

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of 8 Mai 1945 Guelma



Faculty of Mathematics, Computer Science and Material Sciences
Department of Computer Science
Domiciliation laboratory of Information and Communication Sciences and
Technologies

Thesis of doctorate in

Field: Computer Science. **Stream:** Mathematics and Computer Science

Speciality: Artificial Intelligence

Presented by:
Chaima AFIFI

Deploying Artificial Intelligence techniques for Supporting Decisions in the Business Process Area

Defended on: July/08/2025

Before the jury composed of:

Full name	Rank	University	
Hamid SERIDI	Professor	Univ. of 8 May 1945, Guelma	President
Ali KHEBIZI	MCA	Univ. of 8 May 1945, Guelma	Supervisor
Khaled HALIMI	MCA	Univ. of 8 May 1945, Guelma	Co-supervisor
Brahim FAROU	Professor	Univ. of 8 May 1945, Guelma	Examiner
Hassina SERIDI	Professor	Univ. of Badji Mokhtar, Annaba	Examiner
Zine-Eddine KOUAHLA	Professor	Univ. of 8 May 1945, Guelma	Invited

Academic year: 2024-2025

Dedicace

*In the name of Allah, the Most Beneficent and the Most Merciful.
All praises are to Allah for all blesses so that we can accomplish
this thesis. In addition, may peace and salutation be given to the
prophet Muhammad who has taken all human being from the
darkness to the lightness.*

*Greetings and invocation are presented to my parents (dad and
mom), sisters and my husband (Haroun BRINIS) who give me the
force after Allah to accomplish this thesis.*

*Furthermore, i would like to express my genuine gratitude to my
supervisor Ali KHEBIZI, and co-supervisor Khaled HALIMI as well
as all member of the computer science department and LABSTIC
laboratory...*

Abstract

Business Processes (BP) constitute the heart of Information Systems (IS) of modern organizations. Thus, they are intensively utilized, both in the management of various companies' resources and in decision-making and strategic alignment activities. The abstract specifications (*or models*) expressing the business logic behind the BPs are essential conceptual tools useful for various tasks, varying from modeling, analysis, monitoring and maintenance. However, with the spectacular increase in the volume of data handled during the life cycle of BPs, which is often heterogeneous in nature, conventional approaches for modeling and mining BP models prove to be ineffective, hindering decision-making actions.

To overcome these limitations, in this thesis we leverage the latest advancements achieved in the AI area in order to improve decision support systems in the field of BPs management. The first contribution of this thesis consists of a conceptual framework, called DSS for BP (DSS4BP), which allows constructing a Knowledge Graph (KG) that represents the data manipulated by the BPs and their links. The constructed KG is powered by a graphical capsule neural network, and its purpose is to enable predictive analysis of future activities during the progression of a BP. This DSS4BP is based on the G-CAPS-NN architecture trained to discover the KG-BP. This KG excels in capturing complex dependencies within the activity flows contained in the different BPs specifications. Thus, the developed graph promotes a high prediction of future events and a deep contextual understanding of BP variations and evolution. Our second contribution is a chat-bot named BP for Decision Support System (BP-DSS3), which refines the GPT-3.5-turbo chat-bot to assist BP managed making more informed decisions. This BP-DSS3 chat-bot leverages deep learning techniques to provide personalized and domain-specific decision support. After the training phase, it achieves a high level of precision and accuracy when managing real-world scenarios, such as Alignment with Organizational Objectives (AOO) and Risk Management and Contingency Planning (RMCP).

The experiments are conducted basing on real-world data, the two proposed frameworks have demonstrated significant improvements in terms of efficiency, adaptability, and performance compared to traditional approaches. DSS4BP enables organizations to proactively identify inefficiencies and predict future outcomes of deployed business processes, while BP-DSS3 significantly improves decision-making by providing actionable and domain-specific information.

The keywords: Artificial Intelligence, Business Process, Business Process Model, Capsnet, Chat-bots, Decision Making, Event Logs, GPT-3.5-turbo, Knowledge Graphs.

تلخيص

تشكل إجراءات الأعمال قلب أنظمة المعلومات في المنظمات الحديثة. وبالتالي، يتم استخدامها بشكل مكثف، سواء في إدارة موارد الشركات المختلفة أو في أنشطة **صنع القرار** والمحاذاة الاستراتيجية. المواصفات المجردة (أو النماذج) التي تعبر عن المنطق التجاري وراء عمليات الأعمال هي أدوات مفاهيمية أساسية مفيدة لمهام مختلفة، وتتنوع أشكال النمذجة والتحليل والمراقبة والصيانة. ومع ذلك، مع الزيادة المذهلة في حجم البيانات التي يتم التعامل معها أثناء دورة حياة **عمليات الأعمال**، والتي غالبًا ما تكون غير متجانسة في طبيعتها، أثبتت الأساليب التقليدية لنمذجة واستخراج نماذج **عمليات الأعمال** أنها غير فعالة، مما يعيق إجراءات صنع القرار.

للتغلب على هذه القيود، نستفيد في هذه الأطروحة من أحدث التطورات التي تحققت في مجال **الذكاء الاصطناعي** من أجل تحسين أنظمة دعم القرار في مجال إدارة عمليات الأعمال. تتكون المساهمة الأولى لهذه الأطروحة من إطار مفاهيمي يسمى **الرسوم البيانية المعرفية لعمليات الأعمال (DSS4BP)**، والذي يسمح ببناء رسم بياني معرفي (KG) يمثل البيانات التي تتم معالجتها بواسطة عمليات الأعمال وروابطها. تعتمد KG المصممة على شبكة عصبية كبسولة رسومية، والغرض منها هو تمكين التحليل التنبئي للأنشطة المستقبلية أثناء تقديم BP. مساهمتنا الثانية هي روبوت دردشة يُدعى BP لنظام دعم القرار (BP-DSS3)، والذي يعمل على تحسين روبوت الدردشة **GPT-3.5-turbo** لمساعدة BP المُدارة على اتخاذ قرارات أكثر استنارة.

يعتمد DSS4BP على بنية **G-CAPS-NN** المدربة لاكتشاف KG-BP. يتفوق رسم المعرفة هذا في التقاط التبعيات المعقدة داخل تدفقات النشاط الواردة في مواصفات BP المختلفة. وبالتالي، يعزز الرسم البياني المطوّر التنبؤ العالي بالأحداث المستقبلية والفهم السياقي العميق لاختلافات BP وتطورها. من ناحية أخرى، يستفيد روبوت الدردشة BP-DSS3 من تقنيات التعلم العميق لتوفير دعم القرار الشخصي والمحدد للمجال. بعد مرحلة التدريب، يتم تحقيق مستوى عالٍ من الدقة والإتقان عند إدارة السيناريوهات الواقعية، مثل التوافق مع الأهداف التنظيمية (AOO) وإدارة المخاطر والتخطيط للطوارئ (RMCP).

بالاعتماد على التجارب التي أجريت على أساس بيانات واقعية مستمدة من مجالات تطبيقية مختلفة، أظهر الإطاران المقترحان تحسينات كبيرة من حيث الكفاءة والقدرة على التكيف والأداء مقارنة بالطرق التقليدية. يعمل BP-DSS3 على تحسين عملية اتخاذ القرار بشكل كبير من خلال توفير معلومات قابلة للتنفيذ ومحددة للمجال، بينما يمكن KG-BP المؤسسات من تحديد عدم الكفاءة بشكل استباقي والتنبؤ بالنتائج المستقبلية ل**عمليات الأعمال** المنشورة.

الكلمات المفتاحية: الذكاء الاصطناعي، عملية الأعمال، نموذج عملية الأعمال، Caps-net، Chat-bots، اتخاذ القرار، سجلات الأحداث، GPT-3.5-turbo، الرسوم البيانية المعرفية.

Resume

Les processus métiers (PM) sont au coeur des systèmes d'information des organisations modernes. Ils sont intensivement exploités, aussi bien dans la gestion des différentes ressources des entreprises que dans les activités de prise de décisions et d'alignement stratégique. Les spécifications abstraites ou modèles des BP expriment la logique métier véhiculée. Ils sont donc des outils conceptuels incontournable pour toute activité de modélisation, d'analyse et de maintenance. Cependant, avec l'accroissement spectaculaire du volume des données manipulées durant le cycle de vie des BP, et qui sont souvent de nature hétérogène, les approches conventionnels de fouille des modèles de BP s'avèrent inefficaces, ce qui entrave les actions de prise de décisions.

Pour surmonter ces limitations, nous exploitons dans cette thèse les dernières avancées réalisées dans le domaine de l'IA afin d'améliorer les systèmes d'aide à la décision dans le domaine de la gestion des BP. La première contribution de cette thèse consiste en un cadre conceptuel, appelé DSS for BP (DSS4BP), qui permet de construire un Knowledge Graph (KG) qui représente les données manipulées par les BP et leurs liens. Le KG construit est alimenté par un réseau neuronal à capsules graphiques, et son but est de permettre une analyse prédictive des activités futures au cours de la progression d'un BP. Ce DSS4BP est basé sur l'architecture G-CAPS-NN entraînée pour découvrir le KG-BP. Ce graphe de connaissances excelle dans la capture de dépendances complexes au sein des flux d'activités contenus dans les différentes spécifications des BP. Notre deuxième contribution est un chat-bot nommé BP for Decision Support System (BP-DSS3), qui affine le chat-bot GPT-3.5-turbo pour aider les BP gérés à prendre des décisions plus éclairées. Ce chatbot BP-DSS3 s'appuie sur des techniques d'apprentissage profond pour fournir une aide à la décision personnalisée et spécifique au domaine. Après la phase de formation, il atteint un niveau élevé de précision et d'exactitude dans la gestion de scénarios réels, tels que l'alignement avec les objectifs organisationnels (AOO) et la gestion des risques et la planification des mesures d'urgence (RMCP).

Grâce aux expérimentations conduites sur des données réelle, les deux cadres proposés ont démontré des améliorations significatives en termes d'efficacité, d'adaptabilité et de performances par rapport aux approches traditionnelles. DSS4BP permet aux organisations d'identifier de manière proactive les inefficacités et de prédire les résultats futures des processus métiers déployés, tandis que BP-DSS3 améliore considérablement la prise de décision en offrant des informations exploitables et spécifiques au domaine.

Les mots clés: Intelligence Artificielle, Processus Métier, Modèle de Processus Métier, Réseau de Capsules, Chatbots, Prise de Décision, Journaux d'Événements, GPT-3.5-turbo, Graph de Connaissances.

Acknowledgements

I would like to thank Allah, my parents, husband and all who give me the patient, ability and the force to accomplish this thesis. Further, i would like to thank all the members of my thesis committee –Ali KHEBIZI, and Khaled HALIMI– for their help in preparation of this work. Further, i would like to thank the jury members who evaluated this work.

I also gratefully acknowledges the support for this work offered by LABSTIC laboratory and Computer science department.

Contents

Acknowledgements	iv
Abstract	v
List of Figures	ix
List of Tables	xi
Abbreviations	xii
General Introduction	1
I Part I: Background and literature review	10
1 Business Processes	11
1.1 Introduction	11
1.2 Definitions, concepts and examples of BPs	12
1.2.1 Definitions	12
1.2.2 Examples of BPs	12
1.2.3 Concepts of BPs	14
1.2.3.1 BP execution	14
1.2.3.2 BP instances	14
1.2.3.3 Activities or tasks	14
1.2.3.4 BP models	14
1.2.3.5 BP life cycle	15
1.2.3.6 Business Process Management System (BPMS)	17
1.3 BP data management	17
1.3.1 Event-log data	17
1.3.2 Textual data	19
1.4 BP modeling	20
1.4.1 Motivation of BP modeling	21
1.4.2 The categorization of the BP models	21
1.4.2.1 Graphical models	22
1.4.2.2 Formal models	23
1.4.2.3 Language BP models	25
1.5 BP mining approaches	26
1.5.1 Process discovery approaches	28

1.5.2	Conformance checking approaches	29
1.5.3	Process enhancement approaches	31
1.6	Conclusion	33
2	Decision Systems through Knowledge Representation Integration	34
2.1	Introduction	34
2.2	Definitions and application domains of DSS	34
2.2.1	Definitions	35
2.2.2	Application domains of DSS	35
2.3	Deploying DSS in BP domain	36
2.4	Types of DSS	37
2.4.1	Types of DSS based input data	38
2.4.1.1	Data-Driven DSS	38
2.4.1.2	Model-Driven DSS	39
2.4.1.3	Knowledge-Driven DSS	40
2.4.1.4	Document-Driven DSS	40
2.4.2	Types of DSS based on target goal	42
2.4.2.1	Operational decisions	42
2.4.2.2	Structural decisions	42
2.5	Knowledge representation	43
2.5.1	The description and types of knowledge	43
2.5.1.1	The knowledge pyramid	44
2.5.1.2	Types of knowledge	45
2.5.2	The cycle of KR	46
2.6	Integration of KR approaches for an effective DSS system	48
2.6.1	Logical representation	48
2.6.2	Frames representation	49
2.6.3	Graphical representation	50
2.7	Conclusion	53
3	Artificial Intelligence techniques	55
3.1	Introduction	55
3.2	Machine Learning techniques	56
3.3	Artificial neural networks	57
3.3.1	Definitions of ANN	58
3.3.2	Key components of ANN	58
3.3.2.1	Perceptron	58
3.3.2.2	ANN layers	59
3.3.3	ANN learning process	62
3.3.3.1	Forward propagation	62
3.3.3.2	Calculation of the loss function	63
3.3.3.3	Backward propagation	63
3.4	The deep learning models	64
3.4.1	Definitions	64
3.4.2	The difference between DL, ML and ANN	65
3.4.3	Discriminative DL models	66
3.4.3.1	Capsule Neural Networks	67

3.4.3.2	Graph Neural Networks	70
3.4.3.3	Capsule Graph Neural Networks	72
3.4.4	Generative DL large language models	74
3.4.4.1	Transformer architecture	74
3.5	Industrial Projects leaded by GDLLL models and chat-bots systems . . .	76
3.5.1	List of popular GDLLL models	77
3.5.2	Chat-bots systems	78
3.6	Conclusion	81
4	Literature review of the BP-DSS approaches	82
4.1	Introduction	82
4.2	Search methodology	83
4.2.1	Research question	83
4.2.2	Relevant research works selection	83
4.3	Analysis of the BP methods	85
4.3.1	Input	85
4.3.2	Output	86
4.3.3	The employed assumption	88
4.3.4	The type of the used algorithm	93
4.3.5	The evaluation data	97
4.4	Synthesis of the selected works analysis	97
4.5	Road-map and research-gap	104
4.5.1	Discovering and enhancing BP models for decision making	104
4.5.1.1	BP mining field	104
4.5.1.2	Decision support systems outside the BP field	106
4.5.1.3	Deploying KG through Caps-nets in BP field	108
4.5.2	Deploying chat-bot techniques for BP decisions	109
4.5.2.1	Smart BPs decision making	110
4.5.2.2	Fine-tuned generative models for decision making	110
4.5.2.3	Fine-tuned generative models in BP decision making . . .	111
4.6	Conclusion	111
II	Part II: Contributions	113
5	Towards a BP-DSS through KG Capsule Network	114
5.1	Introduction	114
5.2	Development of DSS4BP framework	114
5.2.1	The KG-BP's assumption	116
5.2.2	Event-Log specification	117
5.2.3	Specification of KG-BP model	119
5.2.4	Graph capsule neural net discover KG-BP	121
5.2.4.1	Primary capsule module (PCM)	122
5.2.4.2	Capsule network module (CNM)	122
5.3	Experimental results	124
5.3.1	BPI19 event-log description	124
5.3.2	The evaluation setup	124

5.3.3	The G-Caps-NN evaluation	124
5.3.3.1	The evaluation of the G-CAPS-NN training	125
5.3.3.2	The ablation analysis of the G-Caps-NN	127
5.3.4	Comparison of the G-Caps-NN with the past studies' findings	129
5.3.5	The KG-BP scalability through different performance metrics	132
5.3.5.1	KG-BP optimization	132
5.3.5.2	Event prediction through KG-BP	133
5.3.6	Limitations of the G-Caps-NN	133
5.4	Extended discussions and future perspectives	134
5.4.1	Real-world implementation of DSS4BP	135
5.4.2	Advantages of DSS4BP compared to existing DSS systems	136
5.4.2.1	Improved Process Awareness and Automation	136
5.4.2.2	Enhanced Predictive Decision-Making	137
5.4.2.3	Integration with AI and Knowledge Representation	138
5.4.3	Future perspectives of DSS4BP	139
5.4.3.1	Expanding data validation	139
5.4.3.2	Extending DSS4BP applications	139
5.4.3.3	Integration with Reinforcement Learning	140
5.5	Conclusion	140
6	BP-DSS3: A Business Processes Decision-Support Chatbot through the Language Model GPT-3.5-turbo	141
6.1	Introduction	141
6.2	Motivations	142
6.3	Development of the chatbot BP-DSS3	145
6.3.1	Problem statement and specification	146
6.3.2	Development of the Chatbot BP-DSS3	147
6.3.2.1	Pre-processing stage	147
6.3.2.2	Fine-tuning stage	149
6.3.2.3	Testing the fine-tuned model	150
6.4	Experimental Results	153
6.4.1	Capabilities of BP-DSS3 for decision-making	154
6.4.2	Case studies	156
6.4.2.1	Scenario 1: Alignment with organizational objectives (AOO)	157
6.4.2.2	Scenario 2: Risk management and contingency planning (RMCP)	159
6.4.2.3	Performance metrics evaluation	160
6.4.3	Limitations of BP-DSS3	164
6.5	The process of deploying BP-DSS3 in real-world	166
6.6	Conclusion	168
	General Conclusion	168
	Bibliography	172

List of Figures

1.1	BPM life cycle [1]	15
1.2	The workflow model of the online purchase order event-log data-set	22
1.3	BPMN of online order management application's [2]	23
1.4	The petri net of online order management application's	24
1.5	The automata model of online order management application's	25
1.6	The relationships between the PM approaches [3]	27
2.1	The types of DSS systems [4]	38
2.2	The knowledge pyramid [5]	44
2.3	The cycle of KR in AI [6]	47
2.4	The nodes of the e-commerce graph representation [7]	50
2.5	The relationship of the e-commerce graph representation [7]	50
2.6	The semantic graph of the Mammal animals [8]	51
2.7	The relationship of the e-commerce graph representation [9]	52
2.8	An example of a complex semantic graph of social media [10]	53
3.1	The difference of ML and traditional programming [11]	56
3.2	The perceptron mechanism[12]	58
3.3	The ANN layers [13]	59
3.4	The ANN learning process [14]	62
3.5	The curve of minimizing the error through the gradient descent function [15]	64
3.6	The difference between ML and DL models [16]	65
3.7	The generative and discriminative DL models [17]	66
3.8	Examples of the discriminative DL models [17]	67
3.9	The importance of the dynamic routing of the caps-net [18]	67
3.10	The capsule structure [18]	68
3.11	The feature transformation from local into global ones [18]	69
3.12	The GNN layers [19]	70
3.13	The transformer architecture [20]	75
3.14	The transformer architecture with attention mechanism [21]	76
4.1	Problem statement of using Enhanced BP models for decision making in the BP area	105
5.1	Graph Neural Networks in the AI Research Landscape [22]	115
5.2	The KG-BP representing the loan BP	120
5.3	Graph capsule neural network architecture	121
5.4	Graph BP routing layer	123

5.5	Training the G-CAPS-NN with the complete BPI19 $\mathcal{EL}$ data	125
5.6	The global message flow and accuracy relationship	133
5.7	Integrating the DSS4BP in a real-company	135
6.1	Search volume for “generative AI” has exploded in late 2022 [23]	142
6.2	As the pandemic wanes, many workers are adopting a hybrid work schedule [23]	145
6.3	Illustration of the BP decision-making problem using chatbots technology	146
6.4	Architecture and functionalities of the Developed chatbot BP-DSS3 . . .	147
6.5	A formatted excerpt of the BPI19 dataset	148
6.6	The setup of OpenAI API client code of the GPT3.5-turbo	150
6.7	The code for uploading the dataset	151
6.8	The code for fine-tuning process of the GPT3.5-turbo model	152
6.9	The code for utilizing the fine-tuned GPT-3.5-turbo model	153
6.10	The GPT-3.5-turbo trained with PET dataset	154
6.11	The GPT-3.5-turbo trained with BPT19 dataset	155
6.12	Responses of the chat-GPT to questions of the AOO BP (part 01)	157
6.13	Responses of the chat-GPT to questions of the AOO BP (part 02)	157
6.14	Responses of the chat-GPT to questions of the AOO BP (part 03)	158
6.15	Response of the BP-DSS3 chatbot for the AOO business process (part 01)	158
6.16	Response of the BP-DSS3 chatbot for the AOO business process (part 02)	159
6.17	Response of the general chat-GPT to suggest the possible future risks . .	159
6.18	Response of the BP-DSS3 chatbot for predicting future risks (part 01) . .	160
6.19	Response of the BP-DSS3 chatbot for predicting future risks (part 02) . .	160
6.20	Deploying the BP-DSS3 in a real-company	166
6.21	The Chatbot-G-Caps-NN architecture for BP decision making	171

List of Tables

1.1	An example of the online purchase order event-log data-set	18
1.2	Comparison of different types of PM approaches	27
2.1	Comparison between Semantic Graph and Knowledge Graph.	53
3.1	Non-linear activation functions [24]	60
3.2	Advantages and Disadvantages of Activation Functions [24]	61
4.1	Overview of the BP based conventional PM and PM based DL approaches	99
5.1	A sample retrieved from an event log of the bank loan system	117
5.2	The Notations used throughout the paper with examples	118
5.3	The G-CAPS-NN Results through 04 metrics	126
5.4	Performance Comparison of Different Model Configurations	128
5.5	Comparison of G-Caps-NN accuracy with related works	131
5.6	The message under clusters	132
5.7	The prediction evaluation based KG-BP features	134
5.8	Business Functions Enabled by DSS4BP	136
5.9	Comparison Between DSS4BP and Traditional DSS	137
5.10	Comparison of DSS4BP and Traditional Predictive DSS	138
5.11	Comparison between DSS4BP and Traditional DSS	139
6.1	The confusion matrix of BP-DSS3 and Chat-GPT	161
6.2	A performance evaluation of the test case studies	162

Abbreviations

ABM	A gent B ased M odeling
AI	A rtificial I ntelligence
AOO	A lignment with O rganizational O bjectives
ANN	A rtificial N eural N etwork
BP	B usiness P rocess
BPM	B usiness P rocess M anagement
BPMN	B usiness P rocess M odel and N otation
Caps-Net	C apsule N etwork
CNM	C apsule N etwork M odule
DL	D eep L earning
DRL	D eep R epresentation L earning
DSS	D ecision S upport S ystem
DSS4BP	D ecision S upport S ystem f or B P mining
DT	D ecision T rees
DTR	D ecision T rees R egression
EL	E vent L og
ES	E xpert S ystem
FN	F alse N egative
FP	F alse P ositive
GAN	G enerative A dversarial N etworks
G-Caps-NN	G raph C apsule N eural N etwork
GCNN	G raph C onvolutional N eural N etworks
GNN	G raph N eural N etwork
GMF	G lobal M essage F low
HMM	H idden M arkov M odels
IOT	I nternet O f T hings
KG	K nowledge G raph
KG-BP	K nowledge G raph B usiness P rocess
KNN	K - N earest N eighbors

KPI	K ey P erformance I ndicators
LR	L inear R egression
MAE	M ean A bsolute E rror
ML	M achine L earning
MMM	M ap M iner M ethod
MSE	M ean S quared E rror
NLP	N atural L anguage P rocessing
OBM	O riented B ased M odeling
PM	P rocess M ining
PMC	P rimary C apsule M odule
RL	R einforcement L earning
RMCP	R isk M anagement and C ontingency P lanning
RNN	R ecurrent N eural N etworks
RMSE	R oot M ean S quared E rror
RQ	R esearch Q uestion
TN	T rue N egative
TP	T rue P ositive
TQM	T otal Q uality M anagement

General Introduction

This first chapter is dedicated to expound the general context of the thesis and the addressed research questions, followed by a summary of the main contributions. Then, it presents the structure of the manuscript and the list of the achieved scientific production.

1. Context of the thesis

Nowadays, the recent advancements realized in the field of Information and Communication Technologies (ICT) have completely revolutionized the way in which organizations' Information Systems (IS) are perceived and managed. In this context, the Business Processes (BPs) [25] are considered pivotal components inevitable for understaing the functioning of organizations. These BPs are serving as structured activities or tasks aimed to achieve a specific business goals.

In these circumstances, the increasing complexity of these modern organizations, coupled with the growing volume of data generated by these processes, have led to the emergence of advanced methodologies and technologies for process analysis, optimization, and decision-making. These methodologies aim to improve efficiency, enhance productivity, and support executive decision-making by providing insights into the performance and compliance of BPs.

In recent years, business Process Mining (PM) [26] has gained prominence as a discipline that extracts and evaluates BP models using execution data recorded in organizational IS. These data (*often referred as Event Logs (ELs)*), consist of a set of traces representing the historical sequence of activities performed during the execution of a single process instance. From a PM perspective, analyzing and exploiting ELs can be very beneficial for companies. Thus, in order to assess and monitor its activities, organizations can uncover the BP models contained in the EL and which describe their activities with the associated constraints governing their execution. In a nutshell, BPs models are considered as effective tools facilitating decision-making in business.

To attempt this BP decision-making goal, several approaches ranging from conventional Machine Learning (ML) to Deep Learning (DL) approaches were developed.

Despite the utility of conventional PM techniques [27–31], they are often limited by the complexity of the generated BP models, which can be challenging to interpret and deploy them. This complexity is primarily due to the increasing volume of data in ELs. To address this limitation, modern approaches that take advantage of Deep Representation Learning (DRL) [32] have been introduced. This latest type of approaches utilize advanced algorithms to extract features and uncover richer representations of BP models. However, the *task variation*, where the activities contained in the considered BP are performed differently and inconsistently, can reduce the performance of DRL in discovering straightforward BP models.

To tackle the previous issue of *task variation*, selecting the appropriate *input data representations* is a critical step to enhance the performance of the DRL approaches. In this regard, Knowledge Graphs (KGs), recognized for their expressive and structured nature, have emerged as a promising tool for modeling BP data. In fact, these KGs enable the representation of event features as interconnected nodes, providing additional layers of knowledge and ensuring more accurate and complete BP models. By propagating feature values across nodes, KGs can significantly improve the quality of event embeddings, thus addressing the limitations of traditional approaches.

