NTNU

Norwegian University of
Science and Technology

Data Analytics for Chemical Process Risk Assessement: a Representative Case Study to Support Safe Handling of Hazardous Substances

Riccardo Solini

Reliability, Availability, Maintainability and Safety (RAMS)
Submission date: February 2018
Supervisor: Nicola Paltrinieri, MTP

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

RAMS
Reliability, Availability,
Maintainability, and Safety

Data Analytics for Chemical Process Risk Assessment: a Representative Case Study to Support Safe Handling of Hazardous Substances

Riccardo Solini

February 2018

MASTER THESIS
Department of Mechanical and Industrial Engineering
Norwegian University of Science and Technology

Supervisor 1: Nicola Paltrinieri
Supervisor 2: eng. Sarah Bonvicini

Abstract

Potential of data generation has exponentially increased nowadays. Through the World Wide Web, for instance, information of any kind are collected and stored in databases. In the industrial sector, a huge amount of data is set to be collected by the so-called industry 4.0. Also the Seveso III Directive, which contains the rules for preventing and fighting major accidents in establishments handling dangerous substances, advances the need of monitoring and analyzing data in order to improve the safety management system of the plants. In this context, the discipline of machine learning is suggested. It consists in methods through which computers automatically retrieve knowledge from data. On the basis of this, they are able to support or take decisions.

However, data are still not exploited as they should be and opportunities to learn are lost. It is necessary to improve the use of such data and increase our knowledge.

This work suggests an approach to analyze heterogeneous data about past accidents in process industries and extract important information to support safety-related decision making. The knowledge retrieved should help improving the evaluation of the risk picture, by predicting the consequence on humans.

The machine learning tool used to analyze the data is the open-source library TensorFlow. Through its use, different models are built - a linear model, a deep neural network model and a combination of the two. The models, on the basis of specific inputs, may be able to make predictions about the number of people killed or number of people injured. The tuning of the model's parameters is carried out using the past accident data contained in the MHIDAS database as training data set. To evaluate the performance of the model another data set is needed. For this reason, a new database has been built. Ammonia plants, which fall under the Seveso III Directive, have been taken into consideration. Accidents occurred in these establishments - or in similar sections of other plants - have been take into account, investigating public accident databases, books, articles and journals. Their data have been collected and registered in a common database using MHIDAS keywords. A set of simulations have been performed not only to validate the models, but also to identify their limitations. A good model for accident prediction needs to be able to predict rare events - *i.e.* the ones with the highest number of people killed or injured. This condition is obtained if the value of the statistic metric "recall" is high. For this reason, the results returned by the simulations have been analyzed considering the value of the area under the curve precision recall as a priority. From this, it is possible to understand if the value of recall can be of interest or not. The results obtained have shown a common trend and a type of model that, in general, have had better prediction skills than the other ones.

Contents

Chapter 1

Introduction

1.1 Background

We are in the big data era (Murphy, 2012). The volume of data generated nowadays is, indeed, enormous. From business, society, science and engineering, medicine, and almost every other aspect of human life, all kind of information is collected and stored in databases (Jiawei Han, 2012). In the industrial field, an increasing volume of data is set to be collected by the new generation of industry, the so-called industry 4.0 (Paltrinieri et al., 2017).

At the same time, artificial intelligence (AI) has become an important field, with many practical applications. Intelligent software to automate routine labors or to give support in decision making are now possible to use and research topics aim to make them even more reliable. Developing methods through which AI systems - *i.e.* essentially computers - can automatically learn from data and make intelligent decisions based on the knowledge retrieved from them, is the natural consequence of this whole context. That is how the machine learning discipline was born (Greenberg et al., 2012).

The Seveso III Directive (2012/18/EU) - which provides the rules and the indications for preventing major accidents to occur in establishments handling hazardous substances, *i.e.* the Seveso sites - advances the necessity of collecting and analyzing data. For instance, it suggests, for the first time since it was issued in 1982, the use of indicators - such as safety performance indicators (SPIs) - for the ongoing assessment of compliance, with the directive, of the establishment's safety management system. Another important point stressed in the 2012/18/EU, is about learning from past accident or near misses. In order to gain knowledge from these events, all the information regarding them has to be collected, without omitting any detail. Moreover, an efficient system to share it is required, ensuring that everyone can have access to the information.

However, the quantity of data generated is not exploited as it should be. Data are collected but not ably analyzed and so opportunities to learn are lost. In this data-rich information-poor situation, it is necessary to use the advanced tools available nowadays and retrieve information from data (Jiawei Han, 2012).

1.2 Objectives

Today's risk analysis challenges consists essentially in 5 points (Paltrinieri and Comfort, 2018).

1. Dinamicity - how to continuously update and improve risk assessment, keeping track of the constant evolution of the industrial system.

2. Cognition - how to learn from past accidents to improve risk assessment, in order to avoid accident repetition.

3. Data processing - how to process the huge quantity of data coming from the industrial system and extract useful risk information.

4. Emergence - how to be prepared for emerging risk, not known before, such as the one deriving from new technologies.

5. Usability - how to provide a real support to the industry.

This work focuses mainly on points 2 and 3.

The purpose is to find an efficient way to analyze heterogeneous data of past accidents, occurred in process industries, and extract, from them, important information to support safety-related decision making. In other words, the information retrieved should help improving the evaluation of the risk picture for plants handling hazardous substances.

This answers both the challenge of "Cognition" - since data from past accidents are taken into account - and of "Data processing".

1.3 Approach

The idea is to consider sites handling hazardous substances, falling under the Seveso III Directive. With respect to that, past accidents from ammonia plants - or plants containing similar

sections - have been taken into consideration. Their features have been obtained investigating accident reports from different sources and have been collected in a common database using keywords.
The database have been used for the evaluation of different models - *i.e.* a linear model, a deep neural network model and a combination of the two - trained with the data from MHIDAS accident database (HSE, 1999). The machine learning tool used to perform the simulations is the open source library TensorFlow. The results returned have been analyzed and compared in order to detect the model that best predicts the consequences on human - in terms of people killed or people injured.

1.4 Outline

The thesis is structured in seven chapters. Chapter 1 is an introduction of the topics discussed in this work. It presents the problem to investigate, it describes the objectives established and it introduces the approach used to meet the tasks. Chapter 2 is about risk assessment in Seveso sites. Here, the well known Seveso disaster is described and the Seveso III Directive, currently in force, is introduced. The chapter continues with a digression about how the classical risk assessment could benefit from a dynamic approach, introducing the dynamic risk analysis (DRA). Finally, a note about the importance of having an efficient system to exchange information about past accidents or near misses. Chapter 3 is about the creation of the ammonia plant accident database. Data mining, as the tool which provides the necessary steps to manage data in order to make them suitable for the simulations, is first described. Next, the accident sources used to collect data are presented and, finally, the ammonia plant accident database is introduced and its features illustrated. Chapter 4 is about the methods used to reach the thesis objectives. The machine learning tool TensorFlow is first described, the models used are introduced and the information for computing and evaluating the simulations are indicated. In chapter 5 the results are highlighted. Chapter 6 contains the elaboration and the discussion of the results, highlighting the limitations of the work. Finally, chapter 7 sums up the topics discussed and highlights the main results obtained.

Chapter 2

Risk Assessment in Seveso Sites

2.1 Introduction

On Saturday 10 July 1976 an accident occurred in a plant manufacturing 2,4,5-triclorophenol (TCP) in Meda, near Seveso. A runaway reaction occurred, causing a pressure increase in the reactor and the rupture of the safety disk. This led to the release, into the atmosphere, of a cloud containing 2,3,7,8-tetrachlorodibenzodioxine (TCDD), a very toxic chemical. Between the 220000 people exposed, 250 developed the skin disease chloracne and 450 were burned by caustic soda. Even though no one was killed, this is one of the most famous of all chemical plants accidents, which changed the approach to risk assessment forever. As a matter of fact, it urged the enactment, in Europe, of the so-called Seveso Directive, whose focus is on preventing major accidents to occur and ensuring adequate protection to the citizens, the communities and the environment. The industrial sites considered, which fall under the directive, are establishments handling big quantity of dangerous substances, called Seveso sites, able to cause major accidents. Nowadays, the Seveso III directive is in force (2012/18/EU), refining the previous Seveso II Directive (96/82/EC).

The chapter firstly describes the Seveso accident, presenting how TCP was usually manufactured and highlighting, then, what went wrong, showing causes and consequences of the disaster. The Seveso III Directive is described next and its main points are introduced. The dynamic risk analysis, as a new approach to risk assessment, which allows to take into account the dynamicity of the industrial environment and to exploit the huge quantity of data collected nowadays, is then presented. Finally, a note about learning from past accidents and how important it is to have an efficient system to exchange information about them is reported.

2.2 The Seveso Accident

2.2.1 Manufacturing Process of TCP

The site at Seveso was operated by Indsutrie Chimiche Meda Società Azionaria (ICMESA) (Lees, 2005). The manufacturing process of 2,4,5-trichlorophenol (TCP), described below, is referred to the one used by the ICMESA factory, at the time of the accident.

The plant used a discontinuous total batch process, which consisted in 2 stages, as shown in figure 2.1.

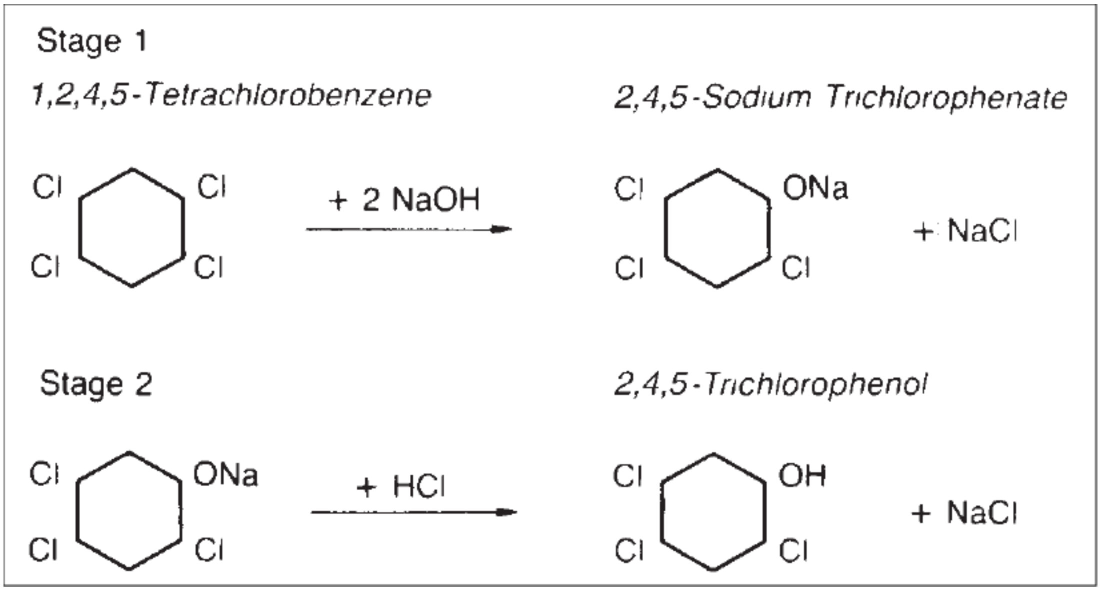

Figure 2.1: Reaction scheme for production of TCP (Lees, 2005).

The first stage consistsed in an alkaline hydrolysis, using caustic soda, of 1,2,4,5-tetrachlorobenzene (1,2,4,5 TCB) at temperatures between 140 to 170 °C and at atmospheric pressure (ARIA, 2018), forming sodium 2,4,5-trichlorophenate. The second one involved the acidification, using hydrochloric acid, of the mixture just formed, to produce TCP.

The installation consisted of a 10000 m^3 Cr-Mo-Ni alloy vessel, equipped with a stirrer (ARIA, 2018). The reactor diagram is shown in figure 2.2.

The reactor was equipped with different safety features (ARIA, 2018):

- an oversized condenser for rapid cooling;

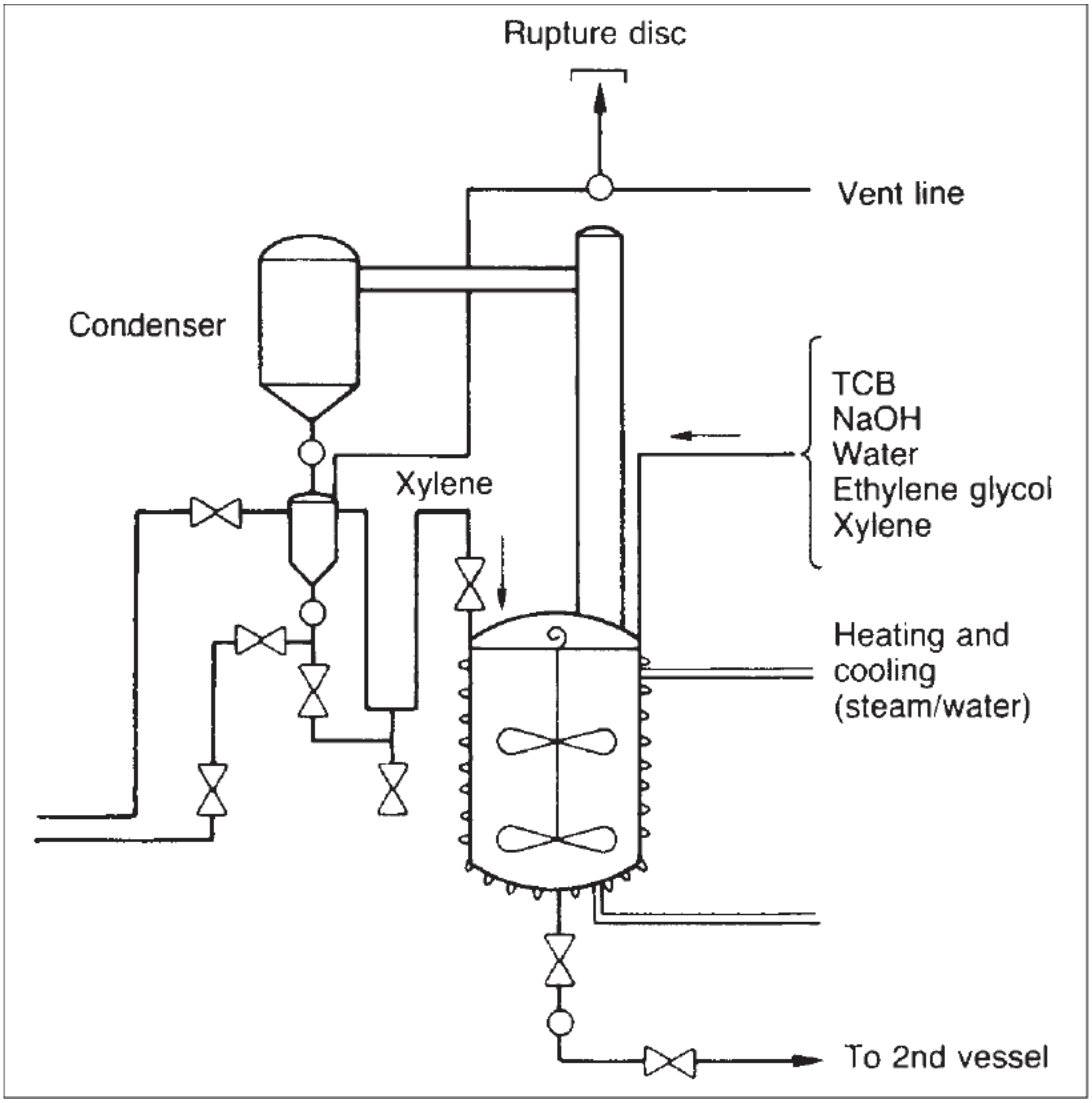

Figure 2.2: Reactor system at Seveso (Sambeth, 1983b).

- a 3000 litre reserve of water to flood the contents of the reactor and to rapidly cooled it to avoid temperature increasing;
- a rupture disk - whose tare pressure was set at 3,8 bar - venting direct to atmosphere (Lees, 2005).

Exothermic reactions were not expected below 230 °C. For this reason, the reactor had a jacket with steam flowing in it at 12 bar. At this pressure, its saturation temperature is 188 °C (Lees, 2005) and, consequently, the reaction mixture could not be heated beyond 180 °C (Sambeth, 1983a).

However, the controls on the reactor were relatively primitive. For instance, it was not equipped with a temperature alarm (Lees, 2005) and the 3000 litre of water to rapidly cooled down the system could not be opened automatically, only manually (ARIA, 2018). Moreover, no catchpot to collect possible discharges was provided (Kletz, 2001).

At the start of the operation - *i.e.* at the beginning of the first stage - the following substances, in the indicated quantity, were fed into the reactor (Sambeth, 1983a):

- 1,2,4,5 TCB, 2000 kg;
- NaOH, 1050 kg;
- ethyleneglycol, 3300 kg;
- xylene, 600 kg.

The ethyleneglycol fed was used as solvent (Sambeth, 1983a).

During the reaction, water was produced. For this reason, xylene was inserted in the system: this way, mixing with it and producing the azeotropic mixture, water removal was promoted (Sambeth, 1983a). As a matter of fact, the reactor was equipped with a condenser that separated and sent away the water, feeding back the xylene to the reactor (ARIA, 2018).

In this stage, the initial mixture is heated at 170 °C using the steam flowing in the jacket of the reactor. The reaction is carried out at this temperature for 6-8 hours (Sambeth, 1983a). At the end of the reaction, the entire xylene and part of the ethyleneglycol were vacuum distilled to be reused during the subsequent synthesis. The mixture is then cooled down to 50-60 °C adding water (ARIA, 2018).

Stage 1 lasted 11 to 14 hours. The duration of every step of this stage is listed below (Sambeth, 1983a):

- charging: 1 hour;

- reaction: 6-8 hours;
- vacuum distillation of the mixture: 3-4 hours;
- addition of water: 15 minutes.

The second stage, as already mentioned, was about converting the sodium 2,4,5-trichlorophenate, obtained in stage 1, in TCP. This was done operating in an acid environment, using an aqueous solution of hydrochloric acid (ARIA, 2018). The crude TCP is finally purified by fractional distillation (Sambeth, 1983a).

The conventional process - normally used in industry - for the synthesis of TCP was different than the one used in the Seveso plant. The solvent used was methanol, not ethyleneglycol, and the operating pressure was 20 bar. The process modification was introduced by Givaudan, ICMESA‘s parent company. Givaudan required TCP for making the bacteriostatic agent hexachlorophene, and, with the conventional process, TCP contained impurities unacceptable for this application (Lees, 2005).

In the reaction, the formation of 2,3,7,8-tetrachlorodibenzodioxine (TCDD) as by-product is unavoidable (Lees, 2005). At normal operating conditions just described - *i.e.* at 170 °C -, the quantity produced is really low: it would be unlikely to exceed 1 ppm of TCP (Lees, 2005). However, TCDD can be produced in important quantities under high temperature conditions (ARIA, 2018): this is exactly what happened in the well known Seveso accident, described next.

2.2.2 The Accident

On Friday 9 July 1976 the TCP production began. It started exactly at 4 p.m., *i.e.* 10 hours later than the usual time (ARIA, 2018). At the time, Italian law required the plant to shut down for the weekend, even though it was in the middle of a batch (Kletz, 2001). The end of the shift of the following Saturday morning - the 10th of July 1976 - was at 6 a.m., so the required 14 hours to complete stage 1 were available (Sambeth, 1983a).

In a normal operation, just before the end of the shift, almost 50% of the ethyleneglycol used would have been distilled. However, in the morning of the 10th of July 1976, only 15% was actually separated from the reaction mixture (Sambeth, 1983a). At about 5 a.m., the operator interrupted the distillation. Fifteen minutes later also the strirring process was stopped (Sambeth, 1983a). Finally, also the temperature recorder was switched off and, according to the Italian law previously mentioned, the installation was shut down for the weekend and it remained unsupervised. The last temperature registered was 158 °C. Thus, the operator had no reason to believe that it would have been dangerous to shut down the reactor and leave it without supervision (Sambeth, 1983a). As a matter of fact, as stated before, it was believed that the expected

temperature at which an exothermic reaction would have occurred was around 230 °C.
Approximately after 6 hours and 30 minutes from the end of the last shift - precisely at 12.37 p.m. of the 10th of July 1976 (Sambeth, 1983a) - the rupture disk broke and a reddish cloud containing TCDD, one of the most toxic chemicals known (Lees, 2005), was released into the atmosphere (ARIA, 2018). The gas emissions lasted about 20 minutes. Luckily, a maintenance foreman was passing by and, hearing the noise produced by the discharge, opened the cooling water supply cooling down the reaction mixture (Lees, 2005). If he had not done so, the discharge would have been greater (Kletz, 2001).
In the meantime, the temperature recorder was switched on again, showing that the temperature of the reactor‘s contents was above 200 °C (Sambeth, 1983a).

2.2.2.1 The Causes

According to Sambeth (1983a), the information known at the time of the accident were not enough to be able to predict the disaster. Even though similar incidents already occurred in the past, he affirmed that the results obtained during the investigations proved the "simultaneous and unforeseeable occurrence of several factors, most of which were unknown prior to the accident".
After detailed investigation following the accident, it was discovered that, at around 180 °C, weak exothermic reactions started. However, their effect began to be important only after 190-200 °C (Sambeth, 1983a). The reactor mixture in the ICMESA plant in Seveso, when left unsupervised for the weekend stop, was at a temperature of 158 °C. Consequently, it was really difficult to understand how the reactor could reach a temperature high enough to start these exothermic reactions.
The answer came from the study of the thermal aspects of the whole installation. The temperature situation of the reactor, once left unsupervised, is shown in figure 2.3.

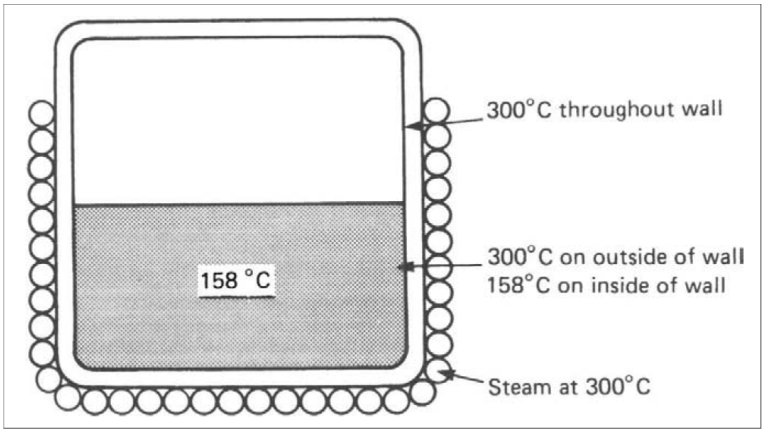

Figure 2.3: Temperature situation of the Seveso reactor left unsupervised (Kletz, 2001).

The steam turbine feeding the jacket of the reactor was on reduced load and became superheated, at a temperature of about 300 °C (Lees, 2005). The temperature of the liquid stayed at around 160 °C - the liquid boiling point (Kletz, 2001). Thus there was a temperature gradient in the wall of the reactor in correspondence of the wet part: around 160 °C in the inner wall and around 300 °C in the outside wall. However, the wall in the upper part of the reactor - *i.e.* the one not touched from the liquid - was at 300 °C right through (Kletz, 2001). Once the steam was isolated and the stirring was stopped, the wall in the wet part of the reactor dropped to the temperature of around 160 °C, while the wall in the upper part, which remained hotter, heated the upper few centimetres of the surface liquid. It reached a temperature of around 190 °C and the weak exothermic reactions started. Because of this localized overheating, temperature began to increase and the runaway known from literature - *i.e.* the exothermic reactions at 230 °C - occurred. This, led to a rise in pressure and, finally, to the rupture of the safety disk (Kletz, 2001).

Although Sambeth (1983a) stressed the unpredictability of the accident, Kletz (2001) states that the runaway could be avoided if:

- legistlators had not drafted laws which did not give the freedom to complete a batch before the weekend;
- the batch was not stopped at an unusual step;
- a well done hazard and operability study (HAZOP) had been performed on the design.

Esepcially with a HAZOP, the rise of the temperature of the steam supplying the reactor, when the turbine was in low load, and, consequently, the need of measuring its temperature, would have been brought to light.
Finally, it is important to underline that, if a catchpot to collect the discharge from the rupture disk would have been installed, the release would have not affected the surrounding area and people, and the consequences, described next, would have not existed (Kletz, 2001).

2.2.2.2 The Consequences

The toxic cloud spread to the southeast of the plant, impacting a farming area (ARIA, 2018). Once the situation with the installation was under control, the managers of the plant visited the houses close by, informing the people living there not to consume anything from the garden (Lees, 2005). However, the nature of the cloud was not directly made public and, at first, no evacuation plan was predicted. Finally, 14 days later - *i.e.* on the 24th of July -, after a general strike called by the ICMESA work force, the arrest of the production director and the general manager of ICMESA and a long series of meeting (Lees, 2005), an evacuation plan was studied and the prohibition to consume food produced in the impacted areas was extended (ARIA, 2018).

