UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO ACADÊMICO DO AGRESTE
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

FLÁVIO DE OLIVEIRA FREIRE

APPLICATION OF ASSOCIATION ANALYSIS AND NATURAL LANGUAGE PROCESSING TO IMPROVE MAINTENANCE MANAGEMENT

Caruaru
2020

FLÁVIO DE OLIVEIRA FREIRE

APPLICATION OF ASSOCIATION ANALYSIS AND NATURAL LANGUAGE PROCESSING TO IMPROVE MAINTENANCE MANAGEMENT

Master's thesis presented to UFPE to obtain a Master's degree as part of the requirements of the Graduate Program in Industrial Engineering of the Academic Center of Agreste.

Concentration area: Process Optimization and Management

Advisor: Rodrigo Sampaio Lopes

Co-advisor: Do Phuc

Caruaru

2020

Catalogação na fonte:
Bibliotecária – Maria Regina Borba - CRB/4 - 2013

F866a Freire, Flávio de Oliveira.
Application of association analysis and natural language processing to improve maintenance management. / Flávio de Oliveira Freire. – 2020.
90 f.; il.: 30 cm.

Orientador: Rodrigo Sampaio Lopes.
Coorientador: Do Phuc.
Dissertação (Mestrado) – Universidade Federal de Pernambuco, CAA, Programa de Pós-Graduação em Engenharia de Produção, 2020.
Inclui Referências.

1. Gerenciamento de recursos de informação. 2. Processamento de linguagem natural (Computação). 3. Mineração de dados (Computação). 4. Manutenção produtiva total. I. Lopes, Rodrigo Sampaio (Orientador). II. Do Phuc (Coorientador). III. Título.

CDD 658.5 (23. ed.) UFPE (CAA 2020-70)

FLÁVIO DE OLIVEIRA FREIRE

APPLICATION OF ASSOCIATION ANALYSIS AND NATURAL LANGUAGE PROCESSING TO IMPROVE MAINTENANCE MANAGEMENT

Thesis presented to the Graduate Program in Industrial Engineering at the Academic Center of Agreste of the Federal University of Pernambuco, as a partial requirement to obtain the title of Master in Production Engineering.

Approved on: 22/07/2020

EXAMINATION COMMISSION

_____

Prof. Dr. Rodrigo Sampaio Lopes (Advisor)

Federal University of Pernambuco

_____

Prof. Dr. Do Phuc (Co-advisor)

Lorraine University

_____

Prof. Dr. Marcelo Hazin Alencar (Internal Examiner)

Federal University of Pernambuco

_____

Prof. Dr. Cristiano Alexandre Virgínio Cavalcante (External Examiner)

Federal University of Pernambuco

To everyone who reads and extracts knowledge from it.

ACKNOWLEDGMENTS

In these two and a half years of my master's degree, I have shared moments and knowledge with many wonderful people – friends, classmates, and professors. Looking backward, I can see how much I have improved through these years through many challenges overcome with the support of very wonderful people. I cannot name all of them, but they know I am referring to them. To all of them, Thank you very much!

Yet, to some special ones, I want to write some words.

Many thanks to my professor advisor Rodrigo who has challenged and guided me to the very best of my skills, whom I admire and appreciate working with.

Many thanks to my family, especially my mother Roseli, my father Neto, my brothers Fernando, Filiphe, Fabricio, and my sister Maria Clara for all the support, confidence, and love that have been given me through all my life.

And very special thanks to the God I have been discovering through all these years of my life, Jesus Christ, Who has been guiding me even when I do not realize it.

“The highest reward for a person's toil is not what they get for it, but what they become by it.”
John Ruskin

“A maior recompensa para o trabalho do homem não é o que ele ganha com isso, mas o que ele se torna com isso.” **John Ruskin**

RESUMO

Com o avanço da tecnologia nos diversos setores industriais, empresas tem gerado grandes quantidades de dados a todo momento. Esses dados não apenas revelam um histórico da empresa, mas escondem padrões relevantes que, se explorados estrategicamente, podem conceder vantagens competitivas a mesma. Nesse sentido, Data Science é uma ciência que traz solução para essa e outras questões através de uma grande variedade de técnicas que não apenas limpam, estruturam e extraem informações de bases de dados, mas também auxiliam o processo de tomada de decisão. No âmbito da gestão da manutenção, o registro de falhas representa um ativo importante, mas este tem sido pouco explorado em relação aos padrões e relacionamentos de falhas existentes que pode fornecer melhorias importantes nos sistemas de gerenciamento de manutenção. A Análise de Associação é uma técnica sofisticada de Data Science usada para identificar relações de causa e efeito entre conjuntos de itens das mais diversas naturezas, como dados numéricos e textuais. Além disso, o Processamento de Linguagem Natural (PLN) é um conjunto de técnicas de Data Science que dão suporte ao processamento de dados textuais, superando todos os desafios de linguagem enfrentados ao gerenciar esse tipo de dado, e fornecem partes relevantes a serem exploradas. O processo de extração de conhecimento em bancos de dados é chamado de Knowledge Discovery in Database (KDD) e esse processo visa não apenas extrair informações relevantes dos bancos de dados, mas também apoiar os processos de tomada de decisão. Este trabalho objetiva propor e aplicar um Processo KDD, que unifique técnicas de Processamento de Linguagem Natural com a Análise de Associação para processar um banco de dados de relatórios de falhas e, a partir de seus resultados, implicar melhorias no gerenciamento de manutenção. O output do processo KDD apresentado na aplicação revelou a existência de padrões relevantes e fortes relações de causa-efeito entre o conjunto de códigos de falha e entre conjuntos de palavras apresentadas nas descrições de falha. O conhecimento obtido nesses arquivos foi conectado com melhorias consideráveis nos diferentes processos de gerenciamento de manutenção, como scheduling, designação de trabalho, compra de peças de reposição, distribuição de recursos, FMEA / FMECA / RCM, entre outros.

Palavras-chave: Gestão da Manutenção. Processamento de Linguagem Natural. Análise de Associação. Data Mining.

ABSTRACT

With the advancement of technology in various industrial sectors, companies have been generating large amounts of data at all times. These data not only reveal a company's history, but hide relevant patterns that, if strategically explored, can give the company competitive advantages. For this issue, Data Science has stood out as a science that brings effective solutions through a wide variety of techniques that not only clean, structure and extract information from databases, but also provide useful information/indicators for decision-making processes. In the maintenance management field, the company's failure report database represents an important asset, but has been little explored regarding their existing failure patterns and relationships, which may provide important improvements to the maintenance management systems. The Association Analysis is a sophisticated Data Science technique used to identify cause-and-effect relationships among itemsets of the most diverse nature, like code numbers and words. Also, Natural Language Processing is a set of Data Science techniques that support the textual data processing to overcome all the language challenges faced when managing this type of data, and provide relevant portions of it to be explored. The process of extracting knowledge from databases is called Knowledge Discovery in Database (KDD) and this process aims, not only to extract relevant information from databases, but also to support decision-making processes. This research aims to propose and apply a KDD Process, which unifies Natural Language Processing techniques with Association Analysis to process a failure report database, and out of its results, imply maintenance management improvements. The KDD Process' output in the application section revealed the existence of relevant patterns and strong cause-effect relationships among sets of failure codes and among sets of words presented in the failure descriptions. The knowledge obtained in those files was committed to relevant improvements in different maintenance management processes, like scheduling, team assignment, spare-parts replenishment, resource distribution, FMEA/FMECA/RCM, and so on.

Keywords: Maintenance Management. Natural Language Processing. Association Analysis. Data Mining.

FIGURES LIST

TABLES LIST

SUMMARY

1 INTRODUCTION

Data Science, Big Data, Data Mining, Data Analytics, and even Data-Driven are some of the current terms that have been used to refer to new knowledge fields, approaches, techniques, technologies, etc. that are somehow connected to the knowledge contained within databases. Most certainly, the growth within this area comes from the fact that, over the last years, technology has developed and become much popular throughout the world, such that its users have been generating massive amounts of data.

Notably, a considerable portion of all the generated data has been produced by companies regarding their internal and external processes using computers, sensors, mobile devices, social media, and even satellites. The information hidden in this data may be used to enhance the company's processes in so many areas, like customer relationship, production, quality, maintenance, and so on (MAHMUD et al., 2020).

It is worth mentioning that among all areas, maintenance has been considered an essential one for it provides the possibility to decrease cost, increase productivity, and even improve quality. Nonetheless, an effective and efficient maintenance management system requires, besides the experts' knowledge and evaluations, a relevant database provided with all maintenance data that one may need to improve its operations. This dataset is naturally generated by the companies' processes and may give support to predict reliability on complex systems, measure and optimize maintenance performances of components and actions (DUARTE; CUNHA; CRAVEIRO, 2013).

However, even though companies have been naturally generating vast amounts of data, a big portion of this data is considered irrelevant or useless to improve the company's processes strategically. This useless portion camouflages the relevant patterns to be perceived and used to enrich the company somehow. Moreover, often, this data is poorly labelled and structured, which disturbs, even more, its assessment process. In this case, the data must be labelled, structured, cleaned from its irrelevant part, and then be evaluated under an appropriate technique to finally provide significant pieces of information (RUIZ; CASILLAS, 2018).

The techniques used to process data also depends on the type of data that is to be processed. For instance, the tasks performed when extracting knowledge from alphanumeric data are, in some/many ways, very different from the functions performed when extracting

knowledge from image, audio and video data (DASH et al., 2019; GANDOMI; HAIDER, 2015). Similarly, there is a difference between number-data processing and text-data processing. In short, as the text-data is grounded on the language it is written, it requires an extra effort in areas like linguistics and Natural Language Processing to manipulate ad extract relevant information from this type of data (IGUAL; SEGUÍ, 2017).

In the last few years, Association Analysis techniques have been applied to identify patterns and extract knowledge into a variety of fields, like healthcare, environment, education, road-traffic, market basket, industry and so on; which makes it a flexible approach to assess different types of database. This Data Science approach identifies existing patterns within databases that describe relevant relationships between itemsets on a rule form of "*if X, then Y"*. (TELIKANI; GANDOMI; SHAHBAHRAMI, 2020). However, extracting those patterns from a raw database is not a simple and quick task; it requires set skills and steps to be followed (MAIMON; ROKACH, 2010).

This research aims to explore some Data Science (DS) techniques and apply them to a discrete failure report database to extract relevant patterns and improve maintenance management processes.

1.1 Justification

Given the notorious industrial-technological growth in which companies have been generating data from their processes, it is fundamental to apply DS techniques in industrial databases to increase the awareness of the company's status and strategically improve their operations, whether they are internal or external processes.

Besides, it has been verified the absence of applications of data mining (DM) techniques associated with NLP techniques within the maintenance management field to improve companies' processes as presented in chapter 3, *Literature Review*, of this work.

Under those circumstances, this work focuses on filling that gap, as well as providing new insights for future studies.

1.2 Objectives

1.2.1 General objective

This research aims to propose a knowledge extraction process that unifies some DS techniques used to process number and text data with the Association Analysis task, and then extract relevant knowledge from an industrial database that enables improvement on maintenance management processes.

1.2.2 Specific objectives

- Realize a theoretical framework review on DS techniques to identify useful tasks and techniques that can be used to extract relevant failure patterns;
- Realize a systematic literature review on the chosen techniques within the maintenance field;
- Build a code application with the selected DS techniques and employ it on a maintenance database;
- Relate the application's output to the maintenance management improvements.

1.3 Methodology

The methodology of this research is divided into the following steps:

1) Theoretical Review on the following topics:
   - Knowledge Discovery Database (KDD), aiming to present the structure of the knowledge extraction process;
   - Association Analysis, aiming to show its definition, algorithm, measurements, and other features;
   - Natural Language Processing (NLP), aiming to present its challenges, techniques, and other features;

2) Systematic Literature Review on Association Analysis within the maintenance management field;

3) Creation and application of a KDD Process code based on Association Analysis and NLP to process a maintenance failure report database;

4) Analysis and correlation of the KDD's output with improvements to the maintenance management department.

1.4 Contribution

The main contribution of this research is the exploration of Association Analysis with NLP techniques to process a discrete failure report database, extract knowledge from it, and to identify improvements to the maintenance management system. This process became possible through programming and application of those techniques within a KDD Process structure. It is worth mentioning that this approach had not been found in the articles presented in the *LITERATURE REVIEW* section. Figure 1 pictures the main contribution of this research.

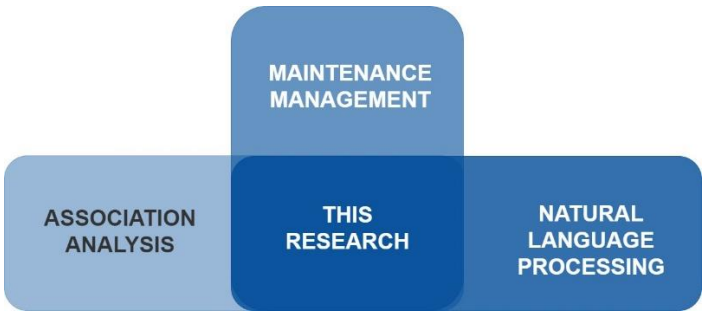


Figure 1 – Thesis contribution.
Source: The author (2020).

1.5 Structure

This research is divided into the following sections. Section 2 presents the theoretical framework review on the DS techniques used in this work. Section 3 presents a literature review on the use of those DS techniques that were somehow connected with maintenance management improvement. Section 4 presents a description of the application of those DS techniques into a failure database, and the output they generated. Section 5 presents the managerial insights obtained from the application presented in section 4. And section 6 presents the final considerations of this research considering all that had already been presented.

2 THEORETICAL FRAMEWORK

Nowadays, by the progress and advances of technology, many companies have the facility to form, gather, store, and even process large amounts of data. This data commonly contains important pieces of information for companies to improve their own business and become more competitive in the market competition field. However, extracting this knowledge may be a very hard task which demands a new 'type of science' called *Data Science* (KOTU; DESHPANDE, 2019).

In general, Data Science (DS) is a computational science used to extract meaningful knowledge from very large data sets based on four main functions: recognizing patterns, classifying items, predicting results; and making decisions (BALLARD et al., 2007). And when it comes to DS tools and techniques, there are almost infinite options from a wide variety of fields, which may be the source of the interchangeable use of the terms *Data Science*, *Machine Learning*, and *Artificial Intelligence* (AI). Nonetheless, these terms may present a slightly different meaning in some contexts, even though they are closely connected as shown by Figure 2 (KOTU; DESHPANDE, 2019).

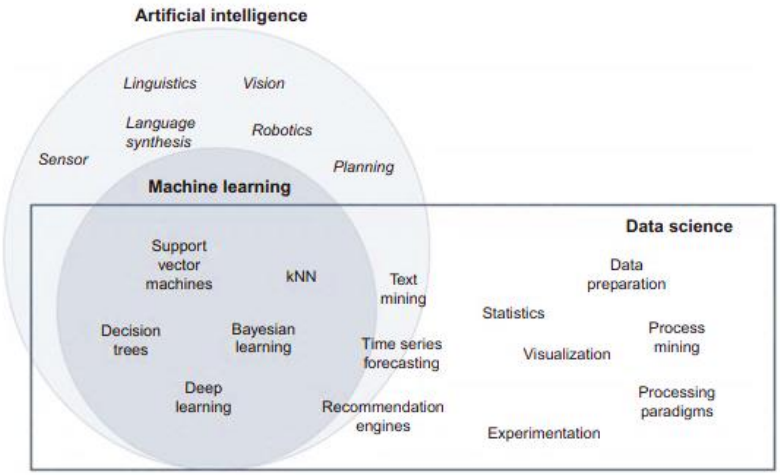


Figure 2 – Data Science, Artificial Intelligence and Machine Learning.
Source: Kotu and Deshpande (2019).

In summary, Artificial Intelligence is much more about skilling the machine to behave like a human cognitively. Not surprisingly, there are uncountable applications for AI, like

automated driving, facial recognition, items organization, etc. These applications, as well as the whole AI, are based on a considerable amount of techniques, like linguistics, decision science, robots, and Machine Learning algorithms. In this science field, learning is considered an essential skill for the machine as it will improve by receiving and analyzing data. As for Machine Learning, it can be viewed as a part of AI or an independent tool used by AI, but regardless of the taxonomy, Machine Learning is what enables the AI learning process. Within Machine Learning are algorithms that, by analyzing data, create models to work with reality and keep improving itself. And lastly, Data Science is what better emblematizes the business application of Artificial Intelligence, Machine Learning, and other tools and techniques within it. In fact, as a result, there are uncountable ways to apply Data Science in real life, for it covers a great variety of fields, like: engineering, medical, manufacturing, advertising, economics, and even literature (KOTU; DESHPANDE, 2019).

Regardless of which one we are referring to, whether it is Data Science, Artificial Intelligence or Machine Learning, the common point about them is that they all include a process of extracting knowledge from some database and apply it into their application. In this research, we are going to explore two specific techniques that are somehow identified with Machine Learning, Artificial Intelligence and Data Science, which are: *Association Analysis* and *Natural Language Processing*. But first, let's explore the aforementioned knowledge extraction process, which is a well-structured process and generally known as Knowledge Discovery in Database (KDD) Process. The next section presents it.

2.1 Knowledge Discovery in Database Process

As technology develops and improves over time, it became easier for companies to create and store large amounts of data. However, it is essential to realize that this data explicitly and implicitly contains relevant patterns that may be used to improve any process or business. Treating the databases, identifying those patterns, and presenting the knowledge through a sophisticated and intelligent computing analysis is a task of the Knowledge Discovery in Database Process. The whole goal of that process is to automate the extraction (or mining) and present it in a meaningful way (MAIMON; ROKACH, 2010).

The KDD process is a complex and well-structured process, with pre-defined steps; however, flexible, which assists companies, managers, and decision-makers in building relevant knowledge to make decisions. In detail, the KDD process is divided into four main steps and some sub-steps as now shown through Figure 3, and then described.

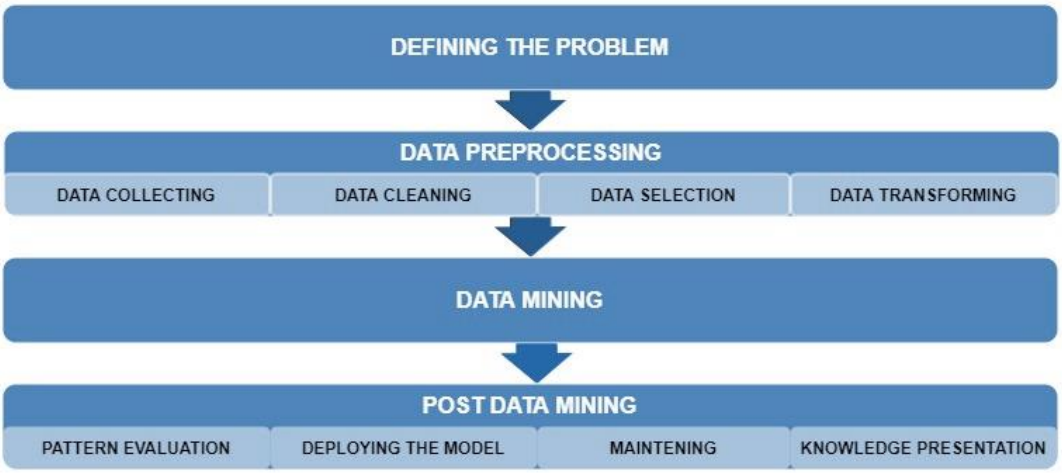


Figure 3 – Knowledge Discovery in Database process.

Source: Author (2020).

Defining the Problem. In this step, the main point is to identify the data to be used and set the goals of the KDD process. It is essential to define feasible and reasonable purposes that can benefit business management once it is achieved (ZHANG; ZHANG, 2002).

Data Preprocessing. The treatment applied to the data in this step depends on its structure, and the goal of the KDD process set previously. This step is composed of four sub-steps. In the first sub-step, *Data Collecting*, data is collected from various sources and combined into one homogeneous and encoded database, since it is more appropriate to work with just one file. In the second sub-step, *Data Cleaning*, the outliers, ambiguous and wrong recordings of the data are properly treated. Then, all recordings are converted to a desired variable format to facilitate its representation or even the next steps of KDD process, i.e., from centimeter to kilometer, from names to number codes. As a result, it has been seen that this sub-step takes a significant part of the whole effort, usually 70 percent or more of the actual *Data Mining* effort. In the third sub-step (ZHANG; ZHANG, 2002). *Data Selection*, the relevant variables, and data set are selected to be used, and the irrelevant variables and data are discarded. Finally, in the fourth

sub-step, *Data Transformation*, the selected data is converted through aggregation and summary into a format suitable for the mining step (ZHANG; ZHANG, 2002).

Data Mining. This is considered the most important step of the KDD Process, for it is this step that the main techniques and algorithms that extract the patterns and knowledge are applied. These patterns and knowledge can be obtained and through many different approaches, for instance: class description, association analysis, classification, prediction, clustering, time-series analysis, and so on (ZHANG; ZHANG, 2002).