In this context, automatic BP decision-making to conduct efficient and effective business activities represent a pivotal challenge, especially in an environments characterized by rapid changes and high level of competition between competitors. Further, the importance of integrating sophisticated approaches for supporting BP decision-making is caused by the globalization, market dynamics, and resource management issues. In this circumstances, the integration of technologies such as Artificial Intelligence (AI), Internet of Things (IoT), and Natural Language Processing (NLP) offers promising solutions for enhancing decision-making frameworks.

The field of DRL in BP discovery is still evolving. While Graph Neural Networks (GNNs) [33] have been explored for BP modeling, their applications remain limited to capturing basic BP representations. In recent years, the spectacular explosion of the mass of data generated daily by the IS of organizations has propelled the significant potential of KGs, GNNs and capsule network to comprehensively discover BP models useful for supporting decision-making processes. Thus, the use of KGs, combined with GNNs, has become a very attractive and promising research area. In particular, conversational AI tools such as ChatGPT, have shown great potential for completely transforming and enhancing Decision Support Systems (DSS). Where, by leveraging the capabilities of language models decision-makers are able

to address complex BP-related queries efficiently, revolutionize decision-making and provide a more effective framework for managing modern BPs.

This thesis is situated at the crossroads of several areas of the computer science discipline. The purpose of this project is to use recent AI techniques ranging from the DL into the generative models to consolidate the decision-making in the BP field. The next section of the chapter aims to present the contributions of this thesis.

2. The problematic of the thesis

To answer the primary research question, *"How can advanced AI methodologies enhance decision-making within the BP domain?"*, we analyzed the BP domain and addressed the following sub-research questions:

- (a) *Can abstract BP models contribute for supporting decision-making activities?*
 - Abstract BP models provide high-level, generalized representations of processes, enabling decision-makers to focus on critical insights rather than low-level operational details. This question investigates how such models can effectively aid decision-making by distilling complex event logs into actionable insights.
- (b) *What are the most efficient knowledge representation models to be used to enhance decision-making in the context of BP?*
 - Efficient knowledge representation is crucial for interpreting, organizing, and leveraging BP data. This question explores representations like Knowledge Graphs (KGs) basing on the dynamic routing to understand their role in structuring BP data for enhanced reasoning and analysis.
- (c) *Based on the relevant knowledge representation models of BP, what are the suitable reasoning frameworks and systems to be deployed to improve decision-support systems?*
 - This question examines their potential in discovering Knowledge Graphs for BP (KG-BP), which encapsulate BP semantics, relationships, and patterns for better decision-making.
- (d) *Can Generative AI models and conversational tools (e.g., chatbots) be exploited to make the decision-making ecosystem more interactive by providing responses to user queries in a more explicit and textual format?*
 - Generative AI has the potential to transform decision-making ecosystems by providing dynamic, user-friendly interfaces for querying BP data. This

question evaluates how AI-driven conversational agents can foster better interaction and understanding of BP insights.

By addressing these questions, we aim to uncover innovative methodologies and tools that significantly enhance decision-making capabilities in the BP domain. In this regard, the contributions of this thesis are described in the following section.

3. The contributions of the thesis

Given the contextual execution conditions of BP, human factors, and the intentional business decisions of an EL data, a large degree of internal heterogeneity between execution data and uniform behavior is enforced. In this context, analyzing those factors from the performance characteristics of the event log data is a challenge for both conventional and DL methods.

(a) Contribution 01:

The first contribution represent an exhaustive literature review expressing a deep analysis of the BP mining techniques. In this contribution, we have identified relevant criteria allowing to compare the different conducted works (*input, output, the employed assumption, the type of the algorithm and the evaluation data*).

At the end of this contribution we classified the techniques into two main groups based on the paradigm used to accomplish the underlying BP goal.

This contribution was published in an **C.AFIFI, A.KHEBIZI, K.HALIMI (2024). A systematic survey of the business process mining-based approaches. International Journal of Business Process Integration and Management, 11(4), 314-331.**

Following this thorough examination, the two classes that emerged contributed to establish the development of our contributions.

In this regard, the four following contributions aim to establish *a BP model as the backbone of the decision-making process*

(b) Contribution 02:

In order to answer the first research question [2a](#), the second contribution focuses to pre-process the event-log data as a starting step for the BP decision-making. This preprocessing stage aims to represent the BP event-log data transitions as Knowledge Graphs (KG). Such KG model will acts as a guideline that serves for providing a solid reference for various business decisions activities, such as BP prediction, recommendation systems, classification and clustering.

This second contribution was published in the **C.AFIFI, A.KHEBIZI,**

K.HALIMI (2023, April). Extracting and Exploiting the Behavior Business Process Graph through Transition-Centric Event-Log data. In 2023 International Conference on Artificial Intelligence and Applications (ICAIA) Alliance Technology Conference (ATCON-1) (pp. 1-6). IEEE.

(c) Contribution 03:

In order to answer the second research question 2b, the third contribution of this thesis aims to enhance the knowledge representation model of the event-log data, by integrating the semantics conveyed by the different existing links and relationships between tasks of BP. in this perspective, we proposed a novel Squash Graph Routing (SGR) mechanism. This routing is based on the feature propagation mechanism to be able to learn unified events feature's of BP and their relationships. Those discovered properties represents the valuable insights for automatically learning the semantic of BP features which can be used in various BP decision-making applications' (*process optimization, conformance checking, event prediction,...etc*).

This third contribution was published in the **C.AFIFI, A.KHEBIZI, K.HALIMI (2023, July). The Squash Behavior Graph Routing between the Business Process Event Capsules. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) (pp. 1-9). IEEE.**

(d) Contribution 04:

In recent years, the capsule graph neural network (Caps-GNN) has propitiated a set of predictive related fields. The power Caps-GNN is the learning of the relationship between the capsules. However, the relationship between the event capsules is currently lacking in the BP area, while being a crucial component for predicting the next action. In this context, before building our customized DL architecture based on the Caps-net, this contribution aims to predict the next activity, basing on the efficiency of the conceived graph capsule network. This contribution was published in the **C.AFIFI, A.KHEBIZI, K.HALIMI (2024, May). Business Process Activities Prediction through Capsule Neural Net framework. Algerian Doctoral Conference in Computer Science (ADCCS'2024)At: Ecole Nationale Supérieure d'Informatique, ESI, in Algiers, Algeria**

(e) Contribution 05:

In order to answer the third research question 2c and after ensuring the efficiency of the Caps-GNN in the BP decision-making field, the fifth contribution aims to discover the KG-BP (*detailed in the contribution 02*) by a new

DL architecture, named a Graph Capsule Neural Network (G-Caps-NN) and inspired by caps-GNN [34]. Through the customized dynamic routing (*detailed in the contribution 03*), such structure is able to learn the embedded event characteristics of the BP and their correlation degree. The conducted experiments, on both virtual and real event log data, exhibits the effectiveness and the efficiency of the suggested modeling in the areas of optimization and prediction tasks.

(This fifth contribution will be submitted)

According to the results of the previous, along with the emergence of the chat-bots, assisting the BP decision-making through virtual assistant is substantial. In this regard, the following contribution *is conducted to develop a chat-bot for BP decision-making*.

(f) **Contribution 06:**

In order to answer the fourth research question 2d, the sixth contribution of this thesis consists to investigate the adoption of innovative language models to improve decision-making in the context of BPs. Exploiting the NLP paradigm in the context of DSS related to BPs, and developing a conversational chat-bot whose capabilities are fine-tuned with domains specific BP data. The developed BP-DSS3 framework constitutes a thorough investigation of the groundbreaking potential of fine-tuned chat-bots for personalized and specific tasks. The conducted experiments exhibits the effectiveness and the efficiency of the suggested BP-DSS3 framework.

This sixth contribution was published in the **C.AFIFI, A.KHEBIZI, K.HALIMI**
doi: 10.1504/IJIEI.2025.10071287. BP-DSS3: A Business Processes Decision-Support Chat-bot through the Language Model GPT-3.5-turbo. International journal of Intelligent Engineering Informatics

4. Thesis organization

Additionally to this General introduction chapter, the thesis is structured into the following two main parts:

(a) **Part I: Related works**

The first part of the thesis attempts to present all the concepts, definitions and techniques that constitute the background needed to understand the remaining chapters. It also, contains a literature review of related works having tackled the initial research question. This part is organized in the following four chapters.

- i. **Chapter 01:** Exposes the necessary background of the BP area. We introduce the concepts associated to the BP domain with their representation models and the underlying techniques and tools.
- ii. **Chapter 02:** This chapter aims to explain the essential principles and terminology of the DSS systems and the KR stage. In this context, this chapter has provided a comprehensive exploration of how KR empowers DSS.
- iii. **Chapter 03:** This chapter aims to explain the advanced AI models. In this case, this chapter aims to present the Artificial Neural Network and the discriminative DL. Furthermore, because conversational languages play an important part in the DSS, this chapter discusses the generative DL models and several chat-bots existing in the real-life.
- iv. **Chapter 04:** this chapter offers a thorough qualitative literature review of BP methods-based PM approaches on one side and the BP based DL approaches on the other. The identified studies are categorized based on their inputs, outputs, fundamental assumptions, algorithm type, and extra-functional properties. This chapter concludes with a broad classification of techniques into two groups based on the paradigm used to accomplish the underlying BP goal.

(b) **Part II: Contributions**

Following a thorough literature analysis of the BP domain, this second section discusses the two contributions of the BP for a decision-making purpose across the next two chapters.

- i. **Chapter 05:** This chapter aims to develop the first contributions (2,3,4 and 5) detailed in the section 3. The motivation, assumption and the experiments were developed in this chapter. At the end, the several limitations of our proposed architecture was discussed.
- ii. **Chapter 06:** This chapter aims to develop the the second contribution 6 detailed in the section 3. The motivation, assumption and the experiments were developed in this chapter. At the end, the several limitations of our proposed architecture was discussed.

Additionally, **the last chapter** of the thesis summarizes the key findings from this work and outlines our futures research perspectives.

5. **List of scientific production** The work carried out within this thesis has been the subject of several scientific productions, presented below.

- **International Journals**

- (a) A systematic survey of the business process mining-based approaches
Class B-scopus: January 2024 International Journal of Business Process
Integration and Management 11(4):314-331
DOI: 10.1504/IJBPM.2024.142658
- (b) BP-DSS3: A Business Processes Decision-Support Chat-bot through the
Language Model GPT-3.5-turbo
Journal B-Scopus: Int. J. of Intelligent Engineering Informatics
- (c) A Road-map for Mining Business Process Models via Artificial Intelli-
gence Technique
Class C: January 2022 International Journal of Informatics and Applied
Mathematics 5(1)
DOI: 10.53508/ijiam.1036234

- **International conferences**

- (a) Extracting and Exploiting the Behavior Business Process Graph through
Transition-Centric Event-Log data
July 2023
Conference: 2023 International Conference on Artificial Intelligence and
Applications (ICAIA) Alliance Technology Conference (ATCON-1)At:
Bangalore, India
DOI: 10.1109/ICAIA57370.2023.10169793
- (b) The Squash Behavior Graph Routing between the Business Process Event
Capsules
July 2023
Conference: 14th ICCCNT IEEE Conference At: IIT-Delhi, Delhi, India
DOI: 10.1109/ICCCNT56998.2023.10307176
- (c) A Review of Business Process Mining and a Trend Analysis of Artificial
Intelligence Powered Approaches
December 2021
Conference: FOURTH CONFERENCE ON INFORMATICS AND AP-
PLIED MATHEMATICS IAM'21At: Guelma, Algeria
- (d) BUSINESS PROCESS MODELING
November 2022
Conference: IAM22 CONFERENCE, Guelma,

- **National conferences**

- (a) Business Process Activities Prediction through Capsule Neural Net frame-
work
May 2024

Conference: Algerian Doctoral Conference in Computer Science (AD-CCS'2024)At: Ecole Nationale Supérieure d'Informatique, ESI, in Algiers, Algeria

Part I: Background and literature review

Chapter 1

Business Processes

1.1 Introduction

Nowadays, Business Processes (BPs) play a critical role in companies management and constitute the cornerstone of operational, tactical and strategic decisions. In the recent years, with the growing complexity of processes, managing and optimizing these workflows have become crucial in ensuring competitiveness and adaptability. In this context, modern Information Systems (IS) have to play a pivotal role in managing BP life-cycle by providing the suitable infrastructure needed for to store, process, and explore vast amounts of data generated during daily operations reflecting BP execution. In the circumstances, Process-Aware Information Systems (PAIS) is developed as a specialized systems that help in the execution and monitoring of BPs. In fact, PAIS not only improves workflow automation, but it also ensures conformity to corporate objectives by enabling real-time analysis and performance evaluation.

The effective management of the data generated by these systems is vital for organizations to derive actionable decisions. Hence, Process Mining (PM) is recognized as a discipline that bridges the gap between data and actionable insights. By leveraging stored EL, PM techniques enable organizations to analyze, visualize, and optimize their BPs. This ensures transparency, uncovers inefficiencies, and provides a solid foundation for data-driven decision-making in modern BP management.

This chapter covers the fundamental concepts and definitions of the BP field, as well as BP models and numerous usual PM methodologies.

1.2 Definitions, concepts and examples of BPs

Nowadays, due to the remarkable growth of connectivity and the widespread use of the internet, firms are now expected to conduct their BP's in a more competitive environment where they must demonstrate a high level of agility and flexibility. In this circumstance, they must strive for an immediate reconfiguration by expanding new BP's and upgrading the already existing ones.

In this context, the BP paradigm becomes inevitable to streamline individual activities and ensure efficient resource of any business. This goal contributed to the development of the Business Process Management (BPM) technology as a thorough and systematic approaches, methodologies, and techniques for fundamentally transforming the way that these businesses operate.

This section presents the definitions of the BP and illustrates them with real-word example, then it exposes the associated concepts and notions.

1.2.1 Definitions

The BP domain refers to the comprehensive set of activities, tasks, and workflows within an organization aimed to achieve specific business objectives. It encompasses several steps from the initial conception of a BPs to its execution, monitoring, and continuous improvement. Effective management of BPs is crucial for ensuring efficiency, consistency, and quality in organizational operations.

The related literature uses the most adopted definitions of BP that are presented bellow.

Definition 1.1. *BP constitutes simply a set of activities performed by one or more businesses in an attempt to accomplish certain business goals [35, 36].*

Definition 1.2. *A BP can be formally defined as a structured, repeatable set of inter-related tasks or activities performed by individuals, systems, or organizations to achieve a specific organizational objective or deliver a service or product to a customer. It involves a sequence of inputs, transformations, and outputs that contribute to meeting a predefined goal [26].*

1.2.2 Examples of BPs

This subsection aims to explain the commonly recognized BPs.

Example 1.1. (Purchase Order Process) *The Purchase Order (PO) process is essential for procurement activities in organizations. It involves the acquisition of goods or services required by an organization while ensuring cost-effectiveness, compliance, and timely delivery. In the following are the steps in order to execute the PO process:*

- **Request for Goods/Services:** *An internal department identifies a need and submits a purchase requisition form detailing the items/services required.*
- **Approval of Requisition and Budget Allocation:** *Managers review the requisition and check if it aligns with the budget and organizational priorities.*
- **Vendor Selection:** *Procurement officers identify and evaluate vendors based on price, quality, and delivery terms.*
- **Issuance of a Purchase Order:** *A formal purchase order document is sent to the selected vendor, detailing the quantity, price and delivery timeline.*
- **Receipt of Goods/Services and Inspection:** *Once the vendor delivers, the organization inspects the goods/services for compliance with the purchase order.*
- **Payment Processing:** *Upon successful receipt and inspection, invoices are processed, and payments are made to the vendor.*

Example 1.2. (Loan Approval Process (Banking Example)) *The loan approval process ensures that loans are disbursed only to eligible applicants while minimizing risk to the bank. The following steps aims to accomplish this goal:*

- **Loan Application Submission:** *Customers submit a loan application with required documents like income proofs and credit history.*
- **Preliminary Review:** *Bank staff perform an initial check to ensure all necessary information is provided.*
- **Credit Assessment:** *The customer's creditworthiness is assessed using credit scores and financial records.*
- **Approval Decision:** *Based on the assessment, the loan is approved or rejected.*
- **Loan Disbursement:** *Approved loans are disbursed to the customer's account.*
- **Repayment Monitoring:** *The bank tracks repayments and takes action in case of defaults.*

In these BP application examples, several key components and stages are involved. Each contributing to the overall functionality and success of BPs. In the following the BP life cycle that manage these components as a unique bloc is described.

1.2.3 Concepts of BPs

This section seeks to describe the basic concepts of the BPs in order to comprehend the subsequent detailed information of the thesis.

1.2.3.1 BP execution

BP execution refers to the actual implementation of a BP, where the defined tasks and activities are carried out in a sequence or parallel manner to achieve the desired outcomes. This execution may involve humans, automated systems, or both, and is often monitored for performance and compliance.

Example 1.3. *For example, the event-log data described in the table 1.1 represents a purchase order BP execution.*

1.2.3.2 BP instances

A BP instance is a specific occurrence of a business process execution. It represents one complete run of the process, from start to finish, and is unique to the context or case being handled.

Example 1.4. *For example, processing an individual customer's order would be a single BP instance within the overall order management process.*

1.2.3.3 Activities or tasks

Activities or *tasks* are the smallest units of work within a BP. They represent specific actions or steps that need to be performed to achieve a particular goal within the process.

Example 1.5. *For example, in an order management process, tasks might include (Request for Goods, Approval of Requisition and Budget Allocation, or Vendor Selection.*

1.2.3.4 BP models

BP models are graphical or textual representations of a business process. They describe the structure, tasks, flows, and dependencies within the process.

Example 1.6. *Examples of BP modeling notations include Business Process Model and Notation (BPMN), Petri nets, and flowcharts. These models are used for understanding, analyzing, and improving processes.*

1.2.3.5 BP life cycle

As shown in the figure 1.1, the life-cycle [37] consists of main steps: *process design and modeling, process execution, process monitoring and control, and process improvement*. Each phase plays a critical role in ensuring that BPs operate efficiently, effectively, and in line with organizational objectives.

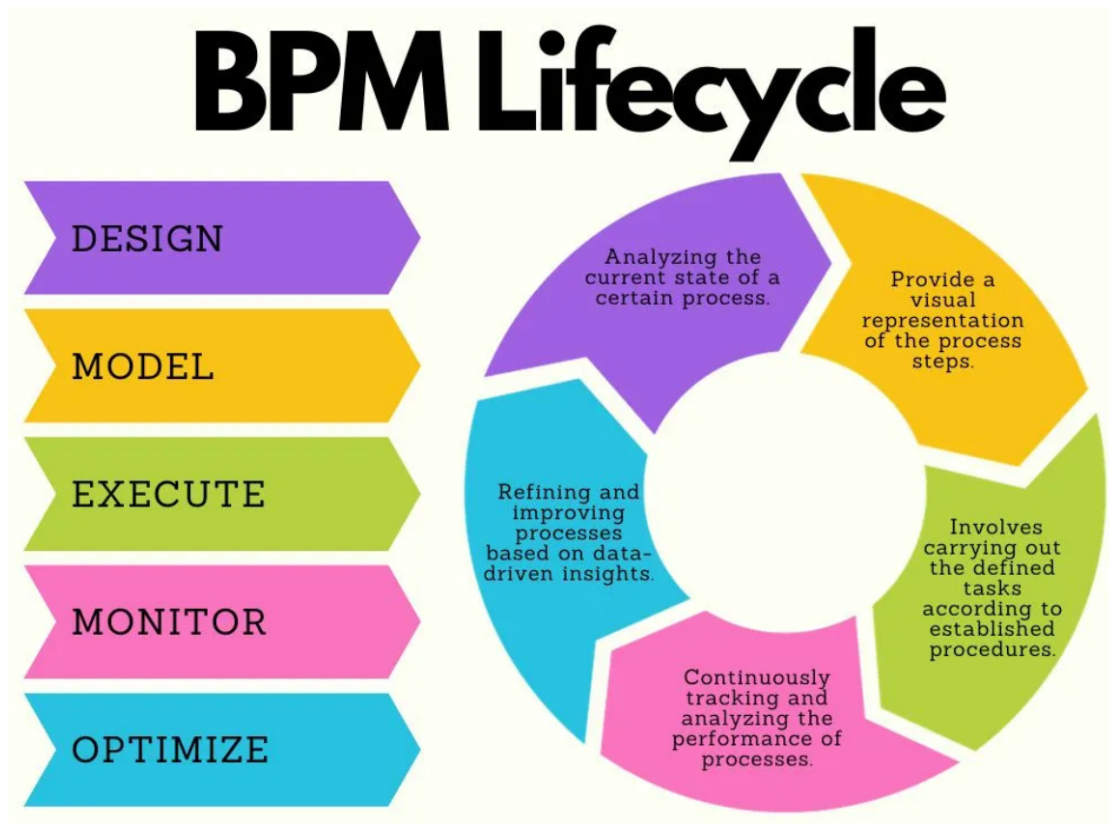


FIGURE 1.1: BPM life cycle [1]

1. **Process design and modeling:** The first phase of the life cycle involves defining the structure of the BPs. This step entails mapping out the sequence of activities, establishing roles and responsibilities, and specifying the decision points and rules that govern the process. Visual tools, such as Business Process Model and Notation (BPMN) diagrams, are often employed to create standardized representations of workflows. These diagrams provide stakeholders with a clear understanding of the process flow, allowing for effective communication and collaboration during the design phase. In this regard, the key considerations during this phase include:
 - **Objective Setting:** Clearly defining the goals the process aims to achieve.
 - **Role definition:** Assigning roles and responsibilities to stakeholders and teams involved in the process.

- **Compliance:** Ensuring the process aligns with organizational policies, industry standards, and regulatory requirements.
- **Decision Points:** Establishing rules and conditions for process branching and decision-making.

Example 1.7. *In a PO process, the design phase involves mapping activities such as submitting a requisition, approving budgets, selecting vendors, issuing purchase orders, and receiving goods. The roles, such as the requester, approver, and procurement officer, are clearly defined, and compliance rules, such as budget thresholds, are incorporated into the design.*

2. **Process execution:** After the design phase, the BP is implemented and executed using the defined workflows, rules, and roles. Process automation tools, such as Business Process Management Systems (BPMS) or Enterprise Resource Planning (ERP) systems, are often used to streamline execution and ensure adherence to the process model. These systems automate repetitive tasks, route activities to the appropriate stakeholders, and track progress in real-time.

Example 1.8. *In the PO process, execution begins when an employee submits a requisition through the system. The system then automatically routes the request to the approver, generates a purchase order upon approval, and sends it to the selected vendor. The system also records the receipt of goods and triggers the payment process based on predefined rules.*

3. **Process monitoring and control:** The third phase focuses on tracking the performance of the business process in real-time to ensure that it operates within the defined parameters. This involves collecting data on process activities and measuring performance against predefined Key Performance Indicators (KPIs), such as process cycle time, error rates, and compliance levels. Monitoring tools and dashboards are used to provide real-time insights into the process.

Example 1.9. *In the PO process, monitoring could involve tracking the time taken for requisition approvals, ensuring vendors deliver goods on time, and verifying that payments are processed accurately. If a delay occurs in the vendor delivery stage, control mechanisms, such as notifying the vendor or escalating the issue, can be triggered.*

4. **Process analysis and optimization:** The final phase involves analyzing process performance data to identify areas for optimization and implementing strategies to enhance efficiency and effectiveness. This phase relies heavily on techniques like PM to extract insights from event logs and uncover bottlenecks, inefficiencies, or

deviations from the desired workflow. Improvement strategies often incorporate methodologies such as Lean, Six Sigma, or Total Quality Management (TQM).

Example 1.10. *In the PO process, improvement initiatives might involve automating low-value approvals to reduce cycle time, introducing a preferred vendor list to minimize procurement delays, or providing training to staff on compliance and process adherence. These changes ensure that the process evolves to meet organizational objectives effectively.*

1.2.3.6 Business Process Management System (BPMS)

In order to manage those critical phases of the BP life cycle, Business Process Management Systems (BPMS) should be deployed as a software application and a set of technologies suited to define, automate, and evaluate the BP's. It serves as a tool to automate workflows, integrate systems, and track performance metrics, ensuring efficient and consistent process execution.

Understanding the characteristics of each type of data structure and BP model is required to achieve those BP decision-making goals. The purpose of the subsequent sections is to outline the various types of data structures that are used in the BP area. This is followed by the different types of output BP models that are intended to be constructed.

1.3 BP data management

Types of BP data refer to various forms of information and data generated, used, or analyzed during the execution of BP. This data helps to understand process performance, identify bottlenecks, and improve overall efficiency. There are several types of BP data that can be classified based on their source, structure, and purpose within the process life-cycle. This section is intended to describe the two types of data (*event log and textual data*) utilized in this area.

1.3.1 Event-log data

Both the discovered BP model or the hand made one can be deployed can be deployed by any company for managing the online order fulfillment process. After its deployment, such a BP is executed by an important number of customers (*human, machines or invoked by others BP's*). Each BP execution creates a unique *case or trace*. Each trace

is composed of a succession of *events*, each of which represents the completion of a certain *process activity*. Each occurrence of the online-order requires the three following characteristics:

- **The trace identification:** refers to a unique reference related to a specific execution of the BP
- **The activity name:** indicates event name in the trace
- **The timestamp:** indicates the needed time to accomplish the execution of an activity
- **The performance attributes:** corresponds to the other additional features that characterize an activity

In the following, the formal definitions of the BP concepts are described through the online-order application.

Example 1.11. Table 1.1, below represents two traces (or cases) containing the event logs of the BP model of the e-commerce order management applications of figure 1.3 [2]. A trace in this table is characterized with the Id attributes (identifier case). In addition, each trace is composed of various events which are described with other properties, such as Activity name, Start time, and End time. Optionally, the trace can be enriched by other attributes, such as its textual description, the employee (who execute the operation) and other needed resources attributes. These attributes set is designated as performance parameters (in our case we have only the the employee (who execute the operation)). The actions inside the same trace are ordered and they constitute a control flow.

TABLE 1.1: An example of the online purchase order event-log data-set

CA	EA			PA
TID	Activity description	Begin time	Final Time	Employee
1	Prepare Order	01/01/2023 09:12	02/01/2023 00:00	Customer A
	Check Order	02/01/2023 09:12	03/01/2023 09:00	Order handling
	Confirmed Order	03/01/2023 09:12	03/01/2023 09:59	Customer A
	Order Accepted	03/01/2023 10:00	03/01/2023 10:00	Customer A
2	Prepare Order	02/01/2023 09:12	03/01/2023 00:00	Customer B
	Check Order	03/01/2023 09:12	04/01/2023 09:00	Order handling
	Prepare shipment	04/01/2023 09:12	05/01/2023 09:59	Shipment
	Order Accepted	05/01/2023 10:00	05/01/2023 10:00	Customer B

Let's now define formally the aforementioned concepts of events and traces:

Definition 1.3. (Event) An event is a tuple $(a, c, t, (d_1, v_1), \dots, (d_m, v_m))$ where a is the activity name, c is the trace ID, t is the timestamp and $(d_1, v_1), \dots, (d_m, v_m)$ (where) are the event or trace attributes and their values [35].

