release rate increases.

Following the condensed phase explosion, a fireball usually occurs.
The resulting fireball expands to reach a diameter of about 8 m (Atkinson, 2021). The fireball can be described through a surface emitter model (TNO, 2005):

$$I = E \cdot F \cdot \tau_a \qquad (12)$$

Where E, which is the specific emissivity, is usually about 250÷350 kW/m^2 (TNO, 2005); F is the view factor, which expresses the proportion of the radiation which leaves the surface of the fire and strikes the target, and can be calculated as $F = \frac{D^2}{4X^2}$ where D and X are displayed in *Figure 97* and represent respectively the fireball diameter and the distance between the target and the fireball's centre. τ_a is the atmospheric transmissivity, which can be calculated as $\tau_a = 2.02(p_w \cdot L)^{-0.09}$, where p_w is the partial pressure of water in air (TNO, 2005). For a value of p_w = 1200 Pa, $\tau_a = 2.02(1200 \cdot L)^{-0.09}$.

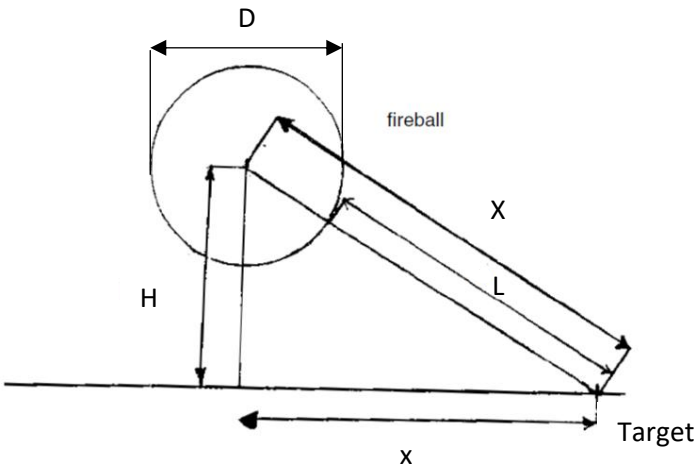


Figure 97. Schematic representation of a fireball, adapted from TNO (2005).

With these data, it is possible to calculate the safety distance corresponding to a value of thermal radiation of 1.5 kW/m^2:

$$1.5 = 300 \cdot \frac{8^2}{4 \cdot X^2} \cdot 2.02(1200 \cdot (X - 4))^{-0.09} \qquad (13)$$

From Equation (13): X = 49.2 m.

Being H = 0.75÷1 D, so considering H = 6 m, $x = \sqrt{X^2 - H^2} = 48.8\ m$.

In this case, since 48.8 m > 5 m, the safety distance is given by the distance from the epicentre of the fire necessary for the target to be protected from the thermal radiation, having hypothesised that the projection of the epicentre of the fireball coincides with the release point – considering the magnitude of x, and that the fireball originates from the aerosol right above the liquid pool, it is a plausible hypothesis.

In the following, the focus is placed on the case in which no condensed phase explosion occurs after ignition of the flammable mixture, which results in a jet fire. The experimental studies conducted by the FFI (Aaneby, Gjesdal and Voie, 2021) consider a liquid hydrogen release rate

of about 0.71 kg/s, which in case of ignition generates an initial fireball followed by a jet fire. The thermal radiation measured at 5 m from the release point is about 61.1 kW/m^2. The thermal radiation decreases with the square of the distance from the source, so in order to reach an acceptable value (< 1.5 kW/m^2) the operator must move from the release point of about 30 m (Aaneby, Gjesdal and Voie, 2021).

If the gaseous flammable mixture hydrogen-air is ignited during an indoor liquid hydrogen release with a release rate of about 0.5 kg/s, an explosion generally occurs due to the confinement of the flame inside an enclosed room. The explosion might take a long time to happen, around 30 minutes, if the enclosed room is completely sealed preventing airflow. The peak overpressure will be of about 0.2 bar, which is not even high enough to determine eardrums rupture on human. If the enclosed room is not perfectly tight, the airflow increases and that determines the explosion to occur immediately after ignition (about 15 s after ignition) with a much higher peak overpressure. A peak overpressure of about 2 bar was measured inside the enclosed room, which provokes its complete destruction (Aaneby, Gjesdal and Voie, 2021).

Separation distances might be reduced by the use of mitigation measures. For example, water sprays to reduce thermal radiation effects were considered in this work, or walls to protect from blast and missiles deriving from explosions may be another effective solution. It is important to check that the mitigation measures will not increase the impact of other hazards. For example, walls to protect against blast and missiles may inhibit the dispersion of hydrogen increasing the time flammable concentrations are present and the likelihood of an ignition. The partial confinement introduced by the wall may also increase the probability of a flammable cloud detonation (Pohanish, 2008).

Chapter 6
Conclusions and further work

A Machine Learning method for predicting the condensation or freezing of air oxygen and the formation of flammable atmospheres in case of liquid hydrogen spills has been developed. The method has involved the creation of several databases.

Three models have been trained and tested on such databases: Linear, Deep and Wide&Deep models. The performances of these models have been evaluated over their predicting capabilities through the use of performance metrics. In general, the models have shown extremely good prediction capabilities when it comes to evaluating the formation of liquid or solid oxygen. A Deep approach seems to be essential in order to obtain high performance metrics.