Site clean up started six months after the accident and lasted almost 5 years. The plant was destroyed in 1982 and, in the site which hosted it, is now a sport complex and a natural park (ARIA, 2018).
The area of land contaminated was about 17 km^2 and the one made uninhabitable was about 4 km^2 (Kletz, 2001). Around 220000 people were exposed to dioxin (ARIA, 2018): about 250 people developed the skin disease chloracne and about 450 were burned by caustic soda (Kletz, 2001). One of the things that worried the people affected the most was about the long lasting health effects. Different studies, such as the one proposed by Professor Pesatori from Milan University, 17 years after the accident, proved that the rate of cancer in the contaminated areas is not higher than normal, except for 2 thyroid cancers (ARIA, 2018).

It is not a fact that the accidents with many people killed - or injured - or with substantial damages are the ones who can teach lessons to be learned the most. For instance, the Seveso accident, even though no one was killed, is one of the best known of all chemical plant accidents (Kletz, 2001), which changed the approach to risk assessment forever (Paltrinieri and Reniers, 2017). Neither the residents or the authorities in Seveso were aware of the real hazard coming from the plant or about the chemical substances used. There was a real problem about the risk communication, which also led to the incapability of taking appropriate decision when the disaster occurred (ARIA, 2018). This, and the need to ensure protection to the citizens and the environment around industrial sites handling hazardous substances, led to the enactment of the so called "Seveso Directive" (82/501/EEC). Throughout the years, different modifications to the directive have been done, and, nowadays, Seveso III Directive (2012/18/EU) is in force, amending and subsequently repealing Seveso II Directive (96/82/EC).

2.3 The Seveso III Directive

The European Parliament and the Council of European Union, on the 4th of July 2012, drafted the Seveso III Directive (2012/18/EU). As the previous Seveso Directives - *i.e.* 82/501/EEC (the Seveso Directive) and 96/82/EC (the Seveso II Directive) -, the 2012/18/EU aims to prevent and reduce the risk of major accidents and to enable the necessary steps to be taken to limit the consequences thereof. Thus, it ensures that a high level of protection for citizens, communities and the environment, throughout the Union, is provided. As already mentioned, it replaces Seveso II Directive, strengthening the level of protection - introducing new guidelines and refining old ones - and reducing the unnecessary administrative burdens, in order to simplify its implementation (2012/18/EU).
The directive is divided in 34 articles, reported in table 2.1, and 7 annexes, listed in table 2.2.

Articles	
Article 1	Subject matter
Article 2	Scope
Article 3	Definitions
Article 4	Assessment of major-accident hazards for a particular dangerous substance
Article 5	General obligations of the operator
Article 6	Competent authority
Article 7	Notification
Article 8	Major-accident prevention policy
Article 9	Domino effects
Article 10	Safety report
Article 11	Modification of an installation, an establishment or a storage facility
Article 12	Emergency plans
Article 13	Land-use planning
Article 14	Information to the public
Article 15	Public consultation and participation in decision making
Article 16	Information to be supplied by the operator and actions to be taken following a major accident
Article 17	Action to be taken by the competent authority following a major accident
Article 18	Information to be supplied by the Member States following a major accident
Article 19	Prohibition of use
Article 20	Inspections
Article 21	Information system and exchanges
Article 22	Access to information and confidentiality
Article 23	Access to justice
Article 24	Guidance
Article 25	Amendment of Annexes
Article 26	Exercise of the delegation
Article 27	Committee procedure
Article 28	Penalties

Article 29	Reporting and review
Article 30	Amendment of Directive 96/82/EC
Article 31	Transposition
Article 32	Repeal
Article 33	Entry into force
Article 34	Addressees

Table 2.1: Seveso III Directive articles.

Annexes	
Annex I	Dangerous substances
Annex II	Minimum data and information to be considered in the safety report referred to in Article 10
Annex III	Information referred to in Article 8(5) and Article 10 on the safety management system and the organisation of the establishment with a view to the prevention of major accidents
Annex IV	Data and information to be included in the emergency plans referred to in Article 12
Annex V	Items of information to the public as provided for in Article 14(1) and in point (a) of Article 12
Annex VI	Criteria for the notification of a major accident to the Commission as provided for in Article 18(1)
Annex VII	Correlation table

Table 2.2: Seveso III Directive annexes.

Table 2.1 and table 2.2 show the object of every article and every annex as they are reported in the 2012/18/EU.

As it is often reminded in the directive, one of its focuses is the collection and the exchange of information between Member States, making the authorities and the public aware about the dangerous substances and the potential dangers coming from the so-called "Seveso sites" - *i.e.* industrial sites handling dangerous substances (Paltrinieri and Reniers, 2017). For instance, operators have to draw up a major-accident prevention policy (MAPP) setting out their overall approach and measures for controlling major-accident hazards. The document has to be sent to the competent authority when required by law and it should be reviewed at least every five years. All the information about the MAPP can be found in Article 8.

Moreover, for sites handling significant quantities of dangerous substances - *i.e.* the "upper-tier establishments" (information about the categorization of the establishments can be found in Annex I) - a safety report needs to be provided, from the operator, to the competent authority.

Information about the safety report is reported in Article 10.
Directive 2012/18/EU focuses also in sharing and exchanging information about major accidents already occurred, in order to prevent unwanted events of similar nature to occur again. It stresses the importance of collecting and exchanging data also about near misses, not only accidents. Article 16, 17, 18, together with Annex VI, and Article 21 are about this topic.
It is finally important to mention that Seveso III underlines the importance of an appropriate safety management system to implement the MAPP. In the last decade, as a way to assess and control risk, focus has been put in evaluating and monitoring early deviations through appropriate indicators (Paltrinieri and Reniers, 2017), whose use and analysis is promoted by the extensive collection of data carried out in the industry nowadays. One of the points of Annex III of the directive is about "monitoring performance". Here, the use of indicators such as safety performance indicators (SPIs) and/or other relevant ones is suggested, in order to monitor the performance of the safety management system. How such suggestions has been received in the the EU member and associate countries can be found elsewhere (Paltrinieri and Reniers, 2017).

2.4 Improvement of the Classical Risk Assessment Approach: the Dynamic Risk Analysis

The accident that occurred in Seveso on Saturday 10 July 1976 changed the approach to risk assessment forever (Paltrinieri and Reniers, 2017). Article 10 and Annex II of the Seveso III Directive are about the safety report and the minimum data and information to be considered in it. There, it is stated that risk analysis has to be carried out to identify the possible major-accident scenarios and its consequences and, thus, ensuring protection and intervention to limit them, for human health and the environment. The safety report should be reviewed and updated every five years, unless the competent authority requests it, major changes are performed or a major accident occurs in the establishment. Quantitative risk assessment (QRA) is usually carried out to comply with the Seveso III Directive (Paltrinieri and Reniers, 2017). Villa et al. (2016) listed some of the achievements of the QRA in the last years, in line with the accomplishments for risk analysis identified by Greenberg et al. (2012). However limitations of this approach have been identified (Villa et al., 2016). QRAs are often developed in design phases and, consequently, they are referred to that specific life cycle phase of the plant. As a matter of fact, available risk reducing measures are different in the design phase compared to the operational phase (Falck et al., 2015). Furthermore, an industrial system - as its safety measures - constantly evolves, highlighting new conditions. The QRA's staticity and inability of updating the risk picture has suggested a new approach, able to consider the dynamicity of the systems: the dynamic risk analysis (DRA). DRA is based on the iteration of risk assessment, as represented in figure 2.4.
The iteration may be performed in three different levels:

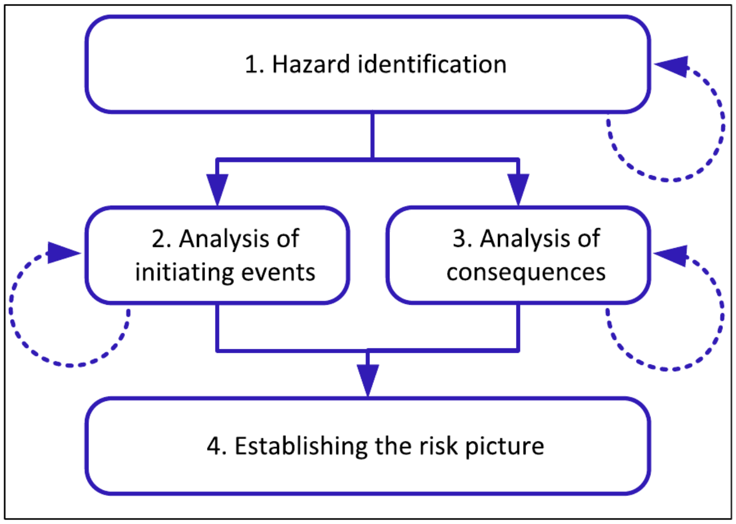
1. Hazard identification
2. Analysis of initiating events
3. Analysis of consequences
4. Establishing the risk picture

Figure 2.4: Risk flow-chart and suggested iterations (Paltrinieri and Reniers, 2017).

- identification of accident scenarios (hazard identification);
- analysis of initiating events, studying the likelihood of occurrence of these events;
- analysis of consequences, studying the consequences of the above-mentioned scenarios.

Each of these steps can be developed in different ways. Paltrinieri and Khan (2016) illustrate several methods to do that. In particular, it is possible to focus on a selection of them, in the perspective of dynamic risk analysis for Seveso sites (Paltrinieri and Reniers, 2017).

The general challenge is to decrease the probability of high impact, low probability (HILP) events, which has the potential of having catastrophic losses, also in terms of human life (Paltrinieri and Reniers, 2017). To do so, risk analysis is iterated - in the different levels mentioned before - integrating new risk notions and lessons learned and monitoring and analyzing the so-called "Small things" - which can be defined as deviations from normal/optimal conditions (Paltrinieri and Khan, 2016). Extreme accidents are the results of a combination of these early warnings. Consequently, focusing in preventing small things from happening, should consent to break this chain of events and decrease the probability, or even prevent, HILP to occur (Paltrinieri and Khan, 2016). Nowadays, a huge amount of data is collected in the industry and many indicators, to monitor and evaluate these early deviations, are used - as suggested in the Seveso III Directive (2012/18/EU). These indicators could be the perfect starting point to iterate the risk analysis and, thus, to compute a DRA.

All in all, risk analysis is constantly improving and new methods to refine it have been studied. Dynamic risk analysis, even though it still has evident limitations as pointed out by Paltrinieri and Reniers (2017), is a promising approach, able to take into account the dynamicity of the industrial environment and to exploit the huge amount of information coming from data collected - for instance indicators. The study should focus on refining the methods and on how to analyze these data, in order to obtain relevant information for safety-related decision making.

2.5 Learning from Past Accidents

As already mentioned, Article 16, 17, 18, together with Annex VI, and Article 21 of the Seveso III Directive give indications, among other things, about lessons to be learned from past accidents or near misses and how to share the information deriving from them. According to Kletz (2001), people do learn from experience of accidents and rarely they repeat the same mistakes again. The issue is about passing the knowledge, without omitting any detail. It is essential to share all the information about them, in order to allow everyone to learn the lesson. Thus, for such a purpose, an efficient system to share the information is required, in order to facilitate the process of acquiring the knowledge and learning the lesson. This is one of the points promoted by the Seveso III Directive too. Moreover, well-written reports - *i.e.* clear, well structured and with all the information needed - are important. According to Kletz (2001), investigating teams should report not only the information needed to make their recommendations, but everything they collect. This way, the readers, analyzing the report, may recognize additional causes or recommendations and draw different conclusions.
All in all, it is important to facilitate the exchange and the comprehension of the information about past accidents - but also near misses of particular technical interest, without omitting any detail. This way, it is easier to access data, analyze them and learn from past accidents.

The work in this thesis suggests an approach to analyze heterogeneous data from chemical process industries and to extract important information. More in detail, it is about collecting data from past accidents and analyze them in order to obtain indications for handling hazardous substances. Thus, it suggests a way to learn lessons from past unwanted events, analyzing the data collected. It is clear that, the more the information available - and the better the reports are written - , the better - and the more - methods like this can be developed and be of real support to the industry.

Chapter 3

Ammonia Plant Accident Database Creation

3.1 Introduction

One of the goals of the work is to analyze in an innovative way heterogeneous data from chemical process industries, in order to support and improve safe handling of hazardous substances. For such a purpose, ammonia plants have been taken into consideration as an example. To do so, the creation of an accident database has been performed. Events in ammonia plants and in plants containing similar sections (for instance, desulfurization sections to clean the raw gas before entering the reformer reactor, reforming sections to obtain syngas necessary for the ammonia synthesis and storage tanks containing relevant substances in relevant physical and chemical conditions) have been considered. Different information from different sources, such as public accident databases, books, articles and journals, have been collected for this purpose. It is important to mention that not all the accident reports consulted were structured the same way and with the same level of detail; for this reason, it has not always been possible to extrapolate all the information needed.

In the chapter, data mining, as the method that provides the tools to collect and manage heterogeneous data in an efficient way, is introduced, the accident sources are presented and the database created is showed and described.

3.2 Data Mining

Nowadays, the data volume generated and collected every day is enormous. For instance, through the World Wide Web it is possible to find and share any kind of information, while all the researches made, all the websites visited and every "click" in general, is recorded in databases. In the industrial field, an increasing volume of data is set to be collected by the so called "industry 4.0". Wireless sensors, for example, are now low cost instruments, often redundant in the systems; the more the sensors, the more the data collected and stored (Paltrinieri et al., 2017).
In general, these days, the number of databases, collecting big quantities of data from business, society, science and almost every other aspects of human life, has consistently increased, and their number is still growing. According to Ian H. Witten (2011), it has been estimated the amount of data stored in the world's databases doubles every 20 months and there is a growing gap between the generation of data and their understanding. In this data-rich but information-poor situation, it is necessary to find a way to retrieve knowledge and, therefore, to exploit the data in an efficient way. Data mining comes to the aid of this.
Data mining, or "knowledge discovery from data" (KDD), is the subject that provides the tools to discover knowledge from data and the ability to use it (Jiawei Han, 2012). In general, its tasks can be classified in two categories (He, 2015):

- "descriptive", whose aim is to characterize and describe a target data set and its properties;
- "predictive", to make predictions on future data performing induction and inference on current data.

In other words, data mining consists in finding and describing patterns in large amounts of data. However, not all the patterns mined are actually of practice interest to a given user. According to Jiawei Han (2012), a pattern is interesting if it is easily understood by humans, valid on new or test data with some degree of certainty, potentially useful, and novel. A pattern can also be considered interesting if it confirms the hypothesis that the user wants to confirm. Either objective or subjective measures of pattern interestingness exist, in order to assess systematically if the pattern found is of any use or not. Generally speaking, from a practical point of view, it is possible to say that the patterns found must lead to some advantages to be worthwhile, usually an economic one (Ian H. Witten, 2011).
The process of data mining consists essentially in seven steps, as indicated below (Jiawei Han, 2012).

1. Data cleaning. Missing values, noise and inconsistencies contribute to inaccurate data and, thus, to inaccurate final results. The step's aim is to avoid this, filling in missing values, smoothing out the noise and detecting outliers, *i.e.* eliminating inconsistent data.

2. Data integration. It consists in merging data from different data sources. It is a challenging step, because it is necessary to deal with heterogeneous data.

3. Data selection. In this step, only relevant data for the analysis are selected. It is especially helpful when huge data sets are involved, because the time spent to analyze them could be too long and the analysis not feasible. In this perspective, it is an useful step if correctly done: the reduced data set will be more efficient and, anyway, the analytical result will be almost the same.

4. Data transformation. Data are transformed or consolidated into appropriate forms for mining. A common schema to register all data to be analyzed has to be found, in order to improve the accuracy and speed of the whole process, and to make the patterns found in the following steps easier to understand.

5. Pattern discovery. Data are here analyzed and the patterns describing them are retrieved.

6. Pattern evaluation. In this step, the patterns extracted in step 6 are evaluated and the interesting ones are identified.

7. Knowledge presentation. Finally, the mined knowledge, by means of interesting patterns, is presented to the users.

Steps from 1 to 4 are the so called preprocessing steps. Preprocessing data is very important. In real-world databases, it is common to have low quality data - they are inaccurate, incomplete or inconsistent - and this leads to low quality results. Improving the data quality contributes to improve the overall quality of the patterns mined.

An important activity not mentioned in the seven reported above, is about the data collection. As a matter of fact, it is essential to use relevant data or, in general, data which help to achieve the goal of the analysis, in order to improve the performance of data mining. This step is required only if the users want to build their own database to reach their own objectives. It is also possible to mine data from pre-existing databases.

3.3 Accident Sources

In order to build the accident database to analyze, since no pre-existing databases about ammonia plant accidents only exist, the data collection step is necessary. In this chapter, the accident sources investigated to obtain and collect data are presented and described.

3.3.1 eMARS

The eMARS database is an accident and near miss database, open to the public, established by the EU's Seveso Directive 82/501/EEC in 1982. The name is an acronym and it stands for "Major Accident Reporting System": as a matter of fact, it was previously called "MARS", then was later renamed "eMARS" after going online (MAHB, 2017).

EU, EEA, OECD and UNECE countries (under the TEIA Convention) provide reports to the Major Accident Hazards Bureau (MAHB) of the European Commission's Joint Research Center (JRC), about chemical accidents and near misses. These data are included into the eMARS database directly from the recognized authority reporting the event. The reports are compulsory for EU Member States when a major accident - as defined by Annex VI of the Seveso III Directive (201218/EU) - occurs in a Seveso establishment. For all the other countries previous listed, reporting the event is voluntary (MAHB, 2017).

The goal of the database is to collect all possible information about accidents, near misses and so on, and sharing them with everyone, in order to learn from these information and to use them as an instrument to prevent future dangerous events. To do so, it has been decided not to show company names and location. This way, reporting the event in an accurate and detailed way is supported (the company will not be judged from anyone) and the focus of the reader goes completely on the accident information and not on the company or on the country associated with it.

3.3.2 Japanese Failure Knowledge Database

The Japanese Failure Knowledge Database (JFKD) is an accident and failure database, whose aim is to make companies learn from past events in order to prevent future accidents and to improve reliability and safety of technology in society. The database started to be provided on the 23rd of March 2005 by the Japan Science and Technology Agency (JST) and it is managed by the Hatamura Institute for the Advancement of Technology (JFS, 2017).

Accidents and failures are divided in sixteen categories: selecting one of these categories it is possible to consult the corresponding accidents (JST, 2017).

3.3.3 Major accidents from *Lees' Loss Prevention in the Process Industries*

Another source of relevant accidents, from which some events have been considered and added to the final database, is the "Lees' Loss Prevention in the Process Industries" book (Lees, 2005).

Loss prevention approach is a wide field and it is rapidly developing. The author of the book felt the need to integrate the basic elements of the subject in a textbook, in order to give assistance to the direct interested, especially engineers. That is how and why "Lees' Loss Prevention in the Process Industries" has been written, as an attempt to meet this need. The book is divided in three different volumes. Volume 3 contains a series of appendices reporting reports or information about past accidents. In Appendix 1, table 1.2, some major accidents in process industries are listed in chronological order: from 1911 to 27.04.1995.

3.3.4 Fatality and Catastrophe Investigation Summaries - OSHA

"Fatality and Catastrophe investigation Summaries" is an accident database managed by "Occupational Safety and Health Administration" (OSHA). OSHA is one of the agencies of the United States Department of Labor (DOL). Its role is to guarantee safety and health to wokers in the workplace, through training, outreach, education, assistance and by setting and enforcing standards (OSHA, 2017).
OSHA's database is a collection of accident summaries developed after an inspection, performed by an OSHA's ispector, in response to a fatality or a catastrophe. It is possible to find summaries about accidents from 1984 to one year earlier than today's date. As a matter of fact, one year is necessary to compute all the steps needed to post online the summary.

3.3.5 National Response Center (NRC) database

The National Response Center (NRC) is one of the first "layers" of the National Response System (NRS). NRS is a multi-layered system: every layer has a function in order to respond effectively to hazardous substance releases. When a release occurs, the organization responsible for the release or spill is required by law to notify the NRC. Its function is to collect data in a national database and to notify the On-Scene Coordinator, which is responsible to evaluate and coordinate the response needed (EPA, 2017).
NRC is managed by United States Coast Guard (USCG). Its database is composed by annual reports available on-line, from 1990, and it is currently updated to 2017 (USCG, 2017).

3.3.6 ZEMA

ZEMA stands for "Zentralen Melde-und Auswertestelle für Störfälle und Störungen in verfahrenstechnischen Anlagen" (Infosis, 2017) which means "Central Reporting and Evaluation Station for Accidents and Faults in Process Plants". It is the german accident database, containing, mostly, events occurred in the german territory. It is managed by "Umweltbundesamt" (UBA), the main environmental protection agency in Germany (UBA, 2017). It has been instituted in 1993 to collect, evaluate and post all the events reportable to the "Störfall-Verordnung (12. BImSchV)" - the 12th Federal Immission Control Ordinance. ZEMA aims for being an important strating point for the development of technology and safety, making companies learn from past mistakes, in order to do not repeat them again. Since 1999, with the arrival of internet, all the

infomation included in the database had been open to the public. Nowadays, it is possible to count more than 570 national reports (Infosis, 2017).
Data are reported in "Jahresberichte", which are annual reports. On ZEMA website, reports from 1995 to 2014 are currently recorded. The last one is a biennal report: 2012-2014 (UBA, 2017). In "Anhang 1" - Appendix 1 - of these reports, a list of accidents of the corrisponding year is included, followed by detailed reports for each accident, in chronological order.

3.3.7 Ammonia Plant Safety and Rrelated Facility, Articles by AIChE

This collection of technical articles was made by American Institute of Chemical Engineers (AIChE, 2001). It consist of 42 volumes collecting a series of papers about new process development, maintenance and troubleshooting, revamping and upgrading of older ammonia facilities. Also, works reporting past accidents can be found. The goal of the collection is to study all the circumstances that led to a scenario or, for a great part of the cases represented, to a near miss. Then, the authors propose solutions as improving the design of some apparatuses, suggesting a different material or redefining organizational factors.
The most representative and significant cases on the ammonia plant were found in eight volumes. In particular, 31 cases contained from the volume 35 to the volume 42 were analyzed. All of these cases are described through detailed technical reports made by researchers and/or companies. Most of the incidents accounted in this collection are caused by mechanical failure (almost 60%). This is because these books are technical manuals. In fact, there are several studies on the mechanical properties of the materials and how they change due to the interactions with the process stream. For example, High Temperature Hydrogen Attack (HTHA) and Stress Corrosion Cracking (SCC) failures are largely discussed.
Another characteristic of this collection is the higher percentage of accidents in the ammonia reactor (up to 45%) respect to the other sources (for example, ARIA: 27%; MHIDAS: 17%). This is probably because the collection is completely focused on the ammonia plant and it is quite difficult to find incidents about ammonia reactors in literature. Moreover, the ammonia reactor is very sensitive to mechanical failure due to substances contained in the process stream (as hydrogen). So, all of these incidents are very interesting for technical manuals like these ones.

3.3.8 ARIA Database

The ARIA (Analysis, Research and information on Accidents) database is managed and administrated by the BARPI (Bureau for Analysis of Industrial Risk and Pollution), in collaboration with the French Minister of the Environment and the General Directorate for Risk Prevention. This database is free and it is possible to consult it on its web page. It collects a series of industrial and technological accidents from all over the world, containing over 46.000 reports about acci-

dents and incidents. An average of 1.200 new events per year are added (ARIA, 2018).
The Database is updated by engineers and technicians. In fact, it is possible to find detailed reports describing the main circumstances, outcomes, accident causes, how they managed the incident and the actions taken to avoid it. Nevertheless, they underline that the cases represented are not exhaustive, but the only aim is to make risk prevention and mitigation.
Looking for incidents occurred in an ammonia plant, 16 inherent cases were found. Some of these cases were integrated with information from other databases. All these cases were accounted as reports. So, an interpretative work during the data preprocessing phase was done.

3.3.9 MHIDAS (Major Hazard Incident Data Service)

In 1986, the Major Hazard Assessment Unit of the United Kingdom Health and Safety Executive (HSE) launched the Major Hazard Incident Data Service (MHIDAS). The database was maintained by AEA Technology. It is based on public domain information sources. In fact, a drawback of this source is the variability of quality and accuracy of reports (HSE, 1999). This database had been updated until mid 1990's (Hare et al., 2009).
All the cases presented in the database are accounted through keywords and a very short description of the accident could be reported. The search was firstly based on looking for accidents occurred in an ammonia plant and then on the main sections of the process. Here are listed only the keywords used for the search that they have produced acceptable results:

- Ammonia plant;
- Ammonia synthesis;
- H_2S;
- Syngas production.