Example 1.12. Table 1.1, below shows a set of events and traces. The set of traces forms the event log data (in other word: the process variant or the execution trace). According to definition 1.3, a is the activity, c is the trace ID, t is the start time and $(A_1, V_1) \dots, (A_m, V_m)$ (where $m \geq 0$) are the event or trace attributes and their values. According to table 1.1, an example of event is: (Prepare order, 1, 01/01/2023 9:12, (Completion time, 02/01/2023 00:00), (The employee, Customer A)).

A trace or case is the sequence of events produced by the execution of a particular procedure. Over the time, the various executions of the activities of the deployed BP, capture generated events reflecting historical execution traces. These executions lead to the formation of database containing the historical data or execution traces (*The event-log data*). The formal definition of a trace is the following.

Definition 1.4. (Trace) A trace $\mathcal{E}$ is a non-empty sequence $\alpha = [e_1, \dots, e_n]$ of events, where $\forall i \in [1 \dots n], e_i \in \mathcal{E}$, and $\forall j, j \in [1 \dots n] e_i.c = e_j.c$. In other words, all events in the trace refer to the same trace [35].

Example 1.13. In table 1, the first trace (Trace ID = 1) stores an event-log composed with a set of the following data:

- (Prepare order, 1, 01/01/2023 9:12, (Completion time, 02/01/2023 00:00), (The employee, Customer A)),
- (Check Order, 1, 02/01/2023 9:12, (Completion time, 03/01/2023 00:00), (The employee, Order handling)),
- (Confirmed Order, 1, 03/01/2023 9:12, (Completion time, 03/01/2023 9:59), (The employee, Customer A)),
- (Order Accepted, 1, 03/01/2023 10:00, (Completion time, 03/01/2023 10:00), (The employee, Customer A))

In this context, one of the substantial components of the BPM life cycle is the BP model. The next section aims to cover the BP modeling methodology, event-log data, and associated concepts.

1.3.2 Textual data

This type of BP data refer to the textual representation of BPs that are described in natural language. This data often includes descriptions of tasks, activities, decisions, roles, and the sequence of events that occur during a BP. Textual BP data can be

found in various sources such as policy documents, guidelines, emails, customer support conversations, or operational manuals, and it plays a crucial role in helping organizations define, monitor, and improve their processes. Below are some common types of textual BP data:

1. **Narrative Process Descriptions:** This is the simplest form of textual BP data, where a BP is described in natural language, often in a paragraph or list format. It is typically used for high-level communication and documentation purposes. for example: *When a customer places an order online, the sales team verifies the order, checks product availability, and confirms the purchase. If the item is in stock, it is shipped; otherwise, the customer is notified about the delay.*
2. **Process Flow Textual Descriptions:** These descriptions outline the flow of activities in a BP, often using conditional and branching statements. They are similar to flowcharts but written in text. for example: *If the payment is successful, the system will generate a receipt and send it to the customer. If the payment fails, the system will notify the customer to retry payment. After three failed attempts, the order is canceled.*
3. **Business Rules and Policies:** Business rules are specific guidelines or constraints that govern how certain processes should be executed. They are typically expressed in a conditional format or as guidelines for making decisions within the process. For example: *If the customer has an active membership, a 10% discount is applied.*

Textual BP data can take many forms, from simple narrative descriptions to structured data formats like JSON or XML. The type of textual data used depends on the audience (*human or machine*), the complexity of the process, and the need for automation or analysis. Businesses often rely on different types of textual BP data to ensure processes are effectively communicated, executed, and optimized over time.

The following section aims to describe the various BP models and their motivations because of the benefits of their that were retrieved from those numerous types of BP data.

1.4 BP modeling

BP models are explicit representations of an organization's processes, detailing the sequence of activities, events, and decisions that contribute to achieving specific objectives. They provide an abstract but formalized way to describe, analyze, and improve

processes, serving as a bridge between business and technology. This section aims to explain the motivation of the BP modeling and the types of the existing BP models.

1.4.1 Motivation of BP modeling

BP modeling is an essential practice in organizations that strive to improve their operations, optimize resource utilization, and enhance decision-making. The motivation for BP modeling stems from the need to understand, document, analyze, and improve organizational workflows. Below are key motivations behind BP modeling:

1. **The company adaptation:** Nowadays, the company's adaptation of the actual process, by adding new procedures, removing updating the content of the previous once. In fact, businesses must recognize that they cannot always forecast what will occur next. As a result, they must continuously supervise the market and improve their business logic as needed.
2. **The business integration and collaboration:** The business integration and collaboration include the creation of common relationships, both internally and externally, in order to achieve goals or solve issues by sharing diverse skill sets, talents, and viewpoints. Advanced digital solutions enable company cooperation between staff and clients in order to increase production and communication in a unified environment.
3. **Process automation:** By modeling processes, organizations can identify repetitive and manual tasks that can be automated using technologies like Robotic Process Automation (RPA), Business Process Management Systems (BPMS), or AI-driven tools. This increases efficiency and reduces manual effort.
4. **Decision making:** BP modeling allows organizations to identify inefficiencies, redundancies, and bottlenecks in existing workflows. Further, by analyzing the BP models, they can predict and recommend activities as well as optimizing their processes. Those improvements can be made to: *reduce costs, minimize time delays, enhance resource allocation.*

1.4.2 The categorization of the BP models

Understanding the various types of BP models and tools available helps organizations to choose the right approach for their needs. Whether using simple flowcharts, sophisticated BPMN models, or data-driven process mining tools, BP modeling facilitates better decision-making, process optimization, and overall business efficiency.

The following categorization of BP models highlights their characteristics based on whether they are *graphical*, *formal*, or *language*-based representations:

1.4.2.1 Graphical models

These models use diagrams and visual elements to represent processes, making them intuitive and accessible to both technical and non-technical stakeholders.

1. **Workflow models:** A workflow model [38] represents the sequence of tasks or activities performed within a process. It captures dependencies, transitions, and the flow of tasks based on the EL data. For instance, the following figure 1.2 describes workflow model discovered from the EL data described in the table 1.1.

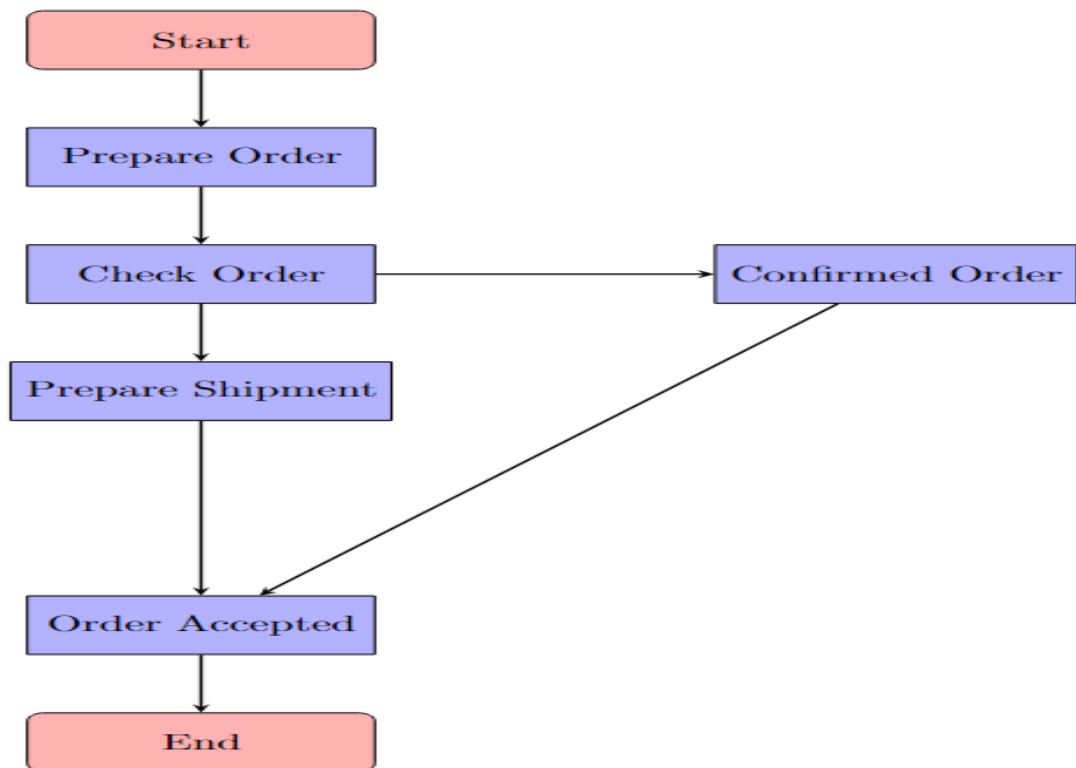


FIGURE 1.2: The workflow model of the online purchase order event-log data-set

Before the modelisation step, the analysis of the EL data described in the table 1.1 concludes the following rules:

- **Prepare Order:** The preparation of a customer's order begins and lasts until the order is ready for review.
- **Check Order:** The order undergoes review by the order-handling team for verification.

- **Confirmed Order:** Once reviewed, the order is confirmed by the customer.
- **Prepare Shipment:** Shipment preparation follows after confirmation for Customer B.
- **Order Accepted:** The final activity when the order is officially accepted.

This workflow reflects common e-commerce processes, showing how different stages (*e.g., shipment preparation*) may vary depending on customer-specific factors. The diagram clearly captures sequential and optional paths. For example:

- Shipment preparation occurs only for specific orders.
- The workflow ensures that all orders are eventually accepted before completion.

2. **Business Process Model and Notation (BPMN):** BPMN [39] is the most commonly used standard for modeling BPs with a focus on simplicity and collaboration. It aims to create standardized, easy-to-understand diagrams for all stakeholders.

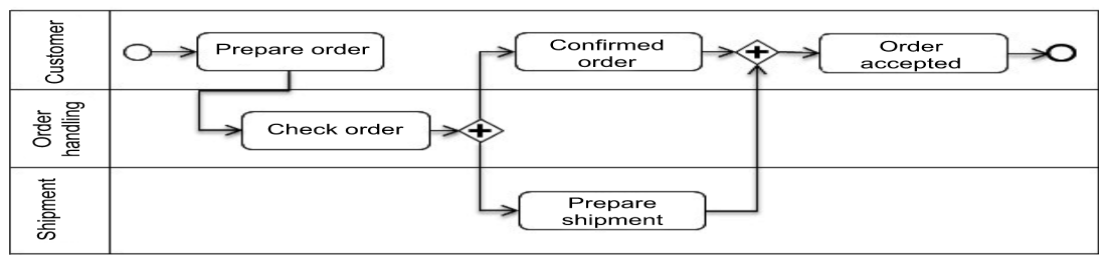


FIGURE 1.3: BPMN of online order management application’s [2]

Figure 1.3 [2] depicts a BPMN language corresponding to an online BP order management [2]. In the figure, each *node* represents a step of the delivery product procedure, while *relation* between nodes (*Prepare order, check order ...etc*) corresponds to activities (*actions*) to be accomplished to pass through nodes.

1.4.2.2 Formal models

Formal models use mathematical and logical frameworks to define processes rigorously. These models are essential for verification, validation, and simulation.

1. **Petri Nets:** A Petri Net [38] is a graphical representation used to describe and analyze processes, especially workflows and concurrent systems. It consists of three main components:

- **Places (circles):** Represent conditions or states in the system.

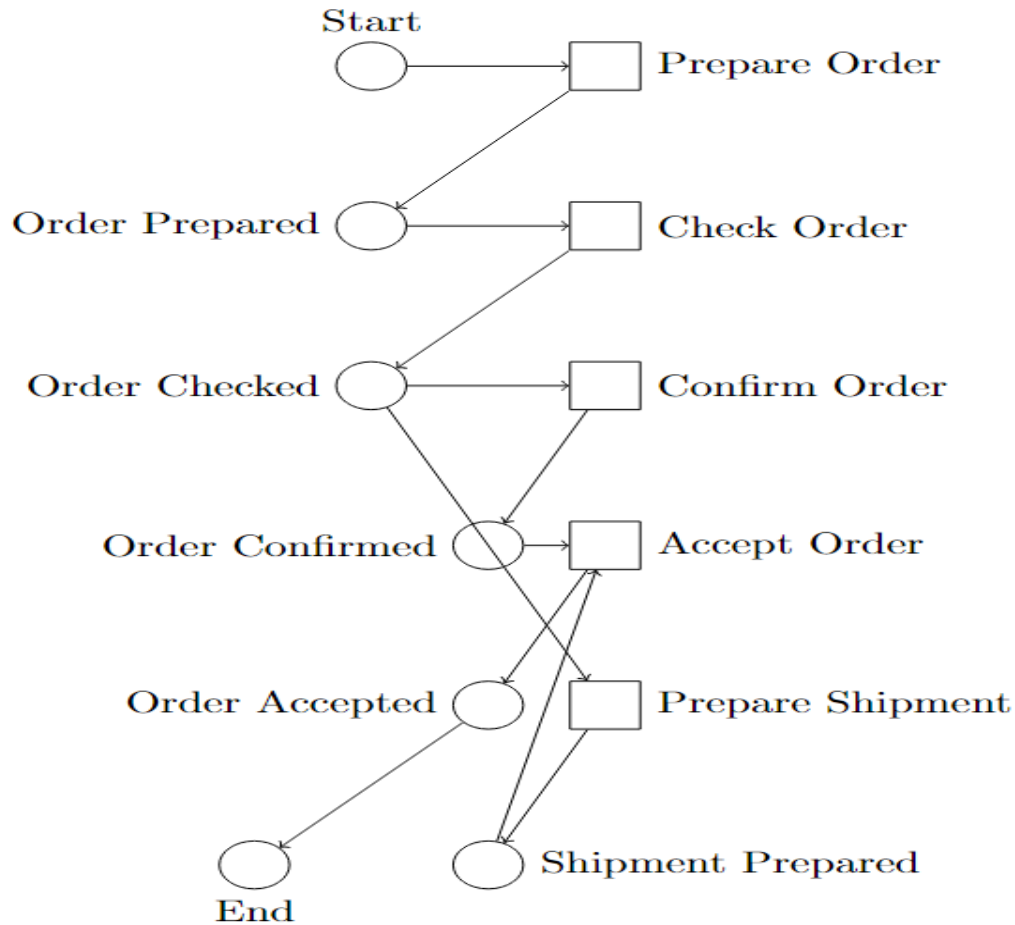


FIGURE 1.4: The petri net of online order management application's

- **Transitions (rectangles):** Represent events or activities that may change the state of the system.
- **Arcs (arrows):** Connect places to transitions or transitions to places, indicating the flow of control or resources.

For instance, the given EL described in the table 1.1 describes two transactions (trace ID 1 and trace ID 2) for an online purchase order process. The figure 1.4 represents a Petri Net model to represent the flow of these activities.

The Petri Net illustrates the sequential and concurrent flows in the online purchase order process. It helps in visualizing dependencies, bottlenecks, and parallel tasks within the process. This representation is especially useful for analyzing and optimizing BPs.

2. **Automata models:** An automata model [40] is a formal representation of the sequence of events or activities within a system or process. For an event log, automata can be used to model the flow of activities as transitions between states. Each activity represents a state change in the automata.

Based on the given event log dataset of the table 1.1, we can construct an automaton for the online POP. As shown in the figure 1.5, each distinct activity in the event log (*Prepare Order*, *Check Order*, *Confirmed Order*, etc.) corresponds to a state or transition in the automaton. The automaton starts at an initial state and ends in a final state, traversing through states based on the sequence of activities in the log.

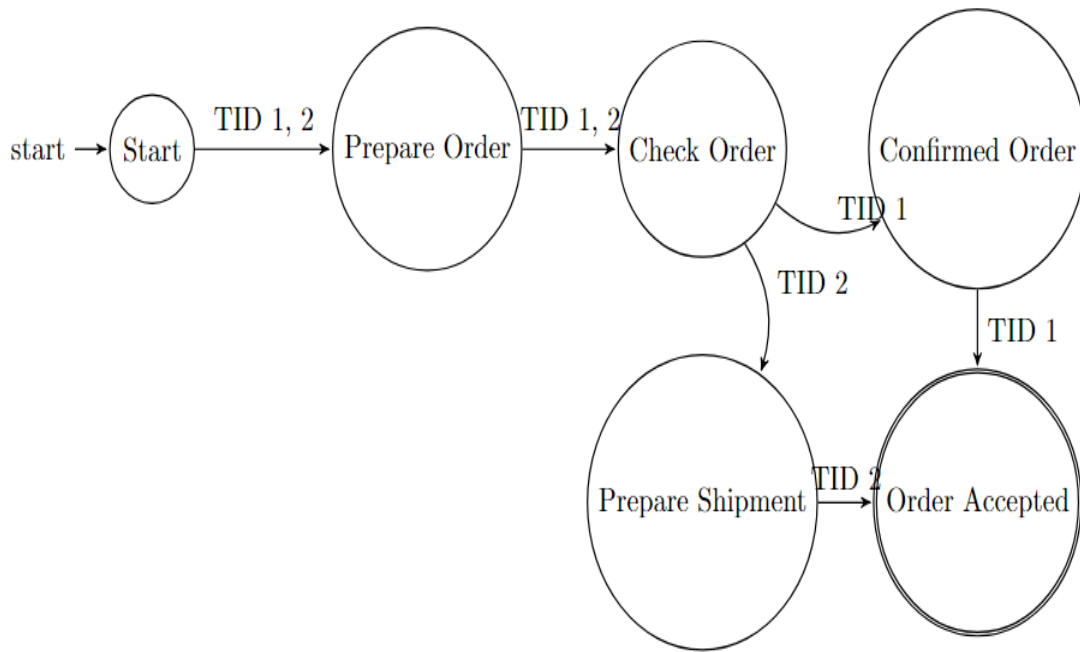


FIGURE 1.5: The automata model of online order management application's

According to the figure 1.5, the following explanations are made:

- **Start State:** The automaton begins with Start.
- **Prepare Order:** Both TID 1 and TID 2 transition to this state.
- **Check Order:** The next state for both traces is Check Order.
- **Divergence:**
 - **TID 1:** Proceeds to Confirmed Order and then Order Accepted.
 - **TID 2:** Proceeds to Prepare Shipment and then Order Accepted.
- **Final State:** Both traces end at Order Accepted.

1.4.2.3 Language BP models

The language BP models are concerned with the textual BP data that is stored in a structured, machine-readable format like XML, JSON, or YAML. This language BP models can be parsed by automated systems. This type of models is commonly used for

BP automation or integration with software tools. for example according to the table 1.1: "process-name": "Case 01",

```
"Activities": [
  "activity": 1, "action": "Prepare order", "actor": "Customer A",
  "activity": 2, "action": "Check order", "actor": "Order handling",
  "activity": 3, "action": "Confirmed order", "actor": "Customer A"
  "activity": 4, "action": "Order accepted", "actor": "Customer A"
]
```

While the advantages of these BP models, they suffer from the complexity and interoperability in the complex and big input execution data. In order to optimize and gain from the complexity for a decision-making goal, the conventional PM and AI-powered approaches appears. In this context, the next sections targets to present the conventional PM approaches used in the BP area.

1.5 BP mining approaches

Process Mining (PM) is is an active research field that combines BPM with data analytic techniques to derive meaningful insights from process execution data [41]. Its approaches aim to support different stages of the BPM life cycle, such as process discovery, analysis, and monitoring [40, 42]. In fact, it targets to identify, monitor, and improve real-world processes by extracting knowledge from event logs [43], which are widely available in today's information systems. More precisely; PM combines BPM with data analytic methods to extract valuable insights from process execution data [44]. The underlying methods are intended to assist several BPM life cycle stages [45], including process discovery, analysis, and monitoring [46]. By extracting information from event logs, which are commonly available in modern information systems, it actually aims to detect, monitor, and enhance real-world processes [47]. More typically:

Definition 1.5. (Process Mining) PM is a collection of methodologies that integrate the disciplines of data science with business process management to support in the analysis of operational business processes through event logs [35].

As depicted in table 1.2 [3], according to the given input and output, the PM ecosystem are being divided into three complementing aspects.

- Without utilizing any prior information, the initial component is discovering BP models in an event log.

TABLE 1.2: Comparison of different types of PM approaches

PM types	Process discovery approaches	Conformance checking approaches	Process enhancement approaches
Attributes	- Log file exploration - Extracting abstract behavioral specifications	- Comparing the event-log and abstract models	- Improvement of the actual BP model
Input	Event-Log data	Event-Log + abstract model	Actual abstract model
Output	Abstract model	Precision and fitness	Enhanced abstract model

- Through executing comparison between the input event log data and the target abstract BP model, the second step enables the conformity checking of BP's (*extracted from an event log or developed manually from scratch*). As a result, the conformance checking test if the reality and its discovered model are compliant and vise versa.
- The third kind is connected to BP model enhancement and it attempts to enrich current process models by using execution information recorded in the event log, or to discover the defects and abnormalities brought by conformance checking.

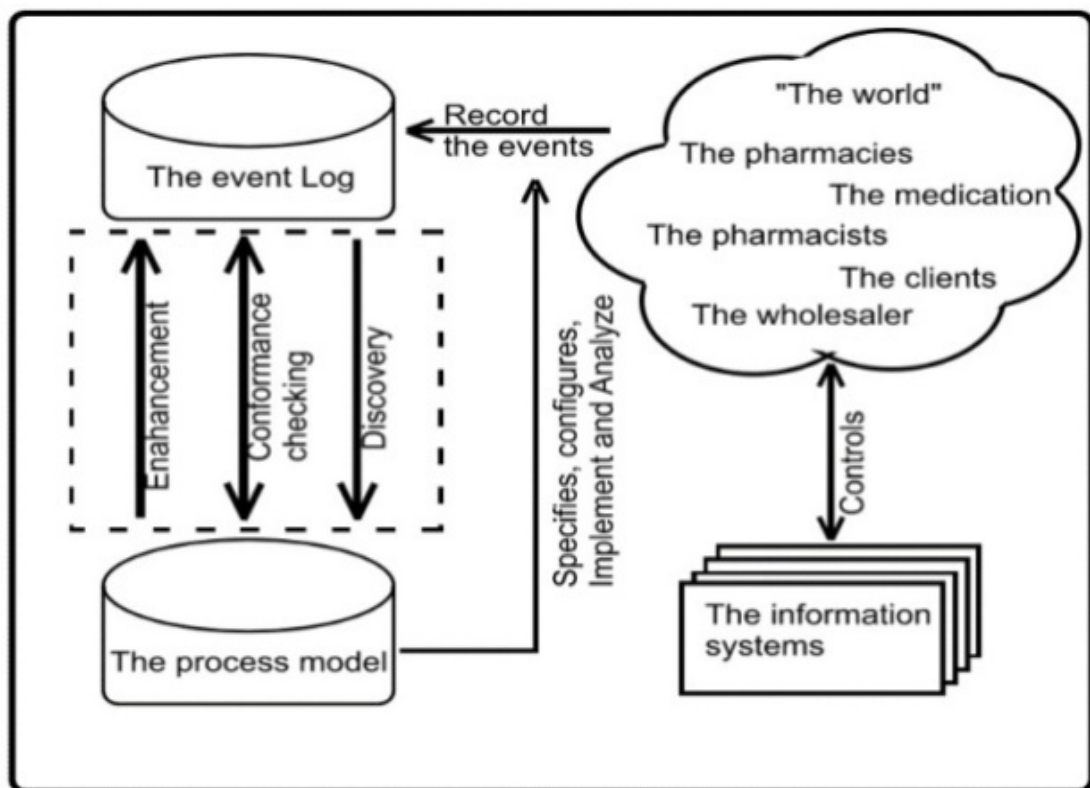


FIGURE 1.6: The relationships between the PM approaches [3]

In this context, this section aim to explain the three types of the BP mining approaches and their impact in the BP decision-making process.

1.5.1 Process discovery approaches

As shown in the figure 1.6, process discovery [48] is the foundational step in process mining. It involves generating a BP model solely based on event logs, without any pre-existing process model or knowledge of the process [49–52]. This approach provides an abstract model that reflects the real-world behavior captured in the event log. The behavior basically depends from the assumption and constraints wanted to attempt by this modeling step.

In this context, multiple methods were developed; the following is a list of the most commonly used conventional approaches:

1. **Alpha-Algorithm:** The Alpha-algorithm [38] is one of the earliest and most fundamental algorithms in process mining, designed to create process models from event logs. It works by identifying causal relationships between activities in the event log, such as determining whether one activity precedes another or whether two activities occur concurrently. Based on these relationships, the algorithm constructs a Petri net, which represents the underlying business process. The Alpha-algorithm is straightforward in its approach and highly effective for structured processes with minimal noise or variations. However, its simplicity can be a drawback in practical applications. It struggles to handle real-world event logs that are often incomplete, noisy, or exhibit complex control-flow constructs such as loops and long-term dependencies. Despite its limitations, the Alpha-algorithm remains a foundational technique, forming the basis for more advanced algorithms in process discovery.
2. **Heuristic Miner:** The Heuristic Miner [53] builds upon the Alpha-algorithm by incorporating heuristic rules to deal with the inherent noise and incompleteness often found in real-world event logs. Instead of relying solely on strict causal relationships, it evaluates the frequency and likelihood of connections between activities to identify patterns that represent the underlying process. By tolerating minor deviations and anomalies in the data, the Heuristic Miner can generate models that are more robust and realistic than those produced by the Alpha-algorithm. This makes it particularly suitable for processes with loops, parallelisms, and infrequent behaviors. However, the heuristic approach introduces a trade-off: while it handles noisy data effectively, the resulting models may lack interpretability and precision when applied to highly complex datasets. Nevertheless, the Heuristic Miner remains a popular choice for discovering process models in scenarios where data quality is less than ideal.

3. **Inductive Miner:** The Inductive Miner [53] is a more advanced algorithm that addresses many of the shortcomings of earlier techniques like the Alpha-algorithm and Heuristic Miner. It uses a divide-and-conquer strategy to construct process models by recursively splitting event logs into smaller parts, which makes it capable of handling both structured and unstructured processes. The algorithm ensures that the generated models are sound, meaning they are free of errors such as deadlocks or unreachable states, which is a significant advantage over other methods. Moreover, the Inductive Miner is highly effective in handling noise and variability in event logs, producing models that closely reflect the real-world process. However, this level of robustness comes at a computational cost. The algorithm can be resource-intensive, particularly for large dataset or highly complex processes. Despite this, its ability to create accurate and reliable models makes the Inductive Miner a preferred choice for organizations dealing with diverse and noisy data.

