



HAL
open science

Data Mining Driven Approaches for Predictive Maintenance, Inventive Design Modelling and Energy Storage Systems

Ahmed Samet

► **To cite this version:**

Ahmed Samet. Data Mining Driven Approaches for Predictive Maintenance, Inventive Design Modelling and Energy Storage Systems. Artificial Intelligence [cs.AI]. Université de strasbourg, 2023. tel-04775592

HAL Id: tel-04775592

<https://hal.science/tel-04775592v1>

Submitted on 10 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



Thesis publicly defended on 23 November 2023 for the

HABILITATION À DIRIGER DES RECHERCHES IN COMPUTER
SCIENCE

Data Mining Driven Approaches for Predictive Maintenance, Inventive Design Modelling and Energy Storage Systems

Author: Ahmed SAMET

Supervisor:	Romuald Boné	Professor, INSA Centre Val de Loire
Reviewer:	Robert J. Howlett	Director of KES International Research
Reviewer:	Salem Benfarhat	Professor, Université d'Artois
Reviewer:	Bruno Cremeilleux	Professor, Université de Caen Normandie
Examiner:	Hubert Cardot	Professor, Ecole Polytechnique de l'Université de Tours
Examiner:	Pierre Gañçarski	Professor, Université de Strasbourg
Examiner:	Osmar R. Zaiane	Professor, University of Alberta
Examiner:	Claudia Hentschel	Professor, HTW Berlin, University of Applied Sciences
Examiner:	Denis Cavallucci	Professor, INSA Strasbourg

“Success consists of going from failure to failure without loss of enthusiasm.”

—Winston Churchill

Contents

List of Figures	x
List of Tables	xii
I Introduction	1
1 Introduction	3
II State of the Art	9
2 Sequential Pattern Mining : Application to Predictive Mining	11
Introduction	12
2.1 Pattern Mining from Sequential Data	12
2.1.1 Background	12
2.1.2 Mining algorithms	14
2.2 Chronicle Mining	18
2.2.1 Chronicle foundations	18
2.2.2 Discovering Frequent Chronicles	20
2.3 Predictive Maintenance and Applications	23
2.3.1 Physical models	23
2.3.2 Data-driven models	24
2.3.2.1 Machine learning-based approaches	25
2.3.2.2 Data mining-based approaches	26
2.3.2.3 Statistical learning-based approaches	27
2.3.3 Hybrid models	27
2.4 Conclusion	28
3 Text Mining Application to TRIZ Inventive Design Modelling	29
Introduction	30
3.1 Representation Learning on Natural Language Processing	30
3.1.1 Non contextual embedding	32

3.1.1.1	N-gram Model	32
3.1.1.2	Bag-of-Words	32
3.1.1.3	Distributed Representation	33
3.1.1.4	Neural Probabilistic Language Model	34
3.1.2	Contextual embeddings models	35
3.1.2.1	Recurrent networks-based encoder	35
3.1.2.2	Transformers	38
3.1.2.3	Convolutional Neural Networks (CNN)	42
3.2	Text Mining for Enhancing Innovation Using TRIZ	43
3.2.1	Contradiction	44
3.2.2	The TRIZ Knowledge Sources	44
3.2.3	Implementation of TRIZ on the Problem Solving	46
3.2.4	Automatizing TRIZ Inventive Process with Machine Learning	47
3.2.4.1	Contradictions and TRIZ parameters extraction	48
3.2.4.2	Inventive principles, physical effects and solution mining	49
3.2.4.3	Estimation of inventiveness level	49
3.2.5	Summary	50
3.3	Conclusion	53

III Scientific Contributions 55

4	Chronicle Mining and Reasoning in Predictive Maintenance 57
	Introduction 58
4.1	Mining Failure Chronicles for Predictive Maintenance 58
4.1.1	CPM Algorithm 59
4.1.2	Clasp-CPM 60
4.1.2.1	Subsequence graph 62
4.1.2.2	Multi-threading 64
4.2	Failure Detection with Chronicles 65
4.2.1	Maximum Confidence-based approach 65
4.2.2	The algorithm for rule base integration with detection of the three issues 67
4.2.3	Chronicle fusion approach 69
4.3	Experiments 71
4.3.1	The SECOM Data Set 71
4.3.2	Evaluation of the chronicle mining 72

4.3.3	Predictive maintenance analysis	75
4.4	Conclusion	78
5	Patent Mining for Inventive Design Modelling	79
	Introduction	80
5.1	Full Framework of TRIZ Patent Inventive Design	80
5.2	Mining Patent TRIZ Contradictions	80
5.2.1	Baseline Approach	82
5.2.2	PaGAN: Patent Generative Adversarial Network	83
5.2.2.1	Document-based approach for contradiction mining	84
5.2.2.2	GAN and semi-supervised training	86
5.3	SAM-IDM for Inventive solution mining	89
5.4	Inventive solution mining using Multiple-Criteria Decision Analysis (MCDA)	91
5.5	Experiments	95
5.5.1	Contradiction mining experiments	95
5.5.2	Contradiction matching experiments	102
5.5.3	Use case example with PatRIS	104
5.6	Conclusion	105
6	Battery Prognostic Using Neural Networks Models	107
6.1	Introduction and related works	108
6.1.1	Literature review of battery ageing analysis	108
6.1.2	Battery ageing datasets	110
6.1.3	Motivation and challenges	111
6.2	AE-LSTM based model for SOH prediction	112
6.2.1	Data pre-processing	113
6.2.2	Encoding branch	114
6.2.3	LSTM branch	115
6.2.4	Smoothing	115
6.3	Window exogenous LSTM for SOH prediction	116
6.3.1	Feature extraction	116
6.3.1.1	Temporal domain	117
6.3.1.2	Statistical domain	117
6.3.1.3	Spectral domain	117
6.3.2	Feature selection	118
6.3.3	Sliding window approach for SOH prediction	118
6.4	SoHAP: SoH explainable prediction based on SHAP	119

6.4.1	Convolutional Neural Network used in SoHAP	120
6.4.2	Post-hoc explainability	121
6.5	Experiments	123
6.5.1	SOH forecasting experiments	124
6.5.1.1	SOH forecasting on the MIT dataset	124
6.5.1.2	SOH forecasting on the NASA PCoE dataset	125
6.5.2	Comparison with other approaches	127
6.5.2.1	MIT prediction results	127
6.5.2.2	NASA prediction results	129
6.5.3	Explainability analysis	130
6.6	Conclusion	133
IV Conclusion & Perspectives		135
7	Conclusions and Perspectives	137
7.1	Pattern mining in predictive maintenance	138
7.2	Mining inventive design using TRIZ theory	139
7.3	Battery prognostic using neural networks models	139
7.4	Perspectives	140
7.4.1	Crossing pattern mining and explainability for smart battery management	140
7.4.2	Crossing TRIZ patent mining and knowledge reasoning	142
7.4.3	Management Strategies, Industrial and Financial Prospects	144
7.4.3.1	Management Strategies and Collective Scien- tific Responsibilities	145
7.4.3.2	Industrial prospect	146
7.4.3.3	Research fundings	146
V Appendices		149
A	TRIZ Contradiction Matrix	151
A.1	39 Generic Engineering Parameters	151
A.2	40 Inventive Principles	153
A.3	76 Inventive Standards	160
A.3.1	Some Extra Precision on Inventive Standards	160
A.3.1.1	Class 1: Building and Destruction of Substance- Field Models	160

A.3.1.2	Class 2: Development of Substance-Field Models	161
A.3.1.3	Class 3: System Transitions and Evolution-Transition to Super-system and Sub-system	162
A.3.1.4	Class 4: Solutions for Detection and Measurement	162
A.3.1.5	Class 5: Standards on Application of Standards	163
B	Evidence theory	165
C	Introduction to Neural Networks	167
C.1	Formal neuron and neural networks	167
C.2	Training	169
C.2.1	Loss	170
C.2.2	Error backpropagation	171
C.3	Training and specialization	172
D	Author's scientific publications	175
E	Research funding	183
E.1	Laboratory and school internal projects	183
E.2	Industrial projects	184
E.3	Regional projects	184
E.4	National projects	185
E.5	European projects	186
E.6	Statistics	187
	Bibliography	189

List of Figures

1.1	The three main contributions of this thesis showing the interaction between data mining and Industry 4.0	5
2.1	Example of timestamped sequence	19
2.2	Example of a chronicle	20
2.3	The chronicle pattern-growth step as illustrated in (Cram, Mathern, and Mille, 2012)	22
3.1	The generic process of NLP (Chen et al., 2021)	31
3.2	An example of Bag-of-Words	33
3.3	An example for the distributed representation of words	34
3.4	The architecture of a Recurrent Neural Network and its equivalent unfolded over a sequence.	36
3.5	Architecture of a Long Short Term Memory neural network	37
3.6	Architecture of a Long Short Term Memory unit	37
3.7	The encoder-decoder structure of the Transformer architecture Taken from (Vaswani et al., 2017a)	39
3.8	BERT's Attention Mechanism (Devlin et al., 2019)	39
3.9	BERT architecture based on Transformers with twelve encoder blocks.	40
3.10	Basic structure of CNN for NLP	42
3.11	Basic operational structure of the convolutional layer	42
3.12	An example of TRIZ contradiction	45
3.13	The processing of solving inventive problems via classical TRIZ	46
4.1	Figure illustrating the steps of the chronicle mining	59
4.2	Subsequence graph: After three steps	63
4.3	Subsequence graph: After four steps	63
4.4	Subsequence graph: seventh step	64
4.5	Subsequence graph: useless nodes in red are marked to be pruned	64
4.6	Different steps used in the frequent failure chronicle mining approach, adapted from (Sellami et al., 2019).	73

4.7	Example of a chronicle	76
5.1	The full process for mining inventive designs	81
5.2	Baseline approach: SummaTRIZ (Guarino et al., 2020)	83
5.3	PaGAN architecture	87
5.4	The structure of MaLSTM developed by (Mueller and Thyagarajan, 2016)	91
5.5	The framework of PatRIS	92
6.1	SOH	111
6.2	IR	111
6.3	Historical Features	111
6.4	Charge current	111
6.5	Architecture summary	113
6.6	SOH-window-XLSTM architecture	116
6.7	I_C RMS	118
6.8	SoHAP architecture. The vector $[e_0, \dots, e_n]$ represents the contributions of each input value on the SoH.	119
6.9	Real VS 25-cycles ahead predicted SOH for battery b2c5	125
6.10	Real VS 300-cycles ahead predicted SOH for battery b3c27	126
6.11	Evolution of the average and minimum MAE and RMSE according to the number of cycles ahead	127
6.12	Real VS 12-steps ahead predicted SOH for NASA battery # 5	128
6.13	Real VS 12-steps ahead predicted SOH for NASA battery # 6	128
6.14	SOH curve for battery b1c1.	132
6.15	Shapley values of features over different cycles.	132
6.16	Shapley values of features over three consecutive cycles.	132
7.1	Cross relation between the different investigated topics	141
7.2	Multimodal explainable approach for battery prognosis	142
7.3	Deep active SAT -based learning for TRIZ matrix finding	144
C.1	Perceptron	168
C.2	Neural Network	169
C.3	Gradient descent	171
E.1	Breakdown of funding sources	188
E.2	Amount of net INSA Strasbourg funding obtained per year	188
E.3	Breakdown of research supervision	188

List of Tables

2.1	Sequence dataset	13
3.1	Summary of the state of the art in TRIZ domain	52
4.1	Sample data set	60
4.2	Sample suffix database : P is the failure event, s -concatenated at the end of each sequence	61
4.3	Sequence enumeration	62
4.4	Details of the functions of Algorithm 8	70
4.5	Number of chronicles of Clasp-CPM, CPM, DCM and FACE w.r.t data set size, sequence's size, dictionary size, maximum events/item and support threshold	74
4.6	Sequences' data set	76
4.7	Quality of prediction on the SECOM data set	78
5.1	Inventive Solutions from Different Domains	92
5.2	A sample of the indicator detail	94
5.3	Sentence classification 1 (with LSTM generator)	97
5.4	Sentence classification 2 (with LSTM generator)	97
5.5	Document classification (with LSTM generator)	98
5.6	Sentence classification 1 (with ANN document classifier)	98
5.7	Sentence classification 2 (with ANN document classifier)	99
5.8	Document classification (with ANN document classifier)	99
5.9	Similarity matching experimental results on the labelled SNLI dataset (left) and the labelled patent sentence pairs (right) for different thresh- old values	103
5.10	Ranking of sentence given for the formulated contradiction	104
6.1	XLSTM performances for SOH prediction on MIT batteries	126
6.2	XLSTM performances for SOH prediction on NASA batteries	128
6.3	Comparison of SOH prediction performances, 50 cycles ahead, on MIT batteries	129
6.4	Architecture comparison	131

List of Abbreviations

TRIZ	T eoriya R esheniya I zobretatelskikh Z adach
SHAP	S Hapley A dditive e x P lanations
FACE	F requency A nalysier for C hronicle E xtraction
HCDA	H euristic C hronicle D iscovery A lgorithm
ML	M achine L earning
SVM	S upport V ector M achine
NLP	N atural L anguage P rocessing
BERT	B idirectional E ncoder R epresentations from T ransformers
BOW	B ag-of- W ords
GRU	G ated R ecurrent U nit
TF-IDF	T erm F requency- I nverse D ocument F requency
LDA	L atent D irichlet A llocation
CPM	C hronicle P attern M ining
FADE	F ailure D etection with C hronicles
FCP	F usion of C hronicle for P rediction
kNN	k N earest N eighbours
GAN	G enerative A dversarial N etworks
SAM-IDM	S imilarity-based A pproach for M erging IDM -related knowledge
MCDA	M ultiple- C riteria D ecision A nalysis
ANN	A rtificial N eural N etwork
FNN	F eedforward N eural N etwork
CNN	C onvolutional N eural N etwork
RNN	R ecurrent N eural N etwork
LSTM	L ong S hort T erm M emory
MaLSTM	M anhattan L ong S hort T erm M emory
PHM	P rognostics and H ealth M anagement
EV	E lectric V ehicles
HEV	H ybrid E lectric V ehicles
TCO	T otal C ost of O wnership
Li-Ion	L ithium- I on
LFP	L ithium F er P hosphate

NMC	Nickel Manganese Cobalt
IR	Internal Resistance
SOH	State Of Health
RUL	Remaining Useful Life
AGV	Automated Guided Vehicles
EoL	End of Life
NAR	Non Linear Auto-Regressive
IR	Internal Resistance
HF	Historical Features
XAI	eXplainable Artificial Intelligence
I	Current
V	Voltage
T°	Temperature
X_D	X during Discharge
X_C	X during Charge

*This thesis is dedicated to the memory of Arnaud
Martin who passed away in August 2023.*

Part I

Introduction

Chapter 1

Introduction

The industry is in the midst of a digital revolution exponentially accelerated by technologies in full growth, such as automation, communication, information technology, and Artificial Intelligence (AI) (Gökalp, Şener, and Eren, 2017). These technologies have opened the gates to new ways of production, in which machines, people, processes, and data (the manufacturing processes actors) involved in manufacturing processes, interact autonomously making intensive use of the Internet and the latest manufacturing technologies, with the primary purpose of developing smarter and more efficient factories (Santos et al., 2017). This new way of production has been called *Industry 4.0*.

The term Industry 4.0 was introduced in the Hanover Trade Fair in 2011 as a concept to denote the new generation of connected robotics, and intelligent factories. This fourth industrial revolution includes concepts as the Internet of Things (IoT), as well as servitization and Cyber-Physical Systems (CPS). It succeeds the three industrial revolutions that influenced the manufacturing domain many years ago.

One of the main goals of Industry 4.0 is to relieve human laborers to focus on work that is more engaging and valuable. Several repetitive applications exist aiming to steer human intervention towards performing, efficient and less wasteful production. Among applications, we may list:

- Augmented reality;
- Taking care of human factor in manufacturing plants, robotics;
- Additive manufacturing (3D printing);
- Monitoring and quality control;
- Predictive maintenance;
- Composites, new materials and assembly automation and robotics;

- Energy efficiency;
- Automation of R&D design and inventive design.

Machine learning (ML) is one of the main technological advances that is allowing Industry 4.0 to gain a foothold in businesses and on factory floors. Machine learning is essentially a subdomain of Artificial Intelligence that allows systems and algorithms to decide without being explicitly programmed to do so. Owing to ML, companies and specifically machines are able to take decision without the human supervision. Machines are now capable of supplying and replacing humans in a few cases when the task is repetitive and transferable to machines.

In this thesis, as highlighted in Figure 1.1, we focus on three distinct applications based on data mining and machine learning, which are: (i) predictive maintenance, (ii) inventive design modelling and (iii) energy efficiency. In these three applications, we investigate the use of data mining techniques for Industry 4.0. Indeed, data mining is a well-known sub-discipline of ML that aims to extract knowledge from large volumes of data and find knowledge when humans are incapable to do so. The following paragraphs describe the various interactions between data mining and Industry 4.0

The first interaction is explored between data mining and predictive maintenance. In this topic, we investigate the use of pattern mining to explain the failure happening in a production line. Concerning predictive maintenance in smart factories, pattern mining has been widely used to discover frequently occurring temporally-constrained patterns, through which warning signals can be sent to humans for a timely intervention (Dousson and Duong, 1999). Among pattern mining techniques, chronicle mining has been applied to industrial data sets for extracting temporal information of events (Cram, Mathern, and Mille, 2012). The extracted temporal information is valuable for predicting potential machinery failures that may appear in the future (Cram, Mathern, and Mille, 2012). However, even though chronicle mining results are expressive and interpretable representations of complex temporal information, domain knowledge is required for users to have a comprehensive understanding of the mined chronicles (Pei, Han, and Wang, 2002). The HALFBACK project - this work was funded by the InterReg offensive science project HALFBACK¹ deals with this theme and aims to support Small and Medium Enterprises (SMEs) in the production process and

¹<https://www.interreg-rhin-sup.eu/projet/halfback-usines-intelligentes-transfrontalieres-hautement-disponibles-dans-le-cloud/>

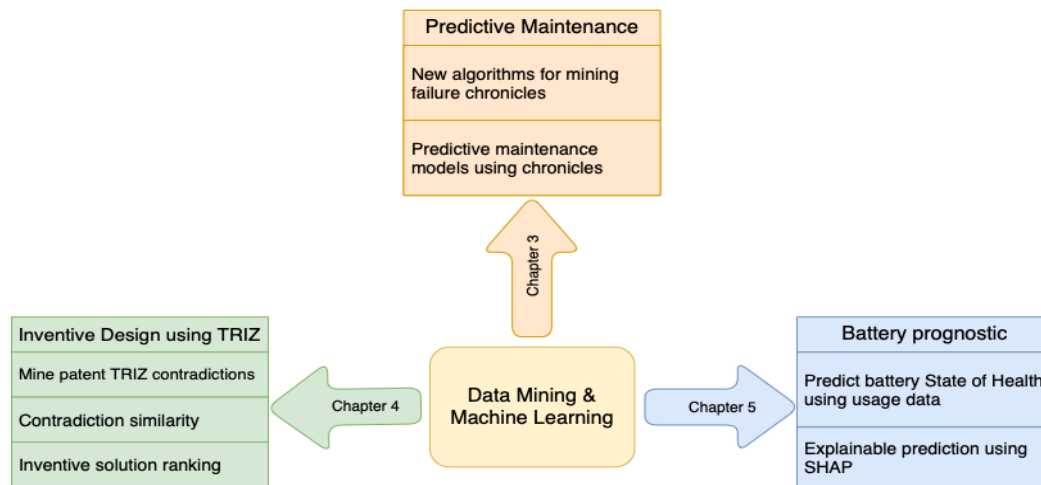


FIGURE 1.1: The three main contributions of this thesis showing the interaction between data mining and Industry 4.0

reduce machine downtime. To do so, a software prototype which predicts and optimizes maintenance planning using the machine's own sensor data was developed during the project. At the same time, options for (partial) outsourcing of the production were indicated so that it did not have to be stopped. An intelligent "machine broker" made possible a statewide coordination of companies and machines.

The second interaction explores the interaction of text mining and TRIZ theory (Altshuller, 1996). The goal is to analyze patent documents to extract TRIZ contradiction. A contradiction is a domain-free formulation of a problem. When comparing problems from different fields, the challenge that arises is the vocabulary used for their description, which is fundamentally different for a problem in mechanics and a problem in chemistry. Thus, Altshuller introduced general parameters (called TRIZ parameters) which can apply to all domains to describe problems (Weight of Stationary Objects, Speed, Power, Waste of Energy...). A contradiction from TRIZ domain is seen as a parameter improvement which leads to another parameter degradation where compromising between these two parameters is not the appropriate path to invent. One should find a solution that both improve the first parameter while also improving the second. The matrix gathers the inventive principles (i.e. the inventive paths to follow) that statistically have the best chance of success on every possible contradiction between TRIZ parameters.

The idea, in this thesis, is to mine and find patents sharing the same contradiction. This work was steered by two PhD students funded by the Chinese government scholarships and one by the INSA Strasbourg scholarship.

The third interaction consists in analyzing Lithium-ion batteries in order to predict the Remaining Useful Life (RUL). The goal is in line with the topic of energy efficiency in Industry 4.0. Indeed, we assume that when the RUL is predictable, the user could be notified and recommendation could be made to extend it. To do so, we focus on analyzing Lithium-ion batteries cells RUL using user usage data such as current, voltage and/or temperature. The problem is then assumed to be a regression problem using exogenous variables. To the best of our knowledge, the problem is not sufficiently investigated in literature. Therefore, we propose a twofold contribution. First, we aim at using the regressive neural networks model to predict the RUL using the usage data. In this part, we investigate, in particular, how to compress the data of the usage time series. Secondly, we intend to make these black boxes more explainable. We add a layer of SHAP model (Lundberg and Lee, 2017) on the top of a predictive model to explain the relation between the predicted output and the inputs. This work was funded by the InterReg offensive science project HALFBACK and VEHICLE². The VEHICLE project focuses on the development of intelligent management strategies based on advanced algorithms. ML models are among the considered solutions to manage the battery system.

This thesis provides 5 chapters including the state of the art and contributions. The order of the chapters has been dictated by the chronological sequence in which work began on these topics. The chapters are listed below:

- **Chapter 2: Sequential Pattern Mining: Application to Predictive Mining** covers the state of the art of sequential pattern mining. We introduce the definitions of a rich pattern called *chronicle*. Finally, we address all the existing approaches dealing with the predictive maintenance.
- **Chapter 3: Text Mining Application to TRIZ Inventive Design Modelling** covers the state of the art in text mining and the full process needed to mine large amounts of data. Particular emphasis is given to deep neural networks to build very rich token representations. We discuss TRIZ and works dealing with patent analysis.

²<https://www.interreg-rhin-sup.eu/projet/vehicule-source-de-stockage-hybride-batterie-li-ion-supercondensateurs-avec-une-machine-synchrone-a-reluctance-variable-pour-les-vehicules-electriques/>

-
- **Chapter 4 : Chronicle Mining and Reasoning in Predictive Maintenance** introduces all failure chronicles mining algorithms in the context of Industry 4.0. The second part of this chapter details three different approaches to use these chronicles on the prediction step.
 - **Chapter 5: Patent Mining for Inventive Design Modelling** covers three main contributions. They are: Mining TRIZ contradictions in patent documents; Matching contradictions with each other using similarity measurement; Ranking similar patents considering several criteria.
 - **Chapter 6: Battery Prognostic Using Neural Networks Models** details two approaches to predict the state of battery cells health using usage data. All approaches are based on neural networks. Finally, we explain the found results using explainable model based on SHAP.
 - **Appendix** provides further details about several models theories. The Evidence Theory, the Neural Network Basic, and TRIZ concepts are highlighted. We also provide the list of publications by the author as well as the projects funding list.

Part II

State of the Art

Chapter 2

Sequential Pattern Mining : Application to Predictive Mining

Contents

Introduction	30
3.1 Representation Learning on Natural Language Processing	30
3.1.1 Non contextual embedding	32
3.1.2 Contextual embeddings models	35
3.2 Text Mining for Enhancing Innovation Using TRIZ	43
3.2.1 Contradiction	44
3.2.2 The TRIZ Knowledge Sources	44
3.2.3 Implementation of TRIZ on the Problem Solving	46
3.2.4 Automatizing TRIZ Inventive Process with Machine Learning	47
3.2.5 Summary	50
3.3 Conclusion	53

Introduction

In this chapter, we review the state of the art of the sequential pattern mining domain. We start by presenting the basics of sequential pattern mining. These definitions are the basis of the chronicle mining approaches of which we review the foundations and the state of the art. All pattern mining definitions are presented formally using illustrative examples. These definitions will serve the contributions of Chapter 4

In the second part of this chapter, we focus on the domain of predictive maintenance, one of the main applicative pillars of Industry 4.0, and the different used-model families. We detail the state of the art of predictive maintenance and the different approaches. A special attention is paid to machine learning and data-mining based approaches.

2.1 Pattern Mining from Sequential Data

Sequential pattern mining is a branch of data mining whereby knowledge is extracted from data structured as *sequences*. In the following subtitles, we give an introduction to the different sequential mining notions used throughout this document.

2.1.1 Background

Definition 1 (item, itemset, sequence). Let $I = \{i_1, i_2, \dots, i_n\}$ be a set of items. An itemset is a subset of items. A sequence is an ordered¹ list of itemsets. A sequence S is denoted by $\langle s_1, s_2, \dots, s_l \rangle$, where s_j is an itemset, i.e. $s_j \subseteq I$ for $1 \leq j \leq l$. l is the size of the sequence.

The problem of sequential pattern mining was proposed by Agrawal and Srikant (Srikant and Agrawal, 1996), as the problem of mining interesting subsequences in a set of sequences.

Example 1. Assuming a sequence dataset of Table 2.1. itemsets are shown between parentheses while items are without. We use single lowercase letters a, b, \dots to denote items. We omit braces for singleton itemset, i.e., writing simply a for $\{a\}$. We omit all commas.

Definition 2 (itemset inclusion). An itemset $\beta = (b_i)_{i \in \{1, \dots, m\}}$ is a sub-itemset of $\alpha = (a_i)_{i \in \{1, \dots, n\}}$, denoted $\beta \sqsubseteq \alpha$, iff there exists a sequence of m integers $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $\forall k \in \{1, \dots, m\}, b_k = a_{i_k}$.

¹A classical order on items is the lexicographical order

TABLE 2.1: Sequence dataset

SID	sequences
10	$\langle a(abc)(ac)d(cf) \rangle$
20	$\langle (ad)c(bc)(ae) \rangle$
30	$\langle (ef)(ab)(df)cb \rangle$
40	$\langle eg(af)cbc \rangle$

Definition 3 (Sequence inclusion). A sequence $S = \langle s_1, s_2, \dots, s_m \rangle$ is a subsequence of another sequence $S' = \langle s'_1, s'_2, \dots, s'_n \rangle$ and denoted $S \preceq S'$, if there exist integers $1 \leq i_1 < i_2 < \dots < i_m \leq n$ such that $s_k \sqsubseteq s'_{i_k}, \forall k \in \{1, \dots, m\}$.

Example 2. In the sequence dataset of Table 2.1, $\langle (ef)c \rangle$ is subsequence of the Sequence 30.

Definition 4 (Sequence support). Let $\mathcal{D} = \{S_1, S_2, \dots, S_N\}$ be a sequence dataset. The absolute support of a sequence S , denoted $\text{supp}(S)$, is the number of sequence \mathcal{D} in which S is a subsequence:

$$\text{supp}(S) = |\{S_i \in \mathcal{D} | S \preceq S_i\}| \quad (2.1)$$

Generally, an itemset (aka pattern) could be either frequent or rare depending on its support. A pattern is considered as frequent if its support is greater than or equal to a given threshold f_{min} , otherwise, it is considered as a rare pattern.

Definition 5 (Support). The support of a sequence is the frequency of occurrence of said sequence in a given sequential database: if S belongs to the sequential database, and S' is a subsequence of S , not necessarily in the database, then S' occurs within S .

The **absolute support** of S is the number of sequences in the sequential database in which S is a subsequence.

The **relative support** of S is the number of entries in the sequential database containing S divided by the size of the database. Its value is between 0 and 1.

We will note $\text{supp}(S)$ the absolute support of a sequence S .

Definition 6 (Frequent Sequences). Given an arbitrary value f_{min} , called minimum support, a sequence is said to be frequent if its support is greater than or equal to f_{min} . We note the set of all frequent sequences \mathcal{FS} .

Example 3. Assuming a minimum support $f_{min} = 2$

Sequence $S_1 = \langle abc \rangle$ is considered frequent, since it occurs in 2 out of 4 sequences in the dataset $\text{supp}(S_1) = 2 > f_{min}$ (SID 10 and 40).

Sequence $S_2 = \langle (ab)c \rangle$ is considered frequent in the dataset since it occurs in 2 out of 4 sequences (10 and 30) in the database $\text{supp}(S_2) = 2 \geq f_{min}$.

Sequence $S_3 = \langle (ab)cf \rangle$ is not considered frequent in the database since it occurs in 1 out of 4 sequences in the dataset $\text{supp}(S_3) = 1 < f_{min}$.

Definition 7 (Closed Frequent Sequences). *The Closed Frequent Sequences set (\mathcal{CFS}) is defined as follows:*

$$\mathcal{CFS} := \{s | s \in \mathcal{FS} \wedge (\nexists \beta \in \mathcal{FS} | s \subseteq \beta \wedge \text{supp}(s) = \text{supp}(\beta))\}$$

In other words, a frequent sequence S is said to be closed if there is not a frequent sequence S' such that $S \preceq S'$ and S' has the same support.

Example 4 (Closed frequent sequences examples). *Referring to Table 2.1, with a relative minsupp of 0.5, one can see that the sequence $S_3 = \langle (ab)f \rangle$ is frequent closed sequence, as it has a support of 0.5 and there exists no other super sequence of S_3 that has the same support.*

Thus, we see that by keeping closed patterns only, we filter simpler, less informative patterns, but we retain those which are more frequent than other, more complex ones.

Remark 1 (Inclusion). *The set of closed frequent sequences is a subset of the frequent sequence set: $\mathcal{CFS} \subseteq \mathcal{FS}$.*

Several algorithms have been proposed to mine frequent and closed frequent sequential patterns (Fournier-Viger et al., 2017). In the following subsections, we detail several main sequential pattern mining algorithms.

2.1.2 Mining algorithms

Pei et al., 2001 propose PrefixSpan algorithm. The essence of this algorithm is to check only prefix subsequences and only their corresponding postfix subsequences are visualized into projected dataset, instead of processing the whole sequential dataset. PrefixSpan uses a direct application of the Apriori (Agrawal and Srikant, 1994) property to reduce candidate sequences.

The input of PrefixSpan is a sequence dataset and a user-specified threshold f_{min} . In the output, PrefixSpan discovers all frequent sequential patterns occurring in a sequence dataset.

Definition 8 (Prefix). A sequence $S' = \langle s'_1, s'_2, \dots, s'_m \rangle$ is a prefix of $S = \langle s_1, s_2, \dots, s_n \rangle$, denoted $S \prec_b S'$, iff $\forall i \in \{1, \dots, m-1\}, s'_i = s_i$ and $s'_m \sqsubseteq s_m$. We have $S' \prec_b S \Rightarrow S' \prec S$.

Definition 9 (Projection). Given two sequences T and S such that T is a subsequence of S , i.e., $S \preceq T$. A subsequence T' of the sequence T (i.e., $T' \preceq T$) is called a projection of T w.r.t prefix S , and noted $T' = T|S$, if and only if: (i) T' has prefix S ; (ii) there exists no proper super-sequence T'' of T' (i.e., $T' \sqsubseteq T''$ but $T' \neq T''$) such that T'' is a subsequence of T and also has prefix S .

Definition 10 (Concatenation operators). Let $S = \langle s_1, \dots, s_n \rangle$ and $S' = \langle s'_1, \dots, s'_m \rangle$ be two sequences. Let \diamond_\bullet be the concatenation operator. We define two kinds of concatenations (Yin, Zheng, and Cao, 2012) :

- $\diamond_i : S \diamond_i S' = \langle s_1, \dots, s_{n-1}, (s_n \cup s'_1), s'_2, \dots, s'_m \rangle$. Here, the first element of S' merges with the last element of S , and then we just append the rest of the second sequence.
- $\diamond_s : S \diamond_s S' = \langle s_1, \dots, s_{n-1}, s_n, s'_1, s'_2, \dots, s'_m \rangle$. This is the usual concatenation.

Definition 11 (New definition of prefix and suffix). Assuming two sequences $S = \langle s_1, s_2, \dots, s_n \rangle$ and $S' = \langle s'_1, s'_2, \dots, s'_m \rangle$. If $S'' = S' \diamond S$, S' is a prefix of S'' and S is a suffix of S'' .

Pei et al., 2001 introduce an algorithm for mining the complete set of frequent sequential pattern called *PrefixSpan*. *PrefixSpan*, as shown in Algorithm 1, is pattern-growth-based sequential pattern mining, based on depth-first search. *PrefixSpan* separates entries into prefixes and suffixes. Frequent prefixes are captured and the prefix's projection becomes a suffix. *PrefixSpan* scans the data once to find length-1 sequential patterns and extends frequent length-1 sequential patterns recursively. Extending is done by setting length-1 as the prefix and the remaining items that start with length-1 as the suffix (referred to as the projected database), and recursively increase the length of the prefix. The projected database will shrink after every iteration as the suffix decreases in length.

Example 5. For $f_{min} = 2$, *PrefixSpan* finds length-1 sequential patterns and prunes those infrequent:

Algorithm 1 PrefixSpan Algorithm

Require: \mathcal{D} : sequence dataset, f_{min} : minimum support threshold, α : a sequential pattern, i : the length of α , $\mathcal{D}|\alpha$: the α -projected database, if $\alpha \neq \langle \rangle$; otherwise, the sequence database \mathcal{D}

Ensure: The complete set of sequential pattern

- 1: Scan $\mathcal{D}|\alpha$ once, find the set of frequent items b such that
 - b can be assembled to the last element of α to form a sequential pattern;
 - or
 - $\langle b \rangle$ can be appended to α to form a sequential pattern.
- 2: For each frequent item b , appended to α form a sequential pattern α' , and output α'
- 3: For each α' , construct α' -projected database $\mathcal{D}|\alpha'$, and call PrefixSpan(α' , $i + 1, \mathcal{D}|\alpha'$).

SID	sequences
10	$\langle a(abc)(ac)d(cf) \rangle$
20	$\langle (ad)c(bc)(ae) \rangle$
30	$\langle (ef)(ab)(df)cb \rangle$
40	$\langle eg(af)cbc \rangle$

→

$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$	$\langle f \rangle$	$\langle g \rangle$
4	4	4	3	3	3	1

PrefixSpan computes the projection databases. Below the projection databases of length-1 frequent sequential patterns:

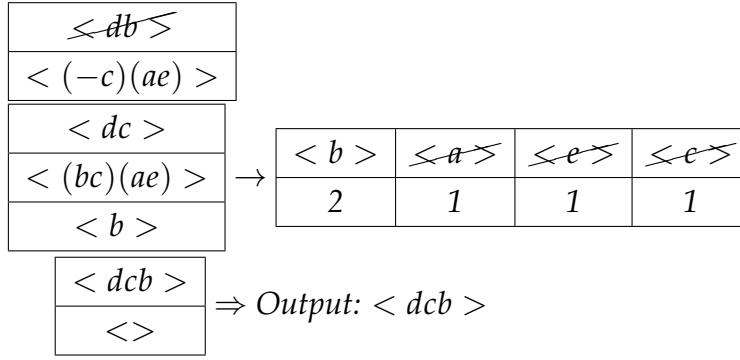
$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$	$\langle f \rangle$
$\langle (-abc)(ac)d(cf) \rangle$	$\langle (-c)(ac)d(cf) \rangle$	$\langle (ac)d(cf) \rangle$	$\langle (cf) \rangle$	$\langle (-f)(ab)(df)cb \rangle$	$\langle (ab)(df)cb \rangle$
$\langle (-d)c(bc)(ae) \rangle$	$\langle (-c)(ae) \rangle$	$\langle (bc)(ae) \rangle$	$\langle c(bc)(ae) \rangle$	$\langle (af)cbc \rangle$	$\langle (abc) \rangle$
$\langle (-b)(df)cb \rangle$	$\langle (df)cb \rangle$	$\langle b \rangle$	$\langle (-f)cb \rangle$		
$\langle (-f)cbc \rangle$	$\langle c \rangle$	$\langle bc \rangle$			

Let us take as example the prefix $\langle d \rangle$:

$\langle d \rangle$
$\langle (cf) \rangle$
$\langle c(bc)(ae) \rangle$
$\langle (-f)cb \rangle$

Find subsets of sequential patterns:

$\langle a \rangle$	$\langle b \rangle$	$\langle c \rangle$	$\langle d \rangle$	$\langle e \rangle$	$\langle f \rangle$	$\langle g \rangle$
1	2	3	0	1	1	1



CloSpan, as shown in Algorithm 2, is a depth-first based algorithm proposed by Yan et al. (Yan, Han, and Afshar, 2003). CloSpan implements the PrefixSpan algorithm. In fact, it is an optimization of the latter, intended to prune the search space by avoiding browsing certain branches in the recursive division process (by detecting in advance the unclosed sequential patterns).

Definition 12 (Total number of items). $I(\mathcal{D})$ represents the total of items and is computed as follows:

$$I(\mathcal{D}) = \sum_{i=1}^n |s_i|$$

Algorithm 2 CloSpan Algorithm

Require: A sequence s , a projected database $\mathcal{D}|s$, a threshold f_{min} .

Ensure: The prefix search lattice L .

- 1: Check whether a discovered sequence s' exists either $s \sqsubseteq s'$ or $s' \sqsubseteq s$, and $I(\mathcal{D}|s) = I(\mathcal{D}|s')$
 - 2: **if** such super-pattern or sub-pattern exists **then**
 - 3: modify the link in L , return
 - 4: **else**
 - 5: insert s into L
 - 6: Scan $\mathcal{D}|s$ once, find every frequent item α such that:
 - s can be extended to $(s \diamond_i \alpha)$,
 - or
 - s can be extended to $(s \diamond_s \alpha)$
 - 7: **if** no valid α available **then**
 - 8: return
 - 9: **for** valid α **do**
 - 10: Call CloSpan($s \diamond_i \alpha$, $\mathcal{D}|s \diamond_i \alpha$, f_{min} , L)
 - 11: **for** valid α **do**
 - 12: Call CloSpan($s \diamond_s \alpha$, $\mathcal{D}|s \diamond_s \alpha$, f_{min} , L)
 - 13: return
-

Since $a < b$, we scan the projected base of a before that of b . Moreover, we can detect that a always appears before f in the dataset. The idea of CloSpan is to say that any sequence of the form $\langle (b) + \alpha \rangle$, with α a sequence added as I-extension or S-extension (SPAM), will have the same support as $\langle (ab) + \alpha \rangle$. It is therefore not necessary to scan the projected base of b and only the results from the projected base of a are retained.

Another analyzed case by CloSpan: $f < g$, we scan the projected base of f before that of g . Moreover, we can detect that g always appears before f in the dataset. In this case, any sequence of the form $\langle (f) + \alpha \rangle$ will have the same support as $\langle (f)(g) + \alpha \rangle$. It is therefore not necessary to scan the projected base of e , but in this case, we can add (g) in front of all the results from the scan on the projected base of b and keep the sequences of the form $\langle (f)(g) + \alpha \rangle$ as the closed result.

2.2 Chronicle Mining

Sequential patterns are mined to extract frequent events or correlations between events. Unfortunately, they lack descriptiveness and temporal details. Therefore, a new type of rich pattern, called *chronicle*, has been introduced (Cram, Mathern, and Mille, 2012). In the following parts, we introduce basics of chronicle mining.

2.2.1 Chronicle foundations

Definition 13 (Event). *An event (Cram, Mathern, and Mille, 2012), is a couple (e, t) with $e \in \mathbb{E}$ and $t \in \mathbb{T}$ where \mathbb{E} is the set of events and \mathbb{T} is the time set.*

These events appear together in their order of occurrence, called timestamped events, which allows us to form a sequence.

Definition 14 (Timestamped sequence). *Let \mathbb{E} be a set of event types, and \mathbb{T} a time domain such that $\mathbb{T} \subseteq \mathbb{R}$. \mathbb{E} is assumed totally ordered by the relation $\leq_{\mathbb{E}}$. A sequence is a couple $\langle SID, \langle (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle \rangle$ such that SID is the index of the sequence and $\langle (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle$ is a sequence of events. The events in the sequence are ordered by \preceq defined as $i, j \in [1, n], i < j \Leftrightarrow (e_i, t_i) \preceq (e_j, t_j) \Leftrightarrow t_i < t_j \vee (t_i = t_j \wedge e_i <_{\mathbb{E}} e_j)$.*

The appearance of timestamped events in a sequence allows us to define *temporal constraints* between them.

Definition 15 (Temporal constraint). A time constraint is a quadruplet (e_1, e_2, t^-, t^+) , denoted $e_1[t^-, t^+]e_2$, where $e_1, e_2 \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $t^-, t^+ \in \mathbb{T}$.

A time constraint $e_1[t^-, t^+]e_2$ is said satisfied by a couple of events $((e, t), (e', t'))$, $e \leq_{\mathbb{E}} e'$ iff $e = e_1$, $e' = e_2$ and $t' - t \in [t^-, t^+]$.

We say that $e_1[a, b]e_2 \subseteq e'_1[a', b']e'_2$ iff $[a, b] \subseteq [a', b']$.

The extraction of temporal constraints between the events of a sequence leads us to define the concept of chronicles (Dousson and Duong, 1999).

Definition 16 (Chronicle). A chronicle is a pair $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ such that:

1. $\mathcal{E} = \{e_1 \dots e_n\}$, where $\forall i, e_i \in \mathcal{E}$ and $e_i \leq_{\mathbb{E}} e_{i+1}$,
2. $\mathcal{T} = \{t_{ij}\}_{1 \leq i < j \leq |\mathcal{E}|}$ is a set of temporal constraints on \mathcal{E} such that for all pairs (i, j) satisfying $i < j$, t_{ij} is denoted by $e_i[t_{ij}^-, t_{ij}^+]e_j$.

\mathcal{E} is called the episode of \mathcal{C} , according to the definition of episode's discovery in sequences (Mannila, Toivonen, and Verkamo, 1997).

The relevance of a chronicle is based essentially on the value of its support. The support of a chronicle refers to the number of its occurrences in a sequence. It can therefore be formalized by the definition below.

Definition 17 (Chronicle support). An occurrence of a chronicle \mathcal{C} in a sequence S is a set $(e_1, t_1) \dots (e_n, t_n)$ of events of the sequence S that satisfies all temporal constraints defined in \mathcal{C} . The support of a chronicle \mathcal{C} in the sequence S is the number of its occurrences in S .

Example 6. Let us illustrate all these basic definitions. Assuming a sequence S of three events $\langle abc \rangle$ represented as follows:

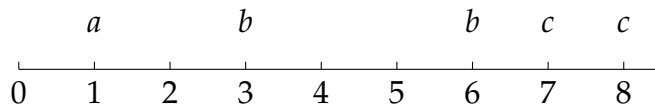


FIGURE 2.1: Example of timestamped sequence

Time constraints that describe the pattern S are noted by $a[2, 5]b$, $b[1, 4]c$ and $a[6, 7]c$.

After the generation of temporal constraints, these events can be represented as a graph, as shown in Figure 2.2.

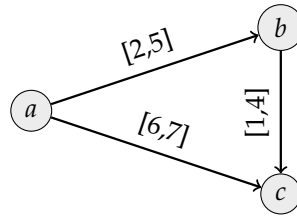


FIGURE 2.2: Example of a chronicle

2.2.2 Discovering Frequent Chronicles

The concept of frequent chronicles discovery is to mine all chronicles whose occurrences in a trace is greater than or equal to a minimum threshold specified by the user.

Several algorithms have been designed for this process, the first one, called FACE for Frequency Analyser for Chronicle Extraction, is introduced (Dousson and Duong, 1999).

To extract frequent chronicles, the algorithm is based on Apriori (Agrawal, Imieliński, and Swami, 1993). This algorithm is based on the idea that a chronicle is not considered frequent if one of its sub-chronicles is infrequent. Otherwise, if all of its sub-chronicles are frequent, this chronicle is candidate to be frequent. The algorithm computes candidate sets of chronicles that can be frequent, it then computes the frequency of each candidate and keeps only the frequent ones. The pseudo-code of the main algorithm is shown in Algorithm 3.

Algorithm 3 FACE Algorithm

Require: \mathcal{D} : Timestamped sequence dataset, fq_{min} : the minimum threshold.

Ensure: the set of frequent chronicles.

- 1: $\Phi^1 \leftarrow \{(a, t_a), \dots \mid A \in \mathcal{L}, \text{supp}(a) \geq fq_{min}\}$
 - 2: **while** $\Phi^{i-1} \neq \emptyset$ **do**
 - 3: $\Phi_c^i \leftarrow \text{generateCandidate}(\Phi^{i-1})$
 - 4: compute $fq(C)$ for all the chronicles C of Φ_c^i
 - 5: $\Phi^i \leftarrow \{C \in \Phi_c^i \mid \text{supp}(C) \geq fq_{min}\}$
 - 6: $i \leftarrow i + 1$
-

The pseudo-code of the procedure "generateCandidate" is shown in Algorithm 4.

The algorithm FACE relies on using first the unconstrained chronicle which is a chronicle model with no time constraint and no order between events.

Algorithm 4 Candidate generation Algorithm

```

1: procedure GENERATECANDIDATE( $\Phi^{i-1}$ )
2:    $\Psi_c^i \leftarrow \text{generateUnconstrainedCandidates}(\Phi^{i-1})$ 
3:   compute support for all the chronicles  $\mathcal{C}$  of  $\Psi_c^i$ 
4:    $\Psi_c^i \leftarrow \{\mathcal{C} \mid \mathcal{C} \in \Psi_c^i \text{ and } \text{supp}(\mathcal{C}) \geq fq_{min}\}$ 
5:    $\Phi_c^i \leftarrow \text{generateConstrainedCandidates}(\Psi_c^i, \Phi^{i-1})$ 
6:   return( $\Phi_c^i$ )

```

FACE is a levelwise mining algorithm that generated unconstrained chronicles and then computes time constraints. For example, using the same sequence of events as Figure 2.1, $fq_{min} = 2$, $i = 3$, we have the previous set of frequent unconstrained chronicles $\{ab, ac, bc, bb\}$ and the set of events $\Omega = \{a, b, c\}$.

Beginning with ab , adding to this one each element of Ω , one by one, we obtain three candidates $\{aba, abb, abc\}$, we compute the support of each candidate and we keep only frequent ones. We process similarly for ac , bc and bb .

The algorithm begins with the extraction of the frequent time constraints to be used for the candidate generation phase. In this part, the algorithm discovers chronicles of size 2. In addition, for each pair of events, only one time constraint is selected. Subsequently, it combines the chronicles discovered to generate larger chronicles.

To conclude, this algorithm is not complete, since there are chronicles that are not extracted.