15 cases are identified as relevant. Three of these cases were integrated with information from other databases. For other three incidents, it is unknown if they occurred in an ammonia plant, but the same technology was employed. For instance, in October 1981 in Czechoslovakia, a case of a catastrophic failure in a synthesis gas reactor that led to a severe flash fire was accounted, even if it is not specified that the accident occurred in an ammonia plant.
It is worth mentioning that 6 over 15 cases found are incomplete for general and specific causes, confirming that the data quality of this database is not always guaranteed.

3.4 The Ammonia Plant Accident Database

The ammonia plant accident database (Appendix B) has been obtained considering all the relevant events from the databases previously described. It consists in a list of dangerous events, to which specific information had been reported. Table 3.1 shows the categories of information considered, the so called "attributes" (Jiawei Han, 2012).

Attribute	***Description***
Date	Date of the event.
Location	Location - *i.e.* country, city, prefecture... - of the event.
Substance	Substances involved in the event.
Incident type	It specifies the typology of the accident (*i.e.* a release and/or an explosion and/or a fire). Keywords in table 3.2 have been used.
Origin	The particular area of the plant and the type of equipment from which the event started. Keywords in table 3.3 have been used.
General cause	The general cause - or causes - which provoked the event. Keywords in table 3.4, under "General cause", have been used.
Specific cause	The specific cause - or causes - which provoked the event. Keywords in table 3.4, under "Specific cause", are used.
Injured	The number of injuries due to the accident.
Evacuated	The number of people evacuated.
Killed	The number of deaths due to the accident.
Damage	The economical damage to the property or due to production loss.
Section	The section of the plant in which the event occurred.
Quantity	The amount (ton) of substances released.

Table 3.1: Information categories in the ammonia plant accident database

As previously mentioned, it is important to stress that the information collected in this database comes from different sources, using different ways to compile the accident reports and with a different level of detail. Reports in the different databases are different. For this reason, for some accidents it has not been possible to register all the information required.

Data collection and the data mining preprocessing steps have been computed to build the database, *i.e.* data have been collected, they have been cleaned from noise and inconsistent and inaccurate ones, information from different source have been taken, only the relevant data have been considered and a first data transformation have been done, finding a common schema to classify the attributes, using a list of keywords, for instance, indicated next.

3.4.1 Keywords

Keywords, to identify and register the incident type, the origin and the causes of the accidents in the database, have been used. Their source is the Major Hazard Incident Data Service (MHIDAS) database (HSE, 1999). Table 3.2 catalogs the keywords used for the incident type. Table 3.3 reports the ones to indicate the origin of the accidents, *i.e.* the particular area of the plant (general origin) and the type of equipment (specific origin) from which the considered event has started. Table 3.4 lists and describes the keywords used to indicate the general cause and the specific cause that provoked the examined accident.

Incident type	
EXPLODE	A release of energy producing gas at such a temperature and pressure and at such a speed as to cause damage to their surroundings.
FIRE	A process of combustion characterized by heat or smoke or flame, or any combination of these.
RELEASE	A release where it is not known whether instantaneous or continuous.

Table 3.2: Incident type keywords (HSE, 1999).

General origin		***Specific origin***	
PROCESS	The incident originated in items of process plant or in an area of process plant.	FIREDEQUIP	Fired process equipment, including furnaces, incenerators, stacks, chimneys.
STORAGE	The incident originated in items/area of storage plant.	HEATXCHANGE	Heat exchangers, including shell and tube, plate exchangers, evaporators, condensers, boilers, reboilers.
		HOSE	Hoses and other similar loading/unloading connections.
		MACDRIVE	Process machinery drives, including electric motors, engines, turbines.
		PIPEWORK	On-plant pipes and associated valves, joints.
		PSVESSEL	Pressurized storage vessels.
		PUMP	Any type of pump, compressor, ejector, fan.
		PVESSEL	Process vessels, including items such as centrifuges, towers, columns, dryers, distillation, absorption, filtration, cyclones, ion-exchange, crystallizer equipment, etc.
		TANKCONTR	A tank having a capacity of >=50L whose shell is fitted with items of service equipment and structural equipment. Capable of carriage by land or sea and of loading or discharge without removal of structural equipment. Possessing stabilising members external to shell, and capable of being lifted when full.

Table 3.3: Origin keywords (HSE, 1999).

General cause		***Specific cause***	
EXTERNAL	External events.	BRITTLE	Brittle failure.
HUMAN	Human factor.	COMPAIR	Compressed air or nitrogen.
INSTRUMENT	Instrument failure.	CONSTRUCT	Construction error.
MECHANICAL	Mechanical failure.	CONTROL	Controller.
PROCOND	Upset process conditions.	CORRODE	Crorrosion.
		DESIGN	Design error.
		ELECTRIC	Electricity.
		EXTNLFIRE	Fire.
		FLANGCOUPL	Leaking coupling or flange.
		GENERAL	General management error.
		GENERALOP	General operational.
		GLANDSEAL	Leaking gland or seal.
		INCOMPAT	Use of incompatible materials.
		INTNLFIRE	Internal fire.
		MAINTAIN	General maintenance.
		METALLURG	Other metallurgical failure.
		OVERHEAT	Overheating.
		OVERPRES	Overpressure.
		VALVE	Leaking or passing valve.
		WELDFAIL	Weld failure.

Table 3.4: Cause keywords (HSE, 1999).

3.4.2 Database Feature Analysis and Description

The number of accidents found and registered in the database is 140. In table 3.5, the distribution of the accidents, from the different sources, in terms of number of events collected, is shown and the weight of each source in the whole database is reported. It is important to indicate the weight, in order to understand the contribution of each database in the overall results.

Source	***Events collected***	***Weight***
Ammonia Plant Safety and Related Facilities	31	22,14%
Aria	12	8.57%
eMARS	21	15%
JFKD	10	7,14%
Lees'	5	3,57%
MHIDAS	11	7,86%
NRC	39	27,86%
OSHA	4	2,86%
ZEMA	5	3,57%
Other	2	1,43%

Table 3.5: Number of events collected, per source, and source weight in the ammonia plant accident database.

The NRC database is the source from which most of the accidents have been collected, followed by the Ammonia Plant Safety and Related Facilities: the accidents taken there, together, represent more than half of the total events. This means that is likely that the overall features of the database will be greatly affected by the characteristics of these two sources.

The time span covered is more than sixty years: it goes from the accident in Ube, Japan, occurred on the 11th of July 1959 - an explosion due to oxygen and syngas in the CO_2 removal section with 44 injuries and 11 deaths - , to the accident in Sulphur, Louisiana, on the 13th of July 2016 - an ammonia release from the reforming sections, with no injured or killed people.

Figure 3.1, reports the subdivision, per continent, of the accident locations in the database.

Most of the events registered occurred in America - almost 50% - , followed by Europe and Asia - almost 30% and more than 20% respectively. Only 2 events took place in Africa - less than 2% of the total. The accidents for which it has not been possible to obtain the location information are around 17% of the total number of events collected, indicated under the category "unknown" in the graph. It is important to report the unknown group, for each attribute, as it represents the quality of the database for the attribute considered: the lower it is, the higher the quality. As a matter of fact, the more data we have about a specific feature, the more it is possible to correctly identify and describe it and, therefore, the higher the quality of the database for that feature.

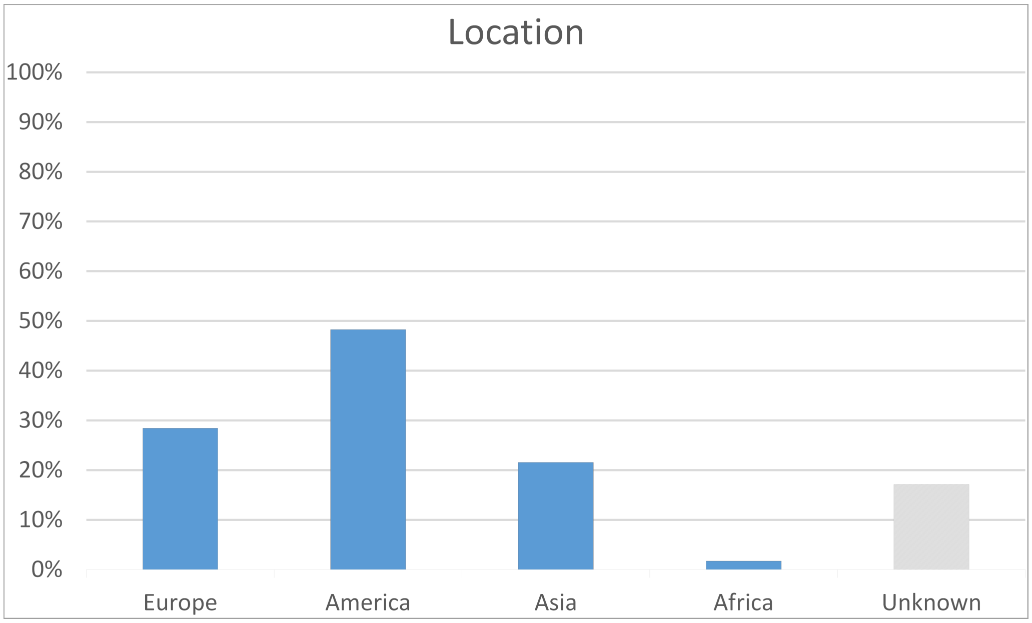

Figure 3.1: Event location subdivsion, per contentinent, in the ammonia plant accident database.

Ammonia plants fall on the Seveso III Directive (2012/18/EU), since many dangerous substances, in big quantities, are handled. Figure 3.2 shows most of these substances and suggests the ones mainly involved in the events collected, *i.e.* the ones responsible for the accidents.

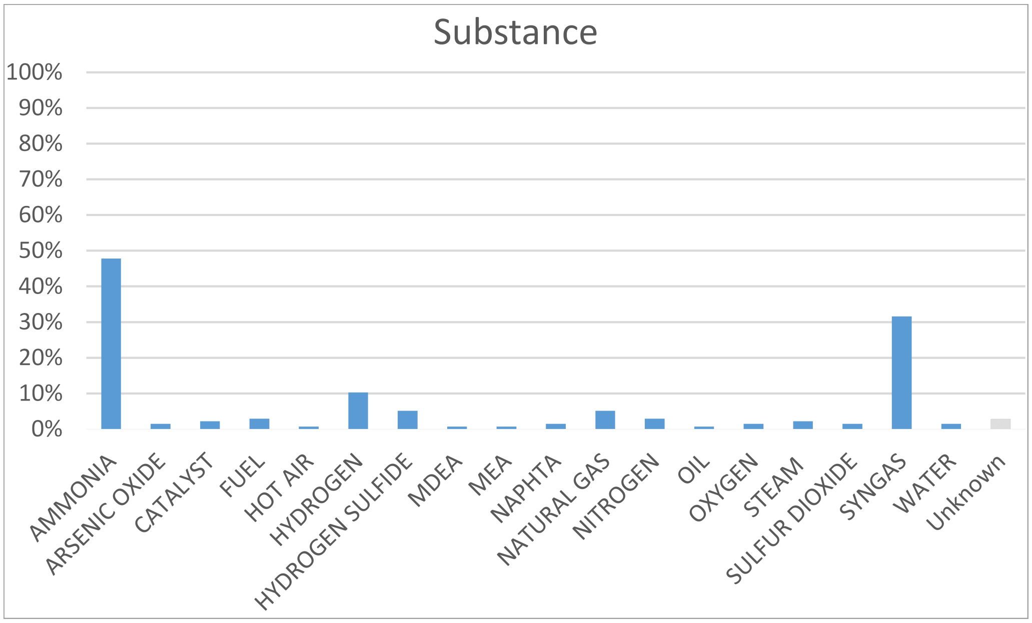

Figure 3.2: Substances involved in the events collected in the ammonia plant accident database.

It is evident that ammonia - present in almost 50% of the events collected - and syngas - more than 30% - are the substances that caused most of the accidents in the database. Other noteworthy substances are hydrogen - more than 10% - , hydrogen sulfide and natural gas - both at around 5%. The quality of the database, for the feature "substance", is quite good, since for only 4 events the substances involved were not indicated. It is important to mention that more than one substance could be responsible for a single event.

In figure 3.3 the percentage subdivision of the incident type, for the accidents registered in the database, is reported.

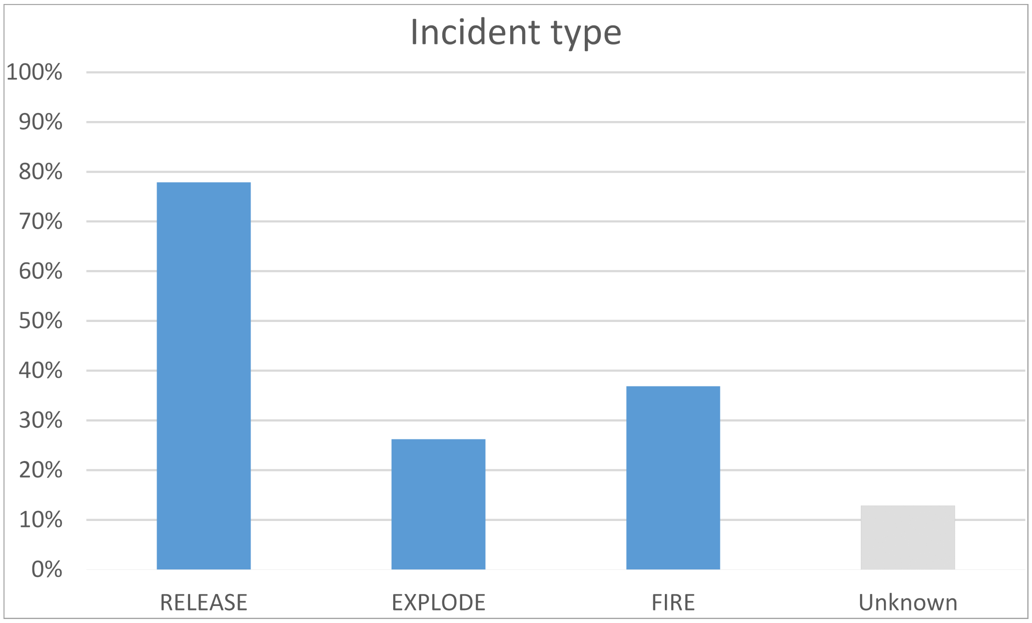

Figure 3.3: Incident type for the accidents collected in the database.

Around 80% of the accidents is due to a release. The percentage is high because, in most of the cases, an accident starts with a release then it evolves into an explosion or a fire. Thus, it is common to find the combination "RELEASE and EXPLODE" or "RELEASE and FIRE" or even "RELEASE,EXPLODE and FIRE" under the "Incident type" field. The percentage of "FIRE", in the events found, is around 40% and the one of "EXPLODE" is around 25%. The unknown events, as for the incident types, are about 13% of the total events registered in the database.

About the "Origin" category, as it is shown in figure 3.4, the events in the database are mostly originated in items or in an area of process plant - almost 95% -, rather than in the storage area - a bit more than 6%. In the sources investigated to collect the accidents, it was possible to take into account more accidents occurred in the storage area than the ones actually registered. However, it has been decided to focus mainly in the process part, this is why only a few of them have been considered - only the most representative ones.

Figure 3.5 shows the different general causes for the accidents collected, quantifying, in percentage, how often each general cause have been met in the registered events.

The main general causes are the "MECHANICAL" one - found in more than 60% of the events

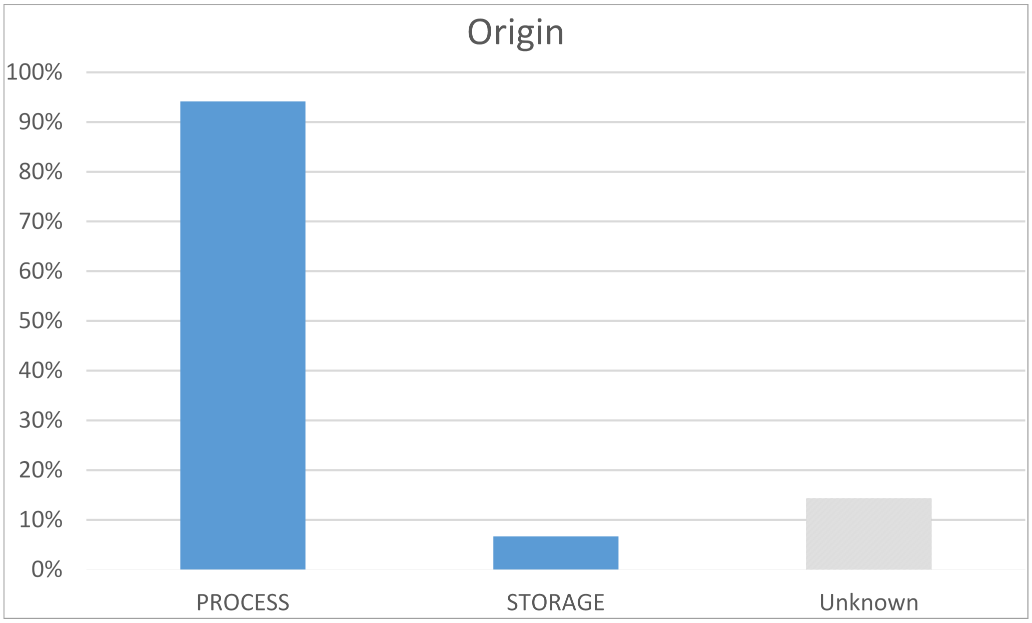

Figure 3.4: General origin of the accidents in the ammonia plant accident database.

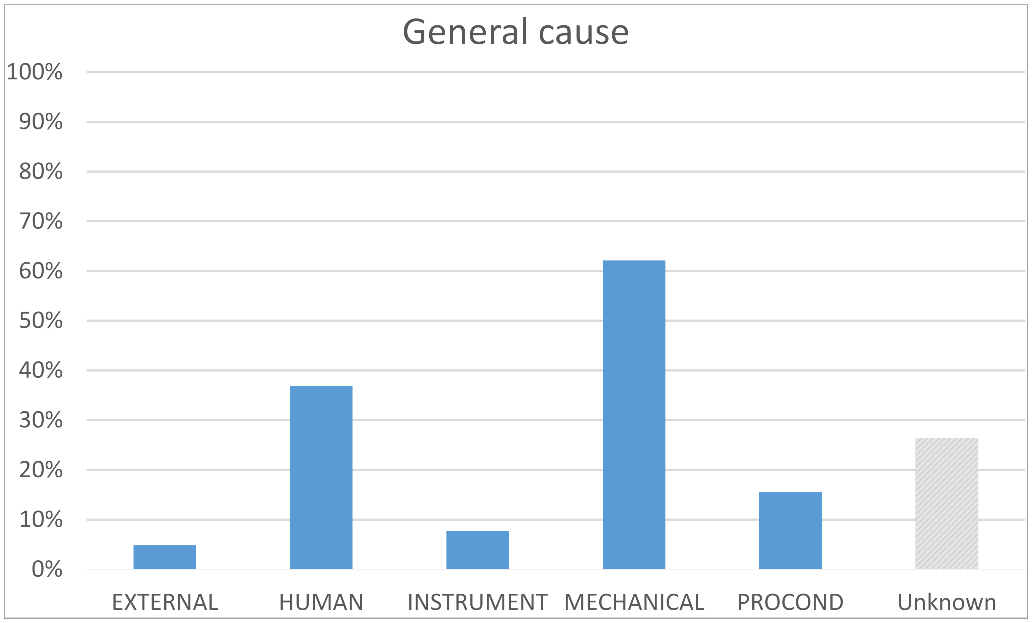

Figure 3.5: General cause for the accidents collected in the ammonia plant accident database.

- and the "HUMAN" one - almost 40%. It is necessary to mention that, also in this case, more than one general cause can be found in a single event. The general cause is not always clearly indicated in reports or it is not indicated at all. This is why the percentage of the unknown category is high: for more than 25% of the total number of events it has not been possible to recognize a general cause. The more the details requested for a specific feature, the less the quality of the database for that feature. That is why for the specific cause, shown in figure 3.6, the percentage of the unknown category is even higher than for the general cause, around 40%. This means that the representation quality of the database, for the specific cause, is not relevant.

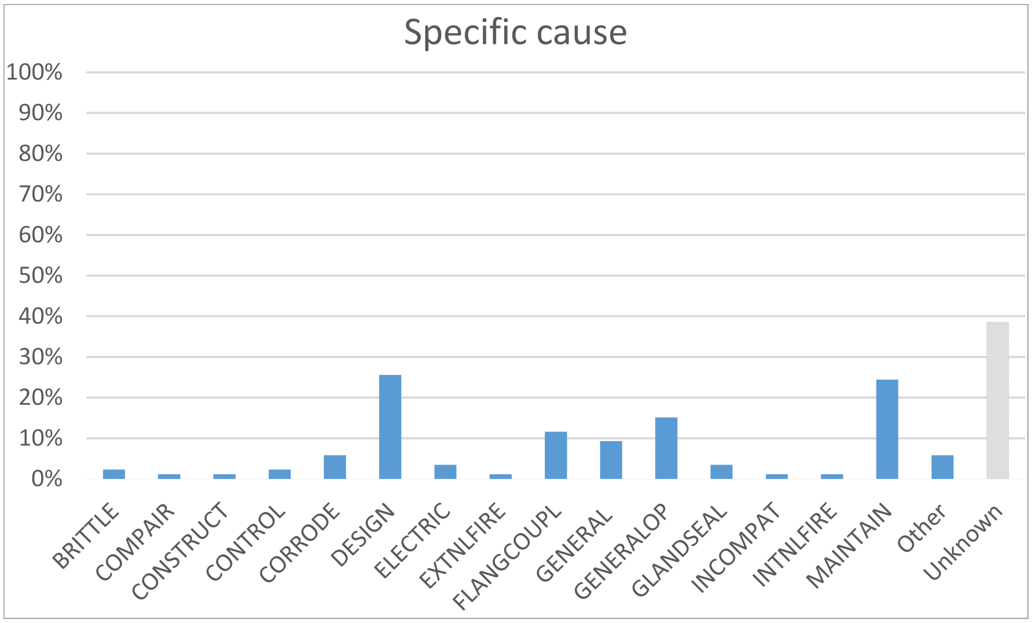

Figure 3.6: Specific cause for the accidents collected in the ammonia plant accident database.

Given these points, the main specific causes are the "DESIGN" one and the "MAINTAIN" one - both present in around 25% of the events for which it has been possible to find a specific cause. Noteworthy also the "GENERALOP" - around 15% - the "FLANGCOUPL" - more than 10% - and the "GENERAL" - almost 10%.

Figure 3.7 shows the consequences of the accidents on humans, the damage on humans, in terms of injuries and deaths. A classification in three categories has been done in order to assess, qualitatively, the magnitude of the accidents collected:

- the "No injured - No Dead" category, including all the events with zero injuries and zero deaths;
- the "Only injured" category, including the accidents with one or more injured but zero deaths;

- the "Dead" category, containing all the events with one or more deaths.

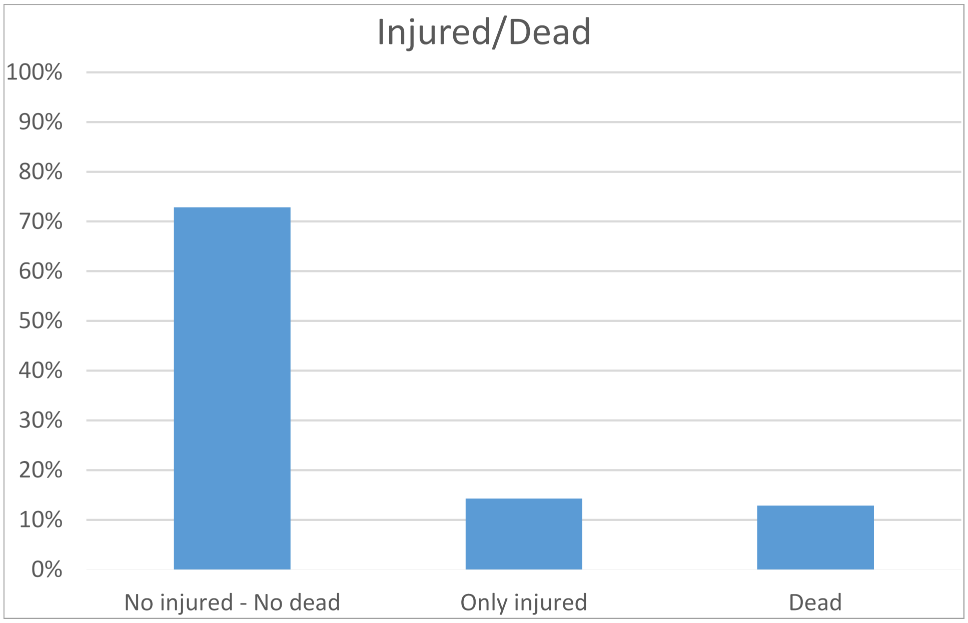

Figure 3.7: Damage on humans, in the accidents collected in the ammonia plant accident database, divided in three categories: "No injured - No dead", "Only injured" and "Dead".