The Machine Learning models built are relatively weak when it comes to predicting whether or not the concentration of gas hydrogen in air will reach the lower flammability limit, giving rise to a flammable cloud. This might be due to the low number of positive labels, which translates into fewer chances to learn: the formation of a flammable cloud seems to be a relatively infrequent event if compared to the condensation or freezing of air components. Furthermore, some data pre-processing functions, despite being normally performed nowadays, might tend to worsen the results. This may be due to the presence of outliers in the database feature columns. Given these drawbacks, other data normalization techniques may be investigated. Moreover, if the aspect of interest is to predict the formation of a flammable cloud so to avoid fires and explosions in case of ignition implementing proper safety measures, further works might focus on finding the most suitable over-sampling technique in order to increase the number of positive events and properly train and evaluate the models.

In any case, the built models have the potential to play an essential role in real-life accidental hydrogen release scenarios. In fact, if detectors reveal the presence of a hydrogen spill, safety measures must be activated. Such safety measures usually comprehend systems shutdown and the personnel evacuation to safe spots placed at great distance from the release point. This leads to a significant economic loss. Through these models, knowing the spill’s flow rate and other parameters easily measured, it would be possible to predict whether or not air oxygen will form liquid or solid deposits on the ground, posing a risk of condensed phase explosions, and if the

gas cloud will reach a concentration of hydrogen such to cause fire in case of ignition. If none of these phenomena are predicted to occur, the safety measures to implement might not determine the shutdown of the whole plant, but only local shutdowns and more limited safety distances.

Appendix A
Codes

A.1 Linear Model

```
from __future__ import absolute_import, division, print_function, unicode_literals
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import precision_recall_curve
import matplotlib.pyplot as plt
from IPython.display import clear_output
import tensorflow as tf
from sklearn import preprocessing

df = pd.read_csv('/content/Database.csv')

columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'TT', 'x', 'y', 'z', 'liquid_oxygen', 'solid_oxygen']

# Second Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'HC', 'x', 'y', 'z', ‘label’]
'''


# Third Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Purge','Sealing', 'P01',
         'P02', 'P03', 'P04', 'PT_01', 'PT_02', 'PT_03', 'PT_04', 'TT', 'x', 'y', 'z', 'liquid_oxygen',
         'solid_oxygen']
'''


df.columns = columns

dftrain = df.sample(frac=0.75, random_state=25)
dfeval = df.drop(dftrain.index)

# Normalization
numeric=df._get_numeric_data()
```

```
for column in numeric:
  xtrain = np.array(dftrain[column]).reshape(-1,1)
  xtest = np.array(dfeval[column]).reshape(-1,1)
  min_max_scaler = MinMaxScaler()
  min_max_scaler.fit(xtrain)
  xtrain_scaled=min_max_scaler.transform(xtrain)
  xtest_scaled=min_max_scaler.transform(xtest)
  dftrain[column]= xtrain_scaled
  dfeval[column]= xtest_scaled

insert_label = input("Insert label type: ")

if insert_label == "liquid_oxygen":
    dftrain=dftrain.drop(columns=['solid_oxygen'])
    dfeval=dfeval.drop(columns=['solid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('liquid_oxygen')
    y_eval = dfeval.pop('liquid_oxygen')
else:
    dftrain=dftrain.drop(columns=['liquid_oxygen'])
    dfeval=dfeval.drop(columns=['liquid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('solid_oxygen')
    y_eval = dfeval.pop('solid_oxygen')


#Second Database
'''
dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
dftrain = pd.read_csv('dftrain.csv')
dfeval = pd.read_csv('dfeval.csv')
y_train = dftrain.pop('label')
y_eval = dfeval.pop('label')
'''


CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low','TT', 'x', 'y', 'z']
```

```
# Second Database
'''
CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low', 'HC', 'x', 'y', 'z']
'''

# Third Database
'''
CATEGORICAL_COLUMNS = ['Purge','Sealing']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity','P01',
                'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'TT', 'x', 'y', 'z']
'''

feature_columns = []
for feature_name in CATEGORICAL_COLUMNS:
  vocabulary = dftrain[feature_name].unique()
  feature_columns.append(tf.feature_column.categorical_column_with_vocabulary_list(featur
e_name, vocabulary))
for feature_name in NUMERIC_COLUMNS:
  feature_columns.append(tf.feature_column.numeric_column(feature_name, dtype=tf.float64
))

crossed = [tf.feature_column.crossed_column(['x', 'y', 'z'], hash_bucket_size=1000),
         tf.feature_column.crossed_column(['Release_rate', 'P01', 'P02', 'P03', 'P04'],
                                  hash_bucket_size=1000),
         tf.feature_column.crossed_column(['x', 'y', 'z', 'TT'], hash_bucket_size=1000)]

# Second Database
'''
crossed = [tf.feature_column.crossed_column(['x', 'y', 'z'], hash_bucket_size=1000),
         tf.feature_column.crossed_column(['Release_rate', 'P01', 'P02', 'P03', 'P04'],
                                  hash_bucket_size=1000),
         tf.feature_column.crossed_column(['x', 'y', 'z', 'HC'], hash_bucket_size=1000)]
'''

def make_input_fn(data_df, label_df, num_epochs=10, batch_size=32):
  def input_function():
    ds = tf.data.Dataset.from_tensor_slices((dict(data_df), label_df))
    ds = ds.batch(batch_size).repeat(num_epochs)
    return ds
  return input_function

train_input_fn = make_input_fn(dftrain, y_train)
eval_input_fn = make_input_fn(dfeval, y_eval, num_epochs=1)

classifier = tf.estimator.LinearClassifier(feature_columns=feature_columns + crossed)
```

```
classifier.train(train_input_fn)

result = classifier.evaluate(eval_input_fn)
clear_output()
print(result)


# Precision-recall curve

predictions = classifier.predict(eval_input_fn)
probs = pd.Series([pred['probabilities'][1] for pred in predictions])
precision, recall, threshold = precision_recall_curve(y_eval, probs)
plt.plot(recall, precision)
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.show()


# Predictions

probs0 = 1-probs
print(probs0)
predict=[]
for item in probs:
  if item >0.5:
    predict.append(1)
  else:
    predict.append(0)
pred=pd.Series(predict)
df_predictions=pd.DataFrame({"Expected": y_eval, "Predictions": pred, "Prob0": probs0, "Pr
ob1": probs})
df_predictions.to_csv('df_predictions.csv', index=False, encoding='utf-8')


# Confusion matrix

from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y_eval, pred)
print(conf_matrix)
```