Process discovery enables organizations to uncover hidden workflows, identify inefficiencies, and gain a better understanding of their operations. Decision-makers can use these insights to optimize workflows, improve resource allocation, and reduce bottlenecks. However, the previously mentioned limitations of each approach, particularly the complexity of the resulting model, make it impossible to be used alone without the additional automatic decisional systems *explained in the next chapter*.

1.5.2 Conformance checking approaches

As shown in the figure 1.6, conformance checking [54] compares an event log with an existing (discovered or manually created) process model to identify deviations, inconsistencies, or compliance issues. This ensures that real-world behavior aligns with the intended or designed process model.

In this context, multiple methods were developed; the following is a list of the most commonly used conventional approaches:

1. **Token-Based Replay:** Token-Based Replay [55] is one of the foundational techniques used in conformance checking. It evaluates the alignment between an event log and a Petri net process model by simulating token flows through the model. For each activity recorded in the event log, the corresponding transition in the Petri net is fired, consuming and producing tokens in the associated places. This process reveals whether the event log adheres to the constraints of the process model.

If a transition in the event log does not align with the model, tokens might be missing or leftover, which indicates deviations or anomalies in the process execution. Metrics such as fitness (*how well the log fits the model*) and precision (*how well the model restricts behavior*) are often used to quantify the alignment.

The strength of this type of algorithm is easy to understand and visualize, making it an effective tool for identifying and explaining deviations. It is especially useful for providing clear, graphical representations of where the process execution diverges from the expected model. However, this approach struggles with complex or noisy event logs that involve loops, parallel executions, or unstructured behavior. It can also become computationally intensive when dealing with very large datasets or highly intricate process models.

2. **Alignments-Based Checking:** Alignments-Based Checking [56] improves upon Token-Based Replay by computing precise alignments between the event log and the process model. Instead of merely replaying tokens, this algorithm identifies the optimal sequence of moves that align the log with the model. These moves can be classified as:

- **Log moves:** Activities present in the event log but not allowed by the model.
- **Model moves:** Activities allowed by the model but missing from the log.
- **Synchronous moves:** Activities that perfectly align between the log and the model.

By comparing the sequence of moves, the algorithm provides a detailed diagnostic of conformance deviations. This precision makes Alignments-Based Checking particularly useful for identifying root causes of discrepancies and suggesting corrective measures.

This approach offers high accuracy and diagnostic capabilities, enabling detailed analysis of deviations and their causes. It is especially effective in situations where precise conformance analysis is required, such as compliance auditing or regulatory reporting. However, this type of approaches can be computationally expensive, particularly for large-scale logs or highly complex process models. The need for extensive computations to determine the optimal alignment sequence makes it less efficient for real-time or large-scale applications.

3. **Entropy-Based Conformance Checking:** Entropy-Based Conformance Checking [56] takes a probabilistic approach to measure deviations, focusing on stochastic and variable processes. This method uses probabilistic models to compute entropy (*a measure of randomness or uncertainty*) for different paths in the process model.

By comparing the entropy of the observed behavior (*from the event log*) to the expected behavior (*from the process model*), this algorithm identifies patterns of compliance or non-compliance.

This approach is particularly effective in environments with high variability, where traditional deterministic algorithms may fail. For example, in healthcare or customer service domains, where processes are flexible and rarely follow rigid workflows, Entropy-Based Checking can provide valuable insights into process compliance and performance.

Entropy-Based Conformance Checking is robust in handling variability and noise in event logs. It is well-suited for processes that are not strictly defined and for identifying trends or patterns in compliance over time. However, this method heavily depends on the quality of the event log. Inaccurate, incomplete, or poorly structured logs can lead to misleading results. Additionally, the probabilistic nature of the approach may require advanced statistical knowledge, making it less accessible for practitioners without expertise in stochastic modeling.

Conformance checking allows organizations to ensure compliance with regulations, policies, or standards. It helps identify process inefficiencies, risks, and areas for improvement, empowering decision-makers to enforce standards and reduce operational risks. However, as explained previously, the earlier algorithms had multiple shortcomings. In this circumstances, the need of the conventional decision making algorithms (*explained in the chapter 3*) and the AI-based approaches (*explained in the chapter 4*) for fulfilling a smart conformance checking mechanism is extremely prominent especially with the generative models that attempt to learn a small space in which the event log data instances can be well rebuilt.

1.5.3 Process enhancement approaches

As shown in the figure 1.6, process enhancement [57] focuses on improving existing process models by incorporating insights from event logs. It identifies areas of inefficiency, suggests improvements, and enriches models with real-world execution data.

In this context, multiple methods were developed; the following is a list of the most commonly used conventional approaches:

1. **Delta Analysis:** Delta Analysis [58] is an algorithmic approach used in process mining to compare an existing business process model with an event log, enabling the identification of discrepancies and opportunities for process improvement. By

analyzing the "delta" or differences between the ideal process model and the real-world process execution captured in the event log, Delta Analysis helps uncover inefficiencies, bottlenecks, and deviations from the intended workflow. It evaluates performance metrics such as execution times, task frequencies, and resource utilization, providing actionable insights for process enhancement.

Delta Analysis excels in directly identifying areas that require improvement within a business process. It provides focused insights into mismatches, allowing for targeted interventions to optimize specific aspects of a process. It is particularly useful in situations where organizations already have established process models but need to refine them based on real execution data. While effective, Delta Analysis often requires manual interpretation of the results. This reliance on human expertise can make the process time-consuming and subjective, especially in complex processes with numerous tasks and actors. Additionally, its efficacy is dependent on the accuracy and comprehensiveness of the event log data.

2. **Genetic Process Mining (GPM):** Genetic Process Mining (GPM) [59] is a sophisticated algorithm that applies genetic algorithms to process mining. It evolves and optimizes process models iteratively by mimicking the principles of natural selection. In this approach, initial models are treated as a population, and "fitness functions" are used to evaluate their alignment with the event log data. Over successive iterations, models are combined, mutated, and optimized to better represent the observed behavior in the event log.

GPM is highly effective for analyzing and optimizing complex processes with multiple variables, including unstructured workflows. It is particularly well-suited for situations where traditional deterministic algorithms, like the Alpha-algorithm, struggle to generate accurate models due to noise or high variability in the event logs. Its evolutionary approach ensures a continuous improvement of models, often resulting in more robust and realistic representations. However, one of the major challenges of GPM is its computational intensity. The algorithm requires significant processing power and time, especially when dealing with large datasets or highly complex processes. Additionally, GPM does not always guarantee convergence to an optimal model, as it depends on the quality of the initial population and the design of the fitness function.

3. **Stochastic Process Mining:** Stochastic Process Mining [60] focuses on enhancing business process models by incorporating probabilistic elements derived from execution data in event logs. Unlike deterministic approaches that assume fixed outcomes, stochastic models account for uncertainties and variabilities in processes, enabling organizations to make predictions about future behavior. These models

integrate probabilities into tasks, transitions, and decision points, reflecting real-world variability in workflows.

This approach is particularly useful for processes that exhibit a high degree of uncertainty or variability, such as those in dynamic industries like healthcare or logistics. By providing probabilistic predictions, Stochastic Process Mining helps decision-makers anticipate potential issues and plan resources more effectively. Its ability to reflect real-world complexity makes it ideal for risk analysis, capacity planning, and performance optimization. However, the effectiveness of Stochastic PM relies heavily on the quality of the event log data. Noisy, incomplete, or biased data can significantly reduce the accuracy of the probabilistic models. Additionally, interpreting the output of stochastic models often requires specialized expertise, which may not be readily available in all organizations.

Process enhancement enables organizations to optimize existing workflows, reduce costs, and improve customer satisfaction. It provides decision-makers with actionable insights to streamline operations and adapt to changing business environments. These environmental variables are handled in an enhanced model using techniques of process discovery and analysis of conformity, where many process shortcomings include flash experiences, process loops, and undesirable process differences was the target of various Conventional ML algorithms (*explained in the chapter 3*) AI-based BP mining approaches (*explained in the chapter 4*) for decision-making solutions.

By leveraging process discovery, conformance checking, and process enhancement, organizations can gain valuable insights into their operations, ensure compliance with standards, and optimize their business processes. The selection of appropriate algorithms for each type of process mining depends on the complexity of the data, the desired outcomes, and the available computational resources.

1.6 Conclusion

The business process domain is essential for the smooth and efficient operation of organizations. Hence, in two last decades, it becomes an inevitable concept for managing modern IS. Thus, understanding its key components and life cycle stages, along with the integration of technology, helps organizations optimize their processes, improve performance, and achieve their strategic goals. As technology continues to evolve, so too will the methods and tools available for business process management, paving the way for more efficient and effective organizational operations.

Chapter 2

Decision Systems through Knowledge Representation Integration

2.1 Introduction

In today's data-driven world, Decision Support Systems (DSS) play a crucial role in aiding complex decision-making across various domains, *including business, healthcare, and finance*. However, the effectiveness of these systems largely depends on how well they represent, structure, and utilize knowledge. Knowledge Representation (KR) provides a formalized way to encode, store, and retrieve information, making DSS more intelligent, interpretable, and adaptive. By integrating KR techniques (*such as logical, frame and graphical representations*), the DSS can move beyond *raw data processing* to *context-aware, knowledge-driven decision-making*. This integration enhances different types of DSS by enabling them to support more transparent, efficient, and automated decisions.

This chapter presents the application domains of the DSS systems including the BP area. Then, the types of the DSS are discussed and explained how the KR approaches are integrated into the DSS systems.

2.2 Definitions and application domains of DSS

DSS have become integral to modern business operations, providing tools and frameworks that assist decision-makers in analyzing data and making strategic choices. DSS

utilizes data, models, and structured algorithms to offer actionable insights and recommendations, supporting efficient, evidence-based decision-making. As business environments become increasingly complex, DSS helps to streamline decisions, optimize operations, and predict outcomes. This section explores the fundamental aspects of DSS, including its definition, application domains, and specific roles in BP domains.

2.2.1 Definitions

A DSS is an interactive computer-based tool designed to assist decision-makers in processing information, analyzing data, and making well-informed decisions. It combines data storage, sophisticated analytical models, and user-friendly interfaces to generate insights in response to specific queries or scenarios [61, 62]. DSS commonly employs conventional algorithms, statistical analysis, and sometimes machine learning techniques to process historical and real-time data. These approaches allow DSS to generate predictive insights, identify trends, and provide recommendations that guide decisions at tactical, operational, or strategic levels. At its core, DSS improves the quality, speed, and effectiveness of decisions, allowing organizations to navigate uncertainties and align actions with business objectives.

In the following the two definitions of DSS systems are presented.

Definition 2.1. *A DSS is a computer-based system that supports managers in decision-making activities by combining models, analytical tools, databases, and user interfaces to improve the effectiveness of decision-making in semi-structured or unstructured problems. [63]*

Definition 2.2. *A DSS is an interactive computer-based system designed to help decision-makers utilize data and models to solve unstructured or semi-structured problems, providing assistance in evaluating alternatives and making informed choices. [64]*

2.2.2 Application domains of DSS

DSS find applications across a broad range of industries and fields [65], each tailored to meet domain-specific requirements. Key application domains include:

- **Healthcare:** In healthcare, DSS assists in patient diagnosis, treatment planning, and healthcare management by analyzing patient data and treatment outcomes, which helps medical professionals make evidence-based decisions.

- **Finance:** Financial DSS models help in portfolio management, risk assessment, investment analysis, and regulatory compliance by processing large volumes of market and client data to provide insights and forecasts.
- **Supply Chain and Logistics:** DSS in supply chain management supports logistics optimization, inventory management, and demand forecasting, enabling better planning, reduced costs, and improved service delivery.
- **Manufacturing:** Manufacturing sectors utilize DSS to optimize production scheduling, quality control, and maintenance planning, enhancing operational efficiency and reducing down times.
- **Retail and Marketing:** In retail, DSS aids in understanding consumer behavior, sales forecasting, and inventory management, enabling companies to offer targeted promotions and enhance customer experience.

Each domain leverages DSS according to its data structures and decision requirements, but the core goal remains the same: to improve the decision-making process through systematic data analysis and predictive modeling.

2.3 Deploying DSS in BP domain

In the BP domain, the role of DSS is critical [66] in supporting efficient and optimized workflows by offering insights that align operational tasks with strategic objectives. DSS provides decision-makers with essential tools for process analysis, monitoring, and re-engineering, enabling better coordination across departments and functions. Its use within Business Process Management (BPM) ensures that processes remain agile, transparent, and responsive to changing conditions or business goals.

In what follows, we illustrate the usefulness of deploying DSS technologies in the BP area.

- **Process Analysis:** DSS enables businesses to analyze current processes in-depth, identify inefficiencies, and optimize workflows. By using data-driven insights, organizations can streamline activities, reduce redundancies, and ensure that each step adds value to the end-to-end process.

Example 2.1. *For instance, a large retailer like Walmart uses DSS to analyze inventory levels, customer demand patterns, and supplier lead times. By integrating real-time data from various stores, warehouses, and suppliers, the DSS*

identifies inefficiencies, such as overstocked items in one location and stock-outs in another. It then recommends optimized restocking schedules, supplier adjustments, and transportation routes to minimize costs and ensure timely product availability. This streamlines the supply chain, reduces redundancies like excess storage costs, and enhances customer satisfaction by ensuring products are consistently available when needed.

- **Monitoring and Control:** With DSS, businesses can monitor process performance through Key Performance Indicators (KPIs) and real-time data, allowing quick identification of bottlenecks or deviations. The system also aids in setting thresholds for alerts, enabling timely intervention and minimizing disruptions.

Example 2.2. *For instance, a retail company might employ a DSS to monitor stock levels in real-time using KPIs such as inventory turnover rate and order cycle time. If the system detects that inventory levels for a high-demand product are nearing a predefined threshold, it triggers an alert for restocking.*

- **Agility and Responsiveness:** DSS empowers businesses to adapt processes in response to evolving market conditions, customer expectations, and regulatory requirements. Its predictive capabilities support proactive decision-making, allowing companies to remain competitive and responsive to change.

Example 2.3. *For instance, during the holiday season, the DSS predicts high demand for certain products, enabling the retailer to proactively adjust stock levels across various locations. Additionally, the system provides recommendations on supplier selection, shipping schedules, and cost-effective logistics routes. This ensures that shelves remain stocked while minimizing overstocking and reducing operational costs, allowing the company to meet customer demands efficiently while maintaining profitability.*

By integrating DSS into the BP domain, organizations can create a structured, data-centric approach to process management, enhance operational efficiency, and drive continuous improvement across BPs. The following section explains the types of DSS algorithms.

2.4 Types of DSS

DSS can be categorized into several types based on their inputs and final goals, components. In the following the two types of DSS systems are explained.

2.4.1 Types of DSS based input data

As shown in the Fig.2.1 [4], four types of input data could classify the DSS systems. The following is an overview of the main types of DSS.

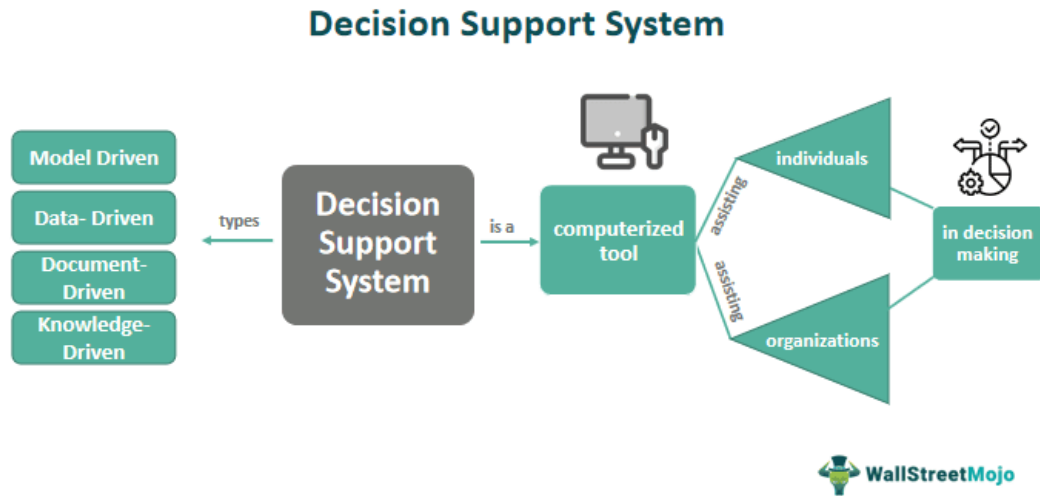


FIGURE 2.1: The types of DSS systems [4]

2.4.1.1 Data-Driven DSS

Data-driven DSS [67] primarily focuses on the analysis of large volumes of data to support decision-making. These systems utilize databases and data warehouses to store and retrieve data for analysis. The characteristics of this type of DSS are the following:

- Emphasizes data collection, storage, and retrieval.
- Often employs tools like data mining, reporting, and querying.
- Suitable for organizations that require data analysis and reporting.

DSS are systems designed to assist decision-makers by providing relevant data, analytics, and insights from different sources. These systems rely heavily on data to guide decisions, support strategic choices, and enhance the operational efficiency of business processes.

Example 2.4. Below are some examples of how Data-Driven DSS can be applied to Business Processes:

- A data-driven DSS can analyze the performance of different marketing campaigns. The system collects data from various marketing channels such as social media,

email marketing, website analytics, and customer interactions. It can generate reports like customer engagement, conversion rates, and sales revenue. By analyzing this data, the system identifies patterns and provides insights into which marketing strategies are effective and which are not, helping marketing managers optimize their campaigns.

- *Data-driven DSS in financial institutions can assess creditworthiness by analyzing historical transaction data, credit histories, income data, and other financial information. Using machine learning algorithms, the system can calculate the probability of default for an applicant, allowing financial institutions to make informed decisions about loan approvals. The system can also track changes in economic conditions, providing real-time insights on market risk factors and advising on risk mitigation strategies.*

2.4.1.2 Model-Driven DSS

Model-driven DSS [68] utilizes mathematical and statistical models to analyze data and simulate different scenarios. These systems focus on using models to evaluate complex decision-making processes. The characteristics of this type of DSS are the following:

- Incorporates various models, such as optimization, simulation, and forecasting.
- Allows decision-makers to perform “what-if” analyses.
- Useful for strategic planning and financial forecasting.

Model-driven Decision Support Systems (DSS) in the context of business processes (BP) are systems that rely on a model of business operations, rules, and procedures to provide insights or recommendations for decision-making. These systems leverage analytical models, data, and sometimes simulation to support business decisions.

Example 2.5. *Below are some examples where Model-Driven DSS can be applied to enhance business processes in various industries:*

- *The DSS uses predictive models, such as logistic regression or decision trees, to predict the likelihood of a customer defaulting on a loan based on historical data. It integrates financial history, current liabilities, and other financial indicators into a comprehensive credit risk model.*
- *The DSS uses models of supply chain logistics, inventory turnover rates, demand forecasts, and warehouse capacities. These models predict future inventory requirements and provide recommendations for restocking and order fulfillment.*

2.4.1.3 Knowledge-Driven DSS

Knowledge-driven DSS [69], leverage specialized knowledge and rules to provide recommendations and insights. They emulate the reasoning of human experts in specific domains. The characteristics of this type of DSS are the following:

- Utilizes a knowledge base and inference engine.
- Offers explanations for recommendations and decisions.
- Ideal for complex problem-solving in areas like medicine, engineering, and finance.

A Knowledge-Driven Decision Support System (KDSS) is a type of decision support system that provides users with knowledge-based advice, recommendations, or insights to assist in the decision-making process. Unlike data-driven systems that primarily focus on collecting and analyzing raw data, knowledge-driven DSS leverage expertise, experience, and best practices to guide decisions. These systems rely heavily on a knowledge base consisting of rules, procedures, models, and heuristic knowledge.

In the context of BP, KDSS can support various operational, tactical, and strategic decisions.

Example 2.6. *Here are some examples of how knowledge-driven decision support systems can be applied in BP:*

- *A knowledge-based system can assist in predicting demand for products, suggesting inventory levels, and optimizing stock levels to avoid overstocking or under stocking. The system uses a combination of historical sales data, expert knowledge about product life cycles, seasonality trends, and market demand to generate forecasts. It can then recommend optimal inventory quantities and reorder points for products.*
- *Investment firms can use a knowledge-driven DSS to make better investment decisions by analyzing past market trends, financial reports, and expert opinions. The system incorporates expert knowledge on economic indicators, market behavior, and financial risk models. It can suggest the best investment strategies (e.g., stocks, bonds, mutual funds) based on the user's goals and risk appetite.*

2.4.1.4 Document-Driven DSS

Document-driven DSS [62] is a type of DSS that focuses on the management, retrieval, and analysis of unstructured or semi-structured documents to support decision-making

processes. These systems help users access and leverage vast repositories of textual, graphical, or multimedia data to make informed decisions. In the following we describe the key features of the Document-Driven DSS.

- Facilitates document management and retrieval.
- Includes features for version control and collaborative editing.
- Provides advanced search capabilities to locate documents based on keywords, metadata, or content.
- Utilizes indexing and categorization to improve accessibility.
- Combines document data with other types of structured data from databases, facilitating comprehensive analysis.
- Supports the integration of multimedia content like videos, images, or audio.
- Includes tools for text mining, natural language processing (NLP), and sentiment analysis to extract meaningful patterns from text-based data.
- Enables visualization and summarization of complex information.

Document-Driven DSS are systems that rely heavily on documents, such as reports, presentations, spreadsheets, and other data formats, to aid decision-making in business processes. These systems focus on managing, organizing, and presenting relevant information from documents in a way that supports business decisions. They are particularly useful when decisions are based on detailed information that is stored in various forms of unstructured or semi-structured documents.

Example 2.7. *Here are a few examples of how document-driven DSS can be applied in BP:*

- *In many organizations, the finance department is responsible for managing and processing invoices. This involves comparing invoices against purchase orders, delivery receipts, and contractual agreements, which are often stored in document form. For example, a Document-Driven DSS could be used to automate invoice approval by scanning invoices, purchase orders, and delivery receipts to ensure that all information matches. The system can generate recommendations such as whether an invoice should be approved, flagged for further review, or rejected.*
- *Financial institutions must analyze risk by reviewing documents such as financial statements, market analysis reports, and credit reports. A Document-Driven*

DSS in this context helps financial analysts make decisions by processing these documents to identify potential risks or investment opportunities. For example, a financial institution may deploy a document-driven DSS to assess credit risks by analyzing credit reports, loan applications, and market conditions. The system could flag loans that are likely to default or suggest risk mitigation actions, such as higher interest rates for higher-risk clients.

2.4.2 Types of DSS based on target goal

DSS can be categorized based on the type of decisions they assist with. Broadly, DSS can support operational decisions and structural decisions, each serving distinct functions in an organization.

2.4.2.1 Operational decisions

Operational decisions are short-term and routine, often requiring real-time or near-real-time analysis of data. These decisions typically involve day-to-day business operations and are often repetitive in nature. Examples include:

- Inventory management (*e.g., restocking levels based on demand predictions*).
- Production scheduling and workflow optimization.
- Customer relationship management (*e.g., automated support ticket routing*).

A DSS for operational decisions often integrates real-time data streams, rule-based logic, and predictive analytics.

2.4.2.2 Structural decisions

Structural decisions (*or strategic decisions*) are long-term, high-level choices that shape an organization's direction. These decisions require extensive data analysis, scenario planning, and often involve significant uncertainty. Examples include:

- Market expansion and entry strategies.
- Capital investment decisions (*e.g., facility construction, technology adoption*).
- Policy and regulatory compliance strategies.

DSS supporting structural decisions typically employ simulation models, multi-criteria decision analysis (MCDA), and machine learning-driven strategic planning tools. These systems are designed to synthesize large volumes of historical and projected data to facilitate evidence-based strategic decision-making.

Each type of DSS relies on different Knowledge Representation (KR) techniques to structure, analyze, and retrieve knowledge efficiently. *Knowledge graphs, ontologies, and rule-based systems* play crucial roles in enhancing DSS by improving interpretability, automation, and inference capabilities. While this KR is the first step in the AI technique (*explained in the following chapter 3*), is substantial step to ensure an effective decision making. In this context, the following section aims to present the KR stage.

2.5 Knowledge representation

Knowledge Representation (KR) [70] describes how systems store, arrange, and use information regarding complex problems. It is an essential component of AI that allows machines to simulate human comprehension and reasoning. Creating data structures and models that effectively capture information about the environment and make it available and usable by AI algorithms for learning, inference, and decision-making is known as knowledge representation. In the following two definitions of the KR in AI area are given.

Definition 2.3. *KR refers to strategies for encoding information about the environment into a format that an AI system can comprehend and utilize to make choices. It is an essential component of AI that bridges the gap between raw data and intelligent thinking. AI systems may use organized knowledge to understand data, form inferences, and solve problems.* [70]

Definition 2.4. *KR the act of structuring and organizing knowledge so that AI systems can process and apply it to reasoning and decision-making.* [71]

Before going in depth through the KR, the following section aims to describe the input knowledge data and their types.

2.5.1 The description and types of knowledge

The purpose of the following section is to describe the types of knowledge and the pyramid of data into knowledge.

2.5.1.1 The knowledge pyramid

The Data, Information, Knowledge, Wisdom (DIKW) pyramid [72] is a widely used depiction of the links between data, information, knowledge, and wisdom. It is also known as the *knowledge pyramid*, *wisdom hierarchy*, and *information hierarchy*. As shown in the figure 2.2, the knowledge pyramid includes strictly defined building pieces, much like other hierarchy models: *data comes first, followed by information, knowledge, and wisdom at the summit*.

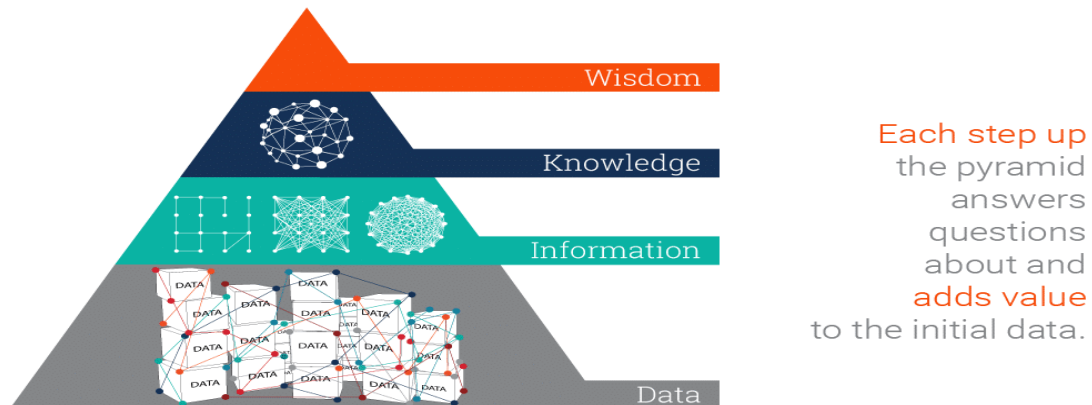


FIGURE 2.2: The knowledge pyramid [5]

Every level of the pyramid adds value to the original data and provides answers to inquiries about it. We climb the pyramid higher the more questions we can answer. To put it another way, we get more information and insights from our data the more we add context and meaning to it.