- **Complete method**

The algorithm by Cram and Mille (Cram, Mathern, and Mille, 2012) is the first complete chronicle discovery algorithm. It starts by building a temporal constraint database, which must be given as input of the main algorithm. A constraint database is said to be complete if all the frequent temporal constraints are stored. HCDA (Heuristic Chronicle Discovery Algorithm) is a generalization of Dousson and Duong (Dousson and Duong, 1999) to extract the complete set of chronicles. The main difference between the two approaches is the selection of frequent temporal constraints. In HCDA, all possible frequent temporal constraints are used to enumerate the frequent chronicles. It is based on a breadth-first search strategy as shown in the example of Figure 2.3. At first step the whole set of frequent temporal constraint is extracted. This set corresponds to the set of chronicles with two events. The

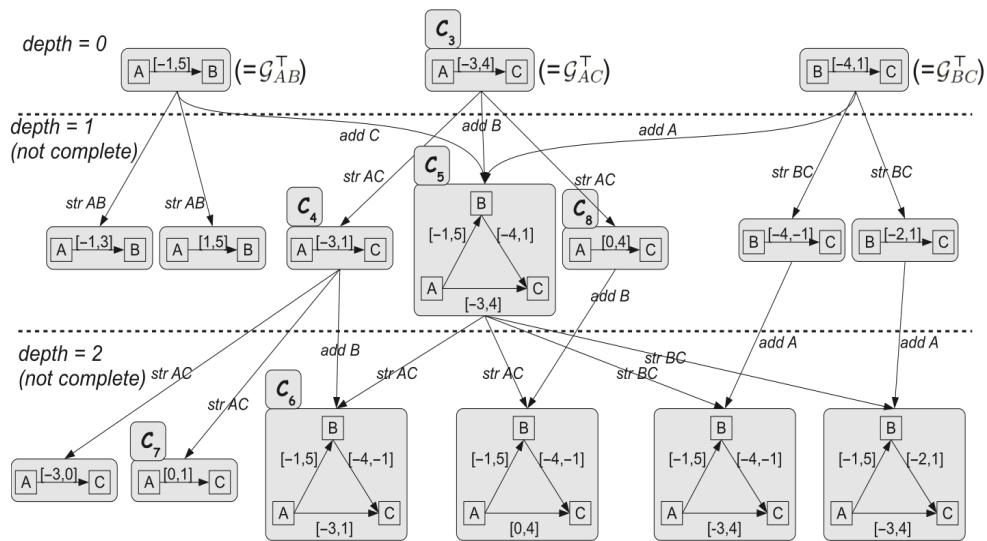


FIGURE 2.3: The chronicle pattern-growth step as illustrated in (Cram, Mathern, and Mille, 2012)

candidate chronicles are then generated by adding new temporal constraints to a frequent chronicle before its frequency test.

ASTPminer (Álvarez, Félix, and Cariñena, 2013) is an apriori based algorithm that extracts the complete set of frequent chronicles from a sequence set. The datasets on which this algorithm is used are obtained from polysomnography tests in patients with sleep apnea–hypopnea syndromes. Contrary to the Huang et al. (Huang, Lu, and Duan, 2012) approach, a same event can occur several times in a sequence. This implies a big time complexity. To tackle this, a window size constraint of 80 seconds, i.e. the constraint that all events of a chronicle occur in a window of 80 seconds, is used to reduce the size of the extracted pattern set.

Data Mining can be defined as the process of analyzing large volumes of data to derive useful insights from it that can help businesses solve problems, seize new opportunities, and mitigate risks. It can be leveraged to answer business questions that were traditionally considered to be too time-consuming to resolve manually.

It is the process of finding patterns in large volumes of data to translate them into valuable information. Data Mining Tools help you get comprehensive Business Intelligence, plan company decisions, and substantially reduce expenses.

Data mining and especially sequential pattern mining were the core challenge of the HALFBACK project. The HALFBACK project aimed to support SMEs in the production process and reduce machine downtime under the so-called topic of predictive maintenance. To do so, a software prototype will be developed during the project which mines data, predicts and optimizes maintenance planning using the machine's own sensor data.

The objective of the next section is to conduct a systematic literature review that provides an overview of the current state of research concerning predictive maintenance including data mining perspective.

2.3 Predictive Maintenance and Applications

In industry, the predictive maintenance task relies on the monitoring of measurable system diagnostic parameters, which identifies the state of a system (Grall et al., 2002). According to the current state of a machine, if any fault or failure exists, a diagnosis can be launched to determine the causes of the fault or failure. Also, the study of the premises characteristic of a failure, analysis about their evolution can be performed. In this way, machine or mechanical system deterioration tendency and the location of a failure can be predicted. Based on the prediction results, maintenance decisions, such as calling the intervention of a machine operator are proposed according to the severity of anomalies, to prevent the halt of the production lines and to minimize economic loss. The use of predictive maintenance techniques has several advantages, such as improved machine availability, improved production efficiency, and reduced maintenance cost (Widodo and Yang, 2007; Rao, 1996).

2.3.1 Physical models

Physical models use explicit mathematical representation to formalize the physical understanding of a degrading machine or equipment (Pecht, 2010). This type of models addresses a predictive maintenance task by solving a deterministic equation or a set of equations derived from extensive empirical data (Sikorska, Hodkiewicz, and Ma, 2011). By using physical models, predictive maintenance of a physical asset is achieved through the interpretation of the acquired knowledge of a manufacturing process, and through the analysis of possible hazards that may cause a failure. Normally, the main steps of a physical model-based predictive maintenance approach include Failure

Modes and Effects Analysis (FMEA), feature extraction, and remaining useful life (RUL) estimation (Pecht, 2009).

The advantages of physical models are their capability of direct incorporation of existing physical mechanisms that have been proved and well-understood by extensive and exhaustive empirical testing (e.g. Paris–Ergo-dan crack propagation laws (Paris and Erdogan, 1963)). As physical understanding of the system improves, the model can be adapted to improve its accuracy. Consequently, when available and sufficiently complete, behavioural models tend to significantly outperform other types of models (Sikorska, Hodkiewicz, and Ma, 2011). Furthermore, changes in the model outputs are described by the residuals (the differences between the reality and the model), which normally have a direct and translatable physical meaning. This eases the interpretation of the outputs of a physical model (Sikorska, Hodkiewicz, and Ma, 2011). Because of these advantages, physical models have been widely used in the predictive maintenance of different physical assets, such as air vehicles (Roemer, Nwadiogbu, and Bloor, 2001), turbine engines (Heng et al., 2009; Sikorska, Hodkiewicz, and Ma, 2011), and sensor default in aircraft military systems (Efthymiou et al., 2012), etc.

The main disadvantage of a physical model is the difficulty of assigning appropriate parameters used in the model. Since the massive and multivariate data required for the assignment of parameters are usually not available, it is hard to quantitatively characterize the system behavior (Sikorska, Hodkiewicz, and Ma, 2011). Also, since fault and failure mechanisms may vary from one equipment to another, physical models are normally equipment-dependent. Thus, it is hard to identify the fault and failure mechanisms of a newly monitored machine component/machine without interrupting system operation (Heng et al., 2009).

2.3.2 Data-driven models

In recent years, rapid advances have been made in the research of data-driven models and approaches for accurate predictive maintenance. Several models perform like a black box that learn the behavior of physical assets directly from their operation data (Javed, Gouriveau, and Zerhouni, 2017). Within a data-driven approach, knowledge about machines is extracted internally from machine operation data, instead of externally from domain experts. Normally, data-driven approaches are classified into machine learning and statistical learning techniques (Dragomir et al., 2009; Peng, Dong, and Zuo, 2010).

2.3.2.1 Machine learning-based approaches

Recently, machine learning (ML) approaches have been proven to be effective solutions for predictive maintenance. The implementation of ML models has been facilitated by the growing capabilities of hardware, cloud computing, and newly introduced state-of-the-art algorithms (Susto et al., 2014).

For supervised and semi-supervised learning, depending on the characteristics of the training data, the learning process of ML-based predictive maintenance could be done using several models (Javed, Gouriveau, and Zerhouni, 2017).

- Supervised learning which is applied to labeled data. This type of methods aims to learn a function that maps an input to an output based on a set of labeled training examples. Typical learning algorithms for supervised learning are Support Vector Machine (SVM) (Scholkopf and Smola, 2001), Artificial Neural Networks (ANN) (Krenek et al., 2016; Javed et al., 2011), Logistic Regression (Kleinbaum et al., 2002), Naive Bayes (Rish et al., 2001), Random Forests (Liaw, Wiener, et al., 2002), and Decision Trees (Quinlan, 1986), etc.
- Semi-supervised learning applied to both labeled and unlabeled data. This type of learning methods is a learning paradigm concerned with the study of how computers and natural systems such as humans learn in the presence of both labeled and unlabeled data (Zhu and Goldberg, 2009). In addition to unlabeled data, semi-supervised learning algorithms are provided with a certain level of supervision, but not for all the training examples. Semi-supervised learning requires less human effort than supervised learning, meanwhile giving higher accuracy than unsupervised learning (Zhu and Goldberg, 2009). Because of these advantages, it has been pervasively implemented in real-world practices. Typical techniques used in this field are Self-Training algorithms (Rosenberg, Hebert, and Schneiderman, 2005), Generative Models (Kingma et al., 2014), Semi-supervised Support Vector Machines (Bennett and Demiriz, 1999), Graph-Based Algorithms (Bryant, 1986), and Multi-view Algorithms (Bickel and Scheffer, 2004), etc.

Facilitated by the capabilities of the advanced techniques and state-of-the-art algorithms, machine learning-based approaches have shown to be promising and effective solutions for industrial predictive maintenance. The algorithms and techniques introduced in this subsection have been widely

used in real-world practices to reduce the maintenance cost and production downtime (Susto et al., 2014).

2.3.2.2 Data mining-based approaches

Unsupervised learning is applied to unlabeled data. In unsupervised learning, the training data is not associated with any corresponding target values. In other words, no labels are given to the learning algorithm, leaving it on its own to discover patterns and structure from the input data. The objective of an unsupervised learning process is to segregate data points with similar traits and assign them into clusters, which is known as *clustering*, or to construct an estimate of the distribution of the input data, which is known as *density estimation*. Common unsupervised learning methods include k-means Clustering (Likas, Vlassis, and Verbeek, 2003), Gaussian Mixture Models (Reynolds, 2015), Self-organizing Maps (Kohonen, 1997), etc. In Rabatel et al. (Rabatel, Bringay, and Poncelet, 2011), a knowledge discovery solution is presented to extract data from historical behavioral data collected by sensors. It is based on association rules, more specifically sequential pattern mining, to extract specialized classes. Using anomaly detection, they compare new patterns with sequential patterns describing normal behavior that were extracted before. In Remil et al. (Remil et al., 2021), the authors present an efficient incident triage model based on 170k incidents reported on more than a thousand application servers over the last 7 years. Artzmüller (Atzmueller, 2015) has provided a review of subgroup discovery techniques and their application. The discovery of subgroups aims to group the predicted objects into subgroups that support the same explanation. Each subgroup is also associated with a description that characterizes it in a unique way. This approach addresses in a new way the problem of explaining the results of the model, in particular when the number of results to explain is large. Because of the low cost of deployment and better applicability than physical models, data-driven models have been widely applied to industrial predictive maintenance. However, the main disadvantage of this type of model is their demand for higher volume of data than physical models. To well train a data-driven model, sufficient run-to-failure data ("sufficient" quantity means that data have been observed for all fault modes of interest (Uckun, Goebel, and Lucas, 2008)) needs to be collected for the model to capture complex relations among data (Javed, Gouriveau, and Zerhouni, 2017). This means a large amount of machine historical data needs to be collected as starting step for obtaining high-quality and accurate data-driven models.

2.3.2.3 Statistical learning-based approaches

Within this kind of approach, predictive maintenance is achieved by fitting the empirical model (a function) as close as possible to the collected data and extrapolating the fitted curve to failure criteria (Javed, Gouriveau, and Zerhouni, 2017). A typical statistical model is a regression method for trend extrapolation, which is based on linear, exponential, or logarithmic functions. The common methods used in statistical learning-based approaches are (i) Stochastic filtering (Kallianpur, 2013); (ii) Particle filters and their variants (Gustafsson et al., 2002); (iii) Hidden Markov models (Beal, Ghahramani, and Rasmussen, 2002); iv) Time series analysis (Hamilton, 1994).

2.3.3 Hybrid models

A hybrid model applies both physics-based and data-driven approaches. As mentioned in the previous subsection, data-driven models are feasible to be used when the required big data is easy to collect. However, it is usually the case that only part of machine historical data can be obtained (Wang, 2016). In this context, the data-driven approach is jointly used with the physics-based approach for effectively identifying machine conditions. Existing hybrid model-based predictive maintenance approaches can be classified into two types: (i) Series approaches and (ii) Parallel approaches.

In series approaches, physical models are combined with online parameter estimation techniques to update model parameters when new data are available (Javed, Gouriveau, and Zerhouni, 2017). Data-driven methods are used to tune the parameters of physical models. This type of approaches have been applied to the predictive maintenance of various equipments, such as power Metal-Oxide-Semiconductor Field-Effect Transistors (MOSFETs) Celaya et al. (Celaya et al., 2011), printed circuit card assemblies (Pecht, 2010), and Lithium-ion (Li-ion) batteries (Saxena et al., 2012), etc.

Within parallel approaches, data-driven models are trained to predict the residuals not explained by the first principle model (Javed, Gouriveau, and Zerhouni, 2017). Normally, the hybrid model used in a parallel approach is created with an individual approach, which is either physics-based or data-driven. Therefore, the accuracy of a parallel hybrid model is normally higher than a series hybrid model (Javed, Gouriveau, and Zerhouni, 2017). However, implementing a parallel hybrid model requires several steps, which

leads to a higher modeling complexity than series approaches-based models. Because of this, parallel hybrid models consume more computational time than series hybrid models.

In general, the different steps for implementing a parallel hybrid model for predictive maintenance are: parameter identification, condition monitoring, feature extraction, healthy baseline creation, anomaly detection, parameter isolation, failure definition, parameter trending, and RUL estimation (Cheng and Pecht, 2009).

2.4 Conclusion

Predictive maintenance is a key technique implemented in smart factories to improve the availability, reliability, and productivity of manufacturing systems. In this chapter, we have reviewed the existing sequential pattern mining techniques. We shed the light on a special pattern called chronicle mining that contains both events and time duration. We detailed as well approaches for industrial predictive maintenance by classifying them into four categories: knowledge-based models, physical models, data-driven models, and hybrid models. The classification was followed by the demonstration of the advantages and disadvantages of the existing models.

Chapter 3

Text Mining Application to TRIZ Inventive Design Modelling

Contents

Introduction	58
4.1 Mining Failure Chronicles for Predictive Maintenance . .	58
4.1.1 CPM Algorithm	59
4.1.2 Clasp-CPM	60
4.2 Failure Detection with Chronicles	65
4.2.1 Maximum Confidence-based approach	65
4.2.2 The algorithm for rule base integration with detec- tion of the three issues	67
4.2.3 Chronicle fusion approach	69
4.3 Experiments	71
4.3.1 The SECOM Data Set	71
4.3.2 Evaluation of the chronicle mining	72
4.3.3 Predictive maintenance analysis	75
4.4 Conclusion	78

Introduction

Scientists have been working on understanding natural languages since the turn of the last century and, as of today, we have reached reasonable success in this area. The research on how to make computers analyze and manipulate natural languages draws from several fields, including computer science, math, linguistics, and neuroscience, and the resulting interdisciplinary area of research is called Natural Language Processing (NLP).

Text mining, is a sub-field of NLP domain, that refers to the process of identifying, enumerating, and analyzing syntactic and semantic characteristics of a text corpus. The purpose of text mining is to bring to light unknown facts, characteristics, patterns and ultimately to lead to a better identification of new text insight. It is one of the main disciplines of Artificial Intelligence. The data can have a constant structure. In this case we speak of "structured data". If it is not the case, we speak of unstructured data. This chapter presents related research works about NLP and the text mining field, including several approaches about the semantic textual similarity computation, natural language understanding, and TRIZ-related applications.

3.1 Representation Learning on Natural Language Processing

Any text mining task relies on several main steps. As illustrated in Figure 3.1 (Chen et al., 2021), the different processing stages of NLP are presented. A data mining pipeline generally consists of several steps from data pre-processing to a choice of model architecture and results analysis. Textual data must be transformed into numeric data to be processed by a computer. First, the tokenization splits the input document into words. Vector representations of the documents or words, called *embeddings*, can then be built. The embeddings may finally be used to mine the targeted information.

Over the past years, deep learning has seen huge advances in several application fields such as computer vision or speech recognition. Deep learning has also been extensively used in the NLP domain. The adjective "deep" in deep learning refers to the use of multiple layers neural networks topology¹. Deep neural networks link predictions to data inputs through sequential non-linear variation. In this chapter, we focus on deep learning models

¹Basics of neural networks are given in Appendix C

widely used in NLP tasks.

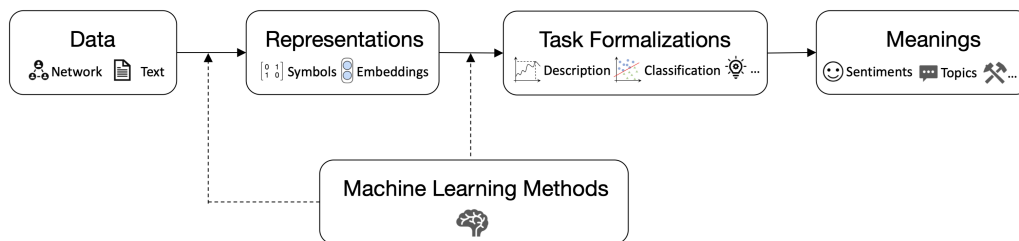


FIGURE 3.1: The generic process of NLP (Chen et al., 2021)

Natural Language Processing, also known as computational linguistics, involves the engineering of computational models and programs to solve the practical problems of human language analysis (Otter, Medina, and Kalita, 2020).

Moreover, NLP involves teaching machines to interpret, classify, manipulate, and generate language. From the early use of handwritten rules and statistical techniques to the recent adoption of deep learning, the NLP domain has provided several models with a wide range of application fields (Zhang et al., 2021).

Over the past few years, with the growing computing power and large-scale text data, distributed representation trained with neural networks and large volumes of data has become the mainstream (Liu, Lin, and Sun, 2020). In detail, distributed representation is mainly driven by the fact that the representation of any single concept is distributed over many, if not all, processing units (Deng and Yu, 2013).

Several text representation models have been introduced for the NLP. These models could be assigned to one of these two families: *contextual* and *non contextual*. In what follows, we summarize a few models commonly used in NLP tasks, and they are detailed in subsection 3.1.1:

- **N-gram Model** (Brown et al., 1992): it predicts the next item in a sequence based on its previous $n-1$ items.
- **Bag-of-Words** (Harris, 1954): a bag of words is a representation of text that describes the occurrence of words within a document.
- **Distributed Representation** (Hinton et al., 1986): a distributed representation occurs when some concept or meaning is represented by the network, but that meaning is represented by a pattern of activity across a number of processing units.

- **Neural Probabilistic Language Model** (Bengio et al., 2003): it learns a joint probability function of a sequence of words given a context.
- **Pre-trained Language Model** (Edunov, Baevski, and Auli, 2019): it includes contextual words representation, the novel pre-training-fine-tuning pipeline, larger corpora, and deeper neural architectures.

3.1.1 Non contextual embedding

3.1.1.1 N-gram Model

N-gram is one of the oldest techniques for word embedding. N-grams could be sequences of characters or words extracted from text (Majumder, Mitra, and Chaudhuri, 2002). It is used to predict a word from previous analyzed words (Brown et al., 1992). It relies on assuming that linguistic items with similar distributions have similar meanings (Harris, 1954). It has led to the emergence of new NLP models such as Word2vec (Mikolov et al., 2013b) and BERT (Devlin et al., 2018). N-grams can be classified into two families: character-based and word-based.

- **Character N-gram** is a collection of n consecutive characters extracted from a word. The main motivation behind this approach is that similar words contain a high percentage of shared N-grams. Typical values for n are 2 or 3. These values respectively correspond to the use of bigrams or trigrams, respectively. For instance, for the word "car", it results in the generation of the bigrams as $*c$, ca , ar , $r*$ and trigrams as $**c$, $*ca$, car , $ar*$, $r**$.
- **Word N-grams** are sequences of n consecutive words extracted from a text. Word-level N-gram models are robust for statistical modeling of language as well as for information retrieval and are not very language-dependent. For instance, for the sentence of "Car is cleaned by Tony.", a created vocabulary set by bi-gram is ["car is", "is cleaned", ... , "by Tony"].

3.1.1.2 Bag-of-Words

The Bag-of-Words model (BOW) is an orderless documentary representation, which is built on the distribution hypothesis (Harris, 1954). The distributional relation about elements' occurrence between the correlation of several aspects of meaning is revealed. The BOW only focuses on words occurrence

All Words		
[“Hi”, “How”, “are”, “you”, “bye”, “see”, “later”]		
“Hi”	→	[1, 0, 0, 0, 0, 0, 0]
“How are you”	→	[0, 1, 1, 1, 0, 0, 0]
“Bye”	→	[0, 0, 0, 0, 1, 0, 0]
“See you later”	→	[0, 0, 0, 1, 0, 1, 1]

FIGURE 3.2: An example of Bag-of-Words

without checking their positions. In this approach, word occurrence is the main feature (Goldberg, 2017).

As a special N-gram model where $n = 1$, it ignores the text’s syntax and word order and sees the text as a combination of several individual words. The occurrence of each word in the text is independent (Soumya George and Joseph, 2014). A bag-of-word vector is produced to represent texts. The BOW first designs a vocabulary of words using every word in the corpus. It maps then the text to a bag-of-word vector where co-occurrence words with the vocabulary are shown as “1” and the inverse as “0” (Ni, Samet, and Cavallucci, 2021). “1” is added each time the word is repeated in the document. For instance, as illustrated in Figure 3.2, the co-occurrence words in bag-of-words are presented as “1” in the corresponding vector. After that, the text is converted into fixed-length vectors of numbers.

3.1.1.3 Distributed Representation

The name “distributed representation” is mainly driven by the fact that the representation of any single concept is distributed over many, if not all, processing units. In many cases, the unit values in the vectors are continuous values, instead of just 1’s and 0’s. Distributed representations for words were first proposed by Hinton et al. (Hinton et al., 1986).

The significant performance of distributed representations in the NLP field has encouraged the resurgence of several works since Mikolov et al. paper in 2012 (Mikolov et al., 2012). It is used in several NLP applications such as word representation, named entity recognition, word sense disambiguation, parsing, tagging, and machine translation (Collobert and Weston,

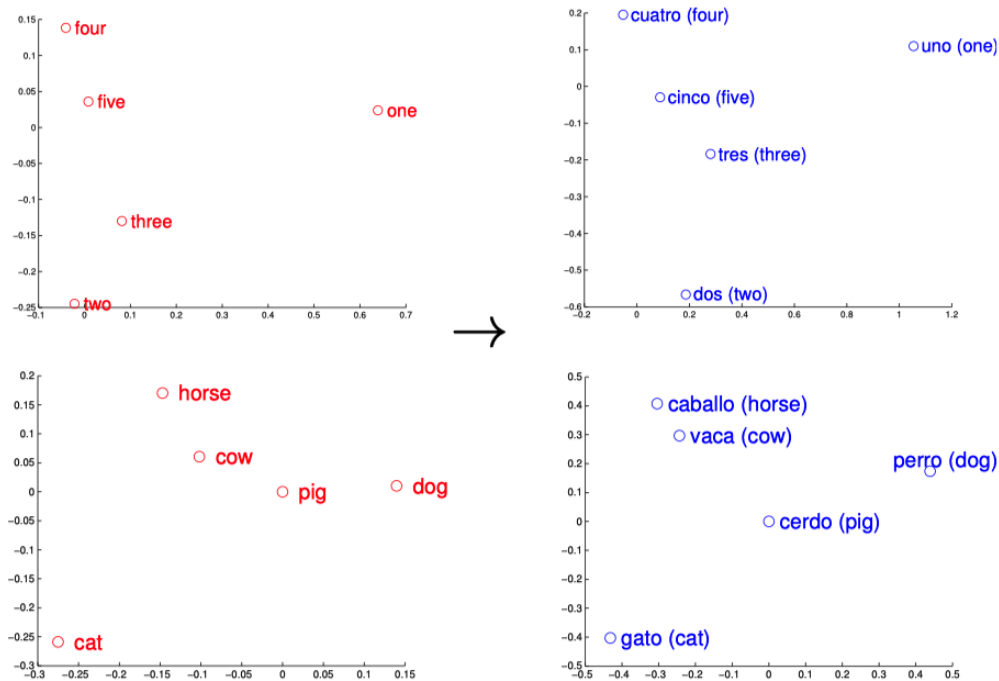


FIGURE 3.3: An example for the distributed representation of words

2008; Turney and Pantel, 2010; Turian, Ratinov, and Bengio, 2010; Collobert et al., 2011; Socher et al., 2011; Zou et al., 2013; Huang et al., 2012). Word2Vec (Mikolov et al., 2013a), surely an iconic model using distributed representation, is a two-layer neural network that can be trained. Training can be done by a given corpus to convert each unique word presentation in the corpus into a computable and structured vector in an a priori chosen space dimension. A word vector is positioned in the vector space so that words sharing similar contexts in the corpus are located close to one another in the space (Mikolov et al., 2013b). Words with similar meanings get a similar representation. For instance, as illustrated in Figure 3.3 (Mikolov, Le, and Sutskever, 2013), with the distributed word representations, five word vectors in English (left) and Spanish (right) are projected down to two dimensions.

3.1.1.4 Neural Probabilistic Language Model

As one of the pioneer practices of distributed representation in NLP (Liu, Lin, and Sun, 2020), the neural probabilistic language model is meant to learn the joint probability function of sequences of words in a language (Bengio et al., 2003).

A computationally efficient probabilistic modeling approach is proposed by Bengio et al. (Bengio et al., 2003) to figure out the issue of the curse of dimensionality, which is that the joint distribution of a large number of discrete variables results in exponentially large parameters. It allows each training sentence to provide the model with an exponential number of semantically adjacent sentences. The model is able to learn a distributed representation of each word and a probability function for the sequence of words, which are represented in terms of these representations, and word embeddings (i.e., low-dimensional word vectors) are brought by it as learned parameters. Semantic meanings of words are indeed encoded by these vectors. Several typical models including Word2Vec (Mikolov et al., 2013b), Glove (Pennington, Socher, and Manning, 2014), and FastText (Bojanowski et al., 2017) are inspired by the neural probabilistic language model to embed words into distributed representations to optimize them as model parameters. It eventually makes it possible to take advantage of longer contexts and significantly improve the model's performance compared to N-gram models.

3.1.2 Contextual embeddings models

Contextual embeddings models move beyond global word representations like Word2Vec and achieve ground-breaking performance on a wide range of natural language processing tasks. Contextual embeddings assign each word a representation based on its context, thereby capturing uses of words across varied contexts and encoding knowledge that transfers across languages. These models are usually generated using neural encoders. Several neural network-based architectures have been used: Recurrent Neural Networks (RNNs) (Mikolov et al., 2010), Transformers and Convolutional Neural Networks (CNNs) (Gu et al., 2018).

3.1.2.1 Recurrent networks-based encoder

Recurrent Neural Networks have signals traveling in both directions by using feedback loops in the network. Features derived from earlier input are fed back into the network which gives them an ability to memorize. Recurrent networks are characterised by a temporal link between the network's outputs h_t and inputs x_t . Indeed, hidden states (the network's memory cells) are updated with respect to the information flowing from the input to the output. Information is therefore "saved" and can be reused later, i.e. for future predictions. Recurrent neural networks are particularly relevant for

NLP tasks as they establish the link between words that appear at different positions. As shown in Figure 3.4, a simplified way of representing the Recurrent Neural Network is by unfolding/unrolling the RNN over the input sequence. For example, if we feed a sentence as input to the Recurrent Neural Network that has 10 words, the network would be unfolded such that it has 10 neural network layers.

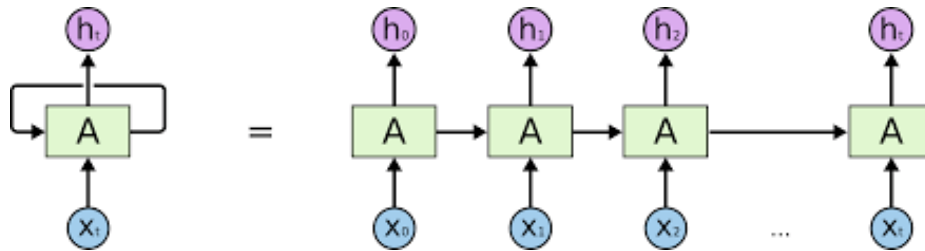


FIGURE 3.4: The architecture of a Recurrent Neural Network and its equivalent unfolded over a sequence.

The most famous RNN is the Long Short Term Memory (LSTM). The internal structure of an LSTM contains "gates" which control the information in the hidden states. The memory vector flows in the upper part of the cell in Figure 3.5. The network's input X_t and the previous network's output h_{t-1} flow in the lower part in the cell. The first block represented in the LSTM unit, shown in Figure 3.6, is the forget gate f_t . The information from the current input X_t and the previous hidden state h_t is passed through the sigmoid activation function (σ in Figure 3.6). If the output value is closer to 0 means forget, and the closer to 1 means to retain. The input gate works as an input to the cell state. It consists of two parts; first, we pass the previous hidden state h_t and current input X_t into a sigmoid function to decide which values will be updated. Then, we pass the same two inputs into the tanh activation to regulate the network. Finally, multiply the tanh output with the sigmoid output to decide which information is important to update the cell state. The hidden state contains information on previous inputs and is used for prediction. The output gate regulates the present hidden state h_t . The previous hidden state h_{t-1} and current input X_t are passed to the sigmoid function. This output is multiplied with the output of the tanh function to obtain the present hidden state. The current state (C_t) and present hidden state h_t are the final outputs from a classic LSTM unit. Formally, the cell state is modified at each inference (Equation 3.1), to take into account the previous state and the input.

$$C_t = f_t \odot C_{t-1} + i_t \odot g_t \quad (3.1)$$

f_t is the forget gate that will select the information to keep in the cell state, i_t is the input gate (weighted formula depending of X_t) that will select the information to add in the cell state given the input, and g_t also depends on the input and the previous output. \odot is the Hadamard product.

In the case of a text analysis, the words are successively fed to the network. It means that the network is only able to analyse the previous words to find the context. A LSTM network is therefore unidirectional. It works either from left to right words or from right to left but it cannot analyse both right and left contexts. To address this problem, two LSTM networks may be stacked, the first analyses the context from left to right and the other one from right to left.

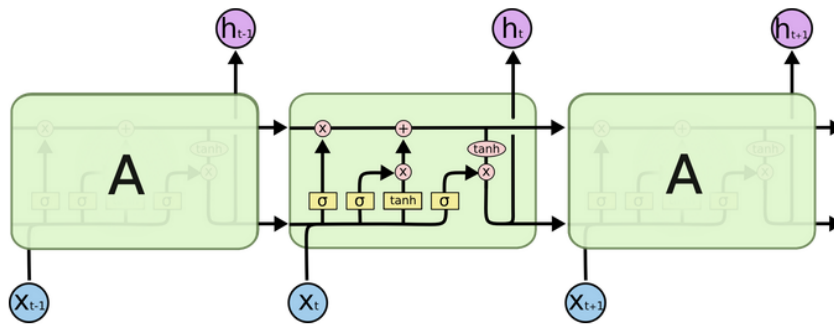


FIGURE 3.5: Architecture of a Long Short Term Memory neural network

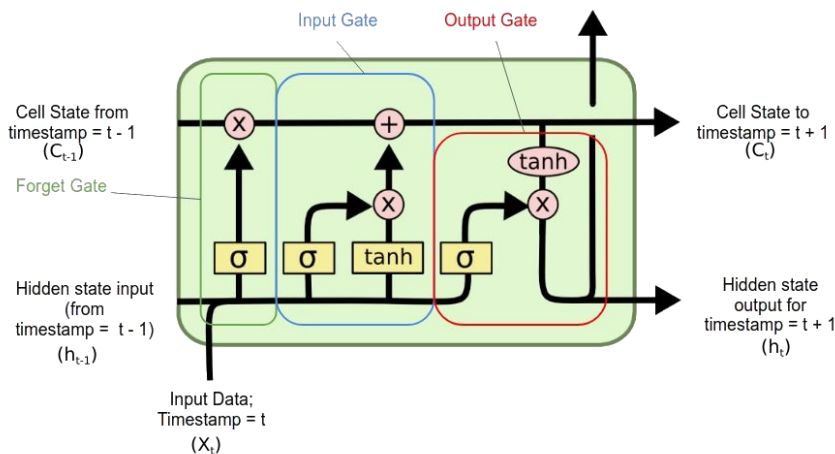


FIGURE 3.6: Architecture of a Long Short Term Memory unit

A Gated Recurrent Unit (GRU) network is very similar to a LSTM. The main difference is the fusion between the hidden states and the output in the GRU network while in a LSTM the output is different from the hidden states.

To build the sentence and the word representations, the hidden states have to be concatenated.

Recurrent Networks show good performance for text analysis and text summarization. They work also very well for generation tasks. The words are predicted one after the other. Nevertheless the main limitation of recurrent networks is the flow of information. To analyze a sentence, the network updates the hidden states at each time step, i.e. for each word. It means that the information coming from the first words is drowned under other words information (Koehn, 2017). Therefore the quality of the transferred information decreases relatively fast. Moreover, research was done to design a full bidirectional network instead of stacking two unidirectional networks. This has led to the development of Transformers and pre-trained encoders.

3.1.2.2 Transformers

Transformers and LSTMs are both popular techniques used in the field of NLP and sequence-to-sequence modeling tasks. Transformers make use of the attention mechanism that enables them to process and capture crucial aspects of the input data. They do this without relying on recurrent neural networks (RNNs) like LSTMs or gated recurrent units (GRUs). This allows for parallel processing, resulting in faster training times compared to sequential approaches in RNNs. The architecture of Transformers typically consists of stacked encoder and decoder layers, with self-attention and feed-forward neural network (FNN) layers in each. The absence of RNN cells contributes to their parallel processing capabilities. Encoders may be designed to be trainable in a non-supervised way. This is especially the case for Transformers networks (Vaswani et al., 2017b) shown in Figure 3.7. They can then be trained on all sort of documents like Wikipedia pages, articles, journals, etc. These models are largely used due to the lack of summarization database. With a small database, it is not possible to learn the encoder and the decoder in the same time without over-fitting and under-performance.

Transformer Networks are interesting for language understanding and language generation because of their bidirectionality. They are composed of several stacked Attention Mechanisms. An Attention Mechanism is designed to select the best information from a flow of data. For instance, it can be used to determine which encoded information is important for the decoder. There are two different Attention Mechanisms: Global Attention where all the information is weighted (according to their relevance) and summed up and

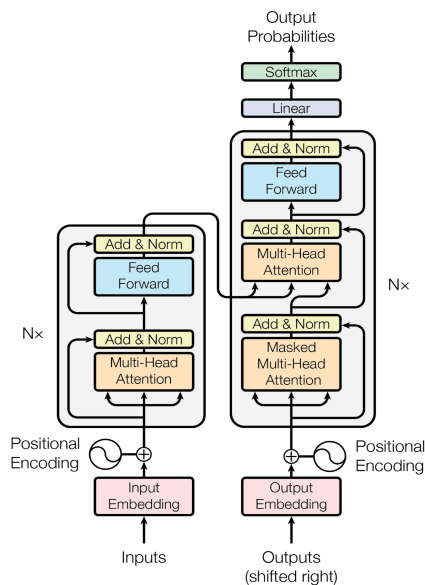


FIGURE 3.7: The encoder-decoder structure of the Transformer architecture Taken from (Vaswani et al., 2017a)

Local Attention where only a part of the information is taken into account to reduce the number of computations, for example.

The stacked Attention Mechanisms are composed of three inputs (see Figure 3.8): the Keys (K), Queries (Q) and Values (V). The Queries represent the information needed by the network. The Keys are the actual information flowing in the network. A dot-product between the Queries and Keys therefore highlights the information that will be selected in the Values with a simple multiplication. Indeed, Values also contain the flowing information in the network. When the Keys, Values and Queries contain the same information, it is called self-Attention. These mechanisms enable the network to study the internal structure of a text.

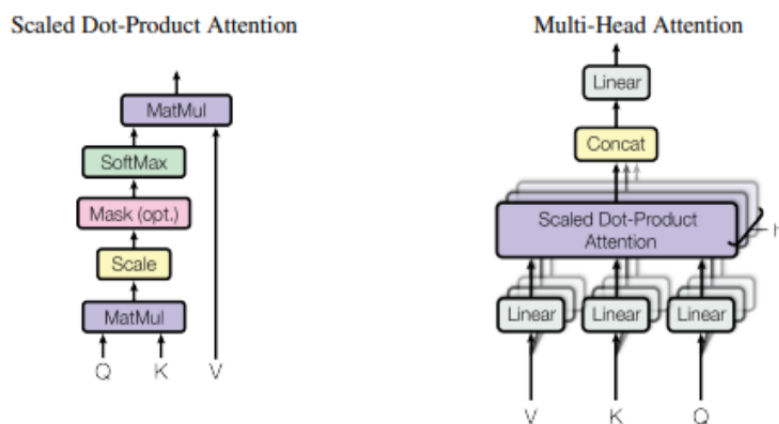


FIGURE 3.8: BERT's Attention Mechanism (Devlin et al., 2019)

If the encoder is already pre-trained on another database, then a small database may be enough to a downstream task without overfitting. These encoders are therefore trained on very large databases (billions of words) and their performances are beyond reach. BERT (Devlin et al., 2019) (Figure 3.9) is the most famous pretrained-encoder. BERT makes use of Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. The input of BERT is a sequence of tokens (i.e words),

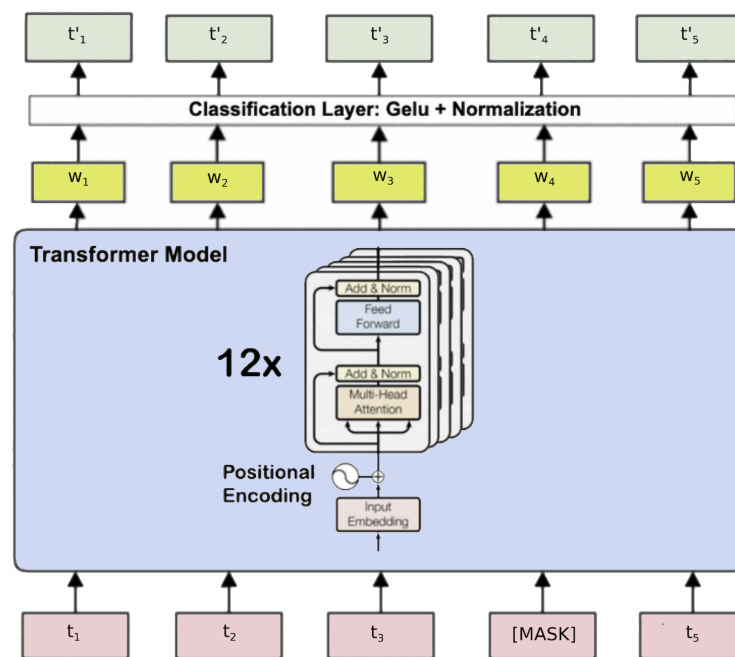


FIGURE 3.9: BERT architecture based on Transformers with twelve encoder blocks.

which are first embedded into vectors and then processed in the neural network. The first token of every sequence is always a special classification token ([CLS]). [SEP] is a special token to separate between two sentences. Before feeding word sequences into BERT, 15% of the words in each sequence are replaced with a [MASK] token. The model then attempts to predict the original value of the masked words, based on the context provided by the other, non-masked, words in the sequence. Therefore, to predict the output words, we: (i) add a classification layer on top of the encoder output (W_i in Figure 3.9); (ii) multiply the output vectors by the embedding matrix, transforming them into the vocabulary dimension; (iii) calculating the probability of each word in the vocabulary with softmax.

BERT uses Wordpiece embeddings input for tokens. Along with token embeddings, BERT uses positional embeddings and segment embeddings for each token. Positional embeddings contain information about the position of tokens in sequence. Segment embeddings help when model input has sentence pairs. Tokens of the first sentence will have a pre-defined embedding of 0 whereas tokens of the second sentence will have a pre-defined embedding of 1 as segment embeddings.

Final Embeddings used by model architecture are the sum of token embedding, positional embedding as well as segment embedding. The final embeddings are then fed into the deep bidirectional layers to get output. The output of the BERT is the hidden state vector of pre-defined hidden size corresponding to each token in the input sequence. These hidden states from the last layer of the BERT are then used for various NLP tasks.

BERT encoder shows incredible results and helped establishing the state of the art in a great number of NLP tasks. Nevertheless, it has some disadvantages. The non-supervised learning is based on the prediction of masked input tokens. To predict one of these tokens, the model should rely on all the other tokens but some of these are also masked. Moreover, it does not take into account the dependencies between masked tokens. The input length is also limited to 512 tokens because of the chosen position embedding for tokens and to reduce the size of the model.

In Lan et al. (Lan et al., 2019), some sharing parameters and learning techniques are presented in order to lighten the BERT model and improve the results. But the global limitations of BERT cited above are not resolved.

The XLNet (Yang et al., 2019) model aims at addressing the limitations of the BERT approach and the classical auto-regressive (recurrent networks) limits by mixing the advantages of both techniques: the ability of generation tasks from the recurrent models and the pretraining and bidirectionality of BERT without its learning issues. XLNet is based on a permutation model. The context words are randomly permuted and the network is trained to predict each word, given the previous words that can come from all over the sentence. The model is therefore able to gather information from all positions on both sides. XLNet performs better than BERT in various NLP tasks mostly because it was trained using much more data than BERT as shown in Zhuang et al. (Zhuang et al., 2021) where the authors compare XLNet and BERT performance with an equal amount of training data.

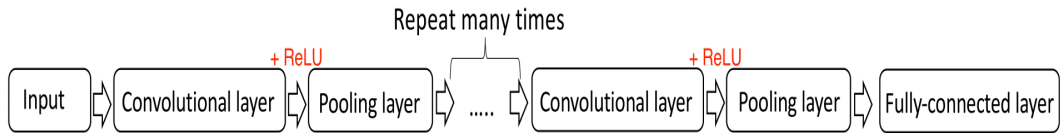


FIGURE 3.10: Basic structure of CNN for NLP

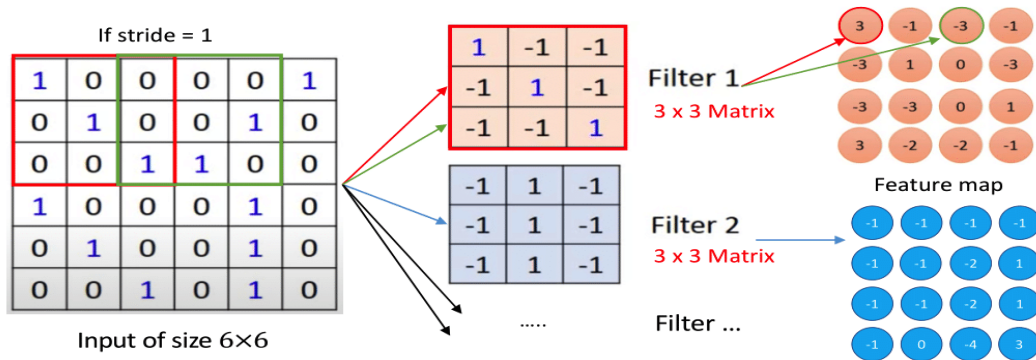


FIGURE 3.11: Basic operational structure of the convolutional layer

3.1.2.3 Convolutional Neural Networks (CNN)

Convolutional networks (Lecun et al., 1998) originate from image processing and are more rarely used in NLP than the two types of networks discussed previously. Nevertheless, they have properties that can also be exploited for language processing. This section presents a brief introduction of the Convolutional Neural Network (CNN) and its main elements whose applications could be highly effective in the field of Natural Language Processing (NLP).

As illustrated in Figure 3.10, a convolutional neural network includes successively an input layer, multiple hidden layers, and an output layer, the input layer being dissimilar according to various applications. The hidden layers, which are the core block of a CNN architecture, consist of a series of convolutional layers, pooling layers, and finally export the output through the fully-connected layer. The convolutional layer is the core building block of a CNN. In short, the input with a specific shape will be abstracted on a feature map after passing the convolutional layer. A set of learnable filters (or kernels) plays an important role throughout this process. Figure 3.11 provides a more intuitive explanation of the convolutional layer. The pooling layer is a concept that can be intuitively understood. The purpose of the pooling layer is to reduce progressively the spatial size of the feature map, which is generated from the previous convolutional layer, and identify important features.

The CNN is often associated with an LSTM type encoder (Zhou et al., 2016). CNN's have a strong ability to extract features locally but are limited when it comes to creating longer term links between several sentences for example. This is why they remain much less used so far in language processing. Recently, Li et al. (Li et al., 2021) showed that it was possible to implement attention mechanisms with convolutional networks.

3.2 Text Mining for Enhancing Innovation Using TRIZ

TRIZ is a human-oriented knowledge-based systematic methodology of inventive problem solving (Savransky, 2000). Originally, it was proposed by Altshuller (Altshuller, 1999) through his analysis of thousands of scientific text documents. The explanation of the definition is as follows:

- **Human-oriented:** As the practice of TRIZ depends on the problem itself and the socio-economic environment, it is human beings, not machines, who set the direction of the heuristic.
- **Knowledge-based :** Knowledge of generic problem solving heuristics is extracted from thousands of patents in the engineering fields. TRIZ uses not only knowledge from the natural and engineering sciences, but also knowledge about the specific problem domain.
- **Systematic :** TRIZ offers an effective application of existing solutions to new problems and the creative process is systematically structured.
- **Inventive problem solving :** The aim of TRIZ is to solve creative problems in which only the main contradiction needs to be solved.

In addition, with the definition of TRIZ, Altshuller proposed three primary findings which are:

- Problems and solutions could be shared among all industries;
- The pattern of technological evolution also repeats itself in different industries;
- The scientific effects used by an innovation go beyond the field in which it was developed.

One of the main theoretical pillars of TRIZ theory for solving inventive problems is contradiction. The subsection below details the definition of contradiction in TRIZ.

3.2.1 Contradiction

The concept of contradiction is one of the two significant parts of TRIZ (together with evolution laws). It is applied in any TRIZ problem-solving process. TRIZ states that in order to obtain creative solutions, contradictions must be eliminated, without allowing for compromise or optimization, and introduces the principle of formulating and eliminating contradictions in a systematic way. There are two typical types of contradictions that are classified (Domb, 1997; Rousselot, Zanni-Merk, and Cavallucci, 2012):

- **Technical contradiction:** A technical contradiction describes the state of a system in which one action causes a useful effect, but also creates an undesirable effect at the same time. It is a typical engineering trade-off when something becomes better but another thing becomes worse. In fact, this occurs when it attempts to improve certain properties or functions of a system but results in the deterioration of other properties in the system (Yan, 2014).
- **Physical contradiction :** A physical contradiction resolves the part of the technical contradiction centered on the parameter, which must contain two opposite values at the same time. This occurs when there are inconsistent requirements for the physical conditions of the same system. For instance, the big size of a laptop screen is able to provide a better watching experience but may make it too heavy to carry around. Therefore, the screen's size values (big screen for watching and small screen for space occupancy) present a physical contradiction.

Figure 3.12 shows a technical contradiction between the volume and the weight. In the example, increasing an aircraft volume for higher number of seats impact negatively the overall weight.