A.2 Deep Model

```
from __future__ import absolute_import, division, print_function, unicode_literals
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import precision_recall_curve
import matplotlib.pyplot as plt
from IPython.display import clear_output
import tensorflow as tf
from sklearn import preprocessing

df = pd.read_csv('/content/Database.csv')

columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'TT', 'x', 'y', 'z', 'liquid_oxygen', 'solid_oxygen']

# Second Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'HC', 'x', 'y', 'z', ‘label’]
'''

# Third Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Purge','Sealing', 'P01',
         'P02', 'P03', 'P04', 'PT_01', 'PT_02', 'PT_03', 'PT_04', 'TT', 'x', 'y', 'z', 'liquid_oxygen',
         'solid_oxygen']
'''

df.columns = columns

dftrain = df.sample(frac=0.75, random_state=25)
dfeval = df.drop(dftrain.index)

# Normalization
numeric=df._get_numeric_data()
for column in numeric:
  xtrain = np.array(dftrain[column]).reshape(-1,1)
  xtest = np.array(dfeval[column]).reshape(-1,1)
  min_max_scaler = MinMaxScaler()
  min_max_scaler.fit(xtrain)
  xtrain_scaled=min_max_scaler.transform(xtrain)
```

```
  xtest_scaled=min_max_scaler.transform(xtest)
  dftrain[column]= xtrain_scaled
  dfeval[column]= xtest_scaled

insert_label = input("Insert label type: ")

if insert_label == "liquid_oxygen":
    dftrain=dftrain.drop(columns=['solid_oxygen'])
    dfeval=dfeval.drop(columns=['solid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('liquid_oxygen')
    y_eval = dfeval.pop('liquid_oxygen')
else:
    dftrain=dftrain.drop(columns=['liquid_oxygen'])
    dfeval=dfeval.drop(columns=['liquid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('solid_oxygen')
    y_eval = dfeval.pop('solid_oxygen')


#Second Database
'''
dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
dftrain = pd.read_csv('dftrain.csv')
dfeval = pd.read_csv('dfeval.csv')
y_train = dftrain.pop('label')
y_eval = dfeval.pop('label')
'''

CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low','TT', 'x', 'y', 'z']

# Second Database
'''
CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low', 'HC', 'x', 'y', 'z']
```

```
'''

# Third Database
'''
CATEGORICAL_COLUMNS = ['Purge','Sealing']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity','P01',
                   'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'TT', 'x', 'y', 'z']
'''

feature_columns = []
for feature_name in CATEGORICAL_COLUMNS:
  vocabulary = dftrain[feature_name].unique()
  feature_columns.append(tf.feature_column.indicator_column(tf.feature_column.categorical_
column_with_vocabulary_list(feature_name, vocabulary)))
for feature_name in NUMERIC_COLUMNS:
  feature_columns.append(tf.feature_column.numeric_column(feature_name, dtype=tf.float64
))


def make_input_fn(data_df, label_df, num_epochs=10, batch_size=32):
  def input_function():
    ds = tf.data.Dataset.from_tensor_slices((dict(data_df), label_df))
    ds = ds.batch(batch_size).repeat(num_epochs)
    return ds
  return input_function

train_input_fn = make_input_fn(dftrain, y_train)
eval_input_fn = make_input_fn(dfeval, y_eval, num_epochs=1)

classifier = tf.estimator.DNNClassifier(feature_columns=feature_columns,
                                        hidden_units=[1024, 512, 256])

classifier.train(train_input_fn)

result = classifier.evaluate(eval_input_fn)
clear_output()
print(result)



# Precision-recall curve

predictions = classifier.predict(eval_input_fn)
probs = pd.Series([pred['probabilities'][1] for pred in predictions])
precision, recall, threshold = precision_recall_curve(y_eval, probs)
plt.plot(recall, precision)
plt.xlabel('Recall')
plt.ylabel('Precision')
```

```
plt.show()


# Predictions

probs0 = 1-probs
print(probs0)
predict=[]
for item in probs:
  if item >0.5:
    predict.append(1)
  else:
    predict.append(0)
pred=pd.Series(predict)
df_predictions=pd.DataFrame({"Expected": y_eval, "Predictions": pred, "Prob0": probs0, "Pr
ob1": probs})
df_predictions.to_csv('df_predictions.csv', index=False, encoding='utf-8')


# Confusion matrix

from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y_eval, pred)
print(conf_matrix)
```