1. **Data:** Data is a collection of facts in a raw or unorganized form such as numbers or characters. However, without context, data can mean little.

Example 2.8. For example, 12012012 is just a sequence of numbers without apparent importance. But if we view it in the context of ‘this is a date’, we can easily recognize 12th of January, 2012. By adding context and value to the numbers, they now have more meaning.

In this way, we have transformed the raw sequence of numbers into *information*.

2. **Information:** Information (*knowing knowledge-that*) is the next building block of the DIKW pyramid. This is data that has been ‘cleaned’ of errors and further processed in a way that makes it easier to measure, visualize and analyze for a specific purpose. By asking relevant questions about ‘who’, ‘what’, ‘when’, ‘where’, etc., we can derive valuable information from the data.

Example 2.9. *For example, personal know-that have been given a basic explication in terms of justified, true, belief (Plato 370 BCE). A person knows that p if, and only if*

- *p is true fact,*
- *The person believes p ,*
- *The person is justified in her belief of p*

3. **Knowledge:** A unit of knowledge can be defined as a piece of information that allows users to reach an outcome when confronted with this specific questions: "**How** are the pieces of this information connected to other pieces to add more meaning and value?"

Example 2.10. *For example, the analysis of public knowledge, impersonal knowledge, would be:*

- *p is true,*
- *p is accepted by the community,*
- *The community has evidence or justification for p .*

When we don't just view information as a description of collected facts, but also understand how to apply it to achieve our goals, we turn it into knowledge.

4. **Wisdom:** Wisdom is the top of the DIKW hierarchy and to get there, we must answer questions such as '*why do something*' and '*what is best*'. In other words, wisdom is knowledge applied in action. We can also say that, if data and information are like a look back to the past, knowledge and wisdom are associated with what we do now and what we want to achieve in the future.

Example 2.11. *For example, the wise person must not only have wide appropriate knowledge, but they must act in accordance with the knowledge they have. They need to use their knowledge when required, and not to ignore it by choice or chance.*

2.5.1.2 Types of knowledge

The knowledge in AI is classified into numerous groups based on how it is used and expressed [6]. Understanding these categories helps in properly organizing knowledge for reasoning and decision-making. The following subsections are the basic forms of knowledge in AI:

- **Declarative knowledge:** This category of knowledge pertains to facts or assertions that characterize the world, frequently expressed as *"knowing what."* It is static and does not need any activities or procedures.

Example 2.12. *"Annaba is the state of Algeria" is a declarative statement.*

- **Procedural knowledge:** This type explains *"how"* to accomplish tasks or solve issues. It is dynamic and action-oriented, with a focus on processes and approaches.

Example 2.13. *Procedural knowledge includes topics like knowing how to solve a mathematical equation.*

- **Structural knowledge:** This form of knowledge is concerned with the relationships between items or concepts, and it organizes knowledge into structures such as hierarchies and networks.

Example 2.14. *Understanding that a dog is a type of animal and a poodle is a particular kind of dog demonstrates structural knowledge.*

- **Meta knowledge:** Meta-knowledge is knowledge about knowledge, specifically knowing which sorts of knowledge to use in particular contexts. Meta-knowledge is critical for systems that need to adapt or improve their performance since it aids in the selection of the best strategy or knowledge base for a particular situation.

Example 2.15. *Meta-knowledge in a medical diagnostic system is understanding which symptoms to prioritize when diagnosing a condition.*

- **Heuristic knowledge:** This sort of knowledge comprises logical thinking, expert estimations, and intuitive judgments based on experience. Frequently used in problem-solving and decision-making processes where precise answers are not possible, AI systems can arrive at good enough solutions quickly.

Example 2.16. *A heuristic might be "If a website loads slowly, refresh the page" - it is not always guaranteed to work, but often useful.*

The fact that KR describes how the previous knowledge can be structured in order to be processed by the machine, the next subsequent sections aim to describe the cycle of this KR and different existing categories that can structure the previous types of data.

2.5.2 The cycle of KR

The cycle of knowledge representation [73] in AI is the cyclical process by which AI systems perceive, learn, represent, and apply knowledge to make informed decisions. This

cycle is critical for developing intelligent systems capable of reasoning and interacting with their surroundings. As shown in the figure 2.3, the primary stages of this cycle are as follows:

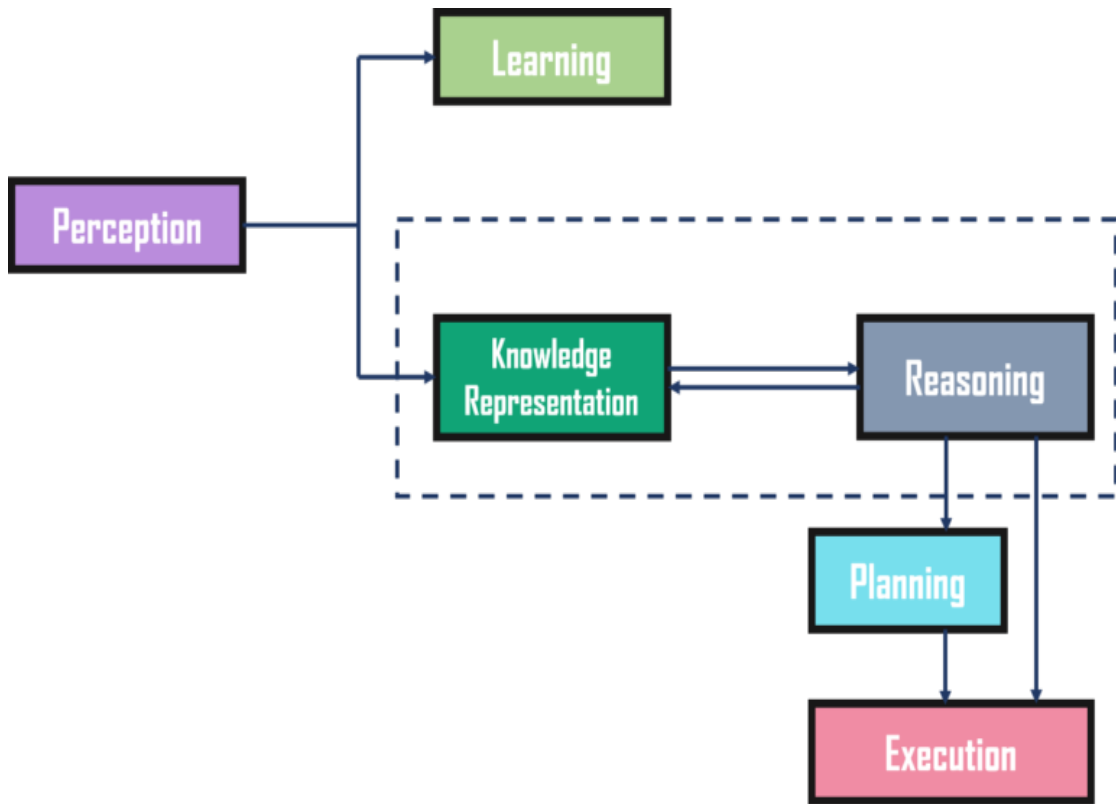


FIGURE 2.3: The cycle of KR in AI [6]

1. **Perception:** The process begins with AI systems perceiving data from their environment. This data could come from sensors, cameras, user inputs, or databases.
2. **Learning:** After perceiving data, the system learns by identifying patterns, relationships, and trends from the data. This can involve supervised learning, unsupervised learning, or reinforcement learning techniques.
3. **KR and reasoning:** In this stage, the system organizes the learned data into structured knowledge, allowing it to reason and draw conclusions. This is where the AI uses knowledge representation techniques to encode the information it has learned.
4. **Planning:** The system uses the represented knowledge to plan actions based on goals and constraints. Planning involves selecting the best course of action based on the available knowledge.

5. **Execution:** Finally, the system executes the planned actions, completing the cycle. After execution, the AI system can receive feedback and adjust its knowledge and actions accordingly.

Every stage of this cycle influences the next, and it remains continuous and adaptable. As AI systems progress through these phases, they continuously improve their knowledge and decision-making skills to become more clever and efficient. The KR step is performed through a set of approaches described in the following subsequent section to effectually ensure the DSS.

2.6 Integration of KR approaches for an effective DSS system

AI systems may represent knowledge using a variety of methods, each with unique applications and advantages. These methods aid AI systems in organizing and reasoning about the knowledge they learn. Some of the most important techniques for knowledge representation for enhancing the DSS are listed below:

2.6.1 Logical representation

Logical representation [73] involves using formal logic systems like *propositional and Logic of Predicate (LP0, LP1)* [74] to represent knowledge in a structured, precise, and unambiguous way.

Example 2.17. *In an expert system for medical diagnosis, logical statements like “If a patient has a fever and a cough, then they might have the flu” can be represented as rules using **propositional logic or predicate logic**.*

- *In Propositional Logic (LP0), we can represent the statement as:*

$$(F \wedge C) \rightarrow Flu$$

where:

- *F: The patient has a fever.*
- *C: The patient has a cough.*
- *Flu: The patient has the flu.*

- In First-Order Logic (LP1), we can represent the statement as:

$$\forall x (Fever(x) \wedge Cough(x) \rightarrow Flu(x))$$

where:

- x : Represents a patient.
- $Fever(x)$: x has a fever.
- $Cough(x)$: x has a cough.
- $Flu(x)$: x has the flu.

This logical representations are precise and allow for powerful inference mechanisms. However, they can be rigid and may not handle uncertainty or incomplete information well.

2.6.2 Frames representation

Frames [73] represent knowledge in structured “frames” or templates, each containing information about an object or situation. Each frame holds slots that store attributes or related information.

Example 2.18. A frame for a car might include slots for attributes such as make, model, color, and owner. This structure allows for easy organization of knowledge about similar objects. Below is an example of how a frame for a car can be represented:

The example above represents a car frame with the following attributes:

- **Make:** The manufacturer of the car, e.g., Toyota.
- **Model:** The model of the car, e.g., Camry.
- **Color:** The color of the car, e.g., Blue.
- **Owner:** The name of the car’s owner, e.g., John Doe.

Frames are effective for representing structured, hierarchical knowledge and allow easy modification or extension of attributes. however, they are limited by the flexibility when dealing with highly dynamic or unpredictable information.

2.6.3 Graphical representation

The graph representations [75] is an organized structures of real-world entities and their relationships. It is typically stored in a graph database, which stores the relationships between data entities. **Entities** in a graph can represent *objects, events, situations, or concepts*. **The relationships between these entities** capture the *context and meaning of how they are connected*. In this context, the key characteristics of the graph are the following:

1. **Nodes:** denote and store details about entities, such as *people, places, objects, or institutions*. Each node has a (*or sometimes several*) label to identify the node type and may optionally have one or more properties (*attributes*). Nodes are also sometimes called *vertices*.

Example 2.19. For example, as shown in the figure 2.4, the nodes in an e-commerce graph typically represent entities such as people (customers and prospects), products, and orders:



FIGURE 2.4: The nodes of the e-commerce graph representation [7]

2. **Relationships link two nodes together:** they show how the entities are related. Like nodes, each relationship has a label identifying the relationship type and may optionally have one or more properties. Relationships are also sometimes called edges.



FIGURE 2.5: The relationship of the e-commerce graph representation [7]

Example 2.20. As shown in the figure 2.5, in the e-commerce example, relationships exist between the customer and order nodes, capturing the “placed order” relationship between customers and their orders:

Based on the previous graph representation, two big advances in the KR approaches was developed untitled **semantic graphs** and **knowledge graphs**. While they are closely related in their *graphical representation of information*, they differ in *their purpose, structure, and applications*. Below is a detailed comparison of the two concepts, with examples to illustrate their differences.

1. **Semantic graph:** emphasizes *meanings and relationships between concepts, often used to model semantics and contextual relationships in a domain* [8]. Its main purpose is to describe concepts, their features, and their interconnections in a way that enables understanding or reasoning about the domain. The relationships can be hierarchical, associative, or other types, and are typically labeled with semantic roles, such as “is-a” or “part-of.”

Example 2.21. As shown in the figure 2.6, the characteristics of the semantic graph that models concepts and their relationships (e.g., the hierarchy of the Mammal animals) are:

- (a) Nodes represent concepts (e.g., "Cat," "Bear").
- (b) Edges represent semantic relationships (e.g., "Is-a," "has" and "'lives-in").
- (c) Focuses on describing the structure or meaning of a domain.

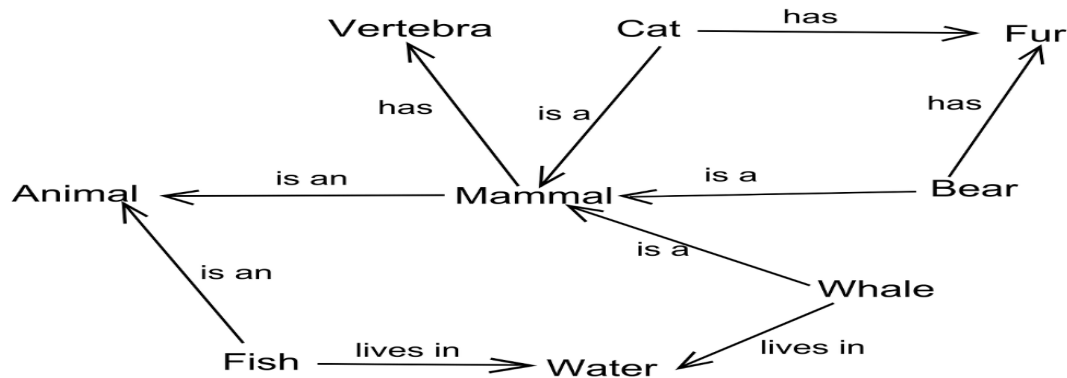


FIGURE 2.6: The semantic graph of the Mammal animals [8]

2. **Knowledge graph:** emphasizes *facts, entities, and relationships, often used to store real-world data for reasoning, querying, and answering questions* [8, 9]. Knowledge graphs are designed to represent and interconnect factual data to enable automated reasoning and querying over large datasets.

Example 2.22. As shown in the figure 2.7, the characteristics of the knowledge graph that represents factual data about organizing products by types and categories are:

- (a) Nodes represent real-world entities (e.g., "Fish," "Fruit").
- (b) Edges represent factual relationships (e.g., "Type-of").
- (c) Captures specific data and relationships for reasoning and querying.

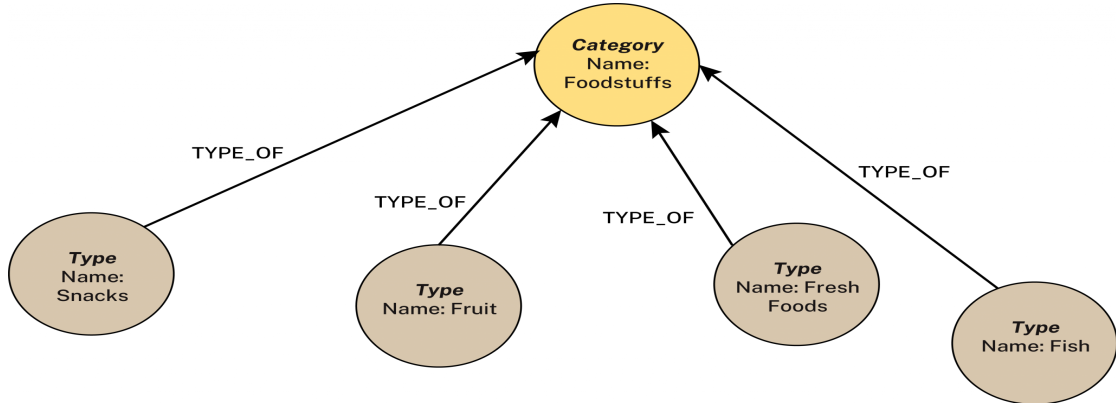


FIGURE 2.7: The relationship of the e-commerce graph representation [9]

3. Comparison between semantic and Knowledge graphs: As shown in the table 2.1, semantic models and knowledge graphs are both crucial in data representation and management, but they serve different purposes and operate at different levels of abstraction.

- A semantic model is a high-level representation that defines the basic meaning of data items and the relationships among them, essentially acting as a dictionary for a domain. It captures the business terms, expected relationships, and types of data that might be associated with these terms. They provide a structured and formal way to define the entities and relationships within a specific domain, ensuring a shared understanding of the data and its meanings.
- On the other hand, a knowledge graph is a more concrete and specific implementation that uses the definitions and structures to represent and link real-world entities and their relationships. Knowledge graphs go beyond simple data storage by focusing on the connections and context between entities, making them powerful tools for data integration, analytics, and sharing.

In this context, the knowledge graphs are a more advanced form of a semantic networks. However, as shown in the figure 2.8, they are typically more expressive and allow for the representation of more complex relationships, including properties and attributes of the entities. This complexity of the knowledge graph makes them *a powerful and rich representation* for ML, Artificial Neural Nets (ANN), and DL models.

Aspect	Semantic Graph	Knowledge Graph
Focus	Emphasizes semantic relationships and meanings.	Emphasizes factual relationships and structured knowledge.
Purpose	Typically used for categorization, classification, and deriving meaning.	Used for query answering, reasoning, and decision-making.
Nodes	Represent concepts, entities, and their semantics (meanings).	Represent entities (real-world things) and their properties.
Edges	Define relationships based on meanings or features.	Define factual and logical relationships between entities.
Features/Attributes	Often explicitly represented as separate feature nodes.	Stored as properties/attributes directly on entities.

TABLE 2.1: Comparison between Semantic Graph and Knowledge Graph.

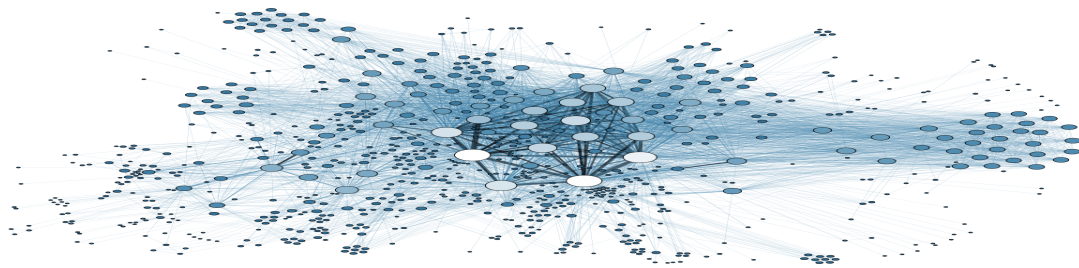


FIGURE 2.8: An example of a complex semantic graph of social media [10]

In summary, DSS are designed to assist decision-makers in analyzing complex problems, synthesizing information, and generating actionable insights. The effectiveness of a DSS is largely determined by how well it represents, organizes, and processes knowledge to support reasoning and decision-making. KR plays a crucial role in different types of DSS by structuring data, enabling inference mechanisms, and improving the interpretability of decisions.

2.7 Conclusion

The integration of KR into DSS has significantly enhanced decision-making by enabling systems to store, structure, and infer knowledge in a more intelligent and interpretable manner. By leveraging techniques (*such as logical, frame and graphical representations*), KR allows DSS to move beyond simple data processing to *context-aware, knowledge-driven decision-making*. Different types of DSS (*model-driven, data-driven, knowledge-driven, and hybrid systems*) benefit from KR by improving *efficiency, accuracy, and*

transparency in decision processes across domains such as business, healthcare, and finance. However, challenges remain in terms of *scalability, explainability, and integration with AI models*, requiring further advancements in KR techniques. Moving forward, the combination of **neural networks, deep learning, and large language models** will continue to refine decision support capabilities, paving the way for *more adaptive, explainable, and autonomous systems*.

This chapter has provided a comprehensive exploration of how KR empowers DSS, setting the foundation for future research and implementation in intelligent decision-making systems. The next chapter presents the AI techniques and their underlying KR models used for an effective decision making process.

Chapter 3

Artificial Intelligence techniques

3.1 Introduction

Artificial intelligence (AI) has become an integral part of modern society, driving advancements in various domains such as e-commerce, social networks, recommendation systems, smart homes, and healthcare. The path to this widespread adoption has been nonlinear, marked by both groundbreaking successes and significant setbacks. Since the introduction of the Turing test, AI research has experienced alternating periods of rapid progress and stagnation, often influencing funding availability and public perception. However, recent breakthroughs, particularly in deep learning, have revolutionized the field, making AI techniques more accessible and widely integrated into everyday applications.

This chapter provides a structured overview of modern AI techniques, with a primary focus on deep learning (DL) algorithms and their applications, particularly in chat-bot systems. Unlike conventional AI methods such as: *expert systems, probabilistic and heuristic models, and many more* [76–79], DL leverages artificial neural networks to achieve remarkable performance in complex tasks. While traditional AI techniques remain valuable in areas requiring explainability and logical reasoning, the dominance of connection methods has reshaped the landscape of AI research and application.

The chapter is organized as follows: The first section introduces artificial neural networks, laying the foundation for understanding deep learning methodologies. We then explore various deep learning algorithms, highlighting their capabilities and impact across different domains. Finally, we examine chat-bot systems, with a particular emphasis on the popular Chat-GPT model, showcasing its role in advancing natural language understanding and human-computer interaction.

3.2 Machine Learning techniques

As shown in the figure 3.1, "**ML**" refers to Machines Learning from data without the need for explicit programming. ML algorithms use statistical approaches to spot patterns and generate predictions or judgments based on training data. While ML is a subset of AI, the word was coined to underline the importance of data-driven learning and computers' capacity to improve their performance when given relevant data.

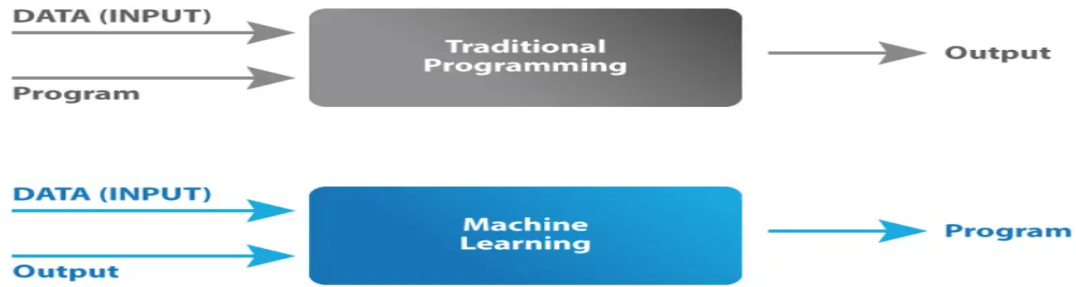


FIGURE 3.1: The difference of ML and traditional programming [11]

In the following, some of the key terminologies are introduced in this section.

1. **ML model** is a mathematical representation of a real-world process.

Example 3.1. *A linear regression model that predicts house prices based on square footage. For example: $\text{Price} = 50,000 + 200 \times \text{Square}$ Here, the model represents the relationship between square and price.*

2. **A feature** is a measurable property or parameter of the dataset.

Example 3.2. *In a dataset predicting house prices: Square footage, number of bedrooms, number of bathrooms, and proximity to a school are features.*

3. **A feature vector** is a set of multiple numeric features used as input to the machine learning model.

Example 3.3. *For a single house in a dataset predicting prices: Feature Vector = [2000, 3, 2, 0.5]*

- 2000: square footage,
- 3: number of bedrooms,
- 2: number of bathrooms,
- 0.5: proximity to a school in kilometers.

4. **Training** is the process where an algorithm learns patterns from input data (training data) to make predictions.

Example 3.4. *A neural network is trained on a dataset of handwritten digits (e.g., MNIST). The input images of digits and their corresponding labels (e.g., 0, 1, 2, etc.) are used to train the model to recognize and predict digits from new images.*

5. **Prediction** using the trained model to predict the target value for unseen data.

Example 3.5. *A trained house price prediction model is given a feature vector [1500,3,2,1]. The model predicts the house price as 220,000 dollar.*

6. **Over-fitting** When a model learns from noise or inaccuracies in the training data and performs poorly on new data.

Example 3.6. *A decision tree trained on a dataset predicts perfectly for the training data but performs poorly on a test dataset because it memorized the training examples rather than generalizing patterns.*

7. **Under-fitting** When a model fails to learn the underlying trends in the data, leading to low accuracy on both training and test data.

Example 3.7. *Fitting a straight line to non-linear data in a regression task. For example, using a linear model to predict a dataset shaped like a parabola will result in under-fitting.*

The Artificial Neural Network (ANN) are the advanced type of ML algorithms described in the following section.

3.3 Artificial neural networks

An ANN is an advanced algorithm in ML that mimics the structure and function of the human brain. It consists of interconnected nodes or neurons organized into layers, including an input layer, one or more hidden layers, and an output layer. Each neuron processes information from the input data, applies an activation function, and passes the output to the next layer. This layered architecture enables ANNs to learn and model complex, non-linear relationships in data, making them highly effective for tasks such as image recognition, natural language processing, and predictive analytics. ANNs are trained using large datasets, allowing them to improve their accuracy over time and make intelligent decisions with minimal human intervention.

This section provides the definitions and major components of the ANN. Forwarded by its learning process.

3.3.1 Definitions of ANN

ANN are computational models inspired by the structure and function of biological neural networks in animal brains. They are composed of interconnected nodes or artificial neurons that process information. Here are some formal definitions:

Definition 3.1. *ANNs are massively parallel systems with large numbers of interconnected simple processors. They are designed to learn from input data and predict outcomes for new data without being explicitly programmed. [80]*

Definition 3.2. *An artificial neural network is a machine learning algorithm based on the concept of a human neuron. The purpose of this review is to explain the fundamental concepts of artificial neural networks. [81]*

Definition 3.3. *ANNs represent a computational paradigm that takes inspiration from the basic workings of the brain to solve complex problems. They are not programmed but rather learn to perform tasks. [82]*

These definitions highlight the core concepts of ANNs, emphasizing their ability to learn from data and perform tasks without explicit programming, inspired by biological neural networks.

3.3.2 Key components of ANN

The key components of an artificial neural network include the following aspects:

3.3.2.1 Perceptron

The perceptron is a fundamental concept in ANN and ML. It is a linear classifier that uses a weighted sum of inputs and an activation function to produce an output.