The most common category in the events collected in the database is the "No injury - No dead" one, found in more than 70% of the accidents. Injuries and deaths are found with lower percentages: a slightly higher one for the "Only injured" category - a bit more than 14% - than the "Dead" one - almost 13%.

Interesting issues come from the analysis of figure 3.8, indicating, in percentage, the occurrence of the events collected in the database in the main sections of the ammonia plant.
It is clear that the most critical section is the reforming one - around 40% of the accidents for which it has been possible to indicate the section occurred there - , followed by the ammonia synthesis one - around 30%. However, in many reports, the section involved in the accident was not clearly stated. As a result, the unknown category's percentage is high - almost 40% - and the quality of the database in describing the critical sections is relatively low.

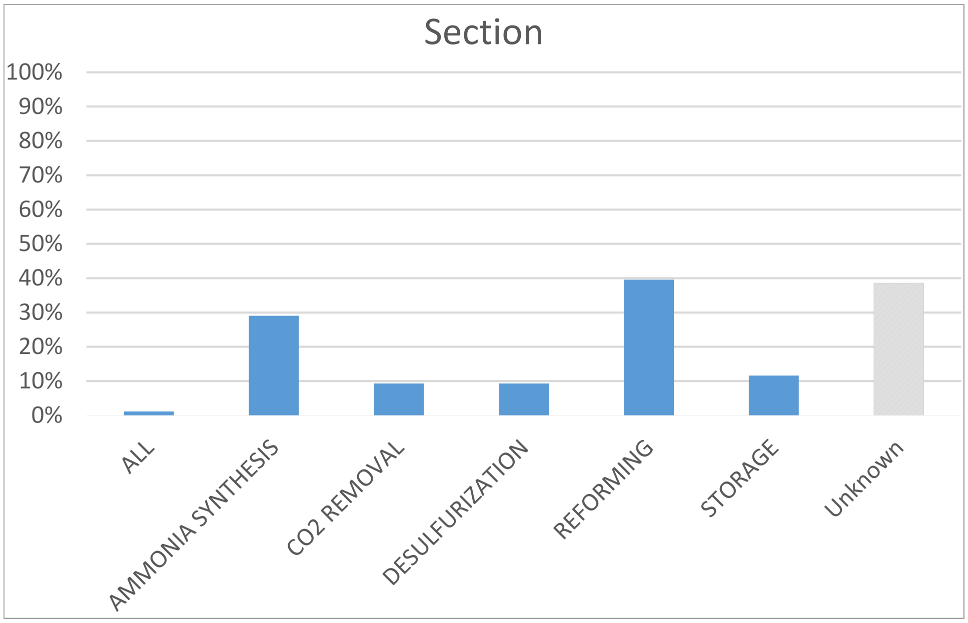

Figure 3.8: Sections involved in the accidents collected in the ammonia plant accident database.

Information about the "damage" and the "quantity" is not often indicated in the reports analyzed. Damage data have been collected only for 22% of the total number of events and quantity data only for 32%. Consequently, the description quality of the database for these two features is really low, and it is not meaningful to analyze them more in detail.

3.4.3 Remarks and Assumptions

The origin and especially the cause of an accident can be tangled; therefore, it is relatively complicated to describe with only one word each field - general origin, specific origin, general cause and specific cause of the examined accident. This is also because the accident reports analyzed to extrapolate the data to register in the database were often not clear and incomplete, *i.e.* not all the information sought were actually reported. For this reason, more than one keyword for category have been used when the particular event needed it. It is also important to stress the limitation, due to the personal interpretation of the information contained in the reports, when assigning the keywords.

The problem deriving from bad written, bad structured or incomplete accident reports is a real issue in the industrial field. For this reason, companies are currently working on methods for teaching the operators to write properly made ones. An example of one of these methods is based on a process of reverse engineering: the operator should write clear standardize structured reports and fill them according to the information he, himself, would like to find in them.

Different assumptions have been made while collecting and registering the data in the ammonia plant accident database. For instance, about the number of injuries and deaths caused by an accident. Because of the issue of incompleteness of the reports, just discussed, sometimes this information is not even mentioned. Damage on humans is a critical, important, information, the first thing to write on a report. If nothing is indicated about that, it can be assumed that the accident did not cause any injury or fatality. Thus, in these cases, a number of 0 injured and 0 killed have been taken.

Occasionally, during the historical analysis, the same accidents, in two or more databases, have been found. When this happened, the event considered has been registered once only, under the database that best describes the event - *i.e.* with a higher number of relevant data - , adding and completing the information needed with the one in the other databases. In the few exceptions in which the number of injured or dead was different, the higher value has been taken, conservatively.

It is finally important to stress, as already mentioned before, that some of the accident sources analyzed have a higher weight on the overall information contained in the database. From these sources a higher number of event has been taken and, consequently, the behavior of the database describing the features will depend more on them. For instance, the NRC database is the source with the highest weight - 27,86% of the events have been taken from them - and it mainly describes accidents occurred in America, releasing ammonia, with no injured and no dead, without going into detail - or without even indicating - when describing the causes of the accident or the section involved. It is possible to see how these information partially reflects the feature description of the database made in the previous subsection.

Chapter 4

Methods: a Machine Learning Application

4.1 Introduction

Throughout the years, artificial intelligence (AI) has become an important field. People look for intelligent software for different purposes. For instance, recognizing images, music or videos - through computers - is nowadays possible. AI is also used to support medical decisions or as a tool for scientific research. In short, there are many practical applications and active research topics about AI (Goodfellow et al., 2016).

In different projects involving artificial intelligence, it was common to hard-code the knowledge of the world, allowing the computer to reason automatically on the basis of that knowledge. Even though the idea was promising, they met several difficulties and none of the projects led to a major success. It was only then that the need, for AI systems, to acquire their own knowledge from data was brought to light and the machine learning discipline was born (Goodfellow et al., 2016).

According to Murphy (2012), machine learning is defined as "a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kind of decision making under uncertainty". In other words, computers learn and understand the world from experience - *i.e.* real data - and they are able to make intelligent decisions based on it (Jiawei Han, 2012).

Nowadays, we are entering the era of big data (Murphy, 2012). A huge amount of data is constantly produced and stored in databases. This should be seen as an opportunity to learn. It is a need to try to analyze and to understand these data, in order to obtain important information. Machine learning can be the perfect tool to do so.

This work suggests an innovative way to analyze data coming from the process industry, in order to obtain support for safety-related decision making. The patterns in the data - and the knowledge coming from there - have been found using machine learning. The methods, developed to efficiently manage the data and retrieve knowledge from them, have been studied together with Kongsberg Digital. As a working tool, the open source software library TensorFlow have been used.

In the chapter, TensorFlow is first described, the models used to analyze the data are then illustrated and, finally, the information needed to run the simulation and to understand the results are indicated.

4.2 TensorFlow

As stated in the offical website (TensorFlow, 2018), TensorFlow is an open-source library for Machine Intelligence. It was first developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization. Thus, it originally started as an in-house tool, but in 2015 was made available for everyone as an open-source software (Vincent, 2017). This allows the community to improve TensorFlow with contributions. According to Unruh (2017), more than 890 external contributors added something to the code. The main communities are Stack Overflow (Stack Overflow, 2018) - monitored by the TensorFlow team - and GitHub (GitHub, 2018) - which has had more than 1000 unique non-Googler contributors -, but the number of repositories using it is constantly growing (Unruh, 2017).

TensorFlow is the primary tool used for a lot of machine learning works in Google products (Dean, 2017), but its domain of applicability is wide. For instance, it found applications in language translations - speech recognition capabilities and image recognition capabilities have improved through its use (Le and Schuster, 2016) - and in medical diagnosis, such as detection of skin cancer and preventing blindness in diabetics (Sandjideh, 2017). More in general, TensorFlow is an accessible framework to create custom tools for a whole range of industries (Vincent, 2017), included the process industry.

TensorFlow uses data flow graphs to execute numerical computation. In these graphs, it is possible to find nodes - which represent mathematical operations - and graph edges - which serve as tensors connected between them. Tensors are multidimensional data arrays and represent the central unit of data in TensorFlow (TensorFlow, 2018).

It also provides multiple application programming interfaces (APIs): a lower one, which consents to have complete programming control, and a higher one. The last one simplifies the mechanics of machine learning - *e.g.* running training loops, running evaluation loops and managing data sets - and it is easier to learn and use. Moreover, it offers a number of commonly used models ready to use out of the box, making it easy to configure them (TensorFlow, 2018).

Generally speaking, the aim of this work is to solve a binary classification problem, described later. To do so, the "tf.contrib.learn" high level API has been used. The models trained and evaluated, whose tools to build and manage are already available in the API, are a linear one - *i.e.* the linear model -, one based on deep learning - *i.e.* the deep model - and a combination of the previous 2, supposed to have the benefits of both of them - *i.e.* the wide&deep model.

4.3 The Models

Even though every model considered is conceptually different, they all go through a process of learning, following, in general, the same steps. The difference is in how and in what they learn. Generally speaking, every model elaborates some inputs to generate the output. To do so, it is first trained with a training data set. The data possess both the values of the inputs and of the output. This is because, during the training step, the model parameters are the ones who have to be quantified. Once the model is trained, it can be used for prediction.
Focusing on binary classification - which is the one of interest for this work - after the training, the input space is divided into two "decision regions" whose boundaries are called decision boundaries (Bishop, 2006). Decision boundaries assume different shapes for different models. Considering the same model, decision boundaries can be different, according, mainly, to the number of steps and the threshold value chosen. This last value is especially important. It is the probability value, according to which, the output is part of the class considered or not. As a matter of fact, at the end of the training, the model returns values - in the interval [0;1] - representing the probability of belonging to a class or to the other one - since binary classification has been considered. For instance, it is assumed that the model returns $P_0 = 0.65$ and $P_1 = 0.35$ - with P_0 as the probability of being part of the class "0" and P_1 as the probability of being part of the class "1". If the threshold value, referred to the class 0, is $T = 0.6$, the model will conclude that the output is in class 0. However, if $T = 0.7$, the model will conclude that the record is in class 1.

4.3.1 The Linear Model

According to Murphy (2012), linear models are the "work horse" of statistics and machine learning. Following what Hastie et al. (2009) indicates, for a linear model, given a vector of inputs $X^T = (X_1, X_2, ..., X_p)$, the output Y is predicted via the equation:

$$Y = \beta_0 + \sum_{j=1}^{p} X_j \beta_j \tag{4.1}$$

where β_0 is the so-called bias - *i.e.* the intercept -, $\beta = (\beta_1, \beta_2, ..., \beta_p)$ is the vector of the model parameters.
Often 4.1 is written in vector form as an inner product:

$$Y = X^T \beta \tag{4.2}$$

including the constant variable 1 in X and the bias, β_0, in the vector of the model parameters, β. The model needs then to be trained with a training set data in order to learn the weights - *i.e.* the values of the model parameters - of every attribute (or feature) - *i.e.* the inputs X_j. Once the weights are known, the model can be use for prediction. However, considered its definition, the linear model is not able to take into account the relative importance of specific combinations of features: after the training, independent weights are assigned to separate features (TensorFlow, 2018). In TensorFlow, for instance, it is possible to get around this limitation introducing new inputs, constituted by the combination of the features believed to be dependent one from the other - *i.e.* the "crossed-columns" or "crossed-features" (TensorFlow, 2018).
Figure 4.1 shows a classification example using a linear model.

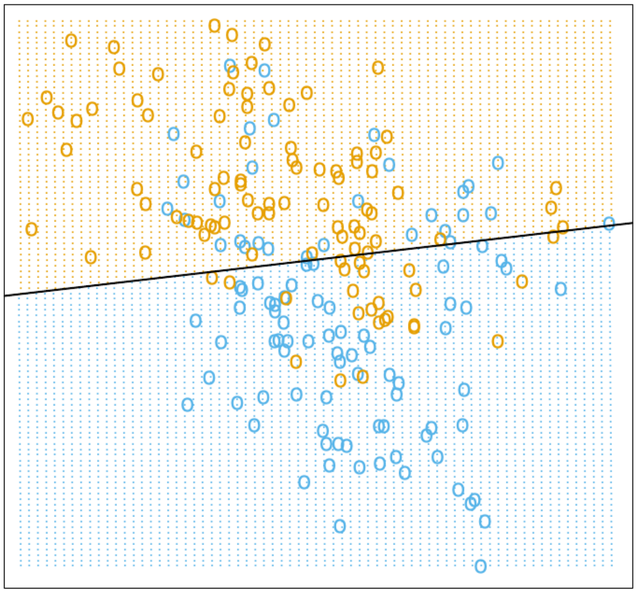

Figure 4.1: Classification using a linear model (Hastie et al., 2009).

It is a binary classification in which, for example, the response Y of the model is coded as 0 for "Blue" and 1 for "Orange". In figure 4.1 are 100 points, each one of them are "Blue" or "Orange". After the training, with definite number of steps and threshold value, the model recognizes two different areas: the lower one is the "Blue" one and the upper one is the "Orange" one. In this

case, the decision boundary is a linear function of the input vector X, since a linear model has been used for the classification (Hastie et al., 2009).
Even though this approach may seem dull in comparison to other modern and more refined methods, linear models are still widely used (James et al., 2014). They train quick, they work well on very large feature sets and can be interpreted easily (TensorFlow, 2018). As a matter of fact, there is a direct correlation between every input and the output, weighted on the model parameters. Thus, it is possible to define the impact of every feature on the output (Hastie et al., 2009). Finally, its simplicity and accessibility make the linear model a good starting point for learning about machine learning (TensorFlow, 2018) and to develop new approaches (James et al., 2014).

4.3.2 The Deep Model

The deep model used for this work is a feed-forward neural network. To understand how it works, a good starting point is considering linear models and how to overcome its limitations (Greenberg et al., 2012). One of the most evident is that the connections between inputs and output is restricted to linear functions. With respect to that, the central idea behind neural networks is to model the target as a nonlinear function of the input features (Hastie et al., 2009).
According to Bishop (2006), the basic neural network model can be described as a series of functional transformations. First, M linear combinations of the input variables $X_1, X_2, ..., X_p$ are constructed, following the equation:

$$a_i = \beta_{i0} + \sum_{j=1}^{p} X_j \beta_{ij} \tag{4.3}$$

with $i = 1, ..., M$.
Quantities a_i are called activations (Bishop, 2006).
As in the linear model, β_{i0} are the bias, β_{ij} are the model parameters.
This is the first layer of the network.

The activations are then transformed using a differentiable, non linear function, called activation function - $h(\cdot)$ - (Bishop, 2006):

$$Z_i = h(a_i) \tag{4.4}$$

This layer is the so-called hidden layer of the neural network. The Z_i are called hidden units (Greenberg et al., 2012). Their name is due to the fact that the training data does not show their values (Greenberg et al., 2012). This can be seen as a limitation, because, unlike the linear model, it is not interpretable and it is not possible to know the weights of the inputs on the outputs.

In general, the hidden layers can be more than one. To make it simple, only one hidden layer is now considered.

Different activation functions exist, but one of the most used is the sigmoid (eq. 4.5):

$$h(a) = \frac{1}{1 + exp(-a)} \tag{4.5}$$

Next, the hidden units are again linearly combined, in order to give the activations - a_k - of the so called output layer (Bishop, 2006):

$$a_k = \beta_{k0} + \sum_{i=1}^{M} Z_i \beta_{ki} \tag{4.6}$$

with $k = 1, ..., K$ - K is the total number of outputs.
β_{k0} and β_{kj} are again, respectively, the bias and the model parameters of this layer.

Finally, a proper activation function is used to give the outputs Y_k:

$$Y_k = \sigma(a_k) \tag{4.7}$$

Combining the various stages, the overall network function takes the form:

$$Y_k(X, \beta) = \sigma\left(\beta_{k0} + \sum_{j=1}^{M} \beta_{kj} h\left(\beta_{i0} + \sum_{j=1}^{p} X_j \beta_{ij}\right)\right) \tag{4.8}$$

where the vector β contains all the biases and the models parameters.
The model just built, represented in equation 4.8, is a non-linear function, giving outputs Y_k from inputs X_j, controlled by a vector β of adjustable factors - *i.e.* the model parameters.

A neural network is typically represented by a network diagram as in figure 4.2.
This is the case of a feed-forward neural network with p inputs, M hidden units - in the only hidden layer -, and K outputs.
The model is called "feed-forward", because information flows through the network, from the inputs to the outputs, without feedback connections - *i.e.* connections through which the outputs are fed back into the model. Neural networks incuding also feedback connections are called recurrent neural networks (Greenberg et al., 2012).
Moreover, it is composed by layers, forming a chain structure. The overall length of the chain gives the depth of the model. The name "deep learning" derives from it (Greenberg et al., 2012).
Finally, the name "neural" is inspired by how the human brain works. Each units represent a

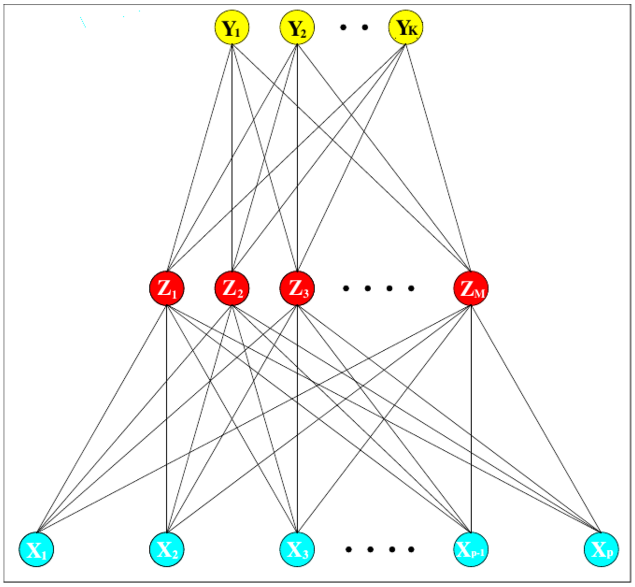

Figure 4.2: Feed-forward neural network (Hastie et al., 2009).

neuron and the connections represents synapses (Hastie et al., 2009). Similar to neurons, the units receive inputs from many other units and they elaborate their own outputs (Greenberg et al., 2012).

Training a deep model of this kind requires more effort than a normal linear model. For instance, the design decisions to be made consist also in building the architecture of the network - how many hidden layers, how many units for each hidden layer and how they should be connected to each other - and in selecting appropriate activation functions (Greenberg et al., 2012). Moreover, the computational effort is higher than for a linear model. However, the decision regions are not so neat and the decision boundaries can be of any kind - they are not limited to be linear, as shown in figure 4.3.
The figure represents a binary classification using two different classifiers - *i.e.* models whose data analysis task is classification (Jiawei Han, 2012). The black continuous line is the decision boundary for the neural network model, the broken purple boundary is referred to the so-called Bayes classifier. The last one has not been considered for this work and more information can be found elsewhere (Hastie et al., 2009).

4.3.3 The Wide&Deep Model

Linear models and deep models are widely used and they form the basis of many important applications (TensorFlow, 2018; Greenberg et al., 2012). The first ones are able to learn and memorize the impact of every feature to the output - for instance from historical data - and reproduce it for new predictions. The weak point is the generalization of the results: to achieve it,

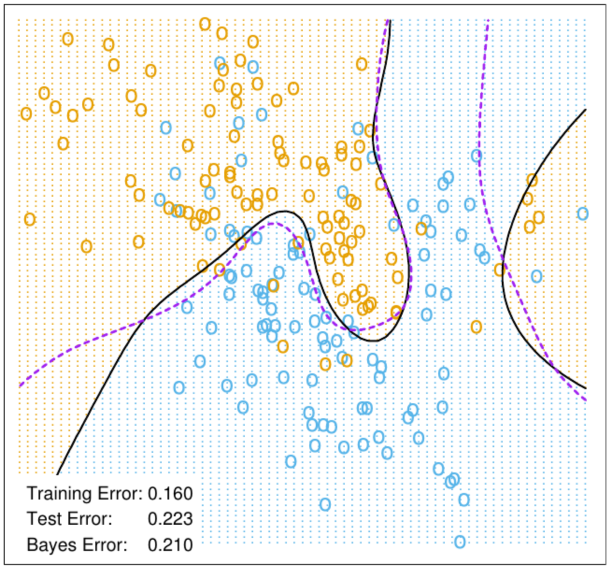

Figure 4.3: Decision boundaries for a neural network model (Hastie et al., 2009).

a higher engineering effort is required. However, the model is not able to take into consideration combinations of features who have not been already manually implemented. The deep neural networks, on the other hand, are able to generalize the results, finding combinations of features that have never or rarely occurred in the past. The problem is that they can over-generalize and make less relevant recommendations (Cheng et al., 2016).

In order to obtain the benefits of both the linear models and the deep neural network models, Cheng et al. (2016) introduce the wide&deep model. The wide component is a generalized linear model of the form of equation 4.1 of section 4.3.1. The deep one, is a feed-forward neural network, discussed in section 4.3.2. The wide&deep model is based on the joint training of the two. This means that all the parameters of the wide part and the deep part are optimized simultaneously and then combined during training time. This way, both memorization - from the wide part - and generalization - from the deep part - are obtained.

In figure 4.4, the 3 models - *i.e.* wide, deep and wide&deep - are represented.

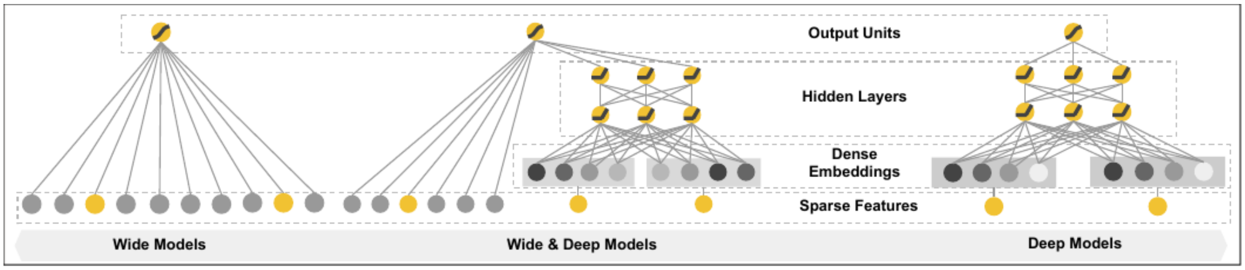

Figure 4.4: From the left: the wide model, the wide&deep model and the deep model configurations (Cheng et al., 2016).

It is clear, from the figure, the linear dependence of inputs and output in the wide models. In deep models, the sparse features are first converted into low-dimensional and dense real-valued vectors, the so-called embedding vectors, then they are fed into the hidden layers and, finally, the output is obtain. The wide&deep models are a combination of the two. In general, for the latter, the wide part only needs to complement the weakness of the deep with a small number of crossed-features, rather than a full-size wide model (Cheng et al., 2016).

4.4 The Simulations

The aim of the work conducted is to analyze accident data from chemical process industries and obtain important information, in order to support safety-related decision making. To do so, the ammonia plant accident database - described in chapter 3 - is considered. Taking as inputs the data collected, the objective is to detect the model that best predicts the consequences on human - in terms of people killed or injured - deriving from the accidents, focusing in particular on the prediction of the rarest event.

In order to do so, a linear model, a deep neural network model and the wide&deep model have been considered. The open-source library TensorFlow has been used as working tool. Here, through the high-level API "tf.contrib.learn", the models just mentioned are already ready to use out of the box.

The evaluation of the model is done using the ammonia plant accident database. However, another accident data set is needed in order to train the model - *i.e.* to define the model parameters. For this purpose, the MHIDAS database (HSE, 1999) has been used. The events in common between the two databases - *i.e.* the MHIDAS and the ammonia plant accident databases - have been canceled from the latter one.

MHIDAS contains 8972 accidents, with the attributes indicated in table 4.1.

The training data set and the one used for evaluation need to have the same attributes, in the same form and order. For this reason, modifications on both the data sets have been done. The final attributes, considered for the work, are the ones in table 4.2.