A.3 Wide&Deep Model

```
from __future__ import absolute_import, division, print_function, unicode_literals
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from sklearn.metrics import precision_recall_curve
import matplotlib.pyplot as plt
from IPython.display import clear_output
import tensorflow as tf
from sklearn import preprocessing
from imblearn.over_sampling import SMOTENC
df = pd.read_csv('/content/Database.csv')

columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'TT', 'x', 'y', 'z', 'liquid_oxygen', 'solid_oxygen']

# Second Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'Wind_Direction_High',
        'Wind_Direction_Low', 'Wind_Speed_High', 'Wind_Speed_Low',
        'HC', 'x', 'y', 'z', ‘label’]
'''

# Third Database
'''
columns = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity', 'Purge','Sealing', 'P01',
         'P02, 'P03', 'P04', 'PT_01', 'PT_02', 'PT_03', 'PT_04', 'TT', 'x', 'y', 'z', 'liquid_oxygen',
         'solid_oxygen']
'''

df.columns = columns

dftrain = df.sample(frac=0.75, random_state=25)
dfeval = df.drop(dftrain.index)

# SMOTE for the Second Database
'''
smotenc = SMOTENC([4],random_state = 101, sampling_strategy=0.3)
         X_oversample, y_oversample = smotenc.fit_resample(df_train[['Time', 'Ambient_pres
         sure',  'Release_rate', 'Humidity', 'Release_orientation',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low','HC', 'x', 'y', 'z']], df_train['label'])
```

```
smote_array = np.concatenate([X_oversample, y_oversample.values.reshape(-1, 1)], axis=1)
dftrain_oversample = pd.DataFrame(smote_array, columns=['Time', 'Ambient_pressure',
          'Release_rate', 'Humidity', 'Release_orientation',
         'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
          'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
          'Wind_Speed_Low','HC', 'x', 'y', 'z', 'label'])

dftrain_oversample.to_csv('dftrain_oversample.csv', index=False, encoding='utf-8')
dftrain = pd.read_csv('dftrain_oversample.csv')
dfeval = pd.read_csv('dfeval.csv')
y_train = dftrain.pop('label')
y_eval = dfeval.pop('label')
'''

# Normalization
numeric=df._get_numeric_data()
for column in numeric:
  xtrain = np.array(dftrain[column]).reshape(-1,1)
  xtest = np.array(dfeval[column]).reshape(-1,1)
  min_max_scaler = MinMaxScaler()
  min_max_scaler.fit(xtrain)
  xtrain_scaled=min_max_scaler.transform(xtrain)
  xtest_scaled=min_max_scaler.transform(xtest)
  dftrain[column]= xtrain_scaled
  dfeval[column]= xtest_scaled

insert_label = input("Insert label type: ")

if insert_label == "liquid_oxygen":
    dftrain=dftrain.drop(columns=['solid_oxygen'])
    dfeval=dfeval.drop(columns=['solid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('liquid_oxygen')
    y_eval = dfeval.pop('liquid_oxygen')
else:
    dftrain=dftrain.drop(columns=['liquid_oxygen'])
    dfeval=dfeval.drop(columns=['liquid_oxygen'])
    dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
    dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
    dftrain = pd.read_csv('dftrain.csv')
    dfeval = pd.read_csv('dfeval.csv')
    y_train = dftrain.pop('solid_oxygen')
    y_eval = dfeval.pop('solid_oxygen')
```

```
# Second Database
'''
dftrain.to_csv('dftrain.csv', index=False, encoding='utf-8')
dfeval.to_csv('dfeval.csv', index=False, encoding='utf-8')
dftrain = pd.read_csv('dftrain.csv')
dfeval = pd.read_csv('dfeval.csv')
y_train = dftrain.pop('label')
y_eval = dfeval.pop('label')
'''
CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low','TT', 'x', 'y', 'z']

# Second Database
'''
CATEGORICAL_COLUMNS = ['Release_orientation']
NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity',
        'P01', 'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04',
         'Wind_Direction_High', 'Wind_Direction_Low', 'Wind_Speed_High',
         'Wind_Speed_Low', 'HC', 'x', 'y', 'z']
'''

# Third Database
'''

CATEGORICAL_COLUMNS = ['Purge','Sealing']

NUMERIC_COLUMNS = ['Time', 'Ambient_pressure', 'Release_rate', 'Humidity','P01',
                'P02', 'P03', 'P04', 'PT_01', 'PT_02','PT_03', 'PT_04', 'TT', 'x', 'y', 'z']
'''


deep_feature_columns = []
for feature_name in CATEGORICAL_COLUMNS:
  vocabulary = dftrain[feature_name].unique()
  deep_feature_columns.append(tf.feature_column.indicator_column(tf.feature_column.catego
rical_column_with_vocabulary_list(feature_name, vocabulary)))
for feature_name in NUMERIC_COLUMNS:
  deep_feature_columns.append(tf.feature_column.numeric_column(feature_name, dtype =
          tf.float64))

crossed = [tf.feature_column.crossed_column(['x', 'y', 'z'], hash_bucket_size=1000),
         tf.feature_column.crossed_column(['Release_rate', 'P01', 'P02', 'P03', 'P04'],
                                  hash_bucket_size=1000),
         tf.feature_column.crossed_column(['x', 'y', 'z', 'TT'], hash_bucket_size=1000)]

# Second Database
'''
crossed = [tf.feature_column.crossed_column(['x', 'y', 'z'], hash_bucket_size=1000),
```

```
        tf.feature_column.crossed_column(['Release_rate', 'P01', 'P02', 'P03', 'P04'],
                                         hash_bucket_size=1000),
        tf.feature_column.crossed_column(['x', 'y', 'z', 'HC'], hash_bucket_size=1000)]
'''

def make_input_fn(data_df, label_df, num_epochs=10, batch_size=32):
  def input_function():
    ds = tf.data.Dataset.from_tensor_slices((dict(data_df), label_df))
    ds = ds.batch(batch_size).repeat(num_epochs)
    return ds
  return input_function

train_input_fn = make_input_fn(dftrain, y_train)
eval_input_fn = make_input_fn(dfeval, y_eval, num_epochs=1)

classifier = tf.estimator.DNNLinearCombinedClassifier(linear_feature_columns=crossed,
                                      dnn_feature_columns=deep_feature_columns,
                                      dnn_hidden_units=[1024, 512, 256])

classifier.train(train_input_fn)

result = classifier.evaluate(eval_input_fn)
clear_output()
print(result)


# Precision-recall curve

predictions = classifier.predict(eval_input_fn)
probs = pd.Series([pred['probabilities'][1] for pred in predictions])
precision, recall, threshold = precision_recall_curve(y_eval, probs)
plt.plot(recall, precision)
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.show()


# Best Threshold for the Second Database
'''
from numpy import argmax
fscore = (1+(1.5)*(1.5))*(precision * recall) / (((1.5)*(1.5))*precision + recall)
# locate the index of the largest f score
ix = argmax(fscore)
print('Best Threshold=%f, F-Score=%.3f' % (threshold[ix], fscore[ix]))
print('Precision=%f, Recall=%.3f' % (precision[ix], recall[ix]))
plt.plot(threshold, fscore[1:])
plt.xlabel('Threshold')
plt.ylabel('Fscore')
plt.show()
plt.plot(threshold, precision[1:], label="Precision",linewidth=2)
```

```
plt.plot(threshold, recall[1:], label="Recall",linewidth=2)
plt.title('Precision and recall for different threshold values')
plt.xlabel('Threshold')
plt.ylabel('Precision/Recall')
plt.legend()
plt.show()
'''


# Predictions

probs0 = 1-probs
print(probs0)
predict=[]
for item in probs:
  if item >0.5:
    predict.append(1)
  else:
    predict.append(0)
pred=pd.Series(predict)
df_predictions=pd.DataFrame({"Expected": y_eval, "Predictions": pred, "Prob0": probs0, "Pr
ob1": probs})
df_predictions.to_csv('df_predictions.csv', index=False, encoding='utf-8')


# Confusion matrix

from sklearn.metrics import confusion_matrix
conf_matrix = confusion_matrix(y_eval, pred)
print(conf_matrix)
```