3.2.2 The TRIZ Knowledge Sources

When dealing with technical contradictions, the TRIZ knowledge sources for solving inventive problems include forty inventive principles. Other TRIZ techniques such as seventy-six inventive standards, and eleven separation

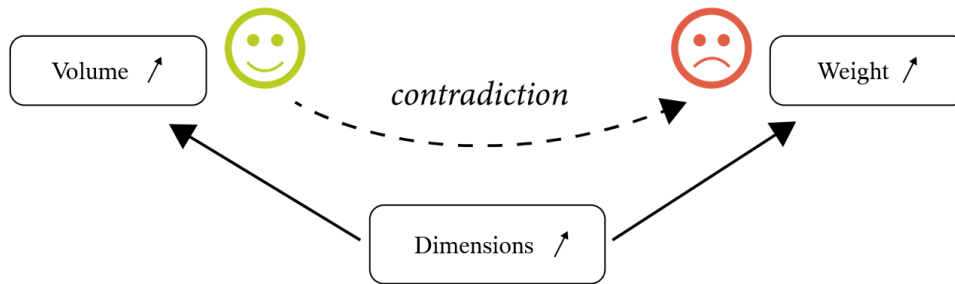


FIGURE 3.12: An example of TRIZ contradiction

methods for removing technical contradictions, provide problem-solving solutions and remove physical contradictions respectively. The full contents of TRIZ matrix, inventive principles and inventive standards are presented in Appendix A.

- Inventive Principles:** Forty inventive principles are a type of tool used to find out innovative and creative solutions for target problems. They derive from the research of TRIZ and patent analysis in order to solve the technical contradictions. Among principles, there is “segmentation”, which proposes breaking down objects into independent parts. This might include manufacturing an object so that it becomes easier to disassemble, or use segmentation to resolve a technical issue. This might be done by using a trailer and truck instead of one large truck or by designing cubicles for an open plan office to enable easy reshuffling of the office layout according to need.
- Standard Solutions:** Most inventions refer to conceptual modifications of physical systems. Therefore, there should be several common approaches for solving problems that apply to the entire group of similar inventive problems. These problems are similar when problems from different fields produce the same physical model. Thus, solving system problems does not always need to identify contradictions. Among solutions, there is "introduce the additive temporarily". An example could be for certain kinds of bone injuries, a metal screw is placed in the bone while healing starts, then removed.

A classical TRIZ problem-solving process can be illustrated in Figure 3.13 (Yan, 2014). In detail, experts should first present the target problem in the form of contradiction by using different tools. After that, they should achieve abstract solutions by using different knowledge. Eventually, with the help of other domains knowledge base like physical, chemical, or geometrical fields,

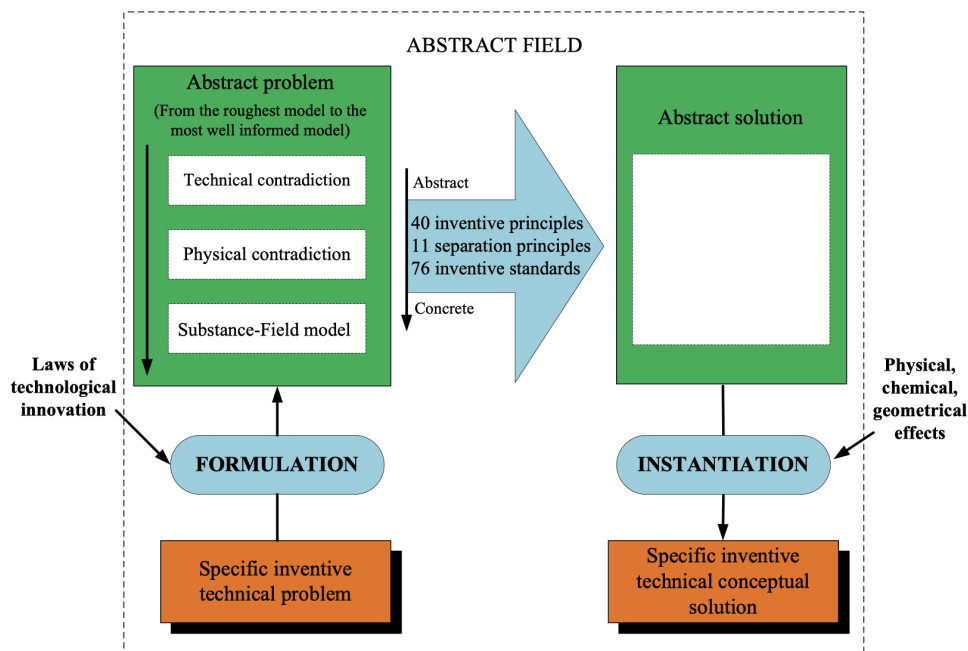


FIGURE 3.13: The processing of solving inventive problems via classical TRIZ

abstract solutions can be instantiated to achieve one or more concept solutions to be implemented in real cases. Altshuller was also interested in the solutioning. So, as with the parameters, he tried to find universal formulations of the ideas leading to solutions. This is what he called the inventive principles. There are 40 of them². The contradictions and the inventive principles allowed him to develop the TRIZ matrix and the inventive problem solving algorithm. The TRIZ matrix is a statistical representation of the inventive principles that were mostly used in the 40,000 inventive patents that were analyzed by Altshuller and the engineers with whom he worked. Each cell corresponds to a contradiction between two evaluation parameters (which are on the abscissa and ordinate). The cells contain, for each contradiction, the indices of the inventive principles that were statistically the most used to solve these contradictions.

3.2.3 Implementation of TRIZ on the Problem Solving

Several models and extensions based on TRIZ have been proposed by researchers. A TRIZ-based patent knowledge management system called TP-KMS is proposed by Ding et al (Ding et al., 2017) to explore the possible inventive principles for solving problems. The structural information of patents,

²Reader may refer to Appendix A for further details about TRIZ matrix

TRIZ theory, and 40 inventive principles are combined to help researchers explore the process of construction innovations. TPKMS requires that users have the corresponding domain knowledge to explore inventive solutions. Moreover, Rahim et al. (Rahim et al., 2018) proposed an approach using the computational thinking model and TRIZ methodology to enhance patent innovativeness. Identifying patterns in the computational thinking model enables the generation of a possible solution. Based on the exploration of trend pattern recognition on the proposed paper, users could explore the inventive solutions among different domains.

Besides, a framework is proposed by Cavallucci et al. (Cavallucci, Roussetot, and Zanni, 2011) which aims at extracting and representing the know-how of domain experts and populating an already constructed ontology of Inventive Design Modelling (IDM). Authors (Yan, Zanni-Merk, and Roussetot, 2011) proposed a method to calculate the semantic distance between short texts and use it to fill the semantic gap between the parameter and the generalized one and to facilitate the use of inventive design techniques. A formal contradiction model applicable to inventive design is highlighted (Roussetot, Zanni-Merk, and Cavallucci, 2012) to promote the related software development. A method based on a synergy between the theory of inventive problem solving and the case-based reasoning (Houssin et al., 2015) is introduced (Negny et al., 2012) to support engineers in preliminary design. Yan et al. (Yan et al., 2013) also presents an inventive method to facilitate the use of the contradiction matrix, using a semantic similarity approach and case-based reasoning.

However, for the aforementioned approaches based on TRIZ, an obvious drawback is that users still need to master the complexity of TRIZ methodology in order to make full use of TRIZ potential. It becomes a significant challenge for most novices to use it. How to automatize the inventive solutions retrieval for users who do not possess TRIZ knowledge is the main purpose of Chapter 5.

3.2.4 Automatizing TRIZ Inventive Process with Machine Learning

Patents can be exploited to serve creative action by providing targeted information on existing inventions with text mining techniques. The mining of TRIZ parameters and inventive principles, in particular, have already been attempted.

3.2.4.1 Contradictions and TRIZ parameters extraction

Chang et al. (Chang, Chang, and Wu, 2017) introduce a method to retrieve TRIZ parameters. They assume that a patent solves one contradiction and two TRIZ parameters are therefore improved (or at least not degraded). The method consists in key phrases detection. One of the parameters is supposed to be in a phrase similar to "to be prevented from worsening" and the second parameter in a phrase with "to be improved". The detection method is therefore very limited and it works only for Chinese patents as the syntax does not vary much.

Souili and Cavallucci (Souili and Cavallucci, 2017) use linguistic tools to extract concepts related to Inventive Design ontology from patents, i.e. problems and partial solutions. Sentiment analysis used as a solution often lies in a "positive" sentence, while a problem lies in a "negative sentence".

Following the work by Souili and Cavallucci (Souili and Cavallucci, 2017), Berdyugina and Cavallucci (Berdyugina and Cavallucci, 2022a) proposed an approach based on antonyms and topic modeling to identify potentially conflicting parameters in patents from the problems and parameters isolated earlier (Souili and Cavallucci, 2017).

Yanhong et al. (Yanhong Liang, Runhua Tan, and Jianhong Ma, 2008), present a methodology to retrieve patents according to the contradictions they solve. Nevertheless, the method uses Wordnet dictionary but it is unclear why, considering that no attempt to test this approach has been made. Therefore, it cannot be really considered as a prior art method. The same authors published another approach in 2009 (Liang et al., 2009), one year later, dealing with patent classification. This time, it was not a classification in accordance with the solved contradictions but in accordance with the inventive principles. First, the initial contradiction must be found. The most suitable inventive principles are then chosen and patents potentially related to the inventive principle are suggested to the user. To measure the relevancy of patents, a traditional TF-IDF (Term Frequency-Inverse Document Frequency) algorithm with Chi-square method is used.

Cascini and Russo (Cascini and Russo, 2007), present a framework to detect the solved contradictions. The worsening factor is supposed to be located in the background of the invention / state of the art and the improving factor in the claims. Nevertheless, the method in use seems to be limited (keywords approach) and no numerical results are shown.

3.2.4.2 Inventive principles, physical effects and solution mining

TF-IDF and LDA (Latent Dirichlet Allocation) clustering may be utilized for physical effects retrieval (Korobkin, Fomenkov, and Kravets, 2017).

Loh et al. (Loh, He, and Shen, 2006) show an attempt to classify patents in accordance with the inventive principle they use to solve a problem. A very small database is used (200 patents) and the patents are only classified using 6 inventive principles due to a lack of precision. He and Loh (He and Loh, 2008) also try to classify patents as a function of the inventive principles. In this case, all the inventive principles are used but, in order to deal with the lack of data and to make the training possible, groups of inventive principles are created. The purpose of the method is to recognize which groups of inventive principles are linked to the patents. Even with this simplification, the results are noisy with a very low recall.

Liang and Tan (Liang and Tan, 2007), present another methodology which is based on keywords, to achieve the same purpose. The idea is to recognize patterns associated with the problems or the way of solving these problems.

Patent parameters of a particular process are mined using semantic databases (Wang et al., 2016). These parameters are then associated with the general parameters from TRIZ matrix. Contradiction solving principles are also clustered to build a new TRIZ matrix with the process patents.

Ni et al. (Ni, Samet, and Cavallucci, 2022) propose a solving method based on problems matching. Their purpose is to avoid mining contradictions. Thus, from an initial problem, their model searches for semantically similar problems in patents. If problems are similar, the solution of the solved one should also apply to the non-solved one. It would at least give an insight on how to solve it. Question answering is coupled with the matching system to find the answer of the problem in the matched patents.

3.2.4.3 Estimation of inventiveness level

The estimation of the level of inventiveness is one of the main challenges for patents analysis in TRIZ domain. Nevertheless, very few methods were proposed.

Li et al. (Li et al., 2012) present a natural language processing and a metric-based citation method. Machine learning is used to infer the degree of inventiveness after a training on a dataset of 75 rated patents. The machine learning model takes as inputs the backward citations, the originality, the backward citation tag but also the knowledge transfer measurement. They

get more than 70% accuracy, which is relatively impressive considering the size of their dataset. Researchers [(Lanjouw and Schankerman, 2004) and (Cremers et al., 1999)] discuss the ways to measure patent quality and the link between patent value and the number of citations, while others (Jugulum and Frey, 2007) try to redefine Altshüller's matrix by replacing the inventiveness in the patents with the notion of robustness. The more an invention is robust, the more reliable it is.

3.2.5 Summary

Table 3.1 summarizes all the contributions in TRIZ domain linked to text or patent mining using machine learning. The different approaches are organized following their purpose and their date. We highlight that, prior to this work, only keywords analysis (Cascini and Russo, 2007) was attempted to retrieve contradictions. It consists in retrieving the phrases positioned after "increase" or "decrease" for example. Nevertheless, the two parts of the contradiction are, thus, processed independently and no real contradiction relation is extracted.

Following this work, Berdyugina and Cavallucci (Berdyugina and Cavallucci, 2022a) extended the keyword-based contradiction mining with the use of antonyms and switched to a deep learning-based approach.

Concerning TRIZ parameter mining which is also part of a full contradiction mining process, several works address this challenge but mostly using keywords and key phrases, which is not very relevant for most of the patents. Automatic language processing and patent mining are not new fields and can therefore motivate the choice of an approach. Contradiction extraction is the identification of two contradictory parameters (in the TRIZ sense) in a textual content. An unsupervised approach does not seem to be feasible here since the modeling of the notion of TRIZ contradiction is difficult and no clustering algorithm would allow the identification of these two parameters in contradiction. A supervised approach is therefore chosen.

It appears that the contradictions are understood owing to the sentences that surround the parameters. For example, in the US06938300 patent, the two following sentences are considered: *When the stroller 1 moves over a lawn or uneven road surfaces, it is necessary for the stroller wheels to have a large diameter so as to ensure the comfort of the baby. However, if each of the front wheel assemblies 11 has two large-diameter front wheels 13, the total volume and weight of the stroller 1 will increase significantly so that it is difficult to push the stroller 1.*

The notion of contradiction is included not in the parameters but in the relationship between the two sentences. This finding implied the development of a two-step approach. First, the modeling of the contradiction relations at the sentence-level and then the extraction of the parameters contained in the sentences in contradiction.

For sentence-level analysis, the closest NLP task is automatic summarization. It consists in selecting the important information in a text with respect to its purpose. The idea is to shift the initial summarization task to a TRIZ summarization which aims at identifying the contradiction relations between sentences. The automatic summarization can be based on sentence classification or on a generative model (see chapter 5).

TABLE 3.1: Summary of the state of the art in TRIZ domain

	Purpose	Approach
(Souili and Cavallucci, 2017)	Problems / Partial Solutions / TRIZ Parameters	Keywords, Sentiment Analysis
(Wang et al., 2016)	Process Parameters	Keywords, WordNet
(Chang, Chang, and Wu, 2017)	TRIZ Parameters	Keywords
(Cascini and Russo, 2007)	Contradictory TRIZ Parameters	Keywords, Patterns
(Berdugina and Cavallucci, 2022a)	Contradictory TRIZ Parameters	Keywords, LDA
(Berdugina and Cavallucci, 2022b)	TRIZ Parameters	Cause-Effect, BERT
(Loh, He, and Shen, 2006)	Inventive Principles	KNN, SVM, Naïve Bayes
(Liang and Tan, 2007)	Inventive principles	Keywords
(He and Loh, 2008)	Inventive Principles	SVM, Naive Bayes, Decision Tree
(Liang et al., 2009)	Inventive Principles	Bag of Words
(Korobkin, Fomenkov, and Kravets, 2017)	Physical effects	LDA, TF-IDF
(Cremers et al., 1999)	Patent value	Manual Analysis
(Lanjouw and Schankerman, 2004)	Patent value	Multiple Indicator Model (Linear)
(Jugulum and Frey, 2007)	Inventiveness / Robustness	Manual Analysis
(Li et al., 2012)	Inventiveness	ANN-Feedforward Neural Network
(Ni et al., 2021)	Inventiveness	MCDA
(Ni, Samet, and Cavallucci, 2022)	Problem/Problem matching	LSTM

3.3 Conclusion

In this chapter, we present the basics of any NLP task. We detail the neural network based approach for word/sentence embedding. In the second part of the chapter, we focus on presenting the TRIZ theory meant for providing inventive solution when the initial problem is presented in a contradiction form. In order to contribute to the field of TRIZ, we investigate the use of data mining techniques to automatize the process of inventiveness. Considering all existing approaches, we finally choose the tree-step approach. As no existing contradiction mining approach has been found, we compare our approach to classical summarization approaches. Contradiction has great importance for matching an input-formulated problem to all contradiction in large volumes of patents. It allows to filter them through their inventiveness and provides the engineers with a set of similar problems solved with inventive solutions. These contributions are detailed in Chapter 5.

Part III

Scientific Contributions

Chapter 4

Chronicle Mining and Reasoning in Predictive Maintenance

Contents

Introduction	80
5.1 Full Framework of TRIZ Patent Inventive Design	80
5.2 Mining Patent TRIZ Contradictions	80
5.2.1 Baseline Approach	82
5.2.2 PaGAN: Patent Generative Adversarial Network	83
5.3 SAM-IDM for Inventive solution mining	89
5.4 Inventive solution mining using Multiple-Criteria Decision Analysis (MCDA)	91
5.5 Experiments	95
5.5.1 Contradiction mining experiments	95
5.5.2 Contradiction matching experiments	102
5.5.3 Use case example with PatRIS	104
5.6 Conclusion	105

Introduction

In this chapter, we are interested in showing how pattern mining is used in predictive maintenance application. As detailed in Chapter 2, we focus in investigating one of the richest and highly descriptive patterns which are chronicles. Two main investigations are addressed in this chapter: (i) How to mine chronicles to detect the signature of a failure from machines sensor sequences; (ii) How to use these mined patterns to prevent future failures.

4.1 Mining Failure Chronicles for Predictive Maintenance

In predictive maintenance context, data come sequentially. The data is recorded until a failure happens. Then, several sequences of sensor data are recorded ending with a failure event. In order to understand the failure through a mining process, several preprocessing steps have to be taken. Figure 4.1 explains the steps used in every mining approach. In fact, to select the most relevant data and to make it suitable for chronicle mining, methods of attribute selection (Novakovic, 2000) followed by the discretization of the data are applied. Subsequently, after the processing of the sequences, we mine closed patterns with a sequential closed pattern mining algorithm. Then, we extract the time constraints between events of patterns. Data preprocessing refers to operations performed on raw data to make them suitable for another processing use. It is a preliminary step that is used in the field of data mining and is needed to transform the data in a format that can facilitate the mining step. Feature selection is used to reduce dimensionality and eliminate irrelevant attributes. Subsequently, the discretization is used since we need a finite set of events for mining process.

Several datasets as SECOM (McCann and Johnston, 2008) request a further transformation to build a sequential dataset. For this purpose, we add sequentialization process to our approach to handle the cases when the timestamped data do not constitute a sequence. In our case, a sequence is a set of ordered events that end with a particular event: the machine failure. Therefore, sequential pattern mining techniques are applicable. After preprocessing, resulting data are ready to use. The idea is to mine the chronicles that can describe a given failure. These chronicles are called *failure chronicles* which are explained in Definition 18.

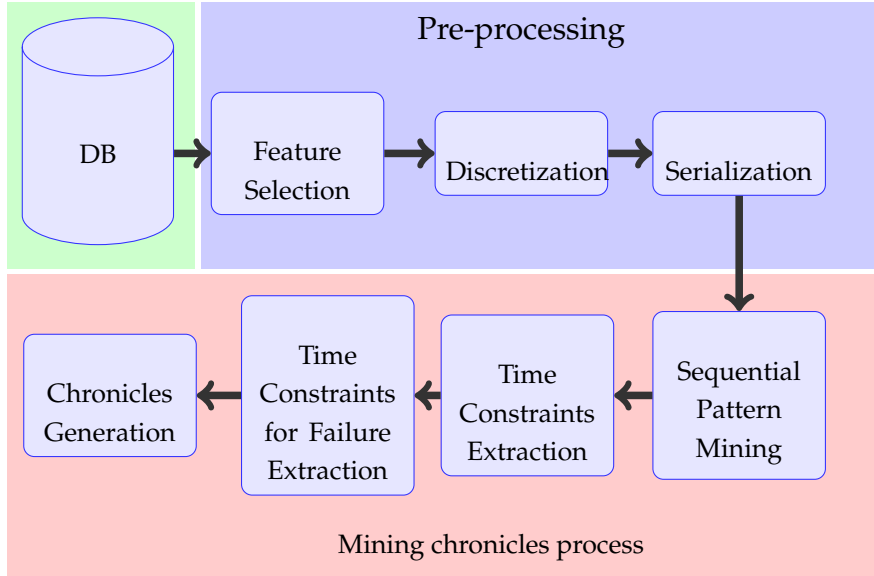


FIGURE 4.1: Figure illustrating the steps of the chronicle mining

Definition 18 (Failure chronicle). Assuming a chronicle $C_F = (\mathcal{E}, \mathcal{T})$. We say that C_F is a failure chronicle if and only if its events are ordered according to their appearances in the sequence and the last event in the chronicle is the one that describes the failure, i.e. for $\mathcal{E} = \{e_1 \cdots e_n \mid e_i \leq_{\mathbb{E}} e_{i+1}, i \in [1, n]\}$, e_n is the failure event.

The contributions of this chapter are related to these failure chronicles. Our contributions are twofold: (i) mining these failure chronicles showing all the steps before reaching a failing status; (ii) predicting future failures once all failure chronicles are mined. In the next section, we detail the mining algorithms to mine failure chronicles.

4.1.1 CPM Algorithm

Algorithm 5 summarizes the general idea of our approach. It runs in three main stages. At first, FrequentSet extracts all the frequent closed patterns in the dataset. This step is carried out by the extraction algorithm of frequent closed sequences CloSpan (Yan, Han, and Afshar, 2003). Then, for each frequent pattern, the *ExtractTimeConstraints* function allows to extract the different time constraints that separate its events.

For each frequent sequence, the conjunction of its temporal constraints extracted by the time constraints function allows to obtain a chronicle which models the sequencing from the first event until reaching the failure.

A sequence may contain one or more events and subsequently it may be described by one or more chronicles. In fact, to get a better detection, we have to specify the sequences covered by the chronicles; thus we can keep

Algorithm 5 Extract Frequent Chronicles

Require: \mathcal{D} : The database of the events, $f_{q_{min}}$: minimum threshold
Ensure: \mathcal{C} : set of frequent failure chronicles

- 1: $\{\text{FS}, \text{IFS}\} \leftarrow \text{FrequentSet}(\mathcal{D}, f_{q_{min}})$
 \triangleright Extraction and index of lines of frequent closed sequential patterns.
- 2: $\mathcal{C} \leftarrow \emptyset;$
- 3: **for** $s \in \text{FS}$ **do**
- 4: $\mathcal{T} \leftarrow \text{ExtractTimeConstraints}(\mathcal{D}, s, \text{IFS})$
- 5: $\mathcal{C} \leftarrow \mathcal{C} \cup \{(s, \mathcal{T})\}$

return \mathcal{C}

only the most relevant chronicle for a given failure, taking for example the support of the chronicle as a reference measure.

4.1.2 Clasp-CPM

The first implementation of chronicle mining (Sellami, Samet, and Tobji, 2018) handled few cases of time constraints extraction. It was dependent on the length of the patterns and treated these on a case-by-case basis. Moreover, chronicle extraction needed two steps of time constraints extraction : one for the events on the patterns, and another to extract constraints between regular events and the breakdown event. To overcome these two major setbacks, two notions were used : subsequence graphs and suffix datasets.

First, we introduce a sample data set, shown in Table 4.1 which will be used for examples in this section.

TABLE 4.1: Sample data set

Seq. Id	Events
1	$\langle (ace, 1), (bdf, 3), (acf, 4), (ace, 8), (bde, 10) \rangle$
2	$\langle (acf, 2), (adf, 6), (ace, 7), (bdf, 9) \rangle$
3	$\langle (adf, 0), (acf, 3), (acf, 7), (bdf, 8), (bce, 12) \rangle$
4	$\langle (acf, 1), (adf, 3), (ace, 4), (bdf, 7), (ade, 9) \rangle$
5	$\langle (adf, 2), (acf, 4), (bde, 7) \rangle$

Definition 19 (Suffix database). Let ω be a sequence. D_ω is said to be the suffix database associated to D if :

$$\forall s \in D_\omega, \exists s' \in D, s = s' \diamond_s \omega$$

Assuming ω is a suffix for all sequences in D_ω , we note \mathcal{FS}_ω and \mathcal{CS}_ω the set of frequent sequences and the set of closed frequent sequences associated to the suffix database D_ω , respectively.

Remark 2.

- We have defined the suffix database with the s -concatenation operator as i -concatenation does **not** preserve the closeness property we need.
- In our application, we will use a sequence of length 1 for ω .
- $\#D = \#D_\omega$, i.e. D and D_ω have both the same number of sequences.
- $\mathcal{FS} \subseteq \mathcal{FS}_\omega$, more precisely $\mathcal{FS}_\omega = \mathcal{FS} \cup \{s \diamond_s \omega \mid s \in \mathcal{FS}\}$. This tells us that if a sequence is frequent in D , then it is also frequent in D_ω , but the converse does not hold.
- $s_\omega \in \mathcal{FS}_\omega \wedge s_\omega = s \diamond_s \omega \Rightarrow s \in \mathcal{FS}$.

Example 7 (Suffix database). Table 4.2 present a suffix database, by s -appending the failure itemset $\{P\}$ at the end of each sequence.

TABLE 4.2: Sample suffix database : P is the failure event, s -concatenated at the end of each sequence

Seq. ID	Events
1	$\langle (ace, 1), (bdf, 3), (acf, 4), (ace, 8), (bde, 10), (P, 10) \rangle$
2	$\langle (acf, 2), (adf, 6), (ace, 7), (bdf, 9), (P, 9) \rangle$
3	$\langle (adf, 0), (acf, 3), (acf, 7), (bdf, 8), (bce, 12), (P, 12) \rangle$
4	$\langle (acf, 1), (adf, 3), (ace, 4), (bdf, 7), (ade, 9), (P, 9) \rangle$
5	$\langle (adf, 2), (acf, 4), (bde, 7), (P, 7) \rangle$

One can choose the timestamp to be different to that of the last element, but the added time should be constant to keep a coherent analysis later.

Proposition 1.

$$\mathcal{CS}_\omega = \{s \diamond_s \omega \mid s \in \mathcal{CS}\} \quad (4.1)$$

Lemma 1. Let ω be the sequence associated to the suffix database D_ω built from the database D . Then

$$\forall s \in \mathcal{FS}, \exists s' \in \mathcal{FS}_\omega, s' = s \diamond_s \omega : \text{supp}(s) = \text{supp}(s') \quad (4.2)$$

This proposition is useful to prove that every closed frequent sequence of a suffix database ends with the sequence ω . We use this fact to improve the previously presented algorithm, as there will be only one *ExtractTimeConstraints* procedure and it has no union of chronicles at the end.

4.1.2.1 Subsequence graph

As chronicle mining algorithms are resource greedy, we are obliged to improve existing works to scale with manufacturing requirements (i.e. large number of sequences, real-time prediction, etc). In addition, to handle the huge volume of sequences, several optimization methods and multi-threading coding are harnessed. A new graph-based approach is introduced to improve the chronicle mining. In this section, we introduce all related graph basics needed for the subsequence graph construction.

A subsequence graph is a data structure built to represent all occurrences of a sequence within another sequence. It was designed similarly to the Knuth-Morris-Pratt algorithm for word matching.

Definition 20 (Subsequence graph). Let $s = \langle s_1, \dots, s_n \rangle$ be a sequence and $p = \langle p_1, \dots, p_m \rangle$, a pattern. A subsequence graph is a couple (X, U) where X are the vertices and U are the (directed) edges. Vertices $v_{i,j}$ correspond to elements of the pattern p_i , indexed by their position on s , j . There is an edge from $v_{i,j}$ to $v_{k,l}$ iff

- $k = i$ (both pattern elements are the same) and $v_{k,l}$, $l > j$, is the first next occurrence (of such pattern element) in s .
- $k = i + 1$ and there is no edge from $v_{i,j}$ to $v_{k,m}$ with $m < l$ ($v_{k,l}$ is the first occurrence of p_{i+1} after $v_{i,j}$).

Example 8 (Graph construction). Assuming the pattern $\langle abbc \rangle$ and the sequence $\langle bacbbacabcbabc \rangle$. Table 4.3 enumerates the sequence :

TABLE 4.3: Sequence enumeration

b	a	c	b	b	a	c	a	b	c	b	a	b	c
1	2	3	4	5	6	7	8	9	10	11	12	13	14

1. First, we initialize an empty list of lists, such that the first element corresponds to the occurrences of a , the second one to the occurrences of b , the third one to the occurrences of b following a b and the fourth one to the occurrences of c .

2. Next, in the first step, b is placed in the second list. Then, a is placed in the first list. In step three, c is placed in the fourth list (figure 4.2). Notice there are no links yet as the elements, while they are individually in the pattern, have not occurred in the order of the pattern.
3. During the fourth iteration, we find a second b . We proceed in four steps : b is placed in the third list, we add links from the second list elements, without links to a third list element, to this b element, then we add b to the second list and we add links from the first list elements, without links to a second list element, to this b (Figure 4.3).
4. We continue the process. During step seven, the graph allows to extract the first occurrence of the pattern in the sequence: $\langle (a, 2), (b, 4), (b, 5), (c, 7) \rangle$ (Figure 4.4).
5. Last, when the whole sequence has been treated, one can perform pruning steps to remove the nodes without links or that do not allow to extract a whole pattern (Figure 4.5).



FIGURE 4.2:
Subsequence
graph: After
three steps

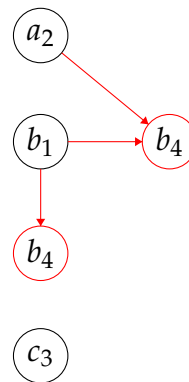


FIGURE 4.3:
Subsequence
graph: After
four steps

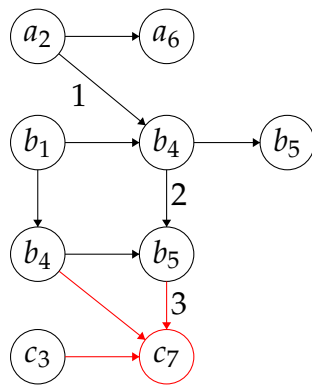


FIGURE 4.4:
Subsequence
graph: sev-
enth step

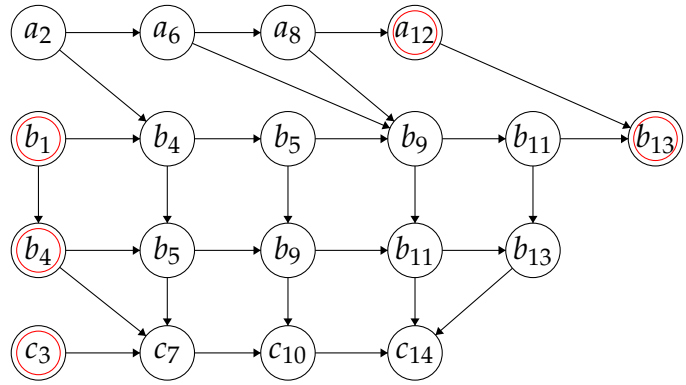


FIGURE 4.5:
Subsequence
graph: useless
nodes in red
are marked to
be pruned

We devised this structure to be able to extract all the information needed, namely timestamps and precedence relations for each element of a pattern in a sequence. It is not consuming in space, as each element is only referenced.

4.1.2.2 Multi-threading

The first step of CPM is frequent closed-sequences mining, which we found hard (if possible) to parallelize. Thus, we enhanced the following steps: time constraint extraction and chronicle building with multi-threading.

Using a Producer-Consumer approach, we defined the following schema:

1. Initialize a certain number of workers, a pool of extracted closed patterns, a pool of time constraints sets and a pool of chronicles.
2. (Time constraint extraction) For half the workers:
 - (a) Take a pattern from the pool.
 - (b) Extract the time constraints associated with it, i.e. build a subsequence graph for each sequence in which the pattern is found and use it to extract the constraints.
 - (c) Put the time constraints in the corresponding pool.
 - (d) Repeat until the pattern pool is empty.
3. (Chronicle Building) For the remaining half the workers:
 - Take the constraints from the pool.

- Build a chronicle using the time constraints.
- Put the chronicle in the corresponding pool.
- Repeat until the constraint pool is empty.

An improvement would be to switch workers' tasks when one of the pools is empty, e.g. if the time constraint pool is empty, then all workers are extracting time constraints, and vice versa.

There is no race conditions when accessing the data set as the sequences are not modified but only read. Our implementation uses blocking queues and linked blocking queues (Java) for the pools. Here again, each element is a reference only, so we do not significantly increase memory usage beyond the extraction step.

4.2 Failure Detection with Chronicles

In this section, we are interested in using the chronicle for the predictive maintenance task. We aim to predict failure using the mined set of failure chronicles.

4.2.1 Maximum Confidence-based approach

After mining of the chronicles characterizing the anomalous sequence, a second step is developed to test the prediction of the mined chronicle which is represented in the Algorithm 6. The algorithm requires to match chronicles to input sequences. Two definitions are detailed.

Definition 21 (Chronicle cover). *Assuming a sequence $S = \langle SID, (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle$ and a failure chronicle C . We say that c covers the sequence SID if and only if the events represented by the chronicle belong to the sequence as well as the time intervals between these events in the sequence belong to the temporal constraints extracted by the chronicle, i.e.,*

$$C \triangleleft S \Leftrightarrow \forall e_i [t^-, t^+] e_j \in C,$$

$$\exists ((e, t), (e', t')) \in S \wedge e = e_i, e' = e_j \wedge t' - t \in [t^-, t^+].$$

Definition 22 (Supported failure chronicle). *Assuming a sequence S and the set of failure covering chronicle, i.e., $\mathcal{C} = \{C \in \mathcal{C}, c \triangleleft S\}$. We say that \mathcal{C}_f is*

the supported failure chronicle if and only if it has the maximal support among all chronicles of the set \mathbf{C} , i.e. $\mathcal{C}_f = \{C \in \mathbf{C} \wedge \nexists c' \in \mathbf{C} \wedge \text{supp}(C') > \text{supp}(C)\}$.

Algorithm 6 represents the general idea of detecting anomalies using both Definitions 21 and 22.

Algorithm 6 Failure Detection with Chronicles

Require: S : Sequence, \mathcal{C} : Chronicles set

Ensure: \mathcal{C}_f

```

1:  $\mathcal{C}_f \leftarrow \emptyset$ 
2: for all  $C \in \mathcal{C}$  do
3:   if ( $\text{COVERAGE}(S, C)$ ) then
4:      $\mathbf{C} \leftarrow \mathbf{C} \cup c$ 
5:  $\text{supp}_{\min} \leftarrow 0$ 
6: for all  $C \in \mathbf{C}$  do
7:   if  $\text{supp}(C) > \text{supp}_{\min}$  then
8:      $\mathcal{C}_f \leftarrow c$ 
9:      $\text{supp}_{\min} \leftarrow \text{supp}(C)$ 
10: if  $\mathcal{C}_f \neq \emptyset$  then
11:   return  $\mathcal{C}_f$ 

```

Algorithm 7 Procedure COVERAGE for computing the covering chronicles

```

1: procedure  $\text{COVERAGE}(S, C)$ 
2:   for all  $(e_1, e_2) \in \mathcal{E}$  and  $(t^-, t^+) \in \mathcal{T}$  do
3:     if  $\exists ((e, t), (e', t')) \in S \mid e = e_1, e' = e_2$  and  $t - t' \in [t^-, t^+]$  then
4:       return true
5:   return false

```

First, the algorithm uses the coverage procedure to check whether the processed sequence is described by the set of chronicles or not. This procedure takes as parameters a sequence and a set of chronicles. For a given sequence, the procedure checks if there is one or more chronicles whose events belong to the sequence. Furthermore, the procedure also checks whether the temporal constraints described by the chronicle are validated by the sequence or not.

If the sequence is shown by the chronicles, the algorithm traverses this set to extract the one with the highest support; it is the supported failure chronicle.

4.2.2 The algorithm for rule base integration with detection of the three issues

One of the main asset of chronicles is their high descriptiveness. In this approach, we investigate the case when an expert formulates rules (in chronicle form) to detect failure. These expert chronicles are used along with those mined from the collected sensor data. To cohabit together mined rules and expert ones, we provide several definitions.

Definition 23 (Chronicle rule redundancy). *A chronicle rule $R : EC \wedge TEC \rightarrow TF$ and a rule $R' : EC' \wedge TEC' \rightarrow TF'$ are considered as redundant when the two rules have the same sets of events, and the events have the same temporal constraints, denoted as*

$$\text{ChroRedundancy}(R, R') \iff (EC = EC') \wedge (TEC = TEC') \wedge (TF = TF'). \quad (4.3)$$

Definition 24 (Chronicle rule conflict). *A chronicle rule $R : EC \wedge TEC \rightarrow TF$ and a rule $R' : EC' \wedge TEC' \rightarrow TF'$ are considered to have conflict when the two rules succeed in the same sets of non-failure events and the same temporal constraints for these non-failure events, but with conflicting temporal constraints of a failure, denoted as*

$$\text{ChroConflict}(R, R') \iff (EC = EC') \wedge (TEC = TEC') \wedge (TF \neq TF'). \quad (4.4)$$

Definition 25 (Chronicle rule subsumption). *A chronicle rule $R : EC \wedge TEC \rightarrow TF$ subsumes the chronicle rule $R' : EC' \wedge TEC' \rightarrow TF'$ when they have the same temporal constraints of a failure, but rule R contains additional restrictions on the set of non-failure events or on the temporal constraints of these non-failure events. This subsumption relationship is denoted as*

$$\text{ChroSubsums}(R, R') \iff (TF = TF') \wedge ((EC' \wedge TEC') \models (EC \wedge TEC)). \quad (4.5)$$

These definitions are used to detect issues with regard to rule base verification. They provide foundations for the rule base refinement approach.

To ensure, that these chronicles obey the same construction of these mined from the data, an ontology is used (Cao et al., 2020). To automate the rule

base integration and refinement process, we propose an algorithm to detect the issues when an expert rule is integrated into the chronicle rule base. The algorithm checks *Redundancy*, *Conflict*, and *Subsumption* between the chronicle rules and an expert rule. The pseudocode is shown in Algorithm 8. The algorithm runs into six major steps:

- For an expert rule R_e , three functions extract different sets of atoms from it. The function *Non-failureEvent* extracts all the non-failure events inside R_e . The function *Non-failureTemporalConstraints* extracts the temporal constraints of these non-failure events, and the function *FailureTemporalConstraints* extracts the temporal constraints of the failure.
- For each chronicle rule R in the chronicle rule base \mathcal{R} , the three functions that are described in the previous step extract the same types of atoms from it (non-failure events, temporal constraints of the non-failure events, and the temporal constraints of the failure).
- For each chronicle rule R and the expert rule R_e , redundancy issue is checked by the algorithm. If they are redundant, the expert rule is removed whilst the redundant chronicle rule is retained in the rule base.
- Rule *Subsumption* is examined after the check of *Redundancy*. For each chronicle rule R and the expert rule R_e , if R subsumes R_e (if R is more specific than R_e), then discard R_e ; If R_e subsumes R , then remove R from the rule base \mathcal{R} and integrate the expert rule R_e into \mathcal{R} .
- After the *Redundancy* and *Subsumption* issues are inspected, the last issue to be examined is rule *Conflict*. If the expert rule R_e is in conflict with a chronicle rule R , then remove R and integrate R_e into the rule base. The reason for doing this is based on the assumption that the reliability of an expert rule is always higher than or equal to a chronicle rule when they have *Conflict*. In this context, when there is a rule *Conflict* issue, the expert rule always have the priority of being integrated into the rule base.
- Return the refined rule base \mathcal{R}' as the algorithm output.

By taking the expert rules as input, Algorithm 8 is applied to check whether there exist issues among the expert rules and chronicle rules. The description of the used functions are defined in Table 4.4. The chronicle rule base is refined and verified progressively for each input expert rule. At last,

Algorithm 8 The algorithm to detect the three issues when an expert rule is integrated into the chronicle rule base.

Require: \mathcal{R} : a chronicle rule base which contains a set of failure chronicles,
 R_e : an expert rule which is in the form of a failure chronicle.

Ensure: \mathcal{R}' : the integrated rule base.

```

1:  $EE \leftarrow \emptyset, TEE \leftarrow \emptyset, TFE \leftarrow \emptyset, EC \leftarrow \emptyset, TEC \leftarrow \emptyset, TFC \leftarrow \emptyset, \mathcal{R}' \leftarrow \emptyset.$ 
2:  $EE \leftarrow Non - failureEvent(R_e)$ 
3:  $TEE \leftarrow Non - failureTemporalConstraints(R_e)$ 
4:  $TFE \leftarrow FailureTemporalConstraints(R_e)$ 
5: for each  $R \in \mathcal{R}$  do
6:    $EC \leftarrow Non - failureEvent(R)$ 
7:    $TEC \leftarrow Non - failureTemporalConstraints(R)$ 
8:    $TFC \leftarrow FailureTemporalConstraints(R)$ 
9:   if  $EE = EC, TEE = TEC, TFE = TEC,$  then
10:     $ChroRedundancy(R_e, R)$ 
11:     $Remove(R_e, \mathcal{R})$ 
12:   else if  $TFE = TFC, EC = EE, TEC \subseteq TE,$  then
13:     $ChroSubsumes(R_e, R)$ 
14:     $Remove(R, \mathcal{R})$ 
15:     $Integrate(R_e, \mathcal{R})$ 
16:   else if  $EE = EC, TE \subseteq TEC, TFE \neq TFC,$  then
17:     $ChroConflict(R_e, R)$ 
18:     $Remove(R, \mathcal{R})$ 
19:     $Integrate(R_e, \mathcal{R})$ 
20:   else
21:     $Integrate(R_e, \mathcal{R})$ 
22:   end if
23: end for each
24:  $\mathcal{R}' \leftarrow \mathcal{R}$ 
25: return  $\mathcal{R}'$ 

```

a refined rule base is obtained, which consists of best-quality chronicle rules as well as expert rules.

4.2.3 Chronicle fusion approach

When considering classifying a sequence as an anomaly or predicting when it is going to happen, we may encounter several failure chronicles that appear in the sequence. Thus, it becomes problematic to cope with different information when predicting. Therefore, in this subsection, we aim to use all the supported failure chronicles rather than use the one maximizing the support. To do so, using fusion framework as the evidence theory (Dempster, 1968; Shafer, 1976b) is a strategy.

TABLE 4.4: Details of the functions of Algorithm 8

Function	Role
<i>Non – failureEvent()</i>	Extract all the non-failure events of this expert rule
<i>Non – failureTemporalConstraints()</i>	Extract the temporal constraints of these non-failure events
<i>FailureTemporalConstraints()</i>	Extract the temporal constraints of the failure
<i>ChroRedundancy()</i>	Mark the rule as redundant
<i>ChroSubsumes()</i>	Mark the rule as subsuming
<i>ChroConflict()</i>	Mark the rule as conflictual
<i>Remove()</i>	Remove a rule from the set
<i>Integrate()</i>	Integrate a rule into the set

We investigate a new approach for fusing several chronicles when we analyse a new incoming sequence. In this approach, we consider not only failure chronicles but also other chronicles not ending with the failure event. Indeed, we assume that a sequence could end up as a normal sequence even if a failure chronicle covers it. The proposed approach rely on the evidence theory. Readers may refer to Appendix B for details about the theory.

Definition 26 (Basic Belief Assignment modeling). *Assuming a chronicle $C_i \in C_f \cup C_{\bar{f}}$ that covers a sequence S , we model the Basic Belief Assignment (BBA) function m_i of C_i in $\theta = \{f, \bar{f}\}$ as follows :*

$$\begin{cases} m_i(f) = \text{Supp}(C_i) \\ m_i(\bar{f}) = 0 \\ m_i(\theta) = 1 - \text{Supp}(C_i) \end{cases} \quad (4.6)$$

Definition 27 (Chronicles combination). *Assuming N chronicles C_i that cover a sequence S , with $m_i, i \in [1, N]$, the basic belief assignment function relative to the i^{th} chronicle. The joint mass function that combines all the m_i mass functions of the chronicles C_i that cover S using the Dempster Rule of combination is defined as follows:*

$$m_{\oplus}(A) = m_1 \oplus \dots \oplus m_N(A); \forall A \subseteq \theta \quad (4.7)$$

To make the decision, we compute the pignistic probability *BetP* for failure (f) and normal (\bar{f}) outcome. The final decision is obtained by retaining the hypothesis that maximized the pignistic probability as follows:

$$x = \text{argmax}_{x_i \in \theta} \text{BetP}(x_i). \quad (4.8)$$

For the prediction task, we developed the FCP algorithm (Fusion of Chronicles for Prediction). It consists in comparing the input sequence (to predict) with every chronicle in terms of events and time constraints. To each matching chronicle, we model a BBA that measures to which degree the chronicle expresses the failure (f) and normal (\bar{f}) behaviour classes. The level of uncertainty is retained using the support of the chronicle. Once all matching chronicles are modelled, we use the Dempster rule of combination to combine all the BBAs. The joint BBA shows the membership of the input sequence to both classes. The final class is computed using the argmax function. If the final class is failure, we display the failure time by aggregating the time constraints of all matching failure chronicles. Algorithm 9 performs the combination of the covering chronicles to predict the status of a sequence using all aforementioned notions.

4.3 Experiments

4.3.1 The SECOM Data Set

Our approaches are validated by conducting experimentation on the SECOM data set (McCann and Johnston, 2008), which contains measurements of features of semi-conductor production within a semi-conductor manufacturing process. In the SECOM data set, 1567 data records and 590 attributes are collected, with each recording being characterized by a timestamp referring to the time that the data is recorded. Each recording is also associated with a label, which is either 1 or -1. The label of every recording explains the correctness of the event, with -1 corresponding to a non-failure event, and 1 refers to a failure. Timestamps are associated with all the records indicating the moment of each specific test point. In total, 104 records represent the failures of production. The data is stored in a raw text file, within which each line represents an individual example of recording with its timestamp. The features are separated by spaces.

However, the data contained in SECOM data set do not have the same types of attributes and values, that some of the information contained in the data is irrelevant to the failure prediction task thus is considered as noise. Moreover, due to the inter-dependency among individual features and the complex behavior of combined features, it is difficult to extract frequent patterns and rules based on the analysis of all the 590 attributes. Thus, in this context, instead of going through the entire data set and use all 590 attributes

Algorithm 9 Fusion of Chronicle for Prediction

Require: S : sequence, \mathcal{C} : chronicles set
Ensure: R : result, $min_time_failure$: minimum time to failure,
 $max_time_failure$: maximum time to failure

- 1: $C_M \leftarrow \{\}$
- 2: **for all** $C \in \mathcal{C}$ **do**
- 3: **if** (coverage(S, C)) **then**
- 4: $C_M \leftarrow C_M \cup C$
- 5: **for all** $C \in C_M$ **do**
- 6: **if** ($C.Type == f$) **then**
- 7: $m \begin{cases} m(f) = Supp(C) \\ m(\bar{f}) = 0 \\ m(\theta) = 1 - m(f) \end{cases}$
- 8: **else**
- 9: $m \begin{cases} m(f) = 0 \\ m(\bar{f}) = Supp(C) \\ m(\theta) = 1 - m(\bar{f}) \end{cases}$
- 10: $m_{\oplus} \leftarrow m_{\oplus} \oplus m$
- 11: $R \leftarrow argmax_{x_i \in \theta} BetP(x_i)$
- 12: **if** ($R == f$) **then**
- 13: Init($C_M, min_time_failure, max_time_failure$)
- 14: **for all** $C \in C_M$ **do**
- 15: **if** ($min_time_failure > C.min_time$) **then**
- 16: $min_time_failure \leftarrow C.min_time$
- 17: **if** ($max_time_failure < C.max_time$) **then**
- 18: $max_time_failure \leftarrow C.max_time$
- 19: **return** $R, min_time_failure, max_time_failure$
- 20: **else**
- 21: **return** R

for failure prediction, feature selection methods (Guyon and Elisseeff, 2003a) are used to identify and select the most relevant attributes in predicting the failures. The selected attributes are subsequently used to extract the key factors and patterns that lead to machine failures. This reduces the data processing time and memory consumption.