Post Data Mining. This step is divided into four sub-steps, which include the evaluation of the patterns obtained, applications of the knowledge acquired, maintenance of the whole KDD process, and presentation of its results. In the first sub-step, *Pattern Evaluation*, the knowledge extracted is evaluated through its utility and applicability in real life. Furthermore, a sensitivity model is tested, as well as its effectiveness through a comparison between its result and the real-world result. In the second sub-step, *Deploying the Model*, the model is tested to produce results in different databases. In this sub-step, it may be required some improvement or upgrade in the model to make it work with computerized systems and new features. In the third sub-step, *Maintaining*, the model is continuously evaluated. As society changes, market, technology, and costumers also change, so the model must be outdated, but follow the new necessities trends of the market. And finally, in the fourth sub-step, *Knowledge Presentation*, appropriate techniques are selected to present the knowledge mined and evaluated from the whole KDD process. This last task shall be carefully done to better express and transfer the acquired knowledge to the users (ZHANG; ZHANG, 2002).

As it is the ‘heart’ of the whole KDD process, some authors, with no harm done, refer to ‘KDD process’ by ‘Data Mining’; however, as seen above, DM is just one step of the whole KDD process (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) The next section presents more details about the DM step.

2.1.1 The Data Mining step

As the core of the KDD process, the DM step contains the mathematical models and algorithms which are used to mine knowledge. Notably, there is a variety of forms to perform

the DM step, which depends on the tools one wants to use and the aspects of knowledge to be mined (MAIMON; ROKACH, 2010).

In general, we can divide the DM tools through categories, tasks, and techniques. The category mostly depends on the structure of the data set. The task depends on the chosen approach, and the type of information one wants to mine. And the technique may depend on the constraints, assumptions, and information structure that one may assume (ZHANG; ZHANG, 2002). Figure 3 portrays the DM hierarchy, followed by a short description of its structure.

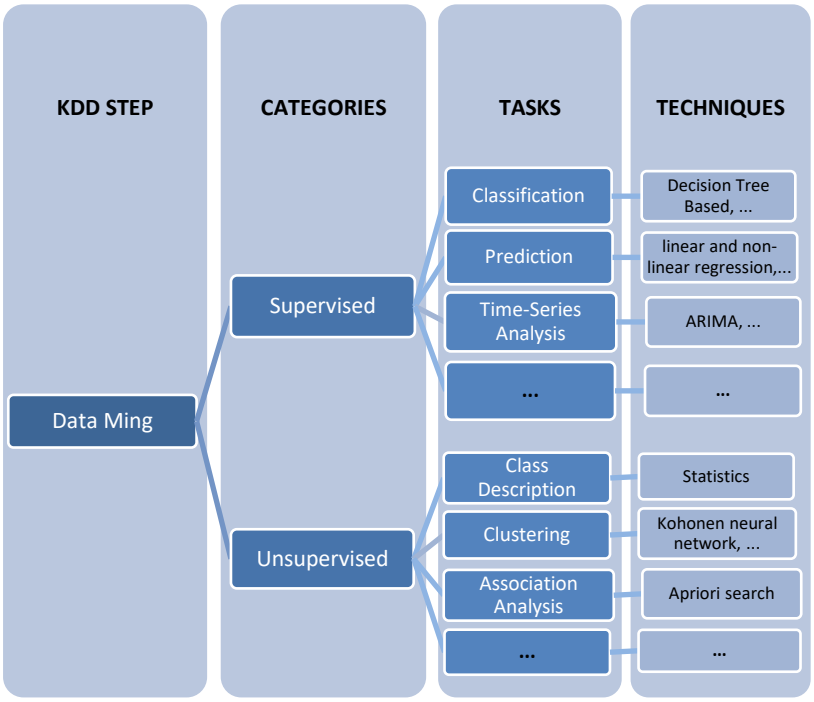


Figure 4 – Data Mining hierarchy of categories, tasks and techniques.

Source: The author (2020).

The DM categories are commonly divided into two groups: supervised (also called predictive) learning models and unsupervised (also called as descriptive) learning models. The main difference between these two categories is based on the characteristics of the dataset required for the learning (or training) phase. For instance, the supervised learning models will require a previous dataset with input and output values to construct the model, and only then, predict the values of a new dataset. In contrast, the unsupervised methods do not require a previous data set to construct its model. In these types of methods, the input data interacts with

itself to outputs a result. In other words, these methods focus on finding patterns that translate the relationship of the data points with themselves; hence, there is no need for a previous dataset. However, even though these categories are structurally different, it is possible to use both categories complementarily into one application (KOTU; DESHPANDE, 2019).

Besides, each category presents a variety of DM tasks, for example, class description, association analysis, classification, prediction, clustering; time-series analysis, and so on. Each DM task represents a specific approach to treat the dataset and extract information (KOTU; DESHPANDE, 2019). Following, it is presented a short explanation of some DM tasks together with a mention of some of its respective techniques

(1) *Classification.* It is one of the most commonly used DM tasks. At first, it is necessary to define the number of groups, the characteristic that those groups describe, and a train data set to build the train model. Then, this model will be used to assign or categorize records into groups. There are several techniques to apply *Classification* into a data set, for instance: Decision Tree-Based Classification, Naïve Bayes Based Classification, Nearest Neighbor Based Classification, Neural Networks Based Classification, and so on (ZHANG; ZHANG, 2002).

(2) *Prediction.* In general, *Prediction* also requires a train data set to build a train model. This data set must present input and output values. Later, the train model will be used to predict the values of a new data set. The predicted values may be financial, binary, or even object-ID values, depending on the situation. Notably, this task may be used in many different areas, like engineering, medical, financial, and so on. Some very known algorithms of this task are: linear and non-linear regression, genetic algorithms, and neural network algorithms (ZHANG; ZHANG, 2002).

(3) *Time-Series Analysis.* This type of DM task is suitable to extract patterns and predict future values based on historical series. In this technique, time is an important variable. Nonetheless, Time-Series Analysis may be understood through four components: trend, which represents the continuous long-term tendency; seasonality, which represents the short-term repetitive conduct of the values; cycle, which represents the long-term repetitive conduct of the values; and noise,

which is used to explain the random part of the values or the part that cannot be explained through the other components. Some very known algorithms for Time-Series Analysis are: Autoregressive Integrated Moving-Average (ARIMA), Auto Regression (AR) and Vector Auto Regression (VAR) (KOTU; DESHPANDE, 2019).

(4) *Class description*. In general, class description (also called summarization, characterization, or generalization) summarizes the data set through a statistical approach, evidencing its dispersion. Some of the variables presented through this task are: mean, variance, quartiles, and so on. In particular, Class description is useful to comprehend the global view of a data set as well as to compare it to a different data set description (ZHANG; ZHANG, 2002).

(5) *Clustering.* This type of task aims to group instances (or objects) based on its attributes (or parameters). The number of groups and the description of the groups may be previously defined, depending on the structure of the algorithm used. The objects will be grouped according to the Euclidean distance of one object to another. As a result, it will be seen that some objects will share the same group (or cluster) according to one instance; however, according to another instance, they will be grouped apart. There are many algorithms for *Clustering*, and some very known ones are: Kohonen Neural Network, Adaptative Resonance Theory, K-Means Clustering, and so on (ZHANG; ZHANG, 2002).

(6) *Association Analysis.* Association Analysis is a widely used task to discover patterns among items in a dataset. This task does not require to build or train a model. Naturally, there are many applications for *Association Analysis* worldwide, like business transactions, financial trends, marketing, and so on. Some known algorithms for *Association Analysis* are: Apriori, mining multiple-level, meta-pattern directed, etc. (ZHANG; ZHANG, 2002).

Naturally, there are other DM tasks and Techniques; however, in this research, it was decided to briefly present some of them as we are going to focus only on the use of the *Association Analysis* task. The next section presents it in detail.

2.2 The Association Analysis

The main product of Association Analysis is association rules. These rules express the relationship between itemsets in the form of $X \rightarrow Y$, which may be translated as 'if X, then Y' where X and Y are considered non-empty itemsets. These itemsets may refer to real objects, numbers, ranges, and so on (ZHANG; ZHANG, 2002).

In general, the association rules are easily understood within a shopping store context. In this context, the checkout counter records all the customers' individual shopping as transactions. Therefore, all those transactions together may hiddenly content important buying patterns that may be used to improve the company's strategy and become more competitive through some practical actions. Example of useful patterns for this context are: every time (or most of the times) when a customer buys the itemset X, he also buys the itemset Y; or even every time (or mostly) a customer buys the itemset X, he does not buy the itemset Y (HAN; KAMBER; PEI, 2012).

Following, it is presented the Association Rules (AR) description based on the definition given by Han, Kamber ad Pei (2012).

- Let $I = \{i_1, i_2, \dots, i_m\}$ be defined as a universe of possible items;
- Let $T = \{t_1, t_2, \dots, t_n\}$ be defined as the transactions database, where each transaction t_n is a non-empty itemset, such that $t_n \subseteq I$;
- Let X and Y be defined as non-empty itemsets, such that $X, Y \neq \emptyset$;
- Let the interception between the itemsets X and Y be empty, such that $X \cap Y = \emptyset$;
- Let R be defined as the association rule $R: X \rightarrow Y$, where X and Y respectively are the antecedent and the consequent itemsets, which are contained in the universe of possible items, such that $X, Y \subset I$;
- Let the rule R holds in the transactions database T with the *support* value s, such that s is defined as the percentage/probability of both itemsets X and Y together in T, $p(X \cup Y)$, which is greater than a threshold value pre-established;
- And let the rule R holds in the transactions database T with the *confidence* value c, such that c is defined as the conditional percentage/probability of X that already contains Y in T, $p(Y \mid X)$, which is greater than a threshold value pre-established.

However, not all rules are worth becoming a practical action, and they must be mathematically and reasonably assessed. For this reason, the next section presents some well-used AR measurements, and later, it is given some strategies to remove redundant/irrelevant rules.

2.2.1 The Association Rules measurements

Filtering the AR while mining them is an essential part of the mining process for not all rules are useful, so it requires some measures to evaluate it when extracting them. The literature presents an immense variety AR measurement (BENITES; SAPOZHNIKOVA, 2014; JU et al., 2015; SHEIKH; TANVEER; HAMDANI, 2004), but this section presents measures used in this research.

2.2.1.1 Support

The *Support* of an itemset X, $supp\ (X)$, represents the fraction (or probability) of that itemset appear in the transaction database T(ZHANG; ZHANG, 2002). It may be written as:

$$supp\ (X) = \frac{|X(t)|}{|T|} = p(X), \text{ where } X(t) = \{t\ in\ T | t\ contains\ X\} \quad (1)$$

However, the construction of an association rule is based on the evaluation of, at least, two different itemsets, so the support must analyze not just one, but, of all analyzed itemsets (ZHANG; ZHANG, 2002). Therefore, the support of the rule $X \rightarrow Y$ may be written as:

$$supp\ (X \rightarrow Y) = \frac{|(X \cup Y)(t)|}{|T|} = p(X \cup Y) \quad (2)$$

Even though support may seem a simple measure, in the literature, it has been constantly presented as a basic measure for filtering AR and constructing more sophisticated measures. Under those circumstances, a decision-maker must define the minimum support threshold, *min supp*, so every mined rule is already at conformity to its value**.**

2.2.1.2 Confidence

Let X and Y be respectively defined as the *antecedent* itemset and the *consequent* itemset of the rule $X \rightarrow Y$. Thus, the confidence of the rule $X \rightarrow Y$ is defined as the conditional probability of the rule $X \rightarrow Y$ based on the *antecedent* itemset (AGGARWAL, 2015). This may be written as:

$$conf\ (X \rightarrow Y) = \frac{supp\ (X \cup Y)}{supp\ (X)} = \frac{p\ (X \cup Y)}{p(X)} \tag{3}$$

It is important to realize that the $supp(X \cup Y) = supp(Y \cup X)$, however, the $conf(X \cup Y) \neq conf(Y \cup X)$ (AGGARWAL, 2015). Similarly to the *min supp*, a decision-maker may define the minimum confidence threshold, *min conf*, as mining rules.

2.2.1.3 Lift

The critic about the *confidence* measure is that, in the rule $X \rightarrow Y$, this measure does not consider $supp\ (Y)$, just the $supp\ (X \cup Y)$ and $supp\ (X)$. Consequently, rules with low values of $supp\ (Y)$ may also be considered. For this reason, it was necessary to create a new measure, *lift* (DUNHAM, 2002). The *lift* equation is now presented.

$$lift\ (X \rightarrow Y) = \frac{supp\ (X \cup Y)}{supp\ (X) \times supp\ (Y)} = \frac{p\ (X \cup Y)}{p(X) \times p(Y)} \tag{4}$$

Also called as *interest*, *lift* considers the dependence or independence of the itemsets X and Y to select reasonable rules. In other words, if $\frac{p(X \cup Y)}{p(X) \times p(y)} = 1$, the itemsets X and Y are considered *independent*, and the rule $X \rightarrow Y$ must be discarded. On the contrast, if the result of the $\frac{p(X \cup Y)}{p(X) \times p(y)} < 1$, the itemsets are considered *negatively correlated*; and complementarily, when $\frac{p(X \cup Y)}{p(X) \times p(y)} > 1$, the itemsets are considered *positively correlated* (HAN; KAMBER; PEI, 2012).

Another practical way to analyze the *lift* is through the approach presented by Zhang e Zhang (2002), here presented through Equation 5.

$$lift\ (X \rightarrow Y) = \left| \frac{p\ (X \cup Y)}{p(X) \times p(Y)} - 1 \right| \quad (5)$$

Through this approach, the decision-maker may establish a threshold *lift* value greater than 0, and just rules with *dependent itemsets*, regardless *positively* or *negatively correlated*, will be considered. Consequently, rules with independent rules will be discarded. It is important to realize that the *min lift* value represents the minimum value for itemsets to be considered *correlated* (ZHANG; ZHANG, 2002).

2.2.1.4 Conviction

The only problem with the *lift* measure is that this is a symmetric measure, so there is no difference between the *lift* of the rule $X \rightarrow Y$ and the *lift* of the rule $Y \rightarrow X$. In contrast, the *conviction* measure takes into account the support of both itemsets; however, it does not produce a symmetric measure (DUNHAM, 2002). The *conviction* of the rule $X \rightarrow Y$ is defined as:

$$conv\ (X \rightarrow Y) = \frac{supp\ (\neg\ Y)}{conf(X \rightarrow \neg\ Y)} = \frac{p\ (X) \times (1 - p(\ Y))}{(1 - p(X \cup\ Y))} \quad (6)$$

where the particle '¬' is used to refer to the complementary part of an itemset. For instance, $(\neg\ Y) = (1 - Y)$.

Consequently, the information brought by *conviction* is different from all others. *Conviction* outputs the ratio of unexpected frequency that X will occur without Y in the rule $X \rightarrow Y$. In other words, *Conviction* expresses how dependent the consequent Y is on the antecedent X to occur, and its value ranges from 0 to infinite, representing the degree of independence of Y on X in the association rule. The greater the conviction value is, the higher is the interest in the rule, and closer to zero the conviction value is, the greater the degree of independence of the rule (JU et al., 2015; KOTU; DESHPANDE, 2019)

2.2.1.5 Leverage

Another well-used measure is *leverage*. This measure ascertains the difference between the occurred and the expected frequency of X and Y to ensures its dependency (MAIMON; ROKACH, 2010). The *leverage* is described as follows:

$$lev\ (X \rightarrow Y) = supp(X \cup\ Y) -\ supp(X) \times supp(Y) \tag{7}$$

The *leverage* values are established in the interval $[-1\ ;\ 1\ ]$. A *leverage* value equals to or under to 0 indicates that the itemsets X and Y are independent, and *leverage* value close to 1 indicates dependency, which makes the rules more interesting (MAIMON; ROKACH, 2010).

These AR measurements here presented are not just to evaluate the rules, but also to mine the rules with the assistance of a mining algorithm. In this research, we have used a very known Algorithm, the *Apriori algorithm*, which is presented in the next section.

2.2.2 The Apriori algorithm

It is important to realize that the number of possible itemsets and AR depends on the number of different items d in a database, and sometimes this number may be extremely high. Equations 8 and 9 respectively describe the number of possible itemsets $M(d)$ and possible rules $R(d)$ that may be generated based on the value of d (TAN; STEINBACH; KUMAR, 2006).

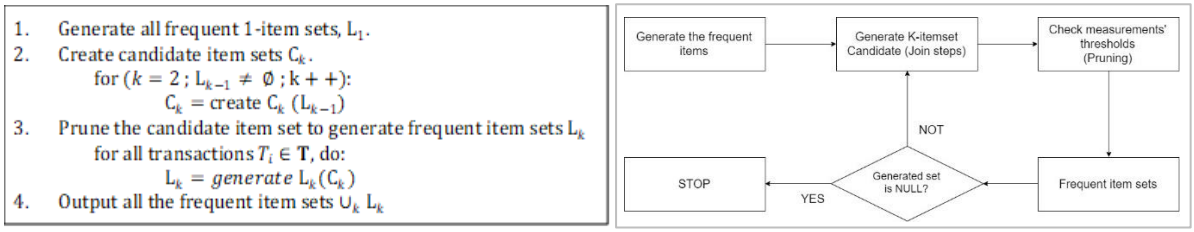
$$M(d) = 2^d \tag{8}$$

$$R(d) = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right] =\ 3^d - 2^d + 1 \tag{9}$$

To exemplify the complexity of this problem, let's suppose $d = 3, 5\ and\ 10$. The respective values of $M(d)$ are 8, 32 and 1024 possible itemsets; and the respective values of

$R(d)$ are 12, 607 and 52.002 possible rules. Naturally, most of the problems that are to be solved through *association analysis* present a considerable number of different items d; consequently, the total number of possible itemsets and rules are impracticable to analyze and compare (TAN; STEINBACH; KUMAR, 2006).

Under those circumstances, the solution to this problem goes to generating relevant rules that already consider threshold values of the measurements previously presented: support, confidence, lift, and so on. One very known and worldwide algorithm for generating rules this way is the Apriori algorithm. This algorithm generates and prunes subsets according to threshold values (of the AR measurements), then transcribes the mined subsets into AR. In general, this algorithm is divided into two main parts. In the first part, the frequent items are generated according to a support threshold value. Then, in the second part, which is divided into three cyclic steps, the frequent itemsets are generated and pruned according to AR measurement threshold values. As the length of itemsets increases in each iteration, the algorithm quits the cycle once it iterates without creating itemsets or once the length of the itemsets reaches the number of different items (HAN; KAMBER; PEI, 2012). Figure 5 presents the Apriori algorithm and the Apriori flowchart.



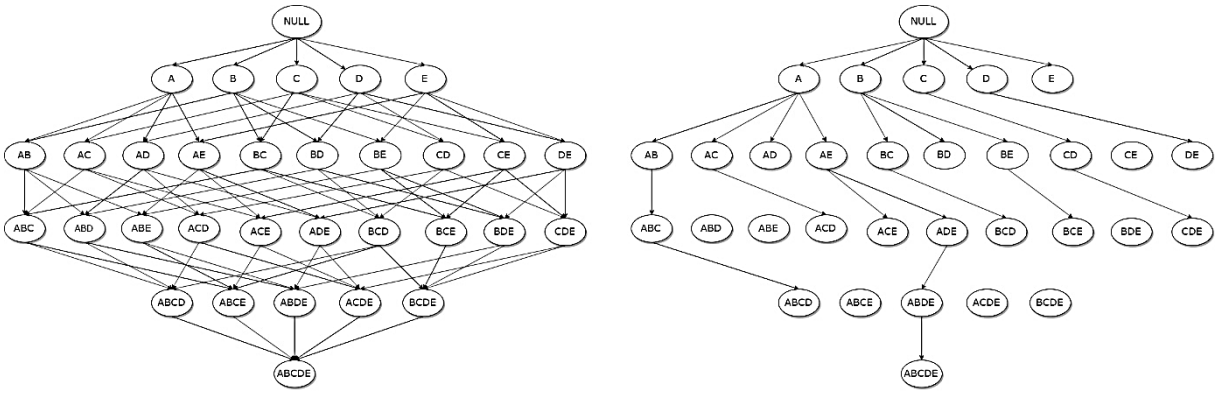
(a) (b)

Figure 5 – (a) Apriori algorithm; (b) Apriori algorithm flowchart.

Source: (a) Adapted from Liu et al. (2017); (b) Adapted from Bagui and Dhar (2019).

To better understand the efficiency and efficacy of this algorithm, let's suppose d=5. In this case, there are 300,000 possible ways to check the creation of the possible 32 itemsets; however, with the use of the Apriori algorithm, this number drastically decreases, depending on the pre-established threshold values. Granted that, Figure 6 illustrates the operation of the Apriori algorithm by contrasting two itemsets network, one created without the help of the

Apriori algorithm (a) and another with it (b). The itemsets that are connected by an arrow represent the itemsets that do not reach a theoretical threshold value.



(a) (b)

Figure 6 – (a) All possible ways of generating itemsets for association rules; (b) Itemsets generated by the Apriori algorithm, considering hypothetic values for A. R. measurements.
Source: Adapted from Tan, Steinbach and Kumar (2006)

As noticed, the Apriori algorithm is really useful for mining the AR, as the rules may be formed through several ways. However, even with the aid of the AR Measurement, many redundant or irrelevant rules may be mined. To treat this issue, a filtering process must be performed to identify and remove those. The next section encompasses this issue.