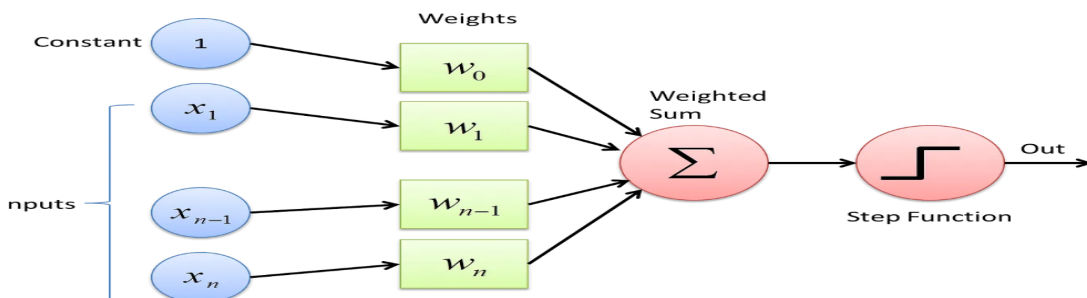


FIGURE 3.2: The perceptron mechanism[12]

As shown in the figure 3.2, the basic mathematical function of a perceptron can be described as follows [12]:

1. **Inputs of the perceptron:** Given inputs $(x_1, x_2, \dots, x_n)$ and corresponding weights $(w_1, w_2, \dots, w_n)$, the perceptron computes the weighted sum as described in the following function:

$$z = \sum_{i=0}^n w_i x_i \quad (3.1)$$

where (w_0) is often referred to as the bias term.

2. **Outputs of the perceptron:** The output of the perceptron is determined by applying *an activation function* ($\mathcal{F}$) to the weighted sum (z).
3. **The activation function:** The role of this activation function is used to map the input between the required values like $(0, 1)$ or $(-1, 1)$. A common choice for the *activation function* is the *step function*, which outputs 1 if (z) is greater than a certain threshold (θ) and 0 otherwise, as shown in the following:

$$y = f(z) = \begin{cases} 1 & \text{if } z > \theta \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

Additionally to this linear activation function, the set of *non-linear ones are described in Table 3.1*. The advantages and disadvantages of each function is summarized in the following table 3.2.

The perceptron algorithm adjusts the weights (w_i) during training to minimize classification errors (*through the loss function and backward propagation*), typically using a learning rule such as the perceptron learning rule or gradient descent methods.

3.3.2.2 ANN layers

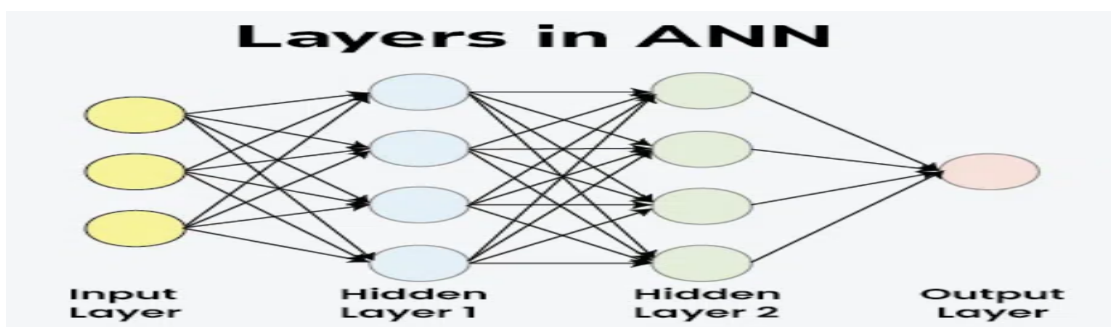


FIGURE 3.3: The ANN layers [13]

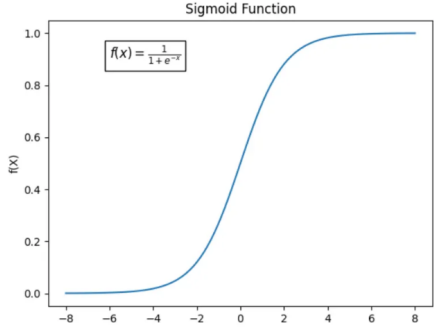
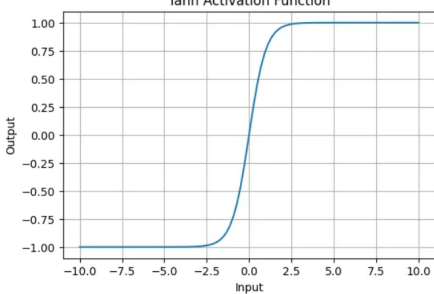
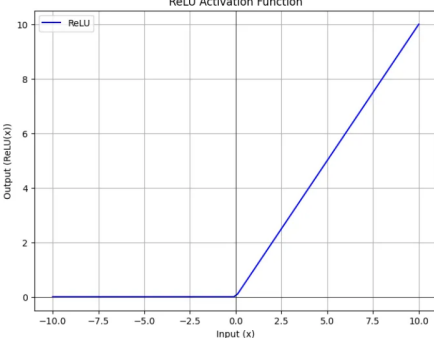
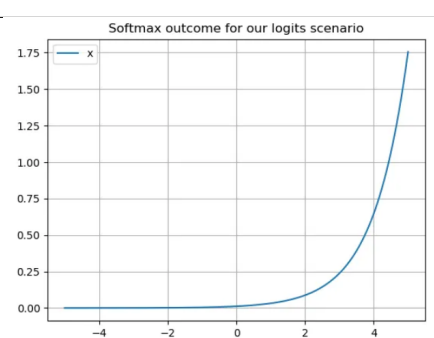
Name	Function	Derivative	Figure
Sigmoid	$\sigma(z) = \frac{1}{1+e^{-z}}$	$f'(z) = f(z)(1 - f(z))^2$	 The plot shows the Sigmoid Function, which is an S-shaped curve. The x-axis ranges from -8 to 8, and the y-axis ranges from 0.0 to 1.0. The curve starts near 0 for negative x, passes through 0.5 at x=0, and approaches 1 for positive x. A legend indicates the formula $f(x) = \frac{1}{1+e^{-x}}$.
tanh	$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$f'(z) = 1 - f(z)^2$	 The plot shows the Tanh Activation Function, which is an S-shaped curve ranging from -1.00 to 1.00. The x-axis ranges from -10.0 to 10.0, and the y-axis ranges from -1.00 to 1.00. The curve passes through 0 at x=0 and approaches -1 for negative x and 1 for positive x.
ReLU	$f(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{if } z \geq 0 \end{cases}$	$f'(z) = \begin{cases} 0 & \text{if } z < 0 \\ 1 & \text{if } z \geq 0 \end{cases}$	 The plot shows the ReLU Activation Function, which is a piecewise linear function. The x-axis ranges from -10.0 to 10.0, and the y-axis ranges from 0 to 10. The function is 0 for x < 0 and increases linearly with a slope of 1 for x ≥ 0.
Softmax	$f(z) = \frac{e^z}{\sum_i e^z}$	$f'(z) = \frac{e^z}{\sum_i e^z} - \frac{(e^z)^2}{(\sum_i e^z)^2}$	 The plot shows the Softmax outcome for our logits scenario. The x-axis ranges from -4 to 4, and the y-axis ranges from 0.00 to 1.75. The curve starts near 0 for negative x and increases rapidly for positive x, approaching 1.75 at x=4.

TABLE 3.1: Non-linear activation functions [24]

As shown in the figure 3.3, ANN is structured into *layers*, including *the input layer*, *one or more hidden layers*, and *the output layer*. Each layer plays a specific role in processing and transforming data. The layers are explained in the following [13]:

1. **Input Layer:** This layer consists of nodes that represent the features of the input data. Each node corresponds to a feature, and its value represents the feature's

TABLE 3.2: Advantages and Disadvantages of Activation Functions [24]

Ac-tiva-tion Func-tion	Advantages	Disadvantages
Sig-moid	- Produces outputs between 0 and 1, suitable for binary classification tasks. - Outputs are independent and not constrained to sum to one.	- Non-zero centered, which can cause gradients to go in different directions during back-propagation. - Prone to the vanishing gradient problem, especially in deep networks. - Computationally expensive compared to ReLU.
Tanh	- Zero-centered output range (-1 to 1), which can help with gradient flow. - Better than sigmoid for handling positive and negative values.	- Still prone to the vanishing gradient problem. - Outputs are not constrained to sum to one.
ReLU	- Computationally efficient and faster training. - Helps overcome the vanishing gradient problem. - Sparse activation, leading to fewer vanishing gradients.	- Non-zero centered. - Non-differentiable at 0, although the gradient is 1 for positive inputs. - Can lead to "dead neurons" where the neuron stops responding to variations in input.
Soft-max	- Produces a probability distribution over multiple classes, suitable for multi-class classification. - Outputs sum to one, making it ideal for tasks requiring normalized probabilities.	- Not suitable for binary classification tasks. - Can be computationally expensive for large networks. - Outputs are interdependent, which can affect training dynamics.

value. The input layer does not perform any computation; it simply passes the input data to the next layer.

2. **Hidden Layers:** These layers are positioned between the input and output layers. Hidden layers perform complex computations on the input data. Each neuron in a hidden layer receives inputs from all neurons in the previous layer, applies a weighted sum, adds a bias term, and passes the result through an activation function. This process allows the network to learn and extract features from the input data.
3. **Output Layer:** This layer produces the final prediction or result based on the computations performed by the hidden layers. The number of neurons in this layer depends on the nature of the problem. For example, in a binary classification task, there might be one neuron for each class, outputting probabilities.

3.3.3 ANN learning process

As shown in the figure 3.4, a neural network's learning (*training*) process is iterative in nature, with computations performed forward and backward through each layer of the network until the loss function is reduced.

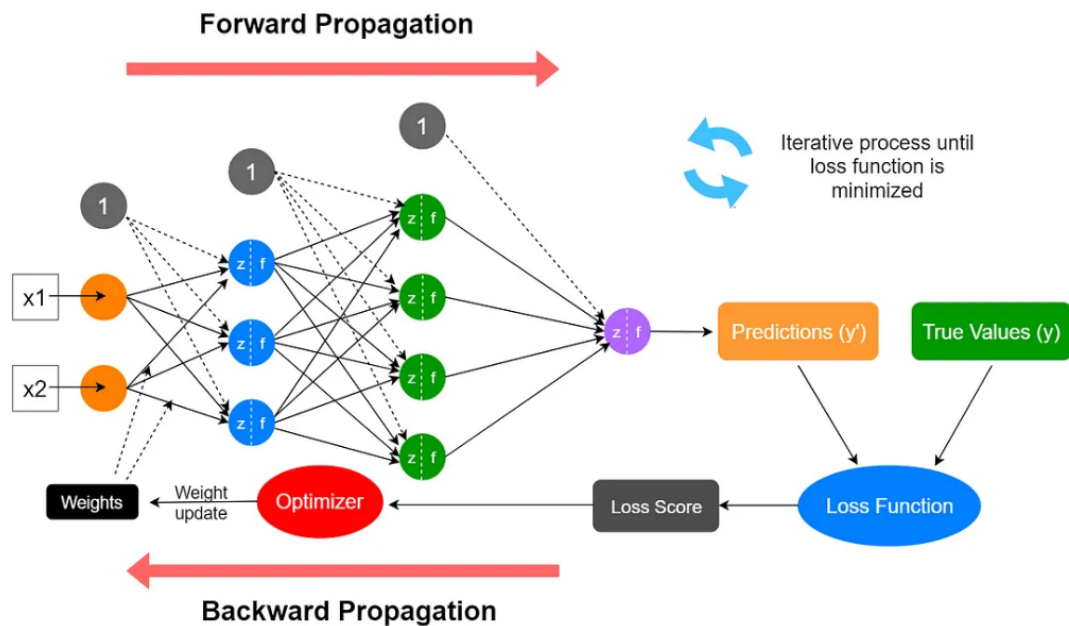


FIGURE 3.4: The ANN learning process [14]

In the following the detailed explanations about the each step from the learning process of the ANN [83].

3.3.3.1 Forward propagation

Forward propagation in neural networks is closely related to the concept of a perceptron. A perceptron is a fundamental unit in artificial neural networks, and forward propagation involves passing data through these units to generate a prediction. This process involves passing data through each layer of the network, starting from the input layer and moving towards the output layer. At each layer, the input data is transformed using a combination of weights, biases, and an activation function.

In the context of forward propagation, the perceptron is one of the simplest forms of a neural network layer. The process of forward propagation in a neural network involves passing the input data through multiple layers of perceptrons, where each layer performs the same operations of weighted sum and activation function explained in the subsection 3.3.2.1. This allows the network to learn increasingly complex features from the input

data. Thus, the perceptron serves as a building block for more complex neural network architectures, and the forward propagation process is essentially the repeated application of the perceptron's operations across multiple layers.

3.3.3.2 Calculation of the loss function

As shown in the figure 3.4, after the forward step the ANN measures the efficacy of ANN by measuring the loss function. The calculation of a loss function involves quantifying the difference between the predictions made by a machine learning model and the actual target values. Different types of loss functions are used depending on whether the task is a regression or a classification problem.

1. **Mean Squared Error (MSE):** This loss function is used for regression tasks and calculates the average of the squared differences between the predicted and actual values [84].
2. **Mean Absolute Error (MAE):** This loss function is also used for regression tasks and calculates the average of the absolute differences between the predicted and actual values [84].
3. **Binary Cross-Entropy Loss (Log Loss):** This is used for binary classification tasks and quantifies the difference between the predicted probabilities and the actual labels [84].
4. **Cross-Entropy Loss:** This is used for multi-class classification tasks and measures the performance of a classification model whose output is a probability value between 0 and 1 [84].

3.3.3.3 Backward propagation

After calculating the loss function, Backward propagation, often referred to as back-propagation is processed. It is a method used to train ANN by adjusting weights and biases to minimize the loss function. As shown in the figure 3.5, it works by computing the *gradient of the loss function* [85] with respect to the weights using the chain rule, and then *updating the weights in the opposite direction of the gradient to reduce the loss*. This process is repeated for each layer of the network, starting from the output layer and moving backward to the input layer, hence the term “back-propagation.”

Deep Learning (DL) extends the concept of ANNs by adding more hidden layers, creating a “deep” network. This architecture allows the network to learn and represent more

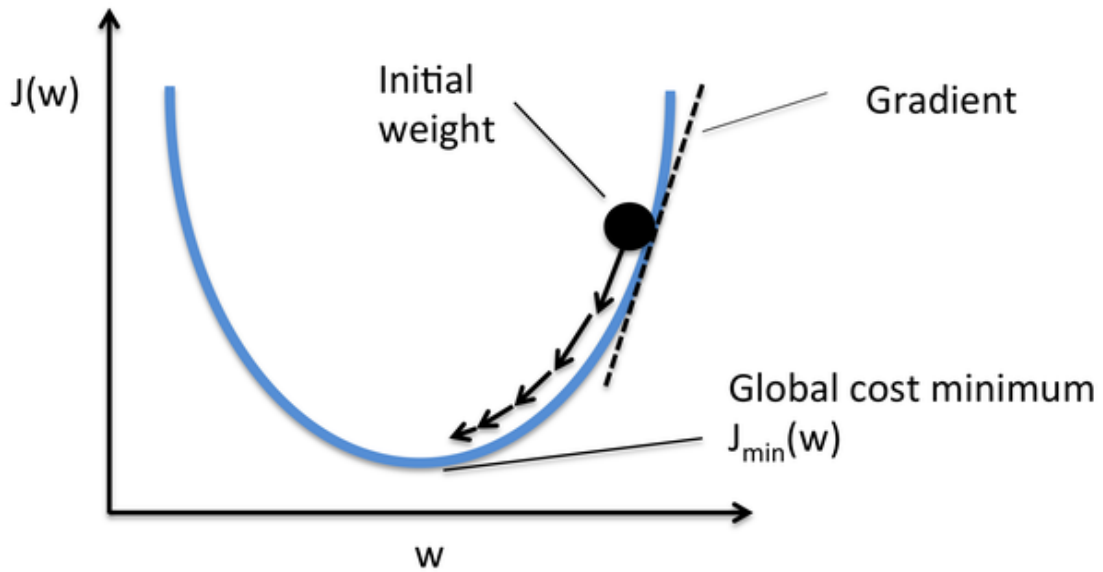


FIGURE 3.5: The curve of minimizing the error through the gradient descent function [15]

complex features and patterns from data. The additional layers enable the network to capture hierarchical representations, which are essential for tasks such as image and speech recognition. The following section aims to explain the variety of DL models used by our contributions.

3.4 The deep learning models

This section aims to presents the definitions of the DL models, followed by its difference with the ML in general and ANN in particular. Finally, a set of discriminative and generative models are presented.

3.4.1 Definitions

DL builds upon the principles of ANNs by incorporating deeper architectures and specialized layers to handle complex and high-dimensional data, leading to significant advancements in various fields such as computer vision, natural language processing, and more. The following definitions highlight the core aspects of DL, including its ability to learn from data independently, its use of multiple layers of processing units, and its capability to handle complex tasks through hierarchical learning.

Definition 3.4. *DL is a type of technology that allows computers to simulate how our brains work. Specifically, it is a method that teaches computers to learn and make decisions independently, without explicitly programming them. Instead of telling a computer exactly what to look for, we show it many examples and let it learn on its own. [86]*

Definition 3.5. *DL is a form of machine learning in which the computer network rapidly teaches itself to understand a concept without human intervention by performing a large number of iterative calculations on an extremely large dataset. It involves multiple layers of nonlinear processing units to extract features from data. [87]*

Definition 3.6. *DL is a ML method using multiple layers of nonlinear processing units to extract features from data. The network learns from each mistake it makes, which is analogous to how humans learn from their errors. These layers of concepts enable the machine to learn complicated concepts by building them out of simpler ones, creating a hierarchy of concepts that are deep with many layers. [88]*

3.4.2 The difference between DL, ML and ANN

DL algorithms are ML algorithms. Therefore, it might be better to think about *what makes DL special within the field of ML and ANN?*

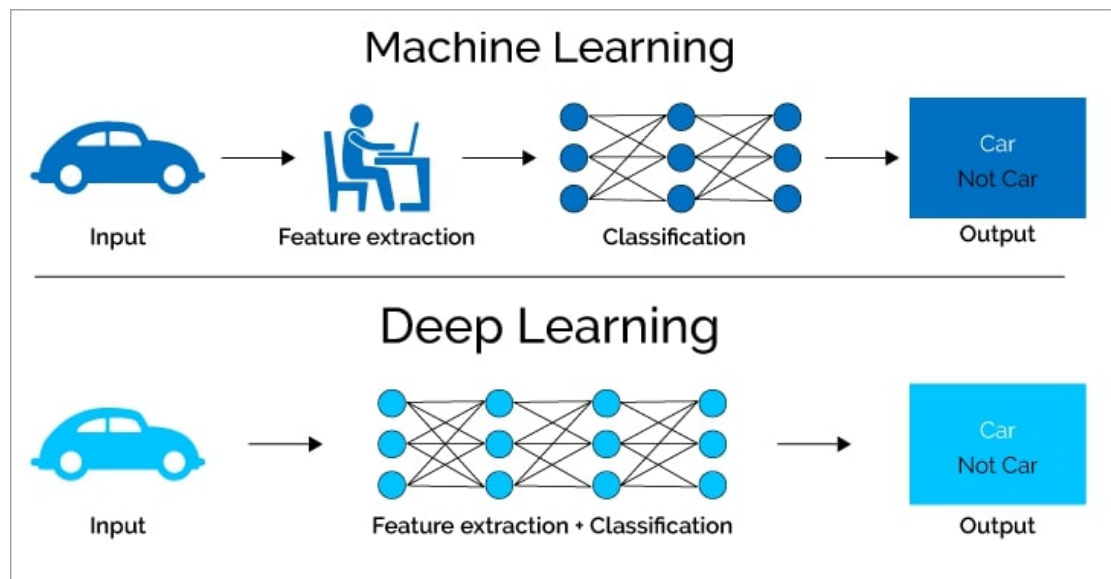


FIGURE 3.6: The difference between ML and DL models [16]

To answer the previous questions, the following aspects are to be considered. [89]:

1. First and foremost, whereas typical Machine Learning algorithms, such as linear regression or decision trees, have a relatively straightforward structure, Deep

Learning is built on an artificial neural network. This multi-layered artificial neural network is complicated and interconnected, much like the human brain.

2. Second, DL algorithms need a lot less human involvement. Although using the conventional machine learning technique, which includes ANN, a software developer would manually choose features and a classifier to organize images, verify that the output meets the necessary standards, and modify the algorithm if it necessary. In this context, the DL algorithm learns from its mistakes and automatically extracts the characteristics, as seen in the figure 3.6.
3. Lastly, compared to a standard machine learning algorithm, Deep Learning needs a lot more data in order to perform well. While machine learning uses a thousand data points, deep learning frequently uses millions. Because of its intricate multi-layer structure, a deep learning system requires a sizable dataset in order to remove irregularities and produce interpretations of superior quality.

In this context, two types of DL models are presented in the following subsequent sections.

3.4.3 Discriminative DL models

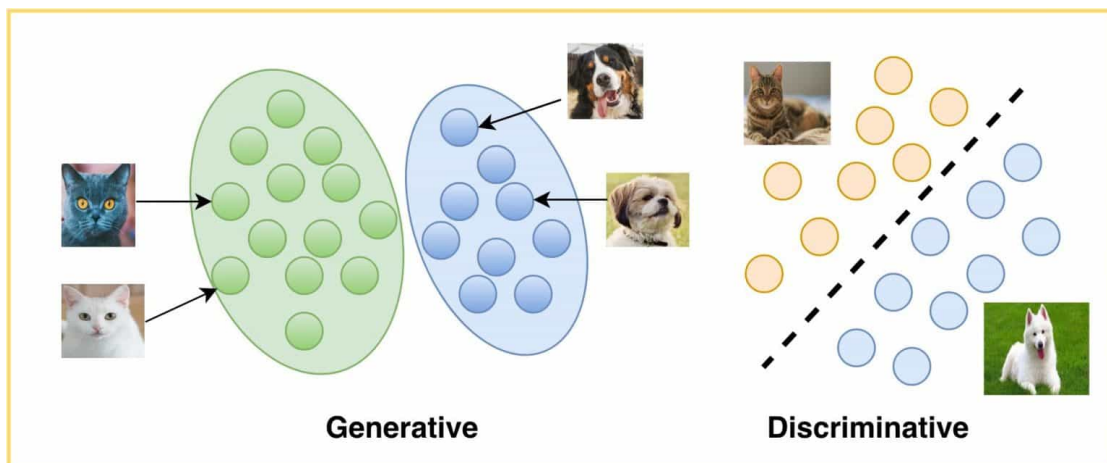


FIGURE 3.7: The generative and discriminative DL models [17]

As shown in the figure 3.7, Discriminative Deep Learning (DDL) models are a type of supervised ML models that focuses on learning the decision boundary between classes. It is widely used in the *classification problems* [90], *object detection* [91], *semantic segmentation* [92], *regression problems* [93], and *language modeling* [94].

In the following a set of the DDL models are described.

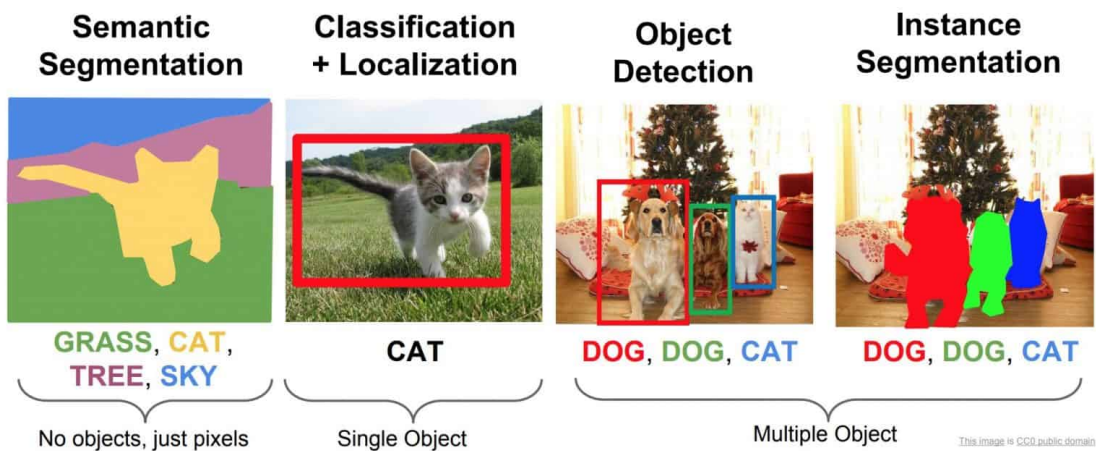


FIGURE 3.8: Examples of the discriminative DL models [17]

3.4.3.1 Capsule Neural Networks

In the AI area, some neural network architectures, like capsule networks (*groups of neurons*) as shown in the figure 3.9 [18], deploy dynamic routing algorithms on input data to adjust the weights and connections between neurons. Hence, they enable the network to continuously learn and adapt its internal representation. In capsule networks, dynamic routing improves the representation of relationships contained in the input data and to selectively route information to relevant capsules. Thus, dynamic routing allows facilitating better representation and understanding of hidden patterns withing data.

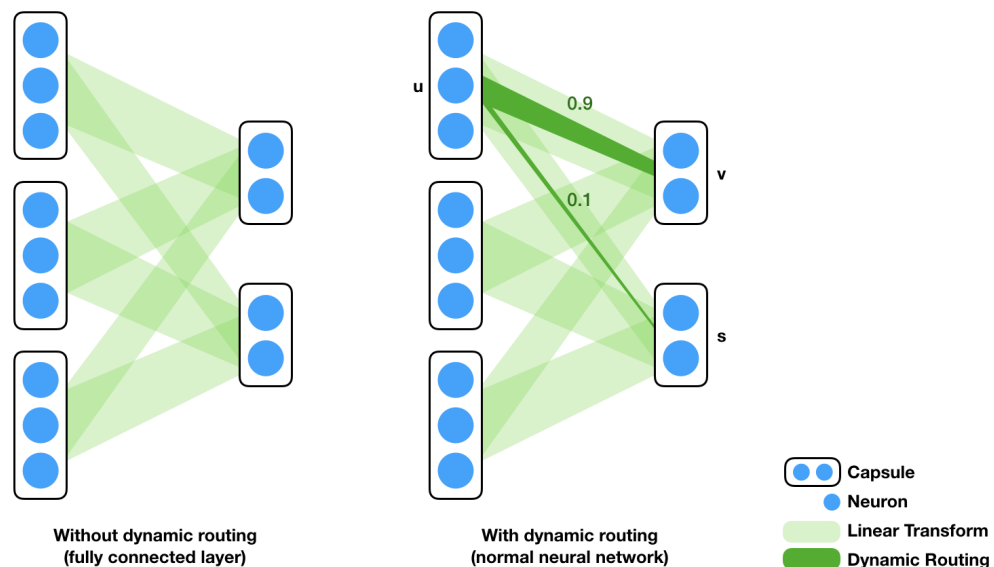


FIGURE 3.9: The importance of the dynamic routing of the caps-net [18]

Since the dynamic routing was proposed by Sabour et al [95], it has been considered as the standard routing method that is based on the squashing function which goes beyond

just collecting the margin between the input and output capsule vectors. In fact, the goal of such function is to reflect the probability for transforming the local features S of an item into global ones V , as it is formally specified in **equation (3.3)**.

$$V = \frac{\|S\|^2 * S}{(1 + \|S\|^2) * \|S\|} \quad (3.3)$$

In this context, CapsNets are designed to preserve spatial hierarchies within data, primarily in images, by capturing pose and orientation information through a structure of capsules. As shown in the figure 3.10, each capsule represents groups of neurons that work together to detect specific features and their relationships within an image.