4.3.2 Evaluation of the chronicle mining

In this first part, we confront *Clasp-CPM* to other algorithms of the state-of-the-art using synthetic data sets. We compared it to a previous brute-force algorithm called *CPM* (Sellami, Samet, and Tobji, 2018), as well as *DCM* (Dauxais et al., 2015) and *FACE* (Dousson and Duong, 1999). In our experiments,

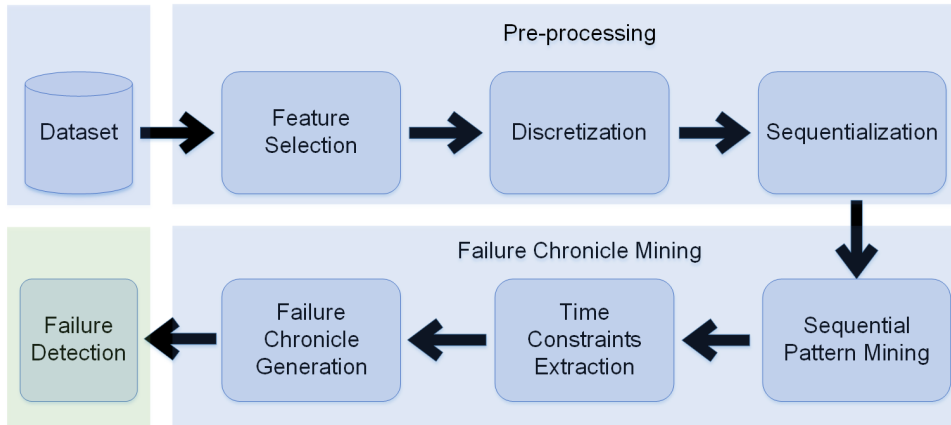


FIGURE 4.6: Different steps used in the frequent failure chronicle mining approach, adapted from (Sellami et al., 2019).

we used different synthetic data sets to test the effect of several parameters on the results. Table 4.5 shows the number of generated chronicles by the aforementioned algorithms for a range of support threshold that goes from 0.4 to 1 for three different synthetic data sets where we change *seq_size*, *dic_size* and *max_items/event*.

As Clasp and Clospan extract the same number of closed patterns (Antonio et al., 2013), the experiments show that both Clasp-CPM and CPM generate the same number of frequent chronicles. This number depends on the different parameters used when generating the data set. Obviously, it increases considerably while increasing the parameters' values. These same experiments done on FACE algorithm (Dousson and Duong, 1999) show that this algorithm generates the highest number of chronicles, since it is an Apriori based algorithm. In fact, Apriori extracts all the frequent patterns, not only the closed frequent ones. In our introduced approach, we bypassed the Apriori-like methods to avoid redundant chronicles, and to optimise the performance of our chronicle mining step. On the other hand, DCM (Dauxais et al., 2015) was designed to consider discriminancy in data, so the comparison with our algorithm is not straightforward.

Extracting chronicles was only possible for threshold values greater than 0.7 as shown in Table 4.5. One hypothesis for these results is, as already mentioned, the use of the discriminance constraint. This parameter was integrated in the epidemiology algorithm to distinguish two populations (positive and negative) and to extract patterns that are frequently present in the positive base, which is not really the case in our approach since we only process a single data set (population) at a time.

Table 4.5 shows that the maximum size of a sequence is the parameter that

TABLE 4.5: Number of chronicles of Clasp-CPM, CPM, DCM and FACE w.r.t data set size, sequence's size, dictionary size, maximum events/item and support threshold

Threshold	Clasp-CPM	CPM	DCM	FACE
DB 500 ; seq_size = 10 ; dic_size = 5 ; max_items/event = 3				
0.4	70103	70103	N/A	74531
0.5	61403	61403	N/A	65312
0.6	8156	8156	N/A	9703
0.7	7180	7180	8336	8854
0.8	471	471	514	630
0.9	380	380	483	520
DB 500 ; seq_size = 10 ; dic_size = 25 ; max_items/event = 7				
0.4	86723	86723	N/A	89641
0.5	81641	81641	N/A	85644
0.6	71187	71187	N/A	74290
0.7	11280	11280	13002	13383
0.8	6521	6521	6701	6824
0.9	4236	4236	4812	5148
DB 500 ; seq_size = 15 ; dic_size = 30 ; max_items/event = 7				
0.4	135681	135681	N/A	140023
0.5	126550	126550	N/A	127670
0.6	27930	27930	N/A	28354
0.7	20998	20998	22601	21564
0.8	930	930	941	950
0.9	784	784	846	897

affects mostly the number of obtained chronicles. On the other hand, increasing the dictionary size and the maximum number of items per event leads to a small increase of the number of chronicles. This behaviour is explained by the fact that small sequences generate fewer closed patterns. A larger sequence size induces more frequent closed sequential patterns, which increases the number of generated frequent chronicles. This experiment result supports our choice to use an attribute selection method in the pre-processing step of our approach. Indeed, the more attributes (measures) we consider, the more our sequences are long. That's why we consider only relevant attributes, avoiding a huge number of "irrelevant" chronicles that would make our approach more costly, without a significant impact on the prediction step.

4.3.3 Predictive maintenance analysis

To evaluate the quality of prediction, we used several measures. The first is to compute the *True Positive Rate* of the extracted chronicles. It is used to measure the number of positive sequences that are correctly classified, i.e., the sequences for which there is at least one chronicle allowing to predict the appearance of the failure without taking into consideration the temporal constraints. Indeed, for each sequence, if its events are described by the chronicle, we have correctly classified the sequence's failure and the chronicle could have predicted what are the events which caused this breakdown. Otherwise, if there is no chronicle that could describe the failure for a given sequence, therefore there is no prediction of failure so the sequence was misclassified. The idea is to bring the approach to a classification problem to apply the cross-validation method (Stone, 1974).

In our experiments, we used the 10 fold cross validation evaluation methods. For each value of $f_{q_{min}}$, the chronicles are extracted from the training sequences. Then, for the test set, we check for each sequence its membership in at least one chronicle among those extracted. The number of sequences validated by the chronicles is computed to estimate its percentage with respect to the sequence set. This procedure is repeated 10 times to validate all the sequences of the data set. This is the same principle used to compute the recall rate (Davis and Goadrich, 2006), which is defined by the number of relevant instances found in relation to the number of relevant instances in the data set.

The True Positive Rate is computed according to this formula :

$$TPR = \frac{TP}{TP + FN} \quad (4.9)$$

where TP (the true positive results) is the number of validated sequences, for which we found at least one chronicle that could have predicted the failure, and FN (the false negative results) is the number of sequences for which no chronicle could predict the failure.

Indeed, if there is no chronicle that describe a sequence S , then we cannot classify the sequence as a "failure", and we state the "normal" case. However, all the sequences of our data set lead to a machine failure, that's why we consider such a sequence classified as false negative.

In Example 9, the first three sequences are described by the given chronicle, since according to this chronicle an event a followed by a b causes a failure, whereas this is not the case for the fourth sequence since the event d

is not described by the chronicle. So the true positive rate for this example is: $\frac{3}{3+1} = 75\%$.

Example 9. Let's take the chronicle shown in Figure 4.7 and the four sequences depicted in Table 4.6.

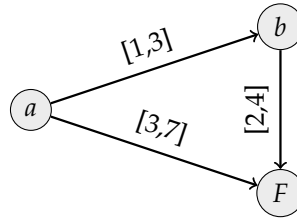


FIGURE 4.7: Example of a chronicle

TABLE 4.6: Sequences' data set

SID	Events
1	$\langle (a, 1), (b, 2), (F, 4) \rangle$
2	$\langle (a, 0), (b, 3), (F, 7) \rangle$
3	$\langle (a, 2), (b, 6), (F, 17) \rangle$
4	$\langle (a, 3), (d, 5), (F, 5) \rangle$

The second measure we used evaluates the precision of the results with consideration of the failure time, i.e., it estimates the percentage of sequences for which time constraints are extracted correctly. Indeed, for each sequence, if the moment predicted by the extracted chronicles is outside the failure appearance interval in the sequence, so the chronicle could not extract the temporal constraints of this failure, and the failure is classified as false positive. These interpretations allow us to apply the following precision formula:

$$Precision = \frac{TP}{TP + FP}. \quad (4.10)$$

Example 10. With the same data from the Example 9, the first two sequences are classified as TP since the temporal constraints between events belong to those extracted by the chronicle, whereas the third sequence is classified as FP since the events are described by the chronicle but the temporal constraints are not checked. So the precision rate is: $\frac{2}{2+1} = 66.66\%$.

These two previous measurements allow to compute the F-measure as follows:

$$F - measure = \frac{2TP}{2TP + FP + FN}. \quad (4.11)$$

Taking always the same example, the F-measure value will be equal to

$$: \frac{2 \times 2}{2 \times 2 + 1 + 1} = 66.6\%.$$

The literature is plentiful of other measures that use the number of true negatives such as specificity measure (also called the True Negative Rate). It measures the proportion of negatives that are correctly identified. In our application case, we can not apply them because a sequence is classified as a true negative if the chronicles are able to predict normal cases for a sequence where there is no failure. This class can not be used in our case since all the available sequences lead to failures. Furthermore, a chronicle is made to predict a failure and not the normal operation of a machine.

The prediction approaches that use chronicles for prediction are limited. In this thesis, we compare our approach to FADE (Sellami et al., 2019). As for FCP (Ben Chrayet, Samet, and Bach Tobji, 2020), FADE classifies the sequence and predicts when it is going to happen using time constraint of the chronicle failure event. For these reasons FADE is a natural comparative reference to FCP. In addition, we adapted the k -NN algorithm introduced in (Fan, Ye, and Chen, 2016) for evaluation. In our adapted version, we choose the k most similar chronicles to our sequence among the chronicles that cover it. So we do not consider all chronicles, just the k nearest chronicles that correspond to the top 30% of the covering chronicles. Second, we combine the obtained classes using the weighted majority vote method, so the weight of a class is proportional to the distance between the sequence and the chronicles that represent the class in question. Knowing that, our approach uses mined patterns, classifies sequences and predict time to failure. We also compare FCP to other neural network based classification approaches (Tax et al., 2017). Table 4.7 shows the results in terms of recall, precision and F-measure on the SECOM data set.

This experiment showed that the FCP outperforms all adapted classifiers. First FCP outperforms FADE since FADE was initially introduced to find failure sequences. When failure is predicted using FCP, the algorithm returns the merged intervals of failing chronicles which is a large time interval comparatively to FADE. Knowing that the SECOM dataset is a 104 sequences built from the timestamped data, it is limited to train neural network approach like (Tax et al., 2017). Additionally, the FCP considers both assumptions that

TABLE 4.7: Quality of prediction on the SECOM data set

Approach	Parameters	Recall	Precision	F-measure
FCP(Ben Chrayet, Samet, and Bach Tobji, 2020)	minsup=0.4	0.78	0.81	0.79
FADE(Sellami et al., 2019)	minsup=0.4	0.72	0.70	0.71
k -NN(Fan, Ye, and Chen, 2016)	k equivalent to 30% of chronicles	0.69	0.71	0.69
LSTM(Tax et al., 2017)	1 shared layer; 2 prediction layers	0.73	0.74	0.73

the sequence could be normal or abnormal one. The model try to capture the characteristics of both situations which make it more performing.

4.4 Conclusion

In this chapter, we investigate the use of data mining techniques and more specifically, pattern mining algorithms for predictive maintenance. We study a rich pattern called chronicle and we provide mining algorithms to extract chronicles from sequential data. Out of high number of mined chronicles, we select those that describe events ending with a failure. We provide three different approaches to predict failure from event sequences. A first algorithm considers the chronicle with the highest support. A second approach deals with extra-information given by an expert in a chronicle format. Finally, the third approach combines information when several chronicles match a input sequence.

Chapter 5

Patent Mining for Inventive Design Modelling

Contents

6.1	Introduction and related works	108
6.1.1	Literature review of battery ageing analysis	108
6.1.2	Battery ageing datasets	110
6.1.3	Motivation and challenges	111
6.2	AE-LSTM based model for SOH prediction	112
6.2.1	Data pre-processing	113
6.2.2	Encoding branch	114
6.2.3	LSTM branch	115
6.2.4	Smoothing	115
6.3	Window exogenous LSTM for SOH prediction	116
6.3.1	Feature extraction	116
6.3.2	Feature selection	118
6.3.3	Sliding window approach for SOH prediction	118
6.4	SoHAP: SoH explainable prediction based on SHAP	119
6.4.1	Convolutional Neural Network used in SoHAP	120
6.4.2	Post-hoc explainability	121
6.5	Experiments	123
6.5.1	SOH forecasting experiments	124
6.5.2	Comparison with other approaches	127
6.5.3	Explainability analysis	130
6.6	Conclusion	133

Introduction

In this chapter, we are interested in mining inventive solutions using TRIZ theory. As detailed in Chapter 3, we investigate neural networks models to mine pertinent data to users. Three main concerns are addressed in this chapter: (i) How to mine contradictions essential for any TRIZ resolution problem using neural approaches; (ii) How to link the contradiction to a plausible solution provided in patent documents; (iii) Finally, how to rank the inventiveness of mined solutions using multiple criteria.

5.1 Full Framework of TRIZ Patent Inventive Design

In order to ensure TRIZ-based inventive design mining, we adopt a multi-stage strategy as shown in Figure 5.1. Patents are downloaded from USPTO¹ and stored in a database. From these patents, we focus on *state of the art* sections in which contradictions are exposed and solutions are shown. Therefore, in this chapter, the contributions are threefold. At first, we mine patents' contribution using two different architectures of neural networks. We assume that the invention of a patent is centred on its TRIZ contradiction. Therefore, assuming a user formulated contradiction, an analyzed patent could provide an inventive solution if they share the same contradiction. Therefore, as a second contribution, we match patents by using the similarity-based approach. Finally, we rank the patents according to their inventiveness by using several criteria in a Multi Criteria Decision Analysis (MCDA) approach.

5.2 Mining Patent TRIZ Contradictions

In this chapter, the focus is on patents' state of the art. Indeed, it is the part that details what is at stake in invention. The state of the art section presents, usually, the goal of the invention and the difficulties encountered by the prior-art current solutions. In order to extract the contradiction between parameters solved by the patent, several scenarios are then possible. A starting problem can be presented like this: "*When primary antioxidants, such*

¹<https://www.uspto.gov/patents>

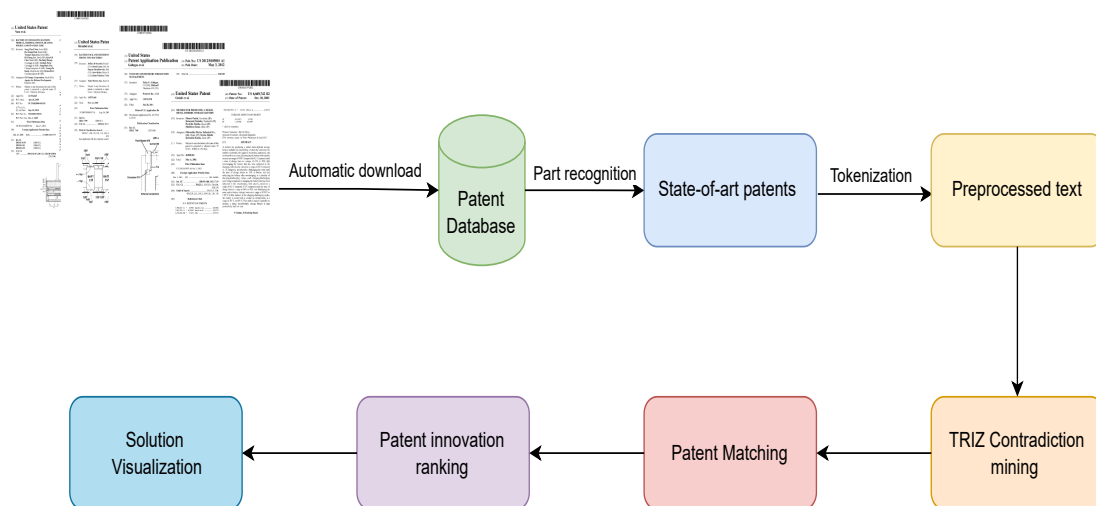


FIGURE 5.1: The full process for mining inventive designs

as hindered phenols, are utilized, polymers can have a more yellow color than unstabilized polymers, therefore decreasing the commercial value of polymers". This sentence contains a parameter that worsens, in this case, the commercial value. Solutions to this problem are then presented but they have disadvantages such as: "Applications employing phosphite additives can result in a reduced quality in the physical properties of polyolefins". In this second sentence, a second parameter is recovered which can be degraded by the application of the solution. Therefore, the solution mentioned in the state of the art allows the improvement of the "commercial value" parameter but leads to the degradation of the "physical properties" parameter. Consequently, there is a contradiction between these two parameters. In the same way, if the parameter to be improved is presented in the state of the art, "This necessitates proper positioning of the upwardly-extending portion of the below-ground structure". The disadvantages of the solutions of the prior art form a contradiction with this parameter as, for example, the risks of injuries: "Therefore, repetitively placing, removing and re-placing such device in the process of determining how best to complete the upper end of the upwardly-projecting portion of the below-ground structure involves considerable physical strain and accompanying risks". Finding this contradiction makes it possible to formulate the problem in a TRIZ theory manner and allows an in-depth description of the object of the invention which, by hypothesis, responds to this (or these) contradiction(s). The sentences containing the parameters to be improved or the parameters involved in the initial problem are associated with the **first part of the contradiction**. The disadvantages of the solutions of the prior art constitute the **second part**

of the contradiction. This choice to separate the two parts of the contradiction is motivated by the fact that the meaning of the contradiction must be detected. Indeed, Altshuller’s matrix presenting the inventive principles as a function of the contradictions between parameters is not symmetrical. Therefore, if a parameter A is in contradiction with a parameter B , the inventive principles that are statistically the most used and therefore the types of solutions are different.

5.2.1 Baseline Approach

The extraction of contradiction sentences is close to an automatic summarization task. This is the reason why our model is based on an extractive summarization model (Guarino et al., 2020), called SummaTRIZ, as it is shown in Figure 5.2. BERT encoder is used in this model. BERT takes as input a series of tokens. Each sentence is separated by a special token [SEP]. Another special token [CLS] is used to represent each sentence. BERT also takes in input indicators of positions of tokens and sentences called Positional Embeddings and Segment Embeddings. The input embeddings then pass into a series of Transformer layers based on attention mechanisms that allow extracting the salient information.

BERT allows building a contextual representation of each input token. These representations integrate a maximum of information coming from adjacent tokens and can then be used for many automatic language processing tasks such as summarization tasks as it is the case here.

A Transformer layer on top of BERT takes as input the token representations [CLS]. The used Transformer on a given sentence representation allows to have global attention on the whole sequence even if the latter was longer than the 512-token limit for BERT. The output of this last layer is thus a contextual representation of each sentence. The limit is set to 1,500 tokens to fit the length of the patent’s state of the art parts.

The baseline model is pre-trained on an extractive summarization task of press articles with the CNN/DailyMail dataset (Hermann et al., 2015). The objective is to train the attention layer above BERT so that it is able to build a representation containing the essential information in the input sequences. Indeed, the training of this layer requires a large variety of documents. Data are cleaned and tokenized using **Stanford-Core-NLP** tokenizer before the standard BERT’s tokenizer. This process was used to pre-train the baseline model on CNN/DailyMail dataset and it was therefore reused for patent processing so that the pre-trained model operates in the same conditions.

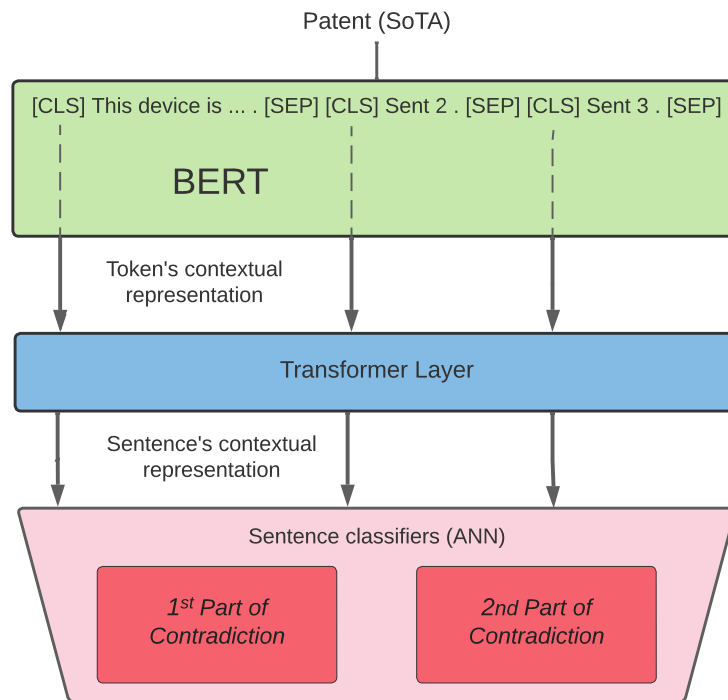


FIGURE 5.2: Baseline approach: SummaTRIZ (Guarino et al., 2020)

Two 2-classes ANN classifiers are used, on top of the Transformer layer, to predict whether each sentence belongs to the first part of the contradiction, the second part of the contradiction, or none of them. Two different classifiers are used because there is a non-zero probability that a sentence contains the whole contradiction and thus should be classified as the first and second part of the contradiction at the same time.

5.2.2 PaGAN: Patent Generative Adversarial Network

The architecture of PaGAN is based on Guarino et al.'s model (Guarino et al., 2020) described above in subsection 5.2.1. BERT encoder allows to build context-wise representations of sentences which are then classified as *first part of contradiction*, *second part of contradiction* or as neutral sentence. For each part of the contradiction, a classifier with two output neurons O_{i0} , O_{i1} is used (with $i \in \{1, 2\}$ the part of the contradiction). The higher the output score of O_{i1} the higher probability that this sentence describes a contradiction.

We improve the model through two contributions. The first contribution is the addition of a document-level classifier, described in subsection 5.2.2.1, that allows predicting whether a document contains a contradiction. The document classifier validates sentences processed by the sentence classifiers by analyzing the whole document. The document classifier also prunes

all the documents without contradictions. A mass of several millions of unlabeled patents is available. The possible use of unlabeled data led us to explore the potential of semi-supervised training to improve baseline results. Our second major contribution is the implementation of a Generative Adversarial Network (GAN) to optimize the model's fine-tuning via semi-supervised learning and cope with training dataset size.

5.2.2.1 Document-based approach for contradiction mining

The limitation of the baseline model is that it does not predict whether a document contains a contradiction or not. Not all analyzed patents contain the TRIZ contradiction. Therefore, the addition of a document-level analysis would allow to automatically prune the patents that do not contain a contradiction.

We propose four different models for this part which cover the main tendencies in document classification: a probabilistic model, a model based on a recurrent network, a Transformer-based model, and an ANN-based model.

Probabilistic model

In this model, we consider that the belonging of a sentence S_i to either the first part or the second part of the contradiction are independent events and thus:

$$P_c(S_1...S_n) = \max_{1 \leq i \leq n} (P_{c1}(S_i)) * \max_{1 \leq i \leq n} (P_{c2}(S_i)) \quad (5.1)$$

with $P_c(S_1...S_n)$ the probability that a contradiction is in a sentence sequence $S_1...S_n$. $P_{c1}(S_i)$ is the probability that sentence i is the first part of the contradiction and $P_{c2}(S_i)$ the probability that sentence i is the second part of the contradiction.

Recurrent model

The probabilistic model is limited in performance because of its postulate which is too strong. A more global analysis of the document seems therefore necessary. The main difficulty encountered is the variable length of the documents. One possibility is to use recurrent networks. A LSTM (Hochreiter and Schmidhuber, 1997) or a GRU (Chung et al., 2014) is used for this model. These recurrent networks take as input a sequence of sentence representations, i.e, in this case, the representations of all sentences in the state of the art. The LSTMs contain a memory vector called Cell State which allows them to select the "useful" information during the iterations. The cell state is modified at each inference using Equation 3.1.

In our case, the "useful" information will be the features related to the presence or not of a contradiction. The features are learned by BERT encoder, so the recurrent network only has to extract the right ones. This is why only one unidirectional cell of LSTM is used. This allows limiting the number of parameters without affecting the results.

Transformer model

The recurrent model has several drawbacks such as its run time or the loss of information due to successive iterations. A transformer layer does not have these drawbacks. In this model, the transformer layer takes the sequence of sentence representations as input and its first output is the document classification score.

ANN model

The recurrent neural network and transformer models do not take into account the results related to the classification of sentences. We, therefore, present a last model based on a multi-layer perceptron. The idea is to keep only decisive information in the decision-making concerning the existence of a contradiction. In this case, because the encoder allows building a very precise contextual representation of the sentences, only the representations associated with the two sentences forming the contradiction are necessary for the decision making. A multi-layer perceptron is used for the classification of the document from the representations of the two sentences having the maximum probabilities of belonging to the contradiction (one sentence for the *first part* and one sentence for the *second part of the contradiction*). Thus, the probability $P_c(S_1...S_n)$ that a contradiction is present in the document becomes:

$$P_c(S_1...S_n) = ANN(\arg \max_{1 \leq i \leq n}(P_{c1}(S_i)), \arg \max_{1 \leq i \leq n}(P_{c2}(S_i))). \quad (5.2)$$

The advantage of this approach is that the model has less chances of overfitting despite the small amount of data since, for the same documents, its inputs vary during the learning process. Indeed, as the sentences scores evolve during the training, for a same document, the selected sentences which go through the ANN model change. The approach makes the encoder integrate the contradiction information in all "main" sentences which make the ANN decision easier.

5.2.2.2 GAN and semi-supervised training

Semi-supervised learning aims to improve the generalization of a model using unlabeled data. It also improves the quality of the representations generated by the model (Weston, Ratle, and Collobert, 2008; Yang, Cohen, and Salakhutdinov, 2016; Kipf and Welling, 2017). Generative Adversarial Networks were introduced by Goodfellow et al. (Goodfellow et al., 2014). The purpose of GANs is to generate new data close to a target data distribution. The GANs were then adapted to semi-supervised learning (Salimans et al., 2016). Croce et al. (Croce, Castellucci, and Basili, 2020) showed the efficiency of this method for sentence classification.

A model called generator G generates dummy data and another model called discriminant D tries to distinguish the data generated among real but unlabeled examples.

The generator G learns to map input noise variables z to the real data distribution p_{data} . Its goal is therefore to minimize $\log(1 - D(G(z)))$. The discriminant, on the contrary, tries to maximize $\log(1 - D(G(z)))$ while associating the right labels to the real data x , i.e. maximizing $\log(D(x))$.

D and G play therefore a two-player minimax game with value function $V(G, D)$:

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (5.3)$$

Model training

An adversarial training for the sentence-level classifiers is implemented as described by Salimans et al. (Salimans et al., 2016). An additional class is created to introduce the probabilities of input data fitting to the target distribution p_{data} . Sentence classifiers have therefore three output neurons: the initial contradiction's classification neurons and another neuron which outputs the probability for the document to be fake (see Figure 5.3). The "contradiction" neurons are therefore involved in the supervised loss (D_{sup_1} or D_{sup_2}) for the two contradictions' classification classes C and \bar{C} (the sentence belongs to the part i of the contradiction or not, Equation 5.4) while the "adversarial" neuron is involved in the unsupervised losses for the *fake* F and *real* \bar{F} classes (Equations 5.5 and 5.6).

$$D_{sup_i} = E_{x, y \sim p_{data}} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} \in (C, \bar{C})))] \quad (5.4)$$

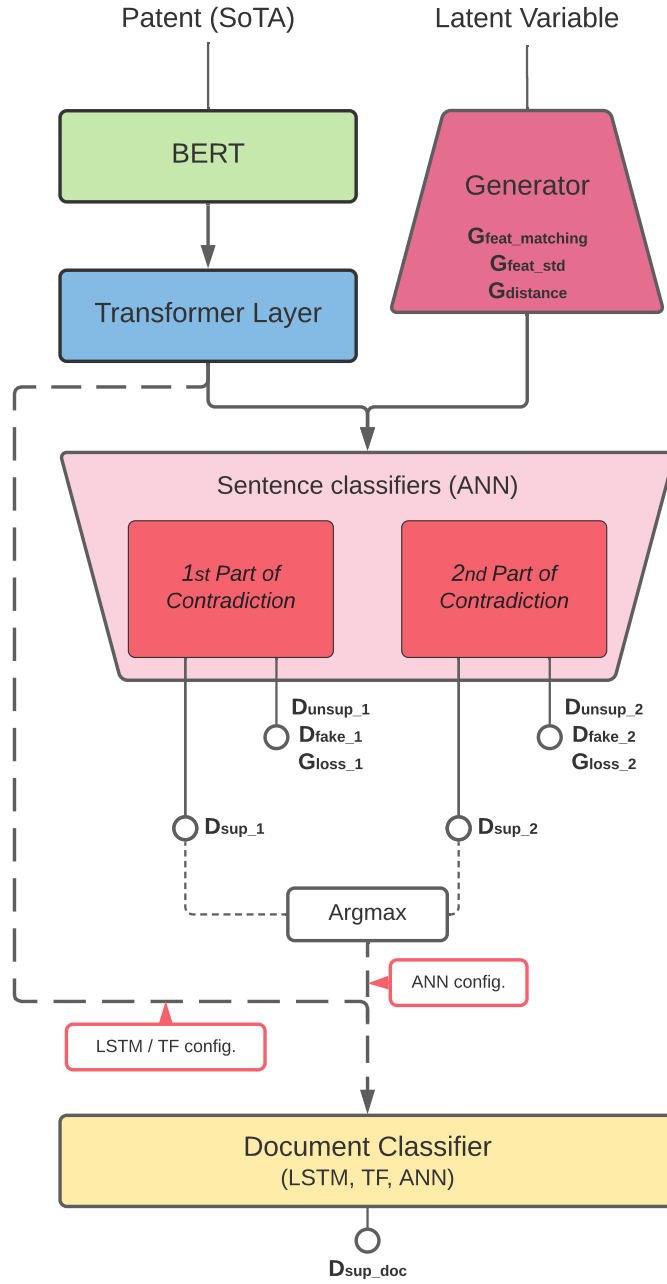


FIGURE 5.3: PaGAN architecture

$$D_{unsup_i} = E_{x \sim p_{data}} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = \bar{F}))] \quad (5.5)$$

$$D_{fake_i} = E_{z \sim p_z} [-\log(P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))] \quad (5.6)$$

D_{unsup} relates to how good is the model at classifying the real data as real data. D_{fake} relates to how good is the model at classifying the fake data as

fake data. The sentences used for the unsupervised losses come from unlabeled patents. As the losses are computed at a sentence level, not many documents are needed (we only use 1/10 of the number of labeled documents). D_{unsup} is therefore back-propagated through the Transformer layer and BERT encoder. This implies that the encoder will learn to integrate new information in the sentence representations to allow better recognition of the "real" data. This richer representation of patents' sentences induces an easier classification for the contradiction which is our main purpose. This mechanism also allows minimizing overfitting. This is even amplified by the fact that two non-supervised losses are computed and back-propagated at the same time (one for each classifier).

The document classifier has only one output neuron that makes the decision between documents that contain a contradiction (D_c class) and those that do not ($\overline{D_c}$ class). It is thus involved in a single supervised loss:

$$D_{sup_doc} = E_{x,y \sim p_{data}}[-\log(P(\hat{y}_d = y_d | x, y_d \in (D_c, \overline{D_c})))] \quad (5.7)$$

As the number of labeled documents is lower than the number of labeled sentences, the document classifier tends to learn faster than the sentence classifiers. An experimental coefficient of 0.1 is therefore applied to D_{sup_doc} so that the learning curves of all classifiers can match.

Generator architecture and training

A few different architectures were implemented for the generator: a fully connected network, a LSTM, and a transformer. The comparison of these different architectures is shown in section 5.5.

The generator, in our case, creates sequences of sentence representations that are plausible in order to deceive the sentence-level classifiers but, as BERT uses the context to encode the sentences, the generator must also produce a plausible context. The loss associated with the generator combines both losses from sentence-classifiers (G_loss_1 and G_loss_2 in Figure 5.3):

$$G_{loss} = \sum_{i=1}^2 E_{z \sim p_z}[-\log(1 - P(\hat{y}_{s_i} = y_{s_i} | x, y_{s_i} = F))]. \quad (5.8)$$

To ensure that the generator converges to the right distributions, we add two feature matching losses. G_{feat_mean} (Equation 5.9) ensures that the generated representations is close to the training data and G_{feat_std} (Equation 5.10) also ensures that the generated documents are not a single representation of a

sentence repeated n times. We, thus, add a constraint on the variety of representations in the same document.

$$G_{feat_mean} = \left\| E_{x \sim p_{data}(x)} f(x) - E_{z \sim p_z(z)} f(G(z)) \right\| \quad (5.9)$$

$$G_{feat_std} = \left\| \sigma_{x \sim p_{data}(x)} f(x) - \sigma_{z \sim p_z(z)} f(G(z)) \right\| \quad (5.10)$$

Finally, to bring variety in the generated documents, we introduce a minimization of the similarity between generated documents via a cosine similarity measure:

$$Cos_{similarity}(A, B) = \frac{A \cdot B}{\|A\| \|B\|} \forall A, B \in \mathbb{R}^n. \quad (5.11)$$

At each iteration, a constant number of documents is generated but they include a random number X of representations with $X \sim \mathcal{N}(\mu, \sigma^2)$ and $\mu = 8, \sigma = 2$. All the documents generated at an iteration i have nevertheless the same length X_i . The computation of the similarity between documents can be done relatively easily by using the similarity between sentences. The similarity between two documents is, thus, defined as the sum of the similarities between the sentences of these documents.

In practice, this additional strong constraint, since it forces the representations to be very different, does not interfere with the convergence of the generator while avoiding mode collapse.

5.3 SAM-IDM for Inventive solution mining

To predict accurately similar problems from different domains, the model's ability of learning long context information has to be considered. Specifically, patent documents contain very long sentences compared to generic texts. In this section, in order to improve this ability, we introduce a novel retrieval model named SAM-IDM in detail. Similarity-based approach for merging IDM-related knowledge (SAM-IDM) aims at preparing inventive solutions by extracting similar contradictions from a large number of patent documents.

At the first step, a period of several patent documents from USPTO are randomly chosen to build a patent database in XML format. PaGAN is then used to extract contradictions from input patents in the second step. A third step consists in designing a reduction strategy to assign IDM-related knowledge into different groups according to different domains they belong to.

It aims to only access inventive solutions of level 3. Altshuller separated the patents' different degrees of inventiveness into five levels. Level 3 designates finding a solution outside the studied domain. This reduction strategy is also able to enhance the performance of the sequential similarity computation by decreasing the search set. After that, in the fourth step, the trained MaLSTM (Mueller and Thyagarajan, 2016) is used to learn sentence semantic meanings in order to predict similarity among different domains contradictions. Several similar contradictions are eventually listed and ranked via their similarity scores. Corresponding partial solutions towards different domains problems can be seen as potential inventive solutions of the input contradiction. Our SAM-IDM approach follows these steps:

- **Reduction of IDM-related Knowledge Set** : SAM-IDM analyses a large number of contradiction sentences. This could lead to a significant performance issue for computing similarity among sentences. Indeed, if we compare this considered contradiction to all of the remaining contradiction sentences, computation time increases drastically. In industrial context, the extensive computation is a major waste of time drawback for companies and engineers. Therefore, following the hypothesis, we specifically divide IDM-related knowledge into several groups according to the different domains they belong to, so that the computational consumption among the same domains patents can be avoided. In detail, we compare the considered contradiction Po_{ij} of patent i from $Domain_{d-1}$ with other contradictions which are from different domains like $Domain_d$. Since several domains tend to contain a much larger number of patents, the proportion of contradiction is therefore high.
- **Manhattan Long Short Term Memory Networks for Similarity Measure of Problem Sentence**: In SAM-IDM, Manhattan LSTM (MaLSTM) (Mueller and Thyagarajan, 2016) is applied to perform the semantic similarity computation among different contradictions. As shown in Figure 5.4, MaLSTM consists of two identical LSTM networks (LSTM $Encoder_1$ and LSTM $Encoder_2$). Same weights are shared in these two LSTM networks to decrease the training time of the model, because of its siamese recurrent architecture. In MaLSTM, the identical sub-network LSTMs is able to learn representations of contradiction sentences Po_{ij} via sequences of word vectors x_T . Furthermore, the word embedding approach provides the semantic meaning to each word

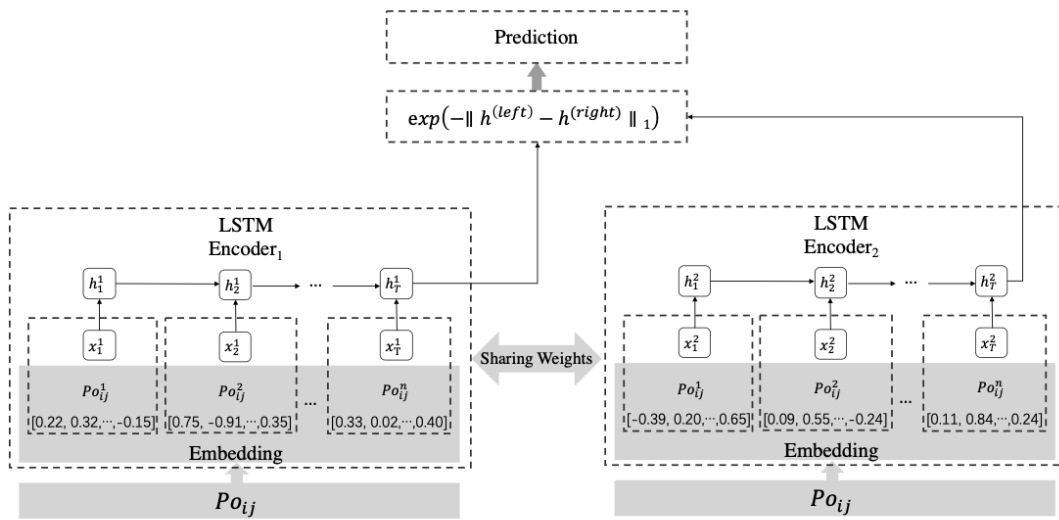


FIGURE 5.4: The structure of MaLSTM developed by (Mueller and Thygarajan, 2016)

Po_{ij} of the problem sentence in a vector representation. We apply labelled Quora dataset containing a large number of similar Quora question pairs to feed LSTMs in order to let it learn hidden semantic representations between labelled similar pairwise sentences. Similarity of the representation space is subsequently used to infer the underlying semantic similarity of the sentences. The Manhattan distance is eventually used to measure semantic similarity among contradictions Po_{ij} from different domains patents.

5.4 Inventive solution mining using Multiple-Criteria Decision Analysis (MCDA)

As illustrated in Figure 5.5, an inventive solutions-ranking model called PatRIS is proposed. In detail, different domains patents P_{at} firstly flow to SAM-IDM model based on LSTM neural networks to achieve similar pairwise contradictions set \mathcal{P} for the target contradictions P_{target} and corresponding SV. After that, IDM-Matching based on XLNet neural networks retrieves solutions \mathcal{S} for given similar problems according to the context information of patent documents. We see these solutions from different domains patents as potential inventive solutions for the target problem.

However, with a large number of patents input, several inventive solutions with the same similarity value might be also generated. It leads to the

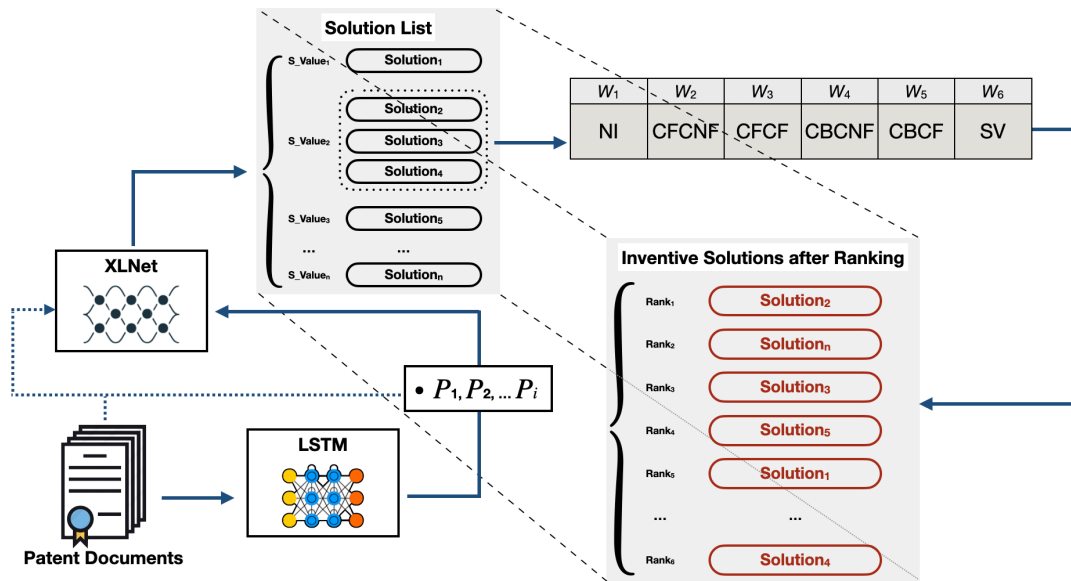


FIGURE 5.5: The framework of PatRIS

TABLE 5.1: Inventive Solutions from Different Domains

Input sentence	Similar Problems	Patent Number	Similarity Value	Domain	Inventive Solutions
Patent Number: US9534284 Domain: C the second metal layer is not provided	the web page is not captured normally	US9535571	0.86	G	associated with a broadcasting application according to an exemplary embodiment of the present invention
	if the wfe instruction is not intended for agent discovery purposes	US9535772	0.83	G	are accessible to the agent as well as the client, such as designated locations in a memory 104
	the scope of the first aspect is not limited to these examples	US9537403	0.86	H	a slow DAC 930, the gear shift can be made gradual. Alternatively, by using a fast DAC, the gear shift can
	the message is not received in step	US9537998	0.86	H	step 301. Alternatively, when the message is transmitted, the mobile

obstacle of ranking solutions only with their similarity values. As shown in Table 5.1, three latent inventive solutions with the same similarity value 0.86 are generated from the real-world U.S. patent sample. We assume that the inventiveness of potential solutions is related to the inventiveness of corresponding patents from which they are extracted. Patent inventiveness indicators as well as the Similarity Value indicator (SV) are leveraged by PatRIS to build the inventive solution ranking model according to the multiple-criteria decision analysis approach. It is able to eventually rank these inventive solutions according to patent features and semantic similarity.

Moreover, as a sub-discipline of the operation research, Multiple Criteria Decision Analysis (MCDA) is able to explicitly evaluate multiple criteria in decision making in order to help understand the inherent trade-off (Greene et

al., 2011). PatRIS is based on the MCDA approach named TOPSIS (Greene et al., 2011) and integrates five patent indicators and semantic similarity value (SV) as the sixth criterion to build a ranking system. Ranking problems could be formulated as the function 5.12. It is designed to maximize the inventiveness of solutions under the selected indicators in order to rank them.

$$\begin{aligned}
 \max f(x) &= (F_1(x), F_2(x), F_3(x), F_4(x), F_5(x), F_6(x)) \\
 F_1(x) &= NI(x_1, x_2, \dots, x_n) \\
 F_2(x) &= CFCNF(x_1, x_2, \dots, x_n) \\
 F_3(x) &= CFCF(x_1, x_2, \dots, x_n) \\
 F_4(x) &= CBCNF(x_1, x_2, \dots, x_n) \\
 F_5(x) &= CBCF(x_1, x_2, \dots, x_n) \\
 F_6(x) &= SV(x_{sv1}, x_{sv2}, \dots, x_{svn})
 \end{aligned} \tag{5.12}$$

subjected to linear constraints:

$$x_i \geq 0, i = 1, \dots, n \tag{5.13}$$

$$x_{sv_i} \in [0, 1], i = 1, \dots, n \tag{5.14}$$

In (5.12)-(5.14), $NI(x)$, $CFCNF(x)$, $CFCF(x)$, $CBCNF(x)$, $CBCF(x)$, and $SV(x_{sv})$ are count of the number of inventors, cited-forward citations with no family, cited-forward citations with family, cited-backward citations with no family, cited-backward citations with family, and semantic similarity respectively. x stands for the vector of the count variable, 0 presents the lower bound of the i^{th} count variable. The linear constraint comes from the consideration of patent cost expenditure (number of inventors), peer recognition (number of backward citations), relevant knowledge references (number of forward citations), and semantic distance (similarity value). As illustrated in Table 5.2, a real-world sample of U.S. patents, the count distribution of indicators has been listed. The higher the number of inventors, the higher the cost invested in the patent. A high number of forward citations shows that the invention contains a high technical impact. A high number of backward citations means that the innovation tends to cite a large number of, and range of, scientific publications. A high similarity value indicates a short distance between the corresponding problem of solution and the target problem. These six criteria are therefore positively related with the corresponding values x_{ij} . PatRIS ranks the mined solutions from the most innovative to the

TABLE 5.2: A sample of the indicator detail

Patent Number	NI	CFCNF	CFCF	CBCNF	CBCF	SV
US9535571	1	1	22	18	0	0.86
US9537403	3	3	14	15	0	0.86
US9535772	2	0	2	4	0	0.83
US9537998	2	0	2	9	0	0.86

least innovative. Different weights w_j are also assigned to these criteria.

As illustrated in formula 5.15, we normalize each value so that all attributes are set in the same range: j -th feature $F_j = \{x_1, x_2, \dots, x_i\}$, $i \in \{0, n\}$ derives from six criteria, and x_{ij} is the value of the i -th solution under the j -th feature. After normalization, values x_{ij} of patent indicators reaching 1 indicate that the corresponding patents are more innovative in the j -th feature F_j than other patents. When x_{svij} is closer to 1, the solution might be more innovative. Attribute weights are also applied to the corresponding values.

$$\text{Normalization}(x_{ij}, F) = \frac{x_{ij}}{\text{sum}(F)} \quad (5.15)$$

TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) with other two typical types of MCDA approaches, namely Weighted Sum approach and Weighted Product approach, are listed below.