The code to build the computational graph on TensorFlow and run the simulations is written in python language. A summary of the code is reported in the following.

```
#import datasets
training_datasets
test_datasets

#attributes
columns=["DA", "PD", "LO",..., "IS2"]

#Define columns
DA= tf.feature_column.categorical_column_with_vocabulary_list(
'DA', ["<1900s", "1900s", "1910s", "1920s", "1930s", "1940s", "1950s", "1960s", "1970s", "1980s",
"1990s"])
PD = tf.feature_column.categorical_column_with_vocabulary_list("PD", ["R", "U","Na"])
LO = tf.feature_column.categorical_column_with_hash_bucket('LO', hash_bucket_size=280)
...
IS2 = tf.feature_column.categorical_column_with_hash_bucket('IS2', hash_bucket_size=20)

#crossed columns crossed=[
tf.feature_column.crossed_column(["DA", "LO"], hash_bucket_size=int(1e5)),
...
tf.feature_column.crossed_column(["IS1", "IS2"], hash_bucket_size=int(1e4))
]

#WIDE MODEL input
#if insert_label_type == "NPI":
base_columns=[DA, PD, LO,..., IS2]
crossed_columns=crossed

#DNN MODEL input
deep_columns = [
tf.feature_column.indicator_column(DA),
tf.feature_column.indicator_column(PD),
tf.feature_column.embedding_column(LO, dimension=8),
...
tf.feature_column.embedding_column(IS2, dimension=4),
]

#CLASSIFIERS

model_wide model_deep model_w_d
#training and evaluation
model.fit(input_fn=get_input_fn(train_file, None, False) , steps=insert_steps)
results=model.evaluate(input_fn=get_input_fn(test_file, 1, False), steps=None)

#print the results
print("%s : %s" %(key, results[key]))

#print the probability values
predictions=list(model.predict_proba(input_fn=get_input_fn(test_file, 1, False)))

print("\n".join(map(str, predictions)))
```

MHIDAS Attributes	
AN	MHIDAS Record Number
ME	Multiple Entry
CR	Contributor
DA	Date of Incident
PD	Population Density
LO	Location
MN	Material Name
MT	Material Type
MH	Material Hazard
MC	Material Code
IT	Incident Type
OG	Origin
NP	Number of People Affected
DG	Damage (US Dollars)
GC	General Cause
SC	Specific Cause
QY	Quantity (Tonnes)
IG	Ignition Time (Seconds)
IS	Ignition Source
KW	Keywords
AB	Abstract
RA	References Available
ID	Inverted Incident Date
KR	Killed
IR	Injured
ER	Evacuated
DR	Damage
QR	Quantity

Table 4.1: MHIDAS database attributes (HSE, 1999).

Final attributes	
DA	Date
PD	Population Density
LO	Location
MN	Material Name
MH	Material Hazard
MC	Material Code
IT	Incident Type
GOG	General Origin
SOG	Specific Origin
NPI	Number of People Injured
NPK	Number of People Killed
NPE	Number of People Evacuated
GC	General Cause
SC	Specific Cause
QY	Quantities (Tonnes)
IS	Ignititon Source

Table 4.2: Attributes considered for the simulations.

In order to accurately asses and compare the criteria to use for building the models, cases with 200, 2000, 20000 and 200000 iteration steps have been used for the training. The simulations are carried out using NPI (Number of People Injured) and NPK (Number of People Killed) as output labels.
Moreover, different classes of simulations have been performed, according to the categorization made on the output label's values. Table 4.4 shows this categorization for NPI.

Output label's categories	
YES	one or more injuries
NO	no injuries
<10	between 1 and 10 injuries
10-100	between 10 and 100 injuries
100-1000	between 100 and 1000 injuries
>1000	more than 1000 injuries

Table 4.4: Categorization for NPI values

The same categorization has been carried out for NPK, considering the deaths instead of the injuries.

All in all, before running the simulations, the output label, the model type, the iteration steps and the output label category have to be chosen. Table 4.5 reports all the possible options.

Output label	***Model type***	***Iteration steps***	***Output label category***
NPI	wide	200	YES
NPK	deep	2000	NO
	wide&deep	20000	<10
		200000	10-100
			100-1000
			>1000

Table 4.5: List of all output labels, model types, iteration steps and output label categories considered for the simulations.

4.4.1 The Evaluation Metrics

In order to evaluate the performance of the model selected - for a considered output label, number of steps and output category - the program compares the values predicted with the real ones. The evaluation metrics used for this work are listed in table 4.6.

Evaluation metrics	
accuracy	It calculates how often predictions match labels (TensorFlow, 2018).
accuracy/baseline_label_mean	It indicates the fraction of text examples belonging to the output category picked (Stack Overflow, 2018).
auc	Area under the ROC curve.
auc_precision_recall	Area under the curve precision/recall.
global_step	Total number of iteration steps.
loss	It measures the discrepancy between the truth value of the values in the output category and the model's prediction (TensorFlow, 2018).
precision/positive_threshold_0.5_mean	It reports the statistic value "precision".
recall/positive_threshold_0.5_mean	It reports the statistic value "recall".

Table 4.6: Evaluation metrics used for the simulations.

Between all these values, the focus is primarily on the recall and on the "auc_precision_recall". The recall can be calculated following the equation (Flach and Kull, 2015):

$$R = \frac{T_p}{T_p + F_n} \tag{4.9}$$

R indicates the recall.
T_p are the so-called "true positives". They are events belonging to the output category chosen - *i.e.* the "positive" events - which were correctly detected.
F_n are the so-called "false negatives". They are events belonging to the output category chosen which were not correctly detected. The reason of the name is because the model detected the events wrongly, as "negatives" - *i.e.* not belonging to the output category selected.
The recall indicates, then, the fractions of events, belonging to the output category chosen, correctly detected (Greenberg et al., 2012). A high value of this metric is obtained when F_n is low. Thus, it indicates that the quantity of positive events not detected from the model is low. However, in this case, it is possible to detect, as positives, also events which are actually negatives. These are called "false positive".
Usually, to correctly asses the performance of a model, the recall is used in combination with the so-called "precision", defined as (Flach and Kull, 2015):

$$P = \frac{T_p}{T_p + F_p} \tag{4.10}$$

where P is the precision, T_p the true positive and F_p the so called "false positive".
The precision indicates the fraction of detections reported by the model that were correct (Greenberg et al., 2012). A high value of precision is obtained when F_p is low. It means that, in this case, almost all the "positives" detected are actually positive. However, it does not take into consid-

eration the false negatives. This means that it is possible to have a really high value of precision without detecting many positive values - *i.e.* detecting them as negative.
For this work, more than having a high value of precision, the purpose is to look for a model able to detect all the positive events, also the rarest ones. That is why it is interesting to focus on the recall. The fact that having high values of recall means, in general, low values of precision - *i.e.* high values of false positives - is not important. Detecting, as positives, values that usually are negative, only makes the model more conservative.
Anyway, the combination of these two statistical values gives, in general, a good idea of the performance of the classifier. As a matter of fact, the curve precision-recall is widely used. An example of the curve is shown in figure 4.5.

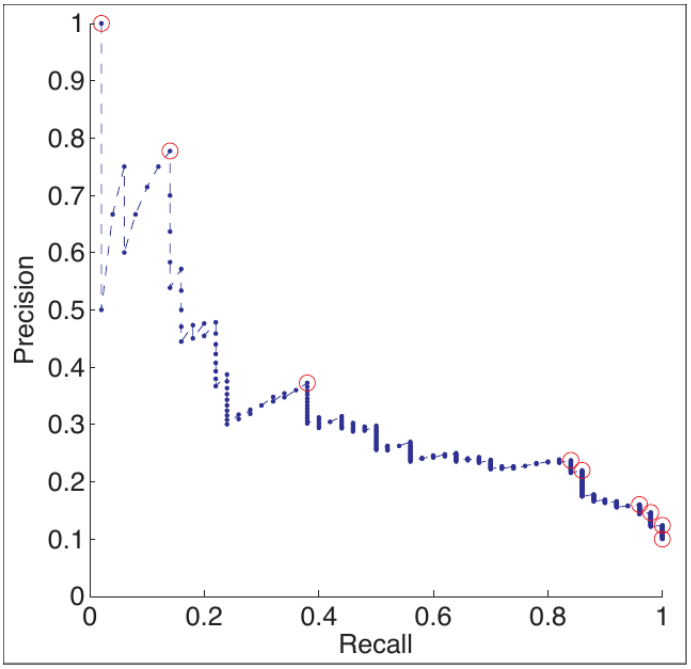

Figure 4.5: Curve precision-recall (Flach and Kull, 2015)

This curve is obtained trading precision for recall varying the threshold (Greenberg et al., 2012). In general, decreasing the threshold referred to the positive values - *i.e.* the probability value defining if the event is positive or not -, the recall increases while the precision decreases.
Rather than using a curve, it is usually preferred to summarize the information with a single number (Greenberg et al., 2012). In order to do so, the area under the curve precision-recall is considered. The bigger this value the more interesting is the model.
The simulations computed for this study automatically return values of precision and recall with a specific threshold chosen by the tool. It is by looking at the "auc_precision_recall" that it is possible to understand if the model is good or not. For instance, the "recall/positive_threshold_0.5_mean" found can be low, but if the "auc_precision_recall" is high enough, it is possible to obtain a good model, for the purpose of this work, just by varying the threshold.

Chapter 5

Results

5.1 Introduction

In this chapter, all the results of the simulations for the linear model, the deep neural network model and the wide&deep model are reported.
The values for the output label category "100-1000" and ">1000" are not recorded for both the output label NPI (Number of People Injured) and NPK (Number of People Killed). This is because the database used for evaluation - *i.e.* the ammonia plant accident database - does not contain events with a number of injuries - or deaths - between 100 and 1000 or higher than 1000.

5.2 The Linear Model

The results obtained from the simulations, using the linear model, are presented in the following.

The Number of People Injured (NPI) label is first considered.

Table 5.1 reports the results obtained with the output label category "YES" - indicating events with one or more injuries.
Table 5.2 indicates the values returned from the model for the output label category "NO" - re-

ferring to events with no injuries.
Table 5.3 reports the results for the category "<10" - events with a number of injuries between 1 and 10.
Finally, table 5.4 contains the results for the category "10-100" - categorizing events with a number of injuries between 10 and 100.

Wide model: NPI - YES

STEPS	200	2000	20000	200000
accuracy	0.34375	0.789062	0.742188	0.789062
accuracy/baseline_label_mean	0.203125	0.203125	0.203125	0.203125
auc	0.638575	0.514517	0.445513	0.388198
auc_precision_recall	0.349323	0.279142	0.23533	0.20392
loss	0.808906	0.603909	0.62173	0.952959
precision/positive_threshold_0.5_mean	0.231481	0.4	0.266667	0.333333
recall/positive_threshold_0.5_mean	0.961538	0.0769231	0.153846	0.0384615

Table 5.1: Wide model: evaluation metric values for the output label NPI and output label category YES.

Wide model: NPI - NO

STEPS	200	2000	20000	200000
accuracy	0.34375	0.789062	0.742188	0.789062
accuracy/baseline_label_mean	0.796875	0.796875	0.796875	0.796875
auc	0.638575	0.514517	0.445513	0.388198
auc_precision_recall	0.876248	0.798864	0.750359	0.76012
loss	0.808906	0.603909	0.62173	0.952959
precision/positive_threshold_0.5_mean	0.95	0.804878	0.80531	0.8
recall/positive_threshold_0.5_mean	0.186275	0.970588	0.892157	0.980392

Table 5.2: Wide model: evaluation metric values for the output label NPI and output label category NO.

Wide model: NPI - <10

STEPS	200	2000	20000	200000
accuracy	0.820312	0.8125	0.820312	0.820312
accuracy/baseline_label_mean	0.179688	0.179688	0.179688	0.179688
auc	0.676605	0.579089	0.40766	0.375776
auc_precision_recall	0.269507	0.202159	0.160256	0.141215
loss	0.471002	0.675007	0.569665	1.0105
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.3: Wide model: evaluation metric values for the output label NPI and output label category <10.

Wide model: NPI - 10-100

STEPS	200	2000	20000	200000
accuracy	0.976562	0.984375	0.984375	0.984375
accuracy/baseline_label_mean	0.0234375	0.0234375	0.0234375	0.0234375
auc	0.424	0.814667	0.764	0.729333
auc_precision_recall	0.0454154	0.539458	0.537017	0.366366
loss	3.20941	0.285693	0.165543	0.0919773
precision/positive_threshold_0.5_mean	0	0.666667	1.0	1.0
recall/positive_threshold_0.5_mean	0	0.666667	0.333333	0.333333

Table 5.4: Wide model: evaluation metric values for the output label NPI and output label category 10-100.

The results obtained performing the simulations using the output label NPK are then shown.
Table 5.5 reports the evaluation metric values returned by the model for the category "YES" - classifying events with one or more deaths.
Table 5.6 shows the results for the output label category "NO" - indicating events with no deaths.
Table 5.7 contains the values returned by the model for the category "<10" - referring to events with a number of deaths between 1 and 10.
Finally, table 5.8 includes the results for the category "10-100" - categorizing events with a number of deaths between 10 and 100.

Wide model: NPK - YES

STEPS	200	2000	20000	200000
accuracy	0.796875	0.882812	0.8125	0.898438
accuracy/baseline_label_mean	0.117188	0.117188	0.117188	0.117188
auc	0.8059	0.733333	0.7	0.684956
auc_precision_recall	0.407194	0.309212	0.298231	0.41536
loss	0.578526	0.396275	0.422599	0.38058
precision/positive_threshold_0.5_mean	0.310345	0.5	0.263158	0.666667
recall/positive_threshold_0.5_mean	0.6	0.2	0.333333	0.266667

Table 5.5: Wide model: evaluation metric values for the output label NPK and output label category YES.

Wide model: NPK - NO

STEPS	200	2000	20000	200000
accuracy	0.796875	0.882812	0.8125	0.898438
accuracy/baseline_label_mean	0.882812	0.882812	0.882812	0.882812
auc	0.8059	0.733333	0.7	0.684956
auc_precision_recall	0.969383	0.953774	0.933252	0.923549
loss	0.578526	0.396275	0.422599	0.98058
precision/positive_threshold_0.5_mean	0.939394	0.901639	0.908257	0.909836
recall/positive_threshold_0.5_mean	0.823009	0.976451	0.876106	0.982301

Table 5.6: Wide model: evaluation metric values for the output label NPK and output label category NO.

Wide model: NPK - <10

STEPS	200	2000	20000	200000
accuracy	0.890625	0.867188	0.914062	0.898438
accuracy/baseline_label_mean	0.109375	0.109375	0.109375	0.109375
auc	0.734336	0.709587	0.780389	0.764098
auc_precision_recall	0.240246	0.236979	0.418801	0.356113
loss	0.663478	0.643362	0.295076	0.404255
precision/positive_threshold_0.5_mean	0	0.2	1	1
recall/positive_threshold_0.5_mean	0	0.0714286	0.214286	0.0714286

Table 5.7: Wide model: evaluation metric values for the output label NPK and output label category <10

Wide model: NPK - 10-100

STEPS	200	2000	20000	200000
accuracy	0.992188	0.992188	0.992188	0.898438
accuracy/baseline_label_mean	0.0078125	0.0078125	0.0078125	0.0078125
auc	0.944882	0.948819	0.767717	0.779528
auc_precision_recall	0.0416668	0.0625001	0.016129	0.0172414
loss	0.0862017	0.0646929	0.0920474	0.16875
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.8: Wide model: evaluation metric values for the output label NPK and output label category 10-100

5.3 The Deep Model

The results obtained from the simulations, using the deep model, are presented in the following.

The Number of People Injured (NPI) label is first shown.

Table 5.9 reports the results for the output label category "YES" - indicating events with one or more injuries.
Table 5.10 shows the values returned by the model for the category "NO" - referring to events with no injuries.
Table 5.11 contains the results obtained for the category "<10" - events with a number of injuries between 1 and 10.
Finally, table 5.12 includes the values returned by the model for the category "10-100" - categorizing events with a number of injuries between 10 and 100.

Deep model: NPI - YES

STEPS	200	2000	20000	200000
accuracy	0.526041333	0.747396	0.53125	0.460938
accuracy/baseline_label_mean	0.203125	0.203125	0.203125	0.203125
auc	0.533874	0.644231	0.407428	0.384804
auc_precision_recall	0.241089	0.337787	0.195943	0.258708
loss	0.698439	0.506459	2.87726	6.46264
precision/positive_threshold_0.5_mean	0.221065	0.500418	0.113636	0.135593
recall/positive_threshold_0.5_mean	0.551282	0.141026	0.192608	0.307692

Table 5.9: Deep model: evaluation metric values for the output label NPI and output label category YES

Deep model: NPI - NO

STEPS	200	2000	20000	200000
accuracy	0.596354	0.687500	0.773438	0.484375
accuracy/baseline_label_mean	0.796875	0.796875	0.796875	0.796875
auc	0.627137	0.585596	0.547134	0.439857
auc_precision_recall	0.844169	0.850263	0.884186	0.802515
loss	0.666904	0.590606	2.58352	4.73528
precision/positive_threshold_0.5_mean	0.785204	0.817195	0.806723	0.772727
recall/positive_threshold_0.5_mean	0.660131	0.784314	0.941176	0.5

Table 5.10: Deep model: evaluation metric values for the output label NPI and output label category NO

Deep model: NPI - <10

STEPS	200	2000	20000	200000
accuracy	0.820312	0.820312	0.71875	0.476562
accuracy/baseline_label_mean	0.179688	0.179688	0.179688	0.179688
auc	0.639683	0.558040	0.484886	0.46087
auc_precision_recall	0.299808	0.265903	0.176128	0.255764
loss	0.500608	0.476159	1.98578	3.18913
precision/positive_threshold_0.5_mean	0	0	0.190476	0.176471
recall/positive_threshold_0.5_mean	0	0	0.173913	0.521739

Table 5.11: Deep model: evaluation metric values for the output label NPI and output label category <10

Deep model: NPI - 10-100

STEPS	200	2000	20000	200000
accuracy	0.976562	0.976562	0.960968	0.960938
accuracy/baseline_label_mean	0.0234375	0.0234375	0.0234375	0.0234375
auc	0.608889	0.580889	0.604444	0.637333
auc_precision_recall	0.0912892	0.0439778	0.052894	0.0865163
loss	0.227724	0.158686	0.2422	0.3475583
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.12: Deep model: evaluation metric values for the output label NPI and output label category 10-100.

The results obtained performing the simulations with the output label NPK are then shown.
Table 5.13 reports the values returned performing the simulations using the output label category "YES" - classifying events with one or more deaths.
Table 5.14 contains the results for the category "NO" - indicating events with no deaths.
Table 5.15 shows the evaluation metric values for "<10" - referring to events with a number of deaths between 1 and 10.
Finally, table 5.16 includes the results for the category "10-100" - categorizing events with a number of deaths between 10 and 100.

Deep model: NPK - YES

STEPS	200	2000	20000	200000
accuracy	0.882812	0.783854	0.65625	0.5625
accuracy/baseline_label_mean	0.117188	117188	117188	117188
auc	0.570206	0.63707	0.570502	0.471976
auc_precision_recall	0.184593	0.183196	0.201393	0.200881
loss	0.523201	0.461321	1.98905	2.63829
precision/positive_threshold_0.5_mean	0	0.128395	0.162791	0.0816327
recall/positive_threshold_0.5_mean	0	0.244445	0.466667	0.266667

Table 5.13: Deep model: evaluation metric values for the output label NPK and output label category YES

Deep model: NPK - NO

STEPS	200	2000	20000	200000
accuracy	0.882812	0.773437	0.59375	0.53125
accuracy/baseline_label_mean	0.882812	0.882812	0.882812	0.882812
auc	0.554179	0.473746	0.602065	0.385546
auc_precision_recall	0.888845	0.872665	0.927423	0.891102
loss	0.561255	0.50782	2.59215	4.05464
precision/positive_threshold_0.5_mean	0.882812	0.887428	0.917808	0.835443
recall/positive_threshold_0.5_mean	1	0.852507	0.59292	0.584071

Table 5.14: Deep model: evaluation metric values for the output label NPK and output label category NO

Deep model: NPK - <10

STEPS	200	2000	20000	200000
accuracy	0.890625	0.880208	0.851562	0.789062
accuracy/baseline_label_mean	0.109375	0.109375	0.109375	0.109375
auc	0.636800333	0.688701	0.693609	0.683897
auc_precision_recall	0.251586	0.270186	0.325247	0.215281
loss	0.820374	0.364296	0.988342	1.33151
precision/positive_threshold_0.5_mean	0	0.111111	0.307692	0.217391
recall/positive_threshold_0.5_mean	0	0.095238	0.285714	0.357143

Table 5.15: Deep model: evaluation metric values for the output label NPK and output label category <10

Deep model: NPK - 10-100

STEPS	200	2000	20000	200000
accuracy	0.992188	0.992188	0.921875	0.921875
accuracy/baseline_label_mean	0.0078125	0.0078125	0.0078125	0.0078125
auc	0.417323	0.571654	0.366142	0.354331
auc_precision_recall	0.00942188	0.00989547	0.00390626	0.00390626
loss	0.0841593	0.0746411	0.22798	0.319463
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.16: Deep model: evaluation metric values for the output label NPK and output label category 10-100

5.4 The Wide&Deep Model

The results obtained from the simulations, using the wide&deep model, are presented in the following

The Number of People Injured (NPI) label is first shown.

Table 5.17 shows the results of the simulations with the output label category "YES" - indicating events with one or more injuries.
Table 5.18 contains the values obtained for the category "NO" - referring to events with no injuries.
Table 5.19 reports the results for the category "<10" - events with a number of injuries between 1 and 10.
Finally, table 5.20 includes the evaluation metric values obtained for the category "10-100" - categorizing events with a number of injuries between 10 and 100.

Wide&deep model: NPI - YES

STEPS	200	2000	20000	200000
accuracy	0.523437	0.786458	0.625	0.679688
accuracy/baseline_label_mean	0.203125	0.203125	0.203125	0.203125
auc	0.539844	0.630845	0.392345	0.530354
auc_precision_recall	0.225683	0.330302	0.202903	0.318757
loss	0.654253	0.518993	1.88773	2.68498
precision/positive_threshold_0.5_mean	0.157827	0.607655	0.133333	0.285714
recall/positive_threshold_0.5_mean	0.602564	0.179487	0.153846	0.384615

Table 5.17: Wide&deep model: evaluation metric values for the output label NPI and output label category YES

Wide&deep model: NPI - NO

STEPS	200	2000	20000	200000
accuracy	0.679688	0.809896	0.34375	0.445312
accuracy/baseline_label_mean	0.796875	0.796875	0.796875	0.796875
auc	0.53821	0.588801	0.454563	0.451169
auc_precision_recall	0.851897	0.867517	0.783502	0.788911
loss	0.628977	0.516108	4.40671	5.30077
precision/positive_threshold_0.5_mean	0.78769	0.810667	0.736842	0.803922
recall/positive_threshold_0.5_mean	0.813725	0.993464	0.27451	0.401961

Table 5.18: Wide&deep model: evaluation metric values for the output label NPI and output label category NO

Wide&deep model: NPI - <10

STEPS	200	2000	20000	200000
accuracy	0.820312	0.820312	0.671875	0.6875
accuracy/baseline_label_mean	0.179688	0.179688	0.179688	0.179688
auc	0.586819	0.632781	0.398137	0.355694
auc_precision_recall	0.216702	0.265357	0.133898	0.141654
loss	0.479485	0.451136	2.47009	3.5826
precision/positive_threshold_0.5_mean	0	0	0.0869565	0.0952381
recall/positive_threshold_0.5_mean	0	0	0.0869565	0.0869565

Table 5.19: Wide&deep model: evaluation metric values for the output label NPI and output label category <10

Wide&deep model: NPI - <10

STEPS	200	2000	20000	200000
accuracy	0.976562	0.981771	0.96875	0.914062
accuracy/baseline_label_mean	0.0234375	0.0234375	0.0234375	0.0234375
auc	0.553333	0.738009	0.636	0.576667
auc_precision_recall	0.0412882	0.0497824	0.078183	0.0362891
loss	0.28707	0.188535	0.284812	0.414595
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.20: Wide&deep model: evaluation metric values for the output label NPI and output label category 10-100

The results obtained performing the simulations with the output label NPK are then shown.
Table 5.21 includes the results obtained from the simulations carried out with the output label category "YES" - classifying events with one or more deaths.
Table 5.22 shows the values for the category "NO" - indicating events with no deaths.
Table 5.23 reports the results for the category "<10" - referring to events with a number of deaths between 1 and 10.
Finally, table 5.24 contains the evaluation metric values for the output label category "10-100" - categorizing events with a number of deaths between 10 and 100.

