

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA  
MINISTERE DE HIGHER EDUCATION AND SCIENTIFIC RESEARCH  
UNIVERSITY MOHAMED BOUDIAF - M'SILA

FACULTY OF TECHNOLOGY  
DEPARTMENT ELECTRONICS

N° : .....



FACULTY : TECHNOLOGY  
OPTION : EMBEDDED SYSTEM

**Thesis presented for obtaining  
The degree of Academic Master  
by: Ahmed KEROUCHA  
Youcef BENHAMIDOUCHE**

**Entitled**

**PREDICTION OF CEMENT FINENESS USING  
MACHINE LEARNING APPROACHES  
(Application in Cement Grinding  
Ball mills- Lafarge Cement M'sila)**

**Defended before the jury composed of:**

.....	University: M'Sila .....	President
Abdelhafidh BENYOUNES	University: M'Sila .....	Advisor
Mounir BOURAS	University: M'Sila .....	Co-Advisor
.....	University: M'Sila.....	Examiner

**Academic year: 2020/2021**

# Acknowledgement

*Alhamdulillah, I praise and thank Allah for His greatness and for giving us the strength and courage to complete this thesis.*

*We would like to thank our coach Dr. Abdelhafid BENYOUNES who agreed to mentor us to make this thesis. We are grateful to him. Throughout this work, we have benefited from his great competence.*

*We would also like to thank all those who contributed directly or indirectly to the realization of this thesis*

# *Dedication*

*I dedicate this humble act to my dear mother for her  
sacrifices. In memory of my father  
To my wife, sons and daughters. To my brothers, to my  
sisters and all the family, to all my friends. To all the  
teachers in the Department of Electronics. And for all  
Muslims*

*Ahmed KEROUCHA*

# Dedication

*I dedicate this humble act to my dear mother for her  
sacrifices. to my dear father*

*To my wife, sons and daughters. To my brothers, to my  
sisters and all the family, to all my friends. To all the  
teachers in the Department of Electronics. And for all  
Muslims*

*Youcef BENHAMIDOUCHE*

# Table of Contents

Introduction.....	1
1 CHAPTER 1: MACHINE LEARNING INTRODUCTION .....	3
1.1 Introduction.....	3
1.2 Artificial Intelligence .....	3
1.1.1 A Little bit of history .....	4
1.1.2 What is machine learning?.....	6
1.1.3 Why Use Machine Learning? .....	7
1.1.4 Applications of Machine Learning.....	8
1.3 Types of Machine Learning .....	9
1.3.1 Supervised Learning: .....	9
1.3.2 Unsupervised Learning: .....	9
1.3.3 Semi-supervised Learning : .....	10
1.3.4 Reinforcement Learning: .....	11
1.4 Machine Learning Algorithms:.....	12
1.4.1 Linear regression.....	12
1.4.2 Support vector machine: .....	13
1.4.3 Decision Tree:.....	17
1.4.4 Random Forest: .....	18
1.4.5 Gradient boosting.....	19
1.4.6 XGBoost .....	20
1.4.7 Feature Importance: .....	20
1.5 Deep Learning.....	21
1.5.1 Deep Neural Networks:.....	22
1.5.2 Recurrent Neural Networks .....	26
1.5.3 Convolution Neural Networks: .....	27
1.6 Conclusion .....	28
2 Chapter 2. Description of Cement grinding Ball Mill system .....	29
2.1 Introduction.....	29
2.2 Main equipment in grinding area .....	30
2.2.1 Ball Mill .....	30
2.2.2 Feed system.....	31
2.2.3 Separator .....	32
2.2.4 Mill bag filter for cleaning of mill ventilation air .....	33

2.2.5	Separator bag filter.....	33
2.3	Process description.....	34
2.4	Control loops.....	35
2.4.1	Recipe of the feed to the mill:.....	35
2.4.2	Constant feed to the mill.....	36
2.4.3	Air flow through separator.....	36
2.4.4	Mill outlet temperature.....	37
2.4.5	Constant mill inlet under pressure.....	37
2.5	Cement Quality control.....	38
2.6	Cement Fineness.....	38
2.7	Case Study.....	39
2.7.1	System description.....	39
2.8	Conclusion.....	40
3	Chapter 3. Machine learning appRoAches for predictive models.....	41
3.1	Introduction.....	41
3.2	Proposed system:.....	41
3.3	Dataset.....	42
3.3.1	Dataset knowledge.....	42
3.3.2	Data export to CSV file:.....	44
3.3.3	Data Collected.....	45
3.3.4	Correlation matrix.....	45
3.3.5	Trend chart.....	46
3.4	Development environment.....	47
3.4.1	Google Colab.....	47
3.4.2	Jupyter Notebook.....	47
3.4.3	Python.....	48
3.5	Preprocessing data.....	48
3.5.1	Steps involved in data preprocessing.....	49
3.6	Model evaluation metrics.....	52
3.7	Simulation result.....	53
3.7.1	Linear regression (LR).....	53
3.7.2	Kernel Ridge Regression (KRR).....	55
3.7.3	Random Forest Regression (RFR).....	56
3.7.4	Neural Network model.....	57
3.7.5	XGBoost for Regression (XGB).....	59

3.8 Conclusion .....	61
Conclusion .....	62
Bibliography .....	<u>63</u>

## LISTE OF FIGURES

Figure 1.1 : Subdomains of AI.....	4
Figure 1.2 : AI Applications .....	4
Figure 1.3 : The traditional approach.....	7
Figure 1.4 : Machine Learning approach .....	7
Figure 1.5: Machine Learning can help humans learn.....	8
Figure 1.6: Reinforcement Learning Components (Fumo 2017, p.4).....	11
Figure 1.7 : Linear relationship.....	13
Figure 1.8: First classification scenario with SVM (Ray 2017, p.4).....	14
Figure 1.9: Second classification scenario with SVM (Ray 2017, p.4).....	14
Figure 1.10: Third classification scenario with SVM (Ray 2017, p.5).....	15
Figure 1.11: Fourth classification scenario with SVM (Ray 2017, p.5).....	15
Figure 1.12: Fifth classification scenario with SVM (Ray 2017, p.6-7).....	16
Figure 1.13: Titanic decision tree (cited in Gupta 2017a, p.2).....	17
Figure 1.14: Relationship between AI, ML and DL (Le 2017, p.10).....	21
Figure 1.15: Illustration of a deep learning model (Goodfellow et al. 2016, p.6).....	23
Figure 1.16: NN types chart (Van Veen 2016, p.1).....	25
Figure 1.17: unrolling the network through time.....	27
Figure 1.18: A recurrent neuron (left) unrolled through time (right).....	27
Figure 2.1: Cement Ball Mill, Lafarge M'sila .....	30
Figure 2.2: Design of mill internals FICEM_Christian Pfeiffer_2017 .....	31
Figure 2.2.3 Cement Feeding system.....	31
Figure 2.4: Limestone and gypsum feeders in Lafarge M'Sila.....	32
Figure 2.5 : Example of Cement air separator .....	32
Figure 2.6 FLSmith Sepax air separator .....	33
Figure 2.7 : Cement grinding with ball mill systems. Infinity for cement equipment .....	35
Figure 2.8 : Recipe of the feed to the mill .....	36
Figure 2.9: Mill feed is supervised by the power of elevator.....	36
Figure 2.10: Air flow control through separator fan power .....	37
Figure 2.11: Mill outlet temperature control.....	37
Figure 2.12 : Mill inlet under pressure control .....	37
Figure 3.1: Architecture of prediction steps.....	41
Figure 3.2 : Third level architecture.....	44
Figure 3.3: Correlation Matrix .....	45

Figure 3.4: Trend chart of correlation of fineness.....	46
Figure 3.5: How to run Colaboratory.....	47
Figure 3.6: Google Colab editor .....	47
Figure 3.7: Jupyter Notebook Logo.....	48
Figure 3.8 Python logo.....	48
Figure 3.9: Before handling outlier     Figure 3.10: After handling outlier.....	51
Figure 3.11: Actual and predicted output for linear regression of test values .....	54
Figure 3.12: Actual and predicted output for linear regression of train values.....	54
Figure 3.13: Histograms of actual and predicted output for Linear regression of test values.....	54
Figure 3.14: Actual and predicted output for kernel ridge regression of test values .....	55
Figure 3.15: Actual and predicted output for kernel ridge regression of train values.....	55
Figure 3.16: Histogram of actual and predicted output for kernel ridge regression of test values .....	56
Figure 3.17: Actual and predicted output for random forest regression of test values .....	56
Figure 3.18: Actual and predicted output for random forest regression of train values.....	57
Figure 3.19: Actual and predicted output for random forest regression of test values .....	57
Figure 3.20: Actual and predicted output for neural network model regression of test values.....	58
Figure 3.21: Actual and predicted output for neural network regression of train values.....	58
Figure 3.22: Histogram of actual and predicted output for Multi Layer Neuron Network test .....	59
Figure 3.23: Actual and predicted output for XGBoost for Regression of test values .....	59
Figure 3.24: Actual and predicted output for XGBoost for Regression of train values.....	60
Figure 3.25: Histogram of actual and predicted output for XGBoost for Regression of test values .....	60

## **LISTE OF TABLES**

Table 3.1 : Variables data .....	44
Table 3.2: Part of data collected.....	45
Table 3.3: comparison Performance evaluation for different techniques. ....	60

## **NOTATION AND ABBREVIATED TERMS**

**ML** : Machine Learning

**DL** : Deep learning

**AI** : artificial intelligence

**CART** : Classification And Regression Trees

**RF** : Random forest

**GBM** : Gradient Boosting Machine

**ANNs** : Artificial Neural Networks

**NNs** : Neural Networks

**MLP** : Multilayer Perceptron

**RNNs** : Recurrent neural networks

**LSTM** : Long Short-Term Memory

**BILSTM** : Bidirectional Long Short-Term Memory

**MAE** : Mean Absolute Error

**MSE** : Mean Square Error

**RMSE** : Root Mean Square Error

**RRMSE** : Relative Root Mean Square Error

# INTRODUCTION

## INTRODUCTION

Cement manufacturing is a complex process that begins with mining and extract raw material that include limestone and clay, then grinding this material to a fine powder, called raw meal, after is heated in a cement rotary kiln at temperature higher than 1450 °C. In this step of process a several chemical change of material was done witch result a new product called clinker. The clinker is grinded in the cement mill with gypsum and limestone to create cement, the end user mixed the powdered cement with water and aggregates to form concrete that is used in construction [1].

The model predictive used in the high-level control of different processes, to optimize the control sequence and control loops of the plant, has become a standard technology. The predictive controller use the soft output to have robustness against plant model. It is one of the best examples of highly non-linear system to be controlled by a linear model [2].

The cement fineness is an important factor in the quality of cement, after collecting cement samples in the laboratory to analyze this collected material by a specified equipment to obtain the fineness or sieve of the produced cement, this procedure was done every one or two hours or more than this duration in the plant with many mills [3].

Machine learning (ML) has the potential to become a general-purpose technology just as computing became general-purpose 70 years ago. Over the past decade, significant progress has been made in the ability to build prediction models, which are the main building blocks in machine learning that turn the data we have into the information we need. This progress can be attributed to three factors: (1) the development of better and more user-friendly machine learning systems, (2) the invention of more performant hardware, and (3) the availability of ever larger amounts of data [4].

Currently ML enjoyed a series of high-profile achievements in, e.g., image classification, speech recognition, automatic machine translation, speech synthesis, and games playing. These successes are enabled partly by software packages that automate learning; partly by the use of

expressive black-box models and predictive accuracy as the metric for validating them; and partly by the unreasonable effectiveness of computational methods, like stochastic gradient descent

The purpose of this thesis is the development of a soft sensor to predict the cement fineness. The data was obtained from the measurable variables of the cement grinding circuit during the regular and real production of cement ball mill process.

The first and essential step is the extraction of raw data from the plant and its preprocessing, analyzing, treating, cleaning raw data and isolation of anomalies for outlier analysis, which allows developing high-quality data and consequently, it improves the performance of the model.

The selection of the model inputs variables is another essential step. Thus, the variables were selected applying correlation algorithm.

In chapter 1 we introduce a few definition of the different techniques of machine learning and the prediction algorithms used in artificial intelligent and see the criteria value to use this technology.

In chapter 2 provides a simplified description of the cement grinding circuit with the main equipment erected the grinding cement area, also the operation process with existing control loops.

In chapter 3 focus at the data collection and the preprocessing of this data, also the result of using the different algorithms, linear regression, kernel ridge regression, random forest regression and neural network model, and discuss the result of each model evaluation metrics.

Finally, the conclusion of implementation of model based on machine learning algorithms.



# CHAPTER 1

## MACHINE LEARNING INTRODUCTION

# **1 CHAPTER 1: MACHINE LEARNING INTRODUCTION**

## **1.1 Introduction**

Machine learning (ML) has the potential to become a general-purpose technology just as computing became general-purpose 70 years ago. Over the past decade, significant progress has been made in the ability to build prediction models, which are the main building blocks in machine learning that turn the data we have into the information we need. This progress can be attributed to three factors: (1) the development of better and more user-friendly machine learning systems, (2) the invention of more performant hardware, and (3) the availability of ever larger amounts of data [4].

In this chapter, we will study several types of machines Learning Algorithms, we will explain the different generalities common to all. Then the general operation of the machine learning algorithm. Then we study the types of algorithms for regression that we will apply in this work. Finally, we will list the main advantages and disadvantages.

## **1.2 Artificial Intelligence**

Artificial intelligence (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans.

- AI is an interdisciplinary science with multiple approaches.
- AI has become an essential part of the technology industry.

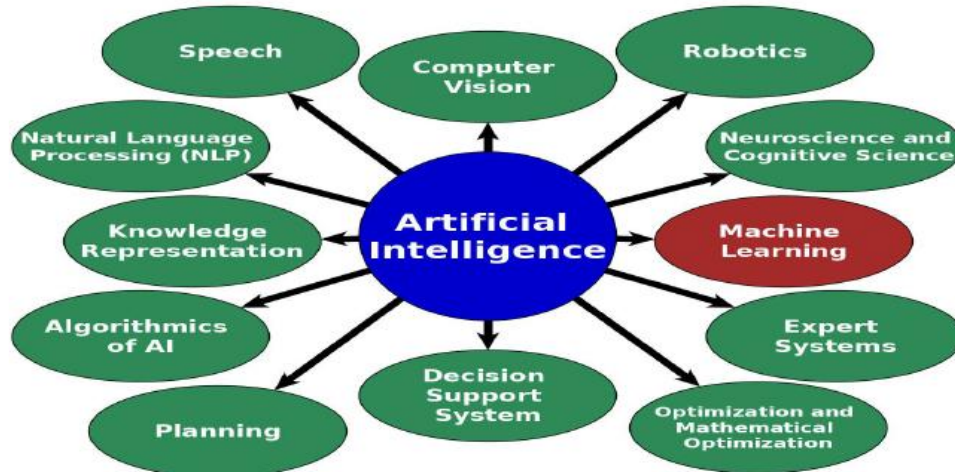


Figure 1.1 : Subdomains of AI

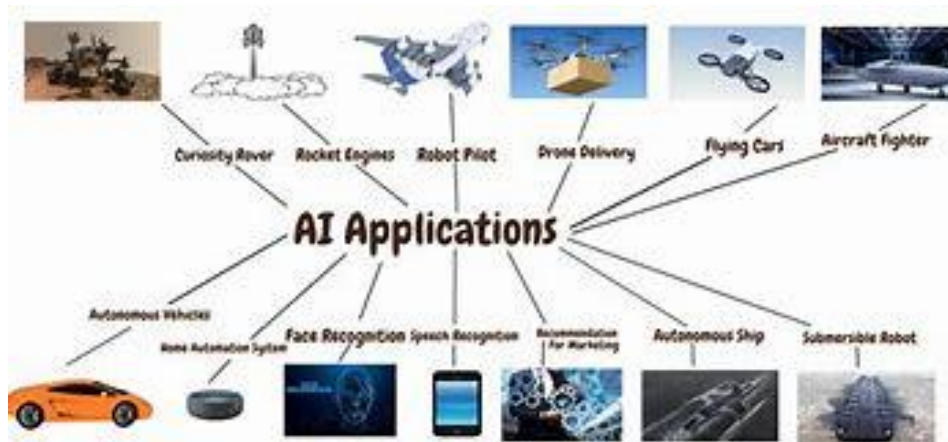


Figure 1.2 : AI Applications

### 1.1.1 A Little bit of history

It might seem that this is a pretty new technology, but, in fact, it isn't. The first ML-related work dates from 66 years ago, in 1950.

In the early days of AI, ML research used mainly symbolic data, and algorithm design was based on logic [5] [6]. At about the same time, Frank Rosenblatt proposed the Perceptron, a statistical approach based on empirical risk minimization [7]. However, this approach remained unrecognized and undeveloped in the following decades.

The real development of statistical learning came after 1986, when David Rumelhart and James McClelland proposed the nonlinear backpropagation algorithm [8]. AI, pattern

recognition, and statistics researchers became interested in this approach and nowadays is highly used in deep learning Neural Networks.

For the sake of curiosity, below there is a chronological list of the most relevant events in this field since the “starting point” in 1950 [¡Error! No se encuentra el origen de la referencia.](#)

1950 – Alan Turing creates the “Turing Test”. This test determined whether a computer had real intelligence or not.

1952 – Arthur Samuel writes the first computer learning program. It played checkers, and the computer was able to improve at the game the more it played, studying which moves made up to winning games.

1957 – Frank Rosenblatt designed the first neural network for computers (the perceptron), which simulates the thought processes of the human brain.

1967 – The “Nearest Neighbours” algorithm was written, allowing computers to begin using very basic pattern recognition.

1979 – Students in Stanford University invent the “Stanford Cart”, which can navigate obstacles in a room on its own.

1981 – Gerald Dejong introduces the concept of Explanation Based Learning (EBL), in which a computer analyses training data and creates a general rule it can follow by discarding unimportant data.

1985 – Terry Sejnowski invents NetTalk, which learns to pronounce words the same way a baby does.

1986 – David Rumelhart and James McClelland propose the nonlinear back propagation algorithm.

1990s – Work on Machine Learning shifts from a knowledge-driven approach to a data-driven approach. Scientists begin creating programs for computers to analyse large amounts of data (Big Data) and draw conclusions (or “learn” from the results).