Appendix B
Tables

B.1 Extract from the First database

Time	Ambien	Release	Humidi	Release	P01	P02	P03	P04	PT_01	PT_02	PT_03	PT_04	Wind_D	Wind_D	Wind_S	Wind_S	TT	x	y	z	liquid_c	solid_o
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	3,563	154,7	-61,5	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	4,875	167	91,1	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	3,25	283,5	190,3	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2,875	436,1	178	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2,813	535,3	61,5	0	0	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	3	523,1	-91,1	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2	406,5	-190,3	0	1	1
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	3,625	253,9	-178	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	4,313	-130,8	-153,7	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	4,25	-100,1	227,8	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	4,438	191,3	475,8	0	0	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2,063	572,8	445,1	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2,125	820,8	153,7	0	1	0
0,01039	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	2,125	787,8	-232,2	0	1	0

B.2 Extract from the Second database

Time	Ambien	Release_	Humidit	Release_	P01	P02	P03	P04	PT_01	PT_02	PT_03	PT_04	Wind_D	Wind_D	Wind_S	Wind_S	HC	x	y	z	label
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,009	21213	21213,2	100	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,025	21213	21213,2	1000	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,001	21213	21213,2	1800	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,003	27716	11480,5	100	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	0,006	27716	11480,5	1000	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	0,074	27716	11480,5	1800	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	0,001	30000	0	100	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,006	30000	0	1000	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	0,007	30000	0	1800	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,012	27716	-11480,5	100	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,001	27716	-11480,5	1000	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,007	27716	-11480,5	1800	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	-0,019	21213	-21213,2	100	0
0,01	964,2	0,228	123,4	vertical	1,84	1,971	1,836	-0,006	-174,4	-206,9	-211,9	-17,56	230,7	241,5	2,476	2,836	0,01	21213	-21213,2	1000	0

B.3 Extract from the Third database

Time	Ambien	Release	Humidit	Purge	Sealing	P01	P02	P03	P04	PT_01	PT_02	PT_03	PT_04	TT	x	y	z	liquid_o	solid_ox
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	-3,563	-200	0	0	1	1
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	-5,188	0	200	0	1	1
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	-8,063	200	0	0	1	1
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	-0,813	0	-200	0	1	1
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	4,063	-500	0	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	3,938	-353,6	353,6	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	3,75	0	500	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	2,188	353,6	353,6	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	2,5	353,6	-353,6	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	3,188	0	-500	0	1	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	3,5	-353,6	-353,6	0	0	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	4,375	-707,1	707,1	0	0	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	4,688	707,1	707,1	0	0	0
0,098	953,4	0,183	108,5	air	none	1,516	1,431	1,377	0,039	-210,1	-201,7	-184,1	-119,1	4,875	707,1	-707,1	0	0	0

Bibliography

Aaneby J., Gjesdal T. and Voie Ø. (2021). *Large scale leakage of liquid hydrogen (LH_2) - tests related to bunkering and maritime use of liquid hydrogen.* (FFI-RAPPORT 20/03101). Available at: https://www.ffi.no/en/publications-archive/large-scale-leakage-of-liquid-hydrogen-lh2-tests-related-to-bunkering-and-maritime-use-of-liquid-hydrogen

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning, in: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*. Available at: osdi16-abadi.pdf (usenix.org)

ActiveWizards (2019), *Artificial Intelligence, Machine Learning and Deep Learning: what is the difference?* Available at: Artificial Intelligence vs. Machine Learning vs. Deep Learning. What is the difference? | by Igor Bobriakov | ActiveWizards — AI & ML for startups | Medium (Accessed: 25 January 2022).