Wide&deep model: NPK - YES

STEPS	200	2000	20000	200000
accuracy	0.882812	0.817771	0.53125	0.75
accuracy/baseline_label_mean	0.117188	0.117188	0.117188	0.117188
auc	0.613471	0.673451	0.494985	0.658112
auc_precision_recall	0.152307	0.207747	0.202501	0.313107
loss	0.295398	0.47763	2.83981	1.44217
precision/positive_threshold_0.5_mean	0.166667	0.244444	0.0909091	0.242424
recall/positive_threshold_0.5_mean	0.0222222	0.288889	0.333333	0.533333

Table 5.21: Wide&deep model: evaluation metric values for the output label NPK and output label category YES

Wide&deep model: NPK - NO

STEPS	200	2000	20000	200000
accuracy	0.757812	0.817708	0.703125	0.523438
accuracy/baseline_label_mean	0.882812	0.882812	0.882812	0.882812
auc	0.750017	0.688594	0.641298	0.374041
auc_precision_recall	0.908476	0.94205	0.94099	0.879091
loss	0.593612	0.451672	1.5891	3.62232
precision/positive_threshold_0.5_mean	0.910115	0.899623	0.894737	0.861111
recall/positive_threshold_0.5_mean	0.80826	0.893805	0.752212	0.548673

Table 5.22: Wide&deep model: evaluation metric values for the output label NPK and output label category NO

Wide&deep model: NPK - <10

STEPS	200	2000	20000	200000
accuracy	0.890625	0.890625	0.804688	0.835938
accuracy/baseline_label_mean	0.109375	0.109375	0.109375	0.109375
auc	0.584378	0.707957	0.593045	0.630639
auc_precision_recall	0.127306	0.224997	0.13449	0.19583
loss	0.415087	0.341036	1.00402	1.05551
precision/positive_threshold_0.5_mean	0	0	0.133333	0.266667
recall/positive_threshold_0.5_mean	0	0	0.142857	0.285714

Table 5.23: Wide&deep model: evaluation metric values for the output label NPK and output label category <10

Wide&deep model: NPK - 10-100

STEPS	200	2000	20000	200000
accuracy	0.992188	0.992188	0.914062	0.976562
accuracy/baseline_label_mean	0.0078125	0.0078125	0.0078125	0.0078125
auc	0.839895	0.699213	0.275591	0.374016
auc_precision_recall	0.0198826	0.0308853	0.00390626	0.00390627
loss	0.0897457	0.0883096	0.26657	0.22804
precision/positive_threshold_0.5_mean	0	0	0	0
recall/positive_threshold_0.5_mean	0	0	0	0

Table 5.24: Wide&deep model: evaluation metric values for the output label NPK and output label category 10-100

Chapter 6

Elaboration and Discussion of the Results

In order to analyze the results and the performance of the models, the area under the curve precision-recall is considered ("auc_precision_recall"). The recall returned by the simulation is, indeed, specific for a certain value of the threshold, chosen by the tool. It is by looking at the area under the curve precision-recall that it is possible to correctly asses the performance of the model: if the value is high, it means that the value of recall achievable, varying the threshold, can be of interest.
Figures 6.1, 6.2, 6.3 and 6.4 shows a comparison between the models, in reference to the label NPK (number of people killed), using 200, 2000, 20000 and 200000 steps respectively.

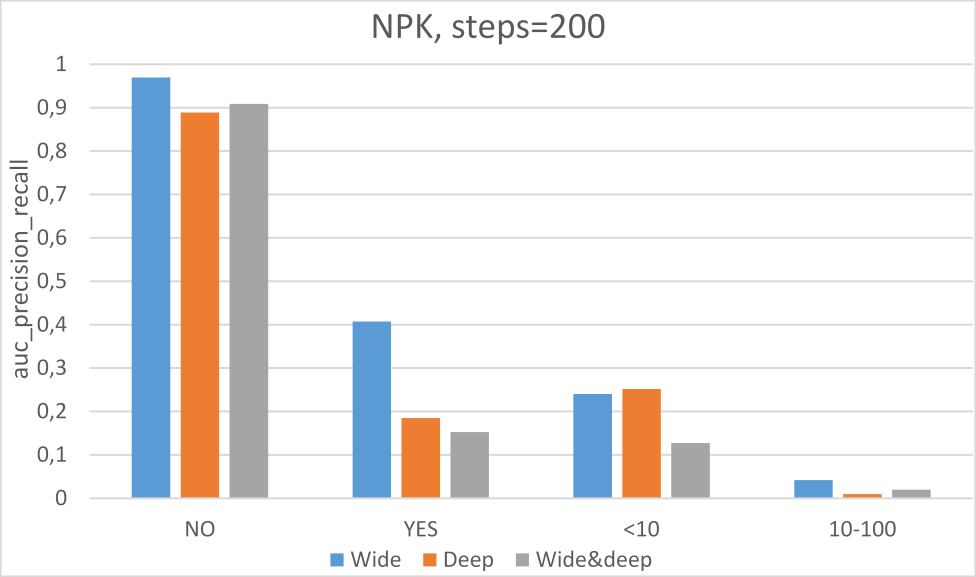

Figure 6.1: Comparison of the three models for the different categories. Output label NPK, iteration steps=200.

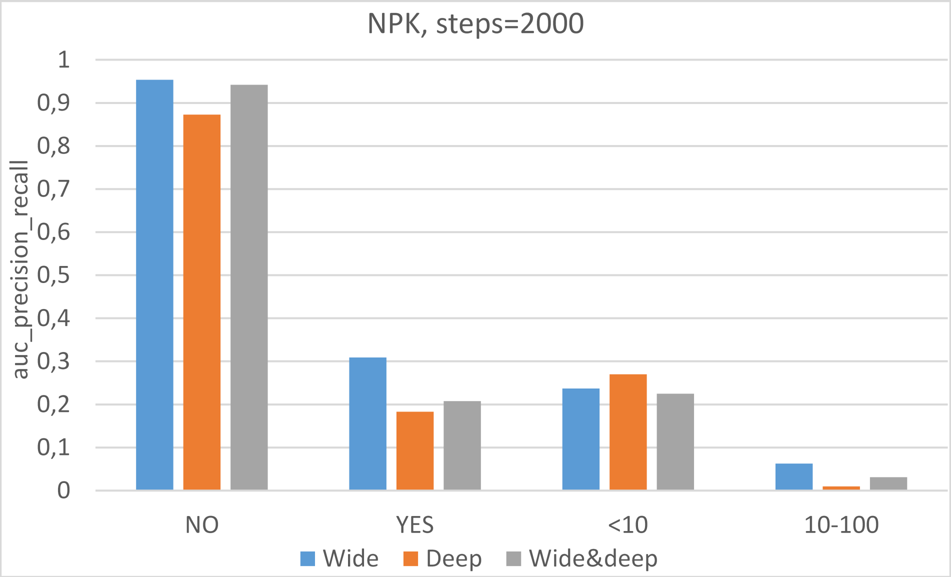

Figure 6.2: Comparison of the three models for the different categories. Output label NPK, iteration steps=2000.

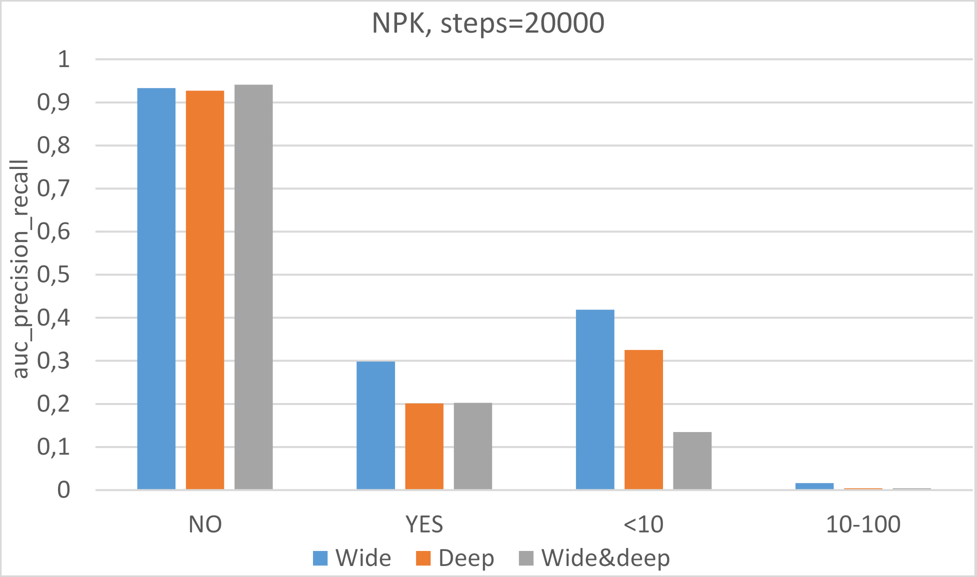

Figure 6.3: Comparison of the three models for the different categories. Output label NPK, iteration steps=20000.

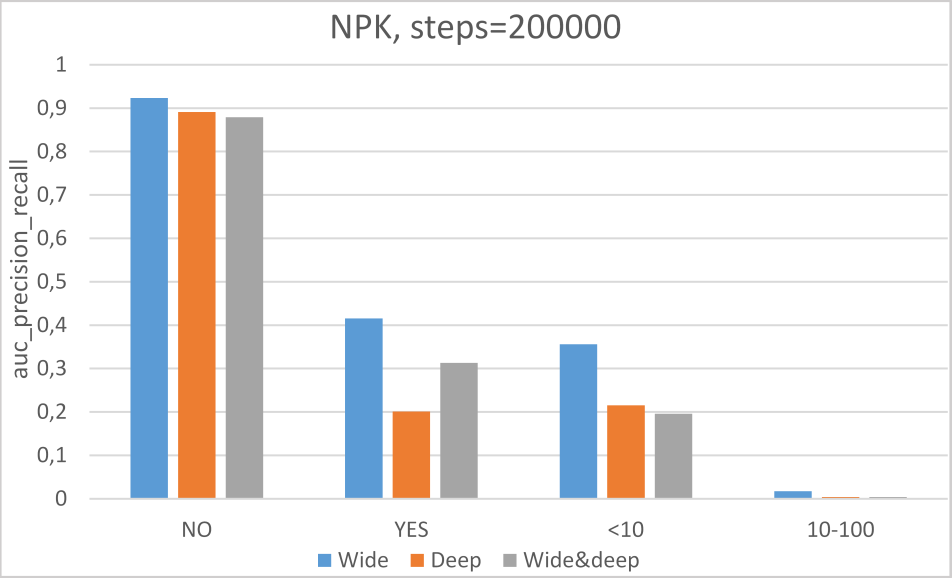

Figure 6.4: Comparison of the three models for the different categories. Output label NPK, iteration steps=200000.

From all the figures, it is possible to notice a common trend: the more the event is rare, the lower the value of the area under the curve precision-recall. This matches the expectations. The prediction skill of a model is, indeed, better for common events. It is more difficult to detect events that have never or rarely happened because there are few chances to learn.
In general, the model which seems to work better is the linear one (wide).
Considering common events - *i.e.* essentially the output label category "NO" - all three models give satisfactory results. For every example taken into account, the wide model is the best. The deep and wide&deep give substantially the same results. The linear model assumes a linear association between the inputs and the event to predict, while the other two have more complex structures. As it is possible to see from a bow-tie diagram, the more a feature is far from the event to predict, the more interactions it has with other features. That is why the deep model and the wide_deep model have been taken into consideration: the output, in these cases, is predicted on the basis of combinations of the inputs. However, the results of the simulations show that these models give worse results than the linear one. A reason can be found in the uncertainty and in the low quality of data collected in the ammonia plant accident database. Many times, some features were not even registered because of the lack of information in the safety reports investigated.
Taking into account rare events the situation does not change: the linear model is, in general, the one which makes better predictions. The only exceptions are for the category "<10" using 200 and 2000 steps. Here, the deep model is the one which works better, followed by the wide and, lastly, by the wide&deep.

Figures 6.5, 6.6, 6.7 and 6.8 shows, instead, the results obtained using the output label NPI for the three models, with, respectively, 200, 2000, 20000 and 200000 iteration steps. All the categories are considered.

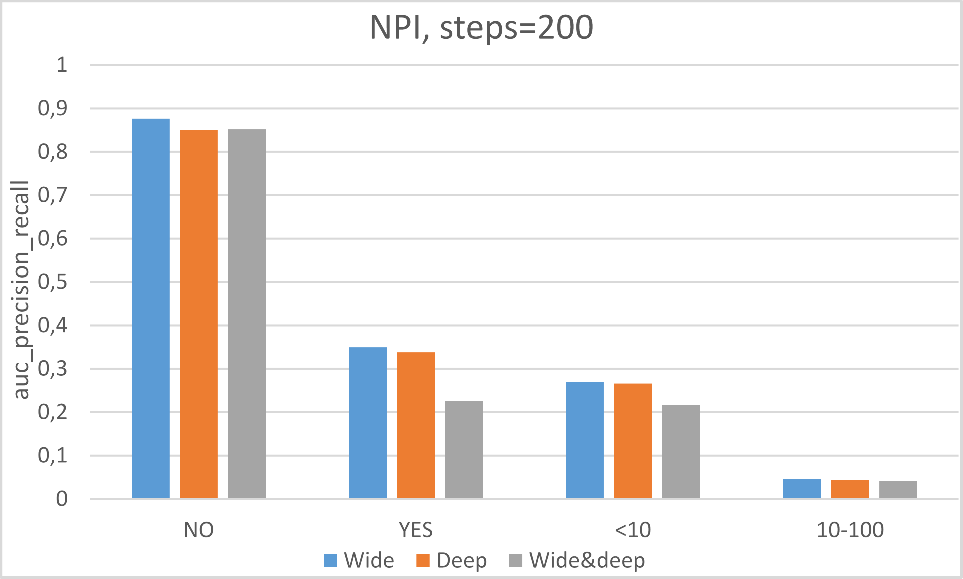

Figure 6.5: Comparison of the three models for the different categories. Output label NPI, iteration steps=200.

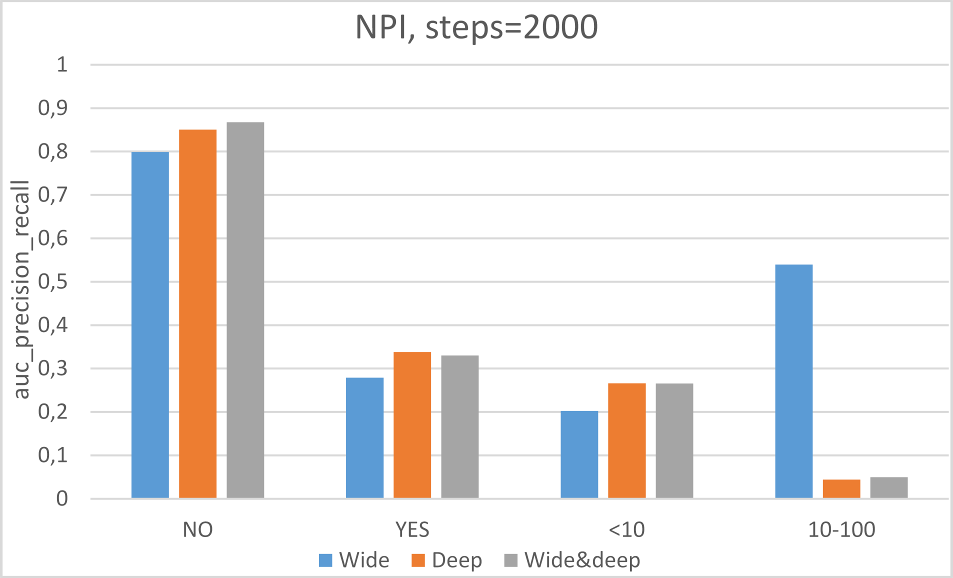

Figure 6.6: Comparison of the three models for the different categories. Output label NPI, iteration steps=2000.

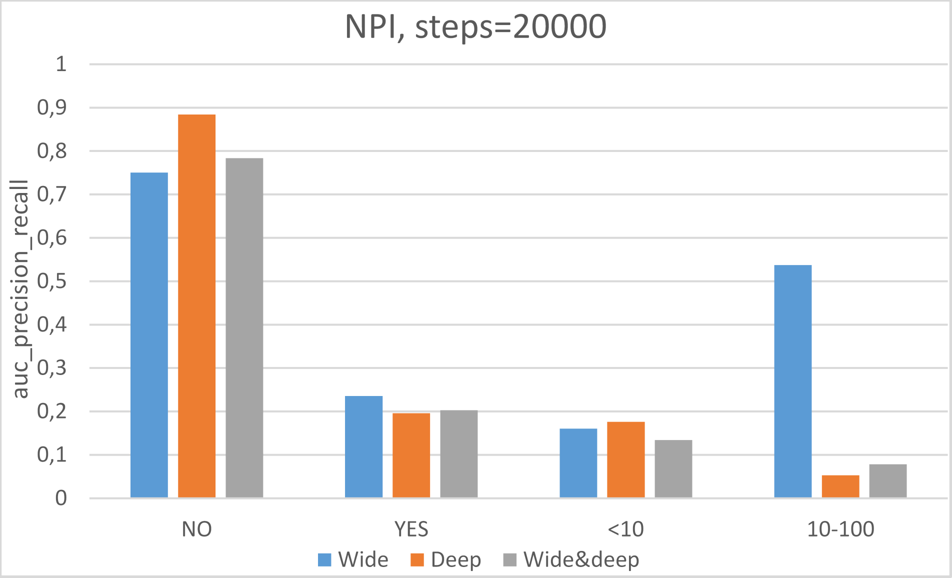

Figure 6.7: Comparison of the three models for the different categories. Output label NPI, iteration steps=20000.

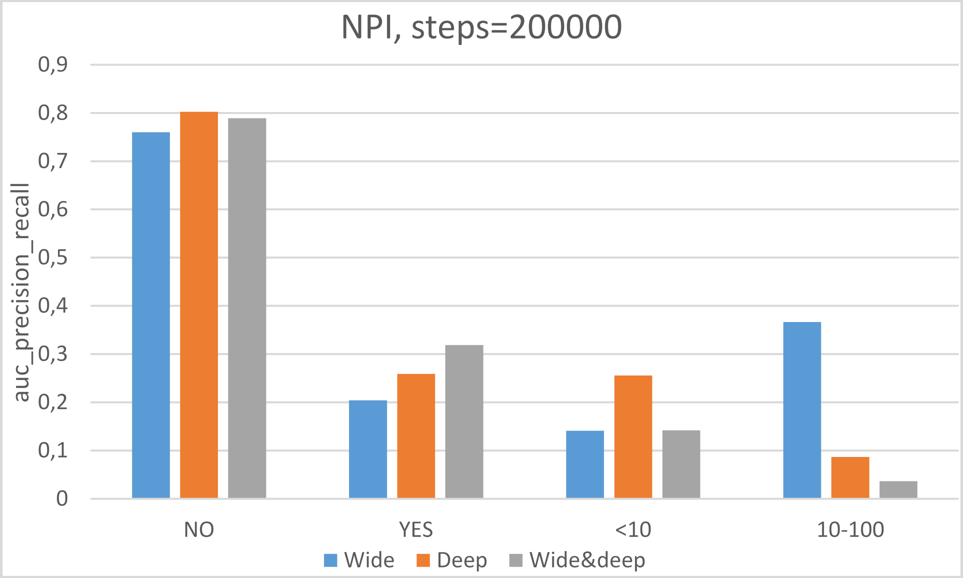

Figure 6.8: Comparison of the three models for the different categories. Output label NPI, iteration steps=200000.

The expected trend is the same one described for the label NPK. However, this does not always reflects the results.
Considering the deep and wide&deep models, there is no difference from what already stated above for the NPK label: the more the event is rare the lower the value of "auc_precision_recall". For the linear model, instead, using 2000, 20000 and 200000 steps, this metric decreases going from the "NO", to the "<10" category and then it registers high values for the category "10-100". This is probably due to the lack of events in the ammonia plant accident database, used for the evaluation. Here are only two events registered as "10-100" for the output label NPI. Detecting one only would make a big difference in the final results. Looking carefully at the database, it is possible to notice that one of the two is associated with a high number of people evacuated (NPE). In the linear model, every input is linearly correlated to the output (NPI in this case), with a certain weight defined during the training. It's likely that, because of the high number of people evacuated associated to the event, the model detected that record, correctly, as positive - *i.e.* belonging to the category "10-100". The deep and the wide&deep model, on the other hand, because of their non-linearity between input and output, were not able to detect them - with the specific threshold considered - as it is possible to see from the recall value returned equal to 0. Thus, their "auc_precision_recall" value is much lower than the one obtained with the linear model. It is important to underline, also, that only in case of 2000, 20000 and 200000 iteration steps the wide one has high value of the area under the curve precision-recall. This means that for a low number of steps, also this model is not able to detect the events belonging to the category "10-100".
In general the wide model works well also for the label "NPI" but it is not always the best one. for number of iteration steps=200, it is the one which predicts better. However, using 2000, 20000 and 200000 steps, the situation changes. In these 3 cases, considering the categories "NO", "YES" and "<10", the deep and the wide&deep give similar results - better than the ones obtained from the linear. In general, the deep model is a bit better than the wide&deep. For the category "<10", as already stated above, the wide model is able to detect one of the two events registered in the database used for evaluation, so it gives much better results.

Another important argument to consider is that, even though the value of recall returned by the model is low or equal to 0, it does not necessarily mean that the model is not good enough for the purpose of this work. As already mentioned in Chapter 4 - section 4.4.1, this is the reason why "auc_precision_recall" is considered: if this is high enough it is possible to obtain good values of the recall just by varying the threshold. For instance, let's consider the simulation carried out using the wide model and choosing NPI, as output label, <10, as output category and a number o iteration steps equal to 2000. The "recall/positive_threshold_0.5_mean" returned is 0, while the "auc_precision_recall" value is 0.202159. Figure 6.9 reports the curve precision recall for this particular case, using threshold equal to: 0; 0,06; 0,08; 0,1; 0,12; 0,14; 0,16; 0,18; 0,2; 0,22; 1.

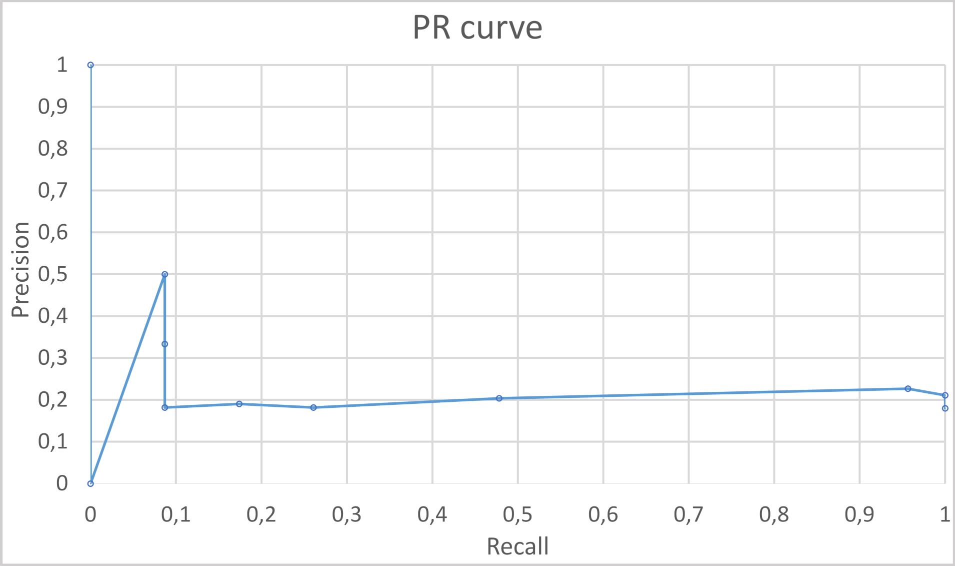

Figure 6.9: Precision-Recall curve (PR curve) for output label NPI, output category <10, iteration steps=2000. Wide model.

The values returned from the model corresponds to the point of coordinates (0;0) in the graph - precision=0;recall=0. This is due to the default configuration of the model, which aims to give high value of accuracy. However, the purpose of this work is to predict accidents, being conservative on the prediction: a high value of the recall is then preferred, even if it could lead to lower values of accuracy and precision. Decreasing the threshold to a value of 0.08, for instance, improves the situation: it brings the value of precision at around 0.2 and the value of recall at around 0.95. All of this means that, using the threshold chosen by the tool, the model could not predict any of the events belonging to the category "<10". Varying the threshold to 0.08, the same model is able to detect almost all of these events.

6.1 Limitations of the Work

The number of accident collected in the ammonia plant accident database are not enough to give a fully correct evaluation of the models.

The lack of detail of the safety reports influenced the quality of the data in the final database and, consequently, the quality of the simulations. It was difficult to comprehend the real information reported. Consequently, the data collected in the ammonia plant accident database, are affected by the personal interpretation of the writer, with respect to the content of the reports.

Chapter 7

Conclusions

Analyzing data and extract knowledge from them has become an important need. Especially in the industrial sector, with the so-called industry 4.0, a big amount of data is set to be collected. This should be seen as an opportunity to learn: not exploiting them would be a waste. The Seveso III Directive, whose focus is laying down the rules to prevent major accidents in establishments handling hazardous substances, also advances the necessity to monitor data in order to improve the safety management system of the plants.

In this work, a way to manage and analyze heterogeneous data from chemical process industry, in order to obtain important information for establishing the risk picture, has been proposed.

The establishment taken into consideration for this work is the ammonia plant, which falls under the Seveso III DIrective. Information about past accidents occurred in these sites have been collected from different sources and they have been stored in a common database. Data have been registered using keywords, in order to make them easier to analyze. The ammonia plant accident database have been used for evaluating three different models - a linear one, one based on deep learning and a combination of the two - trained with the past accident data stored in the MHIDAS database. The aim of the model is to predict the consequence on human - in terms of number of people injured or number of people killed - on the basis of information about past accidents. The machine learning tool used, which provided the instruments to build the models, is the open source library TensorFlow.