1997 – IBM’s Deep Blue beats the world champion at chess. 2006 – Geoffrey Hinton coins the term “deep learning” to explain new algorithms that let computers “see” and distinguish objects and text in images and video.

2011 – Google Brain is developed and its deep neural network can learn to discover and categorize objects much the way a cat does.

2012 – Google’s X Lab develops a machine learning algorithm that is able to autonomously browse YouTube videos to identify the videos that contain cats.

2014 – Facebook develops DeepFace, a software algorithm that is able to recognize or verify individuals on photos to the same level as humans can.

2015 – Amazon launches its own machine learning platform.

2015 – Microsoft creates the Distributed Machine Learning Toolkit, which enables the efficient distribution of machine learning problems across multiple computers.

2016 – Google’s AlphaGo, an artificial intelligence algorithm, beats a professional player at the Chinese board game Go, which is considered the world’s most complex board game and is many times harder than chess. It managed to win five games out of five.

### **1.1.2 What is machine learning?**

Machine learning has become one of the mainstays of the information technology in the past two decades and thus, an important, but hidden, part of our life. The increasing amount of data that is being generated (and stored) daily by individuals and corporations, demands a smart analysis. It is here where machine learning comes to stage as a necessary ingredient for technological progress. As the word stands for, machine learning is the study of computer algorithms capable of learning to improve their performance of a task on the basis of their own previous experience. It focuses in achieving that programmable devices and “machines” learn automatically, by themselves. Basically, it is all about systems learning from data [4].

The field is closely related to pattern recognition and statistical inference. It works with data and processes it to discover patterns that can be later used to analyse new data. It usually relies on specific representation of data, a set of “features” that are understandable for a computer. For example, if text had to be represented, it should be through the words it contains or some other

characteristics such as length of the text, number of emotional words, etc. This representation depends on the task than one is dealing with and is typically referred to as “feature extraction” [4].

### 1.1.3 Why Use Machine Learning?

The traditional approach. If the problem is not trivial, your program will likely become a long list of complex rules pretty hard to maintain. But, Machine Learning approach. The program is much shorter, easier to maintain, and most likely more accurate.

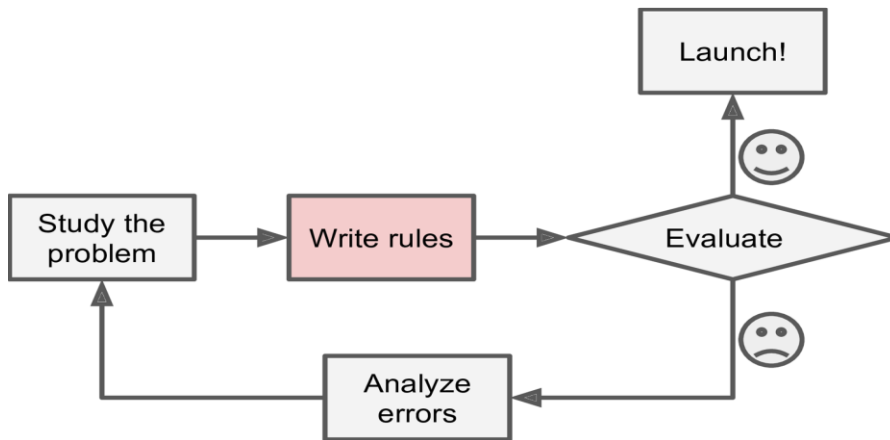


Figure 1.3 : The traditional approach

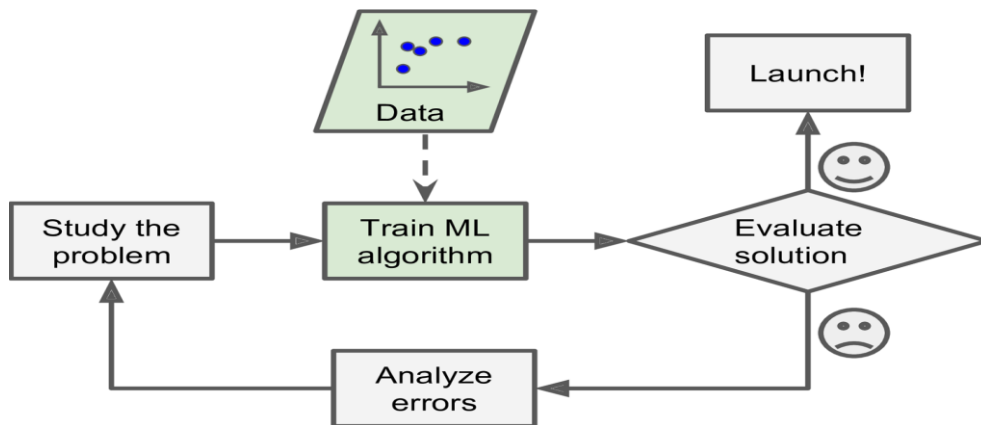


Figure 1.4 : Machine Learning approach

Machine Learning can help humans learn [4] (Figure 1.4): ML algorithms can be inspected to see what they have learned (although for some algorithms this can be tricky). For instance, once the spam filter has been trained on enough spam, it can easily be inspected to reveal the list of

words and combinations of words that it believes are the best predictors of spam. Sometimes this will reveal unsuspected correlations or new trends, and thereby lead to a better understanding of the problem.

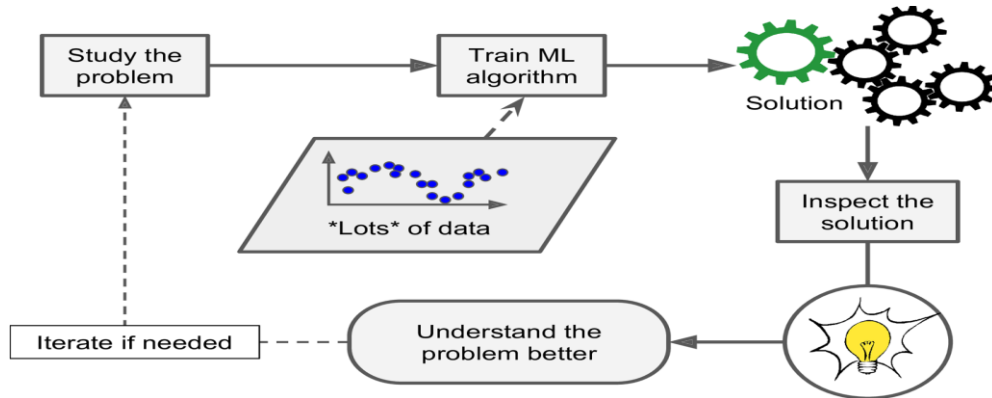


Figure 1.5: Machine Learning can help humans learn

#### 1.1.4 Applications of Machine Learning

Machine learning is currently the preferred approach in the following domains:

- 1) Speech analysis: e.g., speech recognition, synthesis.
- 2) Computer vision: e.g., object recognition/detection.
- 3) Robotics: e.g., position/map estimation.
- 4) Bio-informatics: e.g., sequence alignment, genetic analysis.
- 5) E-commerce: e.g., automatic trading, fraud detection.
- 6) Financial analysis: e.g., portfolio allocation, credits.
- 7) Medicine: e.g., diagnosis, therapy conception.
- 8) Web: e.g., Content management, social networks, etc.

To summarize, Machine Learning is great for:

- Problems for which existing solutions require a lot of hand-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better.
- Complex problems for which there is no good solution at all using a traditional approach: the best Machine Learning techniques can find a solution.
- Fluctuating environments: a Machine Learning system can adapt to new data.
- Getting insights about complex problems and large amounts of data.

## 1.3 Types of Machine Learning

According to what goal to be achieved by using ML, it can be classified into four major types as follows [4]:

### 1.3.1 Supervised Learning:

Supervised learning earned its name because data scientists acts as a guide to teach the algorithm what conclusions it should come up with. It is similar to the way a student learns basic arithmetic from a teacher. This type of learning requires labeled data with the correct answers to be expected from the algorithm's output. For classification and regression problems, supervised learning proved itself to be accurate and fast [9].

- **Classification:** consists of predicting the categorical output value where the data can be separated into specific —classes. Classification 9 has different use cases, such as: determining the weather, if an email is a spam or not or types of animals after being trained on a properly labeled dataset of images with the species and some identifying characteristics [9].
- **Regression:** it's a type of problem where the prediction of a continuous-response value such as stock and housing prices is needed [9].

So, the way it works is modeling relationships and dependencies between the target prediction output and the input features such that it is possible to predict the output values for new data based on those relationships which it learned from the previous datasets [9].

### 1.3.2 Unsupervised Learning:

Conversely, unsupervised learning is more closely aligned with what it is called true artificial intelligence by some experts – the concept that a machine can learn to identify complex processes and patterns without supervision from humans. This approach is particularly useful in cases where the experts doesn't know what to look for in the data and the data itself does not include Targets. Under the many use cases of unsupervised machine learning it's worth mentioning k-means clustering, principal and independent component analysis, and association rules [4].

- **Clustering:** it's a type of a problem where similar things are grouped together. It shares the same concept with classification but in this case, there are no labels provided and the system will understand from the data itself and cluster it. A use case for this would be clustering news, articles depending on their genre, content. (cited in Trevino 2016)

Despite This type of machine learning opens the doors to solving problems that human normally would not tackle, it's not used as widely as the supervised learning due to its complexity and difficulty to implement [9].

### 1.3.3 Semi-supervised Learning :

Until now, the data provided is all labeled with the desired output or not labeled at all. Semi-supervised machine learning is a combination of the two. In many practical situations, the cost to label is quite high and in case of large datasets the task become tedious and very much time consuming. In addition, providing too much labeled data, can force human biases on the model. Even though the unlabeled data is unknown for the network, this data brings useful information about the target group parameters. Which leads to the conclusion, that by including unlabeled data the accuracy of the model can be improved while also saving time and money building it. For example, semi-supervised machine learning could be used in webpage classification, voice recognition or genetic sequencing. In those cases, data scientists can access large volumes of unlabeled data, and the task of labeling all of it would take an overwhelming time [10].

Using the information acquired until now a comparison between these three types of machine learning can be set for the same use case, for example classification:

- Supervised classification: The algorithm will classify the types of the webpages according to the labels provided from the beginning [10].
- Unsupervised clustering: The algorithm will look for patterns and characteristics that help placing webpages into groups.
- Semi unsupervised classification: The algorithm will identify the different groups of webpages based on the labelled data and will use the unlabelled data to define the boundaries of those webpage types and to look for other types that might not be listed in the labelled data.

### 1.3.4 Reinforcement Learning:

Reinforcement Learning is the third main Machine Learning type along with Supervised and Unsupervised Learning. It consists of five important components which are: the agent, environment, state, action and reward. The goal of RL is to maximize the reward and minimize the risk by exploiting its interaction with the environment. The RL algorithm (called the agent) will periodically improve by exploring the environment going through the different possible states. To maximize the performance, the ideal behaviour will be automatically determined by the agents. A feedback (the reward) is what allows the agent to improve its behaviour.

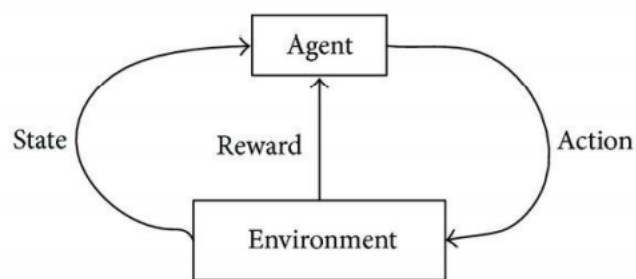


Figure 1.6: Reinforcement Learning Components (Fumo 2017, p.4)

To obtain agents with good results, reinforcement machine learning goes through five main steps. Fumo) describe them in his article as follows:

- The agent examines constantly the input state.
- The agent performs an action according to the function responsible for decision-making.
- The agent will receive reinforcement (reward) after performing its action.
- Information about the reward state will be stored.

In Reinforcement Learning, there are two types of tasks: episodic and continuous:

- **Episodic task:** The task in this case is defined by a starting and an ending point or also called a terminal state. This creates an episode: a list of states, actions, rewards, and new states. Video games are a typical example of this type of tasks.
- **Continuous task:** Opposite to the first type, this one has no terminal state and as its name indicates, continues forever. In this case the agent has to learn how to choose the best

actions and simultaneously interacts with the environment. Automated stock trading is a typical use case of this type of tasks. The agent keeps doing actions and receiving feedback until it's decided to be stopped, since there is no starting point and terminal state.

One of the most used algorithms for Reinforcement Learning is Monte Carlo which is based on collecting the rewards at the end of the episode and then calculating the maximum expected future reward. A second popular algorithm is Temporal Difference Learning that uses a different approach from the first one which is estimating the rewards at each step.

## 1.4 Machine Learning Algorithms:

Since the beginning of the AI implementation, many techniques were used, and many others are emerging until this day. In this subchapter, three different techniques will be discussed and ordered by their introducing date to the public.

### 1.4.1 Linear regression

Linear regression is a technique used for the modeling and analysis of numerical data. It exploits the relationship between two or more variables so that we can gain information about one of them through knowing values of the other. Regression can be used for prediction, estimation, hypothesis testing, and modeling causal relationships [9].

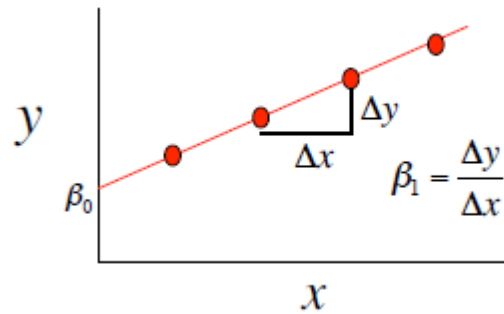
$$\mathbf{Y} = \mathbf{X1} + \mathbf{X2} + \mathbf{X3} \tag{1-1}$$

Dependent Variable	Independent Variable
Outcome Variable	Predictor Variable
Response Variable	Explanatory Variable

When we are examining the relationship between a quantitative outcome and a single quantitative explanatory variable, simple linear regression is the most commonly considered analysis method. (The “simple” part tells us we are only considering a single explanatory variable.) In linear regression, we usually have many different values of the explanatory variable, and we usually assume that values between the observed values of the explanatory variables are also possible values of the explanatory variables. We postulate a linear relationship between the population mean of the outcome and the value of the explanatory variable. If we let Y be some

outcome, and  $x$  be some explanatory variable, then we can express the structural model using the equation:

$$y = \beta_0 + \beta_1 x \quad (1-2)$$



**Figure 1.7 : Liner relationship**

Much of mathematics is devoted to studying variables that are deterministically related to one another, but we're interested in understanding the relationship between variables related in a nondeterministic fashion.

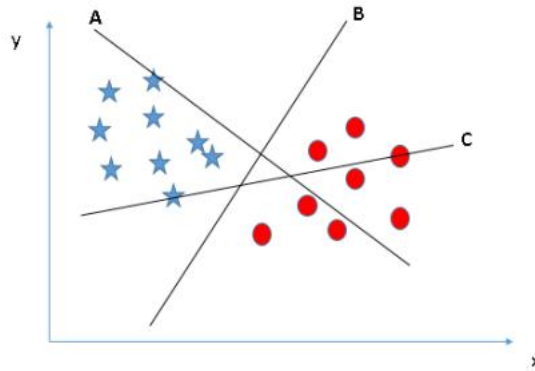
### **1.4.2 Support vector machine:**

Support Vector Machine (SVM) is a supervised machine learning technique which tackles mainly regression and classification challenges. In case of classification, each data item is plotted as points in  $n$ -dimensional space (where  $n$  represents the number of available features) with the value of each feature being the value of a particular coordinate. Afterwards, by classifying the different classes, a hyper-plane will be plotted to separate them clearly.

Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

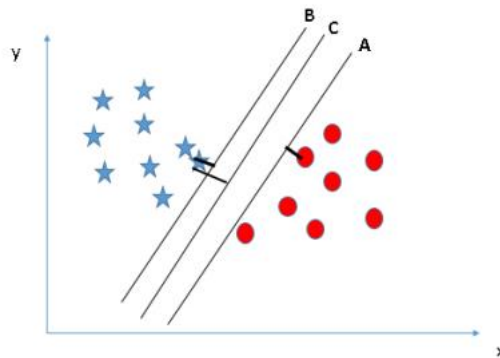
There are several scenarios that can be stumbled on while trying to apply SVM and it can deal with them perfectly to identify the right hyper-plane.

- A, B and C represents three hyper-planes. The one which segregates the two classes better will be selected. As the figure shows, B is the appropriate choice.



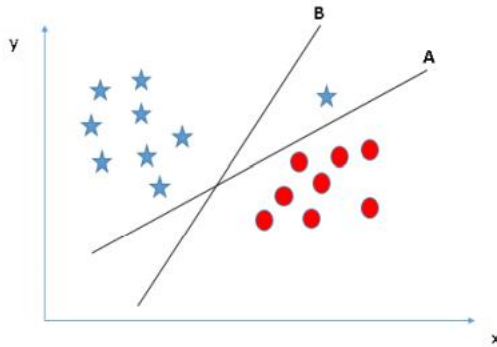
**Figure 1.8: First classification scenario with SVM (Ray 2017, p.4)**

- In this case all three hyper-planes are segregating the classes well. To decide which one from the three is the right one, the distances between the nearest data point and the hyper-plane should be maximized. This distance is called Margin. Another reason for choosing the hyper-plane with the higher margin is robustness, otherwise a misclassification has a high chance to occur when choosing a hyper-plane with a low margin.



**Figure 1.9: Second classification scenario with SVM (Ray 2017, p.4)**

- In this scenario, applying the same logic as the previous scenario won't give a correct classification since B has the higher margin and as the below figure demonstrate, A should be the right choice. Here, the SVM technique will be aware of the situation and won't prioritize the margin maximization over classifying correctly the two classes.