2.2.3 Filtering redundant association rules

Association Analysis is a very useful task to identify and extract patterns from databases, but not all rules mined may be considered relevant. Most of the time, a significant portion of rules mined may be regarded as redundant or irrelevant, and not rarely, this portion is even higher than the portion represented by the relevant/essential rules (ASHRAFI; TANIAR; SMITH, 2004).

Support and confidence are two regularly-used AR measurements to filter rules when mining them, but regardless of those measurement settings, a dilemma is faced. When those measurements are set to a high threshold value, some (or probably many) rules that could be mined are left behind. In contrast, when those measurements are set with a low threshold value, many irrelevant rules are mined together (ASHRAFI; TANIAR; SMITH, 2004).

Through Literature, there are uncountable ways to filter and remove redundant AR, but in the following sections, two different filtering strategies that better fit o be used in our application are presented. The first filtering strategy is called *Simple & strict redundancy* and the second is *Direction-setting redundancy.*

2.2.3.1 Simple and strict redundancy

Aggarwal and Yu (2001) present the redundancy divided into two groups *simple* redundancy and *strict redundancy*.

The authors propose that the simple redundancy is found when different rules present the same itemsets and the same support values, regardless of the confidence value. For example, let's take the rules $R1$, $R2$, and $R3$, where $R1: AB \rightarrow C$, $R2: A \rightarrow BC$ and $R3: BC \rightarrow A$. As can be seen, all three rules present the same itemset (A, B and C) and may present the same support value, which would make 1 of them relevant, and the 2 left redundant.

As for the strict redundancy, the authors depict that type of redundancy is found among rules that can be considered a subset of other rules. To demonstrate this idea, let's take the hypothetical rules $R1: A \rightarrow B$ and $R2: C \rightarrow D$, where $A \cup B = X_i$, $C \cup D = X_j$ and $X_i \supset X_j$. In this case, the rule R2 may be considered a subset the rule R1; and consequently, a redundant rule.

2.2.3.2 Direction-setting redundancy

Liu, Hu and Hsu (2000) present a strategy to identify redundancy among rules that can be "assembled" by a combination of other rules. This strategy is effective in summarizing the whole set of AR that present some redundant ones. To clarify this idea, let's take the rules R1, R2 and R3, $R1: A \rightarrow C$, $R2: B \rightarrow C$ and $R3: A, B \rightarrow C$. In this case, the rule R3 may be considered a redundant rule because the rules R1 and R2 combined already represent the connection relationship found in R3.

The next section presents some AR industry applications.

2.2.4 The Association Analysis industry applications

The literature presents a variety of applications for *association analysis*. Not to mention that each application presents a different context bounded with a different DS method or even knowledge from other science fields. This section presents some *association analysis* applications.

Calçada, Rezende e Teodoro (2019) applied *association analysis* to improve a green fertilizer production process in the Northeast of Brazil. In this application, they used the Association Analysis to identify AR and build an AR Network. Lately, this network was validated through comparison with a decision tree, and key parameters were identified as well as the best legume species to improve productivity.

Ciarapica, Bevilacqua and Antomarioni (2019) applied *association analysis* in the context of creating a framework for environmental risk management on a medium-size refinery. In their application, they used data from geographically different repositories to develop a conceptual model based on the AR. Thus, they USED Social Network Analysis (SNA) to provide a general understanding of the interactions of the risk factors, as well as identify important nodes and patterns. By their application, even though presenting a wide set of objectives and predictive variables, they were able to identify immediate and root causes, cause-effect correlations, and important areas to focus on risk management.

Wen et al., (2019) used *association analysis* to construct a traffic congestion prediction model. Briefly, they applied Association Analysis, then used a genetic algorithm (GA) to mine time series based association rules and construct the model. After that, a clustering technique was applied to generate different traffic environments in a dataset. The results proved a very accurate model.

Lakshmi and Vadivu (2017) used several AR mining algorithms to extract rules from a medical center dataset. Then, they used a multi-criteria decision analysis method to select the best algorithm. The results showed good applicability for rules to describe the relationship between diseases and medicines, diseases and symptoms, and diseases with themselves. This knowledge may be used to improve the management of the health center in many different ways.

Deshmukh and Bhosle (2016) also present a health care application stepped upon an image mining technique. In their paper, they used the association rule principles through image mining to identify patterns in a mammogram database. After filtering the database to remove wrong and unwanted recordings, they applied the Apriori algorithm database to extract the image patterns. The acquired knowledge presented useful information to identify regions of interest (ROI) in mammograms and improve the health service.

Li et al., (2019) applied *Association Analysis* to assess the air quality in 31 regions of China as the industrialization occurred in those regions from 2003 to 2017. They also registered many socioeconomic indicators and the concentration of SO_2, NO_2 and PM_{10}. Provided that, the mined rules supported the analysis to identify ranges of six prior indicators to construct industrialization strategies considering air quality levels.

The next section of this chapter presents the theoretical structure review on the subject of NLP techniques applied when managing textual data and extracting knowledge from it.

2.3 Natural Language Processing

Text Analytics (also referred as Text Mining or Knowledge Discovery in Text) is the process of treating text data and extracting relevant patterns, information and knowledge from it (ALLAHYARI et al., 2017; MAIMON; ROKACH, 2010; MINER et al., 2012).

Text data is a very unstructured type of data; therefore, when processing it, two main issues are faced. The first issue is about storing this type of data for it does not easily fit every database schema or model. The second issue is about extracting relevant information from it for this type of data frequently requires a different science field to be processed that is not required when processing numerical data, called Linguistics (SARKAR, 2016).

Linguistics is the science of Language that encompasses several different areas of it, like phonetics, phonology, syntax, semantics, morphology, lexicon, pragmatics, stylistics, semiotics and so on. However, even though Linguistics represents an enormous knowledge field, not all those areas are considered essential when processing text data. In general, two of those areas are frequently considered useful for NLP: Syntax and Semantics. In short, Syntax depicts all the uses and functions of words, phrases, clauses, sentences and grammar; while Semantics

depicts the study of meaning behind texts and words, as well as the word's base form (also called *lemma*), and so on (SARKAR, 2016).

In general, a text document has a vast number of representations and must be treated before providing any valuable information. This treatment process is often a long full of many tasks process that can be divided into three phases: Preparatory Processing, Natural Language Processing, and Problem-Dependent Tasks (MAIMON; ROKACH, 2010).

The next sections present a description of all those phases of the Knowledge Discovery in Text.

2.3.1 Preparatory Processing

Generally, the Preparatory Processing phase represents all the actions/tasks taken to convert the raw file into an editable text stream format. It is important to mention that those actions depend on the raw file format and are not always necessary. For instance, the raw files may come as a PDF file, a scanned-picture file and even a recorded speech file, so the actions taken to each of those files will be different, but all aiming to convert them into an editable stream text format, but if the raw files come as a TXT, DOCX or even XLSX file, there is no need for conversion (MAIMON; ROKACH, 2010).

2.3.2 NLP phase and tasks

In essence, NLP is a specialized science that capacitates computers to deal with and interpret human language, like English, Portuguese, French, etc. Connected by three major components – Computer Science, Artificial Intelligence and Computerized Linguistics – NLP proposes to clean, standardize and provides means to extract meaningful information from raw textual data (SARKAR, 2016). To achieve that goal, NLP possesses a vast task arsenal that enables one to perform a variety of analyses, but now we are going to explore just the ones used in this research.

2.3.2.1 Tokenization

Once the raw textual file is loaded, the computer usually recognizes the text as a stream of characters (letters, punctuation signals and spaces), not words nor sentences. This pack of characters is poorly capable of providing relevant information for Artificial Intelligence, so the commonly first NLP is converting this character format into a more meaningful one, a tokenized text format (BIRD; KLEIN; LOPER, 2009).

The Tokenization process consists of defining the tokenization technique (which depends on the goal of the text analysis) and transforming the characters into tokens (the minimal textual size component). The most common tokenization techniques are *word tokenization* and *sentence tokenization*, which will partition the text by words and sentences, respectively. In the meantime, there are other types of tokenization that one may choose to apply, for example, *clause tokenization* or *paragraph tokenization*. Moreover, it is possible to apply multiple types of tokenization at once (SARKAR, 2016).

To better understand Tokenization techniques, let's take *Table 1*, which presents two tokenization techniques on a short text taken from the book of Genesis.

TYPE OF TEXT (SIZE)	**TEXT FORMAT**
String (54)	And God said, Let there be light. And there was light.
Word-tokenized (14)	'**And**', '**God**', '**said**', '**,**', '**Let**', '**there**', '**be**', '**light**', '**.**', '**And**', '**there**', '**was**', '**light**', '**.**'
Sentence-tokenized (2)	'**And God said, Let there be light.**', '**And there was light.**'

Table 1 – Example of the sentence and word tokenization.

Source: The author (2020).

Once the computer reads the original string text, it counts the amount of 54 items in it, which are all the characters, including punctuation signals and spaces, but if word tokenization is applied to the original text, the computer will recognize each word and punctuation separately and the original text will be summarized to an amount of 14 items (74% reduction). In contrast, if the sentence tokenization is applied to the original text, the computer will summarize the original text to an amount of 2 items (96% reduction), which are divided by the period signal. The type of tokenization applied depends on the purpose of the text processing and a combination of different tokenization techniques is also possible.

Finally, after the tokenization process, the text file is enabled to go under many other tasks to be cleaned and standardized, hence a more efficient data to be analyzed and interpreted (SARKAR, 2016).

2.3.2.2 Part-of-speech tagging

Efficiently applicable after performing the word tokenization, as it isolates each word separately, the Part-of-Speech (POS) tagging is a process of identifying the syntactic function performed by each word within a sentence and tagging this information to each word. This task enables the application of several other tasks; hence, a range of different analyses, including information retrieval and syntax-sensitive analysis (BRILL, 1995).

The process of POS tagging is based on the list of syntactic classes, which can be from a very narrow list by considering only the basic syntactic classes (like nouns, verbs, adjectives and adverbs and discard unclassified words) until a very wide list by considering many detailed classes (like punctuation signals, conjunctions, determiners, symbols, sub-classes of the basic classes, etc.). The extension of the list depends on how rigorous one wants this task to be (SARKAR, 2016). Santorini (1991) presents a profound work on an extensive POS list (presented by *Table 2*) as well as an explanation of the main issues faced when performing the tagging.

N°.	**TAG**	**DESCRIPTION**	**EXAMPLE(S)**
1	CC	Coordinating Conjunction	*and, or*
2	CD	Cardinal Number	*five, one, 2*
3	DT	Determiner	*a, the*
4	EX	Existential *there*	*there were two cars*
5	FW	Foreign Word	*d'hoevre, mais*
6	IN	Preposition/ Subordinating Conjunction	*of, in, on, that*
7	JJ	Adjective	*quick, lazy*
8	JJR	Adjective, comparative	*quicker, lazier*
9	JJS	Adjective, superlative	*quickest, laziest*
10	LS	List item marker	*2)*
11	MD	Verb, modal	*could, should*
12	NN	Noun, singular or mass	*fox, dog*
13	NNS	Noun, plural	*foxes, dogs*
14	NNP	Noun, proper singular	*John, Alice*
15	NNPS	Noun, proper plural	*Vikings, Indians, Germans*
16	PDT	Predeterminer	*both the cats*

17	POS	Possessive ending	*boss's*
18	PRP	Pronoun, personal	*me, you*
19	PRP$	Pronoun, possessive	*our, my, your*
20	RB	Adverb	*naturally, extremely, hardly*
21	RBR	Adverb, comparative	*better*
22	RBS	Adverb, superlative	*best*
23	RP	Adverb, particle	*about, up*
24	SYM	Symbol	*%, $*
25	TO	Infinitival to	*how to, what to do*
26	UH	Interjection	*oh, gosh, wow*
27	VB	Verb, base form	*run, give*
28	VBD	Verb, past tense	*ran, gave*
29	VBG	Verb, gerund/ present participle	*running, giving*
30	VBN	Verb, past participle	*given*
31	VBP	Verb, non-3rd person singular present	*I think, I take*
32	VBZ	Verb, 3rd person singular present	*he thinks, he takes*
33	WDT	Wh-determiner	*which, whatever*
34	WP	Wh-pronoun, personal	*who, what*
35	WP$	Wh-pronoun, possessive	*whose*
36	WRB	Wh-adverb	*where, when*
37	NP	Noun Phrase	*the brown fox*
38	PP	Prepositional Phrase	*in between, over the dog*
39	VP	Verb Phrase	*was jumping*
40	ADJP	Adjective Phrase	*warm and snug*
41	ADVP	Adverb Phrase	*also*
42	SBAR	Subordinating Conjunction	*whether or not*
43	PRT	Particle	*up*
44	INTJ	Interjection	*hello*
45	PNP	Prepositional Noun Phrase	*over the dog, as of today*
46	-SBJ	Sentence Subject	*the fox jumped over the dog*
47	-OBJ	Sentence Object	*the fox jumped over the dog*

Table 2 – Part-of-speech tags, descriptions, and explanations.

Source: Santorini (1991).

As abovementioned, POS tagging allows a whole range of NLP tasks, and intuitively one common task would be identifying the most frequent words within each syntactic class, but this task faces the linguistic issue of word variations, which is the existence of alternative spellings to the same *root word* to satisfy grammar rules. Prefixes and suffixes primarily identify these alternatives spellings; although, there are uncountable exceptions. For example, the words *reading* and *reads*, both come from the same root word, *read*, and represent the same action, but they are written differently (by adding suffixes), for they are used for different time tenses. Another example would be the word *acknowledge*, which comes from the root word *know* by adding a prefix and a suffix. Given these points, if this issue is not fixed, the final analysis result

may be harmed and somehow inaccurate. To solve this problem, two common tasks may be performed, stemming or lemmatization, but just one of them uses the POS tagging information.

2.3.2.3 Stemming and Lemmatization

Stemming is the process of text normalization in which words are shrunk into their basic forms by cutting off their affixes (prefixes and/or suffixes). This process aims to normalize similar-related words into a singular root form, also called stem or morpheme. For example, the words *walks*, *walked*, *walking* and *walker* would all be stemmed to their root form *walk*. For some DS tasks – like clustering, classification and index searching – stemming is a very attractive task that may work as a leverage efficiency point (MINER et al., 2012; SARKAR, 2016).

The most popular stemmer is the Porter Stemmer, which uses many heuristics to shorten tokenized words into their stem form. Nonetheless, not all stemmed words remain with a meaningful spelling form. For instance, some words that end with *-y,* when stemmed, lose it and may be turned into a meaningless stem form, such as *sky* and *navy.* Another example would be when stemming some irregular verbs that present very distinct conjugated forms that cannot be turned into a singular stem form, like the verb *be,* which may be conjugated as *was*, *were*, *am*, *is*, *are* and *be*. (MINER et al., 2012).

For those reasons, *stemming* may not represent an attractive task for many text analysis applications, but those issues can be solved by applying a more sophisticated task instead, called *lemmatization*. Basically, lemmatization works exactly like stemming regarding the removing affixes part, except that lemmatization also uses the word's lexical core information (nouns, verb, adverbs, etc.) to always convert the "raw" word into its root word, which is always a meaningful one. Not to mention that *lemmatization* works well even with contractions. In other words, *lemmatization* not only solves the issues faced when *stemming*, but also yields a more trustworthy output by processing the words altogether with their syntactic information. One example of a famous lemmatizer would be the *Word Net Lemmatizer,* which is from the *nltk* module and is commonly used for Python programming (MINER et al., 2012; SARKAR, 2016).

2.3.2.4 Case conversion

As the computer recognizes the difference between lowercase and uppercase, all the occurrences of the same word must be written with the same case setting; otherwise, the computer may recognize them as different items. To illustrate this problem, let's take the sentence "*Stemming is a common task to remove affixes from words, but stemming is not as sophisticated as lemmatization*." In this sentence, there are two similar occurrences of the word "*stemming*", but as the first one has the first letter in uppercase, and the second one is all lower cases, the computer would recognize them as two different items. To solve this problem, a simple procedure must be taken, which is converting all text characters to uppercase or lowercase. In other words, the case conversion simplifies the matching of all words/tokens. (SARKAR, 2016).

2.3.2.5 Removing Stopwords

A prevalent task done when applying any text analysis is the removal of words that add little information or are considered irrelevant to the analysis, like *a, an, in, on, at, the, that, etc*. These words are called stopwords (or stop words), and usually, they cover some specific syntactic groups of words, like articles, prepositions, conjunctions, pronouns, and so on. Furthermore, there are standardized stopword lists, depending on the language in which the analysis is performed, but these lists are adaptable and can be increased or decreased as needed. In the long run, after removing stopwords, the text becomes lighter and less noisy (AGGARWAL, 2015; SARKAR, 2016).

2.3.2.6 Removing special characters

Another type of removal that is commonly applied is the removal of special characters. In this case, special characters mean symbols and/or punctuation signals that may have been used in the text. Therefore, these characters are often removed, for they add no value when retrieving information from texts for Machine Learning, Artificial Intelligence, or Data Science

applications. This removing process may be applied at different stages during the text processing, depending on the analyst's will (SARKAR, 2016).

2.3.3 Problem-Dependent Tasks

In general, Problem-Dependent phase mainly refers to the type of application in which one wants to connect the text processing and applied the processed data. Two common groups of tasks to apply that data are: Categorization and Information Extraction (MAIMON; ROKACH, 2010).

Text Categorization (TC) is a task that concentrates on grouping text documents into one or more categories according to specific features. There are two main approaches used to categorize text files, supervised and unsupervised. The supervised categorization approaches are based on a set of pre-classified documents that enables the algorithm to learn and categorize new documents. On the other hand, the unsupervised categorization approaches are based on rules that make the data interact with itself ad classify documents individually (MAIMON; ROKACH, 2010; NIKHATH; SUBRAHMANYAM; VASAVI, 2016).

Different from Information Retrieval (IR) that focuses on selecting documents by a set of criteria, Information Extraction (IE) is a task that focuses on efficiently extracting specific pieces of information from a large collection of documents and present them in a structured form. This task boosters the Text Mining process and can be used when the type of information looked for is explicit and can be readily found in the text. Its efficiency comes skillfully ignoring the irrelevant information and focusing on the relevant one, making its effort computationally less expensive The extracted information comes from exploring four basic elements: Entities, Attributes, Facts and Events (COSTANTINO et al., 1997; MAIMON; ROKACH, 2010; RILLOF; LEHNERT, 1994).

2.3.4 NLP industry applications

This section presents some NLP techniques applied to process text data in different industry applications.

Suzuki, Gemba and Aoyama (2014) present a study in which NLP techniques were used to analyze the opinion of consumers of hotel services from 8 different hotels. Data from consumer review opinions were collected between 2011 and 2012 from hotel review websites. Hotel reviews were collected from customers in 9 countries around the world (Canada, Great Britain, Ireland, Finland, Sweden, Singapore, Holland and Germany). The assessments were divided into 3 main dimensions: the building structure, the cleanliness of the hotel and the quality of the staff. This approach allowed a closer analysis of the quality perceived by customers for a better classification of the hotels and a direction in the improvement conducted by their organizations.

El-Dehaibi and MacDonald (2019) present a study for the creation and evaluation of products that aim to stand out in terms of sustainability. The approach presented by the authors analyzes, through Natural Language Processing, specific characteristics of the products, called Perceived Sustainable Features (PerSFs), divided into the three pillars of sustainability: social, environmental, and economic. The database was collected from online comments made by customers and associated the PerSFs of the product's sustainability. The results of the study support the creation and correction of sustainable products strategically.

Single, Schmidt and Denecke (2020) applied the use of NLP combined with Web Scrapping (also called Web Data Extraction) for hazard assessment in accidents with chemical compounds. The NLP techniques were used to process a textual database containing the descriptive records of accidents. At the same time, Web Scrapping was designed to obtain extra information about the chemical components on the web. Among the aspects analyzed associated with chemical accidents are: the relationships between causes and consequences, substances, and locations. The authors also endorse the relevance of the work in the industrial environment associated with the knowledge of specialists and the sharing of this information among interested agents.

Markou, Kaiser and Pereira (2019) presented in their article a structured model and application for forecasting private transport demand. The model was based on the collection of data through Web Searching techniques and the processing of it through Natural Language Processing and Time Series Analysis. The model was applied in Manhattan (NYC, USA), and the data search was directed to identify 3 specified event features: the title of the event (which can be associated with the event category), the location of the event, and the time of the event.

Based on their model, several applications can be executed concerning transportation planning carried out by public and private companies.

Mahmoudi, Docherty and Moscato (2018) present a sentiment classification model for investors in the financial market. The model shown combines NLP techniques with Deep Learning Network and domain-specific word embeddings. The authors also present the relevance of the use of emojis within this analysis and how it benefited the understanding of the scenario in general.

3 LITERATURE REVIEW

Association analysis was first introduced by Agrawal, Imielinski and Swami (1993) and, since then, this topic has spread over many knowledge fields and has been cited in uncountable articles, books, reviews, conference papers and so on. Then, it is reasonable explore this topic under limited and strategic aspects to better comprehend the context of it and get insights on how to guide the research.

In this context, Systematic Literature Review (SLR) is a structured approach to not only support on managing a significant volume of literature data and providing an overview of it, but also help one to answer some specific research questions and give insights of possible gaps or under-explored aspects of the topic under discussion (PEREIRA; COSTA, 2015; TRANFIELD; DENYER; SMART, 2003).