1. Weighted Sum (WS):

$$\text{Score}_i = \sum_j x_{ij} \times w_j \quad (5.16)$$

2. Weighted Product (WP):

$$\text{Score}_i = \prod_j x_{ij}^{w_j} \quad (5.17)$$

3. TOPSIS: It aims at selecting the alternative that contains the longest geometric distance from the negative ideal solution and the shortest geometric distance from the positive ideal solution. Thus, the optimal goal is to approach the best alternative and stay away from the worst alternative. The main steps are as follows:

Step1: Realize the normalized matrix.

$$\bar{x}_{ij} = \frac{x_{ij}}{\sqrt{\sum_i x_{ij}^2}} \quad (5.18)$$

Step2: Assign weights to each value in order to compute the weighted normalized matrix.

$$xw_{ij} = \bar{x}_{ij} \times w_j \quad (5.19)$$

Step3: Compute the ideal best to mark the attribute as containing the positive impact.

$$idealbest_{positive_j} = Max(xw_{ij}) \quad (5.20)$$

$$idealworst_{positive_j} = Min(xw_{ij}) \quad (5.21)$$

Step4: Compute the Euclidean distance of $idealbest_{positive}$ and $idealworst_{positive}$, respectively.

$$Best_i = \sqrt{\sum_j (xw_{ij} - idealbest_{positive_j})^2} \quad (5.22)$$

$$Worst_i = \sqrt{\sum_j (xw_{ij} - idealworst_{positive_j})^2} \quad (5.23)$$

Step5: Compute $Score_i$ and use it to rank inventive solutions.

$$Score_i = Worst_i / (Best_i + Worst_i) \quad (5.24)$$

5.5 Experiments

5.5.1 Contradiction mining experiments

Contradictions mining relates to a classification problem. This is why the classical classification metrics (Accuracy, Precision, Recall, F1 score) are used for the evaluation. However, they are not sufficient to fully evaluate the mining performance. New metrics are therefore introduced, among them, a few are inspired from extractive summary metrics.

The metrics S and S_m are used to evaluate sentence-level analysis. A labeled document includes n_1 sentences for the first part of the contradiction and n_2 sentences for the second part of the contradiction. The metric S for a labeled document D is defined as the number of correct sentences for the first/second part of contradiction in the n_1/n_2 sentences with the maximum probabilities of belonging to the *first/second part of contradiction*. S_m evaluates

the number of correctly extracted sentences in the $n_1 + margin$ or $n_2 + margin$ best sentences depending on the part of the studied contradiction (we take $margin = 2$ for the experiments). These metrics allow us to assess the importance of the sentences that are extracted.

We call $E_1 = \{S_{10} \dots S_{1n_1}\}$ the set of the n_1 sentences labeled as *first part of contradiction* and $E_2 = \{S_{20} \dots S_{2n_1}\}$ the set of the n_2 sentences labeled as *second part of contradiction*. Each pair $S_{1i}S_{2j}$ forms a contradiction. We consider that a contradiction is extracted if:

$$\arg \max_{1 \leq i \leq n} (P_{c1}(S_i)) \in E_1 \quad (5.25)$$

$$\arg \max_{1 \leq i \leq n} (P_{c2}(S_i)) \in E_2 \quad (5.26)$$

$$P_c(S_1 \dots S_n) > P_{threshold} \quad (5.27)$$

with $P_{c1}(S_i)$ the probability that sentence i is the first part of the contradiction, $P_{c2}(S_i)$ the probability that sentence i is the second part of the contradiction and $P_c(S_1 \dots S_n)$ the probability that there is a contradiction to mine. The CO_{Found} metric evaluates the first two conditions so that it only takes into account the analysis of the sentences. The CO_{Valid} metric is used to evaluate the number of extractions that verify all these conditions, i.e. correct sentences are chosen and the document has a high probability of containing a contradiction. We take $P_{threshold} = 0.5$ for the experiments.

Concerning the hardware and the experimental settings, all the experiments are performed using a four RTX 2080Ti and Intel Core i9-9820X 3.30 GHz machine. Pytorch framework is used for all experiments. Fourfold cross-validation is performed for all experiments (at the best of three trainings). All results, except losses, accuracy, precision, recall, and F1-score are summed up over the fourfolds. Losses are averaged out and the accuracy, precision, recall and F1-score are computed from the summed TP, FP, TN, and FN.

The results of sentences classification are shown in Tables 5.3 and 5.4. Results of document classification are sketched in Table 5.5.

TABLE 5.3: Sentence classification 1 (with LSTM generator)

Model	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
SummaTRIZ _D	0.140	0	0	61959	2276	0.96	0	0	0	548	1158
SummaTRIZ _{TL}	0.115	576	510	61449	1700	0.97	0.53	0.25	0.34	1119	1711
Baseline _{ANN_D}	0.140	0	1	65248	2276	0.97	0	0	0	535	1149
Baseline _{ANN_{TL}}	0.115	575	482	64767	1701	0.97	0.54	0.25	0.35	1098	1710
PaGAN _{PROB}	0.112	575	457	71584	1701	0.97	0.56	0.25	0.35	1168	1736
PaGAN _{ANN}	0.112	532	414	71689	1744	0.97	0.56	0.23	0.33	1187	1760
PaGAN _{LSTM}	0.112	509	368	71704	1767	0.97	0.58	0.22	0.32	1186	1752
PaGAN _{TF}	0.113	649	592	71438	1627	0.97	0.52	0.29	0.37	1143	1759

TABLE 5.4: Sentence classification 2 (with LSTM generator)

Model	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
SummaTRIZ _D	0.171	0	0	60526	3709	0.94	0	0	0	1750	2692
SummaTRIZ _{TL}	0.129	1814	815	59711	1895	0.96	0.69	0.49	0.57	2493	3127
Baseline _{ANN_D}	0.170	0	0	63816	3709	0.95	0	0	0	1766	2692
Baseline _{ANN_{TL}}	0.129	1849	881	62935	1860	0.96	0.68	0.50	0.57	2500	3131
PaGAN _{PROB}	0.120	2091	936	69672	1618	0.97	0.69	0.56	0.62	2619	3226
PaGAN _{ANN}	0.120	2288	1119	69551	1421	0.97	0.67	0.62	0.64	2626	3206
PaGAN _{LSTM}	0.118	2323	1156	69483	1386	0.97	0.67	0.63	0.65	2645	3213
PaGAN _{TF}	0.121	2327	1228	69369	1382	0.96	0.65	0.63	0.64	2631	3220

TABLE 5.5: Document classification (with LSTM generator)

Model	Loss	TP	FP	TN	FN	Acc.	Pre.	Recall	F1 score	CO _{Found}	CO _{valid}
SummaTRIZ _D	-	0	0	1600	1600	0.50	0	0	0	153	0
SummaTRIZ _{TL}	-	386	135	1465	1214	0.58	0.74	0.24	0.36	582	213
Baseline _{ANN_D}	0.529	1274	490	1110	326	0.74	0.72	0.80	0.76	146	96
Baseline _{ANN_{TL}}	0.502	1275	438	1162	325	0.76	0.74	0.80	0.77	580	467
PaGAN _{PROB}	-	335	126	1474	1265	0.57	0.73	0.21	0.33	668	192
PaGAN _{ANN}	0.466	1335	431	1169	265	0.78	0.76	0.83	0.79	666	576
PaGAN _{LSTM}	0.481	1370	507	1093	230	0.77	0.73	0.86	0.79	654	567
PaGAN _{TF}	0.467	1345	427	1173	255	0.79	0.76	0.84	0.80	648	552

TABLE 5.6: Sentence classification 1 (with ANN document classifier)

Setup Generator	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S _m
LSTM _S	0.112	532	414	71689	1744	0.97	0.56	0.23	0.33	1187	1760
FC _S	0.113	575	493	71540	1701	0.97	0.54	0.25	0.34	1151	1760
TF _S	0.116	313	199	71873	1963	0.97	0.61	0.14	0.22	1157	1741
LSTM _{ALL}	0.114	561	438	71672	1715	0.97	0.56	0.25	0.34	1146	1729
TF _{ALL}	0.118	286	170	71863	1990	0.97	0.63	0.13	0.21	1132	1714
LSTM _D	0.112	462	326	71771	1814	0.97	0.59	0.20	0.30	1168	1736
TF _D	0.111	537	387	71730	1739	0.97	0.58	0.24	0.34	1182	1759

TABLE 5.7: Sentence classification 2 (with ANN document classifier)

Setup Generator	Loss	TP	FP	TN	FN	Accuracy	Precision	Recall	F1 score	S	S_m
$LSTM_S$	0.120	2288	1119	69551	1421	0.97	0.67	0.62	0.64	2626	3206
FC_S	0.118	2285	1135	69465	1424	0.97	0.67	0.62	0.64	2630	3231
TF_S	0.130	1735	690	69949	1974	0.96	0.72	0.47	0.57	2602	3192
$LSTM_{ALL}$	0.121	2285	1153	69524	1424	0.97	0.66	0.62	0.64	2639	3217
TF_{ALL}	0.129	1593	587	70013	2116	0.96	0.73	0.43	0.54	2598	3185
$LSTM_D$	0.118	1960	773	69891	1749	0.97	0.72	0.53	0.61	2663	3245
TF_D	0.118	2269	1060	69624	1440	0.97	0.68	0.61	0.64	2637	3231

TABLE 5.8: Document classification (with ANN document classifier)

Setup Generator	Loss	TP	FP	TN	FN	Acc.	Pre.	Recall	F1 score	CO_{Found}	CO_{valid}
$LSTM_S$	0.466	1335	431	1169	265	0.78	0.76	0.83	0.79	666	576
FC_S	0.459	1363	466	1134	237	0.78	0.75	0.85	0.79	632	548
TF_S	0.508	1363	532	1068	237	0.76	0.72	0.85	0.78	630	547
$LSTM_{ALL}$	0.460	1384	467	1133	216	0.79	0.75	0.86	0.80	620	531
TF_{ALL}	0.512	1449	618	982	151	0.76	0.70	0.91	0.79	598	552
$LSTM_D$	0.471	1392	498	1102	208	0.78	0.74	0.87	0.80	641	575
TF_D	0.479	1418	592	1008	182	0.76	0.71	0.89	0.79	653	574

SummaTRIZ (Guarino et al., 2020) is the only existing approach that has tackled contradiction mining. This model only allows sentence classification. In order to compare it to other approaches, we introduce the classification of documents in SummaTRIZ model with a probabilistic model (Part 5.2.2.1). SummaTRIZ_D model is trained on our dataset only. *TL* subscript refers to Transfer Learning; it indicates that the model is first trained on CNN/DailyMail dataset (extractive summarization task) and finetuned with our dataset. For a fairer comparison of the models and to highlight the contribution of the GAN, a SummaTRIZ model with an ANN document classifier (Part 5.2.2.1) is also introduced as *Baseline*. The influence of the document classifier (PROB described in 5.2.2.1, ANN described in 5.2.2.1, LSTM described in 5.2.2.1 and TF described in 5.2.2.1) is also studied with PaGAN. Note that for the sentence classifiers, we use an ANN with a single intermediate layer.

At first, we notice that the dataset alone does not allow reaching a correct level of performance. Indeed, very few sentences have a probability of belonging to *First part of contradiction* or *Second part of contradiction* greater than 0.5, which automatically brings most of the classification metrics to 0. Moreover, the ranking of the sentences made by SummaTRIZ_D and Baseline_{ANN_D} is much worse than that of the other tested setups (other architectures and/or training mode). S and S_m are, in fact, almost twice as low as the other candidates. This means that in the theoretical summaries provided by these models, very few sentences actually contain information about one or more contradictions. The results in terms of document classification are also lower than those of the other setups.

PaGAN shows slightly better sentence classification results for the traditional metrics but also for the new metrics S and S_m with an improvement of respectively 8% and 6% for S and 3% for S_m comparatively to the baseline using transfer learning (Baseline_{ANN_{TL}}). In terms of document classification and the number of contradictions extracted, the difference is much more visible. The loss is about 7% lower for the best configuration of PaGAN. At the level of the contradictions found CO_{found} , in the sense of the good sentences selected regardless of the classification of the document, we see a maximum increase of 15.7% (from 580/1600 to 668/1600). Since document classification is better with GAN, this gap grows even larger when the whole model is considered with both sentence and document classification. Indeed, we observe an increase in the number of contradictions extracted and validated by the document-level classifier of more than 23% to go from 467/1600 to 576/1600

for the best setup of PaGAN. The contribution of adversarial training is thus clearly visible, both in terms of losses and metrics, and whether it is at the level of sentence or document classifications. It is also clear that, even if GAN works only at the sentence level, the learning of better representations by the encoder also has a very positive impact on document classification.

Four different setups of PaGAN were tested with different document classifiers (Probabilistic model, ANN model, LSTM model, Transformer model). The probabilistic approach PaGAN_{PROB} is, unsurprisingly, very poorly performing for document classification since even if an equivalent number of contradictions are found by sentence classification (CO_{found}), very few are actually validated by document classification (less than one third). The other three setups show very similar results. The most important metric is precision since the goal is to limit false positives as much as possible. The approach with the ANN, which presents at the same time high precisions and a number of extracted contradictions slightly higher than the others, seems to be the best choice.

Finally, we studied the impact of the generator architecture as well as the training mode (see Tables 5.6, 5.7, 5.8). Several generator architectures have been implemented: LSTM, Transformer, Fully connected. A study of the impact of adversarial training has also been performed. Initially, only the sentence classifiers are involved in the adversarial training (index S added to the name of the generator architectures in tables 5.6, 5.7, 5.8). In another configuration, we integrate the document classifier to the discriminant which means that we add a term to the unsupervised losses for the discriminant and the generator (index ALL added to the name of the generator architectures in Tables 5.6, 5.7, 5.8):

In a final setup, we remove the terms linked to the sentence classifiers so that only the document classifier plays the role of discriminant (index D added to the name of the generator architectures in Tables 5.6, 5.7, 5.8). The Fully Connected FC generator is only a sentence generator. It takes as input a latent vector and generates a representation. Therefore, only the results of the first training configuration (adversarial training on sentence classifiers) are shown. The LSTM and the Transformer generators generate entire documents with linked sentence representations. That's why document-level adversarial training is relevant.

Tables 5.6, 5.7 show that the architecture of the generator has an effect on the sentence classification metrics. The Transformer-based generators have high precisions. Indeed, for the *first part of contradiction*, Transformers reach a

precision of 0.61 on average against 0.57 for LSTM-based generators and 0.54 for the FC generator. For the *second part of contradiction* the average precision of Transformers is 0.71 against 0.68 and 0.67 for the LSTMs and FC generators). LSTMs show higher recall (0.23 for the *first part of contradiction* and 0.59 for the *second part of contradiction* versus 0.17 and 0.50 for the Transformers). The fully connected generator shows, as for LSTMs, a low precision but a high recall. Based on the sentence classification, the Transformer seems to be the best architecture since it limits the maximum number of false positives.

However, the observation is different for document classification (Table 5.8). Indeed, the LSTMs based and FC architectures show the best precisions (0.75 on average against 0.71 for the Transformers generators) while the Transformers have higher recalls. The document classification is important because it validates the sentence selection. Therefore, the best choice of architecture, validated by the metrics associated with the contradiction extraction CO_{found} and CO_{valid} is an LSTM.

The influence of the training mode is also visible since sentence-level training (*S*) goes for better recall for sentence classification and better precision for document classification and vice versa for document-level training (*D*). The dual-level training *ALL* presents a compromise between the two but it does not improve the precision of document classification compared to the *S* training. Therefore, the latter appears to be the best training mode.

5.5.2 Contradiction matching experiments

We tune our SAM-IDM approach on the Quora set and use the grid search to determine the optimal parameters. In the MaLSTM part, two identical LSTM neural networks share the same configuration with the same parameters and weights. Updating of parameters is mirrored across both subnetworks. In order to capture and learn the semantic similarity of input pairs of sentences, a pre-trained 100-dimensional Word2vec model (Ni, Samet, and Cavallucci, 2019) on training an open-source Wikipedia dataset is initially used to obtain word embeddings as inputs of LSTM. More dimensions usually mean a slower training speed and may contribute to an overfitting problem. For LSTM, we select the epochs among {25, 50, 75, 100}, batch size among {500, 1000, 1500, 2000} due to a large size of training dataset. Besides, for other parameters, we separately select validation split as 0.1, rate of drop as 0.17, rate of drop dense as 0.25, number of LSTM layers as 50, number of dense units as 50, and activation function as ReLU. Specifically, Quora dataset has 403,459 labelled pairwise sentences. Among them, 363,114 pairwise sentences are

TABLE 5.9: Similarity matching experimental results on the labelled SNLI dataset (left) and the labelled patent sentence pairs (right) for different threshold values

Precision	Threshold													
	0.6		0.65		0.7		0.75		0.8		0.85		0.9	
	SNLI	Patent	SNLI	Patent	SNLI	Patent	SNLI	Patent	SNLI	Patent	SNLI	Patent	SNLI	Patent
BOW	62.45%	73.77%	60.24%	72.25%	60.04%	68.06%	61.24%	61.99%	62.25%	59.94%	61.24%	44.51%	62.05%	35.05%
TF-IDF	62.05%	72.70%	62.25%	67.88%	61.85%	66.63%	60.84%	51.65%	60.04%	49.86%	61.45%	42.01%	61.85%	26.27%
Word2vec	43.78%	71.36%	45.38%	70.47%	47.19%	72.79%	49.60%	67.26%	56.22%	61.73%	61.85%	58.70%	62.45%	60.74%
MaLSTM	70.28%	70.12%	71.49%	75.38%	73.69%	75.28%	72.49%	76.89%	81.73%	77.88%	76.51%	72.26%	77.11%	72.52%

used as training dataset and 40,345 pairwise sentences as validation dataset. From the official notification, the ground truth labels are inherently subjective, as the true meaning of sentences can never be determined with certainty. Human labeling is also a ‘noisy’ process, and reasonable people will disagree. As a result, this large size of labelled dataset may include many incorrect labellings. Therefore, instead of avoiding the overfit problem when we train the LSTM, we also try to find a balance point for parameters between underfit and goodfit in order to avoid LSTM to learn many wrong characteristics in wrong labelled dataset even in the goodfit condition. It aims to make the model have a good generalization on the future patent dataset. The maximum sequence length is among {20, 30, 40, 50} because of the patent sentences which are usually longer than normal sentences. Activation function is set as sigmoid. Dropout rates of LSTM encoder layer and dense layer are separately set as 0.17 and 0.25 to prevent neural networks from overfitting. The optimal parameters are highlighted with bold faces.

To show how MaLSTM can pick up similar sentences, we compare the performance of different methods on the labelled sample dataset with different similarity thresholds.

The results are shown in Table 5.9. MaLSTM generally outperforms BOW, TF-IDF, and Word2vec on labelled sample datasets. Specifically, when we set the similarity threshold as 0.8, MaLSTM achieves the promising experimental results similar to the original research results (Mueller and Thyagarajan, 2016). Besides, we also noticed that with the increase of similarity threshold, the number of predicted correct sentences out of similar sentences might be sharply decreased. At the end, our goal is to be able to find inventive solutions after matching contradictions between each other. Therefore, we still need to ensure there are enough (a few) numbers of predicted similar pairs of contradictions that will be flowed to the next stage which is extracting inventive solutions. Based on the previous work (Ni, Samet, and Cavallucci, 2019), the final threshold around 0.85 is the right trade-off between the precision and the obtained number of similar contradiction. So, in this work, we

ultimately define the similarity threshold as 0.8. It avoids two-thirds of extra computation consumption. MaLSTM finally retrieves 327 pairs of similar contradictions from these 2.8 million pairs of contradictions from 8 domains in Step 4. After experts' evaluation in step 5, the precision of MaLSTM on the U.S. patent dataset is 78.59%. The inventive solutions are subsequently extracted from these corresponding solutions of similar contradictions in different domains.

5.5.3 Use case example with PatRIS

In this section, in order to evaluate the quality of the similarity and the ranking, we choose to evaluate on a real use case from the battery domain. Our starting input contradiction is *Having batteries that charge faster reduces the number of replacement batteries needed. But a faster charge increases the temperature and therefore the risk of fire.* This is typically a contradiction that highlights the battery thermal runaway. This sentence is fed to SAM-IDM and then to PatRIS for finding the best similar contradiction in the same domain. Several matching sentences are shown as follows:

TABLE 5.10: Ranking of sentence given for the formulated contradiction

PaTRIS ranking	Sentence
1	Voltage measurement for multiple cell battery pack
2	Heat exchanging system that exchanges heat between refrigerant and a temperature regulated portion
3	Secondary battery charging system and method, and battery pack
4	Germanium-containing active material for anodes for lithium-ion devices

We look deeper into the patent containing the first ranked sentence. The matched sentence extracted from patent US20160097815A1 is *Overcharging may cause severe damages to battery cells and may even become safety concerns. Although the invention is trying to supply the surplus energy to the next cell in line, this system inevitably causes energy waste for charging which is not friendly to the environment.* The invention relates generally to battery charging and protection, and more particularly to voltage measurement for a plurality of rechargeable battery cells. Overcharging lithium-ion or lithium polymer batteries, for example, may cause thermal runaway, and the high temperature developed may lead to cell rupture. Fire hazards have been reported during

charging as extreme cases. Therefore, much attention has been paid to battery over charging and solutions have been designed to avoid battery damage and safety issues. The typical charging system developed applies one charger to charge all cells connected in series in a battery pack. Since all cells are not manufactured in the same way, some cells may charge faster than others. As such, charging a battery pack with a single charger charging a plurality of cells can lead to the sonic overcharging.

5.6 Conclusion

In this chapter, we investigate the problem of helping R&D engineers to mine inventive solutions in the Industry 4.0 domain. Therefore, we develop a threefold contribution: *(i)* mine TRIZ contradiction from patents using two summerization approaches called SummerTRIZ and PaGAN; *(ii)* match contradiction with other user, a similarity-based approach and Siamese LSTM; *(iii)* rank the found output using a multi-criteria decision analysis algorithm. In literature, the TRIZ domain remains poor as far as the data mining-based approach is concerned. The available contributions offer the baseline and a foundation for the next improvement and perspectives discussed in Chapter 7.

Chapter 6

Battery Prognostic Using Neural Networks Models

Contents

7.1	Pattern mining in predictive maintenance	138
7.2	Mining inventive design using TRIZ theory	139
7.3	Battery prognostic using neural networks models	139
7.4	Perspectives	140
7.4.1	Crossing pattern mining and explainability for smart battery management	140
7.4.2	Crossing TRIZ patent mining and knowledge reasoning	142
7.4.3	Management Strategies, Industrial and Financial Prospects	144

6.1 Introduction and related works

In the development of Industry 4.0, energy management and energy efficiency overall has taken center stage. As a matter of fact, given sustainability regulations, specific power needs, and increasing importance of energy to drive the smart factory (i.e stationary application) and electrical vehicles (i.e electrical mobility), there is more attention than ever to energy efficiency in industrial markets.

It is not exactly a secret that factories and many other industrial facilities are not just among the main “consumers” of power; power is equally critical in ensuring continuity and quality of the end-to-end manufacturing process. There is a reason why in Industry 4.0 so much attention is paid to topics such as smart energy consumption in logistic mobility and smart battery management for electric mobility manufacturers.

6.1.1 Literature review of battery ageing analysis

Vehicle manufacturers in the world are undergoing unprecedented technological change (Mesbahi et al., 2017; Li et al., 2019). In the logistic transport field such as electric cars and heavy Automated Guided Vehicles (AGVs), the voices of progress are linked, among other things, to the hybridization and electrification. In both configurations, the effective operation of the systems is mainly linked to the availability of the on-board electrical energy and therefore of the battery. The energy storage systems used in the latest generation of Hybrid Electric Vehicles (HEVs) and pure Electric Vehicles (EVs) are mainly based on Lithium-ion (Li-ion) technology. This is because it has a high specific energy (around 250 Wh/kg) that this technology has established itself on the market of electric traction, especially in the automotive sector (Mesbahi et al., 2017). The challenge for any car manufacturer, wishing to develop a clean vehicle, is therefore not only to optimize its electric power-trains, both in terms of mass, cost and range, but also to bring the battery into line with the life of the vehicle (Mesbahi et al., 2017; Ren et al., 2018). Battery lifetime is therefore a key issue for the development of HEVs and EVs under acceptable cost conditions. In this context, failure as well as unsuitable usage of the battery could lead to serious inconvenience, performance deterioration, accelerated ageing and costly maintenance (Li et al., 2019). To the best of our knowledge, no major work has been established to understand battery

ageing sequences regardless of the battery technology. A better understanding of ageing patterns through battery charging and discharging sequences could enhance the prediction of the Remaining Useful Lifetime (RUL).

In automotive or logistic applications, a battery is considered out of use after reaching 80% of its initial storage capacity. This is referred to as the *End of Life* (EoL) criterion. The *State of Health* (SOH) of a battery is defined as the ratio between the storage capacity of the battery at any time ($Q(t)$) and its initial capacity (Q_{nom}).

$$SOH(t) = \frac{Q(t)}{Q_{nom}} \quad \text{and} \quad SOH(t)\% = \frac{Q(t)}{Q_{nom}} * 100 \quad (6.1)$$

Considering this definition, the *Remaining Useful Life* (RUL) is the operating time left to a battery (often expressed as a number of cycles) before its SOH reaches 80%.

Accurate prediction of RUL of Li-ion battery plays an increasingly crucial role in battery state estimation and health management (Wang et al., 2017). The typical method of Li-ion battery RUL prediction is usually divided into two categories: multiphysical model-based (Virkar, 2011) and data driven approach (Song et al., 2017; Hong et al., 2014; Jorge et al., 2020). The multiphysical model-based approaches are typically difficult to develop especially for battery degradation modeling due to the highly complex chemical reactions inside Li-ion battery. The data-driven method has recently drawn significant attention in Li-ion battery RUL prediction research area (Lipu et al., 2018). The advances in Artificial Intelligence (AI) and Deep Learning introduce new data-driven approaches to this problem. Deep Neural Network (DNN) is especially suitable for high complex non-linear fitting by training multi-layer artificial neural networks, and can achieve better accuracy for complex prediction problems such as multi-battery RUL estimation. Several prediction model-based Recurrent Neural Networks (RNN) (Li et al., 2019; Qu et al., 2019a) or artificial neural networks are introduced. Extensive review of existing battery ageing datasets and data driven approaches are in (Dos Reis et al., 2021; Hasib et al., 2021). All approaches have two major shortcomings: (i) they do not fully consider all the available features to make accurate predictions; (ii) they lack explainability when they make predictions.

6.1.2 Battery ageing datasets

The aim of the Sandia National Laboratories dataset (Preger et al., 2020) is to study the effect of Depth of Discharge (DoD), i.e load current and temperature on battery degradation. Eighty-six cells of three different chemistries (LFP, NMC and NCA) were considered in this study. Three different DoD were used, 0%-100%, 20%-80% and 40%-60% at three different temperatures, 15°C, 25°C and 35°C. The discharge rates used are 0.5C, 1C, 2C and 3C. XC refers to the battery C Rating which is the measurement of current in which a battery is charged and discharged. The capacity of a battery is generally rated and labelled at the 1C (1C current). All tests are conducted beyond 80% SOH and batteries were charged at 0.5C. The NCA cells were not discharged at 3C since it would be destructive.

In this study, only the data measured from the NMC cells tested at a 0-100% depth of discharge is used. Further research on the impact of the depth of discharge on capacity degradation and model performance should be done, as the results show clear evidence that the depth of discharge has major repercussions on a battery's capacity.

The NASA PCoE dataset (Saha and Goebel, 2007) is the most widely used and contains 6 groups of cells for a total of 34. The cycling experiment, which is the continuous process of charging and discharging the battery cells until end of life, was segmented into three parts : charging, discharging, and impedance spectroscopy. Impedance measurements were carried out through an Electrochemical Impedance Spectroscopy (EIS) frequency sweep from 0.1Hz to 5kHz. The charging protocol is the same for all tests, a CC-CV charging protocol with 1.5A and a 4.2V threshold. Various different discharging currents were used.

The MIT Cycle Life dataset contains data from 124 LFP cells cycled using various fast-charging protocols consisting of one-step or two-step constant current charging, switching to 1C CC-CV charging at 80% State of Charge (SOC). The cells were divided into three batches, and tested under a constant temperature of 30°C. Internal resistance is measured during charging at 80% SOC, using an average of ten 30ms or 33ms pulses of $\pm 3.6C$.

The datasets contain data about cycle indexes, voltage, current, temperature, charge and discharge capacity, charge and discharge energy, and battery impedance.

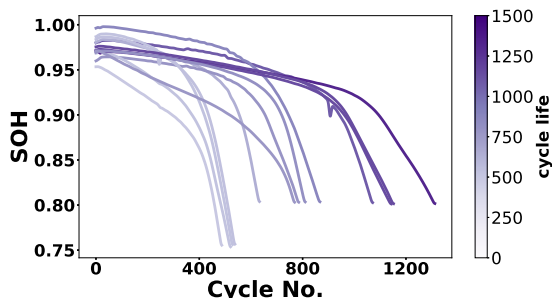


FIGURE 6.1:
SOH

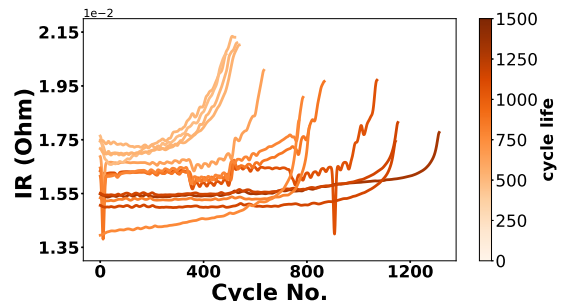


FIGURE 6.2:
IR

FIGURE 6.3: Historical Features

6.1.3 Motivation and challenges

There are two different data types in the ageing datasets of Li-Ion batteries: *historical* data and *local* time series. The historical data refers to the evolution of features according to the cycle number. In Figure 6.1, the SOH of several batteries from the MIT dataset is plotted as a function of the number of cycles. The longer the lifetime of the battery, the darker the curve. In Figure 6.2, the Internal Resistance (IR) of the same batteries is plotted as a function of the number of cycles. For both SOH and IR, one value is stored per each cycle. Therefore, SOH and IR are considered as Historical Features (HF).

The time series data concern mainly the current, the voltage, and the temperature that vary according to the use profile of the battery and that are represented as a function of time inside a given cycle. The time series data is referred to as local since it results from the usage of the battery by the user. Figure 6.4 shows time series data of charge current for several given cycles of the same battery from the MIT dataset.

Historical data and local time series do not vary on the same scale but both carry information on battery ageing. According to the literature, NAR

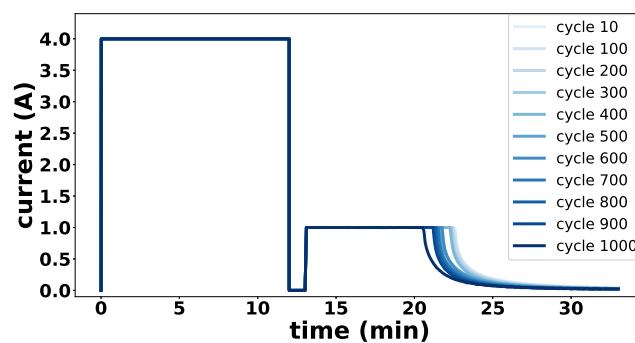


FIGURE 6.4: Charge current

models (Non Linear Auto-Regressive), are very efficient and lead to very accurate predictions of SOH. NAR Neural Networks make use of one single vector that is given as input to the network and that is the target output of this network. That means future values of one unique time series are determined by looking only at its past values, as follows:

$$y(t + 1) = f\{y(t), y(t - 1), \dots, y(t - d)\} \quad (6.2)$$

where d refers to the size of the input time window while t stands for the time stamp. However, building a Prognostic and Health Management (PHM) strategy for Li-Ion batteries should consider all available information about the use of the battery. Studying previous value of SOH provides accurate predictions of SOH for future cycles only if the battery is used in the same manner continuously. If the use conditions vary, the prediction of SOH will not be impacted consequently when using NAR models.

In this chapter, we focus on the problem of battery remaining useful life prediction using usage features. We also investigate the reasons behind battery SOH degradation using eXplainable Artificial Intelligence (XAI).

6.2 AE-LSTM based model for SOH prediction

Two main parameters are predicted, the SOH and the RUL, which both represent a battery's health. We chose to focus on SOH predictions. The majority of papers focus on using historical capacity values as inputs. In most approaches, the conclusion is made: the larger the input window, the better the prediction. We propose a predictive approach based on a sliding window of limited size that performs well on long term predictions of SOH values. The proposed architecture is based on auto-encoders as a mean of feature extraction and dimensional reduction. Local time series data features such as voltage, current, and temperature vary with the usage conditions of the battery and its state of degradation. As a battery SOH decreases, its internal resistance rises and leads to more power loss. Local time series are real time indicators of how the battery is being used by a driver, and are believed to be key factors to observe in our PHM strategy. The following six time series are used to make a prediction: charge voltage (V_c), discharge voltage (V_d), charge current (I_c), discharge current (I_d), charge temperature (T_c) and discharge temperature (T_d).

Figure 6.5 presents the architecture of our AE-LSTM approach. Windows of raw time series are given as inputs to make a SOH prediction. Each of these

time series are then distributed into two branches. One of them is made of the encoding layers of an auto-encoder trained to reconstruct this particular time series. This encoder outputs a code, corresponding to a vector of data representative of the input. The second branch is composed of LSTM layers. The encoder's weights are frozen during the training of the final model. This

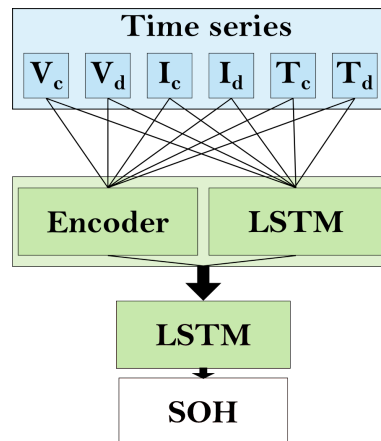


FIGURE 6.5: Architecture summary

allows the first branch to act as a static feature extraction method while the second branch gives more context as it learns from time series of varying lengths.

The outputs of the two sub-systems are then concatenated and fed through LSTM layers to make the SOH prediction. A detailed description of this architecture is available in Sections 6.2.2 and 6.2.3.

6.2.1 Data pre-processing

Data pre-processing plays a key role in data-driven strategies. Data that is not properly processed and shaped can not possibly lead to satisfying results. This is why a step of denoising, outlier excluding and normalization is needed. During data acquisition, several sensors and data alterations may add unwanted noise (electromagnetic noise, vibrations, ...).

The model was built using the Sandia National Laboratories ¹ (Preger et al., 2020) dataset and tested on it as well as on the MIT Life Cycle dataset ² (Severson and Attia, 2019) and the NASA PCoE battery dataset ³ (Saha and Goebel, 2007).

¹ Accessed at <https://www.batteryarchive.org> in March 2021

² Accessed at <https://data.matr.io/1/projects/5c48dd2bc625d700019f3204> in March 2021

³ Accessed at <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/> in March 2021

A first look at the SNL data showed several peaks in most capacity evolution curves, which had to be dealt with using a simple function which detects these peaks and computes their imputed new value instead of using the neighboring ones. These peaks are artifacts indicated on the dataset's download page, and are due to the transitions between normal cycling and capacity checks. After excluding these outliers, the data could be normalized before being used.

The NASA PCoE battery dataset, while being the most used dataset in literature, needed several steps of pre-processing in order to be integrated in our model. The data is gathered during three distinct steps : charge, discharge and electrochemical impedance spectroscopy. We do not use the impedance data; therefore it is excluded. What remains of the data structure makes a clear distinction between charge and discharge data, which was grouped in order to get both pieces of information for each cycle. A few irregularities such as two subsequent charging processes should be excluded. Finally, due to several gaps in the battery capacity measurements, several batteries are excluded from this work. We use batteries #5-7, #18, #32-36, #45-48 and #53-56.

The MIT Life Cycle dataset (Severson and Attia, 2019) is the largest and the most recent battery ageing dataset. This dataset does not need much of pre-processing as the data do not contain any noise. The data is divided into three batches which correspond to three series of tests. Several batteries from each of the batches were selected to conduct performance tests.

The process of normalization is an important step to avoid imbalance between different input data samples. Here, Min-Max normalization is applied to the data :

$$MinMax = \frac{data - \min(data)}{\max(data) - \min(data)} \quad (6.3)$$

The output data ranges from 0 to 1. Note that this normalization process is applied to the input of the models, as the target data is the SOH which is obtained by dividing the capacity values by the nominal capacity of the battery. The batteries are split as follows : 60% for training, 20% for testing and 20% for validation.

6.2.2 Encoding branch

The auto-encoders are models used to reduce their input to a code which is representative enough to be reconstructed. Inputs are windows of padded time series since each and every charge and discharge cycle is of different

length. The sequence-to-sequence auto-encoders are made of LSTM layers of the following sizes : [256,16,RL,256,IS] where RL is a RepeatVector layer with a repetition factor equal to the size of the input window and IS is the size of the input time series. The encoding layers of these models are then used in the main AE-LSTM model.

6.2.3 LSTM branch

In parallel to the encoding layers, the input time series are distributed towards LSTM layers. These two LSTM layers are of size 256 and 32. As the encoding layers extracted from the auto-encoders have their weight frozen, this structure allows the final model to learn while training the final model. After concatenating the outputs of the different branches, the SOH prediction is made using LSTM layers of sizes [512,256,256,32] and a final dense layer with 1 unit and a linear activation function giving the final output.

6.2.4 Smoothing

Certain predictions of the AE-LSTM model present some irregularities. In an attempt to improve these predictions, a post-processing method is investigated. One thing to consider when post-processing SOH predictions is that each value can only be smoothed using the previous ones. Our smoothing method uses a window of predicted capacity values to adjust the next one. These values are assigned a weight in order to give more importance to the most recent predictions. The weights are computed using a simple polynomial function applied to a vector of progressive integers. For example, if using a window of size 5, the vector is initialized as [1,2,3,4,5]. A function $f(x) = x^n$ is then applied to this vector to adjust the weights, where n is a power factor that can be adjusted. With $n = 3$, we get a weight vector equal to [1,8,29,64,125] which we can apply to the window of capacity values preceding the considered prediction.

The smoothing method shows promising results when applied to the predictions but is not consistent enough. The error values presented in the following sections are the ones computed using the predictions without the post-processing stage.

6.3 Window exogenous LSTM for SOH prediction

In this part, we introduce a deep learning model for SOH prediction, based on the extraction of features from temporal curves of current, voltage and temperature, which are used as input to a window - based exogenous LSTM. The architecture is referred to as SOH-window-XLSTM. The global framework of our model is described in Figure 6.6, and the following subsections detail the principles of feature extraction, feature selection, and window LSTM for SOH prediction.

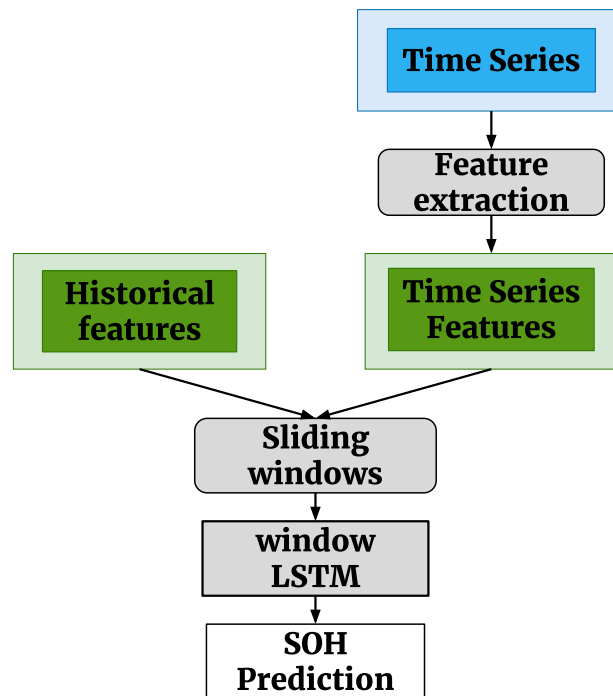


FIGURE 6.6: SOH-window-XLSTM architecture

6.3.1 Feature extraction

In this approach, features are computed and extracted from six different temporal curves: charge current, voltage, temperature and discharge current, voltage and temperature, namely I_C , V_C , T_C , I_D , V_D and T_D . For each of these curves, and at each cycle of the life of each cell, several features are computed in the temporal, statistical and spectral domain. The list of all computed features for each of the aforementioned curves is separated into domains (Barandas et al., 2020). In all equations, s represents the time series signal vector, t is the corresponding time vector and N is the length of s .

6.3.1.1 Temporal domain

- **Total Energy**: $\frac{\sum_{i=1}^N s^2}{t_N - t_0}$
- **Area Under the Curve (AUC)**: $\sum_{i=1}^N (t_i - t_{i-1}) * \frac{s_i + s_{i-1}}{2}$

6.3.1.2 Statistical domain

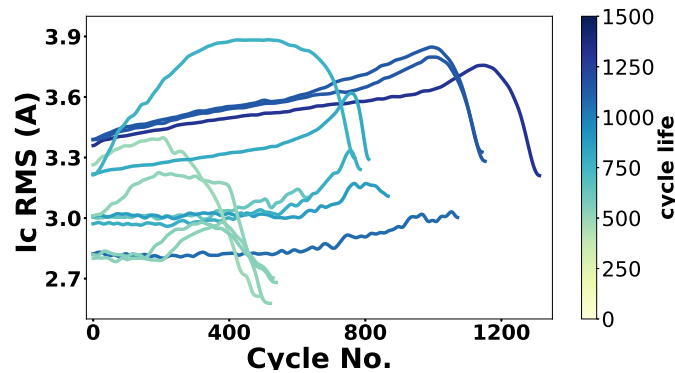
- **Mean** : $\frac{1}{N} \sum_{i=1}^{N-1} s_i$
- **Root Mean Square (RMS)**: $\sqrt{\frac{1}{N} \sum_{i=1}^N s_i^2}$
- **Maximum** : the largest value in s
- **Minimum** : the smallest value in s

6.3.1.3 Spectral domain

The Fast Fourier Transform of the signal is computed ($freq, fmag = fft(t, s)$), from which several features are extracted:

- **Fundamental frequency** : the lowest frequency of the Fourier transform
- F_{max} : the maximum frequency of the Fourier transform
- **Power Bandwidth** : the width of the frequency interval in which 95% of the power of the signal is located

By reducing I , V and T° time series to a combination of several scalar features, local time series data can then be represented at the same scale as HF, in the form of Time Series Features (TSF). For example, the global evolution of the RMS value of charge current is shown in Figure 6.7 over the whole cycle life of different cells, just as historical features can be represented. Every curve corresponds to a different battery, and the darker the curve, the longer the cycle life.

FIGURE 6.7: I_C RMS

6.3.2 Feature selection

After computing features from the original time series, each cycle is represented by a vector of features that includes TSF and HF. This vector has a very high dimensionality, because the same number of features is computed from each of the six time series described earlier. A common practice is to reduce the dimensionality of the learning algorithm input vector by selecting a subset of features that best represents the predicting problem (Guyon and Elisseeff, 2003b).

In our approach, we use wrappers for selecting features (Kohavi and H.John, 2013). Wrappers have the advantage of evaluating the relevance of features according to the performance of a predictive model. The wrapper technique for feature selection uses the training process on a given machine learning model to select the best combination of features. The aim is to obtain the best possible performances with a given algorithm by testing iteratively different subsets, numbers and combinations of features. The strategy is to start with an initial set of features and to add or remove several features after each training process in order to study the impact on the performances. This process is called the Sequential Feature Selection. After the feature selection process, 10 features are kept, mainly coming from charge curves.

6.3.3 Sliding window approach for SOH prediction

The working principle of our LSTM neural network is to observe a sample of past values of any given signals to predict future values of any given signals. The proposed architecture, based on a window exogenous LSTM, makes use of past values of TSF to forecast future values of SOH. Although LSTMs are built to learn long term dependencies in time series, small input windows

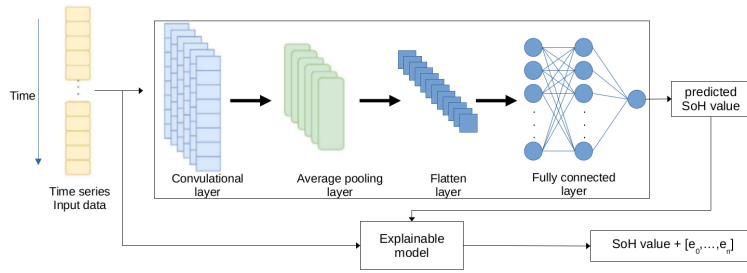


FIGURE 6.8: SoHAP architecture. The vector $[e_0, \dots, e_n]$ represents the contributions of each input value on the SoH.

are used in this model. Rather than giving as input the whole operating data of current, voltage and temperature coming from a battery since its beginning of life, a smaller window of several consecutive cycles is considered. This allows more flexibility in the model. There is no need to store large data sequences, and data preprocessing is made simpler because the size of the window is fixed. The window dataset that is built consists in several overlapping windows of TSF, and the output corresponds to a future value of SOH. Predictions can be made on a very short scale, or for longer time horizons. The training dataset for SOH prediction in this approach is built in a way that several forecasting horizons are made possible. Future values of SOH are predicted from 25 up to 400 cycles ahead. Another advantage of this model is that an online prediction of SOH is made. Most approaches (Qu et al., 2019b; Liu et al., 2012) implement iterative predictions, which consists in updating the input signal given to the predictive model with the last prediction. Our SOH-window-XLSTM is trained with “offline data”, and prediction can be made “online”, on unseen data corresponding to a window of 25 consecutive cycles at any moment of the cycle life of a battery cell.

6.4 SoHAP: SoH explainable prediction based on SHAP

The SoHAP architecture is shown in Figure 6.8 and it is based on a CNN for SoH predictions and the use of SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) for identifying the impact of the inputs regarding the SoH value predicted by the CNN.

Windows of raw time series are used as inputs to the CNN to make SoH predictions. Voltage, current and temperature time series vary with the usage conditions of the battery and its degradation. As a battery degrades, its internal resistance rises and leads to more power loss which can be noticed

in the evolution of its SoH and in usage time series. Usage time series are real time indicators of how the battery is being used by a driver and they are key factors to observe in our prognostics and health management strategy. The following time series are used to make a prediction: charge and discharge voltage, charge and discharge current, charge and discharge temperature.

First, our proposed convolutional neural network is introduced. Then the SHAP method is described; it is used to obtain certain patterns in the data that are associated with events or situations that degrade the SoH of the batteries in an abrupt manner.

6.4.1 Convolutional Neural Network used in SoHAP

In literature, when it comes to prediction SOC and SOH, baseline approaches are based on a CNN architecture (Hannan et al., 2021). Comparatively to architectures of Section 6.2 and Section 6.3 they provided a small error rate (Hannan et al., 2021) and fewer parameters.

Although CNN is the most preferred choice for coupling factor analysis such as classification and segmentation in image applications, in recent years CNN is also employed for SOH and RUL estimation (Hong et al., 2020; Zhou et al., 2020). This is mainly due to the ability of the CNN model to capture local capacity regeneration, thus improving the overall prediction accuracy of the model. In addition, pooling and dropout layers help to improve the training speed of the model and to avoid over-fitting in a deep network.

In our approach we build two CNN models. The first one considers as input only the time series of temperature, current and voltage data of a single cycle, and predicts the SOH at the end of the cycle. The second one considers as input the time series temperature, current and voltage data of three consecutive cycles, and predicts the SOH at the end of the third cycle. Both of them have the same architecture.