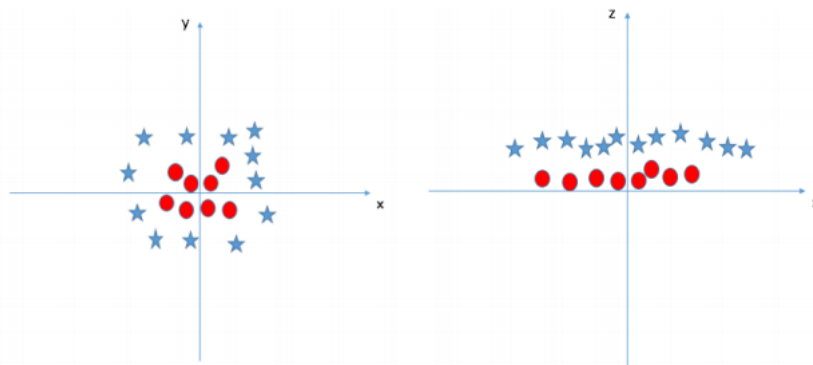
**Figure 1.10: Third classification scenario with SVM (Ray 2017, p.5)**

- In this case segregating the two classes is not possible since one of star class lies in the territory of the other class as an outlier. Luckily SVM robustness will prevent choosing the wrong hyper-plane by ignoring any possible outliers.



**Figure 1.11: Fourth classification scenario with SVM (Ray 2017, p.5)**

- Here the two classes can't be directly separated with a linear hyper-plane, that's why SVM introduces a new additional feature which is:  $z = x^2 + y^2$  to properly separate the two classes. (cited in Ray 2017, p.6)



**Figure 1.12: Fifth classification scenario with SVM (Ray 2017, p.6-7)**

- Having a linear hyper- plane between these two classes is an easy task for SVM. But should the additional feature be added manually as done in the last scenario to have a hyper-plane? It is done automatically by an SVM technique called kernel trick. The kernels are functions that take data which is not linearly separable in a low dimensional space and transform it in a higher dimensional space where it can be linearly separable. It is mostly useful in non-linear separation problem. Said otherwise, based on the labels or defined outputs, it will do some extremely complex data transformations to figure out the process to separate the data.
- The Scikit-learn developers lists several advantages and disadvantages of SVM. They are as follows:
  - The advantages of support vector machines are:
    - Effective in high dimensional spaces.
    - Still effective in cases where number of dimensions is greater than the number of samples.
    - Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
    - Versatile: different kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.
  - The disadvantages of support vector machines include:
    - If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
    - SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

### 1.4.3 Decision Tree:

In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of 16 decisions. (Gupta 2017a, p.1) Random forest is widely used to solve classification and regression problems in machine learning and it's also a commonly used tool in data mining to achieve a particular goal by deriving strategies.

The algorithm is represented as an upside-down drawn tree. In the figure below, the tree splits every time there is a condition (internal node) which are represented with the bold text in black. The outputted decision is called a branch (edge). In case a branch reached its limit and can't be divided anymore, it is identified as a decision (leaf). As shown in the next figure, the leaves are in red and green and represent whether a passenger from the titanic died or survived.

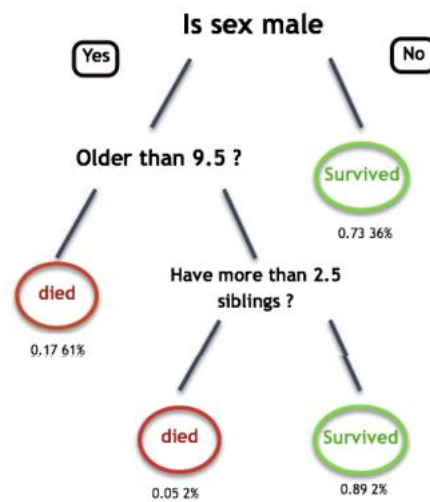


Figure 1.13: Titanic decision tree (cited in Gupta 2017a, p.2)

The main reason why this algorithm is widely used is its simplicity and how the feature importance and the relations in the tree can be easily represented. The above figure represents an example of a classification tree since its objective is predicting and classifying data of the titanic passengers into different classes (died or survived). When in the other hand, a regression tree predicts an output depending on a continuous progressing data. Decision Tree is mostly referred to as CART (Classification and Regression Trees). Finally, Gupta lists in his post (2017a, p.5) the different advantages and disadvantages of CART:

- **Advantages of CART**

- Simple to understand, interpret, visualize.
- Decision Trees implicitly perform variable screening or feature selection
- Can handle both numerical and categorical data. Can also handle multioutput problems.
- It requires relatively little effort from users for data preparation.
- Nonlinear relationships between parameters do not affect tree performance.

- **Disadvantages of CART**

- Decision-tree learners can create over-complex trees that do not generalize the data well, which will result in overfitting.
- Decision trees can be unstable because small variations in the data might result in completely different tree being generated. This is called variance, which needs to be lowered by methods like bagging and boosting.
- Greedy algorithms cannot guarantee to return the globally optimal decision tree. This can be mitigated by training multiple trees, where the features and samples are randomly sampled with replacement.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the data set prior to fitting with the decision tree.

#### **1.4.4 Random Forest:**

This technique falls under the supervised machine learning category. As the name indicates, a forest will be created from a group of decision trees and then will be randomized. Different methods can be used to train the random forest, and the mostly used one is Bootstrap Aggregation (Bagging) method.

As Brownlee describes it in his article (2016a, p.3) the Bagging method is a —very powerful ensemble method and goes further with his explanation —an ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model. As the Random Forest consists of multiple Decision Trees, it's used also for the same purposes: Regression and Classification. Not only that, it shares also almost the same hyperparameters as a decision tree. Fortunately, a decision

tree doesn't have to be combined with a bagging classifier and the —classifier-class of Random Forest can be used here. Furthermore, Regression problems can be solved with using Random Forest regression.

During the process of training, the Forest randomize the model. It also seeks the best feature among a random subset of features while creating the nodes of the tree instead of looking for the most important feature. This behavior will optimize the model since the randomness will result in a wide diversity. This randomness can be also added to other aspects of the training, such as randomizing the thresholds used for each feature. Which oppose to the traditional decision tree method, where the model will look for the most fitting threshold.

#### 1.4.5 Gradient boosting

Gradient boosting machines are a family of powerful machine-learning techniques that have shown considerable success in a wide range of practical applications. They are highly customizable to the particular needs of the application, like being learned with respect to different loss functions.

The common ensemble techniques like random forests rely on simple averaging of models in the ensemble. The family of boosting methods is based on a different, constructive strategy of ensemble formation. The main idea of boosting is to add new models to the ensemble sequentially. At each particular iteration, a new weak, base-learner model is trained with respect to the error of the whole ensemble learnt so far. The first prominent boosting techniques were purely algorithm-driven, which made the detailed analysis of their properties and performance rather difficult. This led to a number of speculations as to why these algorithms either outperformed every other method, or on the contrary, were inapplicable due to severe overfitting. To establish a connection with the statistical framework, a gradient-descent based formulation of boosting methods was derived [8]. This formulation of boosting methods and the corresponding models were called the gradient boosting machines. This framework also provided the essential justifications of the model hyperparameters and established the methodological base for further gradient boosting model development. In gradient boosting machines, or simply, GBMs, the learning procedure consecutively fits new models to provide a more accurate estimate of the response variable. The principle idea behind this algorithm is to construct the new base-learners to be maximally correlated with the negative gradient of the loss function associated with the whole ensemble. The

loss functions applied can be arbitrary, but to give a better intuition, if the error function is the classic squared-error loss, the learning procedure would result in consecutive error-fitting. In general, the choice of the loss function is up to the researcher, with both a rich variety of loss functions derived so far and with the possibility of implementing one's own task-specific loss.

#### **1.4.6 XGBoost**

Extreme Gradient Boosting (XGBoost) is an open-source library that provides an efficient and effective implementation of the gradient boosting algorithm. Shortly after its development and initial release, XGBoost became the go-to method and often the key component in winning solutions for a range of problems in machine learning competitions.

Regression predictive modeling problems involve predicting a numerical value such as a dollar amount or a height. XGBoost can be used directly for regression predictive modeling.

#### **1.4.7 Feature Importance:**

Random Forests makes the measurement of the relative importance of each feature on the prediction very easy. By using the methods provided by the class Sklearn from the machine learning tool scikit-learn, the features importance can be evaluated and measured to discover any impurity that may exist in the forest. The output of each feature will be scaled in a way that the sum of all importance is equal to 1. Since some features don't play a role in the prediction process, they can be dropped and determined by looking through at the feature importance. This step is almost necessary, since the more features taken under consideration in machine learning the more likely the model will be overfitted. Despite that Random Forests are based on Decision Trees, there are some differences between them:

If a decision tree is provided by a training dataset with features and labels, it will formulate some set of rules, which will be used to make the predictions.

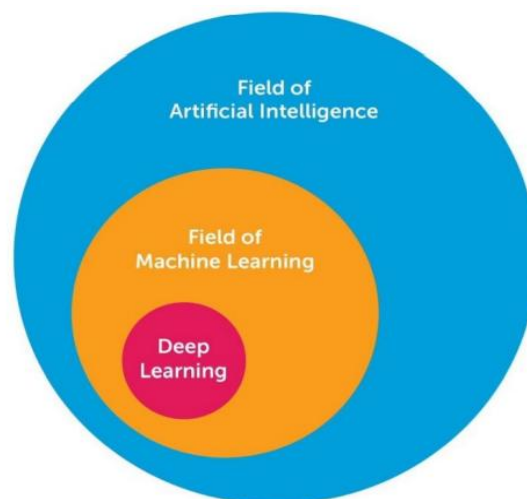
To better understand the concept, a real-life example such as targeted advertisements, can facilitate explaining it. By collecting the ads a user clicked on in a period of time and features that describe his decision, a model can be trained to predict whether that user will visit a certain advertisement site or not. When the features and labels are fed to a decision tree, it will generate some rules. Then a prediction can be made whether the advertisement will be clicked or not. In the

other hand, a model trained with Random Forest will build several decision trees based on random observations and features.

Not to mention that decision trees are exposed to overfitting, while random forest mostly avoid this problem by creating random subsets of the features and building smaller trees using these subsets. Finally, those subtrees will be combined. However, this technique is a double-edged sword, since it slows down computation in case the forest has a large number of trees. Which leads in some cases such as real-time prediction to avoid implementing Random Forest and look for an alternative. As a conclusion, Random Forest are more suitable for use cases that doesn't require big datasets and detection time don't play a big role in the application to avoid any possible complications. For more complex tasks it is preferable to implement another approach

## 1.5 Deep Learning

Since Artificial Intelligence came a while ago, it has a wide range of applications and it's divided into many branches [8].. Deep Learning is a subset of machine learning, which is in itself a subfield of AI. The figure below is a visual representation of the relationship between AI, ML and DL [11].



**Figure 1.14: Relationship between AI, ML and DL (Le 2017, p.10)**

So, what is exactly deep learning and what kind of problems it solves? This question will be answered in depth in the next section.

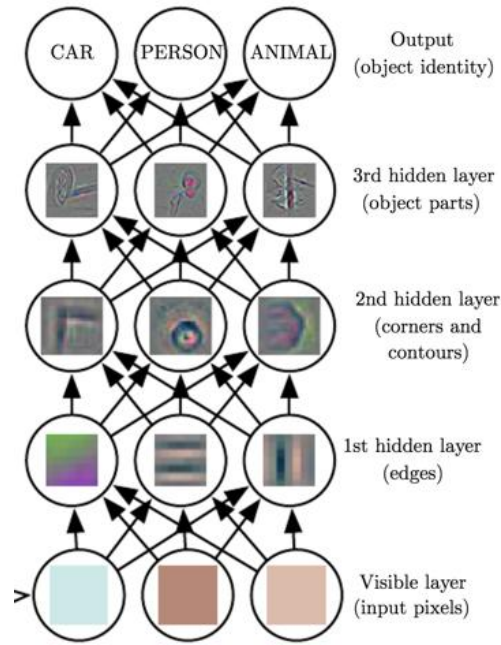
### 1.5.1 Deep Neural Networks:

AI managed since its existence to solve many tasks that meant to be intellectually challenging for humans, yet, it struggled with problems that seems easy and intuitive for human beings such as face or speech recognition. The reason behind this, is many difficult tasks can be translated to mathematical rules which is easy for a computer to understand. In the other hand, other tasks that seems to be easy are hard to be described formally.

A good approach to solve these intuitive problems is to give freedom to computers to learn from previous experiences and understand the world by interacting with it in terms of a hierarchy of concepts. This way humans won't need to specify all the knowledge to computers anymore. This concept allows the computer to solve difficult concepts by building them from simpler ones. If this hierarchy is represented by a graph, it will be formed by many layers and defined by deep. That's why this approach is called deep learning. Since no camera have the quality of the human eye or the no computer can correlate information like the human brain, it became difficult for real-world artificial intelligence applications to extract high-level, abstract features from raw data because of the constant variation of the observed data, such as the change of the angle view in different images and color variation under different circumstances. Because these factors of variation are only identified by nearly human level understanding of the data, it seems at first that such a task is impossible by just representation learning. Here where it comes the role of Deep Learning, since it's based on representing abstract features in terms of other simpler representation. A practical example of this concept is demonstrated by the figure 3.9 where an image of a person (complex concept) is detected by looking for different characteristics in a layer format, such as corners, contours and edges (simple concepts).

The feedforward deep network or multilayer perceptron is considered as a typical example of a deep learning model. Explain it is a mathematical function mapping some set of input values to output values. The function is formed by composing many simpler functions. We can think of each application of a different mathematical function as providing a new representation of the input.

Deep Learning is not only allowing to represent data in the right way, but by adding the concept of depth in its models, it allows the computer to learn multistep computer program. Where each layer of the representation can be thought of as the state of the computer's memory after [11].



**Figure 1.15: Illustration of a deep learning model (Goodfellow et al. 2016, p.6)**

Images which represent scenes from our reality are interpreted by computers as a collection of pixel values. The task of identifying an object or mapping its identity from those values is a difficult task for machines and can be nearly impossible when trying to learn this mapping directly.

Deep Learning approaches this obstacle by introducing the concept of layers where each layer represents simple mappings extracted from the global complex mapping and nested with each other. As shown in the figure, there are two types of layers. The visible layer, which represents the input that's only observed by humans, and the hidden layers that extracts features from the image. The reason behind calling this type of layers —hidden is because their values are not given in the input and must be determined by the model through figuring out which concepts are useful for explaining the relationship in the observed data. As shown in Figure 3.9, every hidden layer visualizes a feature. The first layer is responsible for detection edges, which will help the second hidden layer to detect more complex features. In this case 22 corners and contours. With assembling the detected features, the third hidden layer can figure out connections and detect

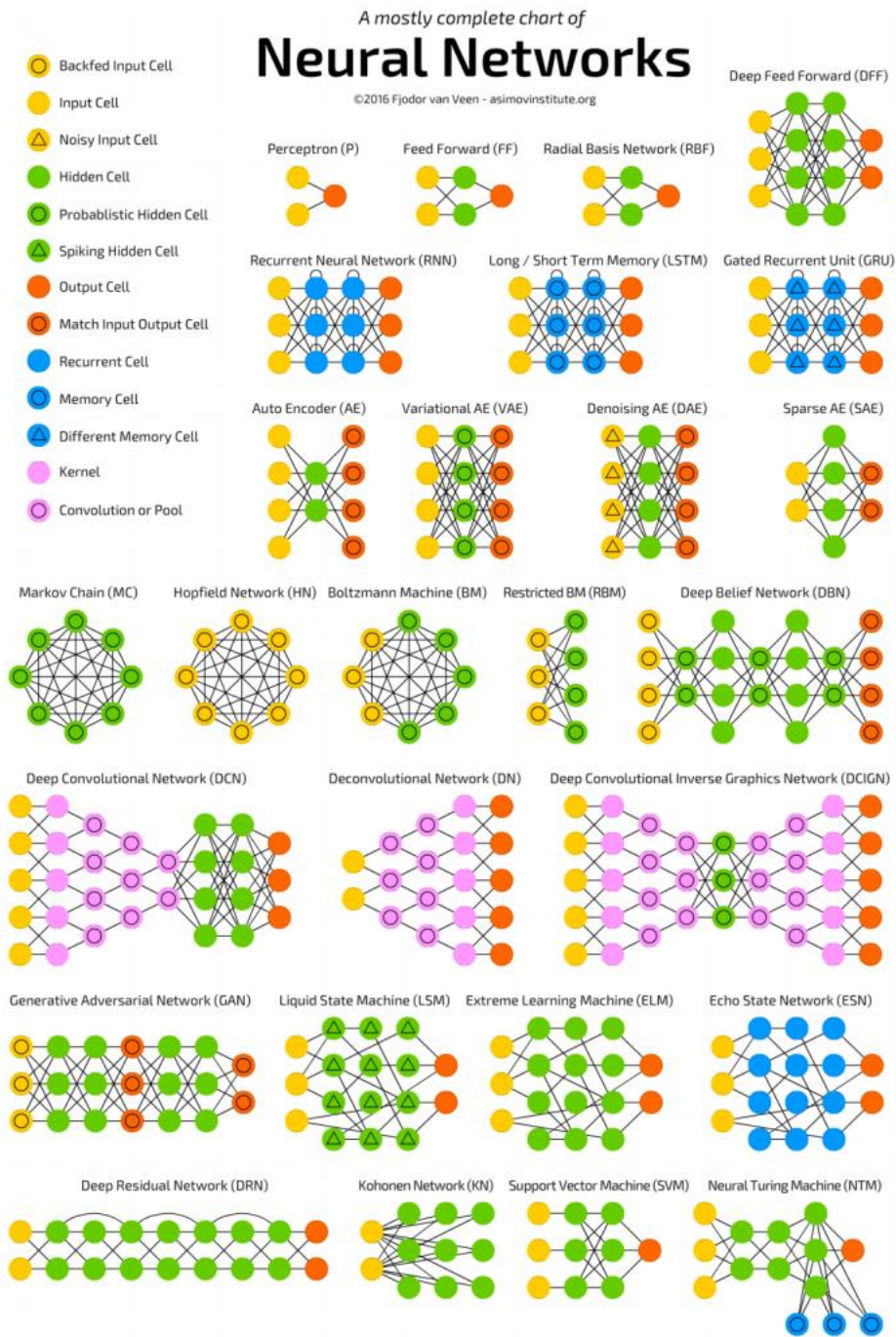
object parts. As a final step, the final object in the input image can be detected by comparing the features from the last hidden layer with the classes provided in the training [11].

An easier way to understand deep learning globally is with some historical context. Goodfellow et al. identified in their book [8]. four key trends of the history of deep learning:

- Deep learning has had a long and rich history, but has gone by many names, reflecting different philosophical viewpoints, and has waxed and waned in popularity.
- Deep learning has become more useful as the amount of available training data has increased.
- Deep learning models have grown in size over time as computer infrastructure (both hardware and software) for deep learning has improved.
- Deep learning has solved increasingly complicated applications with increasing accuracy over time.