Considering that, this chapter presents a systematic literature review on the Association Analysis subject within the maintenance management field, aiming to provide a short overview and insights of it as well as guide this research. Figure 7 presents the steps of the Systematic Literature Review realized in this chapter.

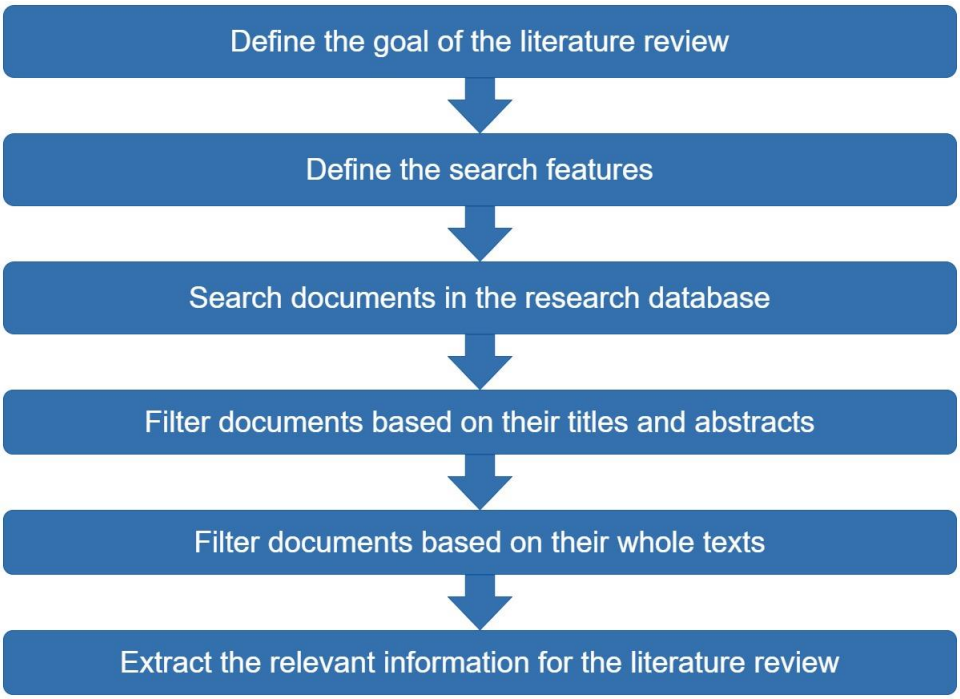


Figure 7 – Systematic Literature Review steps.

Source: The author (2020).

The goal of this literature review is to identify some general statistical aspects and answer four specific research questions. To achieve this goal, some search features and filter have been established. The search features are established to conduct the literature review on finding a reasonable number of documents aligned with the research topic, and the filters are established to discard documents that are not completely linked to the research topic. *Table 3* presents the four research questions intended to be answered in this literature review.

LITERATURE QUESTIONS
QUESTION 1) What maintenance management aspects are most connected with Association Analysis in the publications?
QUESTION 2) What types of industries are most connected with the application of Association Analysis within maintenance management in the publications?
QUESTION 3) What association rules measurements are most applied to filter rules in the publications?
QUESTION 4) What are the other techniques used along with Association Analysis in the publications?

Table 3 – Literature questions to be answered.

Source: The author (2020)

Having established the objectives of this literature review, it became necessary to define the search features to guide this study towards extracting the relevant information and achieving those goals. The features are with regards to the keywords, timespan, type of document, language and research database in which and documents search will be performed. Thus, in this literature review the documents search looked for articles written only in English using the keywords “association analysis (or “association rules” or “association rule”) and maintenance management (or maintenance) dated from 2000 until the present year in the Scopus database. The Scopus database was selected for being recognized as a trustworthy database around the world and present a large number of relevant journals on the subject of this review; and additionally, this database provided an acceptable number of articles to fulfill the objective of this literature review.

This research was carried out in the first semester of 2020 and the results achieved are now described. Table 4 presents the search features and results obtained after the filters until the extraction process.

STEP	RESULTS
Search features	- Keywords: association analysis (or association rules or association rule) and maintenance management (or maintenance); - Years: 2000 - 2020 - Document type: articles - Language: English - Database: Scopus
Database search	- Scopus search string: KEY ("ASSOCIATION ANALYSIS" AND "MAINTENANCE MANAGEMENT") OR KEY ("ASSOCIATION ANALYSIS" AND "MAINTENANCE") OR KEY ("ASSOCIATION RULES" AND "MAINTENANCE MANAGEMENT") OR KEY ("ASSOCIATION RULES" AND "MAINTENANCE") OR KEY ("ASSOCIATION RULE" AND "MAINTENANCE MANAGEMENT") OR KEY ("ASSOCIATION RULE" AND "MAINTENANCE") AND DOCTYPE (ar) AND PUBYEAR > 1999 AND (LIMIT-TO (LANGUAGE , "English")) - Number of articles found: 47
Filter 1	- Number of remaining articles: 28
Filter 2	- Number of remaining articles: 24

Table 4 – Literature Review description and results by step.

Source: The author (2020).

As we can see, after applying the search string, 47 articles were found; however, it was reasonable to admit that not all articles would be interesting for this study. Consequently, filter 1 (read the title and abstract of the articles) and filter 2 (read all articles completely) were were applied to remove the articles that were not relevant to this research. The criteria to remove or not an article when reading it was based on the principle of whether the article presents the *Association Analysis* within or strongly connected to the *Maintenance Management* context. Hence, after filter 1, 19 articles were removed for not having the maintenance management context; and after filter 2, 4 more articles were removed for the same reason.

Out of the removed articles, 20 of them used the word "maintenance" connected to the words "algorithm", "code" or "system," meaning an easy updating version of those. The other 3 articles, only used the word maintenance, but not presented the maintenance management context. Therefore, 24 articles followed to the information extraction step to identify some statistical aspects and answer the 4 literature questions, which are now presented.

3.1 Statistical aspects

This section presents a short statistical analysis of the articles that remained after the third phase of the Literature Review. This analysis covers some general aspects, the number of articles by year, articles by country, most cited article, and so on. But first, the general characteristics of those articles are presented in Table 5.

DESCRIPTION	RESULTS
Timespan	2008:2020
Documents	24
Authors	60
Sources (Journals, Books, etc)	19
Average citations per documents	13.5
Average references per documents	35.62
References	855

Table 5 – Statistical description of the Literature Review.

Source: The author (2020).

As it can be seen in Table 5, 24 articles produced by 60 different authors from 19 different sources fulfilled the requirements for this analysis. The average citation per document is equal to 13.5, yet it is better discussed later, and the average number of references per document equals 35.65 (or 855 references on total).

It is worth mentioning that, even though in the first phase, the timespan was defined to be from 2000 to 2020, the earliest article to be analyzed was published in 2008. However, it does not mean a shortening on the analysis time span, but that from the beginning of the year 2000 until the end of the year 2007, in that research database, there was no record of an article that had strongly aligned association analysis within the context of maintenance management. Similarly, there was not any register in the years 2010, 2011 and 2017. Figure 8 presents the distribution of articles that fulfilled the requirements published over the years.

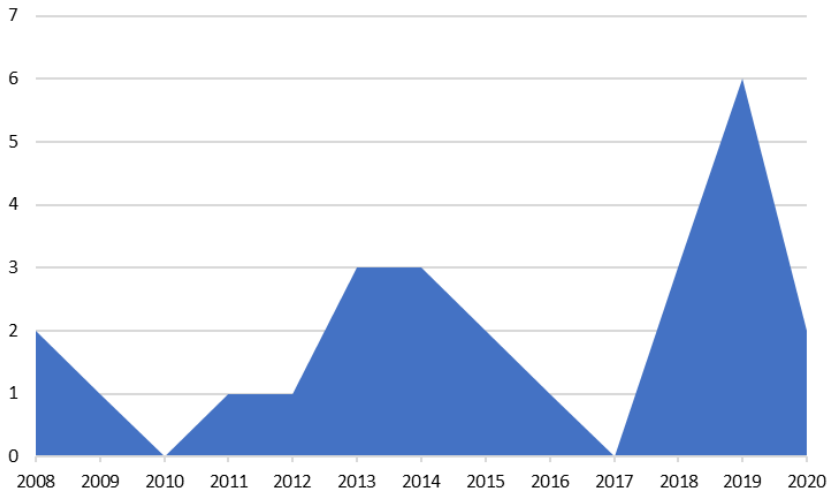


Figure 8 – Number of articles by year.

Source: The author (2020).

Another relevant information is about the origin and the type of team-work applied to those articles. As aforementioned, there were 60 authors and only 24 articles, which made an average of 2.5 authors per article; however, all those articles were located from only 11 countries as follows: China (6 articles), France (4 articles), India (3 articles), Iran (3 articles), Italy (2 articles), Spain (2 articles), USA (1 article), Korea (1 article), Indonesia (1 article), Canada (1 article) and Morocco (1 article). Figure 9 presents this world map distribution.

Interestingly, only 1 out of the 24 articles was produced from authors from different countries, Spain and Italy; all the other articles were of a single nationality. Not to mention that one-quarter of the articles came from China.

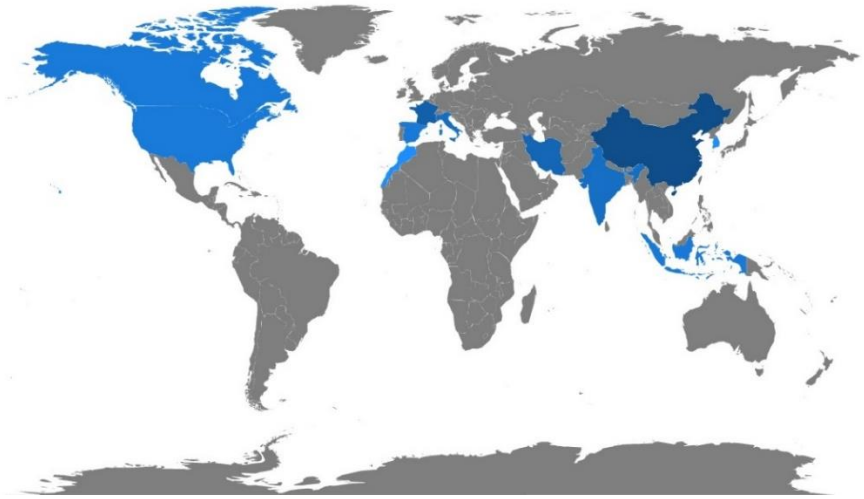


Figure 9 – Distribution of the articles on the world map.

Source: The author (2020).

Regarding the distribution of articles in their respective journals, it is concluded that there is an interesting variety, for most of the journals source only one article. One of the journal sources 3 articles, three journals source 2 articles, and the rest of the journals only source 1 article each. Figure 10 presents all the journals and their respective number of articles.

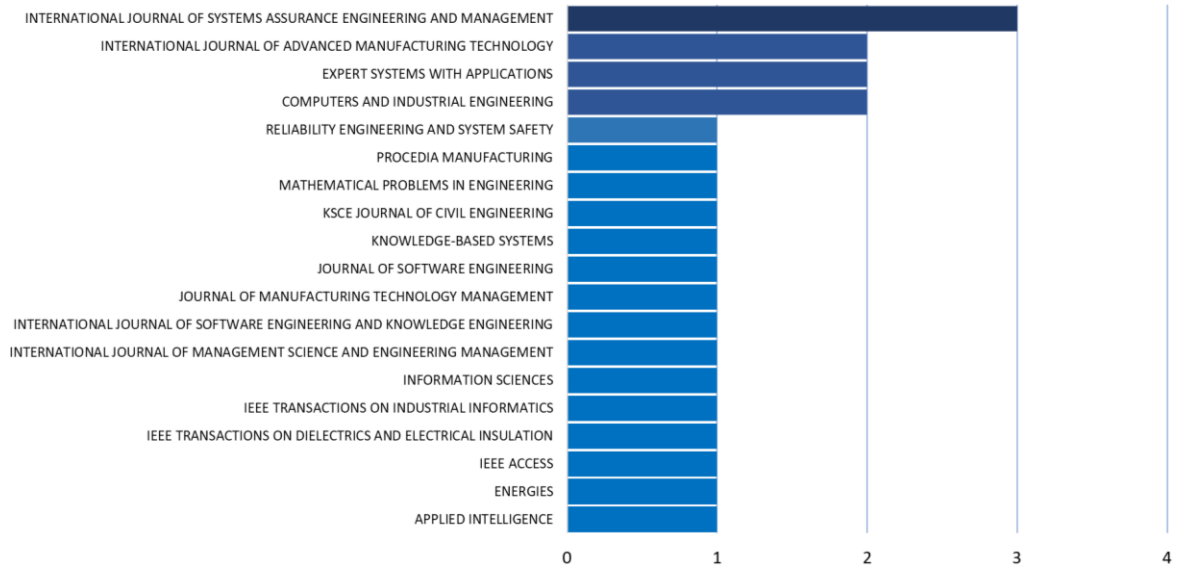


Figure 10 – Distribution of the remained articles along with the journals.

Source: The author (2020).

The number of citations of the 24 remained articles varied significantly as presented through Figure 11; nonetheless, it is reasonable considering that the newest articles tend to have fewer citations than the oldest ones. However, the most cited article and the oldest article are not the same; in fact, there is a considerable citation difference between them, even with the second most cited article and the first oldest one. It is worth mentioning that three of the articles had no citation; however, one of them was published in the current year and two of them in the year before.

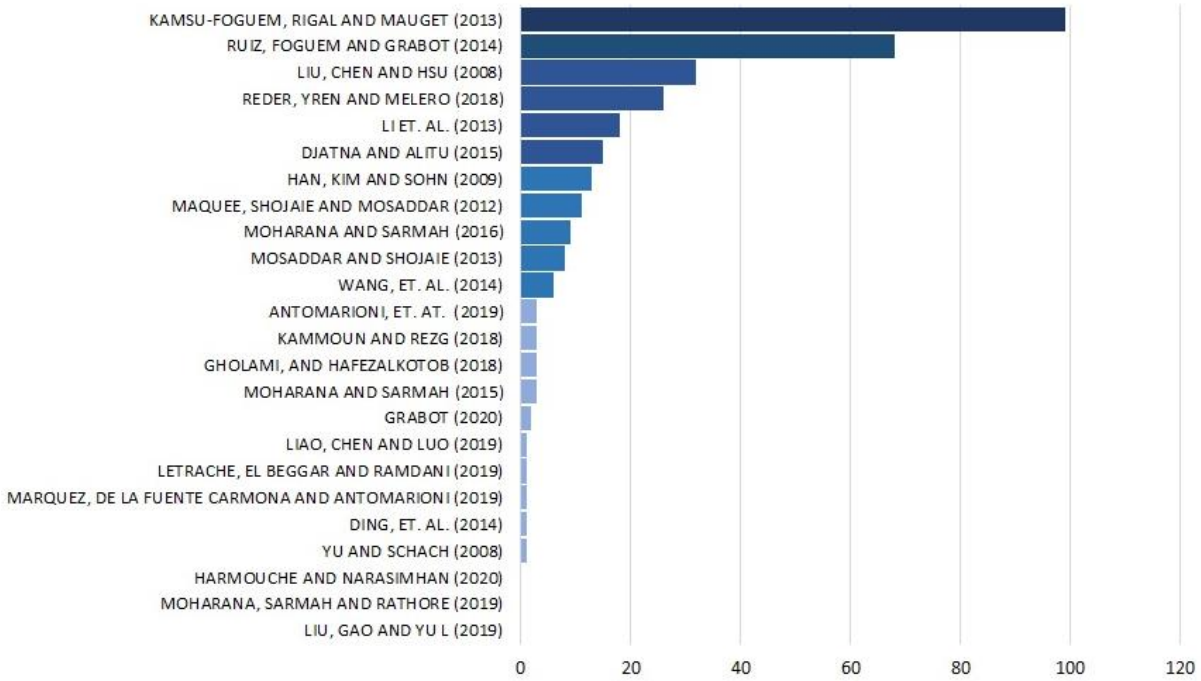


Figure 11 – Number of citations of the remained articles.

Source: The author (2020).

Finally, it is applied a Word Cloud analysis to the remained articles to verify the thematic coherence among them to the intended subjected of this Literature Review. As can be seen through Figure 9, the most frequent words of the word cloud are "*Association",* "*Mining"* and "*Maintenance"* which are strongly related to the purpose of this literature review. Notably, the "*Mining"* term comes from the "*Data Mining", "Association Rules Mining"* and *"Rules Mining",* which are in complete harmony with the subject of this research. Other relevant words also stand out in the word cloud, like: rules, rule, network, model, system, data, and so on.

The next section presents the answer to those four literature questions presented previously.

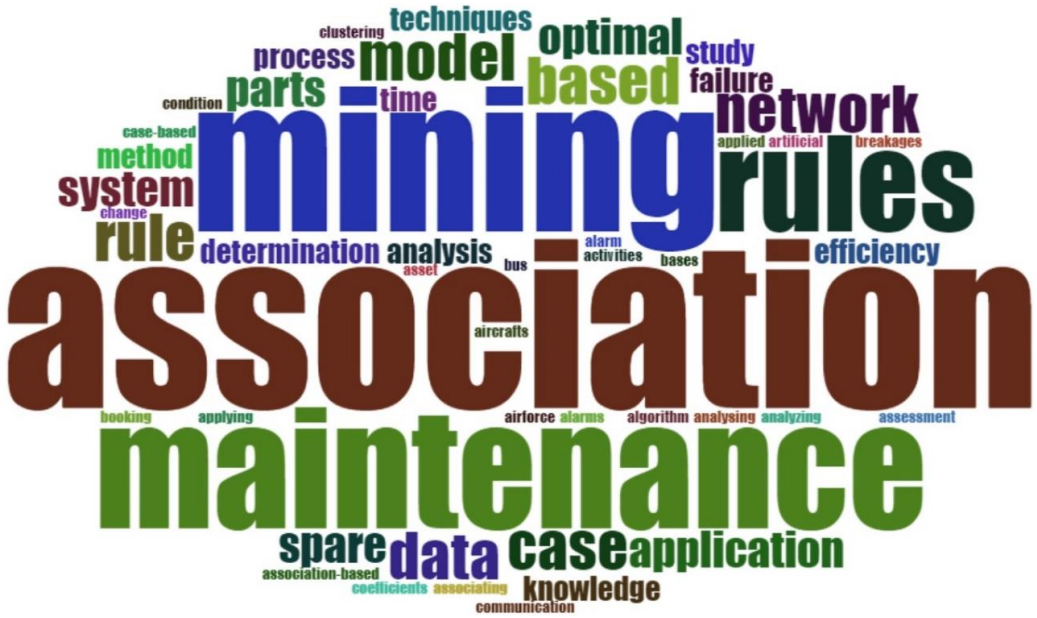


Figure 12 – Word cloud produced by the remained articles.

Source: The author (2020).

3.2 Literature questions

This section of the *Literature Review* focuses on answering the four literature questions presented before. The literature questions are concerning four specific points: the maintenance managements aspects improved through the appliance of Association Analysis; the type of industry context the publications were connected to; the variance of AR measurements found in the articles; and the variance of techniques applied together with Association Analysis to improve maintenance management. These four points are intended to provide a relative, still significant, literature overview and status of the use of Association Analysis in the maintenance management field as well as indicate possible directions for improvements to this research.

First of all, to better compare and visualize all the results of this part of the analysis, all the answers to the four literature questions from all the articles were compiled to a unique table, which is now presented in Table 6; nonetheless, the discussions about those questions with their respective graphs follows henceforward.

ARTICLE REFERENCE	QUESTION 1	QUESTION 2	QUESTION 3	QUESTION 4
Antomarioni, et. at. (2019)	Reliability, CBM	Oil refinery	Support, confidence	Mathematical model optimization
Ding, et. al. (2014)	Alarm monitoring	Flight booking	Support, confidence	Sliding time window model
Djatna and Alitu (2015)	TPM	Manufacturing	Support, confidence, lift, bond	OEE analysis, fishbone diagram
Gholami, and Hafezalkotob (2018)	Scheduling, CBM	General	Support, confidence	Time series models
Grabot (2020)	Scheduling, failure detection	Aerospace, pharmaceutics, automotive, semiconductors	Support, confidence, lift	Adapted
Han, Kim and Sohn (2009)	Failure detection	Aerospace	Support, confidence	Adapted
Harmouche and Narasimhan (2020)	Failure detection	Water distribution	Support, confidence	Clustering
Kammoun and Rezg (2018)	Scheduling	General	Support, confidence	Clustering
Kamsu-Foguem, Rigal and Mauget (2013)	Reliability	Manufacturing	Support, confidence	Adapted
Letrache, El Beggar and Ramdani (2019)	Cost	Data warehousing	Support, confidence	OLAP cube partitioning
Li et al. (2013)	Failure detection, maintenance activity	Energy generation	Support, confidence	Variable weight analysis
Liao, Chen and Luo (2019)	Safety	Technology assembling	Support, confidence	Risk analysis techniques
Liu, Chen and Hsu (2008)	CBM	Hospital	Support, confidence	Adapted
Liu, Gao and Yu L (2019)	Failure detection	Metro system	Support, confidence	Adapted
Maquee, Shojaie and Mosaddar (2012)	Reliability	Collective transport	Support, confidence	Clustering
Marquez, De La Fuente Carmona and Antomarioni (2019)	Reliability, CBM	Asset performance	Support, confidence	Artificial neural network
Moharana and Sarmah (2015)	Spare parts	Mining	Support, confidence	Adapted
Moharana and Sarmah (2016)	Spare parts	General	Support, confidence	Mathematical model optimization
Moharana, Sarmah and Rathore (2019)	Spare parts	Mining	Support, confidence, lift, χ^2	Generalized sequential patterns
Mosaddar and Shojaie (2013)	Reliability, maintenance activity	Automotive	Support, confidence, lift	Clustering
Reder, Yren and Melero (2018)	Failure detection	Energy generation	Support, confidence	Clustering
Ruiz, Foguem and Grabot (2014)	CMMS	Software	Support, confidence	Visual knowledge representation
Wang, et. al. (2014)	Alarm monitoring, reliability	Manufacturing	Support, confidence	Particle swarm optimization
Yu and Schach (2008)	Maintenance activity	Software	Support, confidence	Adapted

Table 6 – Articles' correspondences to the 4 literature questions.