The choices of some parameters for the two convolutional network models developed are described below. Each choice is the result of a series of tests in which only one specific parameter has been modified in order to try to obtain the best possible results. First, regarding the type of pooling used, it was found that the average pooling gave much better results than the maximum pooling. Furthermore, increasing the number of successive convolution layers does not seem to have a noticeable effect on the performance of our model; thus a single convolution layer is placed before the MultiLayer Perceptron (MLP). One of the configurations that gave the best results in terms of size and number of convolution filters was to have 12 filters, each of

size 8. As for the MLP itself, it was decided to have three layers of neurons. The output layer is necessarily composed of a single neuron, since the objective is to perform a regression and not a classification. As for the number of neurons in the first two layers, several tests were performed with different configurations for the two layers (16, 32, 64 or 128 neurons) and we observed a remarkable improvement of the final mean absolute error obtained up to 64 neurons per layer, while going to 128 allowed a rather small improvement but considerably increased the time needed for training.

The final model structure for SOH prediction consists of two layers with 64 neurons each and a fully connected output layer of one unit, since a single value is presented. More details on training and model construction is given in the section 6.5.

6.4.2 Post-hoc explainability

As discussed by Lipton (Lipton, 2016), approaches to creating interpretability can be divided in two main groups: model transparency and post-hoc explainability. While the former tries to explain the model structure, the post-hoc explainability presents a distinct approach to extracting information from learned models. While post-hoc interpretations often do not elucidate precisely how a model works, they may nonetheless provide useful information for experts and end-users of machine learning. Some common approaches to post-hoc interpretations include visualizations of learned representations or models, and explanations by example. One advantage of this concept of interpretability is that we can interpret black box models after-the-fact, without sacrificing predictive performance.

SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017) provides a game-theoretical approach to explain machine learning model outputs. It is an additive feature attribution method to explain the output of any machine learning model. SHAP assigns each feature an importance value for a particular prediction. It combines optimal credit allocation and local explanations using classical Shapley values. The additive in the name emphasizes an important property. The sum of the SHAP values results in the prediction of the model. For a simple linear regression problem, the predictions can be written as:

$$\hat{y}_i = b_0 + b_1x_{1i} + \dots + b_dx_{di} \quad (6.4)$$

where \hat{y}_i is the i -th predicted response, x_{1i}, \dots, x_{di} are the features of current observation, and b_0, \dots, b_d are the estimated regression coefficients. In this case, the \hat{y}_i values correspond to the predicted $S\hat{o}H_i$ and the x_{1i}, \dots, x_{di} to the time series values of voltage, current and temperature. SHAP is a generalization of this concept to more complex neural network models. We define the following:

- F is the entire set of features, and S denotes a subset.
- $S \cup \{i\}$ is the union of the subset S and feature $i \in F$.

Then, the SHAP value is defined to measure the contribution of the i -th feature as a weighted average of all possible differences, as shown in Equation 6.5.

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)] \quad (6.5)$$

The terms of the formula can be interpreted in the following way:

- $f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S)$: captures the marginal contributions of member i .
- $|S|!$: ways in which the set S could have been formed before adding i .
- $(|F| - |S| - 1)!$: Different ways the remaining players could be added.
- $|F|!$: Number of combinations that can be formed with the coalition.

SHAP values are proved to satisfy good properties such as fairness and consistency on attributing importance scores to each feature. But the calculation of SHAP values is computationally expensive. In our case, we use Deep SHAP, which is a model-specific method to improve computational performance through a connection between Shapley values and DeepLIFT (Shrikumar, Greenside, and Kundaje, 2017).

As shown in Figure 6.8, from a given input and its predicted SoH, this method allows to obtain a numerical vector with the Shapley values for each value of the input representing the influence, which can be negative, positive or negligible on the predicted value of the SoH.

6.5 Experiments

In order to compare the performances of our models among themselves and with other approaches described in the literature, several scoring measures were used. In the vast majority of works, the evaluation of models is based on the Root Mean Square Error (RMSE), Root Mean Square Percentage Error (RMSPE) and Mean Absolute Error (MAE). MAE and RMSE are very common error metrics in regression problems. Individual errors are not taken into account in the same way in RMSE and MAE. Large errors and outliers have a greater influence in RMSE due to the squaring information, whereas MAE is a linear score, which means that all the individual errors are weighted equally in the average. RMSPE has the advantage of being scale independent, which is practical to compare forecast performances across different datasets (Hyndman and Koehler, 2006; Armstrong and Collopy, 1992).

We also add the Normalised Mean Square Error (NMSE) in order to compare the performances of our models with future work, and the Standard Deviation of the MAE (σ_{MAE}) in order to evaluate the reliability of the models. These quality measures are expressed as follows:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_{pred,i} - y_i| \quad (6.6)$$

$$\sigma_{MAE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (a_i - MAE)^2} \quad (6.7)$$

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_{pred,i} - y_i)^2} \quad (6.8)$$

$$RMSPE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{(y_{pred,i} - y_i)}{y_i} \right)^2} \quad (6.9)$$

$$NMSE = \frac{\sum_{i=1}^N (y_{pred,i} - y_i)^2}{N * V} \quad (6.10)$$

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|y_{pred,i} - y_i|}{y_i} * 100 \quad (6.11)$$

In all these formulas, $y_{pred,i}$ is the SOH predicted by the model; y_i is the real SOH, and N is the number of samples on which error is calculated; V

is the variance of y . For example, the use of the mean of y as the predicted values would give an NMSE of 1.

We investigate the predicting performances of our different models, for SOH forecasting. The size of the input window stays the same for all tests and was fixed at 25 consecutive cycles. For both models, all error metrics will be shown. For SOH forecasting, the output value is scaled with the maximum-absolute method, so it varies between 0 and 1 (all values are positive). MAE and RMSE measurements will then refer to signals that have an amplitude of 1. Different models are trained according to the input data. As there are important differences between batteries that were tested in the MIT and in the NASA dataset, distinct models need to be trained. Therefore, the results regarding MIT batteries and NASA batteries will be described in two different sections.

6.5.1 SOH forecasting experiments

6.5.1.1 SOH forecasting on the MIT dataset

Table 6.1 shows the predicting performances of our SOH-window-XLSTM model, on MIT batteries, in the case of multi-step predictions. The size of the input window is of 25 cycles, and the SOH prediction can be made from 25 cycles ahead up to 400 cycles ahead. For all cases, two metrics are given: the average prediction error and the minimum prediction error. Different successive trainings are made on different train/validation/test splits, and the average error is computed as the mean of all trainings. The prediction error of the best performing model is also shown for each predicting horizon.

As expected, the best performances are obtained for short term predictions, 25 cycles ahead. As the forecasting horizon grows longer, the predicting performances degrade. The best of our model reaches a 1.14% RMSPE, for a 25 cycles ahead prediction. Nevertheless, with very long term predictions, up to 400 cycles ahead, the RMSPE stays as low as 3.14%. In Figure 6.11, the average and minimum MAE and RMSE of each model, corresponding to each predicting horizon, are plotted. The curves show that the prediction error increases almost linearly according to the number of cycles ahead. The structure of the model does not vary from one predicting horizon to the other, which means that the architecture is quite robust.

In Figures 6.9 and 6.10, a comparison is made between the real SOH curve and the predicted one, for 25 cycles ahead predictions and 300 cycles ahead

predictions. Those predicted curves correspond to predictions that were made with the best performing model, trained to predict 25 or 300 cycles ahead. The batteries chosen to compare real SOH and predicted SOH were picked in the test set of the corresponding model. More precision on the train, validation and test sets are given in the appendix of the article. In both cases, the predicted curves are very close to the real SOH curve.

Given the results presented in Table 6.1, our models stay on average between $1.05 \cdot 10^{-2}$ (25 cycles ahead) and $2.4 \cdot 10^{-2}$ (400 cycles ahead) for the MAE, and the standard deviation of the MAE stays below $2 \cdot 10^{-2}$ (for 400 cycles ahead). Considering that those measures refer to signals that have a $[0,1]$ range, we can conclude that our SOH-window-XLSTM models give accurate predictions, at both short term and long term, with a high reliability.

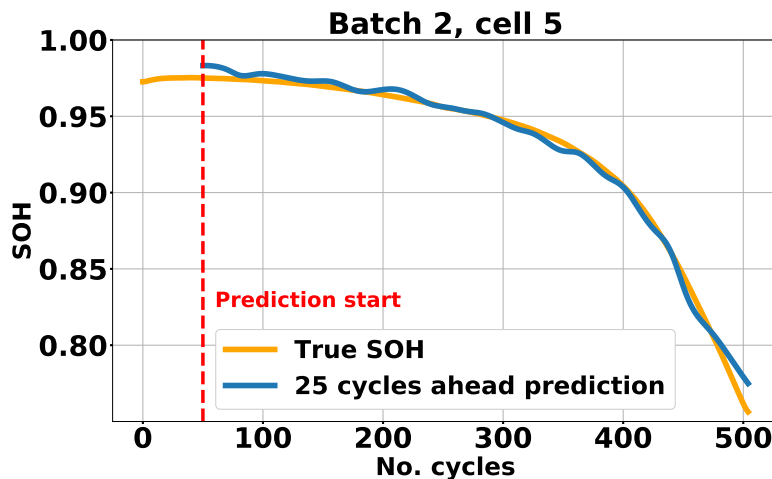


FIGURE 6.9: Real VS 25-cycles ahead predicted SOH for battery b2c5

6.5.1.2 SOH forecasting on the NASA PCoE dataset

Table 6.1 shows the predicting performances of our SOH-window-XLSTM model, on NASA batteries, in the case of multi-step ahead predictions. The size of the input window is of 25 cycles, and the SOH prediction is made only 12 cycles ahead. Contrary to MIT batteries, the ones that were tested by the NASA PCoE have a short cycle life (168 cycles maximum, compared to more than 2000 cycles for some batteries in the MIT dataset). In order to have a reasonable number of training samples, the choice was made to limit the predicting horizon to 12 cycles.

Even though there is not much available training data, the predicting results are still satisfactory and vary in the same range as the ones obtained

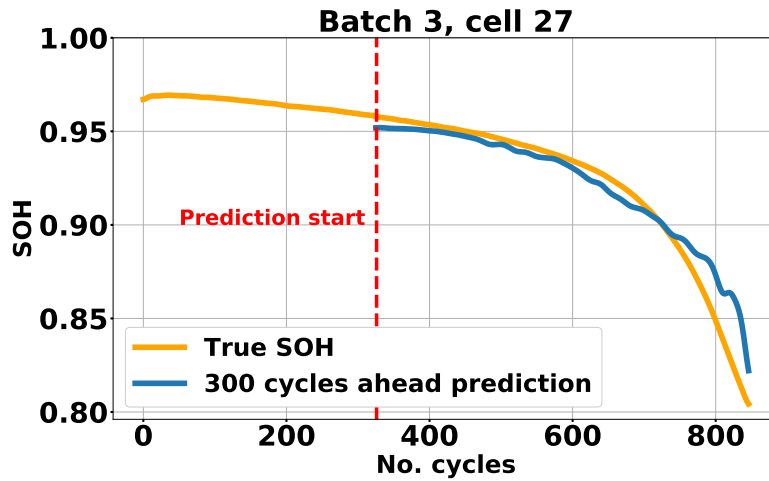


FIGURE 6.10: Real VS 300-cycles ahead predicted SOH for battery b3c27

TABLE 6.1: XLSTM performances for SOH prediction on MIT batteries

No of cycles ahead		MAE	STD	RMSE	RMSPE	NMSE
		$\times 10^{-2}$	$\times 10^{-2}$	$\times 10^{-2}$	%	$\times 10^{-1}$
25	Average	1.05	1.13	1.50	1.70	1.18
	Best	0.77	0.70	1.04	1.14	0.48
50	Average	1.20	1.50	1.90	2.16	1.80
	Best	1.07	1.20	1.60	1.80	1.24
100	Average	1.40	1.60	2.10	2.44	2.10
	Best	1.19	1.10	1.60	1.80	1.20
150	Average	1.30	1.90	2.30	2.67	2.50
	Best	0.90	1.04	1.40	1.60	0.80
200	Average	1.90	2.10	3.00	3.30	3.60
	Best	1.50	1.70	2.20	2.60	2.20
250	Average	1.60	1.90	2.5	2.82	2.70
	Best	1.10	1.50	1.90	2.14	1.40
300	Average	2.00	2.30	3.00	3.49	3.90
	Best	1.50	1.40	2.10	2.50	1.70
350	Average	2.10	2.40	3.20	3.70	4.30
	Best	1.60	1.90	2.50	2.90	2.70
400	Average	2.40	2.80	3.70	4.22	5.70
	Best	2.00	2.00	2.80	3.14	3.10

with MIT batteries. The three models corresponding to batteries B0005, B0006 and B0007 have comparable performances. For these three batteries, the best predicting models have a RMSPE of 1.9% , 3.4% and 2.05% respectively. However, the predicting performances of the model on battery B0018 is slightly lower, with a RMSPE of 5.2% . This might be due to the fact that

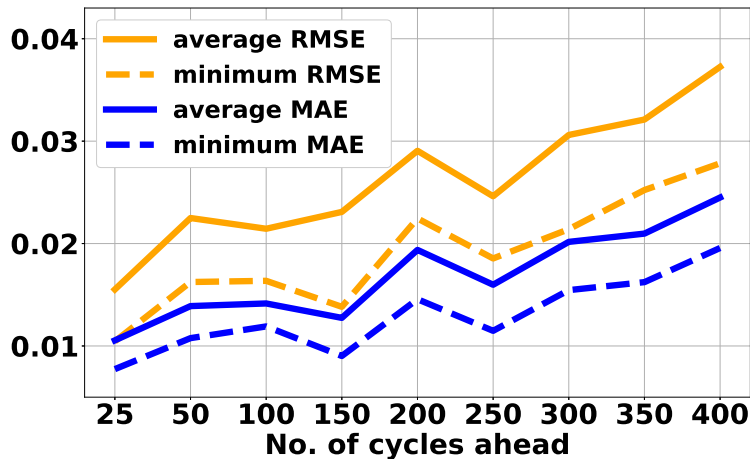


FIGURE 6.11: Evolution of the average and minimum MAE and RMSE according to the number of cycles ahead

battery B0018 has a lower cycle life than batteries B0005 to B0007. Indeed, batteries B0005, B0006, and B0007 are tested during 168 cycles; batteries B0054 and B0055 are tested during 90 cycles. Battery B0018 is the only one to be tested for 132 cycles.

Figures 6.12 and 6.13 show a comparison between the real SOH and the predicted SOH for battery B0005 and battery B0006. In both cases, the two curves are very close. Even though local regeneration phenomena are not taken into account, our model catches the global trend of SOH degradation. With more training data, local variations could be predicted with more accuracy.

Despite the predicting performances being quite good on those four batteries, the fact that the worst performance is obtained with the most "isolated" battery in the dataset shows that a low number of training data impacts negatively the performances of our predictive model. In all cases, the SOH-window-XLSTM trained on MIT data performs better. There are 20 times more exploitable batteries in the MIT dataset than in the NASA dataset, and they have on average a 5-time longer lifetime. Therefore, the MIT SOH-window-XLSTM model is trained with a lot more data, leading to better and more reliable results.

6.5.2 Comparison with other approaches

6.5.2.1 MIT prediction results

There are not many approaches in the literature dealing with the SOH

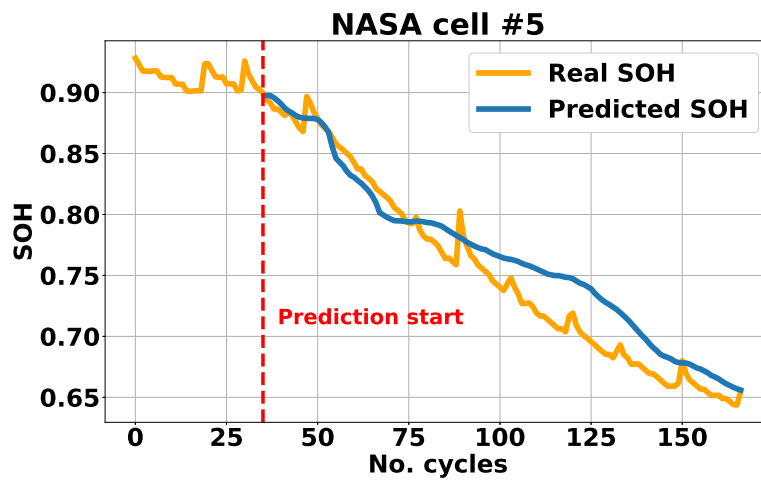


FIGURE 6.12: Real VS 12-steps ahead predicted SOH for NASA battery # 5

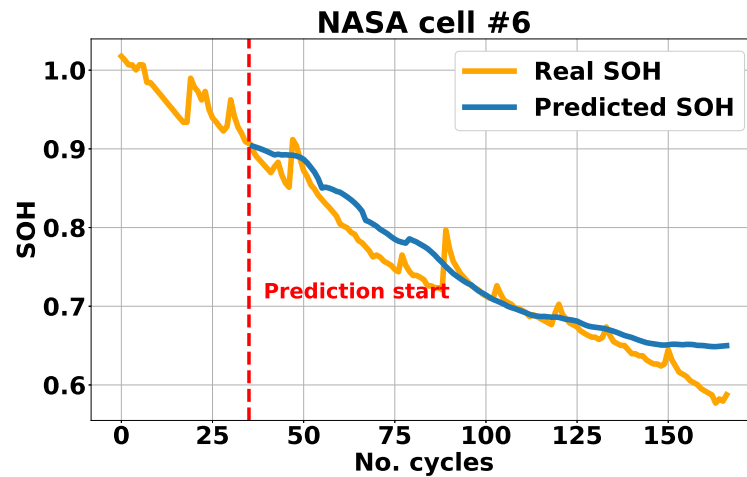


FIGURE 6.13: Real VS 12-steps ahead predicted SOH for NASA battery # 6

TABLE 6.2: XLSTM performances for SOH prediction on NASA batteries

Tested battery No.		MAE	STD	RMSE	RMSPE	NMSE
		$\times 10^{-2}$	$\times 10^{-2}$	$\times 10^{-2}$	%	$\times 10^{-1}$
B0005	Average	1.60	1.10	2.00	2.60	0.67
	Best	1.20	0.70	1.40	1.90	0.32
B0006	Average	2.20	1.60	2.70	4.00	1.00
	Best	1.90	1.30	2.30	3.40	0.70
B0007	Average	2.90	1.30	3.20	4.10	3.00
	Best	1.50	0.80	1.60	2.05	0.70
B0018	Average	4.50	2.50	5.20	7.00	9.00
	Best	3.10	2.40	3.90	5.20	5.30

prediction from the MIT dataset. We compare our predicting performances with those of Audin *et al.* (Audin *et al.*, 2021) that deal with the forecasting of SOH, multi cycles ahead. Their approach is also based on the use of temporal series of current, voltage and temperature as input features to a predictive model. Time series are encoded with an auto encoder coupled with an LSTM. SOH is forecast at different horizons, from a 25 cycles input window.

In Audin *et al.* (Audin *et al.*, 2021), the prediction performances of the AE-LSTM model is given for a 25 cycles input window, and the output is the value of SOH, 50 cycles ahead. We compare their predicting performances to our best performing SOH-window-XLSTM model, for a 50 cycles ahead prediction.

We can see from Table 6.3 that our SOH-window-XLSTM model outperforms the AE-LSTM model. For a 50-cycle ahead SOH prediction, our model shows a RMSE of $1.6 * 10^{-2}$ compared to $2.78 * 10^{-2}$ for the AE-LSTM model.

Our model is less complex as it only uses LSTM in the model itself. The data preprocessing requires low computational abilities as very simple features are extracted from time series.

TABLE 6.3: Comparison of SOH prediction performances, 50 cycles ahead, on MIT batteries

Approach	MAE	STD	RMSE	NMSE
	$*10^{-2}$	$*10^{-2}$	$*10^{-2}$	$*10^{-1}$
XLSTM	1.07	1.20	1.60	0.12
AE-LSTM (Audin <i>et al.</i> , 2021)	2.43	1.40	2.78	8.07

6.5.2.2 NASA prediction results

Most approaches concerning SOH prediction in the literature focus on the NASA datasets. Although there are very few batteries in this dataset, it was the first one published, and the numerous approaches in the literature offer possibilities of comparison. We compare our predictive performances with two articles published in 2019 by Qu *et al.* (Qu *et al.*, 2019b) and 2021 by Liu *et al.* (Liu *et al.*, 2021). Although these two approaches describe clearly their architecture and training process, it is not easy to compare our model with theirs. An exhaustive comparison between the different aspects of their methods and ours is made in Table 6.4. Considering that our architecture is based on the use of TSF only and not past SOH values, it is very hard to compare our SOH-window-XLSTM model to the PA-LSTM (Qu *et al.*, 2019b) or

the LSTM + GPR (Gaussian Process Regression) (Liu et al., 2012). Moreover, our approach aims at building a generic model from a maximum of available data. With MIT batteries, it is easier to test the model on several unseen batteries as there are so many. With the NASA dataset, however, we split the batteries in a way that the validation and test ensemble are fused together, and contain only one battery.

The major difference of our approach is that all available data from all batteries are used to train the model. Training data can be considered as “offline data”, used to learn degradation patterns. Once the model is trained, predictions can be made online from any input window of the test battery. This means that it is not necessary to isolate the first part of the data from a given battery and to use it in model learning to predict future SOH values. In Qu et al. (Qu et al., 2019b), at least 30% of the data is used for incremental learning, and it can go up to 70% of the SOH curve. That means only the last 50 cycles can be forecast by the model. In Liu et al. (Liu et al., 2021), 50% of the SOH data is also used for training. We use input windows of 25 cycles, which represents less than 20% of the lifetime of battery B0018 and less than 15% of the lifetime of batteries B0005, B0006, and B0007.

As shown in Table 6.2, our model has a RMSE of $1.4 * 10^{-2}$, $2.3 * 10^{-2}$ and $3.9 * 10^{-2}$ for batteries B0005, B0006, and B0007 respectively. Our method is outperformed by (Qu et al., 2019b) and (Liu et al., 2021), as the first one reaches an RMSE of $1.19 * 10^{-2}$, $1.97 * 10^{-2}$ and $2.08 * 10^{-2}$ for batteries B0005, B0006, and B0007, and the second one reaches an RMSE of $3.6 * 10^{-3}$ and $4.9 * 10^{-3}$ for batteries B0005 and B0006 (B0007 is not indicated).

However, those results should be analysed regarding the context. Our model is less complex and takes operating data such as current, voltage, and temperature as input. The very positive results obtained by NAR models depend on the nature of the input data. Experimental ageing tests of Li-Ion batteries lead to a steady degradation trend, but one that does not represent the real use of an electric vehicle. By using TSF in a model, the SOH prediction is slightly less accurate, but more flexible and adaptable to different use cases.

6.5.3 Explainability analysis

In this subsection, two interesting results on the explanation provided by SHAP are presented. The first one is about the trained CNN model considering as input only the operating data of one cycle. The second one is about the trained CNN model considering as input three consecutive cycles.

TABLE 6.4: Architecture comparison

	SOH-window-XLSTM	PA-LSTM (Qu et al., 2019b)	LSTM + GPR (Liu et al., 2021)
Used features	Time series (voltage, current, temperature)	SOH	SOH
Proposed architecture	-Feature extraction -Feature selection -LSTM	-EMD -Particle Swarm -Attention mechanisms -LSTM	-EMD -GPR -LSTM
Type of model	-NRX -Generic model for all batteries	-NAR -One model for each battery	-NAR -One model for each battery
Training data	All available batteries	At least 30% of SOH for each battery	50% of SOH for each battery
Type of prediction	-Multi cycles ahead -Online prediction	-One cycle ahead -Iterative prediction	-Multi cycles ahead -Iterative prediction

In the first case, battery b1c1 of the data set was considered as an example. Figure 6.14 shows the SOH curve of this battery. As can be seen, it is a battery whose lifespan follows the expected one. Figure 6.15 shows the impacts of the different input values to estimate the SOHs at different cycles. We can observe that at the beginning of the lifetime (green area in Figure 6.14) the charging and discharging protocols used do not have a negative impact on the SOH; however, these same protocols have a very negative impact when the battery approaches the end of its lifetime (red area in Figure 6.14), mainly the current peaks (darker points in the Figure 6.15). This is a phenomenon that is consistent with expert knowledge. Therefore, here we can see that the explanation provided by SHAP is valid and supports the SOH prediction.

The second case is about the CNN model considering three consecutive cycles as input. This model was mainly done to verify that the battery degradation patterns are repeated from cycle to cycle, i.e., if a pattern has a negative impact in one cycle, it is expected that the same pattern will also have a negative impact in the next cycle. This is linked to the fact that battery aging occurs slowly. Therefore, a large difference in SOH values between two consecutive cycles would not be normal, unless something extreme occurs.

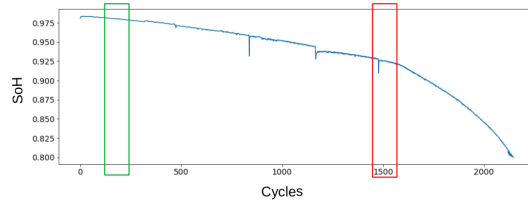


FIGURE 6.14: SOH curve for battery b1c1.

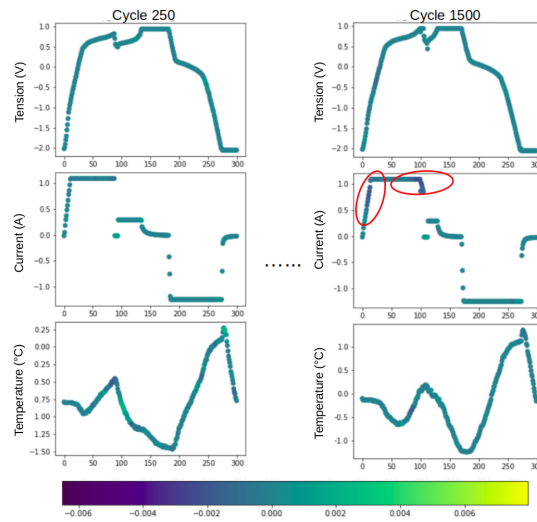


FIGURE 6.15: Shapley values of features over different cycles.

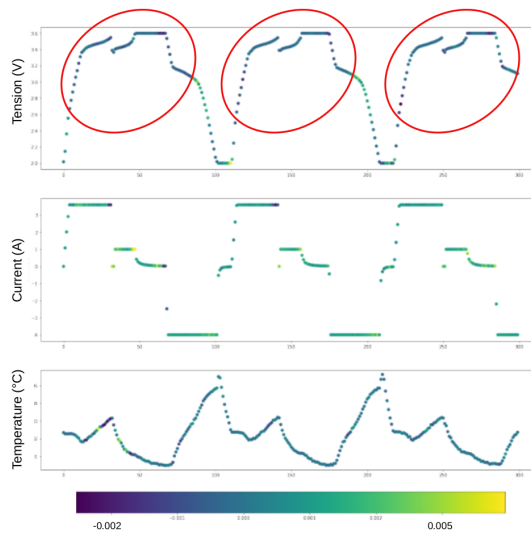


FIGURE 6.16: Shapley values of features over three consecutive cycles.

Figure 6.16 shows three consecutive cycles in the red area of Figure 6.14. From this figure we can observe that sharp increases in tension at that stage of life have a negative impact on the SOH, and we can also note that this pattern is repeated in the three cycles, which also follows the expert knowledge.

Therefore, what can be concluded from this situation is that sharp increases in voltage at this stage of the battery life should be avoided in order to extend its useful lifetime.

6.6 Conclusion

In this chapter, we tackle the problem of State-of-Health prediction of Lithium-ion cells. We propose two different models based on LSTM to predict the SOH-based usage data mainly: the current, the voltage, and the temperature. The models are applied on several public datasets. Additionally, we investigate the explainability of these black models in an attempt to (i) check the relation between entries and output to understand the ageing;(ii) validate, using expert knowledge, the decisions of the models.

Part IV

Conclusion & Perspectives

Chapter 7

Conclusions and Perspectives

The industrial sector has always been crucial to the economic development of countries. Since the end of the 18th century, industries have gone through massive changes that revolutionized the way how products are manufactured, and brought various benefits, mainly related with productivity increase. Nowadays, after three industrial revolutions, the combination of advanced technologies and internet is again transforming the industrial landscape and is being called the 4th Industrial Revolution or Industry 4.0.

The Industry 4.0 topic emerged as a long-term strategy of the German government; it was adopted as part of the High-Tech Strategy 2020 Action Plan in 2011, to ensure the competitiveness of its industry and its factories. Since then, the German government has institutionalized its commitment to industry in creating a platform led by Ministries of Economy and Business, Science and Trade representatives. In other words, Industry 4.0 represents a natural evolution of previous industrial systems, from the mechanization of labor in the 18th century to the automation of production in current days.

These smart factories are equipped with advanced sensors, embedded software and robotics that collect and analyze data and allow for better decision making. Even higher value is created when data from production operations is combined with operational data from ERP, supply chain, customer service and other enterprise systems to create whole new levels of visibility and insight from previously siloed information. These digital technologies lead to increased automation, predictive maintenance, self-optimization of process improvements and, above all, a new level of efficiencies and responsiveness to customers, which was not previously possible.

In this work, we tackle three main pillars of Industry 4.0 applications which are:

- factories' production safety through the investigation of future machine

failures under the major topic of predictive maintenance. We use machine sensor to understand and predict failure using data mining techniques;

- companies' innovation by exploiting one of the main supports of innovation which are the patents. We combine data mining techniques and TRIZ theory to discover and solve industrial problems;
- factories' and logistic transport energy reduction through better understanding of battery degradation and remaining useful life. Machine learning techniques are used to predict and understand battery ageing.

This manuscript presents some research directions to enhance knowledge extraction using data mining techniques for the purpose of Industry 4.0. Three main applications are singled out which are: predictive maintenance, inventive design using TRIZ theory and smart battery management.

7.1 Pattern mining in predictive maintenance

In the preliminary section, we introduce sequential pattern mining and we highlight some of its limitations in application such as Industry 4.0. These limitations are: (i) low descriptive pattern mined from sequential data; (ii) absence of any mining algorithm for mining pattern in the context of Industry 4.0; (iii) absence of any predictive algorithm using chronicles for the purpose of describing failure and predicting them.

Our proposal is to investigate the use of a special type of time-rich pattern called *chronicle* for the task of predictive maintenance. Therefore in Chapter 4, we propose three main contributions:

- two chronicle mining algorithms called *CPM* and *Clasp-CPM*. Both are a two-phase mining algorithm based on closed sequential pattern mining algorithms and time constraint computing. *Clasp-CPM* offers an enhanced performance due to its subsequence graph construction and multi-threading coding.
- three distinct algorithms for predicting future failures using failure chronicle previously mined. *FACE* algorithms that rely on picking the failure chronicle maximizing the confidence value. *FCP* is an algorithm that uses a fusing technique based on the evidence theory. Finally, we

present an approach used to prune chronicles when a few are introduced by an expert. Pruning techniques are based on metrics such as *subsumption, redundancy and conflict*.

7.2 Mining inventive design using TRIZ theory

In Chapter 3, we introduce the TRIZ theory that is widely used to analyze inventive documents such as patents, scientific papers, blogs and point out the provided inventive solution to solve an assumed contradiction. Although, this theory provides a solid formal background for solving inventive problems, automatizing it remains understudied. Indeed, to automatize the process, *contradictions* which are sentences that highlight the invention difficulty should be mined. Contradiction mining is the pillar of any TRIZ automatization process.

In this manuscript, we propose a three-step approach to automatize the process. Therefore in Chapter 5, three main contributions are proposed:

- two main models to mine contradiction from patent documents. First a baseline approach, called *SummerTRIZ* based on transformer and an ANN for text classification. An improvement of this approach is provided by considering the lack of labelled data. Therefore a Generative Adversarial Network based approach, called *PaGAN*, is detailed.
- a similarity-based approach, called *SIAM-LSTM* is introduced to compute similarity between contradictions. Experiments are conducted on laboratory expert labeled dataset of 1,600 patents.
- TRIZ distinguishes 5 levels of inventiveness. In our study, we rank inventive solutions provided by patents regarding several criteria such as citation, number of authors, similarity of the solution to the contradiction. The algorithms use a multiple-criteria decision analysis TOPSIS.

7.3 Battery prognostic using neural networks models

In Chapter 6, we shed light on the problem of energy management in battery storage systems. Indeed, we rely on the assumption that in order to optimize energy consumption in Industry 4.0 applications, we need to address battery lifetime prognosis. To do so, in our manuscript, we investigate the

Remaining Useful Life (RUL) and State of Health (SoH) prediction. We aim to predict future values on the basis of user usage data such as voltage, current, and temperature.

In this manuscript, we propose three main contributions toward battery lifetime prediction using multivariate time series data. The contributions are:

- a model for SoH prediction, called AE-LSTM, based on a LSTM architecture. The LSTM is fed with data extracted from multivariate time series using an auto-encoder. Second, SOH-window-XLSTM similar to AE-LSTM for SoH prediction is proposed using TSF approach for feature extraction.
- SoHAP approach for ageing explanation using explainable artificial intelligence. SoHAP uses the post-hoc SHAP approach to highlight the degradation main features when applied to a convolutional neural network model for SoH prediction.
- approaches are validated experimentally on real-world public data of battery prognosis such as MIT, NASA and SNL.

7.4 Perspectives

The contributions presented in the manuscript can be linked to form perspectives. Indeed, as highlighted in Figure 7.1, data mining approaches, and more specifically chronicle mining, can be used together for explainability purposes and be applied in the field of battery prognostic and predictive maintenance. Similarly for text mining, machine learning and knowledge reasoning can be combined to take TRIZ inventive design a step further.

7.4.1 Crossing pattern mining and explainability for smart battery management

For the time series data, a post-hoc explainable technique, such as SHAP, is bridged to the prediction module to compute the weights of usage data on the degradation. The computed weights of the input feature create a multivariate time series in which it encapsulates all feature/SoX relations. Therefore, the generated explanations turn to be a valuable data to analyze. Two main directions could be investigated in the near future. First, mining techniques such as Matrix profile (Yeh, Kavantzias, and Keogh, 2017) could be investigated to find frequent and/or outlying degradation profiles. This will

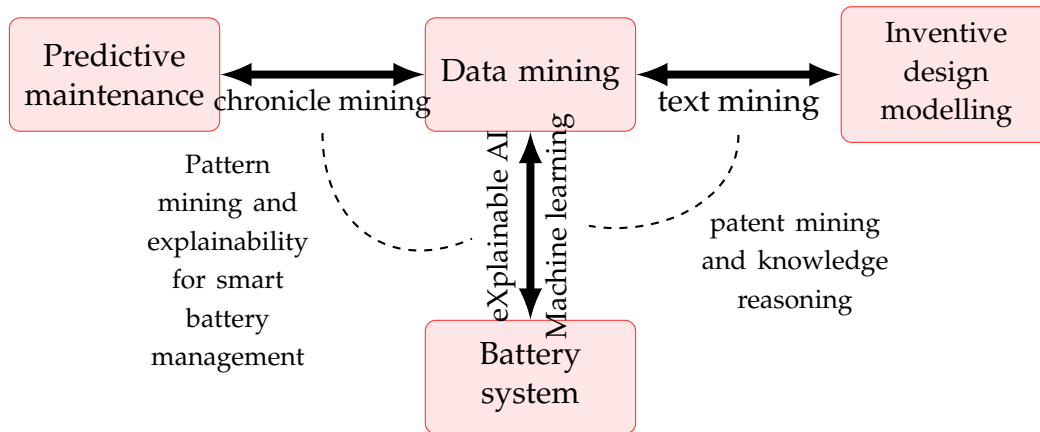


FIGURE 7.1: Cross relation between the different investigated topics

allow to create a causal relationship between input features and SoX degradation. Secondly, explanation, modelled as feature weights or counterfactual (Verma, Dickerson, and Hines, 2020), could further analyze and detect inconsistency, contradiction or find any minimal consistent sets (Jabbour et al., 2016).

As perspective, the idea of using explainable prediction could be explored since it captures all major degradation patterns while being explainable and checkable for an expert if needed. Additionally, it allows the merging of explainable degradation patterns of heterogenous sources (time series, images, expert knowledge, to cite a few). Since several explanations could be mined, as shown in Figure 7.2, a perspective could be to check the possible inconsistent or contradictory explanations. The set of degradation rules constructed is enriched by combining rules coming from the experts as well as rules built from the formalization of the explanations mentioned above. One strategy could be Semantic Technology. These rules could be written in SWRL (Semantic Web Rule Language) format and enrich the formal model representing the domain of the batteries written in OWL (Web Ontology Language). This may raise issues among the rules, such as contradiction, generality, specificity, etc. Therefore, the project develops a strategy for pruning rules according to the reliability of each source to determine which rule should remain in the rule set. In addition, from these rules/explanations, an analysis is intended to determine the causes that gave rise to these explanations by trying to associate them with physical phenomena or battery usage behaviour. The early-fusion approach underscores a preliminary extraction of representative features from the original sensor data that will be fed to the embedding layer(s) to extract multi-perspective hidden features. Afterward,

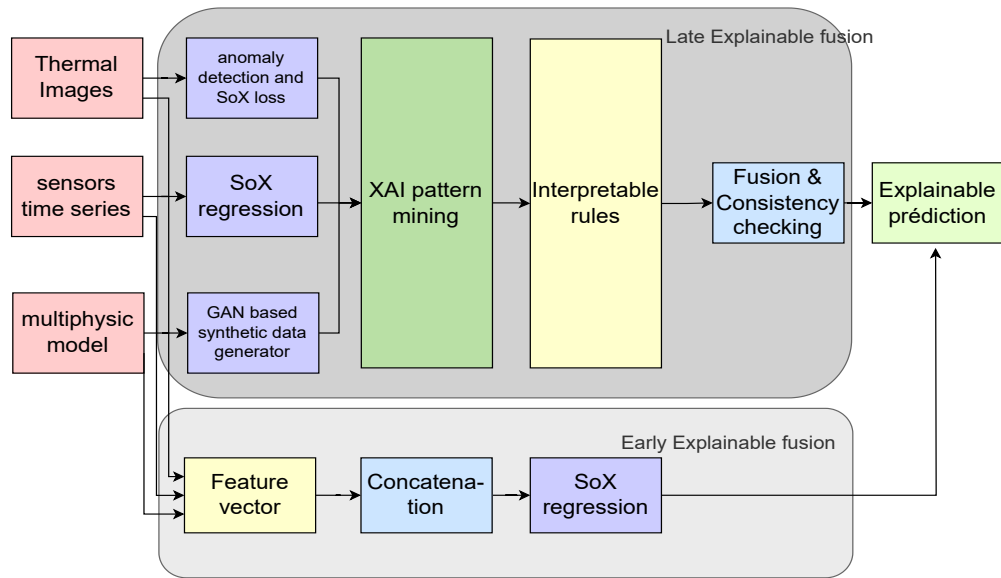


FIGURE 7.2: Multimodal explainable approach for battery prognosis

an LSTM-based neural network model is built. The explainability layer is provided through a model-agnostic based on dependency plots visual explanations. The perspective of exploring explainability with expert knowledge is under investigation in the Horizon Europe accepted project ENERGETIC¹. This idea is going to be explored through multiple collaborations with University of Luxembourg. In fact, knowledge could be brought from time series data but image as well in the use case of battery degradation. SAT or Answer Set Programming Solvers could be another path to instigate contradiction or consistency among generated explanations. This direction is currently jointly investigated with CRIL (University of Artois) and Bochum University (Germany).

7.4.2 Crossing TRIZ patent mining and knowledge reasoning

The advent of Large Language Models (LLMs) has ushered in a wave of excitement and innovation in the realm of patents and knowledge reasoning within companies. LLMs are revolutionizing the patent landscape by significantly streamlining and enhancing the patent analysis and mining. These models can rapidly analyze vast amounts of patent data, helping companies

¹The full list of accepted projects and their details are displayed in Appendix E

identify existing patents relevant to their innovations. These LLMs have a major limitation is the lack of reasoning.

The starting hypothesis is that a patent is a source that might solve a contradiction, i.e. it will allow the improvement of both evaluation parameters at the same time. To automatize the process of inventivity using TRIZ theory, several objectives are targeted: (i) analyse and mine TRIZ contradiction from patents; (ii) mine TRIZ parameters from contradictions; (iii) predict the most plausible inventive principles from TRIZ matrix; (iv) provide the user with inventive solutions from TRIZ Matrix first and from patent claims. To solve these challenges, deep learning models are combined to uncertain theories to increase the robustness of the trained model and to cope with data uncertainty. The potential association of TRIZ and AI is still not explored enough. Currently, the steps of inventiveness using TRIZ theory are based on human expert and engineers' imagination from inventive principles. We could not only automatize the patent inventive idea mining, but also strengthen the process by robust approaches coping with data uncertainty. Conditional Random Fields (CRF), a type of discriminative undirected probabilistic graphical model, is currently investigated for mining TRIZ parameters. A CRF (Lafferty, McCallum, and Pereira, 2001) models the dependencies between neighboring variables (Chu et al., 2016). In classification tasks, the CRF model computes the conditional probabilities $P(Y|X)$ with Y the labels and X the observations. While the first results are meaningful, this model would be further improved to cope with the uncertainty and imprecision. Indeed, we may ask ourselves how to predict the next sentence label when we are uncertain about the previous labels. One way to answer this question is to study the use of an uncertain theory such as the Evidence theory (Dempster, 2008). The strategy could be investigated with the LGI2A laboratory of the University of Artois specialized in the Evidence theory. A joint Master internship supervision is under discussion.

Additionally, to this day, Altshuller inventive principles have never been unproven. As Altshuller has based all his findings on statistical analysis of 400k patents, we aim to use the same approach to prove or to disprove it. To do so, from mined contradiction, we mine TRIZ parameters in the first part and the second part of the contradiction. Once mined, an approach could be using constraint programming and solvers to disprove the matrix. First, all inventive principles are introduced as facts in the program. Then, as long as we analyze patents and we mine parameters we count them. When that number becomes significant, a new constraint is added to the program

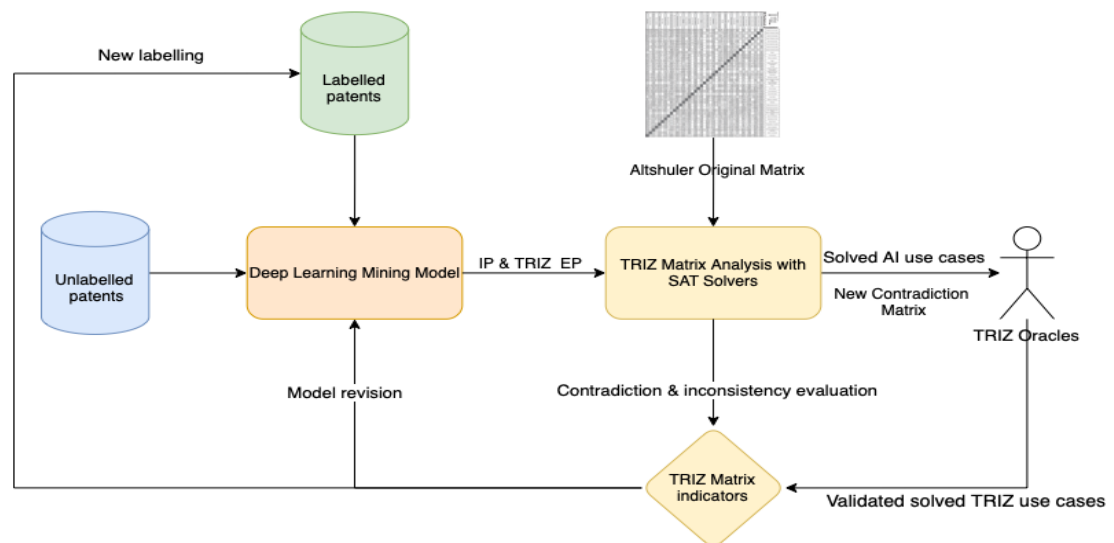


FIGURE 7.3: Deep active SAT -based learning for TRIZ matrix finding

showing the link of the found parameters to the inventive principle. If the constraint is there, we do not bring updates to the set of constraints. On the contrary, if an inventive principle is not sufficiently identified when two parameters are met in the contradiction, a constraint is added to emphasize that this relation is statistically unproven. This process could be integrated as a deep active learning process, as shown in Figure 7.3, in order to single out the unlabeled data that could be labeled by an expert to generate a new TRIZ matrix. This strategy is under study with several German partners such as Kaiserslautern Hochschule and University of Applied Sciences of Berlin.

7.4.3 Management Strategies, Industrial and Financial Prospects

This manuscript provides a comprehensive overview of our research endeavors over the past few years. Our research initiatives have yielded fruitful collaborations in both academic and industrial spheres, resulting in significant advancements in the fields of data mining and industry 4.0. This section of the manuscript delves into the future prospects linked to the future ambition of collective responsibility, call for projects or industrial collaborations.

7.4.3.1 Management Strategies and Collective Scientific Responsibilities

From a scientific standpoint, activities related to responsibility hold equal importance as purely scientific progresses. Beyond administrative duties associated with teaching, there are research obligations, including the leadership of workgroups, research projects, themes, and research teams. My ambition for collective responsibility can be seen at two levels: local and national.

Locally, one of the limitations I noticed when I arrived at INSA Strasbourg in 2017 was that the SDC research team inside the school was limited to two permanent researchers in computer science while the school clearly stated its ambition to open up its specialities to AI. As a result of several project acceptances and internal funding from the school, the team has grown to 3 permanent researchers, 2 PhDs and 4 PhD candidates currently working. Despite this growth, it is not enough to meet the internal and external demands related to the topics addressed in this manuscript. One of our first ambitions is to strengthen the team with other experienced researchers. One possibility is to recruit by applying for junior professor chair in the domain of machine learning and electrical transport.

To fund student training, I have managed to develop the teaching and computing resources center (in french: Centre de Ressources Pédagogiques et Informatiques shortened to CRPI) into a platform by investing more than €250k, obtained among other things, from a response to a call for projects, in new intensive and parallel computing equipments. In the short term, the CRPI aims to bring in more industrial training projects that can be turned into research projects while promoting doctoral training to INSA Strasbourg students.

Nationally, as SDC team leader at INSA Strasbourg I attach as much importance to scientific production as to its dissemination and popularization. Since 2022, I have been an elected member of the administrative board of the French Artificial Intelligence Association (AFIA²). Through this involvement, I am committed to scientific dissemination via participation in the organization of the Plate-Forme d'Intelligence Artificielle (PFIA³) conference, the organization of national scientific days, the organization of tutorials around emerging themes in AI or even by introducing for the first time the call for the best demonstrators. The next step would be taking in charge the full responsibility of one of the AFIA platforms. The AFIA, which is made up of several panels, is a natural facilitator of communication between researchers

²<https://afia.asso.fr>

³<https://pfia23.icube.unistra.fr>

across France. As I am also involved in the industrial panel, my ambition is to disseminate my research to industry as well as to my peers, and to raise awareness of the TRIZ inventive axis or the intelligent battery with a view to responding to joint calls for projects.