To review, the field of Artificial Intelligence encapsulate numerous subfields, such as Machine Learning. Deep Learning is a technique for implementing ML that gives machines a learning curve when provided by data and that's possible thanks to its representation of different features of real data as layers, where each feature defined in relation to simpler features, and more abstract representations computed in terms of less abstract ones[8].

Neural networks are one of the most used computing systems in the field of Machine Learning. They are cable of finding concepts and patterns which are very difficult for humans to figure out. This powerful ability is due to the way this tool imitates how human neural system works [11].



**Figure 1.16: NN types chart (Van Veen 2016, p.1)**

Since the list of networks in the chart is extensive, only the relative ones to the study will be explained.

## 1.5.2 Recurrent Neural Networks

Humans don't start thinking from scratch every second. By reading this essay, you understand each word according to your understanding of the preceding words. You don't throw everything away and start thinking from scratch again. Your thoughts have perseverance.

Traditional neural networks (TNNs) can't do that, and that seems to be a significant gap. For example, imagine that you want to classify the type of event that occurs at each stage of a movie. It is not known how a traditional neural network could use its reasoning on previous events in the film to inform them later.

RNN are a type of Neural Network where the output from the previous steps are fed as input to the current step.

- RNN can predict the future.
- They can analyze time-series data such as stock prices, and tell you when to buy or sell. In autonomous driving systems, they can anticipate car trajectories and help avoid accidents.
- They can work on sequences of arbitrary lengths, rather than on fixed- sized inputs like all the nets, we have considered so far.
- Useful for natural language processing applications such as automatic translation or speech-to-text.

The simplest possible RNN composed of one neuron receiving inputs, producing an output, and sending that output back to itself (figure -left). • We can represent this tiny network against the time axis, as shown in (figure- right). This is called unrolling the network through time. Recurrent Neural Network (RNN) [11].

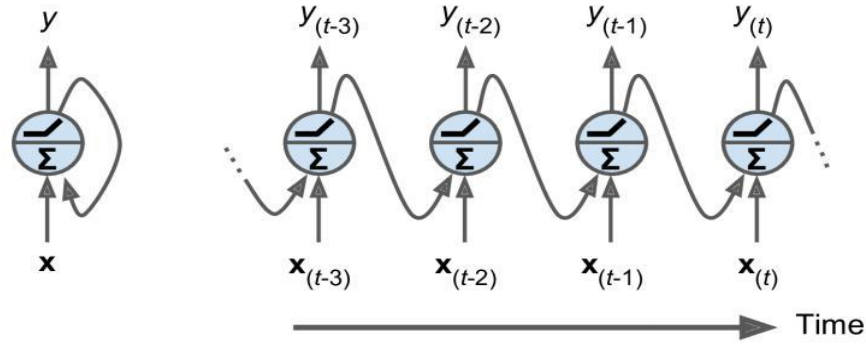


Figure 1.17: unrolling the network through time

You can easily create a **layer** of recurrent neurons. At each time step  $t$ , every neuron receives both the input vector  $\mathbf{x}(t)$  and the **output vector** from the previous time step  $\mathbf{y}(t-1)$ .

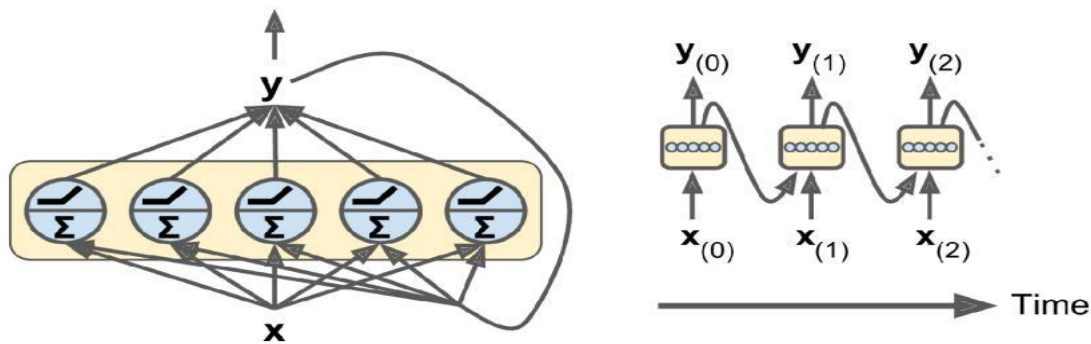


Figure 1.18: A recurrent neuron (left) unrolled through time (right)

### 1.5.3 Convolution Neural Networks:

Good fellow et al. define CNNs in their book [8] a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. This type of network is based on a linear mathematical operation, which is convolution, hence the name Convolutional Neural Network. The concept here, is to replace simple matrix multiplication in one or several layers of the network. This approach proved a huge success in several real-world use cases [11].

## 1.6 Conclusion

In this chapter we have presented the basic concepts of Machine learning and deep learning including its operating principle, its main components and also its limitations. Then, regarding machine learning, we talked about Random forest (RF) and Gradient boosting Machine (GBM), about these methods and their use, and we mentioned the pros and cons of both types.

Then we described a new variant of neural network called Deep Learning (DL). This technique is characterized by its ability to solve the problem of the complexity of training (NN) as well as its power to represent the forms (inputs) in a powerful, automatic and discriminating way.

## CHAPTER 2

## **2 CHAPTER 2. DESCRIPTION OF CEMENT GRINDING BALL MILL SYSTEM**

### **2.1 Introduction**

Cement manufacturing is a complex process that begins with mining and extract raw material that include limestone and clay, then grinding this material to a fine powder, called raw meal, after is heated in a cement rotary kiln at temperature higher than 1450 °C. In this step of process a several chemical change of material was done witch result a new product called clinker. The clinker is grinded in the cement mill with gypsum and limestone to create cement, the end user mixed the powdered cement with water and aggregates to form concrete that is used in construction.

Cement grinding is a crucial stage of the cement production, where both cement quality and process efficiency are decided. Achieving the required fineness with maximum efficiency is down to the design of the ball mill and how well it runs. An inefficient system can push your energy bill right up, making it a lot more expensive to produce the desired product quality. Poor design could also result in overgrinding, inadequate mixing and even inefficient product quality.

In this chapter, brief overview of the cement finish grinding area with the main equipment erected in cement grinding process, ball mill, separator, feed system and the dedusting filters. Also we will see the operation process and different control loops used to reach the process stability.

## 2.2 Main equipment in grinding area

### 2.2.1 Ball Mill



**Figure 2.1: Cement Ball Mill, Lafarge M'sila**

The ball mills, figure 2.1, used for grinding cement have usually two chambers separated by a metallic diaphragm in as shown the Figure 2.2. The first compartment is filled with bigger balls and lifting liners to do coarse grinding. The second compartment is filled with a smaller ball and classifying liners to do the fine grinding. Classifying liners ensure that the bigger balls accumulate at the beginning of compartment and smaller balls toward the end [2]

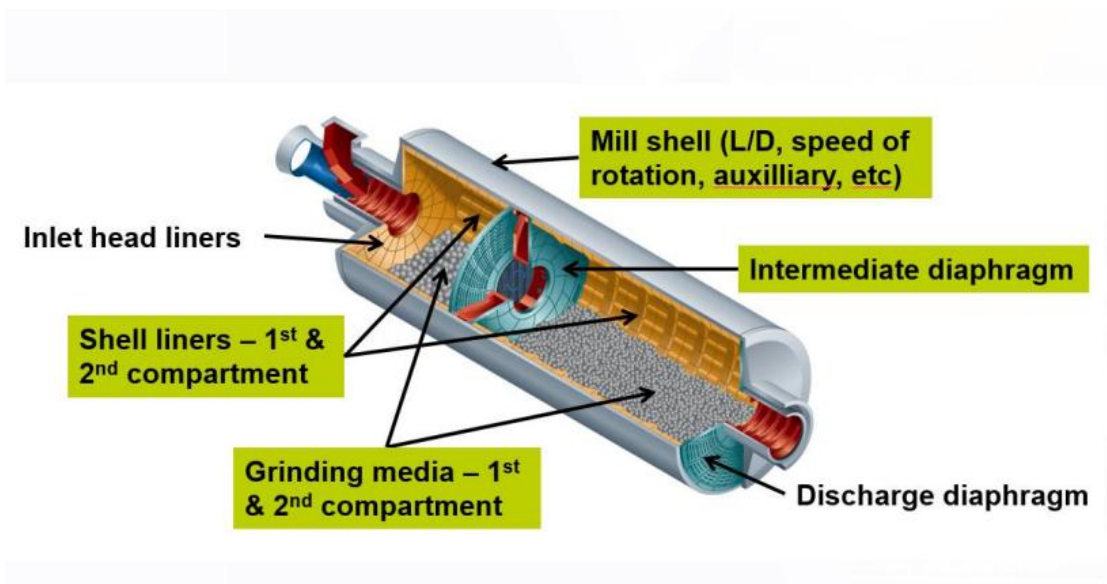


Figure 2.2: Design of mill internals FICEM\_Christian Pfeiffer\_2017

### 2.2.2 Feed system

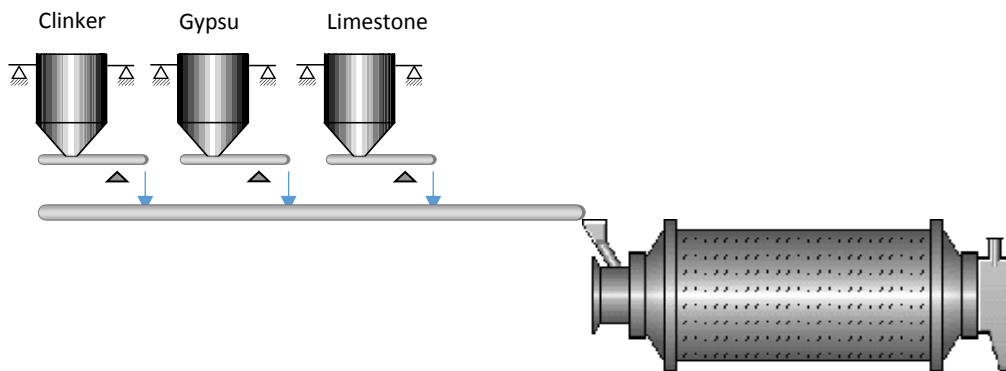


Figure 2.2.3 Cement Feeding system

In order to produce cement with certain quality properties, often the clinker must be mixed with other additive components like gypsum, limestone, slag, pozzolan, depending on the cement type. All component are stored in separate storage and transported to the mill feed bins, then adjusted through weight belt feeders below the mill feed bins before the mixture enters the mill as shown the Figure 2.3 [1]

In the figure 2.4 an example of feeders in Lafarge M'Sila cement plant.



Figure 2.4: Limestone and gypsum feeders in Lafarge M'Sila

### 2.2.3 Separator

The separation is performed by the division of a given material flow into two separate streams, using air as the carrying medium as shown in figure 2.5. One stream should contain the fine particles, and the other, the coarse particles [12].

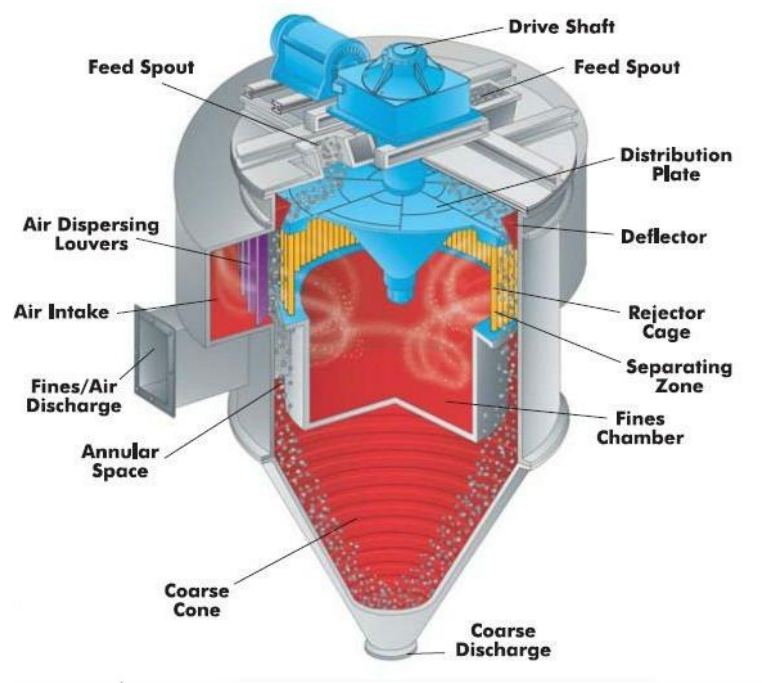


Figure 2.5 : Example of Cement air separator

The material feed inside the separator is subjected to classifying forces. The centrifugal force created by the speed of rotating plate, ascending air current and gravity force. The fine

particles are passed through the separator to become finished product while the coarse particles are returned to the mill as rejects for further grinding as shown in figure 2.5 [12].

The air separator is required to separate the final cement out of the mill product, the air flow required for the separation is generated by a circulation air fan, figure 2.6.

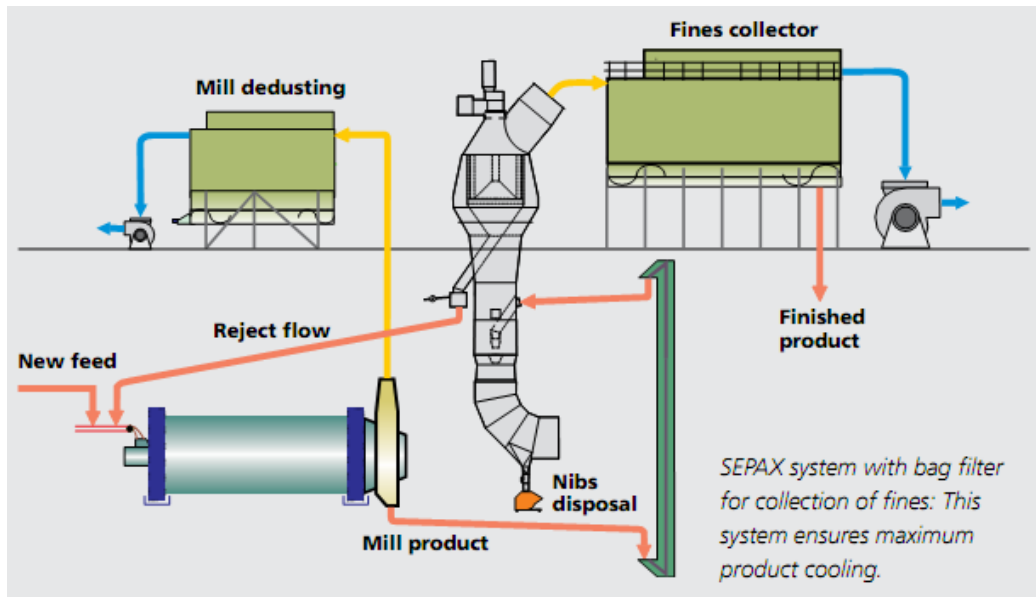


Figure 2.6 FLSmidth Sepax air separator

#### 2.2.4 Mill bag filter for cleaning of mill ventilation air

The Mill ventilation air is generated by mill fan used for cooling and de-dusting the mill, this air passed through a bag filter to separate material and air. The product collected by this filter transported to the separator to classify the coarse and finished product.

#### 2.2.5 Separator bag filter

This filter, called fines collector, separate the fine product from the from the recirculation air which is drawn by separator fan. After filtering, the finished product transported outside the bag filter and the purified gas is discharged into the atmosphere

## 2.3 Process description'

The materials clinker, gypsum and sometimes an additive, are fed in preset recipe to the inlet of the ball mill via three separate weigh feeders, one for each material. A proportioning controller, if installed, can keep a constant ratio between the components in accordance with the selected set-points. All the new material is fed to the ball mill. The speed of the feeders is controlled by the mill capacity [3].

The cement ball mill is divided into two compartment and use vibration or noise sensors to measure the mill fill level.

The ventilation fan of the ball mill ventilates the mill and draws (aspirate) the ventilation air from the mill to a bag filter. The fan and damper should also maintain a constant negative pressure at the mill inlet to keep it free of dust as shown in figure 2.7.

Water vapors developed from dehydration of the gypsum and possible moistures from the gypsum as well as vapors coming from the injection of water at the outlet of the mill are removed from the mill by the ventilation airflow [13].

The mill product is transported via a screw conveyor, a mill elevator and air slides to the separator. In the separator the dispersed material is divided into two streams. The coarse rejects are returned to the mill for further grinding, via a conveyor belt or air slide equipped with a weighing unit. The fine product is transported out of the separator by the separator air and collected in the previous mentioned bag filter.

The collected finished product is transported to the cement silos.

The air from the separator is drawn by the Separator fan to the bag filter, de-dusting this air flow. The major part of the airflow is recycled to the separator. The remaining part of the air flow which corresponds to the false air ingress into the separation system, is released to the atmosphere via the chimney.

The air flow through the separator is controlled by the damper in front of the separator fan. If the air flow is too low, then open the damper in front of the separator fan, until the flow is correct which is indicated by no material falls out at the outlet for material drop-through.