Atkinson, G. (2020). *Pre-normative REsearch for Safe use of Liquid Hydrogen (PRESLHY): Fuel Cells and Hydrogen Joint Undertaking (FCH JU).*

Atkinson G. (2021). Condensed phase explosions involving liquid hydrogen, in*: 9th International Conference on Hydrogen Safety* (ICHS).

Aziz, M. (2021) Liquid Hydrogen: A Review on Liquefaction, Storage, Transportation, and Safety. *Energies*, 14, 5917. https://doi.org/10.3390/en14185917

Arrieta, A., Brickell, P., Camparada, V., ..., Ritlop, D. (2009). *Hazards of inert gases and oxygen depletion.* Available at: https://www.linde-gas.com/en/images/Hazards%20of%20inert%20gases%20and%20oxygen%20depletion_tcm17-13909.pdf

Atallah, S. and Allan, D.S. (1971). Safe separation distances from liquid fuel fires. *Fire Technology*, *7*, 47-56.

Baldissin, M. (2015). *Misure per prevenire la formazione di atmosfere esplosive INERTIZZAZIONE*. 1–58.

Bastani, K., Asgari, E. and Namavari, H. (2019). *Wide and deep learning for peer-to-peer lending*. In: *Expert Systems With Applications*, *134*, 209–224. https://doi.org/10.1016/j.eswa.2019.05.042

Ben-David, S. and Shalev-Shwartz, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Birol, F. (2019). The Future of Hydrogen: Seizing Today's Opportunities. *Report prepared by the IEA for the G20, 82-83, Japan*.

Brink, H., Richards, J. and Fetherolf, M. (2016) *Real-World Machine Learning*. Manning Publications. Available at: https://books.google.no/books?id=DoQAswEACAAJ.

Brownlee, J. (2021). *SMOTE for imbalanced classification with Python.* Available at: https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/ (Accessed: 25 January 2022).

Cao, X. H., Stojkovic, I., and Obradovic, Z. (2016). A robust data scaling algorithm to improve classification accuracies in biomedical data. *BMC Bioinformatics*, 1–11. https://doi.org/10.1186/s12859-016-1236-x

Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Shah, H. (2016). Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7-10).

Chinchor, N., (1992) *MUC-4 Evaluation Metrics*. 22–29. Available at: https://aclanthology.org/M92-1002.pdf

Chollet, F. (2017). *Deep Learning with Python.* Manning Publications.

Crocker, W. P. (1986). Thermal radiation hazards of liquid pool fires and tank fires. In *I. Chem. E. Symp. Ser.* (Vol. 97, pp. 159-184).

Danilov, M., & Karpov, A. (2018). A classification of meteor radio echoes based on artificial neural network. *Open Astronomy*, *27*(1), 318-325.

Davies, P. A. (1993). A guide to the evaluation of condensed phase explosions. *Journal of hazardous materials*, *33*(1), 1-33.

DNV, (2021). *Handbook for hydrogen fuelled vessels.* Available at: https://www.dnv.com/maritime/publications/handbook-for-hydrogen-fuelled-vessels-download.html

Genova, B., Ripani, L. and Silvestrini, M. (n.a.). *Esplosioni: utilizzo di modelli per la previsione degli effetti e per l'investigazione*. 3–28.

GitHub (2020a). *Precision - Recall Curve, a Different View of Imbalanced Classifiers.* Available at: Precision - Recall Curve, a Different View of Imbalanced Classifiers – Sin-Yi Chou – Data Science Enthusiast (sinyi-chou.github.io) (Accessed: 25 January 2022).

GitHub (2020b). Oversampling-Imbalanced-Data. Available at: GitHub - minoue-xx/Oversampling-Imbalanced-Data: MATLAB Implementation of SMOTE related algorithms (Accessed: 25 January 2022).

Google (2020a). *Classification: Accuracy | Machine Learning Crash Course*. Available at: https://developers.google.com/machine-learning/crashcourse/classification/accuracy (Accessed: 25 January 2022).

Google (2020b). *Classification: Precision and Recall | Machine Learning Crash Course.* Available at: https://developers.google.com/machine-learning/crashcourse/classification/precision-and-recall (Accessed: 25 January 2022).

Guru99 (2021). *TensorFlow Binary Classification: Linear Classifier example*. Available at: TensorFlow Binary Classification: Linear Classifier Example (guru99.com) (Accessed: 25 January 2022).

Hastie, T., Friedman, R. and Tibshirani, J. (2009) *The Elements of Statistical Learning*. Springer-Verlag New York. doi: 10.1007/978-0-387-84858-7.

Hydrogen Tools (2022). *Hydrogen leak detection.* Available at: Hydrogen Leak Detection | Hydrogen Tools (h2tools.org) (Accessed: 25 January 2022).

Hochgraf, C. (2009) 'APPLICATIONS – TRANSPORTATION | Electric Vehicles: Fuel Cells', in Garche, J. *Encyclopedia of Electrochemical Power Sources*. Amsterdam: Elsevier, pp. 236–248. doi:10.1016/B978-044452745-5.00863-7.