7.4.3.2 Industrial prospect

The hype surrounding AI within companies pursuing Industry 4.0 is nothing short of transformative. AI, with its machine learning algorithms and data-driven insights, has become the linchpin in the realization of Industry 4.0's potential. Companies are avidly investing in AI to optimize manufacturing processes, from predictive maintenance to electrical transport. Consequently, this has spurred progress in both the industrial and technological realms, with the aim of enhancing the effectiveness, performance, and accessibility of these solutions. Regarding our endeavors in this area at the ICube laboratory and INSA Strasbourg, INSA Strasbourg foundation serves as the bridge connecting our research within the laboratory to businesses. It facilitates the transfer of innovative solutions developed in academic laboratories in Alsace to industrial companies. It is with this in mind that, in collaboration with the SMH team⁴, an Industrial Chair in Intelligent Batteries is beginning to take shape, with a potential start-up date of January 2024. The aim of the chair is to develop hybrid, explainable and interpretable models for predicting battery health. The chair will explore the use of explainable models to understand failures or loss of product quality at the end of the production chain. The aim of this industrial chair will be to anchor its themes within ICube and to strengthen the team.

7.4.3.3 Research fundings

In addition to these scientific opportunities, it is important to recognize the ongoing need to secure funding for research projects at regional, national, and international levels. These projects should engage both academic and industrial collaborators, enabling theoretical advancements to translate into practical industrial applications.

In my research, the applied research part is very important, as the region Grand Est is home to industries of national importance: the region ranks 1st, 2nd or 3rd in France, in terms of employment, for half a dozen sectors,

⁴<https://smh.icube.unistra.fr/en/index.php/Mainpage>

notably automotive, machinery and equipment (production as well as installation and repair), metal products, electrical equipment and plastics. That's why I've always responded to calls for industrial projects such as those from the Grand Est region, ADEME and InterReg, which focus more on technology transfer and applied research. Since 2022, I have won several industrial projects with my colleague Tedjani Mesbahi from the SMH team, including the XAI-BATMAN project, which aims to develop explainable AI models for SOH prediction for electric scooter manufacturing company. In the future, one type of funding that has not been sufficiently explored is CIFRE thesis funding in collaboration with companies. We have been approached by a number of companies, including EDF and WESK companies, and following the acceptance of the ENERGETIC european project, we are considering submitting CIFRE applications with students from our training program.

At the same time, I try to answer to National Research Agency (ANR) and Horizon Europe calls for projects with a low TRL⁵ to explore fundamental research directions or open up new exploratory topics. Since 2023, As matter of fact, the Horizon Europe project ENERGETIC, which aims to develop new explainable hybrid models, is planned to continue over the next few years with new submission under the same program. The same goes for the ANR XQUALITY project, whose results will be used for other European calls for projects.

To date, the themes I have explored concern the application of AI in Industry 4.0 domains. These themes have enabled me to collaborate with several specialties and research teams within ICUBE, such as CSIP⁶, SMH and CSTB⁷, and several INSA Strasbourg departments such as Mechatronics, electrical, Mechanical engineering within the limits of the team's human capacities. One of the aims of being authorised to steer research is to be able to collaborate with other teams from other domains. As INSA Strasbourg is a school known for its architecture and civil engineering training, it is highly interesting to study a link between AI and the ageing of materials such as bitumen. Calls for internal projects or from the INSA group are conceivable in order to reinforce this theme with doctoral students and PhDs.

⁵Technology Readiness Levels (TRL) are a type of measurement system used to assess the maturity level of a particular technology.

⁶<https://csip.icube.unistra.fr/index.php/Accueil>

⁷<https://cstb.icube.unistra.fr/index.php/Accueil>

Part V
Appendices

Appendix A

TRIZ Contradiction Matrix

The "Y-axis" of the contradiction matrix stands for the parameters to be improved and the "X-axis" shows the "undesired results", that is, the parameters to be deteriorated.

A.1 39 Generic Engineering Parameters

1. Weight of moving object
2. Weight of stationary object
3. Length of moving object
4. Length of stationary object
5. Area of moving object
6. Area of stationary object
7. Volume of moving object
8. Volume of stationary object
9. Speed
10. Force (intensity)
11. Stress or pressure
12. Shape
13. Stability of object's composition
14. Strength
15. Duration of action of moving object
16. Duration of action of stationary object
17. Temperature
18. Illumination intensity
19. Use of energy by moving object
20. Use of energy by stationary object
21. Power
22. Loss of energy
23. Loss of substance

24. Loss of information
25. Loss of time
26. Quantity of substance or the matter
27. Reliability
28. Measurement accuracy
29. Manufacturing precision
30. Object-affected harmful factors
31. Object-generated harmful factors
32. Ease of manufacture
33. Ease of operation
34. Ease of repair
35. Adaptability or versatility
36. Device complexity
37. Difficulty of detecting and measuring
38. Extent of automation
39. Productivity

A.2 40 Inventive Principles

The 40 Inventive Principles (IP) are used with the contradiction matrix to solve technical contradictions. This list with the so-called "sub-principles" that intend to help clarify the meaning of the principles was taken from the TRIZ Journal ¹.

- Inventive Principle 1: Segmentation
 - Divide an object into independent parts.
 - Make an object easy to disassemble.
 - Increase the degree of fragmentation or segmentation.
- Inventive Principle 2: Taking out
 - Separate an interfering part or property from an object, or single out the only necessary part (or property) of an object.
- Inventive Principle 3: Local quality

¹<https://the-trizjournal.com/>

- Change an object's structure from uniform to non-uniform, change an external environment (or external influence) from uniform to non-uniform.
- Make each part of an object function in conditions most suitable for its operation.
- Make each part of an object fulfill a different and useful function.
- Inventive Principle 4: Asymmetry
 - Change the shape of an object from symmetrical to asymmetrical.
 - If an object is asymmetrical, increase its degree of asymmetry.
- Inventive Principle 5: Merging
 - Bring closer together (or merge) identical or similar objects, assemble identical or similar parts to perform parallel operations.
 - Make operations contiguous or parallel; bring them together in time.
- Inventive Principle 6: Universality
 - Make a part or object perform multiple functions; eliminate the need for other parts.
- Inventive Principle 7: "Nested doll"
 - Place one object inside another; place each object, in turn, inside the other.
 - Make one part pass through a cavity in the other.
- Inventive Principle 8: Anti-weight
 - To compensate for the weight of an object, merge it with other objects that provide lift.
 - To compensate for the weight of an object, make it interact with the environment (e.g. use aerodynamic, hydrodynamic, buoyancy and other forces).
- Inventive Principle 9: Preliminary anti-action
 - If it will be necessary to do an action with both harmful and useful effects, this action should be replaced with anti-actions to control harmful effects.

- Create beforehand stresses in an object that will oppose known undesirable working stresses later on.
- Inventive Principle 10: Preliminary action
 - Perform, before it is needed, the required change of an object (either fully or partially).
 - Pre-arrange objects such that they can come into action from the most convenient place and without losing time for their delivery.
- Inventive Principle 11: Beforehand cushioning
 - Prepare emergency means beforehand to compensate for the relatively low reliability of an object.
- Inventive Principle 12: Equipotentiality
 - In a potential field, limit position changes (e.g. change operating conditions to eliminate the need to raise or lower objects in a gravity field).
- Inventive Principle 13: "The other way round"
 - Invert the action(s) used to solve the problem (e.g. instead of cooling an object, heat it).
 - Make movable parts (or the external environment) fixed, and fixed parts movable).
 - Turn the object (or process) 'upside down'.
- Inventive Principle 14: Spheroidality - Curvature
 - Instead of using rectilinear parts, surfaces, or forms, use curvilinear ones; move from flat surfaces to spherical ones; from parts shaped as a cube (parallelepiped) to ball-shaped structures.
 - Use rollers, balls, spirals, domes.
 - Go from linear to rotary motion, use centrifugal forces.
- Inventive Principle 15: Dynamics
 - Allow (or design) the characteristics of an object, external environment, or process to change to be optimal or to find an optimal operating condition.

- Divide an object into parts capable of movement relative to each other.
- If an object (or process) is rigid or inflexible, make it movable or adaptive.
- Inventive Principle 16: Partial or excessive actions
 - If 100 percent of an object is hard to achieve using a given solution method then, by using ‘slightly less’ or ‘slightly more’ of the same method, the problem may be considerably easier to solve.
- Inventive Principle 17: Another dimension
 - To move an object in two- or three-dimensional space.
 - Use a multi-story arrangement of objects instead of a single-story arrangement.
 - Tilt or re-orient the object, lay it on its side.
 - Use ‘another side’ of a given area.
- Inventive Principle 18: Mechanical vibration
 - Cause an object to oscillate or vibrate.
 - Increase its frequency (even up to the ultrasonic).
 - Use an object’s resonant frequency.
 - Use piezoelectric vibrators instead of mechanical ones.
 - Use combined ultrasonic and electromagnetic field oscillations.
- Inventive Principle 19: Periodic action
 - Instead of continuous action, use periodic or pulsating actions.
 - If an action is already periodic, change the periodic magnitude or frequency.
 - Use pauses between impulses to perform a different action.
- Inventive Principle 20: Continuity of useful action
 - Carry on work continuously; make all parts of an object work at full load, all the time.
 - Eliminate all idle or intermittent actions or work.
- Inventive Principle 21: Skipping

- Conduct a process , or certain stages (e.g. destructible, harmful or hazardous operations) at high speed.
- Inventive Principle 22: “Blessing in disguise” or “Turn Lemons into Lemonade”
 - Use harmful factors (particularly, harmful effects of the environment or surroundings) to achieve a positive effect.
 - Eliminate the primary harmful action by adding it to another harmful action to resolve the problem.
 - Amplify a harmful factor to such a degree that it is no longer harmful.
- Inventive Principle 23: Feedback
 - Introduce feedback (referring back, cross-checking) to improve a process or action.
 - If feedback is already used, change its magnitude or influence.
- Inventive Principle 24: ‘Intermediary’
 - Use an intermediary carrier article or intermediary process.
 - Merge one object temporarily with another (which can be easily removed).
- Inventive Principle 25: Self-service
 - Make an object serve itself by performing auxiliary helpful functions.
 - Use waste resources, energy, or substances.
- Inventive Principle 26: Copying
 - Instead of an unavailable, expensive, fragile object, use simpler and inexpensive copies.
 - Replace an object, or process with optical copies.
 - If visible optical copies are already used, move to infrared or ultraviolet copies.
- Inventive Principle 27: Cheap short-living objects

- Replace an inexpensive object with a multiple of inexpensive objects, comprising certain qualities (such as service life, for instance).
- Inventive Principle 28: Mechanics substitution
 - Replace a mechanical means with a sensory (optical, acoustic, taste or smell) means.
 - Use electric, magnetic and electromagnetic fields to interact with the object.
 - Change from static to movable fields, from unstructured fields to those having structure.
 - Use fields in conjunction with field-activated (e.g. ferromagnetic) particles.
- Inventive Principle 29: Pneumatics and hydraulics
 - Use gas and liquid parts of an object instead of solid parts (e.g. inflatable, filled with liquids, air cushion, hydrostatic, hydro-reactive).
- Inventive Principle 30: Flexible shells and thin films
 - Use flexible shells and thin films instead of three dimensional structures.
 - Isolate the object from the external environment using flexible shells and thin films.
- Inventive Principle 31: Porous materials
 - Make an object porous or add porous elements (inserts, coatings, etc.).
 - If an object is already porous, use the pores to introduce a useful substance or function.
- Inventive Principle 32: Color changes
 - Change the color of an object or its external environment.
 - Change the transparency of an object or its external environment.
- Inventive Principle 33: Homogeneity

- Make objects interacting with a given object of the same material (or material with identical properties).
- Inventive Principle 34: Discarding and recovering
 - Make portions of an object that have fulfilled their functions go away (discard by dissolving, evaporating, etc.) or modify these directly during operation.
 - Conversely, restore consumable parts of an object directly in operation.
- Inventive Principle 35: Change of physical and chemical parameters
 - Change the object's aggregate state.
 - Change concentration or consistency of the object.
 - Change the degree of flexibility of the object.
 - Change the temperature of the object or environment.
- Inventive Principle 36: Phase transitions
 - Use phenomena occurring during phase transitions (e.g. volume changes, loss or absorption of heat, etc.).
- Inventive Principle 37: Thermal expansion
 - Use thermal expansion (or contraction) of materials.
 - If thermal expansion is being used, use multiple materials with different coefficients of thermal expansion.
- Inventive Principle 38: Strong oxidants
 - Replace common air with oxygen-enriched air.
 - Replace enriched air with pure oxygen.
 - Expose air or oxygen to ionizing radiation.
 - Use ionized oxygen.
 - Replace ozonized (or ionized) oxygen with ozone.
- Inventive Principle 39: Inert atmosphere
 - Use inert gases instead of usual ones.
 - Add neutral parts or additives to the object.

- Inventive Principle 40: Composite materials
 - Change from uniform to composite (multiple) materials.

A.3 76 Inventive Standards

A.3.1 Some Extra Precision on Inventive Standards

The 76 Inventive Standard Solutions (IS) are divided into five classes with various sub-classes that are used depending on the type of engineering problem they solve. The five classes are:

- Class 1: Building and Destruction of Substance-Field Models (13 IS)
- Class 2: Development of Substance-Field Models (23 IS)
- Class 3 Transition to Super-system and Micro level (6 IS)
- Class 4: Standards for Detection and Measuring (17 IS)
- Class 5: Standards on Application of Standards (17 IS)

This appendix describes the classes and the sub-classes.

A.3.1.1 Class 1: Building and Destruction of Substance-Field Models

Class 1 aims to solve problems by building or destroying the Su-Field Models if they are incomplete or have harmful functions. Class 1 contains two sub-classes containing 13 IS:

Sub-class 1.1 Building of Su-Fields (if incomplete) (8 IS)

The major recommendations from this sub-class are:

- Make the Su-Field complete.
- Make it minimally workable by introducing an internal additive.
- Make it minimally workable by introducing an external additive.
- Use minimal - maximal mode (add more and remove the extras; add less and enhance locally).

Sub-class 1.2 Destruction of Su-Field (harms) (5 IS)

The major recommendations from this sub-class are:

- Introduce a third substance between the given two substances.
- Introduce a third substance from the super-system.
- Introduce a third substance that is a modification of one of the given two substances.
- Introduce a sacrificial substance.
- Introduce a field that counteracts the harmful field.

A.3.1.2 Class 2: Development of Substance-Field Models

This class is used to improve the efficiency of engineered systems by introducing small modifications. It provides conceptual solutions on how to improve and develop the system. The main recommendations in this class are:

- Use of chain Su-Fields
- Use of double Su-Fields
- Segmentation (including porosity increase)
- Dynamisation
- Rhythm coordination
- Use of magnetic substances.

Class 2 contains 4 sub-classes and 23 IS:

*Sub-class 2.1 Transition to complex Su-Field Models (2 IS)**Sub-class 2.2 Evolution of Su-Field Models (6 IS)**Sub-class 2.3 Evolution of rhythms (3 IS)**Sub-class 2.4 Complex forced Su-Field Models (12 IS)*

A.3.1.3 Class 3: System Transitions and Evolution-Transition to Super-system and Sub-system

Problems in this class are solved by developing solutions at different levels in the system (super-systems or sub-systems). The main recommendation in this class is how to improve the system by combining elements or combining with other systems. Class 3 contains 2 sub-classes containing 6 IS:

Sub-class 3.1 Simplicity-complexity-simplicity (mono-bi-poly) and increasing flexibility and dynamisation (Transition to super-system and to bi and poly systems; use no links, rigid links, flexible links, "field" links) (5 IS)

Sub-class 3.2 Transition to micro-level (examine the sub-system, use smart substances) (1 IS)

A.3.1.4 Class 4: Solutions for Detection and Measurement

This class is used for solving measuring or detection problems in engineering systems. These solutions contain many distinguished features, especially the use of indirect methods and the use of copies. The major recommendations of this class are:

- Try to change the system so that there is no need to measure/detect.
- Measure a copy.
- Introduce a substance that generates a field (introduce a mark internally or externally).

Class 4 contains 5 sub-classes and 17 IS:

Sub-class 4.1 Indirect Methods (3 IS)

Sub-class 4.2 Create or Build a Measurement System (4 IS)

Sub-class 4.3 Enhancing the Measurement System (3 IS)

Sub-class 4.4 Measure Ferromagnetic-field (5 IS)

Sub-class 4.5 Direction of Evolution of the Measuring Systems (2 IS)

A.3.1.5 Class 5: Standards on Application of Standards

With the help of previous four classes of the Standard Solutions, Class 5 is additionally helpful for the further general improvements and simplification of systems. These Standard Solutions provide recommendations of how to introduce new substances or fields or use scientific effects more effectively after applying the relevant Standard Solutions in the previous four classes.

Class 5 solutions are helpful when simplifying or pruning the system to remove components or reduce the intensity of the associated interactions. The first four classes of standard solutions mentioned above often lead to increased complexity of the solution because something is often added to the system in order to solve the problem. This fifth class of solutions illustrates how to achieve something extra by simplifying, but without introducing anything new. The useful recommendations from this class are:

- Instead of a substance, introduce a field.
- Instead of a substance, introduce a void.
- Introduce a substance for a limited time.
- Introduce a little bit of a substance, but in a very concentrated way.
- Use phase changes.
- Get the substance or environment to change themselves to solve the problem.
- Use segmentation.

Class 5 contains 5 sub-classes and 17 IS:

Sub-class 5.1 Indirect methods for introducing substances under restricted conditions (4 IS)

Sub-class 5.2 Introducing fields under restricted conditions (3 IS)

Sub-class 5.3 Phase transitions (5 IS)

Sub-class 5.4 Clever use of natural phenomena (2 IS)

Sub-class 5.5 Generating higher or lower forms of substances (3 IS)

Appendix B

Evidence theory

In this appendix, we introduce the fundamentals of *evidence theory*. The evidence theory also called the *belief function theory* was introduced by Dempster (Dempster, 1968; Dempster, 2008) in order to represent some imprecise probabilities with upper and lower probabilities. Then, it was mathematically formalised by Shafer (Shafer, 1976a).

Evidence theory is used for representing imperfect (uncertain, imprecise and/or incomplete) information. In this section, we present the main concepts of this theory. The frame of discernment is the set of N possible answers for a treated problem and generally denoted θ . It is composed of exhaustive and exclusive hypotheses:

$$\theta = (H_1, H_2, \dots, H_N).$$

These elements are assumed to be mutually exclusive and exhaustive. From the frame of discernment θ , we deduce the set 2^θ containing all the 2^N subsets A of θ :

$$2^\theta = \{A, A \subseteq \theta\} = \{H_1, H_2, \dots, H_N, H_1 \cup H_2, \dots, \theta\}.$$

This set constitutes a reference to assess the veracity of any proposal. A Basic Belief Assignment (BBA) m is the mapping from elements of the power set 2^θ onto $[0, 1]$ such that:

$$m : 2^\theta \rightarrow [0, 1]$$

having as constraints:

$$\begin{cases} \sum_{A \subseteq \theta} m(A) = 1 \\ m(\emptyset) = 0. \end{cases} \quad (\text{B.1})$$

Each subsets A of 2^θ verifying $m(A) > 0$ is called focal elements. Constraining $m(\emptyset) = 0$ is the normalised form of a BBA and this corresponds to a closed-world assumption (Smets and Kennes, 1994), while allowing

$m(\emptyset) > 0$ corresponds to an open world assumption (Smets and Kennes, 1994).

The belief function offers many advantages. One of its proposed assets is the information fusion allowing extracting the more veracious proposition from a multi-source context. This benefit is granted by the combination rules. Several operators were defined such the conjunctive rule allowing fusion without any normalisation (conflict management). For two sources S_1 and S_2 having respectively m_1 and m_2 as BBA, the conjunctive rule is defined by:

$$m_{\odot}(A) = m_1 \odot m_2(A) = \sum_{B \cap C = A} m_1(B) * m_2(C); \forall A \subseteq \theta. \quad (\text{B.2})$$

A normalised version of conjunctive rule proposed by Dempster (Dempster, 2008) integrates a conflict management approach that redistributes the generated conflictual mass. The Dempster's rule of combination is defined as follows:

$$m_{\oplus}(A) = m_1 \oplus m_2(A) = \frac{1}{1 - K} \sum_{B \cap C = A} m_1(B) * m_2(C) = \frac{1}{1 - K} m_{\odot}(A) \quad (\text{B.3})$$

where $\forall A \subseteq \theta, A \neq \emptyset$ and K is defined as:

$$K = \sum_{B \cap C = \emptyset} m_1(B) * m_2(C) = m_{\cap}(\emptyset) \quad (\text{B.4})$$

K represents the conflict mass between m_1 and m_2 .

In literature, among several functions that were proposed, we distinguish the pignistic probability. The pignistic probability denoted $BetP$ was proposed by Smets (Smets and Kennes, 1994) within his Transferable Belief Model (TBM) approach. TBM is based on the differentiation between the knowledge representation and decision-making level.

In the decision phase, the pignistic transformation consists in distributing equiprobably the mass of a proposition A on its sub-hypotheses, formally:

$$BetP(H_n) = \sum_{A \subseteq \theta} \frac{|H_n \cap A|}{|A|} * m(A); \forall H_n \in \theta \quad (\text{B.5})$$

Appendix C

Introduction to Neural Networks

Neural networks, which are often referred to as Artificial Intelligence, are a family of algorithms inspired by the behavior of the human brain and neurons. The purpose is to draw inspiration from the human brain to build algorithms capable of learning complex tasks as quickly as possible. The concept of neurons was introduced in 1943 by McCulloch and Pitts when computer science was still in its infancy. The real integration of this concept in computer science came more than 15 years later with Rosenblatt in 1959. His goal was to understand how the retina works and to recognize patterns in images. Rosenblatt created the perceptron, considered as the first artificial neuron and the basis of neural networks.

The 1960's and 1970's show a progressive disinterest in artificial neurons, as perceptrons fail to convince; their interest and research is more oriented towards symbolic approaches. Several major works were carried out in the 1980's, notably by Hopking who introduced gradient backpropagation in 1982.

Neural networks were massively adopted in the early 2000's with the development of parallel computing capabilities that allowed the construction of more complex and efficient networks.

C.1 Formal neuron and neural networks

The model of the formal neuron (Figure C.1) is based on that of a real neuron with incoming and outgoing links between the neurons (synapses, axons).

The output of a neuron will be the result of the application of an activation function on the linear combination of its inputs plus a constant called bias. The pre-activation z will thus be defined in this way:

$$z = \sum_{i=1}^n x_i w_i + w_0 \quad (\text{C.1})$$

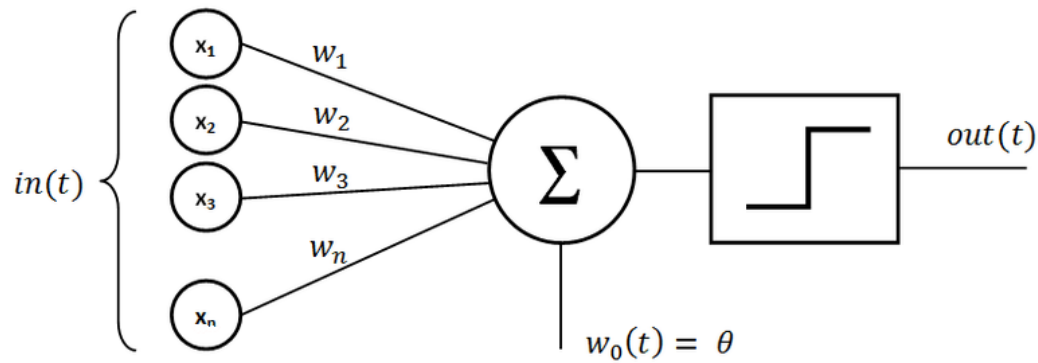


FIGURE C.1: Perceptron

with x_i the input i of the neuron and w_i the weight corresponding to the input i . The weights are in fact parameters associated to the neuron whose value will be learned during the training.

The *sigma* activation function of a perceptron is originally a step function, so the output is binary. The bias replaces the threshold applied to the output to decide when the output should be 1 or 0. However, other activation functions have been developed; the best known of them are the following:

C.1.0.0.1 Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (\text{C.2})$$

C.1.0.0.2 Softmax (N is the number of element in x)

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{c=1}^N e^{x_c}} \quad (\text{C.3})$$

C.1.0.0.3 Rectified Linear Unit (ReLU)

$$\sigma(x) = \max(0, x) \quad (\text{C.4})$$

C.1.0.0.4 TanH

$$\sigma(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (\text{C.5})$$

These activation functions always have the particularity of being non-linear in order to be able to model more complicated behaviors, but they also have the advantage of being differentiable over their entire range of definition, which simplifies the training by gradient descent (Part [C.2](#)).

Neural networks are made mostly of a set of neurons organized in layers as shown in Figure C.2. Each neuron of a layer will be linked to all the neurons of the previous layer. The neurons of the same layer are not linked. Thus, the outputs of the last layer will be functions of all the parameters of all the neurons and of the inputs of the network. A neural network can thus be modeled by a transfer function f such that:

$$\hat{y} = f(x_i, w_i) \quad (\text{C.6})$$

with \hat{y} the network's output, x_i the inputs and w_i the weights (or parameters) of the network.

The number of neurons in the last layer is related to the task of the network. For a regression (prediction of a real value), one neuron in output will be enough. For classification, there will be as many neurons as there are classes. In this case the i neuron will give the probability of the i class.

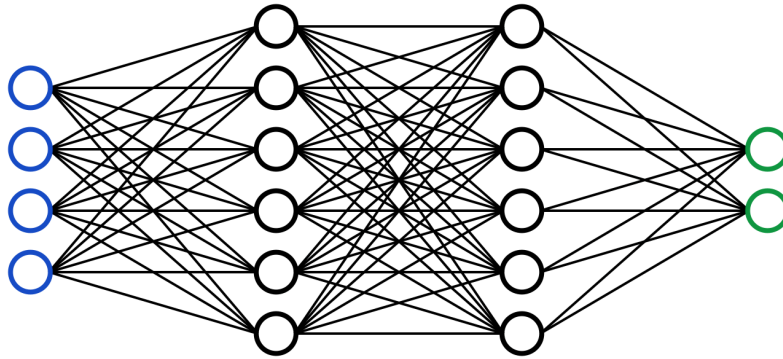


FIGURE C.2: Neural Network

C.2 Training

A neural network can be trained with annotated data, i.e. for a set of N observations x^1, x^2, \dots, x^N , the values of the N variables to predict y^1, y^2, \dots, y^N are provided. We define the objective function as the theoretical function f allowing to predict y^i from an observation x^i :

$$\forall i \in [1, N], f(x^i) = y^i \quad (\text{C.7})$$

The goal of training a neural network is to build a surrogate model for this function. The neural network can be seen as a mathematical function g , non-linear and dependent on the values of the parameters associated to

each neuron. The goal of training is to modify these parameters so that g approximates f . This type of training is called supervised learning.

C.2.1 Loss

To measure the distance between g and f , a measure called Loss is introduced. The Loss L is computed from the distances between g and f for all known pairs $(x^i, f(x^i) = y^i)$. The most known losses for regression tasks (prediction of real values) are the Mean Absolute Error and Mean Square Error:

C.2.1.0.1 Mean Absolute Error (MSE)

$$L = \frac{1}{N} \sum_{i=1}^N |y^i - \hat{y}^i| \quad (\text{C.8})$$

C.2.1.0.2 Mean Square Error (MSE)

$$L = \frac{1}{N} \sum_{i=1}^N \|y^i - \hat{y}^i\|^2 \quad (\text{C.9})$$

For classification (prediction of the membership to a category called class, e.g. "car", "bus") the cross-entropy and the negative log likelihood are the most commonly used. They are in fact very similar. They both measure how correct the model is on the prediction and the cross entropy has an extra term allowing to take into account the other classes and to measure the weight put in the prediction on the bad classes.

C.2.1.0.3 Cross-entropy

$$L = -\frac{1}{N} \sum_{i=1}^N y^i \log(\hat{y}^i) + (1 - y^i) \log(1 - \hat{y}^i) \quad (\text{C.10})$$

C.2.1.0.4 Negative log likelihood

$$L = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}^i) \quad (\text{C.11})$$

C.2.2 Error backpropagation

Once the error between expected and actual predictions is calculated, the network parameters must be updated to decrease this discrepancy. This step is called backpropagation of the error. The name "backpropagation" refers to the mechanism of predicting an output from an input which is called forward propagation.

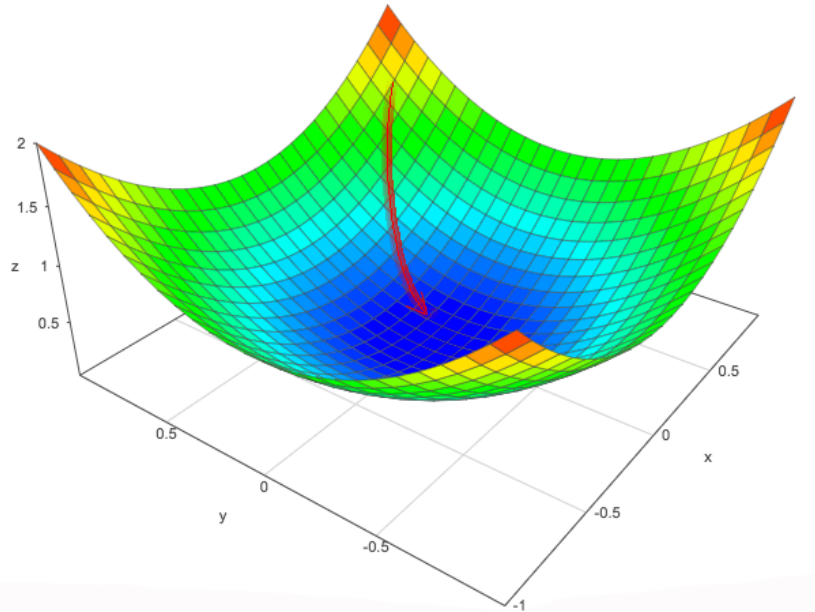


FIGURE C.3: Gradient descent

A simplified representation of the loss as a function of the values of the parameters in the network (here only two parameters) is shown in Figure C.3. At the beginning of the training, the network parameters are randomly initialized and a first loss value is estimated. The goal of the training is to reach the global minimum of the loss function. In a favorable case, with no local minimum, the fastest way is to follow the steepest slope indicated by the gradient $\frac{\delta L}{\delta w}$ (red arrow on Figure C.3). This is the gradient descent. Each parameter is updated in the following way:

$$w = w - lr * \frac{1}{N} \sum_{i=1}^N \frac{\delta L_i}{\delta w} \quad (\text{C.12})$$

with lr the learning rate ,i.e. the speed at which the parameters are updated, and L_i the value of the loss for a couple (x_i, y_i) . The variation of the parameters is then deduced:

$$\Delta w = -lr \frac{\delta L}{\delta w} \quad (\text{C.13})$$

To evaluate the variation of the loss between steps j and $j + 1$, and to verify that it is negative since it is the training purpose, we can use the following formulas:

$$L^{j+1} = L^j + \Delta L \quad (\text{C.14})$$

$$\Delta L = \Delta w * \frac{\delta L}{\delta w} \quad (\text{C.15})$$

This amounts to a linear approximation in the neighborhood of the operating point. With the Δw presented previously we obtain:

$$\Delta L = -lr \frac{\delta L}{\delta w} \frac{\delta L}{\delta w} \quad (\text{C.16})$$

and thus:

$$\Delta L = -lr \frac{\delta L^2}{\delta w} \quad (\text{C.17})$$

which is less than or equal to 0 if we choose $lr > 0$. Thus, by applying this strategy, the loss will converge to a minimum. The larger the learning rate lr is, the greater the capacity to exceed the local minimums and the faster the learning will be but the algorithm will also be less stable and will not always converge. Indeed, as everything relies on a linear approximation in the neighborhood of the operating point, the further away from this point with a large lr the more false this approximation is. In most of the current learning strategies, the lr is adjusted during the training with often a high value at the beginning of the training and a lower value at the end. Certain activation functions, like the sigmoid function, squishes a large input space into a small input space between 0 and 1. Therefore, a large change in the input of the sigmoid function will cause a small change in the output. Hence, the derivative becomes small. A small gradient means that the weights and biases of the initial layers will not be updated effectively with each training session. Since these initial layers are often crucial to recognizing the core elements of the input data, it can lead to overall inaccuracy of the whole network. This problem is commonly called the vanishing gradient problem.

C.3 Training and specialization

The training of a neural network is supervised, so annotated data (dataset) is required. This data must be numerous to allow optimal learning of each of the network parameters. The smaller the amount of data available, the higher

the chances of overfitting. Overfitting occurs when the network performs so well on training data that its ability to generalize and perform well on new data decreases.

However, access to a large amount of annotated data for a specific task is not always guaranteed. This is why transfer learning has been introduced. It consists in learning the network on a task close to, but different from, the goal. Then, the network is trained on the data associated with the target task, which is less numerous, with a very low learning rate to avoid overfitting.

Other training methods are also possible. Reinforcement learning does not use losses as such but aims at maximizing a "reward" according to the prediction of the network through trial and error. The gradient is still used but to maximize the reward instead of minimizing the error.

Appendix D

Author's scientific publications

My scientific production started in 2011 with the publication of my first paper as an attempt to put forward my master's research results. Indeed, in my early days, I attended conferences on the treatment of uncertainty through the theory of belief functions. In addition to conferences ranked and known in the uncertainty community such as IPMU, which, together with ECSQARU form the reference conferences in the belief function theory community, I participated in other conferences such as Belief, IUKM, KSE and ICMSAO which are also carried by influential people in my research area (for example: Arthur Dempster, Glenn Shafer, Thierry Denoeux, Sebastien Destercke). The multidisciplinary nature of my thesis (belief function theory, data mining and machine learning), has changed my attendance of conferences. Indeed, I have opened up to the fields of Artificial Intelligence and data mining, and started targeting others that are thriving. Since 2015, I have participated in 20 international conferences including several ranking A and A*. As far as my publication in international journals is concerned, I have endeavored to add value to my most notable contributions through the extensions. Therefore, most of my papers have been extended in journals with at least a C rank. Additionally, since supervised PhD's are multidisciplinary, I always looked to publish in non-computer science journals, in full agreement with the PhD main supervisors.

Articles in international peer-reviewed journals listed in international databases

Jorge, I., **Samet, A.**, Mesbahi, T., Boné, R.: Time Series Feature extraction for Lithium-Ion batteries State-Of-Health prediction. *Journal of Energy Storage* (to appear). <https://doi.org/10.1016/j.est.2022.106436>.

Guarino, G., **Samet, A.**, Cavallucci, D.: PaTRIZ: A framework for mining TRIZ contradictions in patents. *Expert Syst. Appl.* 207: 117942 (2022). <https://doi.org/10.1016/j.eswa.2022.117942>.

Ni, X., **Samet, A.**, Cavallucci, D.: Similarity-based approach for inventive design solutions assistance. *J. Intell. Manuf.* 33(6): 1681-1698 (2022). <https://doi.org/10.1007/s10845-021-01749-4>

Cao, Q., Zanni-Merk, C., **Samet, A.**, Reich, C., de Bertrand de Beuvron, F., A. Beckmann, C. Giannetti: KSPMI: A Knowledge-based System for Predictive Maintenance in Industry 4.0. *Robotics Comput. Integr. Manuf.* 74: 102281 (2022). <https://doi.org/10.1016/j.rcim.2021.102281>

Cao, Q., Zanni-Merk, C., **Samet, A.**, de Beuvron, F. et Reich, C.: Using Rule Quality Measures for Rule Base Refinement in Knowledge-Based Predictive Maintenance Systems. *Cybernetics and Systems*, 1-16, 2020. <https://doi.org/10.1080/01969722.2019.1705550>

Sellami, C., Miranda, C., **Samet, A.**, Bach-Tobji, M. A, et de Beuvron, F.: On mining frequent chronicles for machine failure prediction. *Journal of Intelligent Manufacturing*, 1-17, 2019. <https://doi.org/10.1007/s10845-019-01492-x>

Samet, A., Bouzemrak, Y., et Lefevre, E. : Evidential Data Mining to design Supply Chain Network under Uncertainty, *Logistics research journal*, 10(1): 8 , 2017.

Samet, A., Lefevre, E. et Ben Yahia, S. : Evidential Data mining: Precise support and confidence, *Journal of Intelligent Information Systems*, 47(1), 135-163, 2016.

Samet, A., Hammami, I., Lefevre, E. et Ben Yahia, S. : Reliability estimation measure: Generic Discounting Approach. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(7), 2015.

Samet, A., Lefevre, E. et Ben Yahia, S. : Integration of extra-information for belief function theory conflict management problem through generic association rules. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 22(04):531-551, 2014 (IF 0.899).

Papers in international conferences with refereed proceedings

Hamouni, A., Arbaoui, S., Ayadi, A., Giustozzi, F., Mesbahi, T., **Samet, A.**: OntoSoC : An ontology-based approach to battery pack SoC estimation. *in proceedings of 27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Athens, Greece 2023 (to appear).(Core 2018 : Rank B).

Heitzmann, T., **Samet, A.**, Soufi, C., Mesbahi, T. and Boné, R.: SocHAP: a new data driven explainable prediction of battery state of charge.*in proceedings of 23RD International Conference on Computational Science*, Prague, Czech republic, 2023. (to appear). (Core 2018 : Rank A)

Venuti, M., **Samet, A.**, Giustozzi, F., Heitzmann, T., Mesbahi, T.: Fouille de séries temporelles pour l'explicabilité de la dégradation de l'état de charge des batteries Lithium-ions. *23ème conférence francophone sur l'extraction et la gestion des connaissances* 2023: 305-312.(ERA 2010 : Rank C).

Guarino, G., Samet, A., Cavallucci, D.: Réseau antagoniste génératif pour la fouille des contradictions TRIZ dans les brevets. *22ème conférence francophone sur l'extraction et la gestion des connaissances* 2022: 379-386.(ERA 2010 : Rank C).

Douard, N., **Samet, A.**, Giakos, G. C., Cavallucci, D.: Bridging Two Different Domains to Pair Their Inherent Problem-Solution Text Contents: Applications to Quantum Sensing and Biology. *in proceedings of TRIZ Future conference* 2022: 61-69

Joël Azemena, H., Ayadi, A., **Samet, A.**: Explainable Artificial Intelligent as a solution approach to the Duck Curve problem. *in proceedings of 26th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Verona, Italy 2022: 2747-2756.([Core 2018 : Rank B](#)).

Ni, X., **Samet, A.**, Chibane, H., Cavallucci, D.: PatRIS: Patent Ranking Inventive Solutions. *in proceedings of the 32nd DEXA Conferences(2)*, virtual conference, 2021: 295-309.([Core 2018 : Rank B](#)).

Guarino, G., **Samet, A.**, Nafi, A., Cavallucci, D.: PaGAN: Generative Adversarial Network for Patent understanding. *in proceedings of 21st IEEE International Conference on Data Mining*, Auckland, New Zealand 2021: 1084-1089.([Core 2018 : Rank A*](#)).

Audin, P., Jorge, I., Mesbahi, T., **Samet, A.**, de Bertrand de Beuvron, F., Boné, R.: Auto-encoder LSTM for Li-ion SOH prediction: a comparative study on various benchmark datasets. *in proceedings of 20th IEEE International Conference on Machine Learning and Applications*, virtually online, 2021: 1529-1536.([Core 2018 : Rank C](#)).

Guarino, G., **Samet, A.**, Cavallucci, D.: Patent Specialization for Deep Learning Information Retrieval Algorithms. TFC 2021: 162-169

Ni, X., **Samet, A.**, Cavallucci, D.: Replicating TRIZ Reasoning Through Deep Learning. *in proceedings of TRIZ Future conference* 2021: 330-339

Jorge, I., Mesbahi, T., Paul, T., **Samet, A.**: Study and simulation of an electric scooter based on a dynamic modelling approach. EVER 2020: 1-6

Guyet, T. , Besnard, P., Ben Salha, N., **Samet, A.** et Lachiche, C. : Énumération des occurrences d'une chronique, *20ème conférence francophone sur l'Extraction et la Gestion des Connaissances EGC2020*, Bruxelles, Belgique, 2020 ([ERA 2010 : Rank C](#)).

Cao, Q., **Samet, A.**, Zanni-Merk, C., de Beuvron, F. et Reich, C. : Combining Evidential Clustering and Ontology Reasoning for Failure Prediction in Predictive Maintenance, *In Proceeding of 12th International Conference on Agent and Artificial Intelligence*, Valletta, Malta, 2020 ([Core 2018 : Rank C](#)).

Ni, X., **Samet, A.** et Cavallucci, D. : An Approach Merging the IDM-Related Knowledge, *In Proceeding of TRIZ Future Conference*, Marrakesh, Morocco 2019, pages 147-158, 2019 ([Conférence référence dans la conception inventive](#)).

Cao, Q., **Samet, A.**, Zanni-Merk, C., de Beuvron, F. et Reich, C. : An Ontology-based Approach for Failure Classification in Predictive Maintenance Using Fuzzy C-means and SWRL Rules, *In Proceeding 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Budapest, Hungary, pages 630-639, 2019 ([Core 2018 : Rank B](#)).

Ayadi, A., **Samet, A.**, de Beuvron, F. et Zanni-Merk, C. : Ontology population with deep learning-based NLP: a case study on the Biomolecular Network Ontology, *In Proceeding 23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, Budapest, Hungary, pages 572-581, 2019 ([Core 2018 : Rank B](#)).

Sellami, C., **Samet, A.** et Bach Tobji, M. A.: Frequent Chronicle Mining: Application on Predictive Maintenance. *In Proceeding 17th International Conference On Machine Learning And Applications ICMLA*, Orlando, Miami, 1388-1393 2018 ([Core 2018 : Rank C](#)).

Samet, A., Guyet, T. , Négrevergne, B., Dao, T-T., Hoang, T. N., et Ho Ba Tho, M. C.: Expert Opinion Extraction from a Biomedical Database. *In Proceeding of 14th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty ECSQARU* Lugano, Switzerland, 135-145, 2017 ([Core 2017 : Rank C](#)).

Samet, A., Guyet, T. et Négrevergne, B.: Mining Rare Sequential Patterns with ASP. In *Proceeding of 27th International Conference on Inductive Logic Programming ILP (Late Breaking Papers)* Orleans, France, 51-60, 2017 ([Core 2017 : Rank B](#)).

Samet, A., Gaudin, T., Lu, H., Wadouachi, A., Pourceau, G., Van Hecke, E., Pezron, I., EL Kirat, K. et Dao, T-T. : Predictive model based on the evidence theory for assessing Critical Micelle Concentration property, In *Proceeding of 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Eindhoven, Netherlands, 510-522, 2016. ([Core 2015 : Rank C](#))

Samet, A., Raddaoui, B., Dao, T-T. et Hadjali, A. : Argumentation Framework based on Evidence Theory, In *Proceeding of 16th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Eindhoven, Netherlands, 253-264, 2016. ([Core 2015 : Rank C](#))

Samet, A. et Dao, T-T., Bounded Support and Confidence over Evidential Databases, In *Proceeding of the International Conference on Computational Science, ICCS 2016*, San Diego, California, 1822-1833, 2016. ([Core 2015 : Rank A](#))

Raddaoui, B. et **Samet, A.**: Mining Frequent Patterns from Correlated Incomplete Databases. In *Proceeding of the 8th International Conference on Agents and Artificial Intelligence, ICAART'2016*, Roma, Italy, 377-384, 2016. ([Core 2015 : Rank C](#))

Samet, A. et Dao, T-T. : Mining over a Reliable Evidential Database : Application on amphiphilic chemical database. In *Proceeding of 14th International Conference on Machine Learning and Applications, ICMLA'2015*, Florida, Miami, USA, pages 1257-1262, 2015. ([Core 2015 : Rank C](#))

Nouaouri, I., **Samet, A.** et Allaoui, H. : Evidential Data Mining for Length of Stay (LOS) Prediction Problem, *In Proceeding of 11th IEEE International Conference on Automation Science and Engineering*, Gothenburg, Sweden, pages 1415-1420, 2015.

Samet, A., Lefevre, E. et Ben Yahia, S. : Belief function classification with conflict management: application on forest image *to appear In Proceedings of 10th International Conference on Signal-Image Technology and Internet-Based Systems*, Marrakesh, Morocco, 2014.

Samet, A., Lefevre, E. et Ben Yahia, S. : Evidential database : a new generalization of databases ? *In Proceedings of 3rd International Conference on Belief Functions*, Belief 2014, Oxford, UK, pages 105-114, 2014.

Samet, A., Lefevre, E., et Ben Yahia, S. : Classification with evidential associative rules. *In Proceedings of 15th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, Montpellier, France, pages 25-35, 2014. ([Core 2015 : Rank C](#))

Samet, A., Lefevre, E. et Ben Yahia, S. : Reliability estimation with extrinsic and intrinsic measure in belief function theory. *In Proceedings of 5th International Conference on Modeling, Simulation and Applied Optimization*, ICMSAO'13, Hammamet, Tunisia, pages 1-6, 2013.

Samet, A., Lefevre, E. et Ben Yahia, S. : Mining frequent itemsets in evidential database. *In Proceedings of the fifth International Conference on Knowledge and Systems Engeneering*, Hanoi, Vietnam, pages 377-388, 2013.

Samet, A., Hammami, I., Lefevre, E. et Hamouda, A. : Generic discounting evaluation approach for urban image classification. *In Proceedings of 3rd international symposium on Integrated Uncertainty in Knowledge Modelling and Decision Making*, IUKM'2013, Beijing, China, pages 79-90, 2013.

Samet, A., Ben Dhiab, Z., Hamouda, A. et Lefevre, E. : Classification of high-resolution remote sensing image by adapting the distance belief function estimation model. *In Proceedings of International Conference on Communications, Computing and Control Applications, CCCA'2011, Hammamet, Tunisia*, pages 1-6, 2011.

Appendix E

Research funding

During the 2017-2023 period, I was involved as scientific lead partner or coordinator in several research projects. In addition to the projects I contributed to, in this section I detail the projects I managed since my arrival at INSA Strasbourg which weigh **€2 261k** of net funding.

E.1 Laboratory and school internal projects

EXERCISE (2019-2021): The EXERCISE project aims to exploit data from national, regional or departmental observatories for decision-making purposes. We propose to analyze data from the "SISPEA" information system for water and sanitation utilities. Data mining and missing data processing approaches are applied to extract correlations between service information and performance indicators.

Role : Project coordinator

Funding : €10k

Funding organization : ICube.

Guillaume Guarino PhD funding (2019-2022): The purpose of this PhD, submitted under the supervision of Denis Cavallucci, was to mine contradiction using Large Language Model and semi-supervised approach. The challenge was to use Generative Adversarial network to find contradiction under the constraint of having a low number of labelled patents

Role : PhD supervisor and project co-writing

Funding : €110k

Funding organization : INSA Strasbourg.

Funded PhD student : Guillaume Guarino.

Yusif Imamverdiyev PhD funding (2023-2026): This PhD is funded by the "college doctoral" and under the supervision of Professor Christophe

Lallemand. This thesis aims to create a lithium-ion battery model that combines the accuracy of multi-physics modeling with the fast computation time and minimal number of parameters of data-driven models. This multimodal approach enables accurate prediction of SoX and RUL, and reduces the need for large amounts of experimental data for model training and aging studies.

Role : PhD supervisor and project co-writing

Funding : €115k

Funding organization : College doctoral Strasbourg.

Funded PhD student : Yusif Imamverdiyev.