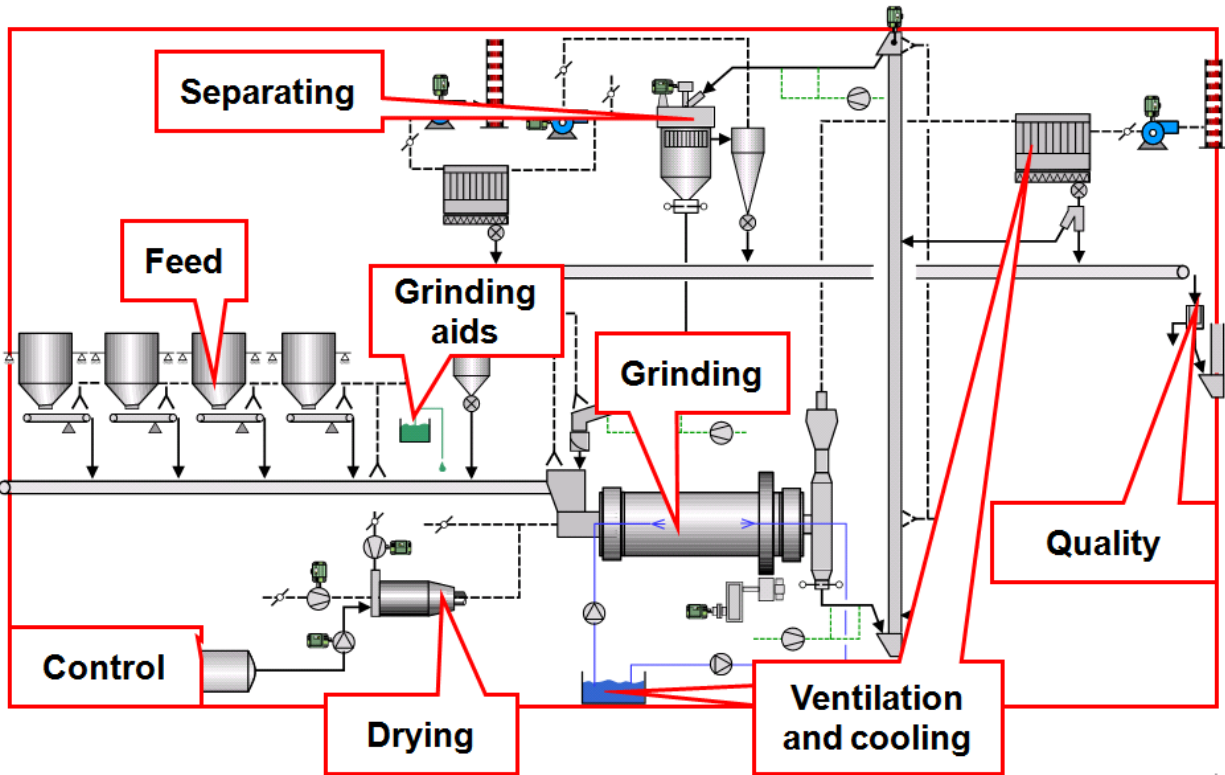


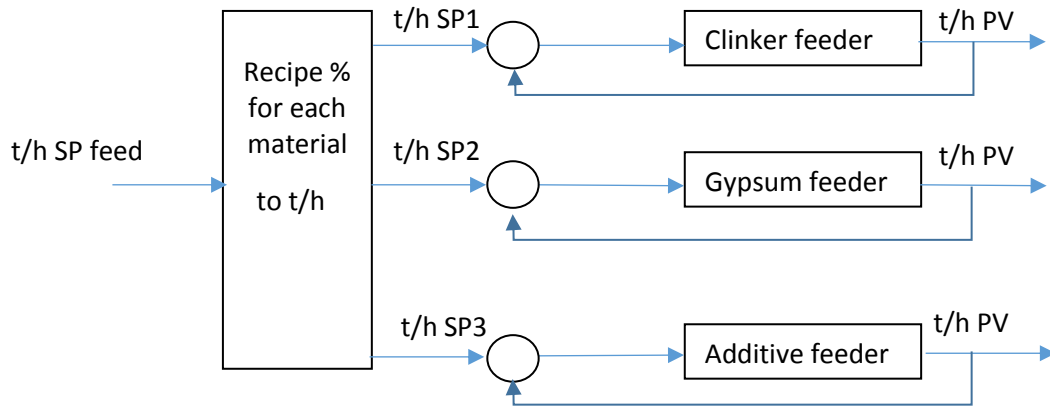
Figure 2.7 : Cement grinding with ball mill systems. Infinity for cement equipment

## 2.4 Control loops

In order to maintain operational stability the system is equipped with a number of control loops. They will take corrective action when set in automatic mode. No controller should be set in automatic mode until the input process value has reached stability at the same level as the controller set-point. The different control loops installed are [13]

### 2.4.1 Recipe of the feed to the mill:

The feed of the mill consists of three weight feeders; clinker, gypsum and an additive. The flow of material controlled by the weigh signals from the respective feeders and speed. They are used to maintain the proportion of the feed components. This means that the above proportional regulation between the three types of materials keeps the proportion constant in percentage, in any case of any changes in the total quantity of the new feed as shown in Figure 2.8.

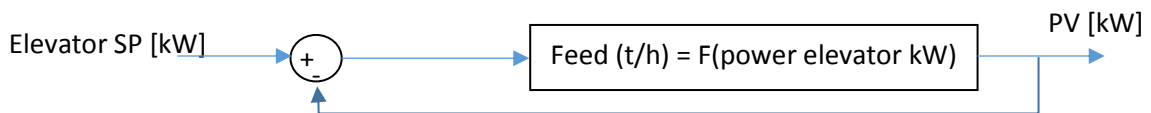


**Figure 2.8 : Recipe of the feed to the mill**

### 2.4.2 Constant feed to the mill

The total mill feed was based on the actual measured fineness compared with the target fineness. The total feed is the total of the fresh new feed and the separator rejects returning to the mill inlet, however, still maintaining the constant composition of the feed.

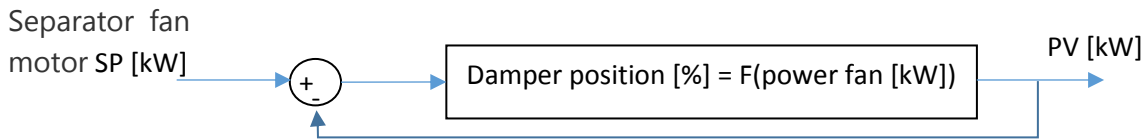
The set-point for the total mill feed is supervised by the power consumption measured on the motor of the mill bucket elevator. This set-point for the total mill feed will automatically be reduced if: The power consumption of the mill elevator is above a set maximum value as shown in figure 2.9.



**Figure 2.9: Mill feed is supervised by the power of elevator**

### 2.4.3 Air flow through separator

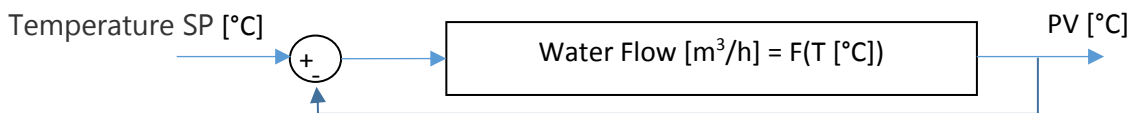
The airflow through the separator is controlled by maintaining a constant power consumption of the separator fan motor. The damper at the fan inlet is used for airflow control.



**Figure 2.10: Air flow control through separator fan power**

#### 2.4.4 Mill outlet temperature

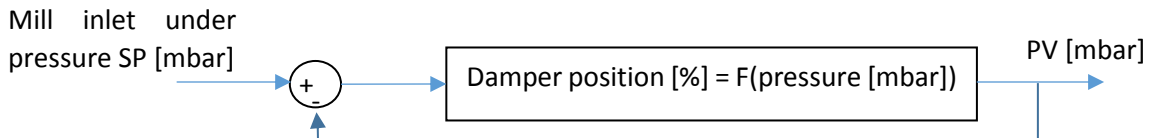
When the mill is in operation, then the mill outlet temperature is kept constant by injection of water at the mill outlet, figure 2.11.



**Figure 2.11: Mill outlet temperature control**

#### 2.4.5 Constant mill inlet under pressure

The mill inlet under pressure is maintained by adjusting the damper of the mill ventilation fan. An under pressure must be present when the mill is producing, in order to maintain the inlet dust free as shown in Figure 2.12.



**Figure 2.12 : Mill inlet under pressure control**

All the above mentioned control loops can be operated in manual mode.

## 2.5 Cement Quality control

The quality control is done by the cement analysis in laboratory by special equipment; the important elements controlled are the SO<sub>3</sub> and fineness (R45).

Cement quality is measured by its chemical composition and its particle size distribution. Blaine is an aggregate number for the particle size distribution measuring the specific surface area of the cement powder.

The result of analysis are the inputs of the quality control system software, it generate new parameters of mill recipe and the operator adjust the set points according to control card of mill operation.

## 2.6 Cement Fineness

Usually fineness is measured hourly by sample analysis in the laboratory

How are the properties of concrete affected by the fineness of the cement? The Effect of Cement Fineness on Concrete is:

Increasing the fineness of a cement increases the amount of mixing water required to achieve a given consistency and reduces the amount of bleeding by the concrete.

The 28-day compressive strength resistance decrease by increasing fineness because the deceasing of elasticity modulus of concrete [14].

The fineness has to be monitored to achieve the product quality specifications.

Cement fineness affecting ball mills grindability, so for every 100 cm<sup>2</sup>/g increase in cement fineness increases the mill power consumption by 1-2 kWh/t for a closed circuit mill & 2-3 kWh/t for an open circuit mill, a sheer waste if this is not required by the market! [15].

The mill is utilized for production of OPC 32.5 grade, OPC 42.5 grade and SRC 52.5 grade. The Blaine's and reject 45 micrometer for the cements of different grades are being maintained as given below:

Product	Blaine Range (cm <sup>2</sup> /g)	R45 (%)
OPC 32.5 grade (Chamile)	4500-4700	10.5-13.5
OPC 42.5 grade (Matine)	4100-4300	8.5-10.5
SRC 52.5 grade (Moukauim)	4500-4700	7.3-10

## 2.7 Case Study

Lafarge cement M'Sila plant is having a closed circuit ball Mills for cement grinding

The plant consist of four identical clinker-grinding installations, however, only one is described.

### 2.7.1 System description

Nominal production feed rate 140 t/h

Electrical consumption mean value 42 kWh/t for mill department.

Electrical consumption mean value 32 kWh/t for mill motor.

Feed system Type Dosimat feeder and Schenck feeder.

Mill Size 46 x 15.5 m.

Main Drive 4716 kW.

Mill Speed 14.6 rpm

Separator Sepax 375M-122 open with 240 kW motor.

Mill Bag filter, for cleaning of mill ventilation air. Type CEH2-18-45

Separator Bag filter Type D24-4.5-2712.

Separator fan Type MTSS 224/224 with 495 kW motor.

Mill ventilation fan Type SPW 100/100

## **2.8 Conclusion**

We have seen in this chapter an overview of grinding cement ball mill with their important parts, then the grinding process operation description with their control loops. Also, we have presented the quality control with the impact of fineness to the cement quality and concrete resistance. The next chapter, we will use a machine learning approaches for predicting the cement fineness, based on the results of laboratory tests and the preprocessed input real data collected from operational of ball mill in M'sila cement plant.

# CHAPTER 3

# 3 CHAPTER 3. MACHINE LEARNING APPROACHES FOR PREDICTIVE MODELS

## 3.1 Introduction

The control of fineness is necessary in order to obtain cement with the desired consistent day-to-day performance.

In this chapter, we will describe the features of the data used and shows the steps of data collected and preprocessing. Also the different techniques of regression to predict fineness, linear regression, kernel ridge regression, random forest regression and neural network model. Finally we give the evaluation of each model.

## 3.2 Proposed system:

The present work considers the prediction of ball mill grinding fineness, which have relation with different features. The fineness values are impacted by large variations in different parameters (feed material, separator speed, reject flow ...). Thus, a model capable of predicting the cement fineness in real-time can improve the two indicators of the cement grinding circuit performance: energy consumption and product quality. The predictive fineness designed with the model from fast sample data (1 min sample) when applied to control with hourly sampled measurements

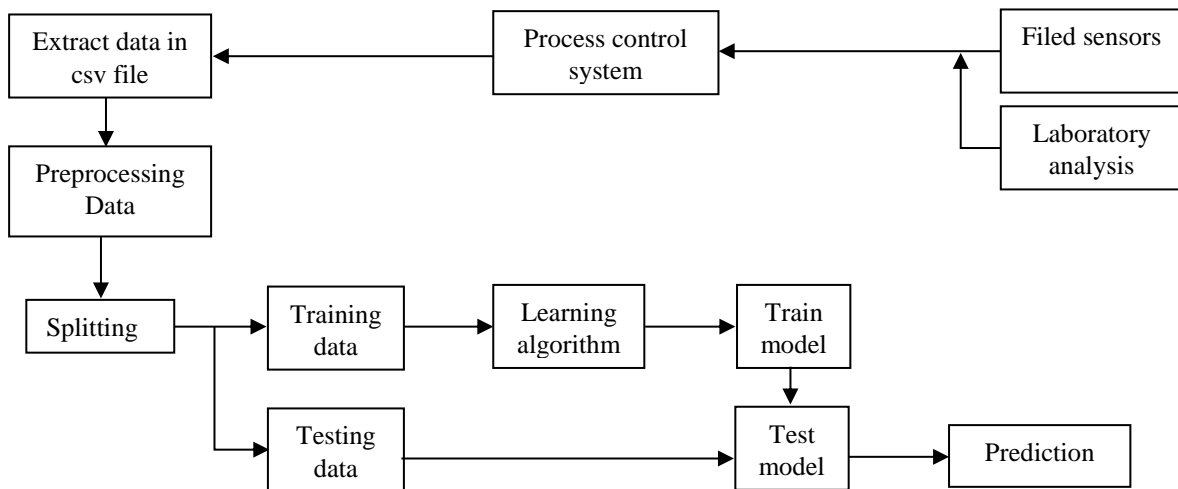


Figure 3.1: Architecture of prediction steps

**Training Dataset:** The sample of data used to fit the model.

**Testing Dataset:** The sample of data used to provide an unbiased evaluation of a final model fit on the training dataset.

**Dataset:** A dataset consists of about two components, the two components are rows and columns. In addition, a main feature of a record is that it is organized in such a way that each row contains an observation.

**Machine learning Algorithm:** An “algorithm” in machine learning is a procedure that is run on data to create a machine learning “model”.

**Machine Learning Model :** A “model” in machine learning is the output of a machine learning algorithm run on data.

### 3.3 Dataset

We will use a real data of operational cement ball mill, which was extracted from control system in Lafarge Cement of M’sila (LCM) located in Hammam Dalaa.

#### 3.3.1 Dataset knowledge

In this data we see a several features affect the fineness of cement. So we discuss brief of each features.

**RHR:** Running hours: [h] the duration of the mill running in hour.

**Production\_Rate:** [t/h] is the total feed of mill resulted by the sum of the feed rate of each feeder (clinker, gypsum and Limestone).

**Clinker:** percentage provided by the clinker feeder according the cement recipe, the clinker is nodules of sintered material produced by heating a homogeneous mixture of raw materials in a kiln to a sintering temperature of approximately 1450°C for modern cements [16].

**Gypsum:** percentage provided by the clinker feeder according the cement recipe, adding gypsum in the cement is to slow down the hydration process of cement once it is mixed with water.

**Limestone:** percentage provided by the limestone feeder according the cement recipe, adding the limestone in the cement grinding increasing the capacity of cement production, and result in lowering product costs.

**Mill\_outlet\_temp:** Mill outlet cement temperature is an important parameter that must be controlled; the high temperature of cement will affect the construction performance of cement

**Rejects\_flow:** [t/h] is the flow of coarse particles leaving the separator and returns to the ball Mill, the reject flow rate at separator outlet is an image of fineness of the cement.

**Sep\_Speed:** [%] is the percentage of separator speed, the separator speed provides a centrifugal force to separate the course from fine particles, is very necessary features to correct the fineness.

**MD\_Power:** Main motor power [kW] has an effect with the variation of cement fineness

**BE\_power:** Becket elevator power [kW], give use the circulating cement load

**R45:** The residue at 45  $\mu$ m, particles size distribution, this analysis consists of passing the collected known material through a sieve to analyze the material retained. This procedure is performed once every two hours

**SO3:** it is obtained by cement analysis with special equipment. The time, for grinding was extended with relatively high SO<sub>3</sub>, the hardness of clinker was correlated with the clinker SO<sub>3</sub> and the adding of gypsum increase the SO<sub>3</sub>.

The flowing table describe the variables used in this model and their limits

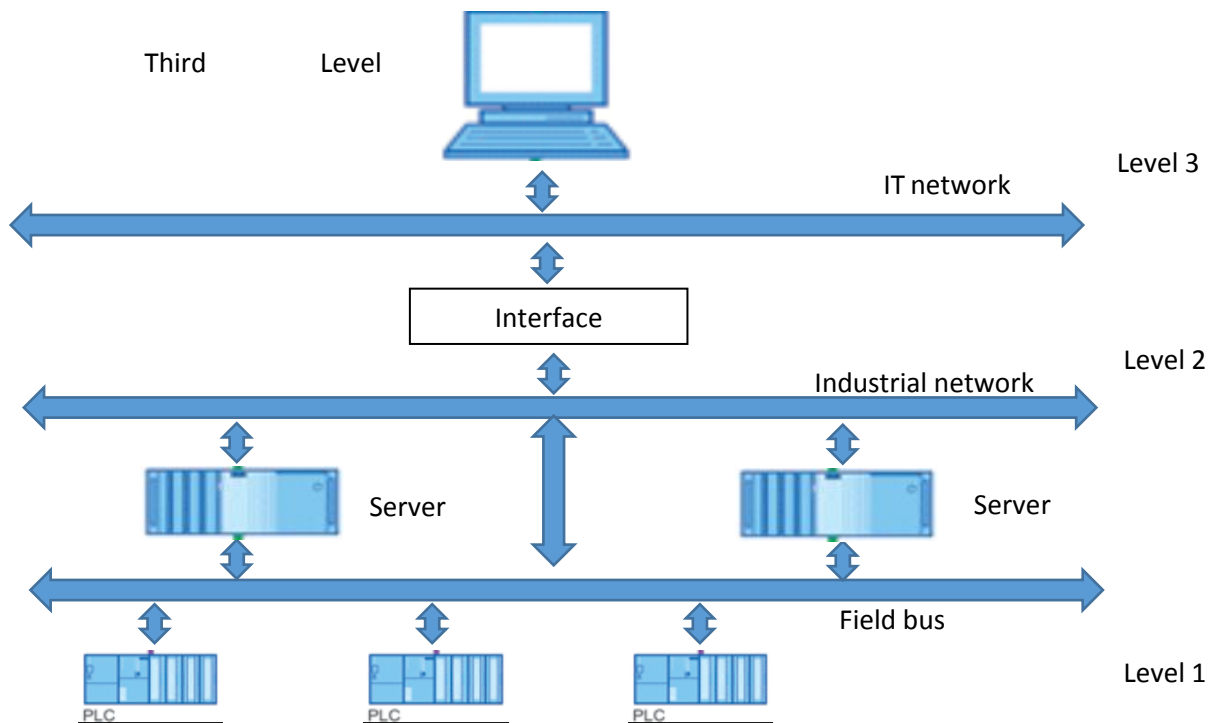
Variable Name	Unit	Notation	Limit H	Limit L
Running hours	H	RHR	1	0.9
Production rate	t/h	Production_Rate	135	168
Clinker percentage	%	Clinker	65.2	94.9
Gypsum percentage	%	Gypsum	2.8	7.1
Limestone percentage	%	Limest	0	27.3

Mill outlet temperature	°C	Mill_Outlet_Temp	88	125
Reject flow	t/h	Reject_flow	0	255
Separator speed	%	Sep_Speed	46.3	61.3
Main drive power	kW	MD_Power	4400	4900
Bucket elevator	kW	BE_Power	22.9	38
Sieve 45 micron	%	R45	7	12.5
Sulfur trioxide	%	SO3	1.03	2.84

**Table 3.1 : Variables data**

### 3.3.2 Data export to CSV file:

We use a functionality of Technical Information System (TIS) to export data from process control system to an CSV file, this software is a third level of control in the plant as shown in the figure 3.2.