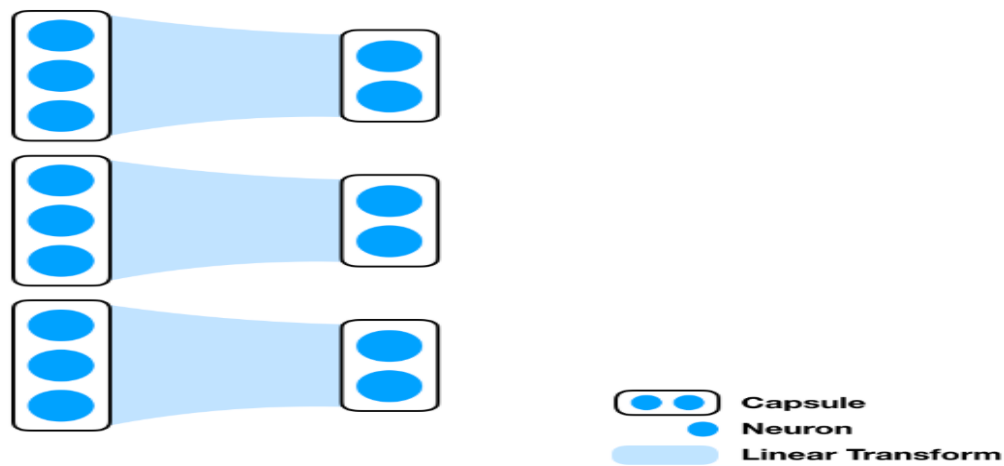


FIGURE 3.10: The capsule structure [18]

These capsules output a vector that includes both the probability of the presence of a feature and its pose (*orientation, position, and scale*) (as is explained in the figure 3.11). This approach allows Caps-Nets to be more robust to affine transformations and to better understand the spatial hierarchies within images.

However, a major limitation of CapsNets is their inability to natively handle graph-structured data [96, 97], which makes them less versatile for tasks that involve complex relationships among entities. Unlike Graph Neural Networks (GNNs), which are built to operate on nodes and edges, CapsNets are limited to grid-like structures (e.g., images), making it challenging to apply them to data that requires an inherent graph representation. The major limitations of the CapsNets are the following:

1. **Lack of Native Graph Processing:** CapsNets are fundamentally designed for regular, Euclidean data, where relationships follow a grid-like pattern. They lack

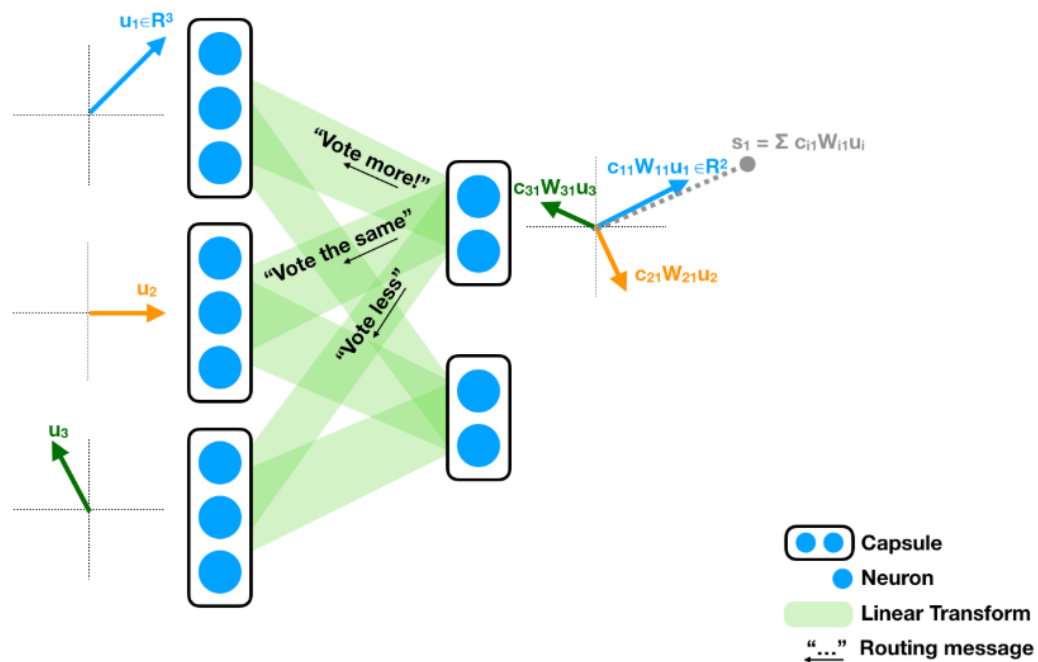


FIGURE 3.11: The feature transformation from local into global ones [18]

the message-passing mechanisms needed to process data in arbitrary graph structures, which makes them unsuitable for domains like social networks, molecular structures, and recommendation systems, where graph representations are critical.

2. **Inflexibility for Complex Relational Data:** Unlike GNNs, which can capture multi-hop relationships through iterative neighborhood aggregation, CapsNets struggle to represent dependencies between non-adjacent data points. This limits their effectiveness for tasks that require understanding complex, non-local relationships, such as in network analysis or bioinformatics.
3. **Inefficiency for Large-Scale Graphs:** The computational demands of CapsNets, combined with their limited adaptability for graph structures, make them inefficient for handling large, sparse graphs. GNNs are optimized to handle sparse, large-scale graphs, making them better suited for applications requiring scalable graph processing.

The GNN explained in the following aims to tackle the previous limitations of the Caps-Networks.

3.4.3.2 Graph Neural Networks

Graph Neural Networks (GNNs) [19] are a class of neural networks designed specifically to operate on graph-structured data. Unlike traditional neural networks, which work on data with fixed structures like images or text, GNNs as shown in the figure 3.12 [19] can handle data represented as nodes and edges, making them suitable for tasks involving complex relationships, such as social networks, biological networks, and recommendation systems. GNNs leverage the connectivity and relational patterns in graph data to make predictions, classify nodes, and generate embeddings.

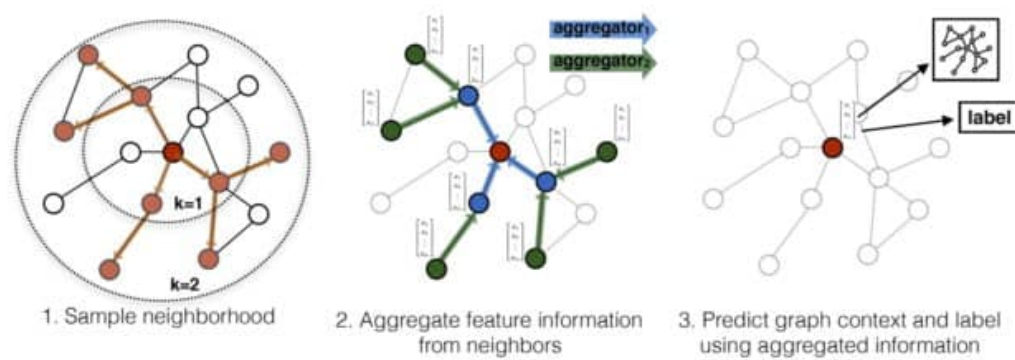


FIGURE 3.12: The GNN layers [19]

1. Key concepts

- **Graph Structure:** In a graph $\mathcal{G} = (\mathcal{U}, \mathcal{W})$, where $\mathcal{U}$ represents a set of nodes (or vertices) and $\mathcal{W}$ represents a set of edges that define connections between nodes. As shown in the figure, each node and edge can have attributes, such as features for nodes and weights for edges, representing their properties.
- **Message Passing:** GNNs use a message-passing framework, where each node aggregates information from its neighbors. During each layer of a GNN, nodes share information with their directly connected neighbors, updating their feature representations by combining their current features with those of their neighbors. This aggregation process allows GNNs to learn the structural dependencies within the graph.
- **Propagation Layers:** GNNs consist of multiple propagation layers. Each layer aggregates neighborhood information and updates node representations, enabling nodes to access information from increasingly distant nodes in the graph with each additional layer. This multi-layer aggregation is what allows GNNs to capture complex dependencies and patterns in the graph.

- **Graph Pooling:** Some GNNs include pooling layers, which condense information from groups of nodes or subgraphs into a single representation, making it easier to perform tasks like graph classification. Pooling layers are useful for tasks where a high-level understanding of the entire graph is required, rather than insights at individual nodes.
- **Readout and Prediction:** The final layer, known as the readout layer, collects information from all nodes in the graph or from specific nodes to produce an output. This output can be used for tasks such as node classification, link prediction, or graph classification, depending on the problem at hand.

2. Applications of GNNs:

- **Social Network Analysis:** In social networks, GNNs can be used to identify influential individuals, detect communities, and recommend new connections based on users' social graphs.
- **Molecular Biology:** GNNs are used to predict molecular properties, simulate protein-protein interactions, and assist in drug discovery by analyzing the structure of molecular graphs.
- **Recommendation Systems:** GNNs help recommendation engines by analyzing user-item graphs to provide personalized recommendations based on user interactions and similarities.
- **Traffic and Urban Planning:** GNNs can be applied to transportation networks for predicting traffic flow, optimizing route planning, and urban development.

Although GNNs excel at learning from graph-structured data and handling complex relational information, they lack some key capabilities found in CapsNets, such as spatial awareness, hierarchical representation, and transformation equivariance. CapsNets are better suited for tasks that require understanding spatial hierarchies and pose information, which makes them particularly effective in image-related applications where GNNs might struggle. Therefore, while both networks have unique strengths, CapsNets are more advantageous in scenarios involving spatial and hierarchical data, while GNNs excel in tasks where relational data in graph format is key.

In this context, the next subsection aims to present the combined architecture named Caps-GNN.

3.4.3.3 Capsule Graph Neural Networks

The Caps-GNN (Capsule Graph Neural Network) [98, 99] is a hybrid model that combines the strengths of Capsule Networks and Graph Neural Networks to leverage both spatial awareness and graph-structured relational learning. This architecture is specifically designed to capture both hierarchical relationships and complex, non-Euclidean data structures, making it powerful for applications such as social network analysis, molecular chemistry, and computer vision tasks involving graphs.

The following is an overview of the Caps-GNN architecture.

- **Graph Neural Network (GNN) Layer** The initial step in Caps-GNN is to use a GNN to generate node embeddings from graph data:
 - **Node Embedding Extraction:** A GNN, such as a Graph Convolutional Network (GCN) or Graph Attention Network (GAT), processes each node by aggregating information from its neighbors. This aggregation builds a feature representation that captures both the node's attributes and the influence of its neighbors.
 - **Message Passing and Aggregation:** The GNN layer performs message passing between nodes, where each node updates its representation based on the embeddings of its connected nodes. After a few layers of aggregation, each node will have an embedding that captures information from a localized neighborhood within the graph.
- **Capsule Layer (Primary Capsules)** Once the initial node embeddings are generated, they are fed into a primary capsule layer:
 - **Primary Capsule Formation:** Each node embedding is mapped to a set of primary capsules. In Caps-GNN, each capsule represents a feature vector (*or "part"*) that captures certain characteristics of the node, such as community structure, type, or functional role in the graph.
 - **Capsule Representation:** Each primary capsule is a vector, not a scalar, which allows it to represent complex properties (such as orientation and intensity) of a node's feature. This helps the network retain more nuanced information about each node's characteristics.
- **Dynamic Routing and Aggregation** The next step involves the use of dynamic routing between the primary and higher-level capsules. This routing process allows Caps-GNN to learn which parts belong together, much like how a Capsule Network would recognize parts of an object in an image.

- **Dynamic Routing Mechanism:** Capsule vectors are routed iteratively to higher-level capsules based on the agreement between their outputs. The dynamic routing process is critical for Caps-GNN because it captures how different parts (or nodes) contribute to the overall graph structure.
- **Higher-Level Capsules (Graph Capsules):** These capsules aggregate the information from multiple nodes and their capsules to form higher-level representations, allowing Caps-GNN to encode complex graph structures. For example, a higher-level capsule might represent a subgraph or a motif within the larger graph, encoding relationships and patterns that emerge from node connections.
- **Readout and Prediction Layer** The final layer of Caps-GNN is the readout and prediction layer:
 - **Graph Embedding Readout:** After the dynamic routing, each graph capsule has a vector representation that can be used for prediction. If the task is node classification, the output for each node capsule can directly be used as the prediction. For graph classification tasks, the graph capsules are pooled into a single vector to represent the entire graph.
 - **Prediction Output:** The readout layer then feeds the graph-level or node-level representations into a classifier, which could be a fully connected neural network, to make the final predictions. These predictions can range from node classification, link prediction, to entire graph classification tasks.

While Caps-GNN brings valuable capabilities in representing hierarchical relationships within graph-structured data, its application to BPs is limited by issues related to the following issues:

- **Scalability:** BPs often involve large and complex graphs (such as workflows with numerous steps, departments, or users). Caps-GNNs, which require significant memory and processing power, may struggle to scale efficiently to handle these large datasets.
- **Temporal modeling:** BPs are typically time-dependent, with actions occurring sequentially and affecting each other over time. Caps-GNN lacks a built-in mechanism to capture temporal dependencies or sequential relationships within process data. This limits its effectiveness in applications requiring time-sensitive insights, such as workflow optimization or process mining.

- **Adaptability to non-graph or evolving data:** Not all BP data is natively graph-structured; some data may be better represented as sequences or in tabular format. Caps-GNN's architecture is optimized for graph data, making it less suitable or efficient for these types of data without substantial pre-processing.

The following subsequent section aims to explain the second type of DL models which are the generative models.

3.4.4 Generative DL large language models

As shown in the figure 3.7, the Generative Deep Learning (GDL) models are designed to create new data that resembles the training data. They can generate various types of content, including images, text, audio, and video [100]. In particular, GDL Large Language models (GDLLL) are specialized in understanding and generating human-like text. They are trained on vast datasets to perform NLP tasks, such as text generation, translation, and summarization.

Notable examples include OpenAI's GPT series. GDLLL models use transformer architectures, which are DL capable of handling sequential data efficiently. They are widely used in applications like chat-bots, virtual assistants, and content translation [101].

The GPT3, Bert and many other models are the most used GDLLL models that are based on the transformer generative model. In the following the transformer architecture is explained and discussed.

3.4.4.1 Transformer architecture

Transformers [75, 102] are an advanced DL network that excels at comprehending context and meaning by evaluating connections in sequential input, such as sentences. The word "transformer" refers to their capacity to transform one sequence into another.

The fundamental benefit of these types of models is their capacity to process the full sequence at once, as opposed to RNNs and LSTMs [103], which process one step at a time. This enables transformer systems to be parallelized, making them faster to train and utilize. As shown in the figure 3.13, the *encoder-decoder architecture*, the *attention mechanism*, and *self-attention* are the main elements of transformer models.

1. Encoder-decoder architecture:

In the transformer model, the encoder takes a series of input data (typically text) and turns it into vectors, such as vectors indicating the semantics and location of

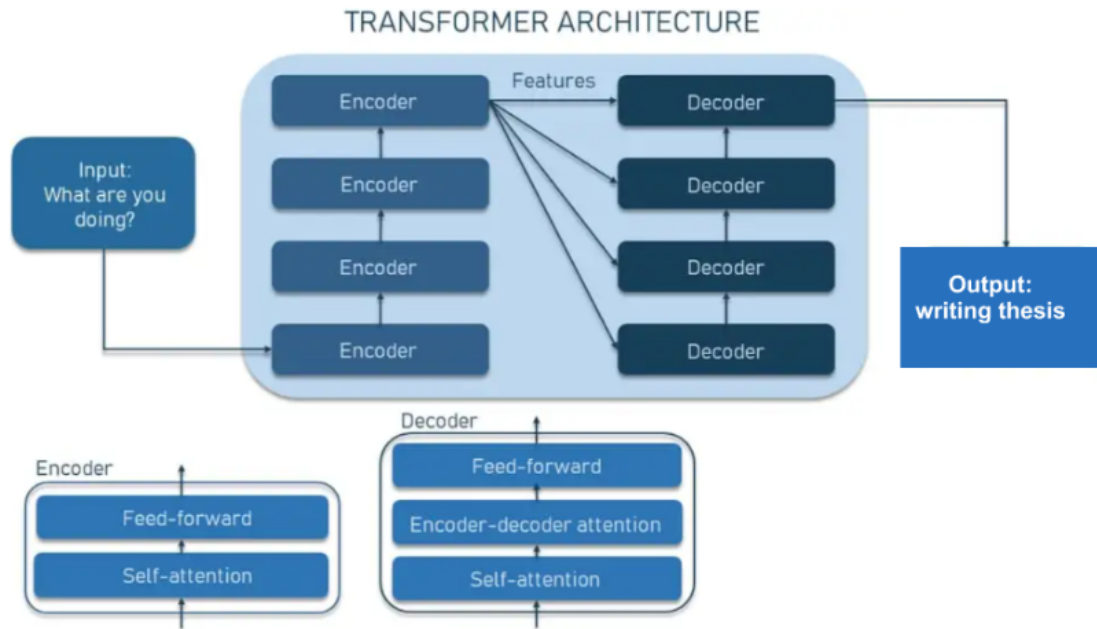


FIGURE 3.13: The transformer architecture [20]

a word in a sentence. This continuous form is commonly referred to as the input sequence's "embedding". The decoder takes the encoder's outputs and utilizes them to construct context before producing the final output.

Both the encoder and the decoder are made up of a stack of identical layers, each with a self-attention mechanism and a feed-forward neural network. There is also encoder-decoder attention in the decoder.

2. Attention and self-attention mechanisms:

As shown in the figure 3.14, the *attention mechanism* is the central component of transformer systems, allowing the model to make predictions based on particular elements of the input. The attention mechanism assigns a weight to each piece of the input, reflecting its significance for the present prediction. These *weights* are then used to create a *weighted total of the input, which is used to construct the forecast*. *Self-attention* is a form of *attention mechanism* in which the model focuses on various elements of the input sequence in order to create a prediction. It signifies that the model is looking at the input sequence several times, each time focusing on a different element of it. In this transformer architecture, the *self-attention mechanism* is applied multiple times in parallel, allowing the model to learn more complex relationships between the input sequence and the output sequence.

Transformers are classified as a type of semi-supervised learning in terms of training. This indicates that they receive unsupervised pre-training using a sizable collection of

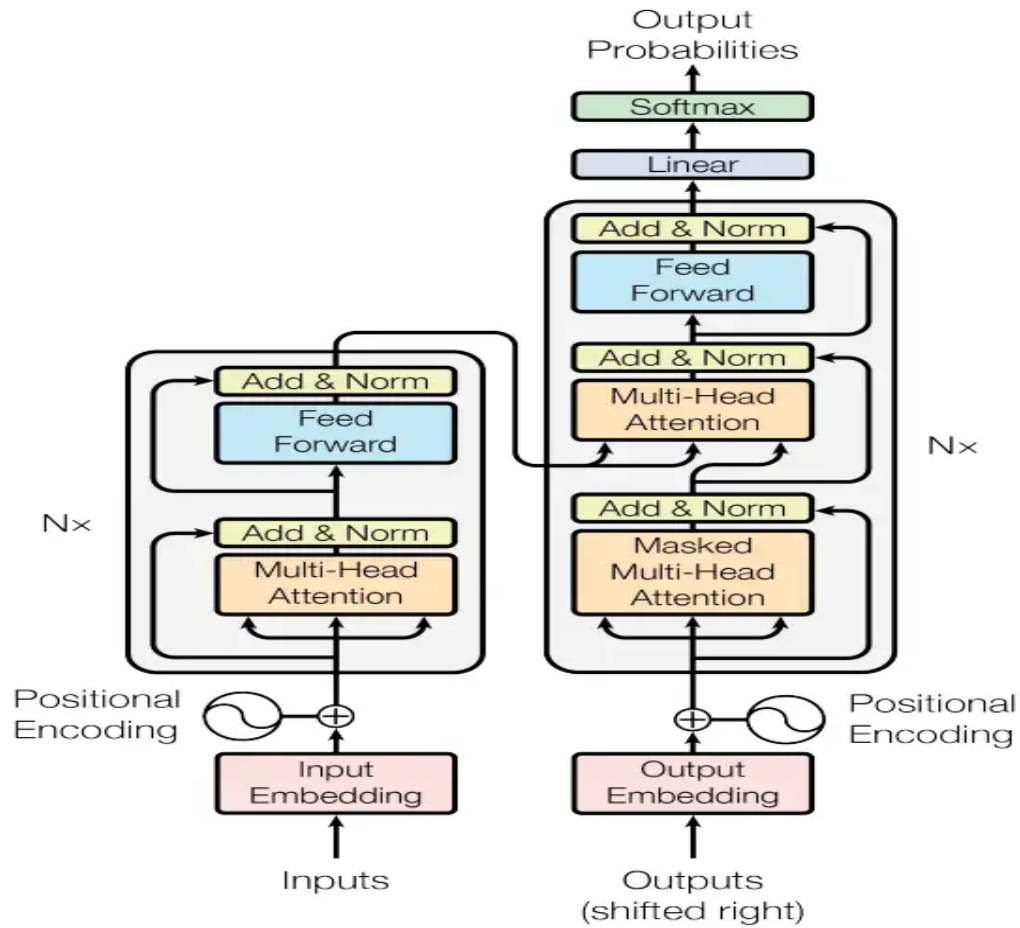


FIGURE 3.14: The transformer architecture with attention mechanism [21]

unlabeled data. Through pre-training, the model is able to identify broad trends and connections within the data. A smaller labeled dataset tailored to the job at hand is then used for supervised training, which refines the model. This adjustment enables the model to function more effectively on that specific task.

In this context the following projects explained in the following section are built from this GDLLL model named the transformer.

3.5 Industrial Projects leaded by GDLLL models and chat-bots systems

Despite the fact that the GDLLL model landscape is always changing due to the interest in new initiatives, we have identified the top four models with the greatest worldwide influence. It is more thorough because the GPT3 is the most often used. After that, two more are briefly explained.

3.5.1 List of popular GDLLL models

This subsection aims to list the set of the GDLLL models.

1. GPT-3 by OpenAI:

GPT-3 (*Generative Pre-trained Transformer 3*) [104] is one of the most advanced and powerful language models developed by OpenAI that is based on transformer architecture. It builds upon its predecessors (*GPT and GPT-2*) with significant improvements in size, performance, and capabilities.

(a) Training:

GPT-3 was trained on a diverse and extensive corpus of text from the internet, which included books, articles, websites, and other sources. This broad training data allows GPT-3 to generate text that is coherent, contextually relevant, and often indistinguishable from text written by humans.

The training process involves predicting the next word in a sequence, given the preceding context, which helps the model learn grammar, facts about the world, and some reasoning abilities.

GPT-3 represents a significant milestone in the development of language models, showcasing the power and potential of AI to understand and generate human language. Its capabilities have far-reaching implications for various fields and industries, driving innovation and transforming how we interact with technology.

(b) Fine tuning the large GPT-3.5-turbo model:

Fine-tuning or customizing is a vital procedure that enables to tailor a pre-trained model to an individual application [105]. With access to the large quantity of data required to train models like as GPT-3.5-turbo, fine-tuning allows to improve the results obtained from these models for a specific use case.

Fine-tuning a model involves many essential procedures.

- i. First, prepare and submit the training data which should be tailored to an application and reflective of the task wanted the model to perform.
- ii. Once the data has been uploaded, it may train a new fine-tuned model which may take some time, but is well worth the effort because it results in a model that is much more tailored to unique requirements.
- iii. Once fine-tuned model has been trained, it may begin utilizing it to get improved outcomes.

In this context, the fine-tuning process is an effective strategy for improving performance on a variety of jobs while saving money in the long term. It

can also help the business to improve its performance through leveraging the capabilities of pre-trained models such as GPT-3.5-turbo.

2. BERT language model by Google:

Google created the pretrained language model BERT (*Bidirectional Encoder Representations from Transformers*) in 2018 [106] . By examining the connections between words in a phrase rather than focusing just on individual words, it aims to comprehend the context of a given text. The model's ability to interpret text both left to right and right to left is shown by the "bidirectional" component.

3. LaMDA by Google:

Google created the LaMDA language paradigm [107] for discussion applications. Unlike conventional models, which are usually task-based, it is intended to produce conversational interaction in a free-form manner, which makes it more organic and subtle. The model gained notice after a Google developer asserted that it seemed sentient since it can give responses that imply an awareness of its own nature.

137 billion parameters of conversation data were used to train LaMDA. This enables it to understand the subtleties of conversations that are open-ended. The concept will be used across all of Google's products, including Workspace, Google Assistant, and search.

The transition from early language models to advanced chat-bot systems such as ChatGPT demonstrates the tremendous improvements in AI and deep learning. ChatGPT demonstrates the capability of fine-tuned language models in developing more interactive and human-like AI systems, with the potential to alter numerous areas of human-computer interaction. The following section aims to explain the chat-bots and the birth of the chatgpt.

3.5.2 Chat-bots systems

This section aims to explain the basic definition, objective and examples of the chat-bots, followed by the birth of chat-GPT.

1. Chat-bot definition and objectives:

A chat-bot [108] is a software application designed to simulate conversation with human users, primarily through text or voice interactions. Powered by various technologies, including rule-based systems, natural language processing (NLP), and machine learning, chat-bots can interpret user requests and respond in a conversational manner. Chat-bots are widely used in customer service, entertainment,

and as personal assistants, capable of answering questions, providing information, and performing tasks like booking appointments or troubleshooting issues. The main objectives of a chat-bot include:

- **Automating Routine Tasks:** Chat-bots are often deployed to handle repetitive inquiries, reducing the workload on human agents. They assist users with frequently asked questions, help desk support, or basic troubleshooting.
- **Enhancing Customer Engagement:** By providing immediate responses, chat-bots improve user experience and engagement. They are available 24/7, providing support at any time and offering quick, accessible interactions with minimal wait times.
- **Personalized Assistance:** Advanced chat-bots, especially those with AI capabilities, can offer personalized recommendations, helping businesses to create a tailored experience for each user.
- **Cost-Effective Service:** Chat-bots provide a cost-effective solution for organizations, enabling them to scale customer service operations without a proportional increase in staffing costs.
- **Data Collection and Insights:** Chat-bots can collect valuable information about customer preferences and behaviors, which businesses can use to enhance their services and products.