Imbalanced Learn, (2022). SMOTENC. Available at: SMOTENC — Version 0.10.0.dev0 (imbalanced-learn.org) (Accessed: 25 January 2022).

JavaTPoint (2021). *Applications of Machine Learning.* Available at: Applications of Machine Learning - Javatpoint (Accessed: 25 January 2022).

Jiao, Y. and Du, P. (2016). Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quantitative Biology*, *4*(4), 320-330.

Kriegeskorte, N. Golan, T. (2019). Neural network models and deep learning. *Current Biology*, *29*(7), R231-R236.

Linde (2021). Hydrogen, refrigerated liquid: Safety Data Sheet. Available at: https://www.lindeus.com/-/media/corporate/praxairus/documents/sds/hydrogen/liquid-hydrogen-gas-h2-safety-data-sheet-sds-p4603.pdf?la=en

Litchfield E.L. and Perlee H.E. (1965). *Fire and Explosion Hazards of Flight Vehicle Combustibles* (Technical Report AFAPL-TR-65-28).

Liu, Y., Wei, J., Lei, G., Chen, H., Lan, Y., Gao, X., ... and Jin, T. (2019). Spread of hydrogen vapor cloud during continuous liquid hydrogen spills. *Cryogenics*, *103*, 102975.

Loukas, S., (2020). *Everything you need to know about min_max normalization: a Python tutorial.* Available at: Everything you need to know about Min-Max normalization: A Python tutorial | by Serafeim Loukas | Towards Data Science (Accessed: 25 January 2022).

Machine Learning Notebook (2021). *Classification: Perceptron* Available at: https://sites.google.com/site/machinelearningnotebook2/classification/binary-classification/perceptron. (Accessed: 25 January 2022).

Malhotra, A., Carson, D., and Mcfadden, S. (2017). Blast pressure leakage into buildings and effects on humans. *Procedia Engineering*, *210*, 386–392. https://doi.org/10.1016/j.proeng.2017.11.092

Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

Nilsson, N. J. (2005). Introduction to machine learning (p. 188).

NIST (2022). *NIST Chemistry WebBook*. Available at: webbook.nist.gov/ (Accessed: 25 January 2022).

Pedregosa *et al*. (2021a), *sklearn.preprocessing.MinMaxScaler*|Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

Pedregosa *et al*. (2021b), *Compare the effect of different scalers on data with outliers*|Scikit-learn: Machine Learning in Python, JMLR 12, pp. 2825-2830, 2011.

Perry, R. H. and Green, D. W. (2008). *Perry's chemical engineers' handbook*. New York: McGraw-Hill.

Pohanish, R. P. (2008). Sittig's handbook of toxic and hazardous chemical carcinogens 5th ed. *Norwich, NY: William Andrew*, 2528.

Pritchard, D. K. and Rattigan, W. M. (2010). *Hazards of liquid hydrogen: Position paper*. HSE 2010 Research Report RR769, Sudbury: HSE Books.

Ramasamy, A., Hughes, A., Carter, N., & Kendrew, J. (2013). The effects of explosion on the musculoskeletal system. *Trauma*, *15*(2), 128-139. https://doi.org/10.1177/1460408613484683

Rana, M. A., and Mannan, M. S. (2010). Forced dispersion of LNG vapor with water curtain. *Journal of Loss Prevention in the Process Industries*, *23*(6), 768-772. https://doi.org/10.1016/j.jlp.2010.08.008

Rana, M. A., Guo, Y., and Mannan, M. S. (2010). Use of water spray curtain to disperse LNG vapor clouds. *Journal of loss prevention in the process industries*, *23*(1), 77-88. https://doi.org/10.1016/j.jlp.2009.06.003

Hooker, P., Willoughby, D., and Royle, M. (2011). Experimental releases of liquid hydrogen. Available at: https://www.h2knowledgecentre.com/content/conference505

Royle, M. and Willoughby, D., (2014). Experimental releases of liquid hydrogen *Releases of unignited liquid hydrogen* (HSE Research Report RR986). Available at: https://www.hse.gov.uk/research/rrhtm/rr986.htm

Seliya, N., & Hulse, J. Van. (2009). A Study on the Relationships of Classifier Performance Metrics. *2009 21st IEEE International Conference on Tools with Artificial Intelligence*, 59–66. https://doi.org/10.1109/ICTAI.2009.25

Shin, T. (2020). *All Machine Learning Models explained*. Available at: All Machine Learning Models Explained in 6 Minutes | by Terence Shin | Towards Data Science (Accessed: 25 Janauary 2022).

Singh, D., and Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing Journal*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Smola, A., and Vishwanathan, S. V. N. (2008). Introduction to machine learning. *Cambridge University, UK*, *32*(34), 2008.

Sowah, R. A., Agebure, M. A., Mills, G. A., Koumadi, K. M., and Fiawoo, S. Y. (2016). New cluster undersampling technique for class imbalance learning. *International Journal of Machine Learning and Computing*, *6*(3), 205-214. https://doi.org/10.18178/ijmlc.2016.6.3.599

Spoelstra, M.B. (2020). Safety aspects relating to the use of hydrogen in confined spaces. Available at: 20201014-IFV-Safety-aspects-relating-to-the-use-of-hydrogen-in-confined-spaces.pdf.

TensorFlow (2021a). *API Documentation.*. Available at: API Documentation | TensorFlow Core v2.7.0 (Accessed: 25 Janauary 2022).