**Figure 3.2 : Third level architecture**

### 3.3.3 Data Collected

In the table 3.2 we shown some lines of data collected from cement plant of ball mill operation

RHR	Production_Rate	Clinker	Gypsum	Limest	Mill_outlet_temp	Rejects_flow	Sep_Speed	MD_Power	BE_power	R45	SO3
1	155.4904418	72.60342582	4.397475709	22.99909847	104.9628359	212.0755208	54.83748061	4511.588379	36.02481308	8.5	2.01
1	149.3069075	72.60588358	4.402253286	22.99186313	102.650191	219.5130193	54.83163287	4479.320955	35.89109643	8.6	2.05
1	149.5593075	72.81816288	4.394917725	22.78691939	102.0616313	212.0542539	54.84230296	4469.18339	35.14561857	8	2.02
1	152.6481983	72.48239976	4.703601673	22.81399857	102.0399301	220.1790365	54.41755892	4484.284212	35.99778748	7.8	2.1
1	157.7416637	72.51184834	4.699842022	22.78830964	108.5538733	195.1610255	53.82269885	4494.487793	34.86444728	8.5	2.02
1	145.1741384	73.47639485	4.506437768	22.01716738	113.9690937	141.1892362	50.00078354	4609.281616	29.37402093	9.1	2.06
1	148.9035263	72.9707113	4.60251046	22.42677824	107.4803785	215.1286891	55.0766784	4504.031372	36.0994929	8.3	2.01
1	140.2600474	73.02609661	4.586340922	22.38756247	105.1079648	221.1006938	55.07191556	4514.59139	35.98047161	8.6	2.16
1	132.4405907	73.00952605	4.598377043	22.39209691	106.4672849	223.6015612	55.07300072	4524.49139	35.54927419	8	2.03
1	155.8815014	73.22142286	4.79616307	21.98241407	111.6698315	190.6147985	54.56175321	4533.355184	33.19735975	10	2.08
1	145.2701166	72.9066152	4.792537794	22.300847	107.9507558	216.878906	56.75956033	4451.218205	36.00265274	9	2.12
1	141.5472881	72.88732394	4.797535211	22.31514085	107.5127494	207.8763013	56.75323079	4454.485189	35.14839204	7.5	2.14
1	149.3027245	72.93209555	4.787733389	22.28017106	107.9618771	217.2628034	56.68878988	4433.50376	36.01953729	8.5	2.1
1	158.2723104	72.89186264	4.791892158	22.3162452	107.8508386	158.9422747	54.17576866	4556.910604	32.20022055	8.9	2.08
1	146.234621	72.9286475	4.792332268	22.27902023	107.9517497	217.3943146	54.90577888	4484.237972	36.09314919	8.7	2.04

Table 3.2: Part of data collected

### 3.3.4 Correlation matrix

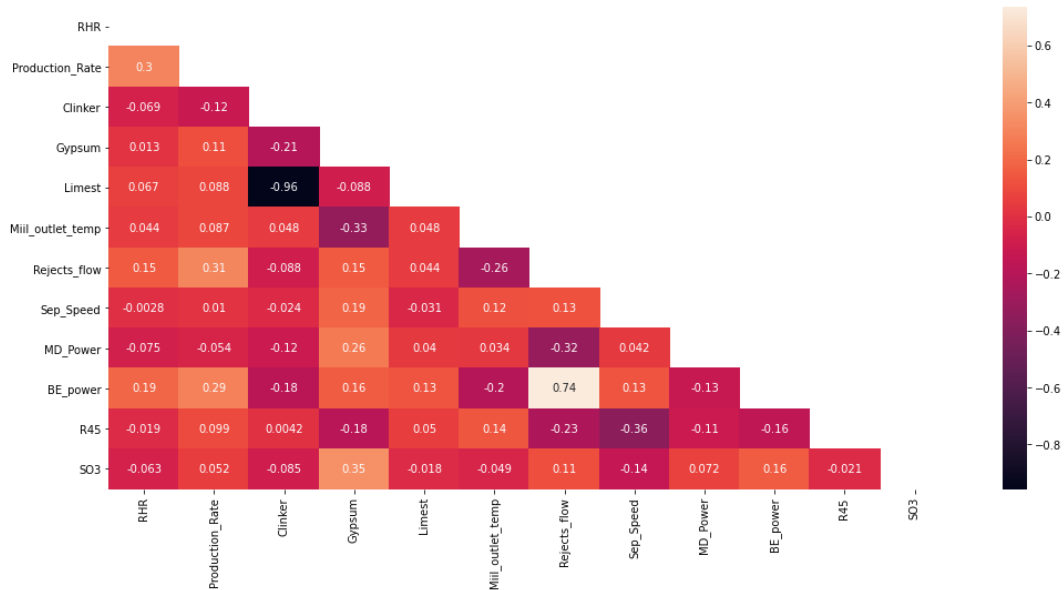


Figure 3.3: Correlation Matrix

The correlation matrix is generated to define the characteristic of data and the relation between features and output, figure 3.3.

### 3.3.5 Trend chart

To see better the relation of fineness according the input features, we chose the first thirty data and plot a trend for the important correlated data with fineness.

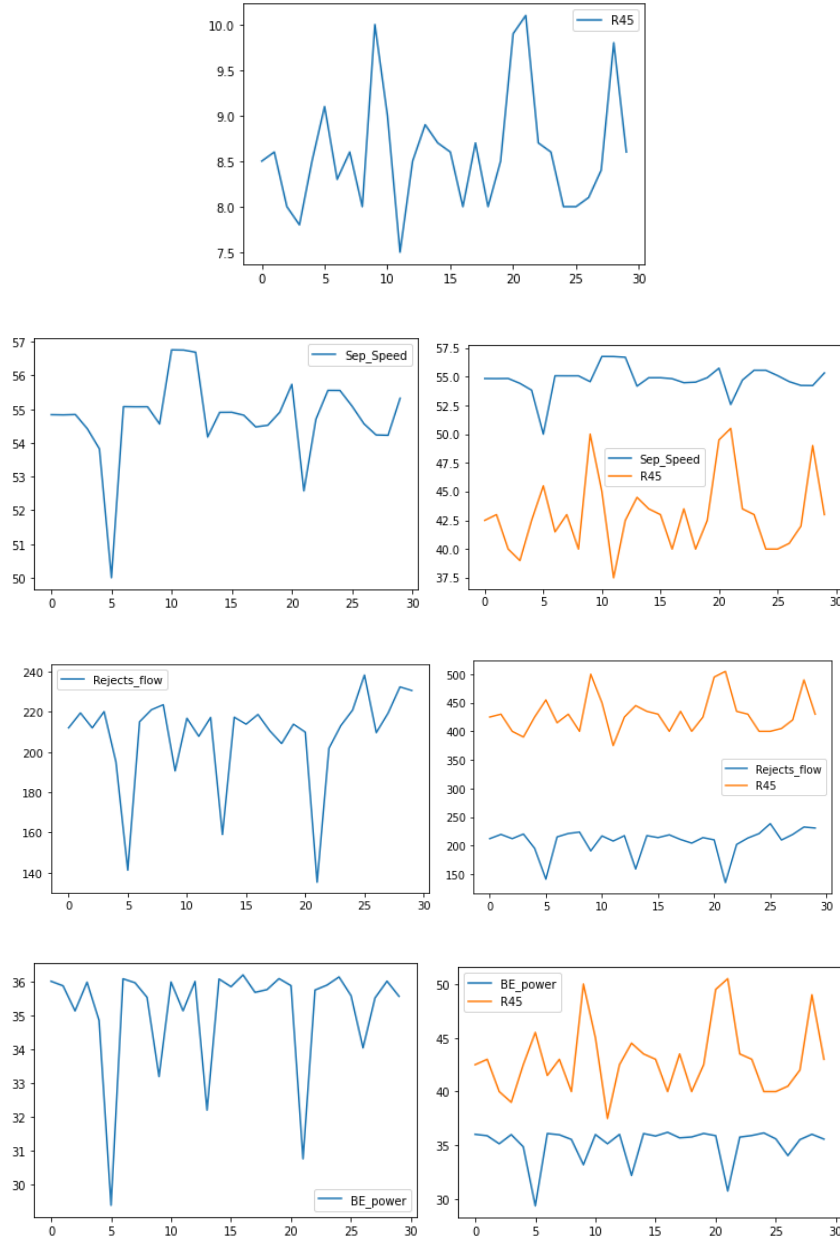


Figure 3.4: Trend chart of correlation of fineness

## 3.4 Development environment

### 3.4.1 Google Colab

Google Colab is a free cloud service intended for training and research in artificial intelligence, based on Jupyter Notebook. This platform allows us to develop and train machine learning models directly in cloud without install python editors in our computer. To use Google Colab just go to drive and execute Colaboratory figure 3.5 and figure 3.6 [17].

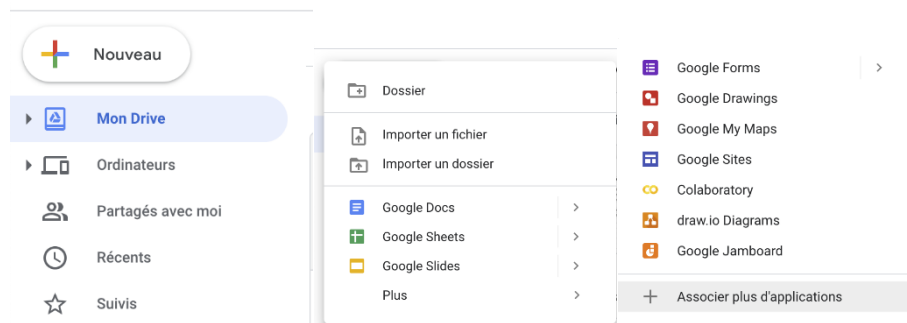


Figure 3.5: How to run Colaboratory



Figure 3.6: Google Colab editor

### 3.4.2 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include:

data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more [18].



**Figure 3.7: Jupyter Notebook Logo**

### **3.4.3 Python**

Python is an object oriented scripting language that was released publicly in 1991. It was developed by Guido van Rossum of the National Research Institute for Mathematics and Computer Science in Amsterdam. Python has rapidly become one of the world's most popular programming languages. It's open source, free and widely available with a massive open source community [19].



**Figure 3.8 Python logo**

In our case, we have use Google Colab to avoid the installation and downloading every time the missing Library.

## **3.5 Preprocessing data**

Data preprocessing is a technique used to convert data to understandable and consistent data. The real data generally contain a missing values, errors and noisy data, so the preprocessing technique is used to correct these problems [20].

### 3.5.1 Steps involved in data preprocessing

- Importing the required Libraries.
- Importing the data set.
- Handling the Missing Data.
- Encoding Categorical Data.
- Splitting the data set into test set and training set.
- Feature Scaling.

#### *Importing the required Library:*

For further process, we have to import some important libraries present in python.

```
import numpy as np
import scipy as sp
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

#### *Importing the data set:*

We use Google drive to store the data set because it is more practical and easy to use data

```
# Connect to the Data
from google.colab import drive
drive.mount("/content/drive")
# Reading data
df = pd.read_csv("/content/drive/MyDrive/Data/Data2.csv")
```

#### *Handling the Missing Data:*

You can see that there are certain numbers of null values present in each feature, so we have to impute any other value to fill null values. The *SimpleImputer* class of *sklearn* library provide basic strategy for imputing missing values. The following snippet used to replace missing values by the mean value of the columns.

```

num= df.select_dtypes(include=['int64', 'float64']).keys()
from sklearn.impute import SimpleImputer
impute=SimpleImputer(strategy='mean')
impute_fit= impute.fit(df[num])
df[num]= impute_fit.transform(df[num])

```

### ***Handling outlier:***

An outlier is a point that is distant from other observed points, and in machine learning it is important to remove them prior to training your model as it may potentially distort your model. There are a number of techniques to remove outliers, and in this chapter we use Tukey Fences.

Tukey Fences is based on Interquartile Range (IQR). IQR is the difference between the first and third quartiles of a set of values. The first quartile, denoted Q1, is the value in the dataset that holds 25% of the values below it. The third quartile denoted Q3, is the value in the dataset that holds 25% of the values above it. Hence, by definition,  $IQR = Q3 - Q1$  [21].

In Tukey Fences, outliers are values that are as follows:

- Less than  $Q1 - (1.5 \times IQR)$ , or
- More than  $Q3 + (1.5 \times IQR)$

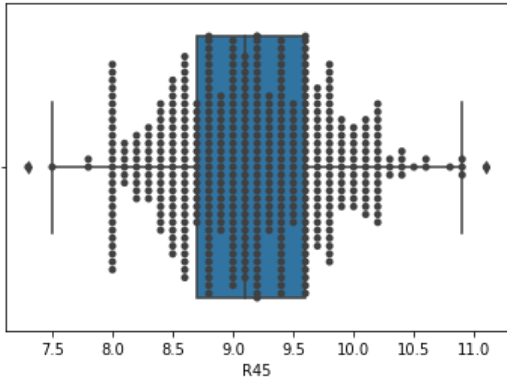
The following code snippet shows the implementation of Tukey Fences:

```

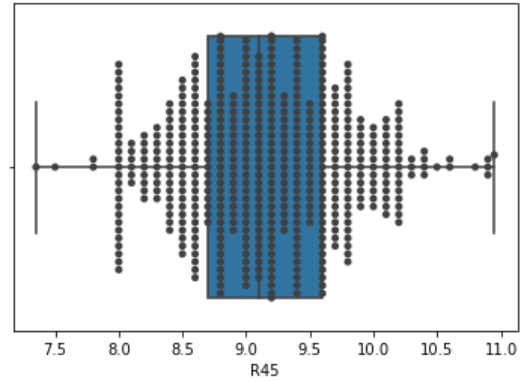
# Remove Outlier
def remove_outlier(col):
    sorted(col)
    Q1, Q3 = col.quantile([0.25, 0.75])
    IQR = Q3 - Q1
    print("Q1 = ", Q1, " Q3 = ", Q3, " IQR = ", IQR)
    lower_range = Q1 - (1.5 * IQR)
    upper_range = Q3 + (1.5 * IQR)
    return lower_range, upper_range

```

In our example the figure 3.9 data before handling outlier values and in the figure 3.10 you see two values are outlier.



**Figure 3.9: Before handling outlier**



**Figure 3.10: After handling outlier**

### *Encoding Categorical Data:*

As Scikit-learn only works with fields that are numeric, you need to encode string values into numeric values. Fortunately, the dataset contains all numerical values, and so no encoding is necessary. After check we see the following information about data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 505 entries, 0 to 504
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   RHR                    505 non-null    float64
1   Production_Rate       505 non-null    float64
2   Clinker                505 non-null    float64
3   Gypsum                 505 non-null    float64
4   Limest                 505 non-null    float64
5   Mill_outlet_temp      505 non-null    float64
6   Rejects_flow          505 non-null    float64
7   Sep_Speed             505 non-null    float64
8   MD_Power              505 non-null    float64
9   BE_power              505 non-null    float64
10  R45                    505 non-null    float64
11  SO3                    505 non-null    float64
dtypes: float64(12)
```

### *Splitting the data set:*

Splitting data set train and test sets is one of the important steps in Machine Learning. Because our model needs to be evaluated before it has been deployed. The main idea is to convert original data set into 2 parts

- train
- test

Where train consists of training data and training labels and test consists of testing data and testing labels.

In the following code the easiest way to split data into inputs and output also splitting data to train and test sets by using scikit-learn.

```
# Split into Input and Output Elements
my_data = pd.DataFrame(data=df)
y = df["R45"].values
X = df.drop(["R45"],axis=1).values
print("X : ",X.shape," y : ",y.shape)
# Split into train and test Elements
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.33, random_state=42)
```

### 3.6 Model evaluation metrics

Evaluation metrics explain the performance of a model, so it is necessary to use mathematical tools to calculate the accuracy of our model, in the following paragraphs some metric tools:

***Mean absolute error (MEA):***

The absolute error is the difference between the predicted values and the actual values. Thus, the mean absolute error is the average of the absolute error, the formula is given by:

$$\text{MAE} = \frac{1}{n} \sum_{j=1}^n |y - y_i| \tag{3.2}$$

- n = The number of errors
- y = Measured values
- yi= predict values.

### *Mean Squared Error (MSE)*

The mean squared error or MSE is similar to the MAE, except you take the average of the squared differences between the predicted values and the actual values it is given by:

$$MSE = \frac{1}{n} \sum_{j=1}^n (y - y_i)^2 \quad 3.3$$

### *Root Mean Square Error (RMSE)*

The root mean square error is the square root of the second sample moment of the differences between predicted values and observed values or the quadratic mean of these differences it is given by the formula:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{j=1}^n (y - y_i)^2} \quad 3.4$$

- $min_{(x)}$ : The minimum value of  $X$ .
- $max_{(x)}$ : The maximum value of  $X$ .