Source: The author (2020)

Most of the articles directly connected the association analysis to improve the reliability and failure detection, but other maintenance aspects were also covered, like, spare parts management, maintenance scheduling, maintenance activity, alarm monitoring, TPM, safety in maintenance activities and maintenance management total cost as presented in Figure 13. Four articles covered more than one maintenance management aspect. Li et al. (2013), for example, connected association analysis with failure detection and maintenance activities; Grabot (2020) bridged the failure detection and maintenance scheduling with association analysis; Marquez, De La Fuente Carmona and Antomarioni (2019) highlighted the improvement achieved when enforcing Condition Based Maintenance (CBM) and association analysis; and Wang, et. al. (2014) connected the improvement achieved on reliability when applying association analysis to alarm monitoring.

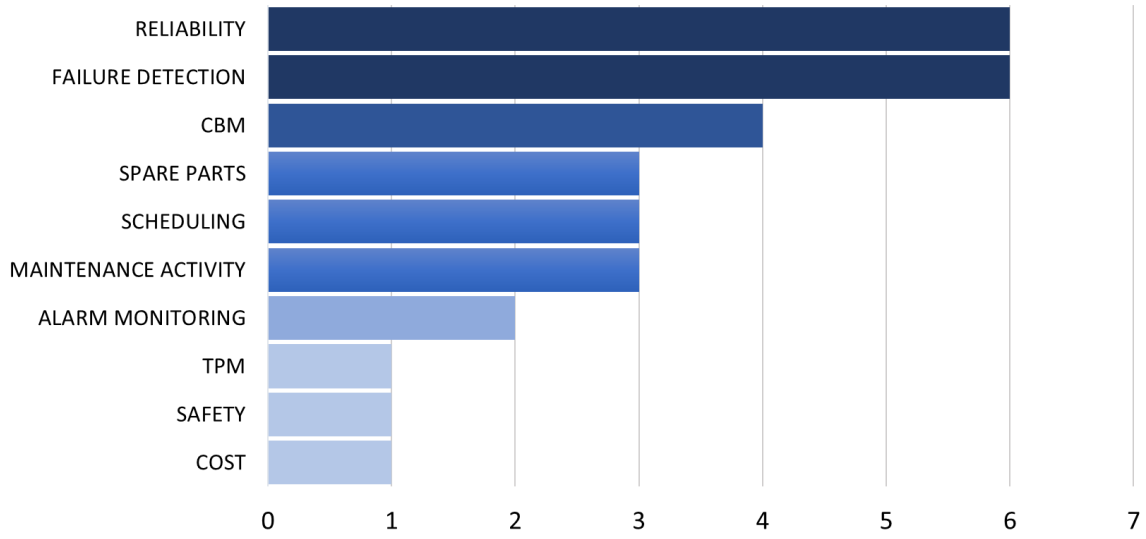


Figure 13 – Ranking of maintenance aspects in the literature review articles.

Source: The author (2020)

With regard to the types of industries presented in the 24 articles, 16 different types were identified, such as software industry, mining industry, energy generation industry, automotive industry, aerospace industry and so on. Three articles (GHOLAMI; HAFEZALKOTOB, 2018; KAMMOUN; REZG, 2018; MOHARANA; SARMAH, 2016) presented the improvement of the maintenance management through/with the support of association analysis in a general manner, not mentioning a specific industry and three articles (DJATNA; ALITU, 2015; KAMSU-FOGUEM; RIGAL; MAUGET, 2013; WANG et al.,

2014) featured the improvement achieved on the maintenance management with the help of association analysis. Grabot (2020) presented the application of association analysis to improve the maintenance management on 5 different companies from 4 different industries: aerospace (assembling-painting and maintenance), automotive, pharmaceutics, and semiconductors. Figure 14 presents the tanking of industry among those articles.

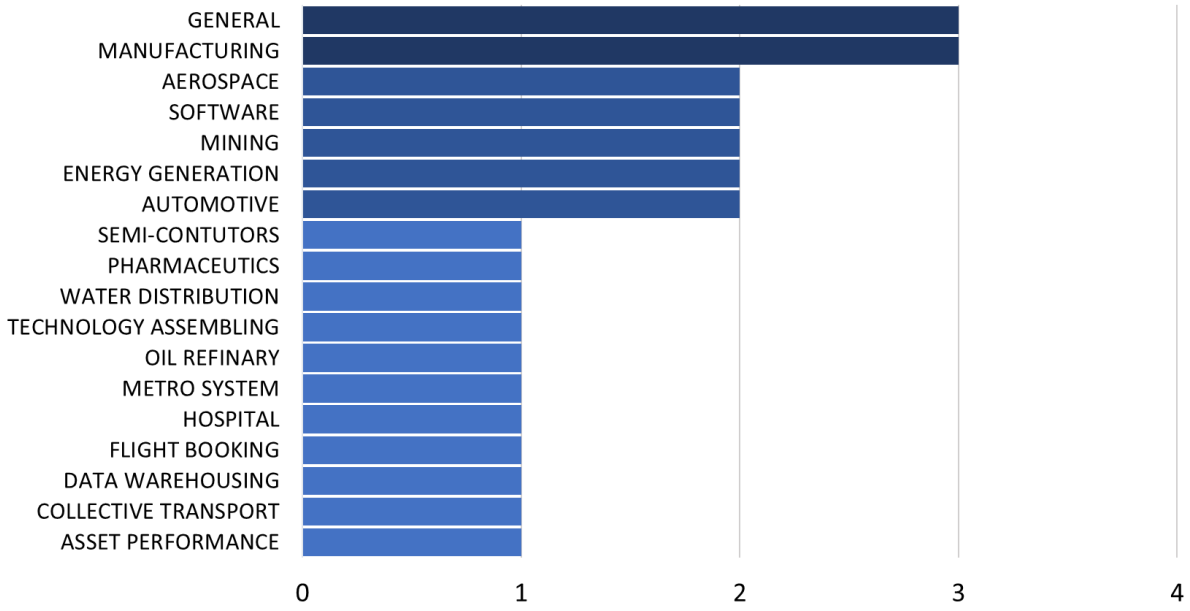


Figure 14 – Ranking of industry markets aligned with maintenance in the literature review articles.

Source: The author (2020)

As for the AR measurements applied to limit the number of rules when mining them, it was verified that in all the articles, the authors applied *support* and *confidence*, which evidences them as the most common measurements used when filtering AR. However, not only those A.R. measurements were applied in the articles; *lift*, for example, was applied in four of the articles (DJATNA; ALITU, 2015; GRABOT, 2020; MOHARANA; SARMAH; RATHORE, 2019; MOSADDAR; SHOJAIE, 2013). Also, two more measures were applied together with *support*, *confidence* and *lift*, which were the *chi-square* (MOHARANA; SARMAH; RATHORE, 2019) and *bond* (DJATNA; ALITU, 2015). Figure 15 presents the A.R. measurement set ranking found in the articles.

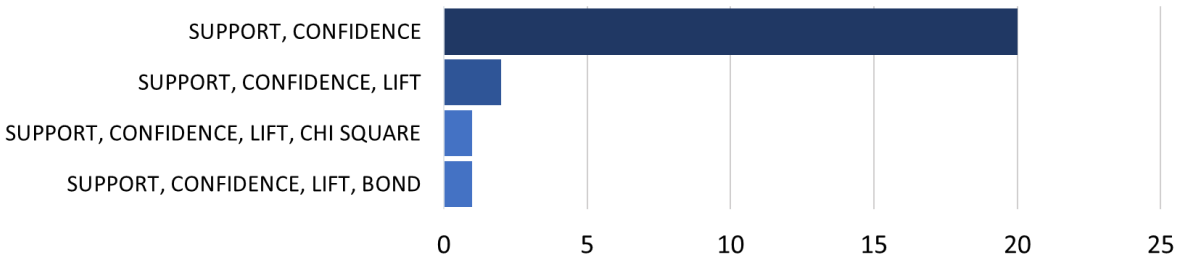


Figure 15 – Ranking of the A.R. measurement sets found in the literature review articles.

Source: The author (2020)

It was also verified that in many articles the maintenance management improvement was operated through the appliance of association analysis aligned with other techniques, like: clustering (MAQUEE; SHOJAIE; MOSADDAR, 2012; MOSADDAR; SHOJAIE, 2013; REDER; YÜRÜŞEN; MELERO, 2018), mathematical model optimization (ANTOMARIONI et al., 2019; MOHARANA; SARMAH, 2016), time series models (GHOLAMI; HAFEZALKOTOB, 2018), OEE analysis (DJATNA; ALITU, 2015), artificial network (MÁRQUEZ; DE LA FUENTE CARMONA; ANTOMARIONI, 2019) and so on. Not surprisingly, in some articles Association Analysis was applied as the main and only technique; however, it adapted to the article's specific context. Figure 16 pictures the ranking of techniques combined with Association Analysis found in the articles.

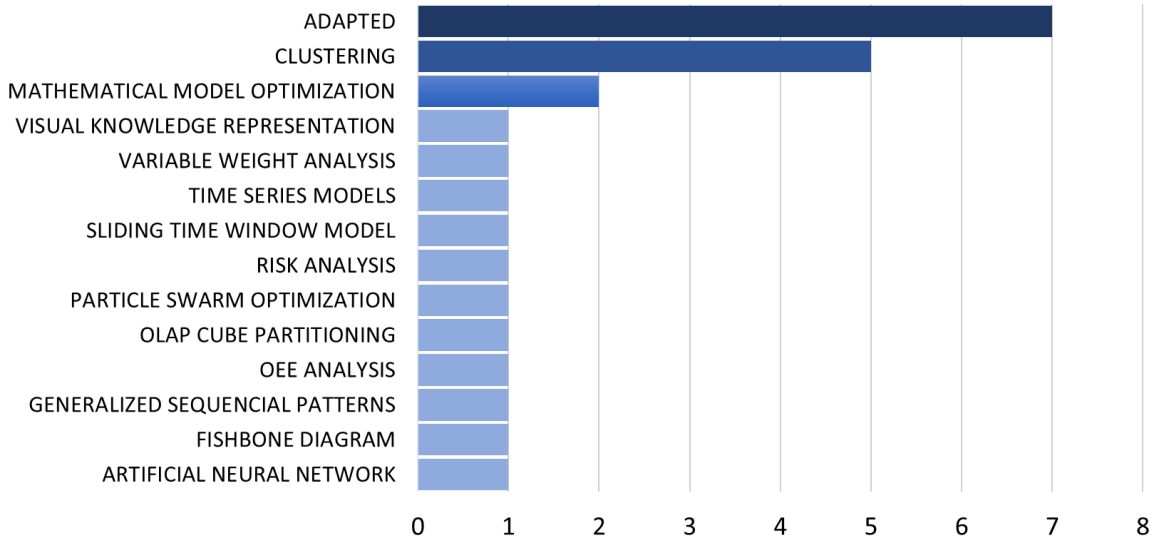


Figure 16 – Ranking techniques aligned with Association Analysis in the literature review articles.

Source: The author (2020)

In the final analysis, it is verified that there is a universe of possibilities to improve maintenance management through association analysis, regardless of the type of industry, the additional technique(s) it is to be combined with, and adaptations one may do. In the next section, it is presented an application of Association Analysis with NLP techniques through a structured KDD process to provide improvements to the maintenance management department of a manufacturing company.

4 PROPOSED KDD PROCESS AND INDUSTRIAL APPLICATION

This section presents an application of association analysis in the case of an industrial company. To extract the hidden patterns from that failure report database, a KDD process was performed. Figure 17 portrays KDD Process flow char, which is divided into 4 phases: Defining the Problem, Data Pre-Processing, Data Mining, and Post Data Mining. All the computational programming was handled through the Python software and the only input file used was an excel failure report file. In contrast, it outputted some excel files, graphs and diagrams to be used to improve maintenance management. All the KDD process is now described.

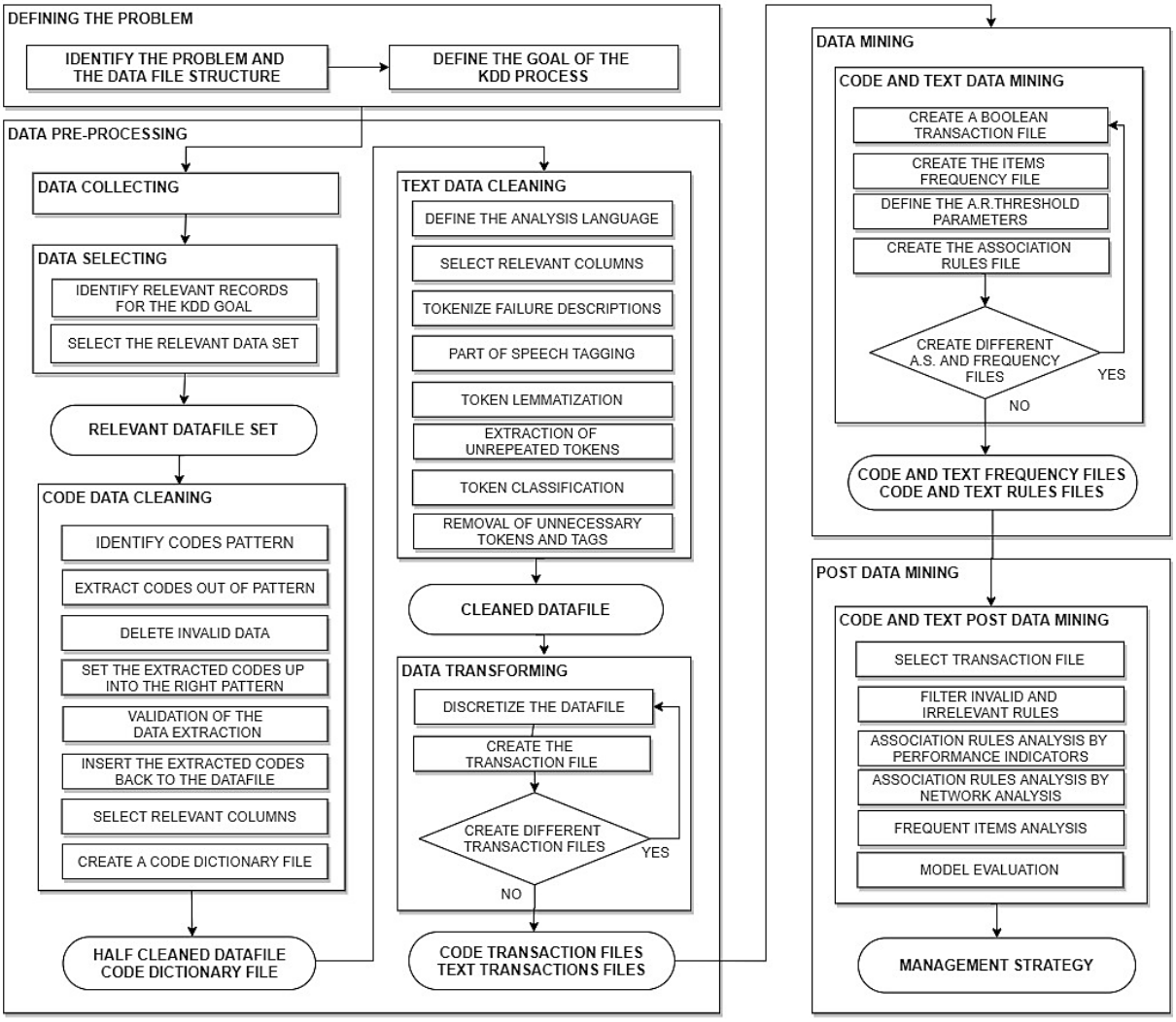


Figure 17 – Flow chart of the KDD process performed.

Source: The author (2020).

DEFINING THE PROBLEM

First of all, to efficaciously apply a KDD process, it is necessary to evaluate the problem in which the datafile comes from; then, rigorously define the goals for the KDD process based on the information available in the datafile.

Our dilemma consisted of a register of continuous failures from a specific production line, so the goal of the KDD process was to identify the main causes of those continuous failures, as well as the occurrence of strong cause-effect relationships among failures.

Then, to extract all the information needed, the *Association Analysis* was selected as the main technique to be used in this KDD process. Additionally, as part of the registration logs was expected to be recorded as failure codes and part as failure descriptions, the *Association Analysis* was of double-appliance. In other words, the *Association analysis* technique was to be separately applied to both code data and text data. Consequently, all the next phases include steps taken to separate and treat all data.

Following, the *Data Pre-processing* phase is explored.

DATA PRE PROCESSING

At first, it is important to understand that this *Data Pre Processing* phase is divided into five sub-phases: *Data Collecting, Data Selecting, Code Data Cleaning, Text Data Cleaning,* and *Data Transforming.* In *Data Collecting,* it will be explained all the sources as well as the structure of the database. In *Data Selecting,* it will be pointed out what part of the original datafile is to be used through all the KDD process and the reduction it represents. The *Code Data Cleaning* details all performed actions with regards to the registration of failures, that covered the removal of the unnecessary logs and features that remained in the data file. After that, the *Text Data Cleaning* presents all the steps taken to clean the text part of the failure description, leaving only relevant tokens for the later phases. Then, in the *Data Transforming phase,* it will present the last fixing structure steps before sending the files to the *DM* phase.

Now, *Data Collecting sub*-phase is presented.

Data Collecting

Our database consisted of a great number of failure data logs recorded by PLCs (Programmable Logic Controllers) and grouped into one single file. This failure report database that was collected from a manufacturing machine from January 7th to December 7th of 2015, and transcribed into a 23 MB Excel file with 500,000 rows, 12 columns and 151 possible types of failures. It is important to realize that this database represents an outstanding amount of 2.85×10^{45} possible itemsets and 1.09×10^{72} possible rules. No human approaches or even simple computational approaches are suitable to analyze this database and produce an equivalent relevant result. In contrast, DS techniques are considered promissory approaches.

Following the *Data Selecting* phase is presented.

Data Selecting

In this subphase, the only goal was to identify the relevant datafile portion to be employed in the KDD process, select it and discard the irrelevant part. This task only demanded a visual analysis of the datafile's structure and a creation of extracting code to treat the relevant part.

After these steps, the original database fell from 500,000 rows and 12 columns to a new database of 500,000 rows and 4 columns. Even though these past steps were simply applied, they represent a reduction of around 67 percent in the whole data set, which represented a considerable decay.

This structure, left in the database guided the construction of the algorithms through the next data pre-processing sub-phases.

Code Data Cleaning

The first step in the *Code Data Cleaning* phase was to treat the failure codes that were out of the general database pattern, which by computer's aid identified only the codes [6413] and [6424]. Later, it is shown the relevance of this step as one of these codes presents a considerable impact on the AR. Additionally, it was identified that some rows were filled with invalid information; inevitably, they must be removed in order not to disturb the following

steps. Table 7 illustrates the general pattern of the database structure, evidencing the codes out of the general pattern and the rows with invalid information.

CODES	**…**	**DESCRIPTION**	**…**	**TIME**	**…**	**PLC**	**...**
700454	…	description	..	time	…	plc	...
6413	…	…	description	…	time	..	plc
++							
100	…	description	…	time	…	plc	...
10610	…	description	…	time	…	plc	...
6413	…	…	description	…	time	...	plc
6424	…	…	description	…	time	...	plc

Table 7 – General appearance of the database evidencing codes out of the general pattern.

Source: The author (2020).

Subsequently, the rows with codes in different patterns and invalid information were taken out from the original file and transcribed into a new file. In this new file, the rows with invalid information were removed; then, the structure of this new datafile was set up according to the original datafile structure. Then, as both datafiles (original and new) had the same structure, the new file was inserted back to the original one; yet, before adding it back, a validation step was applied.

This validation step verified if only the unnecessary data was removed from the original database through a comparison of the amount of data removed with the new datafile and the original datafile. After positively verifying that, the new datafile returned to the original one and the data logs were chronologically ordained, so the failures could be seen over time. As a result of the last steps, all the data logs in the database fit the same structure pattern, and 3,334 rows and one more column with invalid or irrelevant information removed from the database.

At this point, the original database was clean from bugs, and wrongs settings, and irrelevant information. Under this circumstance, this datafile is suitable to be used to be better explored. Thus, a Code Dictionary file was created for future interests. This new file held only one register of each failure code and its description to connect those two pieces of information. In other words, an actual failure code dictionary.