The value of the recall and the area under the curve precision-recall have been used as primary metrics to asses the performance of the models. One of the purposes of this work is, indeed, to find a model able to detect all the events, also the rarest ones, even if the "precision" of the model will not result high. In order to meet this need, the value of the threshold could be lowered. That is why it has been important to consider the value of the area under the curve precision-recall: even though the model, in some cases, returned low values of the recall, the value of the area under the curve was still good. This was an indicator of the fact that, lowering the threshold, the value of the recall would have increased and become of interest.

The general trend of the results showed that the value of the area under the curve tended to decrease the more the events considered in the simulations were rare, as expected. Above all, the linear model gave the best results. Because of the linear dependence between the inputs, weighted on the model parameters tuned during the training, and the output, it showed good prediction skills also in case of rare events.

Appendix A

Acronyms

AI Artificial Intelligence

API Application Programming Interface

DOL (United States) Department of Labor

DRA Dynaic Risk Analysis

EEA European Economic Area

EEC European Economic Community

eMARS Major Accident Reporting System

EPA Environmental Protection Agency

EU Eurpean Union

HAZOP Hazard and operability study

HILP High Impact Low Probability

HSE Health and Safety Executive

ICMESA Industrie Chimiche Meda Società Azionaria

JFKD Japanese Failure Knowledge Model

JRC (European Commssion's) Joint Research Center

JST Japanese Science and Technology

KDD Knowledge Discovery from Data

MAHB Major Accident Hazards Bureau

MAPP Major-Accident Prevention Policy

MHIDAS Major Hazard Incident Data Service

NRC National Response Center

NRS National Response System

OECD Organization for Economic Co-operation and Development

OSHA Occupational Safety and Health Administration

QRA Quantitative Risk Assessment

SPI Safety Performance Indicator

TCB 1,2,4,5-tetrachlorobenzene

TCDD 2,3,7,8-tetrachlorodibenzodioxin

TCP 2,4,5-trichlorophenol

TEIA Transboundary Effects of Industrial Accidents

UBA Umweltbundesamt

UNECE United Nations Economic Commission for Europe

USCG United States Coast Guard

ZEMA Zentrale Melde-und Auswertestelle (für Störfälle und Störungen)

Appendix B

Ammonia Plant Accident Database

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
11/07/1959	Ube, Japan	Oxygen, Syngas	EXPLODE; FIRE	PROCESS - PVESSEL	PROCOND	GENERALOP	44	0	11	240 M yen	CO2 REMOVAL	
1965	Pasadena, TX	Hydrogen; Ammonia	FIRE				3	0	2			
29/12/1966	LUDWIGSHAFEN; GERMANY	AMMONIA, HYDROGEN	RELEASE	PROCESS-PIPEWORK; HEATXCHANG	HUMAN; MECHANICAL	DESIGN	62	0	0			
18/02/1970	DEER PARK; TEXAS; USA	HYDROGEN	EXPLODE	PROCESS-PVESSEL	INSTRUMENT; HUMAN	DESIGN	1	0	0		REFORMING	
21/12/1971	ELLESMERE PORT; CHESHIRE; UK	SYNGAS	RELEASE, FIRE	PROCESS-PUMP	MECHANICAL - EXTERNAL	FLANGCOUPL - EXTNLFIRE	1	0	0	US$ 0.50 x 10E6	REFORMING	
09/07/1972	SAKAI CITY; OSAKA; JAPAN	Ammonia	RELEASE	PROCESS-PIPEWORK	MECHANICAL	CORRODE	0	0	0		AMMONIA SYNTHESIS	
26/09/1973	Kawasaki, Kanagawa, Japan	Hydrogen	RELEASE; FIRE	PROCESS - PVESSEL	HUMAN; MECHANICAL	MAINTAIN; FLANGCOUPL	0	0	0		DESULFURIZATION	
02/10/1973	CAMROSE; ALBERTA; CANADA	HYDROGEN SULPHIDE	RELEASE	PROCESS-PIPEWORK	MECHANICAL	OVERPRES	0	800	0			
16/09/1978	Immingham, UK	Syngas	RELEASE; EXPLODE				0	0	0			
01/04/1980	Tokuyama City, Yamaguchi Pref.	Nitrogen	EXPLODE	PROCESS - PVESSEL	MECHANICAL; HUMAN	BRITTLE; MAINTAIN; COMPAIR	0	0	0		DESULFURIZATION	
01/09/1981	Czechoslovakia	Syngas	RELEASE; EXPLODE				0	0	0			
01/10/1981	CZECHOSLOVAKIA	SYNGAS	RELEASE, FIRE	PROCESS-PVESSEL	HUMAN; MECHANICAL	MAINTAIN; OVERPRES	29	0	6		REFORMING	12
05/01/1982	FEDMIS; SOUTH AFRICA	HYDROGEN, AMMONIA	EXPLODE; FIRE	PROCESS-PVESSEL			0	0	0		REFORMING	
11/11/1982	LAKE CHARLES; CALIFORNIA; USA	NITROGEN, HYDROGEN	EXPLODE; FIRE	PROCESS-PIPEWORK			1	0	0			
01/07/1984	CHICAGO; ILLINOIS; USA	MEA	RELEASE, EXPLODE	PROCESS-PVESSEL	MECHANICAL; HUMAN	WELDFAIL - DESIGN	17	0	17	US$ 100.00 x 10E6	H2S REMOVAL	<85
07/06/1905	NORWAY	HYDROGEN	EXPLODE, FIRE	PROCESS-PUMP	HUMAN, MECHANICAL	GENERAL OP, DESIGN	1	0	2		CO2 REMOVAL	7
06/07/1985	Clinton, IA	Syngas	RELEASE; EXPLODE				0	0	0	14,7 M $		
21/01/1986	Kawasaki, Kanagawa, Japan	Fuel	RELEASE; EXPLODE	PROCESS - FIREDEQUIP	HUMAN	GENERALOP; VALVE; DESIGN	1	0	0		REFORMING	
November, 1986	LOUISIANA	SYNGAS		PROCESS-PVESSEL	MECHANICAL		0	0	0		REFORMING	
November, 1986	CANADA	SYNGAS	EXPLODE	PROCESS-PVESSEL	MECHANICAL	OVERHEAT	0	0	0		REFORMING	
04/10/1987	Sakai, Osaka, Japan	Syngas; Naphta	RELEASE; FIRE	PROCESS - PIPEWORK	MECHANICAL	FLANGCOUPL	1	0	0		REFORMING	
23/12/1987		Syngas; Natural gas; Ammonia	EXPLODE; FIRE; RELEASE	PROCESS - PIPEWORK	MECHANICAL	METALLURG	0	0	0	0,65 M euro	CO2 REMOVAL	
09/06/1988		Hydrogen; Nitrogen	EXPLODE	PROCESS - PVESSEL	HUMAN; MECHANICAL	MAINTAIN; OVERPRES	1	0	1		AMMONIA SYNTHESIS	
23/02/1989		Hydrogen; Nitrogen	RELEASE; EXPLODE	PROCESS - PIPEWORK	MECHANICAL	MAINTAIN; INCOMPAT	0	0	2		AMMONIA SYNTHESIS	
20/03/1989	Jonova, LITHUANIA	Ammonia	RELEASE, FIRE	STORAGE-TANKCONTNR	HUMAN	GENERAL OP	57	32000	7		STORAGE	7000
13/09/1989		Ammonia	RELEASE	STORAGE - PSVESSEL			6	0	0		STORAGE	
28/03/1990		Ammonia	RELEASE	PROCESS - PIPEWORK	HUMAN	MAINTAIN; GENERAL	1	0	0			

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
29/05/1990	Columbus, GA	Ammonia	RELEASE		HUMAN	GENERAL	0	0	0			
06/06/1990	Pollock, LA	Ammonia	RELEASE	PROCESS - PUMP	MECHANICAL		0	0	0			0,159
17/08/1990	Arita, Wakayama, Japan	Syngas; Naphta	RELEASE; FIRE	PROCESS - PIPEWORK	MECHANICAL	FLANGCOUPL	0	0	0	1 M yen	REFORMING	
31/08/1990		Ammonia	RELEASE	STORAGE - PSVESSEL	HUMAN	GENERAL	3	3 (operators evacuated the tower and washed themselves)	0		STORAGE	
02/11/1990	Enid, OK	Ammonia	RELEASE	PROCESS - PUMP	MECHANICAL	GLANDSEAL	0	0	0			1,06
03/01/1991	Luling, LA	Syngas		PROCESS - PVESSEL			0	0	0		REFORMING	
25/01/1991	Donaldsonville, LA	Ammonia		PROCESS - PIPEWORK	MECHANICAL		0	0	0			0,068
13/06/1905	FRANCE	Ammonia	RELEASE	PROCESS-PVESSEL	MECHANICAL, HUMAN	WELDFAIL	0	0	0		AMMONIA SYNTHESIS	
04/02/1991	Bayonne, NJ	Ammonia	FIRE				0	0	0			
19/02/1991	Geismar, LA	Ammonia		PROCESS - PVESSEL	MECHANICAL		0	0	0			
13/06/1905	GERMANY	Ammonia	RELEASE	PROCESS-PVESSEL	MECHANICAL, HUMAN	WELDFAIL	0	0	0		AMMONIA SYNTHESIS	
19/03/1991	Kawasaki, Kanagawa, Japan	Syngas	RELEASE; EXPLODE; FIRE	PROCESS - HEATXCHANG	PROCOND; MECHANICAL	FLANGCOUPL	0	0	0			
02/04/1991	Westwego, LA	Ammonia	RELEASE				0	0	0			
09/04/1991	Lake Charles, LA	Ammonia	RELEASE				0	0	0			0,068
26/04/1991	Enid, OK	Ammonia	RELEASE	PROCESS - PUMP	MECHANICAL		0	0	0			0,103
05/05/1991	Montebello, CA	Syngas					0	0	0			0,045
08/05/1991		Ammonia	FIRE; RELEASE	PROCESS - PIPEWORK	MECHANICAL	CORRODE; VALVE	0	0	0		AMMONIA SYNTHESIS	20
11/05/1991	Beaumont, TX	Syngas	RELEASE	PROCESS - PUMP	MECHANICAL		0	0	0			2,268
17/05/1991	Donaldsonville, LA	Ammonia	RELEASE	PROCESS - PUMP	MECHANICAL		0	0	0			0,68
24/05/1991	Donaldsonville, LA	Ammonia		PROCESS - PIPEWORK	MECHANICAL		0	0	0			0,068
13/08/1991	Enid, OK	Ammonia	RELEASE				0	0	0			0,053
06/11/1991	Catoosa, OK	Ammonia	RELEASE	HOSE	MECHANICAL		0	0	0			0,204
19/03/1992	Geismar, LA	Ammonia					0	0	0			
09/04/1992	Geismar, LA	Ammonia		PROCESS - PIPEWORK	PROCOND		0	0	0			
01/06/1992		Ammonia	FIRE; RELEASE	PROCESS - PIPEWORK	MECHANICAL	METALLURG	0	0	0		AMMONIA SYNTHESIS	0,6
19/06/1992	Geismar, LA	Ammonia	RELEASE				0	0	0			0,045
August, 1992	ENGLAND			PROCESS-PVESSEL	MECHANICAL	MAINTAIN, DESIGN	0	0	0		REFORMING	

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
03/08/1992	Geismar, LA	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			0,045
02/09/1992	Donaldsonville, LA	Ammonia		PROCESS - PIPEWORK			0	0	0			
16/10/1992	Sodegaura-city, Chiba prefecture	Syngas	RELEASE; EXPLODE; FIRE	PROCESS - HEATXCHANG	HUMAN; MECHANICAL	MAINTAIN; FLANGECOUPL	7	0	10			
21/12/1992	Waggaman, LA	Ammonia	FIRE; RELEASE	PROCESS - PVESSEL		INTNLFIRE	0	0	0		REFORMING	
08/02/1993	GERMANY			PROCESS-PIPEWORK	MECHANICAL	DESIGN, WELDFAIL	0	0	0		AMMONIA SYNTHESIS	
05/05/1993	PAKISTAN	SYNGAS	RELEASE	PROCESS-HEATXCHANG	MECHANICAL	METALLURG	0	0	0		AMMONIA SYNTHESIS	
01/08/1993	IJMUIDEN; NETHERLANDS	Ammonia	EXPLODE	PROCESS			0	0	0	US$ 3.80 x 10E6		
14/09/1993	NETHERLANDS	HOT AIR		PROCESS-PIPEWORK	PROCOND	MAINTENANCE	0	0	0		AMMONIA SYNTHESIS	
29/09/1993	GERMANY	SYNGAS	RELEASE	PROCESS-PIPEWORK	MECHANICAL	METALLURG, MAINTAIN	0	0	0		AMMONIA SYNTHESIS	
July, 1994	CANADA	SYNGAS	RELEASE	PROCESS-PVESSEL	MECHANICAL	OVERHEAT	0	0	0		REFORMING	
18/08/1994	Kawasaki, Kanagawa, Japan	HYDROGEN, AMMONIA	RELEASE; FIRE	PROCESS - PIPEWORK	HUMAN; MECHANICAL	MAINTAIN; FLANGECOUPL	0	0	0	10000 yen	DESULFURIZATION	
19/11/1994	TEXAS	SYNGAS	RELEASE	PROCESS-PUMP	MECHANICAL	VALVE, DESIGN	0	0	0		REFORMING	
01/12/1994	Ribercourt.dreslincourt, FRANCE	Ammonia	RELEASE	PROCESS-PIPEWORK	HUMAN	MAINTAIN, DESIGN	2	0	1			5
25/04/1995	INDIA	SYNGAS, AMMONIA	RELEASE, FIRE	PROCESS-PVESSEL	MECHANICAL	WELDFAIL	0	0	0		AMMONIA SYNTHESIS	
18/06/1905	GEORGIA, USA			PROCESS-PVESSEL	MECHANICAL	METALLURG	0	0	0		AMMONIA SYNTHESIS	
05/03/1996	INDIA	OIL	RELEASE, FIRE	PROCESS-MACDRIVE, PIPEWORK	MECHANICAL		0	0	0		AMMONIA SYNTHESIS	
19/04/1996		Ammonia	RELEASE	STORAGE - PIPEWORK	HUMAN	MAINTAN	0	0	2		STORAGE	
01/05/1996	LA, USA	SYNGAS	RELEASE, FIRE	PROCESS-PIPEWORK	HUMAN	MAINTAIN, GENERAL OP	0	0	0		AMMONIA SYNTHESIS	
14/05/1996	PORT LISAS; TRINIDAD & TOBAGO	Ammonia	RELEASE	PROCESS			13	0	0			
13/04/1997	NORWAY	SYNGAS	EXPLODE	PROCESS-PIPEWORK	HUMAN	GENERAL OP, MAINTAIN	0	0	0		CO2 REMOVAL	
09/05/1997	QUATAR	SYNGAS		PROCESS-MACDRIVE	PROCOND; MECHANICAL		0	0	0		REFORMING	
July, 1997	MALAYSIA	CATALYST	RELEASE	PROCESS-PVESSEL	MECHANICAL	METALLURG	0	0	0		AMMONIA SYNTHESIS	
01/08/1997	OTTMARSHEIM, FRANCE	Ammonia	RELEASE	PROCESS-PIPEWORK	PROCOND	ELECTRIC	0	0	0		AMMONIA SYNTHESIS	
04/08/1997	Toluouse, FRANCE	SYNGAS	RELEASE		EXTERNAL	ELECTRIC	0	0	0		AMMONIA SYNTHESIS	
October, 1997	SOUTH AFRICA	FUEL	EXPLODE	PROCESS-PVESSEL	HUMAN	GENERAL OP, DESIGN	2	0	0		REFORMING	
07/11/1997	INDIA	WATER	RELEASE	PROCESS-PIPEWORK	MECHANICAL	MAINTAIN, GENERAL OP	0	0	0		REFORMING	
01/12/1997	LOUISIANA	SYNGAS	FIRE	PROCESS-PVESSEL	MECHANICAL	OVERHEAT, WELDFAIL	0	0	0		REFORMING	

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
21/12/1997		Fuel; Oxygen; Steam; Syngas	RELEASE; FIRE	PROCESS - PVESSEL	MECHANICAL	FLANGCOUPL	2	0	0		REFORMING	Hydrocarbon: 42; Oxygen: 46; CO: 9; Hydrogen: 26
27/03/1998	Toluouse, FRANCE	Ammonia	RELEASE	PROCESS-HEATXCHANG	HUMAN	DESIGN	0	0	0		STORAGE	10
20/06/1905	PAKISTAN	CATALYST	RELEASE	PROCESS-PVESSEL	MECHANICAL		0	0	0		AMMONIA SYNTHESIS	
27/05/1998	Kashima, Ibaragi, Japan	Syngas	RELEASE; EXPLODE; FIRE	PROCESS - PVESSEL			6	0	1	18 M yen	DESULFURIZATION	
08/07/1998	INDIA	MDEA, HYDROGEN	RELEASE, FIRE	PROCESS-PIPEWORK	MECHANICAL, INSTRUMENT	METALLURG	0	0	0		CO2 REMOVAL	
29/07/1998	CANADA	NATURAL GAS	FIRE	PROCESS-PIPEWORK	PROCOND; MECHANICAL	DESIGN	0	0	0		REFORMING	
31/08/1998	BANGLADESH	WATER, AMMONIA, SYNGAS		PROCESS-PVESSEL	MECHANICAL	DESIGN, METALLURG	0	0	0		CO2 REMOVAL	
16/11/1998	CANADA	SYNGAS		PROCESS-PIPEWORK	HUMAN	GENERAL OP, WELDFAIL	0	0	0		REFORMING	
24/04/1999	INDIA	SYNGAS	FIRE	PROCESS-PIPEWORK	PROCOND; MECHANICAL	DESIGN	0	0	0		REFORMING	
01/05/1999		CATALYST	RELEASE	PROCESS-HOSE	PROCOND		0	0	0		AMMONIA SYNTHESIS	8500
17/06/1999	INDONESIA	SYNGAS	FIRE, EXPLODE	PROCESS-PVESSEL	PROCOND	BRITTLE,DESIGN	0	0	0		AMMONIA SYNTHESIS	
20/08/1999	ALASKA	SYNGAS	FIRE, EXPLODE	STORAGE	HUMAN	GENERAL OP, DESIGN	3	0	0		CO2 REMOVAL	
09/05/2000	QUATAR	STEAM		PROCESS-PUMP	EXTERNAL; MECHANICAL		0	0	0		REFORMING	
24/05/2000		Syngas	EXPLODE; FIRE	PROCESS - PVESSEL	MECHANICAL; HUMAN	WELDFAIL; MAINTAIN	8	0	3		CO2 REMOVAL	
03/09/2000		Hydrogen; Hydrogen sulfide	EXPLODE; FIRE; RELEASE	PROCESS - PUMP	HUMAN; PROCOND	GENERAL; DESIGN; GENERALOP	1	0	0	13,72 M euro (material damage); 68,6 M euro (production loss)	DESULFURIZATION	
02/02/2001		Natural gas; Syngas	RELEASE; FIRE	FIREDEQUIP	HUMAN	MAINTAIN	5	0	2			
15/04/2002	Tomokomai, Hokkaido, Japan	Hydrogen sulphide; Hydrogen	RELEASE; FIRE	PROCESS - PIPEWORK	MECHANICAL	CORRODE	0	0	0	7-8 billion yen	DESULFURIZATION	
11/08/2002	Büttel, Schleswig-Holstein	Ammonia	RELEASE	STORAGE - PSVESSEL	PROCOND; INSTRUMENT	DESIGN; GENERAL; OVERPRES	0	0	0		STORAGE	450
17/11/2002		Syngas; Fuel	RELEASE; FIRE	PROCESS - HEATXCHANG	HUMAN	CONSTRUCT	1	0	0	4,3 M euro (material damage); 1,7 (production loss)	REFORMING	
11/04/2003		Ammonia	EXPLODE; FIRE; RELEASE	PROCESS - PVESSEL	PROCOND	OVERPRES	0	100	1		AMMONIA SYNTHESIS	
28/05/2003	Köln-Worringen, Nordrhein-Westfalen	Arsenic oxide	RELEASE	PROCESS - HEATXCHANG			0	0	0			0,75
17/03/2004	Büttel, Schleswig-Holstein	Ammonia	RELEASE	PROCESS - PIPEWORK	MECHANICAL		0	0	0			9

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
26/06/2004		Hydrogen sulfide; Sulfur dioxide	FIRE	PROCESS - FIREDEQUIP	MECHANICAL	METALLURG	2	600	0	6 M euro (material damage); 22,5 M euro (production loss)	DESULFURIZATION	
16/08/2004	Köln-Worringen, Nordrhein-Westfalen	Arsenic oxide	RELEASE	PROCESS - PIPEWORK	MECHANICAL; HUMAN	DESIGN	0	0	0			0,22
04/01/2005		Ammonia	EXPLODE; RELEASE	STORAGE - PSVESSEL	PROCOND	GENERALOP	1	0	1	1 million $	STORAGE	
19/02/2005	Coffeeyville, KS	Ammonia	RELEASE	PROCESS - HEATXCHANG			0	0	0			0,045
23/05/2005	Fort Dodge, IA	Ammonia	RELEASE	PROCESS - PIPEWORK		OVERPRES	0	0	0			
31/05/2005	Claremore, OK	Ammonia	RELEASE	PROCESS - PIPEWORK		OVERPRES	0	0	0			0,161
11/06/2005	Claremore, OK	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			0,063
24/06/2005	Fort Dodge, IA	Ammonia	FIRE				0	0	0			0,045
28/06/2005	Coffeeyville, KS	Ammonia	RELEASE				0	0	0			1,113
12/07/2005	Fort Dodge, IA	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			
31/08/2005	Donaldsonville, LA	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			0,227
12/09/2005	Claremore, OK	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			0,506
20/11/2005	Salt Lake City, UT	Hydrogen sulfide	RELEASE				0	0	0			0,045
24/04/2006	Gonfreville l'orcher, FRANCE	SYNGAS	RELEASE	PROCESS-PIPEWORK	HUMAN	FLANGCOUPL,MAINTAIN	0	0	0	300000 euro	REFORMING	
01/06/2006		Syngas	RELEASE; FIRE; EXPLODE	PROCESS - PIPEWORK	human	glandseal	2	0	0	2 M euro	REFORMING	1,45
26/11/2006	GRANDPUITS-BAILLY-CARROIS, FRANCE	Ammonia	RELEASE	PROCESS-PIPEWORK	HUMAN	GENERAL OP, MAINTAIN	0	0	0		STORAGE	
29/11/2006	GRANDPUITS-BAILLY-CARROIS, FRANCE	SYNGAS	EXPLODE	PROCESS-PIPEWORK	HUMAN	FLANGCOUPL, GENERAL OP	0	0	0		REFORMING	
23/08/2007		Syngas	RELEASE	PROCESS - PIPEWORK	HUMAN; INSTRUMENT	DESIGN; GENERAL	0	0	1		REFORMING	
26/06/2009		Natural gas	EXPLODE	PROCESS - FIREDEQUIP	EXTERNAL; HUMAN	GENERAL; DESIGN	2	0	0		REFORMING	
03/07/2009	NANCY, FRANCE	Ammonia	RELEASE		INSTRUMENT	CONTROL	0	0	0		AMMONIA SYNTHESIS	2,5
22/07/2009	GRANDPUITS-BAILLY-CARROIS, FRANCE	Ammonia	RELEASE	PROCESS-HOSE	INSTRUMENT	CONTROL	35	0	0		STORAGE	
07/11/2009	Gonfreville l'orcher, FRANCE				EXTERNAL, HUMAN	ELECTRIC, DESIGN	0	0	0		ALL	
11/04/2010	Vatva GIDC	Ammonia	EXPLODE	PROCESS - PVESSEL	PROCOND; INSTRUMENT	OVERHEAT; OVERPRES	12	0	1			
15/04/2010	GRANDPUITS-BAILLY-CARROIS, FRANCE	NATURAL GAS	FIRE	PROCESS-PIPEWORK	MECHANICAL		0	0	0		REFORMING	
28/06/2010		Steam	EXPLODE	PROCESS - PIPEWORK	MECHANICAL	METALLURG	0	0	0	2 M euro	REFORMING	
24/07/2010		Syngas; Natural gas	EXPLODE; FIRE; RELEASE	PROCESS - PIPEWORK	MECHANICAL	GLANDSEAL; VALVE	5	0	0	12 M euro	AMMONIA SYNTHESIS	Hydrogen: 2,5; Natural gas: less than 10