## **3.7 Simulation result**

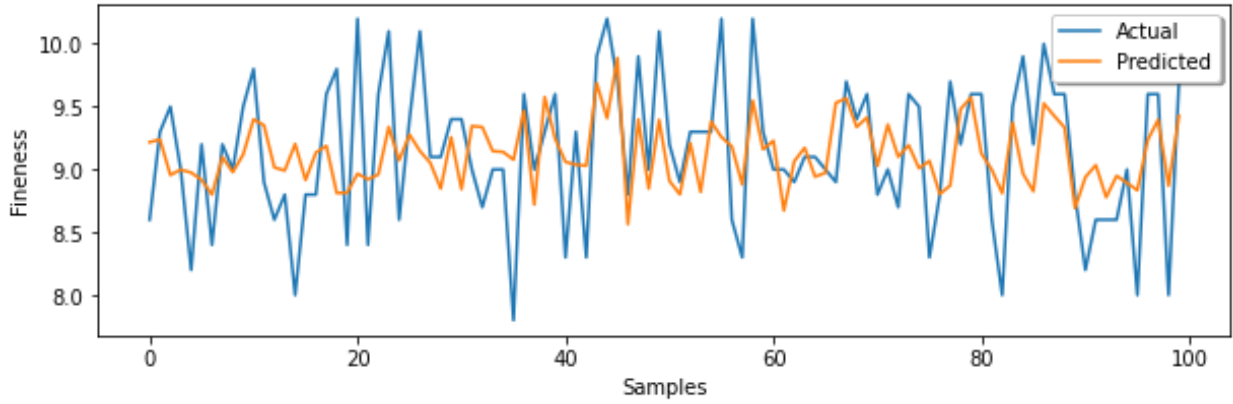
To predict the fitness result we will use different known algorithms

### **3.7.1 Linear regression (LR)**

In general, you'll use a linear model to make a prediction by calculating a weighted sum of the input features. Let's use the linear regression function from the scikit-learn library to find the values of fitness and calculate the accuracy:

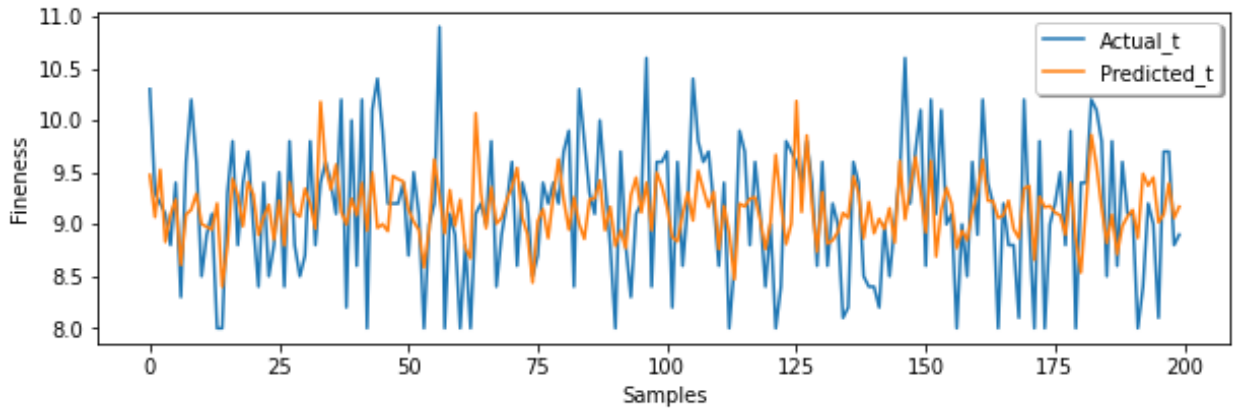
```
Mean Absolute Error      : 0.40247736671698736
Mean Squared Error      : 0.24802074264189186
Root Mean Squared Error : 0.49801680959772016
```

The figure 3:11 represented the comparison between actual and predict fitness result of test values.



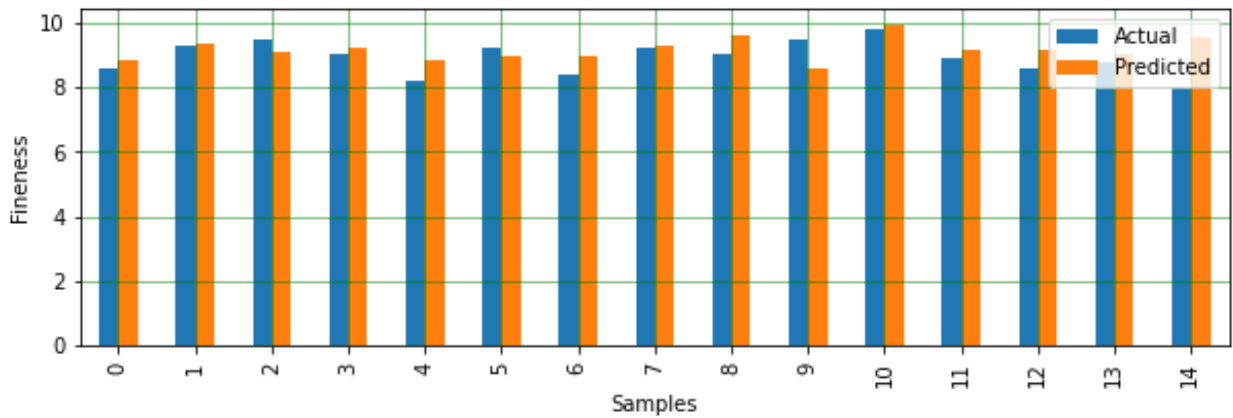
**Figure 3.11: Actual and predicted output for linear regression of test values**

The figure 3:12 represented the comparison between actual and predict fitness result of training values.



**Figure 3.12: Actual and predicted output for linear regression of train values**

The figure 3:13 represented the histogram of actual and predict fitness result of test values



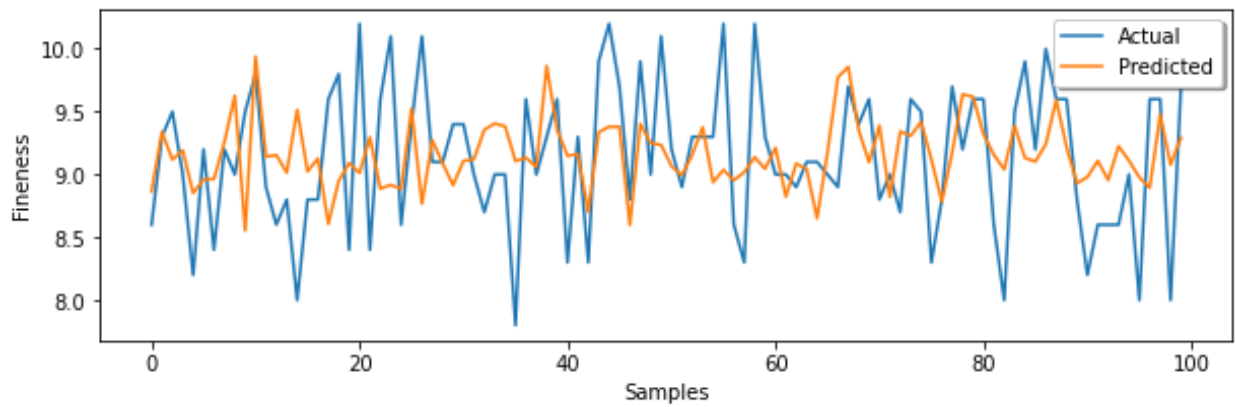
**Figure 3.13: Histograms of actual and predicted output for Linear regression of test values**

### 3.7.2 Kernel Ridge Regression (KRR)

In order to explore nonlinear relations of the regression problem we use the Kernel ridge regression algorithm, and the result of metric evaluation below:

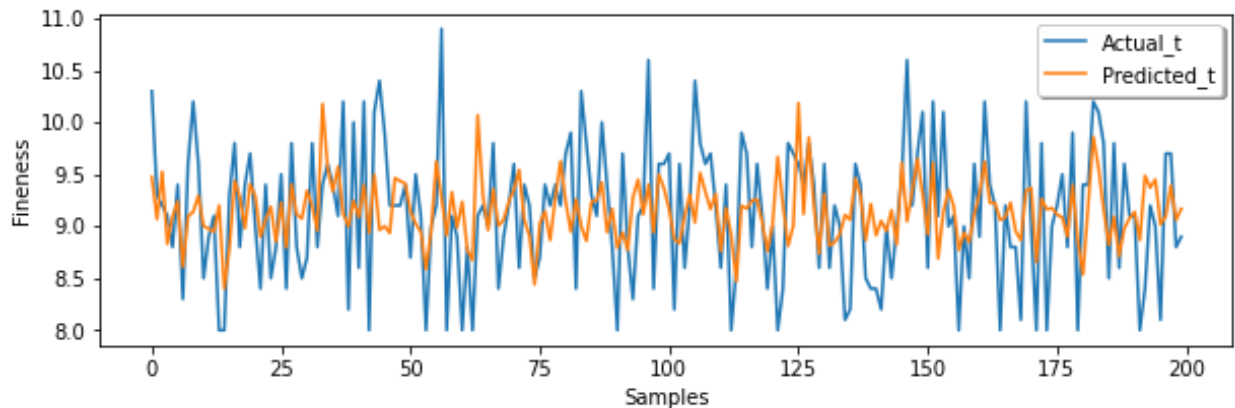
```
Mean Absolute Error      : 0.45868855750206666
Mean Squared Error      : 0.33423249686634265
Root Mean Squared Error : 0.5781284432255022
```

The figure 3:11 represented the comparison between actual and predict fitness result of test values.



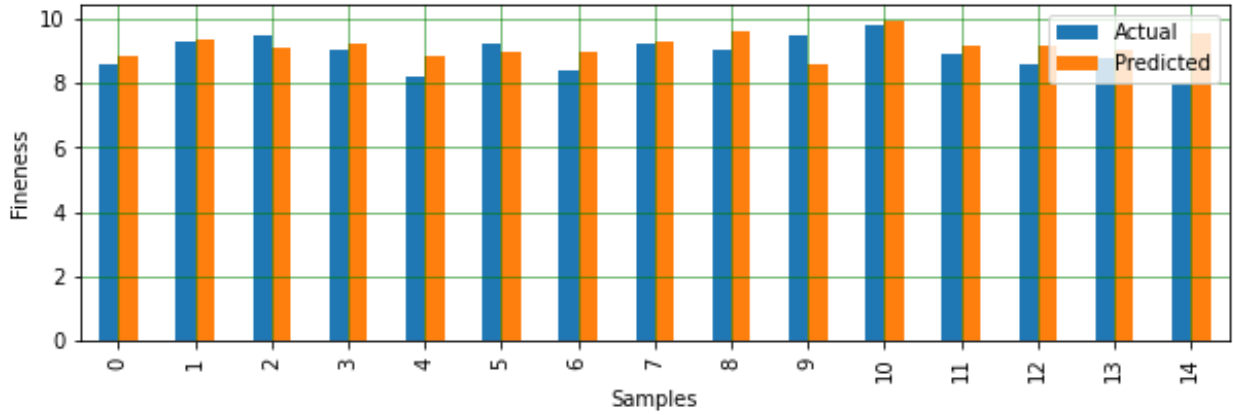
**Figure 3.14: Actual and predicted output for kernel ridge regression of test values**

The figure 3:15 represented the comparison between actual and predict fitness result of training values.



**Figure 3.15: Actual and predicted output for kernel ridge regression of train values**

The figure 3:16 represented the histogram of actual and predict fitness result of test values



**Figure 3.16: Histogram of actual and predicted output for kernel ridge regression of test values**

### 3.7.3 Random Forest Regression (RFR)

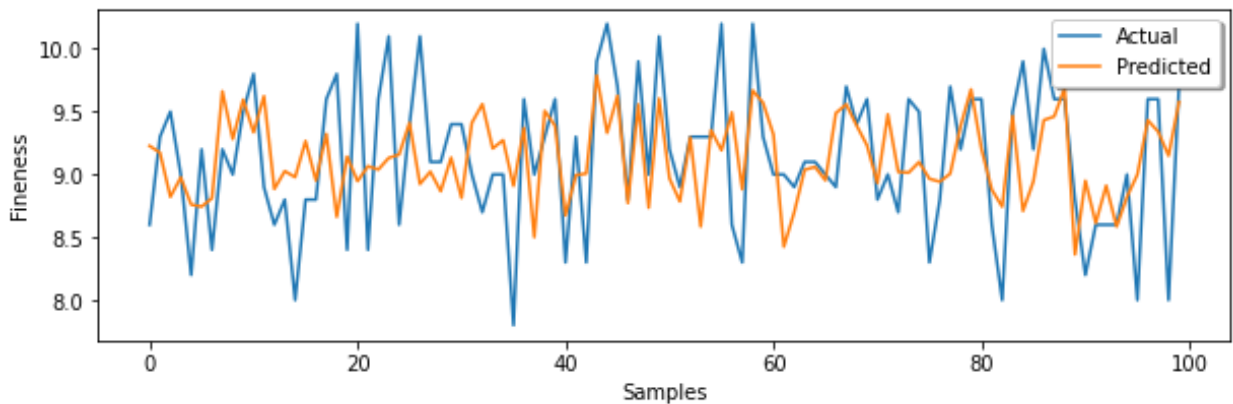
From sklearn.ensemble import the class RandomForestRegressor, the result of metrics evaluation are:

```

Mean Absolute Error      : 0.41889900990098805
Mean Squared Error      : 0.27858575683167996
Root Mean Squared Error: 0.5278122363413716

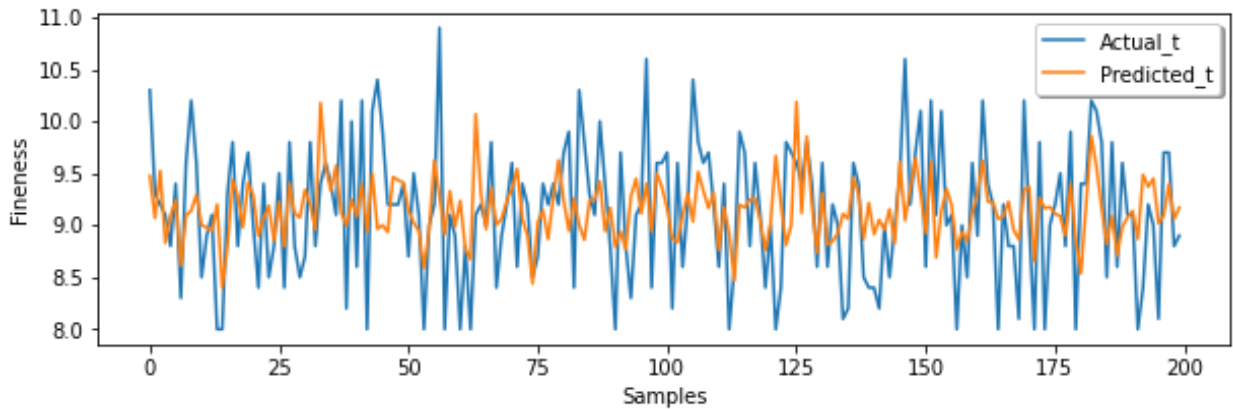
```

The figure 3:17 represented the comparison between actual and predict fitness result of test values using random forest regression.



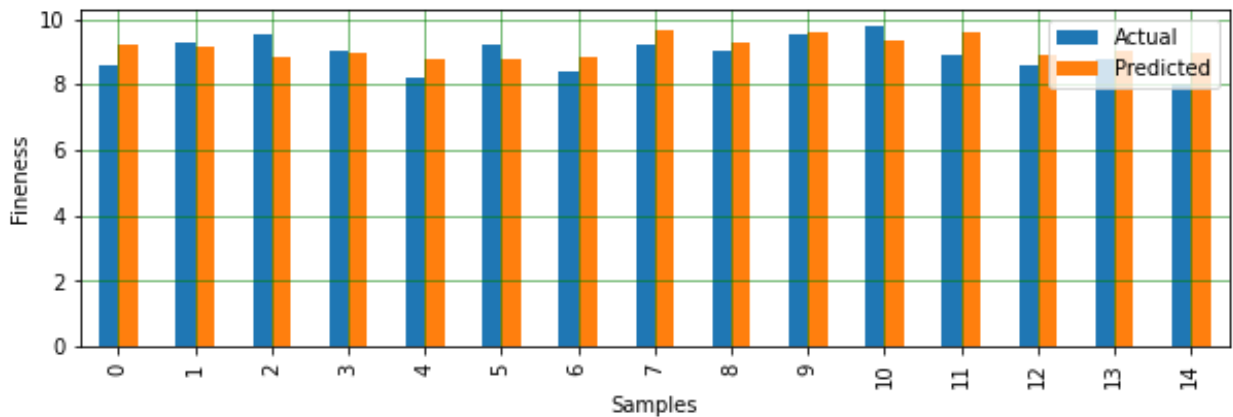
**Figure 3.17: Actual and predicted output for random forest regression of test values**

The figure 3:18 represented the comparison between actual and predict fitness result of training values.



**Figure 3.18: Actual and predicted output for random forest regression of train values**

The figure 3:19 represented the histogram of actual and predict fitness result of test values using random forest regression.