At this point in the *Pre-Processing* phase, after ending the *Code Data Cleaning* sub-phase, two relevant files were produced: the *half cleaned Database* file and the C*ode Dictionary*

file. The *Code dictionary* file is intended to support the KDD Process in identifying and connecting failure codes with failure descriptions and *half-cleaned Database* file holds all the data logs which contain the failure codes, time occurrence, and failure descriptions. Nonetheless, the failure description is useful to understand the failure behind each code, it is also true that the failure description sequence and the items within the sequence may also carry relevant error patterns, perhaps more useful than the one from the code data. However, extracting relevant patterns from a textual data is not an easy task, it requires a more complex pre-processing process, yet worth it.

Given these points, the next section explores the text cleaning process in the failure description part.

Text Data Cleaning

In this part of the *Pre-Processing* phase, all actions related to text cleaning were to be taken. It is worth remembering that all failures include their respective codes descriptions, and those descriptions contain essential keywords to describe them. Identifying those keywords (which may refer to mechanical parts, processes, resources, machine estate, etc.) may provide a holistic perspective of the failure condition of the company. Therefore, the main goal of this text cleaning is to remove the unnecessary words from the description and identify the keywords to serve as input for the *DM* phase.

Notably, the first step of the *Text Cleaning* sub-phase was to define English as the base language for the text cleaning sub-steps. This step, although simple, is relevant for the NLP techniques performed in the next steps.

In general, after having defined the base language for the text cleaning, the next steps were: selecting the relevant column and tokenizing the failure descriptions. In detail, the '*Description'* column was the only relevant column, for this textual analysis. Then, a word tokenization process was performed to convert the failure descriptions from string format into ngram format and removes all the space characters. This new format allows the algorithm to recognize words (henceforward also called "tokens") and punctuation signals as individual objects, not characters, which enables the next steps concerning text processing. Table 8 presents the result produced after the tokenization step in one of the failure descriptions.

TEXT CLEANING STEP (number of objects)	FAILURE DESCRIPTION OUTPUT
Original description (29)	“THE CHANNEL CANNOT MOVE TOOL.”
Tokenized description (6)	“THE”, “CHANNEL”, “CANNOT”, “MOVE”, “TOOL”, “.”

Table 8 – Example of a failure description after tokenization.

Source: The author (2020).

Up to this point, it is already possible to verify an improvement in the failure description length, which came as a characters-size description and leaves as a tokens-size description. However, despite this improvement, there were too many tokens that only played unnecessary linguistic functions and were not essential to identify the core message of the original failure description. In other words, without those tokens, it is still possible to understand the main message of the failure description.

To demonstrate this idea, last take the previous failure description presented. Initially, that failure description was [“THE”, “CHANNEL”, “CANNOT”, “MOVE”, “TOOL”, “.”]. It is reasonable to affirm that not all those tokens are essential to understand the main idea behind this failure description. To verify that, let’s take the tokens “THE”, “CANNOT” and “.” out from that ngram. As a result, the following ngram is outputted [“CHANNEL”, “MOVE”, “TOOL”]. With this ngram, it is easily reasonable to understand that the failure occurred to the machine parts “CHANNEL” and “TOOL” during the process “MOVE”. The linguistic function (part of Speech – POS) played by the tokens “CHANNEL”, “MOVE” and “TOOL”– which respectively were “NOUN”, “VERB” and “NOUN” – also reveals their importance to decide if the token is essential or not. Similar to the failure description presented, all other failure descriptions contain unnecessary tokens that may be removed from the failure description without causing any harm to their prior message.

Another linguistic problem is the word form variations. In this problem, a word may present itself through different forms, but having the same meaning. In this circumstance, the algorithm counts each word form as a different item, though referring to the same issue. For that purpose, a Lemmatization process must be performed so the datafile may proceed to the next steps with only the base form of each word, when possible. To achieve that accomplishment, a Part of Speech Tagging algorithm must be applied first. This algorithm takes the ngram failure descriptions, analyses the POS function of each token in the sentence, and

assigns a POS tag to each one of them. These tags provide a piece of essential information to the Lemmatization algorithm to shrink the words efficiently.

For instance, let's take the description Lemmatization of the codes [520221], [520229] and [702033] presented in Table 9. In that case, the word "tight" is presented in three different word forms (continuous, base and simple past); even though, in all those forms, it is reasonable to assume that the same issue is expressed in those words, which is a failure described through the tightening process, whether causing the failure or being the failure itself. Then to apply Lemmatization, the algorithm took the ngram descriptions first and assigns a POS tag to every token according to its POS function in the failure description, creating tuple pairs. Hence, with the information brought by each tuple pair, the algorithm lemmatizes the token, when possible, and leaves the failure description with base form words.

CODE	**DESCRIPTION**
520221	['CHECKING', 'THE', '**TIGHTENING**', 'COURSE', 'OF', 'FS1', 'CUTTER', 'REQUIRED', '. ']
	[('CHECKING', 'VBG'), ('THE', 'DT'), **('TIGHTENING', 'VBG')**, ('COURSE', 'NN'), ('OF', 'IN'), ('FS1', 'NNP'), ('CUTTER', 'NN'), ('REQUIRED', 'VBN'), ('.', '.')]
	['CHECK', 'THE', '**TIGHT**', 'COURSE', 'OF', 'FS1', 'CUTTER', 'REQUIRE', '. ']
520229	['CUTTING', 'AXIS', 'DOES', 'NOT', '**TIGHT**', '. ']
	[('CUTTING', 'VBG'), ('AXIS', 'NN'), ('DOES', 'MD'), ('NOT', 'RB'), **('TIGHT', 'VB')**, ('.', '.')]
	['CUT', 'AXIS', 'DO', 'NOT', '**TIGHT**', '. ']
702033	['CROSS-DRILL', 'BAR', 'ADAPTER', '1', 'SUPPORT', '; ', 'ADAPTER', 'INCORRECTLY', '**TIGHTED**', '. ']
	[('CROSS-DRILL', 'NNP'), ('BAR', 'NN'), ('ADAPTER', 'NN'), ('1', 'CD'), ('SUPPORT', 'NN'), (':', ':'), ('ADAPTER', 'NN'), ('INCORRECTLY', 'RB'), **('TIGHTENED', 'VBD')**, ('.', '.')]
	['CROSS-DRILL', 'BAR', 'ADAPTER', '1', 'SUPPORT', '; ', 'ADAPTER', 'INCORRECT', '**TIGHT**', '. ']

Table 9 – Three failure description examples for Lemmatization.

Source: The author (2020).

The algorithm applied presents a rigorous tagging process with 36 possible POS tags; however, for punctuation signals and special characters tokens, the algorithm assigns the same item as POS tags since they play a different linguistic role in a sentence. However, this issue did not affect this application as punctuactions are to be removed from the analysis. Table 10 presents all 36 POS functions with their respective sigla and meaning.

SIGLA	MEANING	SIGLA	MEANING
CC	Coordinating conjunction	PRP$	Possessive pronoun
CD	Cardinal number	RB	Adverb
DT	Determiner	RBR	Adverb, comparative
EX	Existential there	RBS	Adverb, superlative
FW	Foreign word	RP	Particle
IN	Preposition or subordinating conjunction	SYM	Symbol
JJ	Adjective	TO	To
JJR	Adjective, comparative	UH	Interjection
JJS	Adjective, superlative	VB	Verb, base form
LS	List item marker	VBD	Verb, past tense
MD	Modal	VBG	Verb, gerund or present participle
NN	Noun, singular or mass	VBN	Verb, past participle
NNS	Noun, plural	VBP	Verb, non-3rd person singular present
NNP	Proper noun, singular	VBZ	Verb, 3rd person singular present
NNPS	Proper noun, plural	WDT	Wh-determiner
PDT	Predeterminer	WP	Wh-pronoun
POS	Possessive ending	WP$	Possessive wh-pronoun
PRP	Personal pronoun	WRB	Wh-adverb

Table 10 – Part of Speech Tags (POS TAGs) scheme applied in this KDD process.

Source: The author (2020).

At this point in the text cleaning, the failure description contains only of tuples that carry pairs of relevant items: lemmatized tokens and their respective POS tags. This couple of features displays the relevance of each word (or token) in the failure description and may perfectly support the decision-making process of removing unnecessary tokens. In detail, the tuple pairs were classified according to the maintenance role identified in the failure description. The maintenance classes were set according to 5 different industrial aspects: *process*, *machine part*, *machine state*, *process resource*, and neutral (not relevant). As a result, 699 unrepeated tuple pairs were assigned into the 5 different maintenance classes: machine part (15,02%), process (12,59%), state (4,15%), resource (2,72%) and neutral (65,52%).

The *Machine part* class carries tokens that refer to the mechanical pieces involved in the failure, like axis, channel, brooch, etc. The *Process* class carries tokens that refer to the activity which the machine was executing when the failure happened, like moving, sliding, tightening, etc. The *Resource class* carries tokens that refer to resources used in the process, like water,

gas, oil, etc. The *State* class carries tokens that relate to the machine status that may have caused the failure or was caused by the failure, like stopped, clogged, closed, etc. And the *Neutral* class carries tokens that are hardly related to maintenance or failure aspects (like interior, manual, etc.) and tokens that only perform a linguistic function (like while, already, and, etc.).This last class was intended to keep the reference of all the tokens that should be removed from the failure description file.

Then, all "neutral" tuple pairs and left in the datafile are removed from it, leaving the datafile with only useful (and lemmatized) tokens. After removing all the remaining POS tags information, the datafile is completely cleaned and ready to be transformed into the right format in the next sub-phase: *Data Transforming*.

Data Transforming

In this sub-phase, the datafile is to be transformed to an appropriate structure format to be used in the *DM* phase; however, as the failure codes were recorded by PLCs, one final issue must be treated: one failure may get registered multiple times, and yet be the same failure occurrence. In this case, a discretization process based on a time interval is needed to remove all the repeated logs and leave only the first occurrence of each failure code within each discretization time (Δt). This process required a time reference to define how long a failure code may keep being registered and still be the same failure occurs, so that every repeated failure within this time span is to be removed from the datafile.

To evaluate the effects of different values of the discretization time (Δt), in this KDD process, it was set up 6 different periods for the discretization time, $\Delta t_d = \{10\ min, 20\ min, 30\ min, 40\ min, 50\ min, 60\ min\}$, starting from 10 minutes until 60 minutes. All the 6 files produced in this discretization process proceed in the KDD process to be further explored in the next steps. Figure 18 illustrates the discretization process applied in this KDD process

Finally, the last step of the *Data Transforming* phase was the transformation step, which converts the database from the "one-failure-by-row" format into the "one-transaction-by-row" format. This new format is suitable for the Apriori algorithm posteriorly used to mine the AR. However, before creating the Transaction file, it is necessary to define a "transaction time".

This new period is used to group the failure data logs into the transactions. For that purpose, we intended to evaluate this decision by grouping the data logs by 6 multiples of 10 minutes, similarly as performed in the discretization process, $\Delta t_t = \{10\ min, 20\ min, 30\ min, 40\ min, 50\ min, 60\ min\}$, creating 36 different files. Table 11 illustrates the conversion of the datafile structure into the new format, the Transaction format. TID refers to the Transaction ID number, which represents the set of failures gathered.

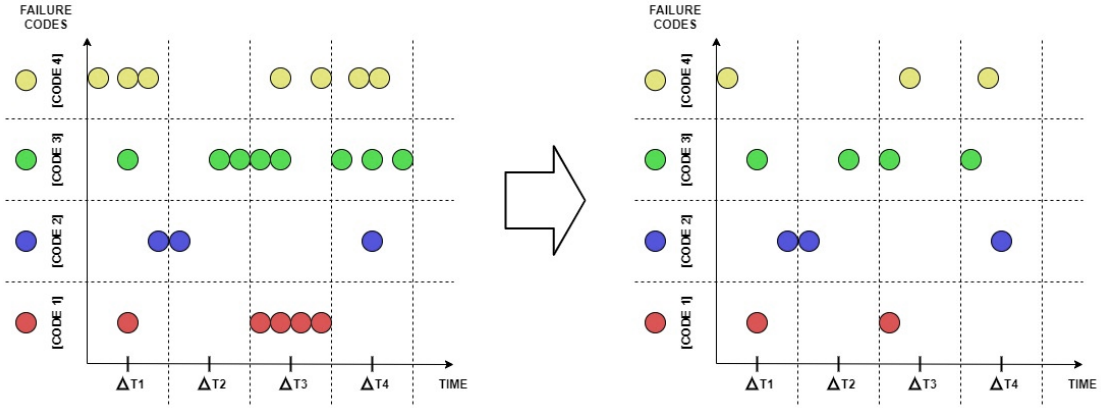


Figure 18 – The discretization process.

Source: The author (2020).

	TID	TIME	CODE	DESCRIPTION
BEFORE CONVERSION	1	2015-06-15 00:45:44	520221	['CHECK', 'TIGHT', 'CUTTER']
	1	2015-06-15 01:48:45	520229	['CUT', 'AXIS', 'TIGHT']
	1	2015-06-15 01:52:45	702033	['CROSS-DRILL', 'BAR', 'ADAPTER', 'SUPPORT', 'TIGHT']
	...	...	...	…
	m	2015-07-14 22:13:07	700454	['CHANGER', 'DOOR', 'CLOSE', 'TOOL']
	m	2015-07-14 22:15:11	601011	['CUTTER', 'SPINDLE']
	m	2015-07-14 23:18:42	600914	['TIGHTEN', 'COUPLE', 'SUPPLY', 'FLUID']

	TID	CODE TRANSACTION	TEXT TRANSACTION
AFTER CONVERSION	**1**	[520221, 520229, 702033, ...]	[‘CHECK’, ‘TIGHT’, ‘CUTTER’, ‘CUT’, ‘AXIS’, ‘CROSS-DRILL’, ‘BAR’, ‘ADAPTER’, ‘SUPPORT’, ...]
	...	…	…
	m	[…, 700454, 601011, 600904]	[…, CHANGER, DOOR, CLOSE, TOOL, CUTTER, SPINDLE, TIGHTEN, COUPLE, SUPPLY, FLUID]

Table 11 – Before and after the transaction format conversion.

Source: The author (2020).

Provided that, the 36 Transaction files corresponded to a different combination of a discretization time (Δt_d) and a transaction time (Δt_t). Not surprisingly, the number of transactions in each Transaction file varied according to the combination of Δt_d and Δt_t applied. Then, to facilitate the Transaction files referring, the 36 files received identification codes. This identification code was designed according to the combination of discretization time and transaction time set to create them. Table 12 presents the code naming scheme, and *Figure 19* presents the number of transactions contained in each one of the 36 files.

Discretization Time (min)	**Transaction Time (min)**					
	10	**20**	**30**	**40**	**50**	**60**
10	D10T10	D10T20	D10T30	D10T40	D10T50	D10T60
20	D20T10	D20T20	D20T30	D20T40	D20T50	D20T60
30	D30T10	D30T20	D30T30	D30T40	D30T50	D30T60
40	D40T10	D40T20	D40T30	D40T40	D40T50	D40T60
50	D50T10	D50T20	D50T30	D50T40	D50T50	D50T60
60	D60T10	D60T20	D60T30	D60T40	D60T50	D60T60

Table 12 – All transaction files code names.

Source: The author (2020)

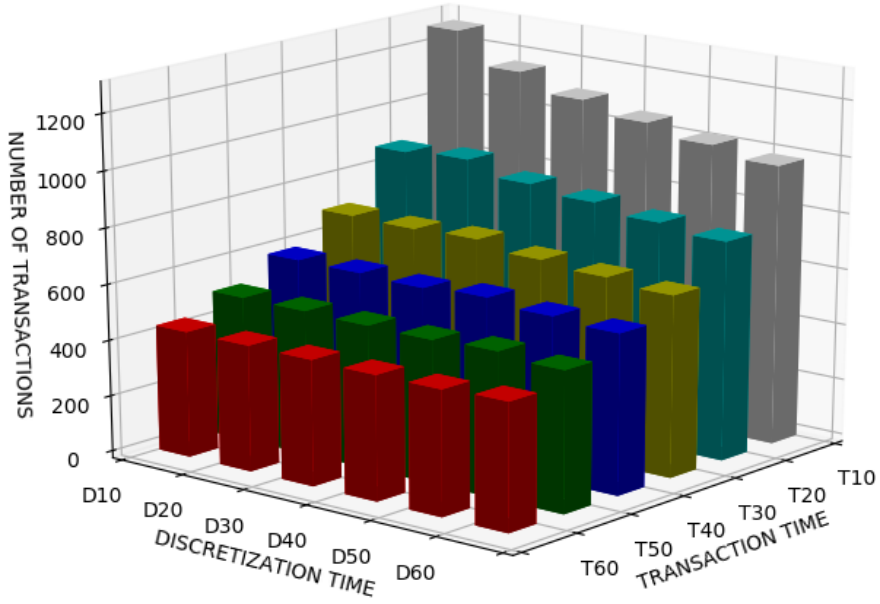


Figure 19 – The number of transactions by different combinations of Δt_d and Δt_t.

Source: The author (2020).

Intentionally different from the first datafile, the Transaction file highlights the co-occurrence relation among failure codes and descriptions within a single transaction time. In other words, the Transaction files are intended to support the mining process to identify cause-effect relationships among failure codes and features, which may be composed of one or more mechanical parts, processes, resources, or status.

Following, the *DM* phase is explored.

DATA MINING

Code and Text Data Mining

Being the "heart" of the KDD process, the DM phase defines the type of information to be mined by the technique to applied. In this KDD process intends to identify hidden cause-effect relationships among failure codes and failure description tokens; consequently (and based on the amount of data to be dug in), it is very suitable to use the *Association Analysis* task.

Furthermore, the key points of this phase lie in the structure and content of the input file the DM technique requires. The understanding of the structure allows us to set the file to be properly processed and the content defines the character of the information that will be processed. In our case, the *Association Analysis* technique is to cover both types of transaction, the *Code Transaction* and *Text Transaction*. And although the same method is to be applied, different analyses are to be realized as the input files' content differs.

The first step of this phase is converting the transaction files into the boolean transaction format. This new format must consider the presence or the absence of all items (codes or tokens) in every transaction. Moreover, the boolean files are of double-service, not only to support the AR mining algorithm, but also to support the item frequency counting process, which is presented only in the *Post Data Mining* phase. Table 13 portrays the structures of the *Boolean Transaction* file.

TID	CODE TRANSACTION					TEXT TRANSACTION				
	3000	4071	…	700454	700704	ADAPTER	AXIS	...	COOLING	ROTATION
1	0	0	…	0	0	0	0	…	0	0
2	1	0	…	1	1	1	0	…	0	1
3	0	1	…	0	1	0	1	…	0	1
…	…	…	…	…	…	…	…	…	…	…
n-2	0	0	…	1	0	0	0	…	0	0
n-1	0	1	…	1	0	0	1	…	0	0
n	0	0	…	1	1	0	0	…	0	1

Table 13 – Boolean Transaction file structure.

Source: The author, 2020.

Then, the association analysis technique was applied to both types of transactions, and for that purpose, the support and confidence threshold values had to be defined first. In summary, the higher these parameters are, the fewer rules are mined. For this reason, both parameters were defined with a low threshold value, both equal to 0.1, so the rules mined can be later analyzed to whether they are relevant or not. It is worth mentioning that even though only the support and confidence parameters were included into the mining process, for filtering purposes, the *lift, conviction and leverage* measure are presented in the *AR analysis* step to evaluate the generated rules

After having set the parameters threshold values, the association analysis must be performed to all Transaction files created as the consequence of the decisions made when defining the "discretization time" and the "transaction time". In those steps, 36 new files we created and all those files must be processed into the association mining step producing a different number of code and text rules. Table 14 summarizes the total number of code rules and text rules provided by each one of those 36 files. Some transaction files yielded too many rules, causing memory errors.

	CODE RULES						TEXT RULES					
	T10	T20	T30	T40	T50	T60	T10	T20	T30	T40	T50	T60
D10	14	20	39	70	113	3,735	725,890	-	-	-	-	-
D20	12	20	32	54	99	3,669	688,234	-	-	-	-	-
D30	9	16	41	52	84	3,622	643,851	798,368	-	-	-	-
D40	8	13	34	75	81	1,990	627,715	677,543	-	-	-	-
D50	6	11	22	40	123	1,832	575,152	684,961	-	-	-	-
D60	5	6	13	27	69	1,495	554,367	567,337	981,789	-	-	-

Table 14 – Number of code and text rules mined in each Transaction file.
Source: The author (2020).

In addition to the AR mining, the frequency of each item (code and token) within each transaction file were raised, creating 36 *Frequency* files. The Frequency file aims to support the company in different decision making aspects than the ones supported with *Association Analysis.* The Frequency files are presented and discussed in the next sections.

Up to this moment, the *DM* phase produced the 36 *AR* files (with their respective 36 *Item Frequency* files). As shown above, the number of rules in each Transaction file is determined by the values assigned to the discretization time and the transaction time. The files created with shorter Δt_d and longer $\Delta \mathrm{t_t}$ trended to produce greater numbers of AR. Complementarily, files created with longer Δt_d and shorter $\Delta \mathrm{t_t}$ trended to produce fewer rules. Furthermore, it is reasonable assuming that not all files and not all rules are worth working with; and therefore, a selecting file process and rule filtering process are required. The next phase of this KDD process explores this issue.