TensorFlow (2021b). *tf.estimator.Estimator*. Available at: tf.estimator.Estimator | TensorFlow Core v2.7.0 (Accessed: 25 Janauary 2022).

TensorFlow (2022a). *Build a Linear Model with Estimators.* Available at: Build a linear model with Estimators | TensorFlow Core (Accessed: 25 Janauary 2022).

TensorFlow (2022b). Available at: https://www.tensorflow.org/. (Accessed: 25 Janauary 2022).

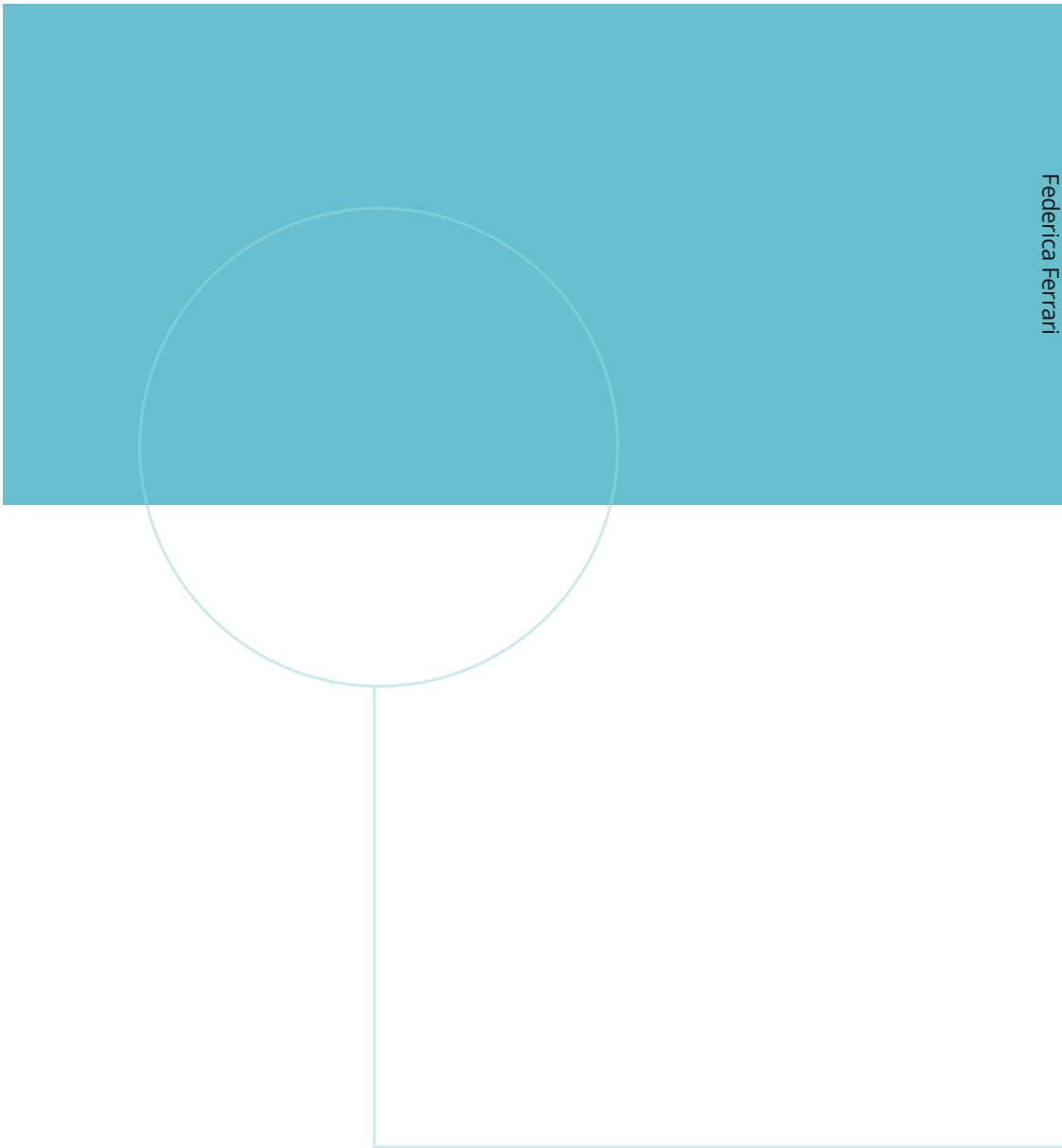
Thomas, J. K., Eastwood, C., & Goodrich, M. (2015). Are unconfined hydrogen vapor cloud explosions credible?. *Process Safety Progress*, *34*(1), 36-43.

TNO (2005) Methods for the calculation of Physical Effects due to releases of hazardous materials (liquids and gases) (“Yellow Book”).

Tutorials Point (2018). *Tensorflow Tutorial.* Available at: MySQL (tutorialspoint.com)

Verfondern, K., Cirrone, D., Molkov, V., Makarov, D., Coldrick, S., … Kuznetsov, M. (2021). *Handbook of hydrogen safety: Chapter on LH_2 safety*. Available at: Microsoft Word - 2021-01 PRESLHY_ChapterLH2-v3.doc (hysafe.info)

Wade, C., Spearpoint, M. J., Bittern, A., and Tsai, K. (2007). Assessing the Sprinkler Activation Predictive Capability of the BRANZFIRE Fire Model. *Fire Technology 43*(3), 175–193. https://doi.org/10.1007/s10694-007-0009-5



Federica Ferrari

NTNU
Norwegian University of
Science and Technology