**Figure 3.19: Actual and predicted output for random forest regression of test values**

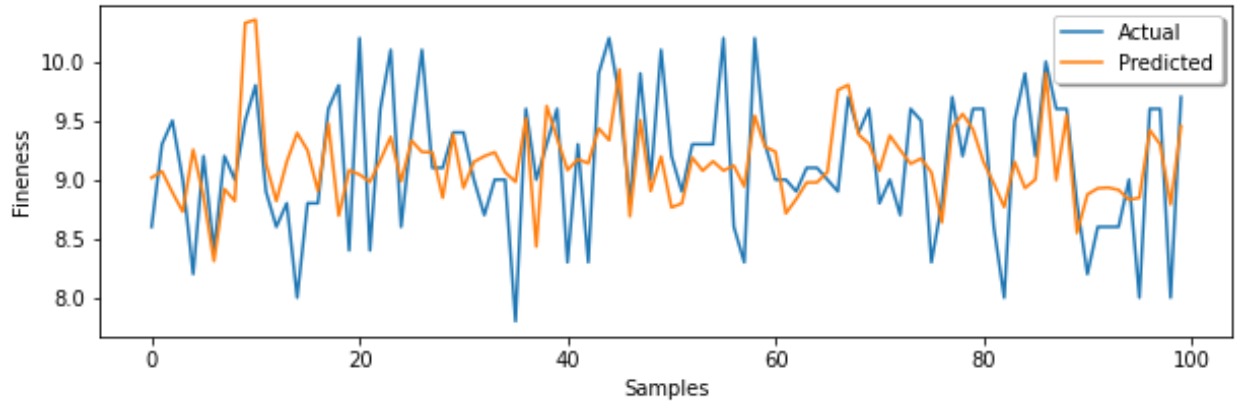
### 3.7.4 Neural Network model

Concerning the number of neurons in the hidden layer, we try an interval between 2 to 500 layers, the number 125 give us the minimum value of errors.

```

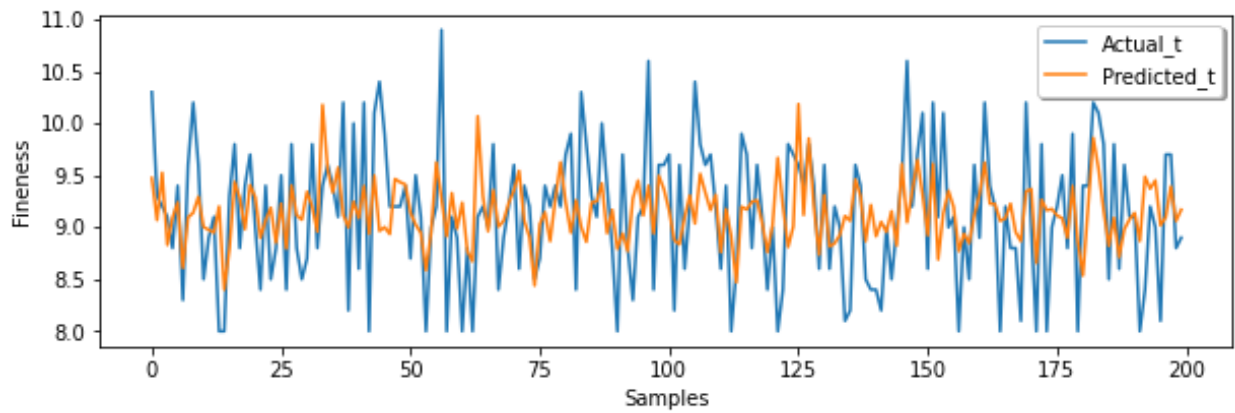
Mean Absolute Error      : 0.40880764873030667
Mean Squared Error      : 0.262730164267622
Root Mean Squared Error : 0.5125721064080858
  
```

The figure 3:20 represented the comparison between actual and predict fitness result of test values using neural network regression.



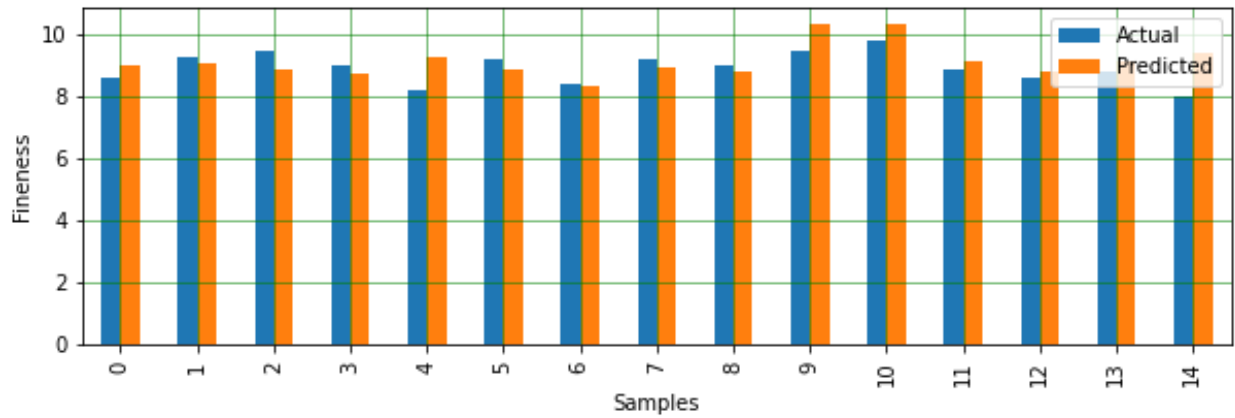
**Figure 3.20: Actual and predicted output for neural network model regressor of test values**

The figure 3:21 represented the comparison between actual and predict fitness result of training values.



**Figure 3.21: Actual and predicted output for neural network regressor of train values**

The figure 3:22 represented the histogram of actual and predict fitness result of test values using neural network model.



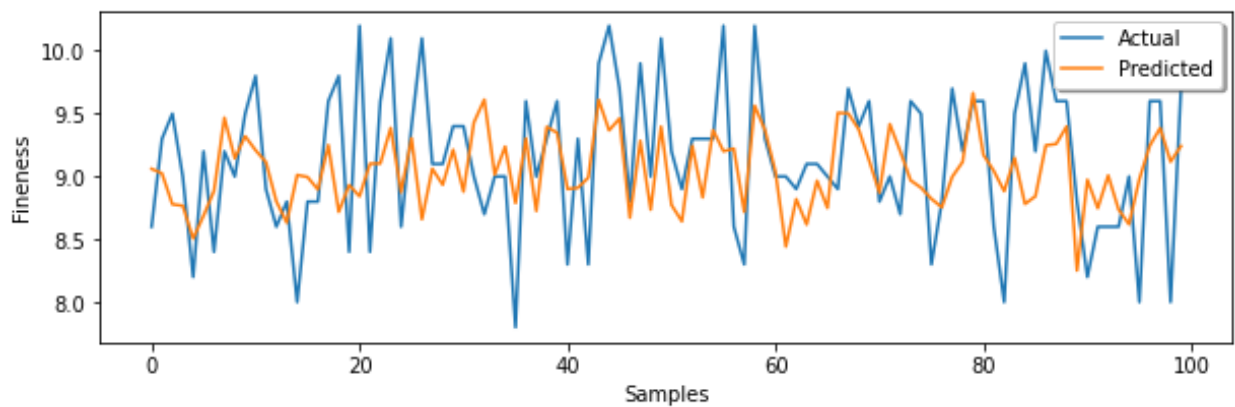
**Figure 3.22: Histogram of actual and predicted output for Multi Layer Neuron Network test**

### 3.7.5 XGBoost for Regression (XGB)

The result of several involved in regression give similar result, as compared to other methods

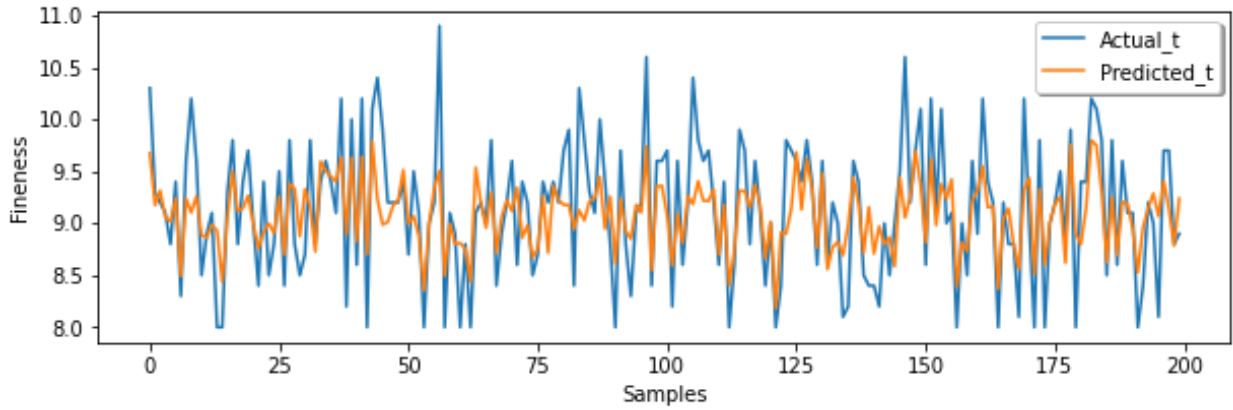
Mean Absolute Error : 0.4305752272653107  
 Mean Squared Error : 0.28060287909507725  
 Root Mean Squared Error: 0.5297196230979906

The figure 3:23 represented the comparison between actual and predict fitness result of test values using neural network regression.



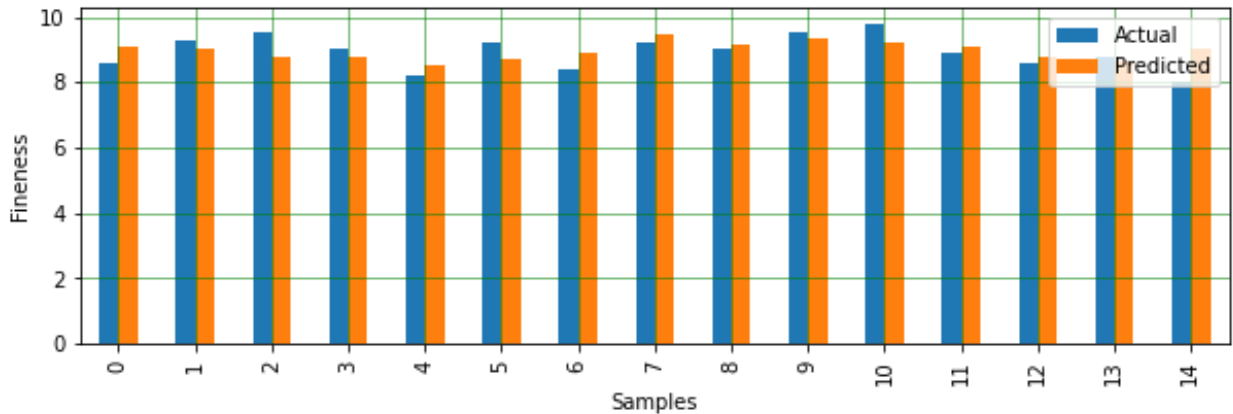
**Figure 3.23: Actual and predicted output for XGBoost Regressor of test values**

The figure 3:24 represented the comparison between actual and predict fitness result of training values.



**Figure 3.24: Actual and predicted output for XGBoost Regressor of train values**

The figure 3:25 represented the histogram of actual and predict fitness result of test values using neural network model.



**Figure 3.25: Histogram of actual and predicted output for XGBoost for Regression of test values**

We compare the evaluation criteria of techniques in table below; we note that the values are approximately similar.

All techniques give us an accepted result of predict fitness using artificial intelligent.

Technique	LR	KRR	RFR	ANN	XGB
MAE	0.403	0.459	0.419	0.409	0.431
MSE	0.248	0.334	0.334	0.263	0.281
RMSE	0.498	0.578	0.578	0.528	0.530

**Table 3.3: comparison Performance evaluation for different techniques.**

### **3.8 Conclusion**

In this chapter, we learn more about the techniques of machine learning and how can develop the adequate models to be applied in the industrial process.

The linear regression and neural network give the best result of fineness prediction more than kernel ridge regression and random forest regression after adjustment of parameters in each technique.

*CONCLUSION*

## CONCLUSION

In this thesis we have seen how can predict the fineness of cement grinding process from offline data extracted from a real industrial system from Lafarge M'sila plant. The preprocessing data was done in two steps, manually by selecting only the data of one cement product type and the second step of preprocessing was done with the programming. The techniques used in this work are linear regression, kernel ridge regression, random forest regression and neural network regression. The model that give the best result is linear regression and the multi layers neural network by comparison between the fineness obtained through laboratory results and the estimated by model.

The purpose of fineness predict is to create a soft sensor controlling cement fineness in cements industries. To reach this we must including in the model all cement type, and improve the model to have the minimum difference between the fineness obtained in the laboratory and predict by model.

The future research could include all type of cement and implemented the fineness predict model online in real factory during cement production to test the performance of the model. In addition, we can use this result of fineness prediction in control loops of separator speed control using a high-level control system.

# BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] LH\_Academy, Écrivain, *LH Cement Grinding and Blending*. [Performance]. LafargeHolcim, 2020.
- [2] M. GURUPRASATH, Robust Model Predictive control of Cement Mill circuits, TIRUCHIRAPPALLI-620015: DEPARTMENT OF CHEMICAL ENGINEERING., 2011.
- [3] K. Andreatta, «Soft Sensor for Online Cement Fineness Predicting in Ball Mills,» chez *International Seminar of Science and Applied Technology (ISSAT 2020)*, Espírito Santo, Vitória 29040-780, Brazil, 2020.
- [4] A. Géron, Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems., USA: O'Reilly Media, 2019.
- [5] R. J. Solomonoff, A new method for discovering the grammars of phrase structure languages. In COMMUNICATIONS OF THE ACM (Vol. 2, No. 7, pp. 20-20), vol. Vol 2, 1515 BROADWAY, NEW YOURK, 1959.
- [6] Z. Pawlak, Rough sets: Theoretical aspects of reasoning about data, vol. Vol. 9, USA: Springer Science & Business Media, 2012.
- [7] F. Rosenblatt, «The perceptron, a perceiving and recognizing automaton Project Para.,» Cornell Aeronautical Laboratory., Buffalo, Neu York, 1957.
- [8] J. L. R. D. E. & P. R. G. McClelland, Parallel distributed processing (Vol. 2, pp. 20-21)., Cambridge, MA, USA: MIT press., 1986.
- [9] C. Nikki, «Supervised vs Unsupervised Machine Learning.,» 2017. [En ligne]. Available: <https://www.datascience.com/blog/supervised-and-unsupervised-machine-learning-algorithms> . [Accès le 16 07 2018].
- [10] C. Nikki, «What is Semi-supervised Learning?,» 2018. [En ligne]. Available: <https://www.datascience.com/blog/what-is-semi-supervised-learning>. [Accès le 24 07 2018].
- [11] I. B. Y. C. A. & B. Y. Goodfellow, Deep learning (Vol. 1, No. 2). :, Cambridge, MA, New York: MIT press., 2016.
- [12] «INFINITY FOR CEMENT EQUIPMENT,» [En ligne]. Available: <https://www.cementequipment.org/home/air-separators-2/>. [Accès le 5 Jun 2021].
- [13] FLSMIDTH , «Mill plant (cement) Process instructions,. A/S, VIGERSLEV ALLÉ 77 DK-2500 VALBY, DENMARK,» VALBY, DENMARK, 2005.
- [14] Analytics Vidhya, «Concrete Strength Prediction Using Machine Learning (with Python code),» Concrete construction , [En ligne]. Available: [https://www.concreteconstruction.net/how-to/cement-fineness\\_o](https://www.concreteconstruction.net/how-to/cement-fineness_o). [Accès le 31 05 2021].

- [15] T. Nowack, Écrivain, *Improvements in New & Existing Cement Grinding Mills*. [Performance]. 2014.
- [16] The Coal Handbook: Towards Cleaner Production: Coal Utilisation, 2013.
- [17] «Google Colab: The Ultimate Guide,» [En ligne]. Available: <https://ledatascientist.com/google-colab-le-guide-ultime/>. [Accès le 10 jun 2021].
- [18] «jupyter,» 10 jun 2021. [En ligne]. Available: <https://jupyter.org/>.
- [19] H. D. Paul Deitel, Python for Programmers, USA: Pearson Education, Inc., 2019.
- [20] J. Daniel, «Data Preprocessing in Machine Learning,» Medium, [En ligne]. Available: <https://medium.com/analytics-vidhya/data-preprocessing-in-machine-learning-model->. [Accès le 9 jun 2021].
- [21] W.-M. Lee, Python Machine Learning., Indianapolis: Published by John Wiley & Sons, Inc., 2019.

## **Abstract**

The cement fineness is an important factor in the quality of cement, after collecting cement samples in the laboratory to analyze this collected material by a specified equipment to obtain the fineness or sieve of the produced cement, this procedure was done every one or two hours or more than this duration in the plant with many mills.

The purpose of this thesis is the development of a soft sensor to predict the cement fineness using the machine learning model. The data was obtained from the measurable variables of the cement grinding circuit during the regular and real production of cement ball mill process.

We have us a several techniques to predict the cement fineness. The result of fineness obtained by predicting model was compared with the fineness result of laboratory tests, it give an accepted result.

The developed model need to be tested in a cement grinding system to improve and demonstrate the ability of the model

**Keywords:** Ball mill process, cement fineness, machine learning.

## **Resume**

La finesse du ciment est un facteur important dans la qualité du ciment, après avoir collecté des échantillons de ciment en laboratoire pour analyser ce matériau collecté par un équipement spécifié pour obtenir la finesse ou le tamis du ciment produit, cette procédure a été effectuée toutes les une ou deux heures ou plus que cette durée dans l'usine avec de nombreux moulins.

Le but de cette thèse est le développement d'un capteur souple pour prédire la finesse du ciment en utilisant des modèles d'apprentissage automatique. Les données ont été obtenues à partir des variables mesurables du circuit de broyage du ciment au cours de la production régulière et réelle du processus de broyage à boulets de ciment.

Nous disposons de plusieurs techniques pour prédire la finesse du ciment. Le résultat de finesse obtenu par le modèle prédictif a été comparé au résultat de finesse d'essais en laboratoire, il donne un résultat accepté.

Le modèle développé doit être testé dans un système de broyage de ciment pour améliorer et démontrer la capacité du modèle

**Mots-clés :** Procédé de broyage à boulets, finesse du ciment, apprentissage automatique.

## ملخص

تعتبر نعومة الإسمنت عاملاً مهماً في جودة الإسمنت ، بعد جمع عينات الإسمنت في المختبر لتحليل هذه المادة المجمعة بواسطة جهاز محدد للحصول على نعومة أو غربال للإسمنت المنتج ، ويتم هذا الإجراء كل ساعة أو ساعتين أو أكثر من هذه المدة حسب عدد المطاحن في المصنع.

الغرض من هذه الأطروحة هو تطوير جهاز استشعار ناعم للتنبؤ بنعومة الإسمنت باستخدام نموذج التعلم الآلي. تم الحصول على البيانات من المتغيرات القابلة للقياس لدائرة طحن الإسمنت أثناء الإنتاج المنتظم والحقيقي لعملية طاحونة الإسمنت.

لدينا عدة تقنيات للتنبؤ بنعومة الإسمنت. تمت مقارنة نتيجة الدقة التي تم الحصول عليها من خلال التنبؤ بالنموذج مع نتيجة نعومة الإسمنت المتحصل عليها في الاختبارات المعملية ، حيث أعطت نتيجة مقبولة.

يجب اختبار النموذج المطور في نظام طحن الإسمنت لتحسين وإثبات قدرة النموذج.

**الكلمات المفتاحية:** عملية مطحنة بالكرات ، نعومة الإسمنت ، التعلم الآلي.