POST DATA MINING

Code and Text Post-Data Mining

In this phase, all the data produced is carefully filtered, selected, and analyzed, producing the final results. These results are to be presented through charts and graphs, aiming to provide relevant information to improve maintenance management. First of all, there will be applied a selecting procedure to elect one of the Transaction files (and consequently one of the Frequency files) and discard the others. Then, this Transaction file is to go under a filtering procedure to drop invalid and irrelevant rules in the file. Following, the remaining text and code rules are to be analyzed by their performance indicators and then employed under a network analysis to construct relationship diagrams. And finally, the Frequency file and the whole KDD process are to be evaluated regarding their management contributions to the maintenance department.

The process of selecting the proper Transaction file was based on the construction of a pairwise comparison matrix, which aimed to identify the Transaction files whose rules were presented in all other Transaction files, regardless of the combination of Δt_d and $\Delta \mathrm{t_t}$ that

originated them. In this case, as the number of rules varied from file to file, all AR files could be classified as subsets, supersets or different sets when compared to each other. For that purpose, all the comparisons intended to identify interrelationship among all files.

In detail, the algorithm compared all files by pairs, $A_i \subseteq B_j \; \forall \; i \neq j$, where A and B are the Transaction files, and $i = \{1,2 \ldots n\}$ and $j = \{1,2 \ldots n\}$ are the indexes of each AR file. As a result, those comparisons originated a $n \times n$ colored-matrix, such that when $A_i \subseteq B_j$ is true (which means that all rules of the A.R. file "A" are also presented in the A.R. file "B"), the color green was assigned, and when $A_i \subseteq B_j$ is false, (which means that not all rules of the A.R. file "A" are also presented in the A.R. file "B"), the color red was assigned. The color white was assigned when there was no need of comparison for the A.R. files "A" and "B" were referring to the same file because of indexers i and j refer to the same file. Figure 20 presents the colored-matrix and Figure 21 shows a general interrelationship diagram of all AR files.

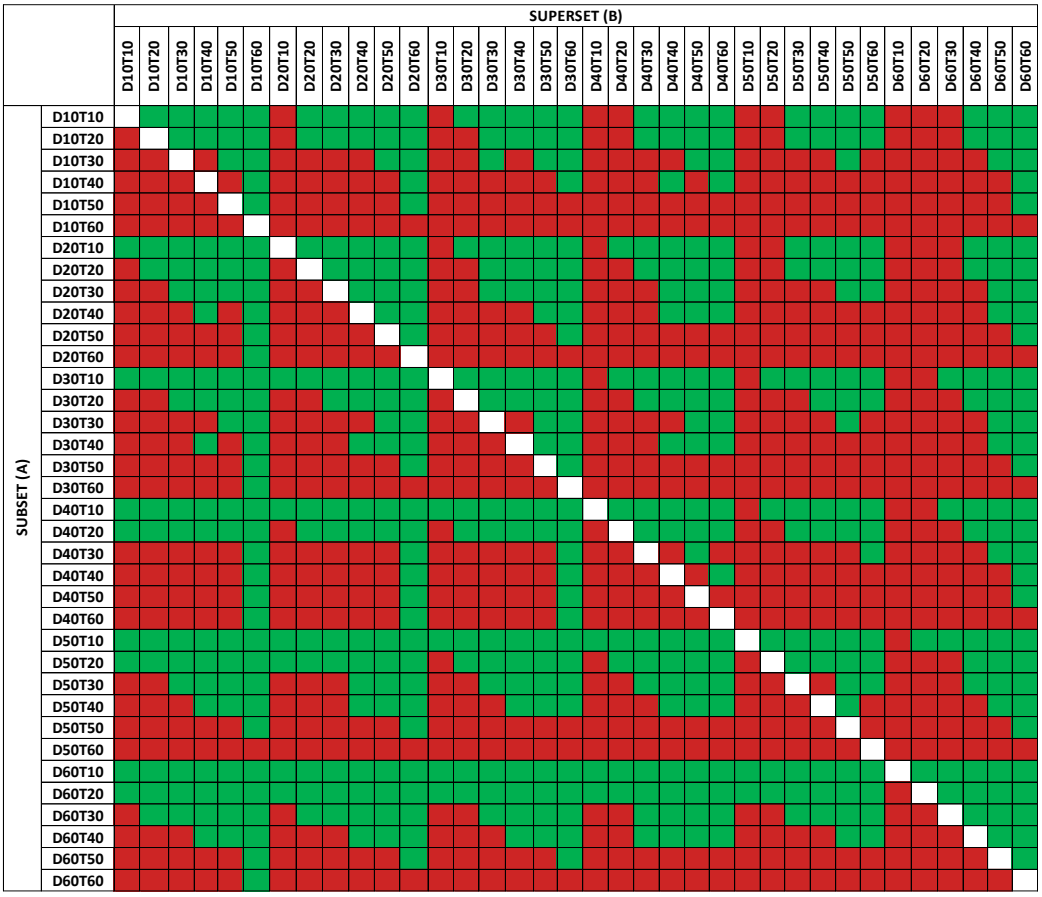


Figure 20 – Transaction files pairwise comparison matrix.

Source: The author (2020).

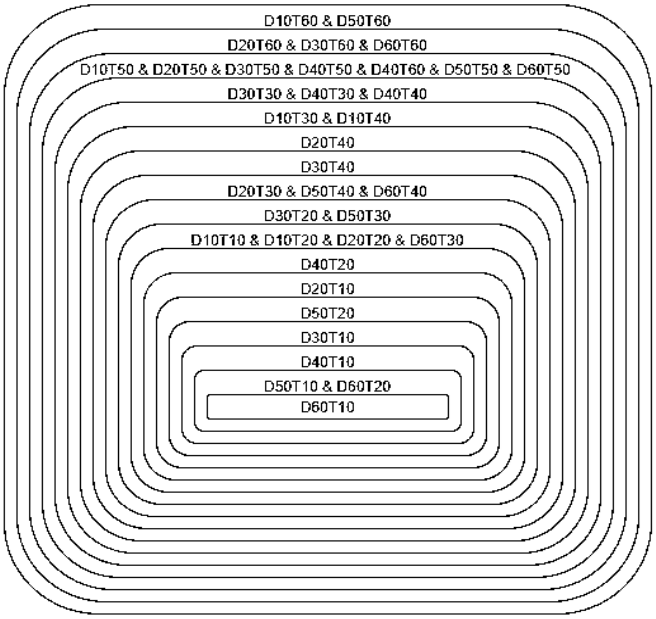


Figure 21 – Transaction files interrelationship diagram.

Source: The author (2020).

As it can be seen, out of 36 AR files, only 2 files (D10T60, and D50T60) could not be considered a subset in any case, 34 files could be considered a subset at least once, and only 1 file (D60T10) was classified as subsets in all cases. Therefore, the selecting file decision came down to the interrelationship diagram. Hence, the file which was considered as a subset in most of the cases, which also was the file with longest the discretization time and the shortest transaction time, the file D60T10 (that held 5 code rules and 554,367 text rules), was selected to carry on in this KDD process for being the most representative file.

After selecting the proper AR file, the next step was filtering and removing any invalid or irrelevant rule. The filtering process was divided into four parts. In the first part, which was directed to treat only text rules, only the rules that could be associated with more than two codes remained; otherwise, the rule is removed. In other words, taking the equation $N_i = \sum_{i=i}^{n} \sum_{j=1}^{m} (A_i \cup B_i) \subseteq D_j$, where A_i and B_i are respectively the antecedent and the consequent token sets of the rule X_i, D_j is the token set within each failure description in the dictionary file, n is the total number of text rules, and m is the total number of failures presented in the dictionary, the algorithm removes a rules X_i when $N_i < 2$. The goal behind this action was to remove rules that originated within a single type of failure. For instance, taking the lemmatized token set of one of the rules $D = ['CUT', AXIS', 'TIGHT']$ it may originate rules by itself, like

X_1: $['CUT', 'AXIS'] \rightarrow ['TIGHT']$ or X_2: $['CUT'] \rightarrow ['AXIS']$. After this process, the number of text rules fell 99%, from 554,367 to 38 rules.

The second part of the filtering rules process, designed to filter rules based on their support and confidence values, aimed to identify the parameters' values for all rules and remove rules that presented their confidence values equal to 1 for this type of rule indicates that their antecedents and consequents are always together as combinations of tokens from the same failure description. Table 15 presents the number of code rules and text rules within the ranges of support and confidence as evidence that this problem only occurred on the text rules for linguistic reasons. As noticed, the support parameter is much more effective in changing the number of rules than the confidence parameter and after this filtering and removing process, the number of text rules reduced 42%, which meant a fell from 38 to 22 rules.

CONFIDENCE VALUES	CODE DATA MINING										TEXT DATA MINING									
	SUPPORT VALUES										SUPPORT VALUES									
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
0.1	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.2	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.3	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.4	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.5	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.6	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.7	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.8	5	1	0	0	0	0	0	0	0	0	38	36	17	2	0	0	0	0	0	0
0.9	4	0	0	0	0	0	0	0	0	0	26	24	15	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	16	15	7	0	0	0	0	0	0	0

Table 15 – Number of code and text rules mined by different values of support and confidence.

Source: The author (2020).

The third part of the filtering rules process was based on the *simple redundancy* presented by Aggarwal and Yu (2001), which was presented here in the section *Filtering redundant association rules*. Even though this filter was applied to both types of rules, code rules and text rules, it only affected the text type. Not surprisingly, this filter affected oly the text type of rules because in that type the order of words (or tokens for this matter) in a sentence depends on the grammar rules of the language that the sentence is written, not on chronological orders. In other words, for text rules, the cause-consequence direction arrow is not meaningful, hence

the hypothetical rules [“A” → “B”] and [“B” → “A”] can be better written as [“A” ↔ “B”], both items happening together with no definition of single cause or consequence. As a result, after applying this filter, the number of text rules reduced by 40%, from 22 to 13 rules.

The last part of the filtering rules process was based on the technique presented by Liu, Hu and Hsu (2000), also presented in the section *Filtering redundant association rules.* This filter considers a summarization of rules focusing on rules that have fewer items in the antecedent and the consequent. For instance, let’s take the following hypothetical rules: [“A” → “C”], [“B” → “C”] and [“AB” → “C”], where “A”, “B” and “C” are item sets. In this case, the last rule would be considered redundant, therefore removed from the rules list. The application of this filter, especially, had an impact on both types of rules, causing a fell on the code rules from 7 to 3; and on text rules, a fell from 13 to 8. Table 16 and Table 17 respectively presents the list of remaining code rules and text rules.

N.º	ANTECEDENTS	CONSEQUENTS	SUPPORT	CONFIDENCE	LIFT	LEVERAGE	CONVICTION
1	[601011]	[600914]	0.278	0.814	2.075	0.144	3.274
2	[700454]	[600914]	0.243	0.858	2.185	0.132	4.275
3	[6413]	[600914]	0.196	0.901	2.294	0.111	6.118

Table 16 – List of remaining code rules.

Source: The author (2020).

N.º	ANTECEDENTS	CONSEQUENTS	SUPPORT	CONFIDENCE	LIFT	LEVERAGE	CONVICTION
1	['COUPLE']	['FLUID']	0.383	0.915	2.389	0.222	7.275
2	['COUPLE']	['SUPPLY']	0.383	0.915	2.325	0.218	7.152
3	['SUPPLY']	['FLUID']	0.383	0.973	2.541	0.232	23.257
4	['DOOR']	['CLOSE']	0.354	0.963	2.666	0.221	17.346
5	['BLOCK']	['CHANNEL']	0.150	0.960	1.796	0.066	11.638
6	['CHANNEL']	['TOOL']	0.438	0.820	1.537	0.153	2.596
7	['SPINDLE']	['CUTTER']	0.276	0.817	2.318	0.157	3.554
8	['TIGHTEN']	['CUTTER']	0.244	0.873	1.636	0.095	3.677

Table 17 – List remaining text rules.

Source: The author (2020).

As shown above, eleven rules were presented after the filtering process, 3 code rules and 8 text rules. These had to be assessed and validated concerning their respective values of *support*, *confidence*, *lift*, *leverage*, and *conviction*. Notably, all rules are complying with the support and confidence pre-established threshold values.

Regarding the *lift* values, al of them are superior to 1, in other words, the failure codes between the antecedent and consequent sides are positively correlated, considering no *minimum interest* value was not previously defined. The minimum *lift* value found was 1.537, which makes the current *minimum interest* equals to 0.537; and the maximum lift value was 2.666, which makes the *maximum interest* value equals to 1.666.

And concerning relative frequency rule analysis, the *leverage* values are presented. Particularly, all the correlations built through the rules were considered dependent on the view of the scenario here presented. For instance, the highest and the lowest *leverage* values are respectively 0.066 and 0.232.

Additionally, to highlight the relevance of the mined rules, the *conviction* values of each rule were mined and presented above. It is important to realize that the *conviction* value represents the tax of frequency the rule will present itself assertively until be presented as unassertively. The minimum and maximum *conviction* value respectively are 2.596 and 23.257.

In the final analysis, all the remained rules were validated based on the AR measurements. Then, in the next step, they were used to build the network diagrams to better explore the connections among the rules' components. As a result, 5 diagrams were built, one for the code rules and four for the text rules. *Table 18* presents all the diagrams separated by the type of rule, and additionally, next to each diagram, a box containing all codes that can be associated with the tokens within the text diagrams.

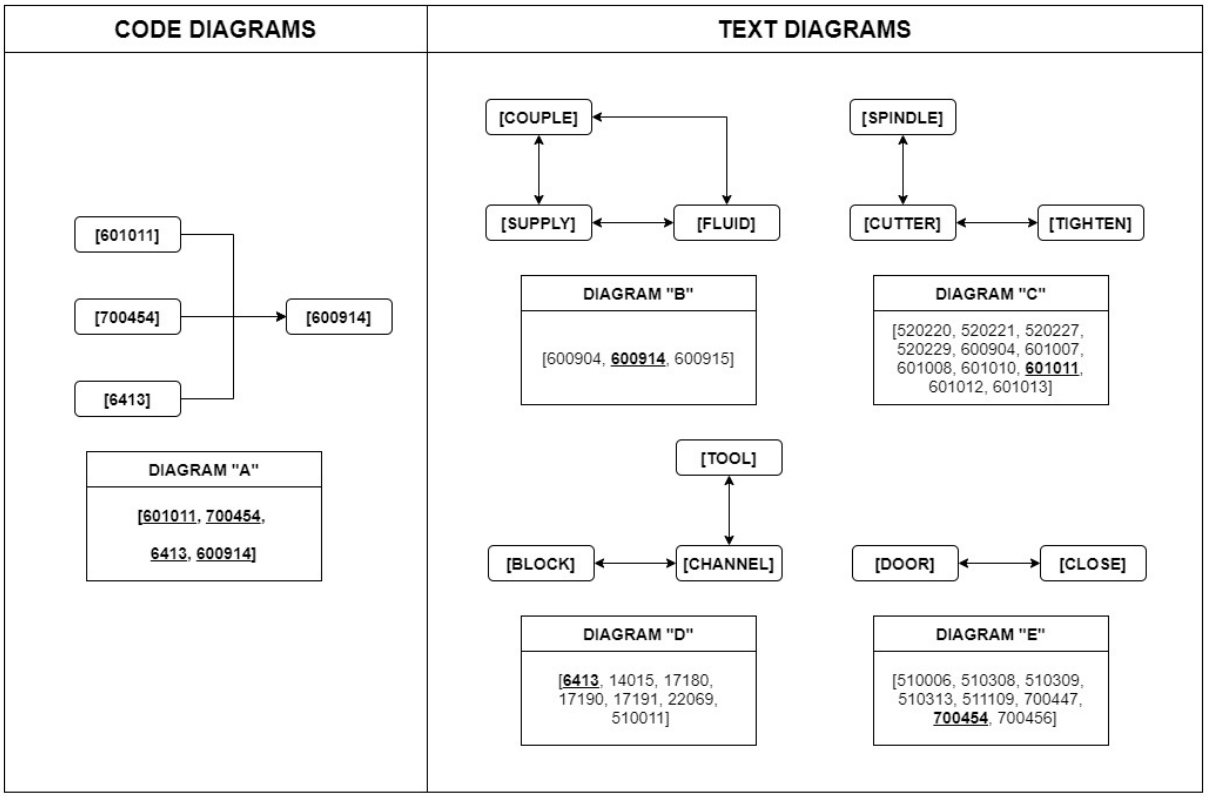


Table 18 – Network diagrams for the remained association rules.

Source: The author, 2020.

As presented above, only 4 failure codes were responsible for the construction of the code rules. The codes [6413], [601011] and [700454] and were only in the antecedent part, and the code [600914] was the only one in the consequent part. This fact provided the existence of just one stream of cause-effect relationship among those failure codes, which simplifies the execution of maintenance actions. Not surprisingly, the same 4 codes were associated with the text diagrams, which is evidence of the dependency of the code data and the text data.

Amid the text rules, 11 tokens formed the base items (antecedents and consequents) in the construction of the 4 text diagrams. Notably, the double-headed arrows in those diagrams indicate the double direction of the cause-effect relationship caused by the uncertainty brought by the grammatical and achronological order of words. It is worth mentioning that those codes inside the boxes are evidence of failures fixed through the same maintenance actions, or at least, have their occurrence decreased

Moreover, as mentioned in the DM Phase, the Frequency files presents the frequency of each item (code and token) to identify the most common items and cross-check them with the

results presented by the association analysis. Table 19 presents the 10 most common codes, and Table 20 presents the 10 most common tokens within each maintenance class in the Frequency file D60T10. The general 10 most frequent tokens are highlighted.

CODE	DESCRIPTION
600914	FLUID SUPPLY COUPLING NOT TIGHT.
6413	CHANNEL TOOL REACHED THE MONITORING LIMIT.
601011	CUTTER SPINDLE NOT IN OFF STATE.
700454	TOOL CHANGER DOOR ON THE WORKING SPACE NOT CLOSED.
700025	BEZEL MISSING CLAMPING PRESSURE.
700704	ELECTRICAL CABINETS RELATIVE AIR HUMIDITY EXCEEDED.
601114	MAIN SPINDLE GEAR REDUCTION IS ACTIVE.
700205	WEIGHT BALANCING ON HYDRAULIC OIL TEMPERATURE
510012	THE MAIN SPINDLE IS NOT IN OPERATING MODE AS AXIS.
601115	THE MAIN SPINDLE IS NOT IN OPERATING MODE AS AXIS.

Table 19 – Description of the ten most frequent failure codes and its frequency.

Source: The author, 2020.

MACHINE PART	PROCESS	RESOURCE	STATUS
CHANNEL	TIGHTEN	SUPPLY	ACTIVE
TOOL	COUPLE	FLUID	EXCEED
SPINDLE	MONITOR	PRESSURE	STOP
DOOR	REACH	AIR	OVERTEMPERATURE
CUTTER	CLOSE	HUMIDITY	NON-COMPLIANT
CHANGER	OPERATE	OIL	SAFE
BEZEL	BLOCK	WATER	FILTER
CABINET	REDUCTION	CHARGE	OPEN
GEAR	MEASURE	DATA	EMERGENCY
PROGRAM	TEST	POWER	FORCED

Table 20 – Ten most common token through all failure descriptions by classes.

Source: The author (2020).

Comparatively, all codes and tokens mined in the AR were also present in the most frequent items lists. This fact highlights the connection between both techniques as the *association analysis* requires a minimum support value, which works exactly as a percentual frequency threshold. In contrast, many codes associated with the text diagrams were not present in the *Frequency Code* list for this last one only presents the ten most common ones.

In the final analysis, the *KDD Process* produced significant output data to improve the maintenance plan strategically. Before applying the *KDD Process*, the only data available was a raw excel file with all relevant data dispersed, dislocated, and puzzled among a great portion

of irrelevant data. Thereunto, the KDD Process cleaned, structured, filtered and analyzed the file through proper techniques, then extracted knowledge to improve the company strategically. In the first phase of it, the relevant portion of the raw data was select, then this part went under code and text cleaning processes, and finally, transformed into a proper format for the *DM* technique. Thus, in the second phase, the *DM* technique was parallelly applied to the failure code content and the failure description content, both coupled with items' frequency counting processes. As a result, the DM phase outputted 36 AR files and 36 Frequency Item files. Finally, in the last phase, Post Data Mining, the 72 files went under a selecting file process and a filtering rules process, which prioritized the analysis to an AR file with 11 rules (3 of the code type and 8 of the text type) and on Frequency Item file. The AR file originated 5 cause-effect diagrams (1 from the code content and 4 from the text content), together with 2 frequency items lists (1 from the code content and 1 from text content). Compared to the raw data, these output files represent a very small and simply-comprehensible summary of the most frequent failures and the interrelationship among them, which not only indicates the main failures to be treated, but also provides a strategic order approach to manage them.

In general, the KDD Process focused on the most effective failures to be fixed and provided a possible cause-effect sequence strategy to be followed when planning the maintenance schedule. After all, out of the 151 failure codes, the KDD Process' output focused only on 4 of them, but these failures represented a considerable amount of 30.56% of all data logs; the *Text Data Mining* covered 28 possible failure codes, but this amount represented 37.62% of all data logs, a short increased when compared to the first one. Lastly, the *Codes Frequency List*, even though it did not express any relationship among all failure codes, can be used as a final approach to the maintenance action planning.