E.2 Industrial projects

XAI-BATMAN (2020-2022): This project was funded by the company MOBION. MOBION is an electric scooter manufacturer. The project aims to understand the battery ageing using explainable AI. The scientific challenge was to extend current semantic models to integrate battery ageing concepts.

Role : Project coordinator

Funding : €220k

Funding organization : MOBION company through SATT connectus.

Funded Postdoc : Franco Giustozzi.

E.3 Regional projects

Pacte de compétence (2021-2023): This regional project aimed to fund new equipment for the INSA Strasbourg school. The purpose was to fund 35 computers with GPU. Two main class computers were updated for the sake of new deep learning lectures.

Role : Project contributor in the preparatory department

Funding : €134k

Funding organization : Region Grand Est.

XPLANE-Region (2022-2025): This project is funded by the region Grand Est and the ADEME. The project aims to reach an interpretable AI for battery ageing. We aim to combine data mining techniques in addition to the explainable model.

Role : Project coordinator

Funding : €55k

Funding organization : Region Grand Est.

Funded PhD student : Theo Heitzmann.

SIMILAR (2023-2025): This project is funded by the region Grand Est. This funding aims to extend by 8 months a postdoc position funded by XQUALITY project. The project intends to develop a consistency approach for explainable AI. The main challenge is to build a set of consistent explanations for a quality assurance application.

Role : Project coordinator

Funding : €25k

Funding organization : Region Grand Est.

Funded postdoc : Amel Hidouri.

E.4 National projects

XAIHyBATMAN (2021-2022): This project was funded by the CNRS through its AMORCE program in order to support the Horizon Europe project submission that led to the acceptance of the ENERGETIC project.

Role : Project coordinator

Funding : €15k

Funding organization : National Scientific Research Center (CNRS)

XPLANE (2022-2025): This project is funded by the region Grand Est and the ADEME. The project aims to reach an interpretable AI for battery ageing. We aim to combine data mining techniques in addition to the explainable model.

Role : Project coordinator

Funding : €55k

Funding organization : The French Agency for Ecological Transition (ADEME)

Funded PhD student : Théo Heitzmann.

XQUALITY (2023-2026): The XQuality project is researching hybrid and explainable AI approaches to help manufacturing companies implement intelligent and automated quality assurance. The project combines data-based machine learning, semantic technologies and expert knowledge to monitor and explain product and process quality targets in a company. The goal is to develop an AI-based system that will assist the staff in identifying the main causes of quality issues as early as possible, to achieve reliability engineering in the domain of manufacturing, by means of new quality assurance models.

Role : Partner scientific leader

Funding : €176k

Funding organization : National Research Agency (ANR)

Funded postdoc : Amel Hidouri.

Funded PhD student : Slimane Arbaoui.

E.5 European projects

HALFBACK (2017-2020): The HALFBACK project aimed to develop a digital tool for anticipating production equipment deficiencies, optimizing maintenance planning and proposing available outsourcing solutions. INSA Strasbourg's work falls within the field of predictive maintenance. The aim is to develop new algorithms for real-time machine diagnostics. To date, two research themes have been addressed: (i) Sequential pattern mining for fault origin determination; (ii) Predicting the date of machine failure.

Role : Partner scientific leader

funding : €300k

Funding organization : INTERREG V Rhin Supérieur.

Funded PhD student : Quishi Cao, Ines Jorge.

DA-HPC-OR (2018-2020): The DA-HPC-OR project involved the analysis of data from the consortium's high-performance computing (HPC) systems: NEMO at the University of Fribourg (NEMO-UniFR), sciCORE at the University of Basel (sciCORE-UniBas) and HPC at the University of Strasbourg (HPC-UniStra). The objectives of the project are to analyze the data collected at NEMO-UniFR in order to improve their research and operational activities, and to offer monitoring perspectives. The proposed approach includes: system and application monitoring; legal compliance through de-identification and anonymization; and data analysis. The methods used are as follows: HPC monitoring, legal control, de-identification, anonymization, data aggregation, data mining and information extraction.

Role : project co-contributor

Funding : €10k

Funding organization : EUCOR.

Fessenheim (2020-2022): The project aims to find alternative energy sources after the Fessenheim nuclear power plant shutdown. Among the

selected solutions are the use of battery in stationary application and the use of AI for smart energy management.

Role : project co-contributor

Funding : €10k

Funding organization : EUCOR.

ENERGETIC (2023-2026): ENERGETIC is a Horizon Europe project that aims at developing a next generation BMS for optimizing batteries' systems utilisation in the first and the second life on a path towards the reduction of the Total Cost of Ownership. ENERGETIC's vision is not just predicting and monitoring the remaining useful life of a Li-ion battery with digital twin but also inspecting the reasons for degradation by investigating the model explainability.

Role : project co-coordinator (with Tedjani Mesbahi).

Funding : €1M

Funding organization : Horizon Europe.

Funded postdoc : Lakhdar Mamouri.

Funded PhD student : Slimane Arbaoui, Marwa Zitouni, Cyrine Soufi.

E.6 Statistics

Since joining INSA Strasbourg and ICube laboratory in 2017, I have had €8006k of total project funding of which €2261k is the net amount. The projects are local (funded by INSA Strasbourg or ICube), regional (mainly Région Grand Est), Industrial (PRT, PFE and company research project), national and European. Over the six years spent at INSA Strasbourg and ICube laboratory I brought an average of €372k of funding per year.

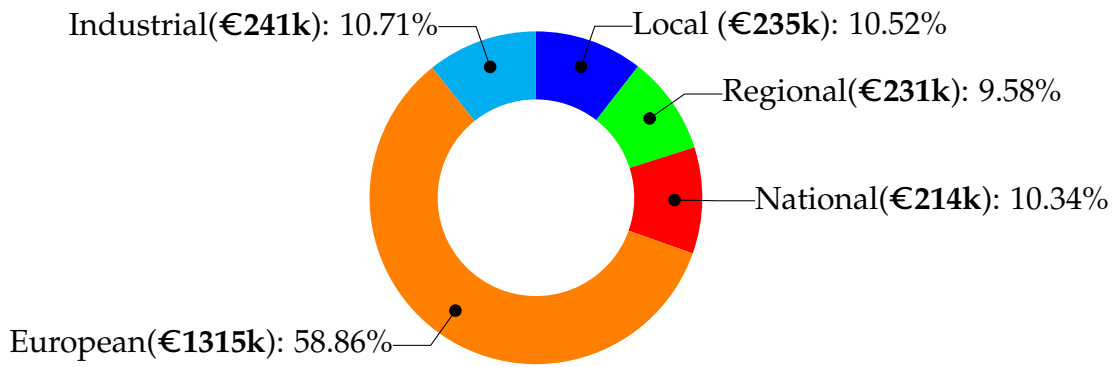


FIGURE E.1: Breakdown of funding sources

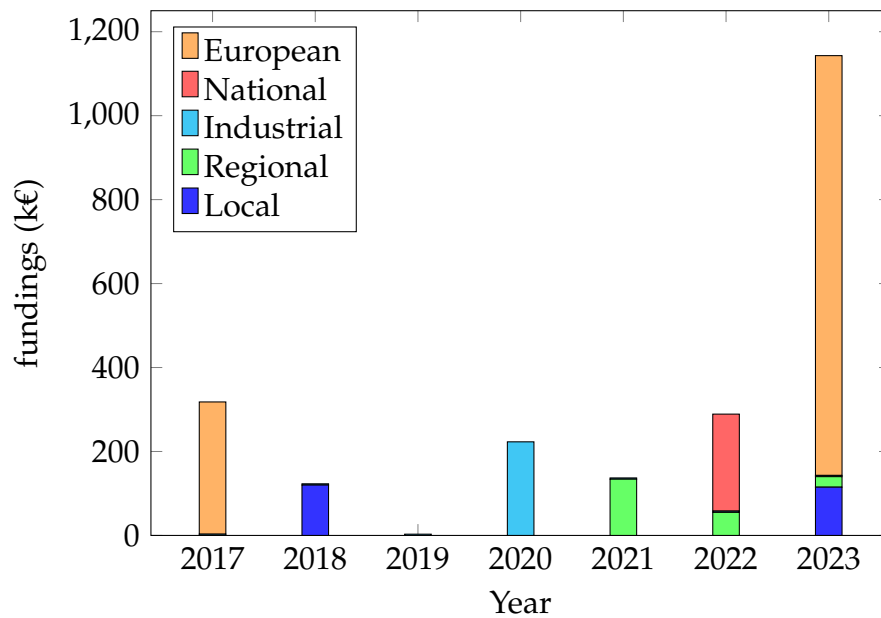


FIGURE E.2: Amount of net INSA Strasbourg funding obtained per year

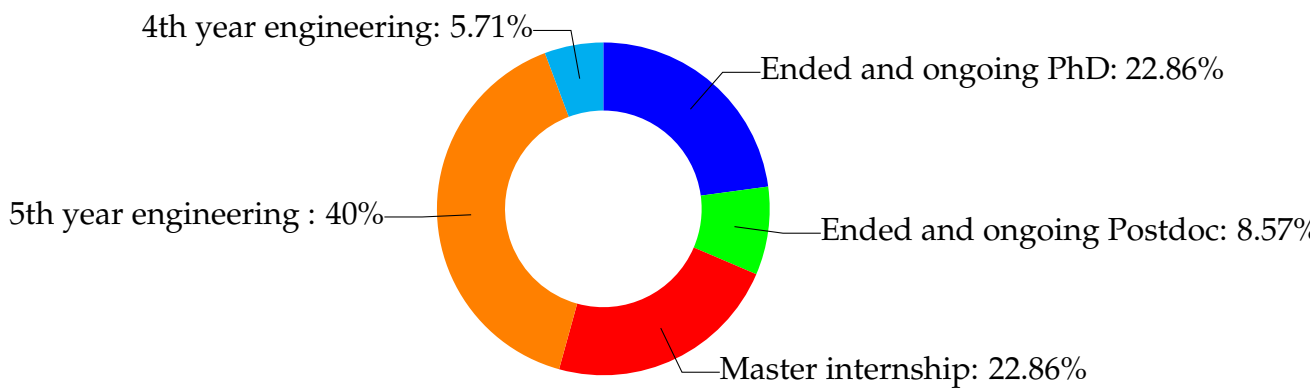


FIGURE E.3: Breakdown of research supervision

Bibliography

- Agrawal, Rakesh, Tomasz Imieliński, and Arun Swami (1993). “Mining association rules between sets of items in large databases”. In: *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pp. 207–216.
- Agrawal, Rakesh and Ramakrishnan Srikant (1994). “Fast algorithms for mining association rules”. In: *Proc. 20th int. conf. very large data bases, VLDB*. Vol. 1215. Santiago, Chile, pp. 487–499.
- Altshuller, Genrikh Saulovich (1999). *The innovation algorithm: TRIZ, systematic innovation and technical creativity*. Technical innovation center, Inc.
- Altshuller, Guenrich (1996). *And Suddenly the Inventor Appeared : TRIZ, the Theory of Inventive Problem Solving*. en. 1st. Technical Innovation Center, Inc. ISBN: 2-9521394-1-5.
- Álvarez, Miguel R., Paulo Félix, and Purificación Cariñena (2013). “Discovering Metric Temporal Constraint Networks on Temporal Databases”. In: *Artif. Intell. Med.* 58.3, 139–154. ISSN: 0933-3657. DOI: [10.1016/j.artmed.2013.03.006](https://doi.org/10.1016/j.artmed.2013.03.006). URL: <https://doi.org/10.1016/j.artmed.2013.03.006>.
- Antonio, Gomariz et al. (2013). “ClaSP: An Efficient Algorithm for Mining Frequent Closed Sequences”. In: *Proceedings of the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Gold Coast, Australia, April 14-17*, pp. 50–61.
- Armstrong, J Scott and Fred Collopy (1992). “Error measures for generalizing about forecasting methods: Empirical comparisons”. In: *International journal of forecasting* 8.1, pp. 69–80.
- Atzmueller, Martin (2015). “Subgroup discovery”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 5.1, pp. 35–49.
- Audin, Paul et al. (2021). “Auto-encoder LSTM for Li-ion SOH prediction: a comparative study on various benchmark datasets”. In: *20th IEEE International Conference on Machine Learning and Applications, ICMLA 2021, Pasadena, CA, USA, December 13-16, 2021*. Ed. by M. Arif Wani et al. IEEE, pp. 1529–1536. DOI: [10.1109/ICMLA52953.2021.00246](https://doi.org/10.1109/ICMLA52953.2021.00246). URL: <https://doi.org/10.1109/ICMLA52953.2021.00246>.

- Barandas, Marília et al. (2020). "TSFEL: Time Series Feature Extraction Library". In: *SoftwareX* 11, p. 100456. ISSN: 23527110. DOI: [10.1016/j.softx.2020.100456](https://doi.org/10.1016/j.softx.2020.100456).
- Beal, Matthew J, Zoubin Ghahramani, and Carl E Rasmussen (2002). "The infinite hidden Markov model". In: *Advances in neural information processing systems*, pp. 577–584.
- Ben Chrayet, Achraf, Ahmed Samet, and Mohamed Anis Bach Tobji (2020). "Evidence Theory Based Combination of Frequent Chronicles for Failure Prediction". In: *Scalable Uncertainty Management - 14th International Conference, SUM 2020, Bozen-Bolzano, Italy, September 23-25, 2020, Proceedings*. Ed. by Jesse Davis and Karim Tabia. Vol. 12322. Lecture Notes in Computer Science. Springer, pp. 226–233. DOI: [10.1007/978-3-030-58449-8_16](https://doi.org/10.1007/978-3-030-58449-8_16). URL: https://doi.org/10.1007/978-3-030-58449-8_16.
- Bengio, Yoshua et al. (2003). "A neural probabilistic language model". In: *The journal of machine learning research* 3, pp. 1137–1155.
- Bennett, Kristin P and Ayhan Demiriz (1999). "Semi-supervised support vector machines". In: *Advances in Neural Information processing systems*, pp. 368–374.
- Berdyugina, Daria and Denis Cavallucci (Apr. 2022a). "Automatic extraction of inventive information out of patent texts in support of manufacturing design studies using Natural Languages Processing". In: *Journal of Intelligent Manufacturing*, pp. 1–15. DOI: [10.1007/s10845-022-01943-y](https://doi.org/10.1007/s10845-022-01943-y).
- (Dec. 2022b). "Exploitation of causal relation for automatic extraction of contradiction from a domain-restricted patent corpus". In: *Proceedings of TRIZ Future 2022*. Brussels, Belgium: ETRIA.
- Bickel, Steffen and Tobias Scheffer (2004). "Multi-view clustering." In: *ICDM*. Vol. 4, pp. 19–26.
- Bojanowski, Piotr et al. (2017). "Enriching word vectors with subword information". In: *Transactions of the Association for Computational Linguistics* 5, pp. 135–146.
- Brown, Peter F et al. (1992). "Class-based n-gram models of natural language". In: *Computational linguistics* 18.4, pp. 467–480.
- Bryant, Randal E (1986). "Graph-based algorithms for boolean function manipulation". In: *Computers, IEEE Transactions on* 100.8, pp. 677–691.
- Cao, Qiushi et al. (2020). "Combining chronicle mining and semantics for predictive maintenance in manufacturing processes". In: *Semantic Web* 11.6, pp. 927–948. DOI: [10.3233/SW-200406](https://doi.org/10.3233/SW-200406). URL: <https://doi.org/10.3233/SW-200406>.

- Cascini, Gaetano and Davide Russo (Jan. 2007). "Computer-aided analysis of patents and search for TRIZ contradictions". In: *International Journal of Product Development* 4, pp. 52–67. DOI: [10.1504/IJPD.2007.011533](https://doi.org/10.1504/IJPD.2007.011533).
- Cavallucci, Denis, Francois Rousselot, and Cecilia Zanni (2011). "Using patents to populate an inventive design ontology". In: *Procedia Engineering* 9, pp. 52–62.
- Celaya, Jose et al. (2011). "Prognostics of power MOSFETs under thermal stress accelerated aging using data-driven and model-based methodologies". In.
- Chang, Hsiang-Tang, Chen-Yen Chang, and Wen-Kuei Wu (May 2017). "Computerized innovation inspired by existing patents". In: pp. 1134–1137. DOI: [10.1109/ICASI.2017.7988268](https://doi.org/10.1109/ICASI.2017.7988268).
- Chen, Huimin et al. (2021). "From Symbols to Embeddings: A Tale of Two Representations in Computational Social Science". In: *Journal of Social Computing* 2.2, pp. 103–156.
- Cheng, Shunfeng and Michael Pecht (2009). "A fusion prognostics method for remaining useful life prediction of electronic products". In: *2009 IEEE International Conference on Automation Science and Engineering*. IEEE, pp. 102–107.
- Chu, Xiao et al. (2016). "CRF-CNN: Modeling Structured Information in Human Pose Estimation". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2016/file/6f3ef77ac0e3619e98159e9b6febf557-Paper.pdf>.
- Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". English (US). In: *NIPS 2014 Workshop on Deep Learning, December 2014*.
- Collobert, Ronan and Jason Weston (2008). "A unified architecture for natural language processing: Deep neural networks with multitask learning". In: *Proceedings of the 25th international conference on Machine learning*, pp. 160–167.
- Collobert, Ronan et al. (2011). "Natural language processing (almost) from scratch". In: *Journal of Machine Learning Research* 12, Aug, pp. 2493–2537.
- Cram, Damien, Benoît Mathern, and Alain Mille (2012). "A complete chronicle discovery approach: application to activity analysis". In: *Expert Systems* 29.4, pp. 321–346.

- Cremers, Katrin et al. (Aug. 1999). "Citation Frequency And The Value Of Patented Inventions". In: *The Review of Economics and Statistics* 81, pp. 511–515. DOI: [10.1162/003465399558265](https://doi.org/10.1162/003465399558265).
- Croce, Danilo, Giuseppe Castellucci, and Roberto Basili (July 2020). "GAN-BERT: Generative Adversarial Learning for Robust Text Classification with a Bunch of Labeled Examples". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, pp. 2114–2119. DOI: [10.18653/v1/2020.acl-main.191](https://doi.org/10.18653/v1/2020.acl-main.191). URL: <https://www.aclweb.org/anthology/2020.acl-main.191>.
- Dauxais, Yann et al. (Sept. 2015). "Chronicles mining in a database of drugs exposures". In: *ECML Doctoral consortium*. Proceedings of the ECML Doctoral consortium. Porto, Portugal. URL: <https://hal.inria.fr/hal-01184100>.
- Davis, Jesse and Mark Goadrich (2006). "The Relationship Between Precision-Recall and ROC Curves". In: *Proceedings of the 23rd International Conference on Machine Learning*. ICML '06. Pittsburgh, Pennsylvania, USA: ACM, pp. 233–240. ISBN: 1-59593-383-2. DOI: [10.1145/1143844.1143874](https://doi.org/10.1145/1143844.1143874). URL: <http://doi.acm.org/10.1145/1143844.1143874>.
- Dempster, Arthur P (1968). "A generalization of Bayesian inference". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 30.2, pp. 205–232.
- (2008). "Upper and lower probabilities induced by a multivalued mapping". In: *Classic works of the Dempster-Shafer theory of belief functions*. Springer, pp. 57–72.
- Deng, Li and Dong Yu (2013). "Deep learning for signal and information processing". In: *Microsoft research monograph*.
- Devlin, Jacob et al. (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805*.
- (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *NAACL-HLT*.
- Ding, Zhikun et al. (2017). "A new TRIZ-based patent knowledge management system for construction technology innovation". In: *Journal of Engineering, Design and Technology*.
- Domb, Ellen (1997). "Contradictions: air bag applications". In: *TRIZ Journal* 2.7.
- Dos Reis, Gonçalo et al. (2021). "Lithium-ion battery data and where to find it". In: *Energy and AI* 5, p. 100081.

- Dousson, Christophe and Thang Vu Duong (1999). "Discovering Chronicles with Numerical Time Constraints from Alarm Logs for Monitoring Dynamic Systems". In: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. IJCAI '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 620–626. ISBN: 1-55860-613-0. URL: <http://dl.acm.org/citation.cfm?id=646307.687741>.
- Dragomir, Otilia Elena et al. (2009). "Review of prognostic problem in condition-based maintenance". In: *2009 European Control Conference (ECC)*. IEEE, pp. 1587–1592.
- Edunov, Sergey, Alexei Baevski, and Michael Auli (2019). "Pre-trained language model representations for language generation". In: *arXiv preprint arXiv:1903.09722*.
- Efthymiou, K et al. (2012). "On a predictive maintenance platform for production systems". In: *Procedia CIRP* 3, pp. 221–226.
- Fan, Yujie, Yanfang Ye, and Lifei Chen (2016). "Malicious sequential pattern mining for automatic malware detection". In: *Expert Systems with Applications* 52, pp. 16–25.
- Fournier-Viger, Philippe et al. (2017). "A survey of sequential pattern mining". In: *Data Science and Pattern Recognition* 1.1, pp. 54–77.
- Gökalp, Ebru, Umut Şener, and P Erhan Eren (2017). "Development of an assessment model for industry 4.0: industry 4.0-MM". In: *International Conference on Software Process Improvement and Capability Determination*. Springer, pp. 128–142.
- Goldberg, Yoav (2017). "Neural network methods for natural language processing". In: *Synthesis Lectures on Human Language Technologies* 10.1, pp. 1–309.
- Goodfellow, Ian J. et al. (2014). "Generative Adversarial Nets". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS'14. Montreal, Canada: MIT Press, 2672–2680.
- Grall, Antoine et al. (2002). "Continuous-time predictive-maintenance scheduling for a deteriorating system". In: *IEEE transactions on reliability* 51.2, pp. 141–150.
- Greene, Randal et al. (2011). "GIS-based multiple-criteria decision analysis". In: *Geography compass* 5.6, pp. 412–432.
- Gu, Jiuxiang et al. (2018). "Recent advances in convolutional neural networks". In: *Pattern recognition* 77, pp. 354–377.

- Guarino, Guillaume et al. (2020). "SummaTRIZ: Summarization Networks for Mining Patent Contradiction". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 979–986.
- Gustafsson, Fredrik et al. (2002). "Particle filters for positioning, navigation, and tracking". In: *IEEE Transactions on signal processing* 50.2, pp. 425–437.
- Guyon, Isabelle and André Elisseeff (2003a). "An introduction to variable and feature selection". In: *Journal of machine learning research* 3.Mar, pp. 1157–1182.
- Guyon, Isabelle and André Elisseeff (Jan. 2003b). "An Introduction of Variable and Feature Selection". In: *J. Machine Learning Research Special Issue on Variable and Feature Selection* 3, pp. 1157 –1182. DOI: [10 . 1162 / 153244303322753616](https://doi.org/10.1162/153244303322753616).
- Hamilton, James Douglas (1994). *Time series analysis*. Vol. 2. Princeton university press Princeton, NJ.
- Hannan, Mahammad Abdul et al. (2021). "SOC Estimation of Li-ion Batteries With Learning Rate-Optimized Deep Fully Convolutional Network". In: *IEEE Transactions on Power Electronics* 36, pp. 7349–7353.
- Harris, Zellig S (1954). "Distributional structure". In: *Word* 10.2-3, pp. 146–162.
- Hasib, Shahid A et al. (2021). "A comprehensive review of available battery datasets, RUL prediction approaches, and advanced battery management". In: *Ieee Access* 9, pp. 86166–86193.
- He, Cong and Han Tong Loh (2008). "Grouping of TRIZ Inventive Principles to facilitate automatic patent classification". In: *Expert Syst. Appl.* 34, pp. 788–795.
- Heng, Aiwina et al. (2009). "Rotating machinery prognostics: State of the art, challenges and opportunities". In: *Mechanical systems and signal processing* 23.3, pp. 724–739.
- Hermann, Karl Moritz et al. (2015). "Teaching Machines to Read and Comprehend". In: *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*. NIPS'15. Montreal, Canada: MIT Press, 1693–1701.
- Hinton, Geoffrey E et al. (1986). "Learning distributed representations of concepts". In: *Proceedings of the eighth annual conference of the cognitive science society*. Vol. 1. Amherst, MA, p. 12.
- Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, 1735–1780. ISSN: 0899-7667. DOI: [10 .](https://doi.org/10.1162/08997669756868688)

- 1162/neco.1997.9.8.1735. URL: <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Hong, Seonri et al. (2020). "State of Health Estimation for Lithium-Ion Batteries Using Long-Term Recurrent Convolutional Network". In: *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, pp. 1929–1934. DOI: [10.1109/IECON43393.2020.9254275](https://doi.org/10.1109/IECON43393.2020.9254275).
- Hong, Sheng et al. (2014). "An adaptive method for health trend prediction of rotating bearings". In: *Digital Signal Processing* 35, pp. 117–123.
- Houssin, Remy et al. (2015). "TRIZ theory and case based reasoning: synergies and oppositions". In: *International Journal on Interactive Design and Manufacturing (IJIDeM)* 9.3, pp. 177–183.
- Huang, Eric H et al. (2012). "Improving word representations via global context and multiple word prototypes". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 873–882.
- Huang, Zhengxing, Xudong Lu, and Huilong Duan (2012). "On mining clinical pathway patterns from medical behaviors". In: *Artificial intelligence in medicine* 56.1, pp. 35–50.
- Hyndman, Rob J. and Anne B. Koehler (2006). "Another look at measures of forecast accuracy". In: *International Journal of Forecasting* 22.4, pp. 679–688. ISSN: 01692070. DOI: [10.1016/j.ijforecast.2006.03.001](https://doi.org/10.1016/j.ijforecast.2006.03.001).
- Jabbour, Said et al. (2016). "A MIS partition based framework for measuring inconsistency". In: *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*.
- Javed, Kamran, Rafael Gouriveau, and Nouredine Zerhouni (2017). "State of the art and taxonomy of prognostics approaches, trends of prognostics applications and open issues towards maturity at different technology readiness levels". In: *Mechanical Systems and Signal Processing* 94, pp. 214–236.
- Javed, Kamran et al. (2011). "Improving data-driven prognostics by assessing predictability of features." In.
- Jorge, Inès et al. (2020). "New ANN results on a major benchmark for the prediction of RUL of Lithium Ion batteries in electric vehicles". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1246–1253.
- Jugulum, Rajesh and Daniel D. Frey (2007). "Toward a taxonomy of concept designs for improved robustness". In: *Journal of Engineering Design* 18.2, pp. 139–156. DOI: [10.1080/09544820600731496](https://doi.org/10.1080/09544820600731496).

- Kallianpur, Gopinath (2013). *Stochastic filtering theory*. Vol. 13. Springer Science & Business Media.
- Kingma, Durk P et al. (2014). "Semi-supervised learning with deep generative models". In: *Advances in neural information processing systems*, pp. 3581–3589.
- Kipf, Thomas N. and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks". In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. ICLR '17. Palais des Congrès Neptune, Toulon, France. URL: <https://openreview.net/forum?id=SJU4ayYgl>.
- Kleinbaum, David G et al. (2002). *Logistic regression*. Springer.
- Koehn, Philipp (2017). "Neural machine translation". In: *arXiv preprint arXiv:1709.07809*.
- Kohavi, Ron and George H. John (2013). "Wrappers for feature subset selection". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7920 LNCS.97, pp. 654–678. ISSN: 16113349.
- Kohonen, Teuvo (1997). "Exploration of very large databases by self-organizing maps". In: *Proceedings of International Conference on Neural Networks (ICNN'97)*. Vol. 1. IEEE, PL1–PL6.
- Korobkin, D. M., S. A. Fomenkov, and A. G. Kravets (2017). "Extraction of physical effects practical applications from patent database". In: *2017 8th International Conference on Information, Intelligence, Systems Applications (IISA)*, pp. 1–5. DOI: [10.1109/IISA.2017.8316402](https://doi.org/10.1109/IISA.2017.8316402).
- Krenek, Jiri et al. (2016). "Application of artificial neural networks in condition based predictive maintenance". In: *Recent developments in intelligent information and database systems*, pp. 75–86.
- Lafferty, John D., Andrew McCallum, and Fernando C. N. Pereira (2001). "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 282–289. ISBN: 1-55860-778-1. URL: <http://dl.acm.org/citation.cfm?id=645530.655813>.
- Lan, Zhenzhong et al. (2019). *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. arXiv: [1909.11942](https://arxiv.org/abs/1909.11942) [cs.CL].
- Lanjouw, Jean and Mark Schankerman (Feb. 2004). "Patent Quality and Research Productivity: Measuring Innovation with Multiple Indicators". In:

- Economic Journal* 114, pp. 441–465. DOI: [10.1111/j.1468-0297.2004.00216.x](https://doi.org/10.1111/j.1468-0297.2004.00216.x).
- Lecun, Y. et al. (1998). “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11, pp. 2278–2324. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- Li, Pengfei et al. (2021). “ACT: an Attentive Convolutional Transformer for Efficient Text Classification”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.15, pp. 13261–13269. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/17566>.
- Li, Xiaoyu et al. (2019). “Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks”. In: *Journal of Energy Storage* 21, pp. 510–518.
- Li, Zhen et al. (Oct. 2012). “A Framework for Automatic TRIZ Level of Invention Estimation of Patents Using Natural Language Processing, Knowledge-Transfer and Patent Citation Metrics”. In: *Comput. Aided Des.* 44.10, 987–1010. ISSN: 0010-4485. DOI: [10.1016/j.cad.2011.12.006](https://doi.org/10.1016/j.cad.2011.12.006). URL: <https://doi.org/10.1016/j.cad.2011.12.006>.
- Liang, Y. et al. (2009). “Computer-aided classification of patents oriented to TRIZ”. In: *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 2389–2393.
- Liang, Yanhong and Runhua Tan (2007). “A Text-Mining-based Patent Analysis in Product Innovative Process”. In: *Trends in Computer Aided Innovation*. Ed. by Noel León-Rovira. Boston, MA: Springer US, pp. 89–96. ISBN: 978-0-387-75456-7.
- Liaw, Andy, Matthew Wiener, et al. (2002). “Classification and regression by randomForest”. In: *R news* 2.3, pp. 18–22.
- Likas, Aristidis, Nikos Vlassis, and Jakob J Verbeek (2003). “The global k-means clustering algorithm”. In: *Pattern recognition* 36.2, pp. 451–461.
- Lipton, Zachary C. (2016). *The Mythos of Model Interpretability*.
- Lipu, MS Hossain et al. (2018). “A review of state of health and remaining useful life estimation methods for lithium-ion battery in electric vehicles: Challenges and recommendations”. In: *Journal of cleaner production* 205, pp. 115–133.
- Liu, Datong et al. (2012). “Data-driven prognostics for lithium-ion battery based on Gaussian process regression”. In: *Proceedings of IEEE 2012 Prognostics and System Health Management Conference, PHM-2012*, pp. 1–5. DOI: [10.1109/PHM.2012.6228848](https://doi.org/10.1109/PHM.2012.6228848).

- Liu, Kailong et al. (2021). "A Data-Driven Approach with Uncertainty Quantification for Predicting Future Capacities and Remaining Useful Life of Lithium-ion Battery". In: *IEEE Transactions on Industrial Electronics* 68.4, pp. 3170–3180. ISSN: 15579948. DOI: [10.1109/TIE.2020.2973876](https://doi.org/10.1109/TIE.2020.2973876).
- Liu, Zhiyuan, Yankai Lin, and Maosong Sun (2020). *Representation learning for natural language processing*. Springer Nature.
- Loh, Han, Cong He, and Lixiang Shen (Mar. 2006). "Automatic classification of patent documents for TRIZ users". In: *World Patent Information* 28, pp. 6–13. DOI: [10.1016/j.wpi.2005.07.007](https://doi.org/10.1016/j.wpi.2005.07.007).
- Lundberg, Scott M. and Su-In Lee (2017). "A Unified Approach to Interpreting Model Predictions". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*. Long Beach, California, USA: Curran Associates Inc., 4768–4777. ISBN: 9781510860964.
- Majumder, P, M Mitra, and BB Chaudhuri (2002). "N-gram: a language independent approach to IR and NLP". In: *International conference on universal knowledge and language*.
- Mannila, Heikki, Hannu Toivonen, and A. Inkeri Verkamo (1997). "Discovery of frequent episodes in event sequences". In: *Data Mining and Knowledge Discovery* 1, 259–289.
- McCann, M and A Johnston (2008). *Secom Dataset, UCI Machine Learning Repository*.
- Mesbahi, Tedjani et al. (2017). "Dynamic model of li-ion batteries incorporating electrothermal and ageing aspects for electric vehicle applications". In: *IEEE Transactions on industrial electronics* 65.2, pp. 1298–1305.
- Mikolov, Tomas, Quoc V Le, and Ilya Sutskever (2013). "Exploiting similarities among languages for machine translation". In: *arXiv preprint arXiv:13-09.4168*.
- Mikolov, Tomáš et al. (2010). "Recurrent neural network based language model". In: *Eleventh Annual Conference of the International Speech Communication Association*.
- Mikolov, Tomáš et al. (2012). "Statistical language models based on neural networks". In: *Presentation at Google, Mountain View, 2nd April* 80, p. 26.
- Mikolov, Tomas et al. (2013a). "Distributed representations of words and phrases and their compositionality". In: *arXiv preprint arXiv:1310.4546*.
- Mikolov, Tomas et al. (2013b). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

- Mueller, Jonas and Aditya Thyagarajan (2016). "Siamese recurrent architectures for learning sentence similarity". In: *Thirtieth AAAI Conference on Artificial Intelligence*.
- Negny, Stéphane et al. (2012). "Toward an eco-innovative method based on a better use of resources: application to chemical process preliminary design". In: *Journal of Cleaner Production* 32, pp. 101–113.
- Ni, Xin, Ahmed Samet, and Denis Cavallucci (2019). "An Approach Merging the IDM-Related Knowledge". In: *International TRIZ Future Conference*. Springer, pp. 147–158.
- (2021). "Similarity-based approach for inventive design solutions assistance". In: *Journal of Intelligent Manufacturing*, pp. 1–18.
- (2022). "Similarity-based approach for inventive design solutions assistance". In: *J. Intell. Manuf.* 33, pp. 1681–1698.
- Ni, Xin et al. (2021). "PatRIS: Patent Ranking Inventive Solutions". In: *Database and Expert Systems Applications*. Ed. by Christine Strauss et al. Cham: Springer International Publishing, pp. 295–309. ISBN: 978-3-030-86475-0.
- Novakovic, Jasmina (Dec. 2000). "Using Information Gain Attribute Evaluation to Classify Sonar Targets". In.
- Otter, Daniel W, Julian R Medina, and Jugal K Kalita (2020). "A survey of the usages of deep learning for natural language processing". In: *IEEE Transactions on Neural Networks and Learning Systems*.
- Paris, Paul and Fazil Erdogan (1963). "A critical analysis of crack propagation laws". In.
- Pecht, Michael (2009). "Prognostics and health management of electronics". In: *Encyclopedia of Structural Health Monitoring*.
- Pecht, Michael G (2010). "A prognostics and health management roadmap for information and electronics-rich systems". In: *IEICE ESS Fundamentals Review* 3.4, 4_25–4_32.
- Pei, Jian, Jiawei Han, and Wei Wang (2002). "Mining sequential patterns with constraints in large databases". In: *Proceedings of the eleventh international conference on Information and knowledge management*, pp. 18–25.
- Pei, Jian et al. (2001). "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth". In: pp. 215–224.
- Peng, Ying, Ming Dong, and Ming Jian Zuo (2010). "Current status of machine prognostics in condition-based maintenance: a review". In: *The International Journal of Advanced Manufacturing Technology* 50.1-4, pp. 297–313.

- Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.
- Preger, Yuliya et al. (2020). "Degradation of commercial lithium-ion cells as a function of chemistry and cycling conditions". In: *Journal of The Electrochemical Society* 167.12, p. 120532.
- Qu, Jiantao et al. (2019a). "A neural-network-based method for RUL prediction and SOH monitoring of lithium-ion battery". In: *IEEE access* 7, pp. 87178–87191.
- (2019b). "A Neural-Network-Based Method for RUL Prediction and SOH Monitoring of Lithium-Ion Battery". In: *IEEE Access* 7, pp. 87178–87191. ISSN: 21693536. DOI: [10.1109/ACCESS.2019.2925468](https://doi.org/10.1109/ACCESS.2019.2925468).
- Quinlan, J. Ross (1986). "Induction of decision trees". In: *Machine learning* 1.1, pp. 81–106.
- Rabatel, Julien, Sandra Bringay, and Pascal Poncelet (2011). "Anomaly detection in monitoring sensor data for preventive maintenance". In: *Expert Systems with Applications* 38.6, pp. 7003–7015.
- Rahim, Zulhasni Abdul et al. (2018). "The application of computational thinking and TRIZ methodology in patent innovation analytics". In: *International Conference of Reliable Information and Communication Technology*. Springer, pp. 793–802.
- Rao, BKN (1996). *Handbook of condition monitoring*. Elsevier.
- Remil, Youcef et al. (2021). "Interpretable summaries of black box incident triaging with subgroup discovery". In: *2021 IEEE 8th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, pp. 1–10.
- Ren, Lei et al. (2018). "Remaining useful life prediction for lithium-ion battery: A deep learning approach". In: *Ieee Access* 6, pp. 50587–50598.
- Reynolds, Douglas (2015). "Gaussian mixture models". In: *Encyclopedia of biometrics*, pp. 827–832.
- Rish, Irina et al. (2001). "An empirical study of the naive Bayes classifier". In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22, pp. 41–46.
- Roemer, Michael J, EO Nwadiogbu, and G Bloor (2001). "Development of diagnostic and prognostic technologies for aerospace health management applications". In: *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*. Vol. 6. IEEE, pp. 3139–3147.

- Rosenberg, Chuck, Martial Hebert, and Henry Schneiderman (2005). "Semi-Supervised Self-Training of Object Detection Models." In: *WACV/MOTION 2*.
- Rousselot, François, Cecilia Zanni-Merk, and Denis Cavallucci (2012). "Towards a formal definition of contradiction in inventive design". In: *Computers in Industry* 63.3, pp. 231–242.
- Saha, B. and K. Goebel (2007). "Battery data set". In: *NASA Ames Prognostics Data Repository*.
- Salimans, Tim et al. (2016). "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2016/file/8a3363abe792db2d8761d6403605aeb7-Paper.pdf>.
- Santos, Maribel Yasmina et al. (2017). "A big data analytics architecture for industry 4.0". In: *World Conference on Information Systems and Technologies*. Springer, pp. 175–184.
- Savransky, Semyon (2000). *Engineering of Creativity: Introduction to TRIZ Methodology of Inventive Problem Solving*. CRC Press. ISBN: 9781420038958. URL: <https://books.google.fr/books?id=a5LMBQAAQBAJ>.
- Saxena, Abhinav et al. (2012). "Designing data-driven battery prognostic approaches for variable loading profiles: Some lessons learned". In: *European conference of prognostics and health management society*, pp. 72–732.
- Scholkopf, Bernhard and Alexander J Smola (2001). *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.
- Sellami, Chayma, Ahmed Samet, and Mohamed Anis Bach Tobji (2018). "Frequent Chronicle Mining: Application on Predictive Maintenance". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1388–1393.
- Sellami, Chayma et al. (2019). "On mining frequent chronicles for machine failure prediction". In: *Journal of Intelligent Manufacturing*, pp. 1–17.
- Severson, Kristen A. and Peter M. Attia (2019). "Data-driven prediction of battery cycle life before capacity degradation". In: *Nature Energy* 4, 383–391.
- Shafer, G. (1976a). *A Mathematical Theory of Evidence*. Princeton University Press.
- Shafer, Glenn (1976b). *A mathematical theory of evidence*. Vol. 42. Princeton university press.
- Shrikumar, Avanti, Peyton Greenside, and Anshul Kundaje (2017). "Learning Important Features Through Propagating Activation Differences". In.

- Sikorska, JZ, Melinda Hodkiewicz, and Lin Ma (2011). "Prognostic modelling options for remaining useful life estimation by industry". In: *Mechanical systems and signal processing* 25.5, pp. 1803–1836.
- Smets, P. and R. Kennes (1994). "The Transferable Belief Model". In: *Artificial Intelligence* 66.2, pp. 191–234.
- Socher, Richard et al. (2011). "Semi-supervised recursive autoencoders for predicting sentiment distributions". In: *Proceedings of the 2011 conference on empirical methods in natural language processing*, pp. 151–161.
- Song, Yuchen et al. (2017). "Data-driven hybrid remaining useful life estimation approach for spacecraft lithium-ion battery". In: *Microelectronics Reliability* 75, pp. 142–153.
- Souili, Achille and Denis Cavallucci (July 2017). "Automated Extraction of Knowledge Useful to Populate Inventive Design Ontology from Patents". In: pp. 43–62. ISBN: 978-3-319-56592-7. DOI: [10.1007/978-3-319-56593-4_2](https://doi.org/10.1007/978-3-319-56593-4_2).
- Soumya George, K and Shibily Joseph (2014). "Text classification by augmenting bag of words (BOW) representation with co-occurrence feature". In: *IOSR J. Comput. Eng* 16.1, pp. 34–38.
- Srikant, Ramakrishnan and Rakesh Agrawal (1996). "Mining sequential patterns: Generalizations and performance improvements". In: *International conference on extending database technology*. Springer, pp. 1–17.
- Stone, Mervyn (1974). "Cross-validators choice and assessment of statistical predictions". In: *Journal of the royal statistical society. Series B (Methodological)*, pp. 111–147.
- Susto, Gian Antonio et al. (2014). "Machine learning for predictive maintenance: A multiple classifier approach". In: *IEEE Transactions on Industrial Informatics* 11.3, pp. 812–820.
- Tax, Niek et al. (2017). "Predictive business process monitoring with LSTM neural networks". In: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 477–492.
- Turian, Joseph, Lev Ratinov, and Yoshua Bengio (2010). "Word representations: a simple and general method for semi-supervised learning". In: *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 384–394.
- Turney, Peter D and Patrick Pantel (2010). "From frequency to meaning: Vector space models of semantics". In: *Journal of artificial intelligence research* 37, pp. 141–188.

- Uckun, Serdar, Kai Goebel, and Peter JF Lucas (2008). "Standardizing research methods for prognostics". In: *2008 International Conference on Prognostics and Health Management*. IEEE, pp. 1–10.
- Vaswani, Ashish et al. (2017a). "Attention is all you need". In: *arXiv preprint arXiv:1706.03762*.
- (2017b). "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon et al. Curran Associates, Inc., pp. 5998–6008. URL: <http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf>.
- Verma, Sahil, John Dickerson, and Keegan Hines (2020). "Counterfactual explanations for machine learning: A review". In: *arXiv preprint arXiv:2010.10596*.
- Virkar, Anil V (2011). "A model for degradation of electrochemical devices based on linear non-equilibrium thermodynamics and its application to lithium ion batteries". In: *Journal of Power Sources* 196.14, pp. 5970–5984.
- Wang, Gangfeng et al. (Mar. 2016). "Extraction of Principle Knowledge from Process Patents for Manufacturing Process Innovation". In: vol. 56, pp. 193–198. DOI: [10.1016/j.procir.2016.10.053](https://doi.org/10.1016/j.procir.2016.10.053).
- Wang, K (2016). "Intelligent predictive maintenance (IPdM) system–Industry 4.0 scenario". In: *WIT Transactions on Engineering Sciences* 113, pp. 259–268.
- Wang, Yujie et al. (2017). "Remaining useful life prediction of lithium-ion battery based on discrete wavelet transform". In: *Energy Procedia* 105, pp. 2053–2058.
- Weston, Jason, Frédéric Ratle, and Ronan Collobert (2008). "Deep Learning via Semi-Supervised Embedding". In: *Proceedings of the 25th International Conference on Machine Learning*. ICML '08. Helsinki, Finland: Association for Computing Machinery, 1168–1175. ISBN: 9781605582054. DOI: [10.1145/1390156.1390303](https://doi.org/10.1145/1390156.1390303). URL: <https://doi.org/10.1145/1390156.1390303>.
- Widodo, Achmad and Bo-Suk Yang (2007). "Support vector machine in machine condition monitoring and fault diagnosis". In: *Mechanical systems and signal processing* 21.6, pp. 2560–2574.
- Yan, Wei (2014). "(Meta) Knowledge modeling for inventive design". PhD thesis. Université de Strasbourg.
- Yan, Wei, Cecilia Zanni-Merk, and François Rousselot (2011). "An Application of Semantic Distance between Short Texts to Inventive Design." In: *KEOD*, pp. 261–266.

- Yan, Wei et al. (2013). "Ontology matching for facilitating inventive design based on semantic similarity and case-based reasoning". In: *International Journal of Knowledge-Based and Intelligent Engineering Systems* 17.3, pp. 243–256.
- Yan, Xifeng, Jiawei Han, and Ramin Afshar (2003). "CloSpan: Mining: Closed sequential patterns in large datasets". In: *Proceedings of the 2003 SIAM international conference on data mining*. SIAM, pp. 166–177.
- Yang, Zhilin, William W. Cohen, and Ruslan Salakhutdinov (2016). "Revisiting Semi-Supervised Learning with Graph Embeddings". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML'16. New York, NY, USA: JMLR.org, 40–48.
- Yang, Zhilin et al. (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems* 32. Ed. by H. Wallach et al. Curran Associates, Inc., pp. 5754–5764. URL: <http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf>.
- Yanhong Liang, Runhua Tan, and Jianhong Ma (2008). "Patent analysis with text mining for TRIZ". In: *2008 4th IEEE International Conference on Management of Innovation and Technology*, pp. 1147–1151.
- Yeh, Chin-Chia Michael, Nickolas Kavantzias, and Eamonn Keogh (2017). "Matrix profile VI: Meaningful multidimensional motif discovery". In: *2017 IEEE international conference on data mining (ICDM)*. IEEE, pp. 565–574.
- Yin, Junfu, Zhigang Zheng, and Longbing Cao (2012). "USpan: an efficient algorithm for mining high utility sequential patterns". In: *Proceedings of the ACM SIGKDD international conference on Knowledge discovery and data mining, Beijing, China, August 12 - 16*, pp. 660–668.
- Zhang, Daniel et al. (2021). "The ai index 2021 annual report". In: *arXiv preprint arXiv:2103.06312*.
- Zhou, Danhua et al. (2020). "State of Health Monitoring and Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Temporal Convolutional Network". In: *IEEE Access* 8, pp. 53307–53320. DOI: [10.1109/ACCESS.2020.2981261](https://doi.org/10.1109/ACCESS.2020.2981261).
- Zhou, Peng et al. (2016). "Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling." In: *COLING*. Ed. by Nicoletta Calzolari, Yuji Matsumoto, and Rashmi Prasad. ACL, pp. 3485–3495. ISBN: 978-4-87974-702-0. URL: <http://dblp.uni-trier.de/db/conf/coling/coling2016.html#ZhouQZXBX16>.

- Zhu, Xiaojin and Andrew B Goldberg (2009). "Introduction to semi-supervised learning". In: *Synthesis lectures on artificial intelligence and machine learning* 3.1, pp. 1–130.
- Zhuang, Liu et al. (Aug. 2021). "A Robustly Optimized BERT Pre-training Approach with Post-training". English. In: *Proceedings of the 20th Chinese National Conference on Computational Linguistics*. Huhhot, China: Chinese Information Processing Society of China, pp. 1218–1227. URL: <https://aclanthology.org/2021.ccl-1.108>.
- Zou, Will Y et al. (2013). "Bilingual word embeddings for phrase-based machine translation". In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1393–1398.