Date	Location	Substance	Incident type	Origin	General cause	Specific cause	Injured	Evacuated	Killed	Damage	Section	Quantity (ton)
02/04/2011	Brunsbüttel, Schleswig-Holstein	Ammonia	RELEASE	PROCESS - PIPEWORK	MECHANICAL	CORRODE	0	0	0			25
09/11/2011	Ludwigshafen, Rheinland-Pfalz	Syngas; Natural gas	FIRE	PROCESS - PIPEWORK	HUMAN	MAINTAIN	0	0	0	500000 euro	REFORMING	0,22
07/12/2012	Rostock-Peez, Mecklenburg-Vorpommern	Ammonia	RELEASE	STORAGE - PSVESSEL			0	0	0		STORAGE	0,3
14/01/2014		Syngas	RELEASE; FIRE	PROCESS - PIPEWORK	INSTRUMENT; MECHANICAL	CORRODE	0	0	0		REFORMING	Hydrogen: 0,7
09/05/2014	Enid, OK	Ammonia	RELEASE	PROCESS - PIPEWORK	PROCOND		0	0	0			0,045
14/07/2014	Garyville, LA	Hydrogen; Hydrogen sulfide	RELEASE	PROCESS - PIPEWORK			0	0	0			
02/09/2014	St.James, LA	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			
28/10/2014	Dodge City, KS	Ammonia	RELEASE	PROCESS - PIPEWORK			0	0	0			0,045
31/03/2015	Enid, OK	Ammonia	RELEASE	PROCESS - PIPEWORK	MECHANCAL		0	0	0			
05/11/2015	St.James, LA	Ammonia	RELEASE				0	0	0			
13/07/2016	Sulphur, LA	Hydrogen sulphide; Sulfur dioxide	RELEASE; FIRE	PROCESS - PIPEWORK	MECHANICAL		0	0	0		REFORMING	

Appendix C

The TensorFlow Code used to Run the Simulations

```
#!/usr/bin/env python

import tensorflow as tf
import tempfile
import pandas as pd

tf.logging.set_verbosity(tf.logging.INFO)

#input information
insert_label_type=input("\nInsert label type.\nInsert NPI for the injuries.\nInsert NPK for the fatalities.\n")
insert_model=input("Insert model type.\nValid model types: wide, deep, wide&deep.\n")
insert_steps=int(input("Insert number of steps for the training.\n"))
insert_model_dir=input("Insert model directory.\nFor example: /tmp/model\n")
insert_label=input("Insert label value ("+insert_label_type+").\nValid values: YES, NO, <10, 10-100, 100-1000, >1000.\n")

#import datasets
train_fileYN_inj = "/home/riccardo/Scrivania/my_codes/Datasets/eventdataYN_inj_full.csv"
train_fileYN_kill = "/home/riccardo/Scrivania/my_codes/Datasets/eventdataYN_full.csv"
test_ammYN_inj = "/home/riccardo/Scrivania/my_codes/Datasets/Ammdata_YN_inj.csv"
test_ammYN_kill = "/home/riccardo/Scrivania/my_codes/Datasets/Ammdata_YN_killed.csv"
train_fileCAT_inj = "/home/riccardo/Scrivania/my_codes/Datasets/eventdataCAT_inj_full.csv"
train_fileCAT_kill = "/home/riccardo/Scrivania/my_codes/Datasets/eventdataCAT_full.csv"
test_ammCAT_inj = "/home/riccardo/Scrivania/my_codes/Datasets/Ammdata_CAT_inj.csv"
test_ammCAT_kill = "/home/riccardo/Scrivania/my_codes/Datasets/Ammdata_CAT_killed.csv"

if insert_label_type == "NPI":
train_fileYN = train_fileYN_inj
```

```
train_fileCAT = train_fileCAT_inj
test_fileYN = test_ammYN_inj
test_fileCAT = test_ammCAT_inj
elif insert_label_type == "NPK":
train_fileYN = train_fileYN_kill
train_fileCAT = train_fileCAT_kill
test_fileYN = test_ammYN_kill
test_fileCAT = test_ammCAT_kill
else : print("\n———The label type inserted is not correct.———\n")

if insert_label == "YES" or insert_label == "NO": train_file = train_fileYN
test_file = test_fileYN
else :
train_file = train_fileCAT
test_file = test_fileCAT

#attributes
if insert_label_type == "NPK":
columns=["DA", "PD", "LO", "MN1", "MN2", "MN3", "MN4", "MN5", "MN6", "MN7", "MN8", "MN9",
"MH1", "MH2", "MH3", "MH4, "MH5", "MH6", "MH7", "MH8", "MH9", "MC1", "MC2", "MC3", "MC4",
"MC5", "MC6", "MC7", "MC8", "MC9", "IT1", "IT2", "IT", "GOG1", "SOG1", "GOG2", "SOG2", "NPK",
"NPE", "GC1", "GC2", "GC3", "SC1", "SC2", "SC3", "QY", "IS1", "IS2"]
else :
columns=["DA", "PD", "LO", "MN1", "MN2", "MN3", "MN4", "MN5", "MN6", "MN7", "MN8", "MN9",
"MH1", "MH2", "MH3", "MH4, "MH5", "MH6", "MH7", "MH8", "MH9", "MC1", "MC2", "MC3", "MC4",
"MC5", "MC6", "MC7", "MC8", "MC9", "IT1", "IT2", "IT", "GOG1", "SOG1", "GOG2", "SOG2", "NPI",
"NPE", "GC1", "GC2", "GC3", "SC1", "SC2", "SC3", "QY", "IS1", "IS2"]

df_train=pd.read_csv(train_file, names=columns, skipinitialspace=True)
df_test=pd.read_csv(test_file, names=columns, skipinitialspace=True)
#input function
if insert_label_type == "NPK": def get_input_fn(data_file, num_epochs, shuffle):
df_data=pd.read_csv(tf.gfile.Open(data_file),
names=columns,
skipinitialspace=True,
engine="python")
df_data=df_data.dropna(how="any", axis=0)
labels=df_data["NPK"].apply(lambda x: insert_label in x).astype(int)
return tf.estimator.inputs.pandas_input_fn(
x=df_data,
y=labels,
num_epochs=num_epochs,
shuffle=shuffle,
num_threads=1)
else :
```

```
def get_input_fn(data_file, num_epochs, shuffle):
df_data=pd.read_csv(tf.gfile.Open(data_file),
names=columns,
skipinitialspace=True,
engine="python")
df_data=df_data.dropna(how="any", axis=0)
labels=df_data["NPI"].apply(lambda x: insert_label in x).astype(int)
return tf.estimator.inputs.pandas_input_fn(
x=df_data,
y=labels,
num_epochs=num_epochs,
shuffle=shuffle,
num_threads=1)

#Define columns
DA= tf.feature_column.categorical_column_with_vocabulary_list(
’DA’, ["<1900s", "1900s", "1910s", "1920s", "1930s", "1940s", "1950s", "1960s", "1970s", "1980s",
"1990s"])
PD = tf.feature_column.categorical_column_with_vocabulary_list("PD", ["R", "U","Na"])
LO = tf.feature_column.categorical_column_with_hash_bucket(’LO’, hash_bucket_size=280)
MN1 = tf.feature_column.categorical_column_with_hash_bucket(’MN1’, hash_bucket_size=1450)
MN2 = tf.feature_column.categorical_column_with_hash_bucket(’MN2’, hash_bucket_size=375)
MN3 = tf.feature_column.categorical_column_with_hash_bucket(’MN3’, hash_bucket_size=170)
MN4 = tf.feature_column.categorical_column_with_hash_bucket(’MN4’, hash_bucket_size=90)
MN5 = tf.feature_column.categorical_column_with_hash_bucket(’MN5’, hash_bucket_size=30)
MN6 = tf.feature_column.categorical_column_with_hash_bucket(’MN6’, hash_bucket_size=15)
MN7 = tf.feature_column.categorical_column_with_vocabulary_list("MN7", ["AEROSOLS","FUEL
OIL", "Na", "NITRIC ACID", "OIL", "STYRENE"])
MN8 = tf.feature_column.categorical_column_with_vocabulary_list("MN8", ["HG SEED DRESS", "Na",
"NATURAL GAS", "TOLUENE DIISOCYANATE"])
MN9 = tf.feature_column.categorical_column_with_vocabulary_list("MN9", ["Na",
"ORG.PHOS.PESTIC"])
MH1 = tf.feature_column.categorical_column_with_vocabulary_list("MH1", ["AS", "CD", "CO", "EX",
"FI", "Na", "OX", "TO"])
MH2 = tf.feature_column.categorical_column_with_vocabulary_list("MH2", ["AS", "CD", "CO", "EX",
"FI", "Na", "OX", "TO"])
MH3 = tf.feature_column.categorical_column_with_vocabulary_list("MH3", ["AS", "CO", "EX", "FI",
"Na", "OX", "TO"])
MH4 = tf.feature_column.categorical_column_with_vocabulary_list("MH4", ["CO", "FI", "Na", "OX",
"TO"])
MH5 = tf.feature_column.categorical_column_with_vocabulary_list("MH5", ["CO", "FI", "Na", "TO"])
MH6 = tf.feature_column.categorical_column_with_vocabulary_list("MH6", ["CO", "FI", "Na", "TO"])
MH7 = tf.feature_column.categorical_column_with_vocabulary_list("MH7", ["CO", "FI", "Na"])
MH8 = tf.feature_column.categorical_column_with_vocabulary_list("MH8", ["FI", "Na", "TO"])
MH9 = tf.feature_column.categorical_column_with_vocabulary_list("MH9", ["Na", "TO"])
```

```
MC1 = tf.feature_column.categorical_column_with_hash_bucket(’MC1’, hash_bucket_size=350)
MC2 = tf.feature_column.categorical_column_with_hash_bucket(’MC2’, hash_bucket_size=180)
MC3 = tf.feature_column.categorical_column_with_hash_bucket(’MC3’, hash_bucket_size=95)
MC4 = tf.feature_column.categorical_column_with_hash_bucket(’MC4’, hash_bucket_size=55)
MC5 = tf.feature_column.categorical_column_with_hash_bucket(’MC5’, hash_bucket_size=23)
MC6 = tf.feature_column.categorical_column_with_hash_bucket(’MC6’, hash_bucket_size=15)
MC7 = tf.feature_column.categorical_column_with_vocabulary_list("MC7", ["Na", "1267", "2031", "1223", "1950", "2055"])
MC8 = tf.feature_column.categorical_column_with_vocabulary_list("MC8", ["Na", "1971", "2588", "2206"])
MC9 = tf.feature_column.categorical_column_with_vocabulary_list("MC9", ["Na", "2588"])
IT1 = tf.feature_column.categorical_column_with_hash_bucket(’IT1’, hash_bucket_size=15)
IT2 = tf.feature_column.categorical_column_with_hash_bucket(’IT2’, hash_bucket_size=15)
IT3 = tf.feature_column.categorical_column_with_hash_bucket(’IT3’, hash_bucket_size=15)
GOG1 = tf.feature_column.categorical_column_with_hash_bucket(’GOG1’, hash_bucket_size=15)
SOG1 = tf.feature_column.categorical_column_with_hash_bucket(’SOG1’, hash_bucket_size=15)
GOG2 = tf.feature_column.categorical_column_with_vocabulary_list("GOG2", ["PROCESS"])
SOG2 = tf.feature_column.categorical_column_with_vocabulary_list("SOG2", ["PIPEWORK", "HEATXCHANG"])
NPE = tf.feature_column.numeric_column(’NPE’)
GC1 = tf.feature_column.categorical_column_with_hash_bucket(’GC1’, hash_bucket_size=12)
GC2 = tf.feature_column.categorical_column_with_hash_bucket(’GC2’, hash_bucket_size=10)
GC3 = tf.feature_column.categorical_column_with_hash_bucket(’GC3’, hash_bucket_size=10)
SC1 = tf.feature_column.categorical_column_with_hash_bucket(’SC1’, hash_bucket_size=72)
SC2 = tf.feature_column.categorical_column_with_hash_bucket(’SC2’, hash_bucket_size=65)
SC3 = tf.feature_column.categorical_column_with_hash_bucket(’SC3’, hash_bucket_size=50)
QY = tf.feature_column.categorical_column_with_vocabulary_list("QY", ["Na", "1000-10000", "100-1000", "10-100", "from1to10", "<1", ">10000"])
IS1 = tf.feature_column.categorical_column_with_hash_bucket(’IS1’, hash_bucket_size=12)
IS2 = tf.feature_column.categorical_column_with_hash_bucket(’IS2’, hash_bucket_size=20)

#crossed columns crossed=[
tf.feature_column.crossed_column(["DA", "LO"], hash_bucket_size=int(1e5)),
tf.feature_column.crossed_column(["MN1", "MN2", "MN3", "MN4", "MN5", "MN6", "MN7", "MN8", "MN9"], hash_bucket_size=int(1e6)),
tf.feature_column.crossed_column(["MH1", "MH2", "MH3", "MH4", "MH5", "MH6", "MH7", "MH8", "MH9"], hash_bucket_size=int(1e6)),
tf.feature_column.crossed_column(["MC1", "MC2", "MC3", "MC4", "MC5", "MC6", "MC7", "MC8", "MC9"], hash_bucket_size=int(1e6)),
tf.feature_column.crossed_column(["IT1", "IT2", "IT3"], hash_bucket_size=int(1e4)),
tf.feature_column.crossed_column(["GOG1", "SOG1"], hash_bucket_size=int(1e4)),
tf.feature_column.crossed_column(["GOG2", "SOG2"], hash_bucket_size=int(1e4)),
tf.feature_column.crossed_column(["GC1", "GC2", "GC3"], hash_bucket_size=int(1e5)),
tf.feature_column.crossed_column(["SC1", "SC2", "SC3"], hash_bucket_size=int(1e5)),
tf.feature_column.crossed_column(["IS1", "IS2"], hash_bucket_size=int(1e4))
```

```
]

#WIDE MODEL input
#if insert_label_type == "NPI":
base_columns=[DA, PD, LO, MN1, MN2, MN3, MN4, MN5, MN6, MN7, MN8, MN9, MH1, MH2, MH3,
MH4, MH5, MH6, MH7, MH8, MH9, MC1, MC2, MC3, MC4, MC5, MC6, MC7, MC8, MC9, IT1, IT2, IT3,
GOG1, SOG1, GOG2, SOG2, NPE, GC1, GC2, GC3, SC1, SC2, SC3, QY, IS1, IS2]
crossed_columns=crossed

#DNN MODEL input
deep_columns = [
tf.feature_column.indicator_column(DA),
tf.feature_column.indicator_column(PD),
tf.feature_column.embedding_column(LO, dimension=8),
tf.feature_column.embedding_column(MN1, dimension=11),
tf.feature_column.embedding_column(MN2, dimension=9),
tf.feature_column.embedding_column(MN3, dimension=7),
tf.feature_column.embedding_column(MN4, dimension=6),
tf.feature_column.embedding_column(MN5, dimension=5),
tf.feature_column.embedding_column(MN6, dimension=4),
tf.feature_column.indicator_column(MN7),
tf.feature_column.indicator_column(MN8),
tf.feature_column.indicator_column(MN9),
tf.feature_column.indicator_column(MH1),
tf.feature_column.indicator_column(MH2),
tf.feature_column.indicator_column(MH3),
tf.feature_column.indicator_column(MH4),
tf.feature_column.indicator_column(MH5),
tf.feature_column.indicator_column(MH6),
tf.feature_column.indicator_column(MH7),
tf.feature_column.indicator_column(MH8),
tf.feature_column.indicator_column(MH9),
tf.feature_column.embedding_column(MC1, dimension=8),
tf.feature_column.embedding_column(MC2, dimension=7),
tf.feature_column.embedding_column(MC3, dimension=7),
tf.feature_column.embedding_column(MC4, dimension=6),
tf.feature_column.embedding_column(MC5, dimension=5),
tf.feature_column.embedding_column(MC6, dimension=4),
tf.feature_column.indicator_column(MC7),
tf.feature_column.indicator_column(MC8),
tf.feature_column.indicator_column(MC9),
tf.feature_column.embedding_column(IT1, dimension=5),
tf.feature_column.embedding_column(IT2, dimension=5),
tf.feature_column.embedding_column(IT3, dimension=5),
tf.feature_column.embedding_column(GOG1, dimension=5),
tf.feature_column.embedding_column(SOG1, dimension=5),
tf.feature_column.indicator_column(GOG2),
tf.feature_column.indicator_column(SOG2),
NPE,
tf.feature_column.embedding_column(GC1, dimension=4),
tf.feature_column.embedding_column(GC2, dimension=3),
tf.feature_column.embedding_column(GC3, dimension=3),
tf.feature_column.embedding_column(SC1, dimension=6),
tf.feature_column.embedding_column(SC2, dimension=6),
tf.feature_column.embedding_column(SC3, dimension=6),
tf.feature_column.indicator_column(QY),
tf.feature_column.embedding_column(IS1, dimension=4),
tf.feature_column.embedding_column(IS2, dimension=4),
]

#to make the model run on the cpu
run_config = tf.estimator.RunConfig().replace(
session_config=tf.ConfigProto(device_count=’GPU’: 0))

#CLASSIFIERS

model_wide= tf.contrib.learn.LinearClassifier(
feature_columns=base_columns+crossed_columns,
model_dir=insert_model_dir,
)

model_deep= tf.contrib.learn.DNNClassifier(
feature_columns= deep_columns,
hidden_units=[1024, 512, 256],
model_dir=insert_model_dir,
)

model_w_d= tf.contrib.learn.DNNLinearCombinedClassifier(
model_dir=insert_model_dir,
linear_feature_columns=crossed_columns,
dnn_feature_columns=deep_columns,
dnn_hidden_units=[1024,512,256],
)

if insert_model == "wide":
model=model_wide
elif insert_model == "deep":
model=model_deep
elif insert_model == "wide&deep":
model=model_w_d
```

```
else :
print("\n———The model type inserted is not correct.———\n")

#training and evaluation
model.fit(input_fn=get_input_fn(train_file, None, False) , steps=insert_steps)
results=model.evaluate(input_fn=get_input_fn(test_file, 1, False), steps=None)

#print the results
model_dir=insert_model_dir
print("\nmodel directory= %s" %model_dir)
print("\n" + insert_model + " model, " + insert_label_type + ": " + insert_label + "\n")
for key in sorted(results):
print("%s : %s" %(key, results[key]))

#print the probability values
predictions=list(model.predict_proba(input_fn=get_input_fn(test_file, 1, False)))

print("\n".join(map(str, predictions)))
```

References

2012/18/EU. DIRECTIVE 2012/18/EU of the EUROPEAN PARLIAMENT AND OF THE COUNCIL of 4 July 2012 on the control of major-accidents hazards involving dangerous substances, amending and subsequently repealing Council Directive 96/82/EC. Seveso III.

82/501/EEC. COUNCIL DIRECTIVE of 24 June 1982 on the major-accident hazards of certain industrial activities. Seveso I.

96/82/EC. COUNCIL DIRECTIVE 96/82/EC of 9 December 1996 on the control of major-accidents hazards involving dangerous substances. Seveso II.

AIChE (1994-2001). *Ammonia Plant Safety and Related Facilities*, volume 35-42.

ARIA (2018). ARIA - La référence du retour d'expérience sur accidents technologiques. https://www.aria.developpement-durable.gouv.fr. Access date: January 2018.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Cheng, H., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., and Shah, H. (2016). Wide & Deep Learning for Recommender Systems. *CoRR*, abs/1606.07792.

Dean, J. (2017). TensorFlow: Machine Learning for Everyone. https://www.youtube.com/watch?v=mWl45NkFBOc.

EPA (2017). National Response System. https://www.epa.gov/emergency-response/national-response-system. Access date: October 2017.

Falck, A., Flage, R., and Aven, T. (2015). Risk assessment of oil and gas facilities during operational phase. *Safety and Reliability of Complex Engineered Systems*.

Flach, P. and Kull, M. (2015). Precision-Recall-Gain Curves: PR Analysis Done Right. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems 28*, pages 838–846. Curran Associates, Inc.

GitHub (2018). GitHub - TensorFlow page. https://github.com/tensorflow. Access date: January 2018.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Greenberg, M., Haas, C., Cox, A., Lowrie, K., McComas, K., and North, W. (2012). Ten Most Important Accomplishments in Risk Analysis, 1980–2010. *Risk Analysis*, 32(5):771–781.

Hare, J. A., Johnson, M. P., and Fullam, B. (2009). Learning from Process Safety Incidents. *IChemE SYMPOSIUM SERIES*, (155):104–112.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2 edition.

He, Z. (2015). 1 - an overview of data mining. In He, Z., editor, *Data Mining for Bioinformatics Applications*, pages 1 – 10. Woodhead Publishing.

HSE (1999). *MHIDAS (Major Hazard Incident Data Service) database*. Harwell, UK: AEA Techology, Major Hazards Assessment Unit, Health and Safety Executive.

Ian H. Witten, Eibe Frank, M. A. (2011). *Data Mining. Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, 3rd edition.

Infosis (2017). Infosis - Informationssystem zum Stand der Sicherheitstechnik. Infosis/ZEMA. http://www.infosis.uba.de/index.php/de/zema/index.html. Access date: October 2017.

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2014). *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated.

JFS (2017). JFS (Japan For Sustainability) news. https://www.japanfs.org/en/news/archives/news_id026011.html. Access date: October 2017.

Jiawei Han, Michelin Kamber, J. P. (2012). *Data Mining. Concepts and Techinques*. Morgan Kaufamnn, 3rd edition.

JST (2017). Japanese Failure Knowledge Database. http://www.sozogaku.com/fkd/en/. Access date: October 2017.

Kletz, T. (2001). *Learning from Accidents*. Gulf Professional Publishing, Oxford, 3rd edition.

Le, Q. V. and Schuster, M. (2016). A Neural Network for Machine Translation, at Production Scale. https://research.googleblog.com/2016/09/a-neural-network-for-machine.html.

Lees, F. P. (2005). *Lees' Loss Prevention in the Process Industries*. Elsevier Butterworth-Heinemann, Oxford, 3rd edition. Editor: Sam Mannan.

MAHB (2017). eMARS (Major Accident Reporting System). https://minerva.jrc.ec.europa.eu/en/emars/content/. Access date: October 2017.

Murphy, K. P. (2012). *Machine Learning: a Probabilistic Perspective*. Cambridge, MA.

OSHA (2017). OSHA (Occupational Safety and Health Administration). Fatality and Catastrophe Investigation Summaries. https://www.osha.gov/. Access date: October 2017.

Paltrinieri, N. and Comfort, L. (2018). Learning about risk. Submitted to Safety Science.

Paltrinieri, N. and Khan, F. (2016). *Dynamic Risk Analysis in the Chemical and Petroleum Industry: Evolution and Interaction with Parallel Disciplines in the Perspective of Industrial Application*. Butterworth-Heinemann.

Paltrinieri, N., Landucci, G., and Rossi, P. S. (2017). Real-Time Data for Risk Assessment in the Offshore Oil&Gas Industry. Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE, 3B-2017.

Paltrinieri, N. and Reniers, G. (2017). Dynamic risk analysis for Seveso sites. *Journal of Loss Prevention in the Process Industries*, 49:111 – 119. The Seveso disaster and its 40-year legacy to process safety.

Sambeth, J. (1983a). The Seveso Accident. *Chemosphere*. Vol 12, No 4, pp. 681-686.

Sambeth, J. (1983b). What Really Happened at Seveso. *Chemical Engineering*. Vol 90.

Sandjideh, A. M. (2017). Announcing TensorFlow 1.0. https://developers.googleblog.com/2017/02/announcing-tensorflow-10.html.

Stack Overflow (2018). Newest 'tensorflow' Questions. https://stackoverflow.com/questions/tagged/tensorflow. Access date: January 2018.

TensorFlow (2018). TensorFlow - Official website. https://www.tensorflow.org/. Access date: January 2018.

UBA (2017). UBA (Umweltbundesamt) - Zentrale Melde-und Auswertestelle für Störfälle und Störungen. http://www.umweltbundesamt.de/themen/wirtschaft-konsum/anlagensicherheit/zentrale-melde-auswertestelle-fuer-stoerfaelle. Access date: October 2017.

Unruh, A. (2017). What is TensorFlow machine intelligence platform? - Learn about the Google-developed open source library for machine learning and deep neural network research. https://opensource.com/article/17/11/intro-tensorflow.

USCG (2017). National Response Center database. http://www.nrc.uscg.mil/. Access date: October 2017.

Villa, V., Paltrinieri, N., Khan, F., and Cozzani, V. (2016). Towards dynamic risk analysis: A review of the risk assessment approach and its limitations in the chemical process industry. *Safety Science*, 89:77 – 93.

Vincent, J. (2017). Google's latest platform play is artificial intelligence, and it's already winning. https://www.theverge.com/2017/5/18/15657256/google-ai-machine-learning-tensorflow-io-2017-platform-play.

Acknowledgment

I would first like to thank the Norges Teknisk-Naturvitenskapelige Universitet (NTNU) for hosting me when writing this Master Thesis. The friendly environment made me feel at home and the stimulating background pushed me to give my best.

I would like to acknowledge the University of Bologna for forming me these years both from the professional and human point of view.

I would like to thank my supervisor Nicola Paltrinieri. His knowledge on the thesis' subject, the time he spent for me and his friendly attitude have been of great help to me.

I would like to thank my co-supervisor Sarah Bonvicini for the help and the advises she gave me despite the distance.

I would like to acknowledge Kongsberg Digital for the support in developing the work.

Finally, I must express my very profound gratitude to my family, my girlfriend Katja and my friends. Without their support and their encouragement all of my accomplishments would have not been possible.