The next chapter summarizes the strategic implications on the maintenance management planning enabled through this KDD Process.

5 MANAGERIAL INSIGHTS

In general, the KDD Process presented aimed to deal with a discrete failure report database, extract significant failure patterns from it, and support the proposition of a strategic approach to fix them as well as improvements to the maintenance management processes. Before applying the KDD Process, no prior knowledge or any expert information was available beyond a raw excel file that held registration of all failures during a year. This file was partitioned, filtered, converted, processed under a DS techniques, and critically assessed before providing any relevant information, which was transcribed into 5 files: a Failure Code Dictionary, a Code Frequency List, a Classified Token Frequency List, and an AR Package (AR tables and diagrams). All those files add value to the maintenance management in a different way and can be associated/aligned or provide support to the application of a different set of techniques.

The first file produced was the *Failure Code Dictionary*. Although simple, that file supported the KDD Process when producing some of the other files, and provided a useful and quick way to identify the meaning behind each failure code. Considering all 151 possible failures, it is reasonable to assume the need for a quick way to associate the failure codes to their respective description, for fixing procedures. Besides, this file represents an initial step for further studies and improvements, like FMEA/FMECA/RCM, providing the failure description information.

As for the Code Frequency List, this file grants an ordered frequency list of all 151 failure codes that a manager can use to improve maintenance scheduling by focusing on the exploration of the most frequent ones through some techniques like: Fishbone diagram and Risk analysis tecniques to better understand their causes and consequences, respectively. Hence, it may be found that some of those failures are caused by inefficient maintenance procedures, which requires maintenance team training or other techniques, for this list opens the possibility for a large range of tools to be applied to continuously improve maintenance management. Also, the frequency information contributes to a FMEA/FMECA/RCM project.

On the other hand, the Categorized Failure Token List corroborates with the improvement of maintenance management in many distinct aspects, as it is not limited by the failure code itself, but counts the tokens occurrences indistinctly among all failure descriptions. In short,

there are 4 useful classes: *Machine Part*, *Process*, *Resource*, and *Status*. The *Machine Part* class contributes to the spare parts purchasing management, which may use it to acquire joint purchase strategies to lower costs. The *Process* class may indicates processes that are being misperformed among many different failure codes, which after improved may correct all failures at once. One valuable strategy would be applying the fishbone diagram technique to identify the common point about those processes that may be causing the failure. Extra team training or specialized maintenance teams may be considered as solutions, also. The *Resource* class is associated with the fail of providing processes' essential insource resources (like oil, heat, etc.) and/or outsource resources (like electricity, water, etc.). Both types require different approaches to deal with, for when the issue is about insourced resources, a manager should track its providing process, which may come from the same or another department, and try to optimize it. On the other hand, when the issue is about outsourced resources, the closeness of the supply chain is a crucial factor to solve the failure. For both types, techniques like Business Process Management (BPM), Quality Tools, and Supply Chain Management Methods are all favored. The *Status* class only aggregates information to the other classes in describing the failures and may be used in a FMEA/FMECA/RCM project to better understand the gravity of each failure.

As for the AR Package (charts and diagrams), it provides a clever understanding of the failure occurrences, for even though Frequency Lists represent relevant achievements, they do not offer any interrelational information about the failure codes or tokens to managers build their maintenance plans strategically. With only those lists' information, a manager would probably plan his maintenance actions based on the decreasing frequency codes order, which seems a sound decision; however, that maintenance plan may not be as efficient and effective as expected if some failures were unknowingly caused by a set of other failures. Nonetheless, this issue is solved by identifying cause-effect relationships through *Association Analysis* techniques. By applying those techniques, a manager would strategically excel in his maintenance planning, and save money and time by ordering maintenance actions prioritizing the most frequent "cause" failures, rather than the "effect" ones. Notably, the same codes presented in the code diagram can be found in the text diagram boxes, and in the first positions of the Failure Code List. This fact highlights the connection between the code and the text data, and provides an initial direction to which failures should be prioritized first.

In the final analysis, this KDD Process provided a relevant amount of information to improve maintenance management. Without any sophisticated input, it was able to describe failures, identify their frequency ranking and key-words, and even diagnose cause-effect relationships. Table 21 summarizes all this analysis.

FILE	INFORMATION	PRACTICAL UTILITY	TECHNIQUE SUGGESTION
Code Dictionary	Failure codes and their respective description.	Provide relevant information to the KDD Process when producing the other files, and a quick way to identify failures by their codes.	– FMEA / FMECA / RCM;
Code Frequency List	Frequency ranking of all failure codes	Identify the most frequent failures and direct the effort to diagnose and treat their causes.	– Maintenance scheduling – Fishbone diagram – Risk analysis techniques – FMEA / FMECA / RCM; – Maintenance team training
Classified Token Frequency List	Frequency ranking of all relevant tokens presented in the failure descriptions divided into the four maintenance classes: Machine Part, Process, Resource, and Status.	Provide support to improve maintenance management processes according to the information provided by each maintenance class.	– Machine Part: spare parts purchasing methods; – Process: fishbone diagram, maintenance team training, maintenance team specialization; – Resource: Business Process Management (BPM), Quality tools, Supply Chain Management Methods – Status: Support FMEA / FMECA / RCM;
AR Package (Tables and Diagrams)	List of code and text rules with their respective AR measurements values, and their resulting AR diagrams.	Identify cause-effect relationships among failure, and direct the effort for strategic maintenance planning and further studies.	– Maintenance scheduling – Fishbone diagram – Risk-analysis – Time-Series Model analysis – Failure Prognostic analysis

Table 21 – KDD Process' output analysis.

Source: The author (2020)

6 CONCLUSIONS

In conclusion, this research contributes to the advancement of knowledge in the Data Science field as well as with the maintenance management field. The points discussed here put into perspective the mutual-benefit relationship that exists between the mentioned areas, as well as its applications in real life. This relationship was explored in this research through all the chapters presented.

Initially, a brief bibliographic framework review on KDD Process, Association Analysis, and Natural Language Processing intended to provide the necessary knowledge to contextualize and describe the techniques applied in this research. In general, the KDD Process directed the application's methodology and the code construction in the application chapter, and the Association Analysis and NLP enabled the knowledge extraction process.

Then, the literature review highlighted the evolution of the Association Analysis within the maintenance management field through the years. It was also presented the occurrence of those topics in several countries around the world; the most common journals that presented publications on those topics together; and so on. Additionally, four literature questions were answered concerning the maintenance management aspects, the type of industry, the most common A.R. measurements applied in the mining process, and the list of additional techniques performed with Association Analysis in those articles. The literature review also made it possible to identify the absence of a joint application of Association Analysis with Natural Language Processing as presented in chapter 4 of this study.

The application chapter detailed the existing challenges and gains when performing the aforementioned techniques. As can be seen, the pre-processing phase constituted the major challenge in all the KDD process as the database was completely unstructured. Besides, the Pre-processing phase also faced all the issues found when handling textual data, here handled through NLP techniques. On the other hand, the DM phase represented the heart of the entire process, for it was the phase where all the relevant information was extracted through Association Analysis and frequency counting. And Finally, in the last phase, Post-Data Mining, the patterns and information extracted were filtered and profoundly assessed. Hence, the analysis and presentation of that data enabled the identity company's managerial improvements, as well as providing insights that guide the company strategically.

Finally, the *MANAGERIAL INSIGHTS* chapter provided a holistic view of the benefits of the application presented in the previous chapter, and how all the output files can benefit the maintenance management system. Some of the techniques there presented may not require much effort, and some of them certainly do require it, but the main point lies in the continuous and strategic improvement of the maintenance management system that can be achieved through those techniques.

Furthermore, some extra literature contributions can be mentioned in this research. Initially, it concerns the processes of producing and selecting the most representative Transaction file, which could be missed in the process. Then, the way of applying the Association Analysis technique to multiple items (tokens) inside text vectors (text descriptions) at once, resulting in an enormous number of text rules. And lastly, the text rule filters applied that efficiently and effectively reduced the number of rules to a manageable set for further analysis and discussion. These contributions enriched the KDD Process as well as this research.

The contributions presented in this research are related to the maintenance management field; however, they provide guidelines for the application of the same techniques in other fields as well as, and suggestions for aggregation with other techniques. For future researches in the field of maintenance management, it is mentioned the association of the techniques presented here with the techniques presented in the *MANAGERIAL INSIGHTS* chapter, especially the FMEA / FMECA / RCM.

REFERENCES

AGGARWAL, C. C. **Data Mining: The Textbook**. Cham: Springer, 2015.

AGGARWAL, C. C.; YU, P. S. A new approach to online generation of association rules. **IEEE Transactions on Knowledge and Data Engineering**, v. 13, n. 4, p. 527–540, 2001.

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. Mining Association in Large Databases. **Proceedings of the 1993 ACM SIGMOD international conference on Management of data - SIGMOD '93**, p. 207–216, 1993.

ALLAHYARI, M. et al. A Brief Survey of Text Mining: Classification, Clustering and Extraction Techniques. 2017.

ANTOMARIONI, S. et al. A predictive association rule-based maintenance policy to minimize the probability of breakages: application to an oil refinery. **International Journal of Advanced Manufacturing Technology**, v. 105, n. 9, p. 3661–3675, 2019.

ASHRAFI, M. Z.; TANIAR, D.; SMITH, K. A new approach of eliminating redundant association rules. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 3180, p. 465–474, 2004.

BAGUI, S.; DHAR, P. C. Positive and negative association rule mining in Hadoop's MapReduce environment. **Journal of Big Data**, v. 6, n. 1, 2019.

BALLARD, C. et al. **Dynamic Warehousing: Data Mining Made Easy**. North Castle: International Business Machines Corporation, 2007.

BENITES, F.; SAPOZHNIKOVA, E. Evaluation of hierarchical interestingness measures for mining pairwise generalized association rules. **IEEE Transactions on Knowledge and Data Engineering**, v. 26, n. 12, p. 3012–3025, 2014.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural Language Processing with Python**. Sebastopol: O'Reilly, 2009.

BRILL, E. Transformation-Based Error-Driven Learning and Natural Language Processing : A Case Study in Part-of-Speech Tagging. 1995.

CALÇADA, D. B.; REZENDE, S. O.; TEODORO, M. S. Analysis of green manure decomposition parameters in northeast Brazil using association rule networks. **Computers and Electronics in Agriculture**, v. 159, n. February, p. 34–41, 2019.

CIARAPICA, F.; BEVILACQUA, M.; ANTOMARIONI, S. An approach based on association rules and social network analysis for managing environmental risk : A case study from a process industry. **Process Safety and Environmental Protection**, v. 128, p. 50–64, 2019.

COSTANTINO, M. et al. Natural language processing and information extraction: Qualitative analysis of financial news articles. **IEEE/IAFE Conference on Computational Intelligence for Financial Engineering, Proceedings (CIFEr)**, p. 116–122, 1997.

DASH, S. et al. Big data in healthcare: management, analysis and future prospects. **Journal of Big Data**, v. 6, n. 1, 2019.

DESHMUKH, J.; BHOSLE, U. Image Mining using Association Rule for Medical Image dataset. **Procedia - Procedia Computer Science**, v. 85, n. Cms, p. 117–124, 2016.

DING, J. et al. Alarm association rules mining in flight booking system based on sliding time window model. **Journal of Software Engineering**, v. 8, n. 4, p. 419–427, 2014.

DJATNA, T.; ALITU, I. M. An Application of Association Rule Mining in Total Productive Maintenance Strategy: An Analysis and Modelling in Wooden Door Manufacturing Industry. **Procedia Manufacturing**, v. 4, n. Iess, p. 336–343, 2015.

DUARTE, J. C.; CUNHA, P. F.; CRAVEIRO, J. T. Maintenance database. **Procedia CIRP**, v. 7, p. 551–556, 2013.

DUNHAM, M. **Data Mining: Introductory and Advanced Topics**. 1. ed. Bergen: Pearson, 2002.

EL-DEHAIBI, N.; MACDONALD, E. F. Extracting customer perceptions of product sustainability from online reviews. **Proceedings of the ASME Design Engineering Technical Conference**, v. 2B-2019, p. 1–13, 2019.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, n. 3, p. 37–53, 1996.

GANDOMI, A.; HAIDER, M. Beyond the hype: Big data concepts, methods, and analytics. **International Journal of Information Management**, v. 35, n. 2, p. 137–144, 2015.

GHOLAMI, P.; HAFEZALKOTOB, A. Maintenance scheduling using data mining techniques and time series models. **International Journal of Management Science and Engineering Management**, v. 13, n. 2, p. 100–107, 2018.

GRABOT, B. Rule mining in maintenance: Analysing large knowledge bases. **Computers and Industrial Engineering**, v. 139, n. November 2018, p. 105501, 2020.

HAN, H. K.; KIM, H. S.; SOHN, S. Y. Sequential association rules for forecasting failure patterns of aircrafts in Korean airforce. **Expert Systems with Applications**, v. 36, n. 2 PART 1, p. 1129–1133, 2009.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3. ed. Waltham: Elservier, 2012.

HARMOUCHE, J.; NARASIMHAN, S. Long-Term Monitoring for Leaks in Water. **IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS**, v. 16, n. 1, p. 258–266, 2020.

IGUAL, L.; SEGUÍ, S. **Introduction to Deep Learning**. Cham: Springer, 2017.

JU, C. et al. A Novel Method of Interestingness Measures for Association Rules Mining Based on Profit. **Discrete Dynamics in Nature and Society**, v. 2015, n. 2, 2015.

KAMMOUN, M. A.; REZG, N. Toward the optimal selective maintenance for multi-component systems using observed failure: applied to the FMS study case. **International Journal of Advanced Manufacturing Technology**, v. 96, n. 1–4, p. 1093–1107, 2018.

KAMSU-FOGUEM, B.; RIGAL, F.; MAUGET, F. Mining association rules for the quality improvement of the production process. **Expert Systems with Applications**, v. 40, n. 4, p. 1034–1045, 2013.

KOTU, V.; DESHPANDE, B. **Data Science: Concepts and Practice**. Cambridge: Elsevier, 2019.

LAKSHMI, K. S.; VADIVU, G. ScienceDirect ScienceDirect Extracting Association Rules from Medical Health Records Using Multi-Criteria Decision Analysis. **Procedia Computer Science**, v. 115, p. 290–295, 2017.

LETRACHE, K.; EL BEGGAR, O.; RAMDANI, M. OLAP cube partitioning based on association rules method. **Applied Intelligence**, v. 49, n. 2, p. 420–434, 2019.

LI, L. et al. Condition assessment of power transformers using a synthetic analysis method based on association rule and variable weight coefficients. **IEEE Transactions on Dielectrics and Electrical Insulation**, v. 20, n. 6, p. 2052–2060, 2013.

LI, T. et al. Mining of the association rules between industrialization level and air quality to

inform high-quality development in China. **Journal of Environmental Management**, v. 246, n. June, p. 564–574, 2019.

LIAO, P. C.; CHEN, H.; LUO, X. Fusion Model for Hazard Association Network Development: A Case in Elevator Installation and Maintenance. **KSCE Journal of Civil Engineering**, v. 23, n. 4, p. 1451–1465, 2019.

LIU, B. et al. A study on the use of discrete event data for prognostics and health management : discovery of association rules. **PHM Society European Conference**, p. 1–7, 2017.

LIU, B.; HU, M.; HSU, W. Multi-level organization and summarization of the discovered rules. **Proceeding of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, p. 208–217, 2000.

LIU, C. H.; CHEN, L. S.; HSU, C. C. An association-based case reduction technique for case-based reasoning. **Information Sciences**, v. 178, n. 17, p. 3347–3355, 2008.

LIU, Y.; GAO, S.; YU, L. A novel fault prevention model for metro overhead contact system. **IEEE Access**, v. 7, p. 91850–91859, 2019.

MAHMOUDI, N.; DOCHERTY, P.; MOSCATO, P. Deep neural networks understand investors better. **Decision Support Systems**, v. 112, p. 23–34, 2018.

MAHMUD, M. et al. A Survey of Data Partitioning and Sampling Methods to Support Big Data Analysis. **Big Data Mining and Analytics**, v. 3, n. 2, p. 85–101, 2020.

MAIMON, O.; ROKACH, L. **Data Mining and Knowledge Discovery Handbook**. 2. ed. Boston: Springer, 2010.

MAQUEE, A.; SHOJAIE, A. A.; MOSADDAR, D. Clustering and association rules in analyzing the efficiency of maintenance system of an urban bus network. **International**

Journal of Systems Assurance Engineering and Management, v. 3, n. 3, p. 175–183, 2012.

MARKOU, I.; KAISER, K.; PEREIRA, F. C. Predicting taxi demand hotspots using automated Internet Search Queries. **Transportation Research Part C: Emerging Technologies**, v. 102, n. March, p. 73–86, 2019.

MÁRQUEZ, A. C.; DE LA FUENTE CARMONA, A.; ANTOMARIONI, S. A process to implement an artificial neural network and association rules techniques to improve asset performance and energy efficiency. **Energies**, v. 12, n. 18, 2019.

MINER, G. et al. **Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications**. Oxford: Elsevier, 2012.

MOHARANA, U. C.; SARMAH, S. P. Determination of optimal kit for spare parts using association rule mining. **International Journal of Systems Assurance Engineering and Management**, v. 6, n. 3, p. 238–247, 2015.

MOHARANA, U. C.; SARMAH, S. P. Determination of optimal order-up to level quantities for dependent spare parts using data mining. **Computers and Industrial Engineering**, v. 95, p. 27–40, 2016.

MOHARANA, U. C.; SARMAH, S. P.; RATHORE, P. K. Application of data mining for spare parts information in maintenance schedule: a case study. **Journal of Manufacturing Technology Management**, v. 30, n. 7, p. 1055–1072, 2019.

MOSADDAR, D.; SHOJAIE, A. A. A data mining model to identify inefficient maintenance activities. **International Journal of Systems Assurance Engineering and Management**, v. 4, n. 2, p. 182–192, 2013.

NIKHATH, A. K.; SUBRAHMANYAM, K.; VASAVI, R. Building a K-Nearest Neighbor Classifier for Text Categorization. **International Journal of Computer Science and**

Information Technologies, v. 7, n. 1, p. 254–256, 2016.

PEREIRA, V.; COSTA, H. G. A literature review on lot size with quantity discounts: 1995-2013. **Journal of Modelling in Management**, v. 10, n. 3, p. 341–359, 2015.

REDER, M.; YÜRÜŞEN, N. Y.; MELERO, J. J. Data-driven learning framework for associating weather conditions and wind turbine failures. **Reliability Engineering and System Safety**, v. 169, n. January 2017, p. 554–569, 2018.

RILLOF, E.; LEHNERT, W. Information Extraction as a Baiss for High-Precision TExt Classification. **ACM Transactions on Information Systems**, v. 12, p. 296–333, 1994.

RUIZ, E.; CASILLAS, J. Adaptive fuzzy partitions for evolving association rules in big data stream. **International Journal of Approximate Reasoning**, v. 93, p. 463–486, 2018.

RUIZ, P. P.; FOGUEM, B. K.; GRABOT, B. Generating knowledge in maintenance from Experience Feedback. **Knowledge-Based Systems**, v. 68, p. 4–20, 2014.

SANTORINI, B. Part-of-Speech Tagging Guidelines for the Penn Treebank Project. 1991.

SARKAR, D. **Text Analytics with Python**. Berkeley: Apress, 2016.

SHEIKH, L. M.; TANVEER, B.; HAMDANI, S. M. A. Interesting measures for mining association rules. **Proceedings of INMIC 2004 - 8th International Multitopic Conference**, p. 641–644, 2004.

SINGLE, J. I.; SCHMIDT, J.; DENECKE, J. Knowledge acquisition from chemical accident databases using an ontology-based method and natural language processing. **Safety Science**, v. 129, n. February, p. 104747, 2020.

SUZUKI, T.; GEMBA, K.; AOYAMA, A. Hotel classification visualization using natural

language processing of user reviews. **IEEE International Conference on Industrial Engineering and Engineering Management**, p. 892–895, 2014.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. **Introduction to Data Mining**. Boston: Pearson, 2006.

TELIKANI, A.; GANDOMI, A. H.; SHAHBAHRAMI, A. A survey of evolutionary computation for association rule mining. **Information Sciences**, v. 524, p. 318–352, 2020.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review* Introduction: the need for an evidence- informed approach. **British Journal of Management**, v. 14, p. 207–222, 2003.

WANG, Y. et al. An algorithm for mining of association rules for the information communication network alarms based on swarm intelligence. **Mathematical Problems in Engineering**, v. 2014, 2014.

WEN, F. et al. Computers & Industrial Engineering A hybrid temporal association rules mining method for traffic congestion prediction. **Computers & Industrial Engineering**, v. 130, n. 6, p. 779–787, 2019.

YU, L.; SCHACH, S. R. Applying association mining to change propagation. **International Journal of Software Engineering and Knowledge Engineering**, v. 18, n. 8, p. 1043–1061, 2008.

ZHANG, C.; ZHANG, S. **Association Rule Mining: Models and Algorithms**. Berlin ; Heidelberg ; NewYork ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Tokyo : Springer, 2002.