2. Example of existing chat-bots:

The following represents the examples of the existing chat-bots:

- **Siri (Apple):** Siri [109] is Apple's virtual assistant integrated into iOS, macOS, and other Apple devices. Siri uses voice recognition to help users with tasks such as setting reminders, searching the web, and controlling smart home devices.
- **Alexa (Amazon):** Amazon's Alexa [110] is an AI-powered assistant embedded in Echo devices and other smart gadgets. It can perform tasks like answering questions, playing music, and integrating with various smart home services.
- **Google Assistant:** Google Assistant [111], available on Android devices and Google Home, is an AI-driven chatbot that performs voice and text-based tasks, such as controlling smart devices, providing personalized recommendations, and assisting with various user queries.
- **Woebot:** Woebot [112] is a mental health chatbot that leverages cognitive-behavioral therapy (CBT) principles to assist users in managing mental health

challenges. Woebot uses conversational techniques to help users identify and address negative thinking patterns.

- **Replika:** Replika [113] is an AI chatbot designed to provide companionship and emotional support. Unlike utility-focused chatbots, Replika is conversational and learns from users to create a personalized, friendship-like experience.

3. The birth of Chat-GPT:

ChatGPT, developed by OpenAI [114], represents a significant advancement in chatbot technology, bringing conversational AI to a new level of sophistication. Launched in November 2022, ChatGPT is built upon OpenAI's GPT-3 and GPT-4 language models, which are large-scale, transformer-based models trained on a diverse dataset covering a wide array of subjects and conversational styles. ChatGPT can understand context, generate coherent and contextually relevant responses, and adapt its answers based on user input, making it capable of engaging in more human-like conversations. The key features of the Chat-GPT are the following:

- **Contextual Understanding:** Unlike traditional rule-based chatbots, ChatGPT uses natural language understanding to interpret complex queries and respond accurately. It can track conversation context, allowing for multi-turn interactions where the chatbot "remembers" previous parts of the dialogue.
- **Dynamic Responses:** ChatGPT can generate detailed and varied responses, unlike static responses in earlier chatbots. This allows for a more interactive, engaging conversation experience.
- **Multi-Domain Knowledge:** Trained on a broad dataset, ChatGPT can respond to inquiries across a wide range of topics, from scientific information to everyday knowledge, making it a versatile tool for different applications.
- **Adaptability:** ChatGPT's underlying model can be fine-tuned for specific applications or industries, allowing it to support specialized domains such as customer service, education, healthcare, and content creation.

Since its launch, ChatGPT has gained widespread use across sectors, from customer support and content generation to education and personal productivity. It has also inspired the development of newer generative AI models in various industries, demonstrating the transformative potential of conversational AI. This superiority of Chat-GPT is caused by the advancements built in DL and transformer-based language modeling, specifically with versions like GPT-3 and GPT-4.

3.6 Conclusion

The advancements in deep learning have fundamentally transformed AI applications, enabling sophisticated problem-solving capabilities that extend beyond traditional computational methods. While conventional AI techniques remain relevant in scenarios demanding interpretability and structured reasoning, the dominance of deep learning underscores a paradigm shift toward data-driven intelligence. This chapter has highlighted the key components of deep learning frameworks and their role in modern AI applications, particularly in natural language processing and chatbot development.

Building upon this foundation, Chapter 4 delves into AI-driven decision support systems, specifically within the context of business process management. By integrating deep learning models with decision-support mechanisms, the next chapter explores how AI can enhance business intelligence, process optimization, and automated decision-making. The transition from general AI techniques to their application in business process decision-support systems serves as a bridge between theoretical advancements and practical implementations, paving the way for innovative solutions in enterprise AI.

Chapter 4

Literature review of the BP-DSS approaches

4.1 Introduction

While the cornerstone goal of the PM algorithms is to capture the behavior reflected in an event-log data, their major downside is the hand-crafted manner in which the necessary features are extracted from the event-log data. As any other ML approach, such characteristics or data representations are critical to the performance of the BP techniques, and consequently the efficiency and the quality of the made BP decisions. In this context, the shift from the hand-crafted into automatic features' extraction in the BP area, was addressed by various DL approaches. Such DL techniques are deployed to mine extra-knowledge from the execution data (*as Process schema discovery, process prediction, anomaly detection, process recommendation and so on*)

In this context, this chapter offers a thorough qualitative literature review of BP methods based PM approaches on one side and the BP based DL approaches on the other. The conducted studies are categorized based on a set of relevant identified criterion, namely: *the inputs, outputs, fundamental assumptions, algorithm type, and extra-functional properties*.

The chapter begins by describing the search methodology, followed by an examination of the BP methods. The above initiatives are then combined in a related work table. Finally, this chapter outlines our road-map for addressing current gaps and shortcomings. This road-map will help us to define our research contributions in this BP decision making field.

4.2 Search methodology

We performed a Systematic Literature Review (SLR) of BP mining approaches using the SLR criteria stated in [115]. In accordance with these criterion, we start by formulating the search query that meets our search objectives. Thus, a search expression was developed from the study topic in order to retrieve related materials from academic digital libraries. The subsections that follow describe the SLR steps used in this work.

4.2.1 Research question

The primary goal of this study is to examine various techniques for evaluating and classifying the BP mining approaches and techniques [116]. To narrow the search space, our Research Question (RQ) is formulated as follows:

Comparing with the conventional PM and ML approaches, how the advanced AI methodologies enhance decision-making within the BP domain ?

Given the contextual conditions, human factors, and the intentional business decisions of exploiting an event logs data, a large degree of internal heterogeneity between execution data and uniform behavior is enforced. In this context, analyzing those factors from the performance characteristics and the potential decisions to be made from the event log data is a challenge for both conventional and deep learning methods. In fact, although traditional PM approaches are quite powerful for extracting manually the features generated from the event log data, they still fail to meet the automatic features extraction that DL models process from the extracted data structure (graph). In this context, the multidisciplinary nature of this field, along with the emergence of these two types of PM methods, leading to two distinguished decision-making techniques, contributes to the field's blurriness, which makes it difficult to articulate the methodologies used in the BP domain.

In what follows, we discuss the process of selecting the relevant works related to our RQ.

4.2.2 Relevant research works selection

Through RQ, the following keywords were taken into account to find pertinent works linked to the research subject:

- "the Business Process" a relevant study must consider the logic hidden behind the BP for decision support system;

- "Event-log data / BP model" a relevant study must consider event logs or the BP model as inputs;
- "BP Feature extraction" a relevant study should understand the BP features as well as the extraction tools;

We formulated a search query using the previous keywords and submitted it to the search engine Google Scholar (GS). As it's known, GS is the largest academic search engine in the world, which includes additional academic databases including IEEE Xplore and ACM Digital Library [117]. The returned papers are those containing at least one of the terms listed above in *the title, keywords, or body of the document*.

The search yielded around 90 distinct papers published between 1994 and April 2023. Over the past three years, it has been observed that there is an upsurge in research publications on BP mining, indicating that the issue is garnering more and more attention by the scientific community.

To exclude irrelevant results and to avoid examining marginal studies, we adopted the following inclusion criteria:

- **Criteria 1 (Cr01):** The research focuses on BP mining algorithms for obtaining important characteristics (*This criterion was evaluated by reading the title and abstract, as well as the final experiments*). .
- **Criteria 2 (Cr02):** The work has been referenced at least three times (*the number has been reduced for articles published before 2017*).

The list of the selected papers that met the previous inclusion criteria was then evaluated further in light of the following exclusion criteria:

- **Criteria 3 (Cr03):** The study does not suggest any particular approach for mining the BP, and the overview publications were excluded.
- **Criteria 4 (Cr04):** The suggested method focuses on extracting features from BP execution data in order to address a particular issue (*prediction, recommendation, process discovery, etc.*).
- **Criteria 5 (Cr05):** A particular event log data-set or a specific BP model should be used as the technique's input to extract the features.
- **Criteria 6 (Cr06):** Either a PM or a DL approach is used as a technique to extract the features.

The application of the exclusion criteria (Cr01...Cr06) conducted to the identification of only 32 relevant studies from the 90 works suggested in the previous stage as a response to our RQ.

The following section describes the analysis and the classification of the identified 31 studies.

4.3 Analysis of the BP methods

The RQ can be addressed by classifying the selected 32 works using several dimensions that describe the typology of the current approaches and their features. Each research, in particular, may be examined by taking into account the following aspects (*the analysis dimension's*):

- **Input data:** (*Event log data or BP model or textual BP data*)
- **Output:** (*Process prediction, event recommendation, anomaly detection, process discovery, conformance checking ...etc.*)
- **The employed assumption** (*homogeneous or asymmetrical behavior...etc.*)
- **The type of the algorithm** (*Conventional PM or PM based DL*)
- **The evaluation data** (*real-life or synthetic logs*)

4.3.1 Input

The structure of the input data has an important impact on the success and performance of the target method in ML and PM in general, and in DL techniques in particular. All of the proposed techniques, use an event-log data as input as the once described in the table 1.1 and the table 5.1. Despite the effectiveness of the graph structure on the DL, only some of the suggested works adopt the graph models as an input structure. Numerous factors could be perceived in this context as cues for the data used in each study. According to our analysis, the pivotal factors throughout these considerations are *the primary goal of the PM approach (factor 01)* and *the effectiveness of the adopted data in the DL model (factor 02)*.

Through those two factors, the selected works are divided into several categories in order to clarify how and why the input data was adopted throughout the analysis of the research papers.

- **Factor 01:** The overall works of that category, select the form of their input data according to the PM goal that is ranging from discovering, conformance checking and enhancement issues.
 - The studies [27–31, 118] use the log within its raw form to discover the BP-model.
 - Unlike the previous work category, other studies employs the raw form of the event-log data and the BP-model graph with the PM approaches [119–121] to check the conformity of the discovered model with the original execution
 - In the last category of the PM approaches, the works [122–124] are used based on a BP-graph model to enhance the actual discovered model with further performance features of an event-log data.
- **Factor 02:** The works of this category aim to achieve a powerful DL model whatever the PM purpose was (*process discovery, conformance checking, process enhancement, recommendation, prediction, etc*). While the theory behind the graph structure makes it a powerful tool for different ML and DL approaches, few of works forward it as an input data of their PM or DL approach. Primarily, the work [125], aim to optimize the input petri-net BP model using a PM discovery approach, where the authors reached that goal by a DL approach through an event-log data [126]. Moreover, the authors [33] model an event-log data as a graph structure, then it was learnt by a GNN to discover a petri-net BP model.

The key difference of those works of both factors, rooted toward the proposed approach, as well as the assumption behind their suggested technique, framework and architecture.

In what follow, we explain how the BP mining approach can vary based on the output data.

4.3.2 Output

The output of Business Process (BP) mining consists of discovered process models, compliance reports, and enhanced process insights, depending on the chosen methodology. These outputs support data-driven decision-making by enabling organizations to refine their operations, reduce inefficiencies, and ensure adherence to business rules. Managers can leverage these insights to streamline workflows, allocate resources efficiently, and implement proactive measures for continuous process improvement. The classification of the output related works are analyzed in the following.

- **Model-based approaches:** The target models range from formal to graphical representations. Many studies generate process models that are easily understandable by end-users as the output. These process models, also known as BP-models, describe the behavior observed in a collection of traces within event-log data. The authors in [33] generate a petri-net model to describe the behavior of distributed systems. Further, the authors [122] build a BPMN model communicating process information across different stakeholders, but it is not as suitable for modeling parallel or concurrent processes as Petri nets. To overcome this issue, the authors in [27–31, 127] discover a workflow models typically include nodes representing tasks and decision points, along with connectors representing the flow of information between them. Workflow models can be used to model simple processes to support business decisions, but they may not be suitable for complex processes that involve parallel or concurrent activities.

In summary, Petri nets are best suited for modeling complex, distributed processes with parallel or concurrent activities, while BPMN is ideal for modeling simpler processes with a clear, sequential flow. Workflow models are best suited for automating processes and ensuring consistent execution, but may not be as effective for modeling complex or distributed processes.

- **Decisive-based approaches:** this type of approaches on BP refer to the methods and techniques that involve making decisions based on specific business rules and resources constraints within a process. Those approaches that incorporate DL and PM techniques are gaining increasing attention. They aim to improve decision-making processes by leveraging large amount of data and various management rules suitable for advanced analytic.

By analyzing process data and applying ML algorithms, decision-based approaches can identify patterns [128], measure the variation between the discovered model and the original data [120, 121] and make predictions about future outcomes [129–134], as well as, detect anomalous behavior [124, 135–138]. Further, the chat-bot [139] can lead to more accurate decision-making and better overall performance of the business process. Leveraging the large amounts of data generated by business processes, these approaches can uncover hidden patterns and insights that may not be apparent through traditional methods.

Overall, decisive-based approaches on business process have the potential to bring significant benefits to organizations, including better decision-making, increased efficiency, and improved competitiveness.

- **Optimization-based approaches:** this type of approaches in BP aim to enhance the efficiency and effectiveness of business processes by utilizing mathematical optimization techniques. Mathematical models are developed to represent the BP's constraints and objectives, such models are solved to find the optimal solution that maximizes efficiency and effectiveness while minimizing costs, time, and other resources. The peak of those mathematical models are those who optimize BP models and executions, as well as the ones that target to improve the quality and implementation of the discovered models in real-world scenarios. This can be achieved through process automation, continuous process improvement, and the use of advanced technologies such as ML, AI, and data analytic. By optimizing BP models and executions, organizations can lower costs, boost productivity, enhance customer satisfaction, and improve overall organizational performance. This perspective is essential for organizations to remain competitive in today's rapidly changing business environment.

According to the selected works, the authors [125] optimize through an abstraction PM approach the complex BP-model generated from an event-log that suffer from the problem of inaccuracy and high-complexity. Furthermore, the authors [128], propose a semi-supervised approach, that aim to discover the BP-behavioral pattern instead of creating a complex BP-model that may contain either complex or over-generalized fragments. Moreover, a DRL approach [126] is applied to activities, traces, logs and models in order to learn highly informative features in low-dimensional vectors. Finally, to a real-time optimization of BP-models in an environment that includes some degree of uncertainty [140] as well as an optimal resource allocation [141] a RL models is applied.

One of the essential factors in obtaining an output from input data using PM or DL algorithms is the underlying assumption. The following subsection will delve into the various assumptions employed in the field of BP mining.

4.3.3 The employed assumption

The success of a ML method is highly tied to the employed assumption, which aids in the generation of an output from input data using a PM or DL algorithm. The assumption in the context of BP mining refers to a specific behavior of the event-log data. A mismatch between the assumed behavior and the actual log characteristics can lead to erroneous models, reduced interpretability, or poor generalization and decision making. In this context, Validating assumptions is an important stage in BP mining.

In the following the selected works are classified through the possible deployed hypotheses:

- **Behavior modeling based assumptions:** This type of assumption refers to the specific hypothesis made about the behavior of the process under analysis and is used as a guideline during the modeling activities. Often, these assumptions are expressed by expert knowledge or basing on the previous empirical studies. For instance, when analyzing a BP, one may assume that certain activities are always performed in a certain sequence, or that certain conditions must be satisfied before a particular activity can be executed. Those assumptions are then used to construct a formal model of the process at hand, such as a Petri net or a workflow diagram, which can be used to analyze the process's behavior and identify potential bottlenecks, inefficiencies, or compliance issues.

The well known assumption is the intention-based modeling, this type of behavioral hypothesis is captured from the event-log data and allows to characterize the implicit intentions of actors (*i.e., individuals or organizations*) and their decision-making processes during the BP progression. The ultimate goal of such models aim to improve the understanding of the actors' behaviors, motivations, and objectives when performing tasks within the BP. Intention-based models often incorporate Agent-Based Modeling (ABM) techniques, which allow the creation of computational models of individual actors' behaviors and interactions in a BP, as well as the simulation of different scenarios while evaluating the impact of changes on the BP actor's intentions and behaviors. Leveraging Hidden Markov Models (HMM), the authors in [27] model the association between users' activity logs and the techniques they employ to achieve their goals as evidence of this modeling trend. Furthermore, some actionable insights that can refer to potential deviating patterns can be identified through PM and DL techniques. These patterns are considered as a second goal of the intentional modeling feature.

The second type of the behavioral BP based assumption is materialized by the Oriented-Based Modeling (OBM). Inspired from the area of complex-system modeling, this type of assumption is based on powerful technique for structuring and organizing elementary items and their relationships. In fact, by defining classes and the relationships [31], the process can be splitted into smaller and more manageable parts that can be designed and implemented more easily [29, 142]. Additionally, OBM can help to identify potential issues or areas for improvement in the process [30, 123] by highlighting areas where classes or behaviors may be incomplete or inconsistent.

Moreover, Transition-oriented modeling is a type of BP modeling that focuses on the transitions or changes that occur between different stages or states of a process. This approach [27–30, 127] views the process as a series of transitions between states, and represents these transitions in the model using directed edges. In this type of modeling, the focus is made on capturing the specific actions or events that trigger transitions between states, and the conditions that must be met to trigger those transactions. The result is a highly structured model that provides a detailed representation of the process flow and the conditions under which it operates.

Actually, the type of behaviors can be backed by the homogeneous or asymmetrical hypothesis of a BP. Homogeneous behavior described by [131, 132], refers to the situation where all events in the log have similar characteristics and can be grouped together based on common attributes or values. Asymmetrical behavior employed by [30, 118, 123], on the other hand, refers to the situation where the behavior of events in the log differs depending on their attributes or values.

- **Abstraction based assumptions** Abstraction is a key concept when modeling BP because it allows us to represent complex BPs in a simplified, structured and understood way. In this perspective, abstraction-based assumptions refer to the assumptions made during the modeling process that involve simplifying or focusing on certain aspects, considered as levels of concerns, instead of others which are perceived as not interesting. In other words, abstraction-based assumptions involve ignoring or simplifying certain details or complexities in order to make a problem more tractable or to focus on key aspects of the problem. Here after, the most remarkable works within abstraction-based modeling are analysed.
 - One common abstraction-based assumption in BP modeling is the assumption of a linear process flow, where each task is performed in a sequential order [125]. This way of perceiving simplified BP the representation of the process, but may not reflect the actual flow of work in the organization.
 - Another abstraction-based assumption is the use of generic task labels [123, 127, 141]. These labels provide a high-level view of the tasks involved in the process, but may not accurately reflect the specific actions taken by individuals or departments within the organization.
 - Other common abstraction-based assumptions include the use of simplified data models [29], the exclusion of non-critical tasks or exceptions, and the assumption of perfect execution of tasks without any delays or errors.

Although abstraction-based assumptions are useful for simplifying complex problems and making them more manageable, they can also introduce biases or errors into the analysis if they are not based on sound reasoning or evidence. Therefore, it is important to carefully consider the assumptions being made in any model or analysis, and to evaluate their validity and potential impact on the resulting models.

- **Validation based assumptions** Validation-based assumptions on BP models refer to the assumptions that are made regarding the validation and verification of the BP models. These assumptions are crucial for ensuring that the elaborated BP models are accurate and reliable.

Different view-points are the pillar of that validation stage. The authors in [119], employs a multi-perspective declarative process model as an approach to model BP's taking into account multiple perspectives or viewpoints. This approach allows a more comprehensive understanding of the process and can lead to an effective conformance checking mechanisms. Moreover, the authors in [120] argue that there is no need to spend unnecessary time to compute the exact fitness value on conformance checking stage. This statement is partially true, where the conformance checking on BP involves comparing the expected behavior (*as specified in the process model*) with the actual behavior (*as captured in the event log*), and computing a measure of fitness that represents the degree of conformance between the two representations. The exact computation of this fitness measure can be computationally expensive, especially for large event logs and complex process models. However, it is important to emphasize that the accuracy of the fitness measure can have a significant impact on the usefulness of the conformance checking results. In some cases, an approximate or coarse-grained measure of fitness may be sufficient to identify the most significant deviations of the expected behavior. In other cases, a precise and detailed measure of fitness may be necessary to accurately diagnose and address process performance issues. Therefore, it is important to carefully consider the trade-offs between computational efficiency and accuracy when performing conformance checking on BP, and to select an approach that balances these factors appropriately based on the specific needs of the analysis. Between both those works, the authors in [121], introduce a mixed-paradigm process modeling as a promising approach for BP conformance checking that allows for more comprehensive and adaptable representations of complex BP's. This fundamental assumption of that approach is based on the integration of different modeling languages and techniques, such as declarative and procedural modeling, to provide a more comprehensive representation of BP's. To ensure that business processes

comply with desired performance and quality criteria, this assumption-based approach allows for the use of various conformance checking techniques. Those methods are including model-based and data-driven techniques, which gives users more flexibility in adapting to changing business requirements.

The analysis of the selected works highlights the following seven common properties of validation-based assumptions on BP models.

- **Completeness:** This assumption stipulates that the BP model includes all relevant processes, activities, and events.
- **Consistency:** The assumption that the BP model is internally consistent and free of logical contradictions.
- **Relevance:** The assumption that the BP model is relevant to the business process it represents and is useful for the intended purpose.
- **Accuracy:** The assumption that the BP model accurately reflects the real-world business process it represents.
- **Verifiability:** The assumption that the BP model can be verified through testing, simulation, or other means.
- **Usability:** The assumption that the BP model is user-friendly and can be easily understood and used by stakeholders.
- **Scalability:** The assumption that the BP model can be scaled up or down as needed to accommodate changes in the business process.

By making these validation-based assumptions, BP designers can ensure that the conceived models are reliable and effective for their intended purpose.

- **Anomaly-based assumptions:** The assumption pertaining to anomaly-based behavior during BP modeling is significant in that it assists managers in detecting and reducing potential conceptual hazards. As a result, the overall performance of the BP in question is enhanced, and the target model is significantly refined. By detecting early the anomalies, the corrective actions can be taken to prevent further deviations and improve the process efficiency. In this context, the anomalous BP behavior refer to the idea that certain behaviors or events in a business process may be considered unusual or deviant from the norm. These anomalous behaviors can occur due to errors, fraud, or other unexpected circumstances.

By analyzing the selected works [124, 135–138], We discovered that representing typical behaviors as a first step to manage anomalous behavior during BP modeling is inevitable. This is accomplished by analyzing historical data and identifying patterns of behavior that are considered standard. Once typical behavioral patterns are established, anomalous behavior can be detected by assessing the deviation

of the current behavior from the established norm. Indeed, it is crucial to account for aberrant behavior while modeling BP in order to make sure that systems are reliable and capable of discovering and responding to unexpected events. In many cases, this activity will include manual intervention, such as human review and inquiry, or in other circumstances, automated responses, such as warnings or notifications.

- **Algorithm theory based assumptions:** The foundation of the deployed algorithm impacts considerably the modelization of the BP at hand.

The following hypothesis were highlighted during the examination of the selected works.

- First, the generative models have a substantial impact in solving the unsupervised BP issues as (*model discovery, anomaly detection, etc*). The embedding vectors of those generative models are the key that can be used for automated inference tasks such as trace clustering. [126]
- Second, the CNN's architectures can outperform RNNs for modeling tasks related to the prediction of sequence of activities. [126, 129]
- Third, various BP issues still get inaccurate results even with the out-performance of the employed DL model. This generalization issue is caused by insufficient training data [130].
- The prediction of the future events is mainly based on previously event-log data [131].
- Finally, the RL reinforced by the attention mechanism of the transformer model is the prospect DL application model on real-time management of the BP issues that take a long-term behavior captured on the event-log data. To solve the anomaly detection on BP, the supervised and unsupervised learning techniques are unable to tackle it due to the missing label in the BP description, the authors in [138] suggest a prior knowledge of true anomalies.

4.3.4 The type of the used algorithm

To mine Business Process (BP) models, two distinct types of algorithms are widely employed: Process Mining (PM) algorithms and Deep Learning (DL)-based approaches. PM algorithms, such as Alpha Miner, Heuristic Miner, and Inductive Miner, focus on extracting structured process models from event logs by leveraging dependency relations, frequency-based heuristics, or recursive decomposition techniques. These algorithms provide interpretable representations of processes but often struggle with handling noise, unstructured behavior, and complex dependencies. On the other hand,

DL-based approaches, such as Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Transformer-based models, aim to learn latent patterns in event sequences, enabling tasks like next-activity prediction, anomaly detection, and process optimization. While DL methods excel in handling large-scale and unstructured event data, they often sacrifice interpretability and require extensive labeled datasets for training. The choice between conventional PM and PM-based DL algorithms depends on the specific requirements of the BP mining task, balancing factors like model accuracy, interpretability, scalability, and the ability to generalize across different process environments

The selected works are classified in the two following categories:

- **Conventional PM approaches:** This family of algorithms uses the conventional PM techniques to discover a process model, and then compare it with the original execution data to highlight the differences and make the updates on the discovered model. As an illustration of such approaches, the authors in [27] presents a novel approach of PM, called Map Miner Method (MMM). The goal of this technique is to construct purposeful process models from process logs automatically. MMM studies the link between users' activity records and their intention-fulfilling strategies using Hidden Markov studies. The scalability of earlier methods that infer a process model from such logs becomes a limitation in terms of performance when BP-related event logs become more widely available. Fast response times are especially important for exploratory analysis that looks into the manifold parameter settings of discovery algorithms, maybe with the help of a software-as-a-service tool. Common methods for finding process models, however, always parse and analyze all of the event data that is available, even when a tiny portion of a log might have already produced a high-quality model. The authors in [28] provide a process discovery architecture that uses sampling to drastically decrease the size of an event log and statistical pre-processing of the log.

On the top of that, the heterogeneity of those event-logs data is an another issue on PM approaches. Fani. al [29] applied another approach supported by the ProM framework to discover a straightforward BP model. The findings demonstrated that subgroup identification in the PM domain might yield actionable insights such as trends in deviating situations. Furthermore, the authors of [30] provide a method for overcoming PM's disadvantages by using heuristic mining to extract business process models and a Markov chain as a sequence clustering tool in the data pre-processing step. Furthermore, in many applications, there are many candidate IDs, resulting in diverse perspectives on the same procedure. In this context, one event may be associated to many cases (*convergence*), and several

occurrences of the same action inside a case (*divergence*). The event data must be "flattened" in order to generate a typical process model. There are usually several options, resulting in disjointed perspectives. As a result, the overview is soon lost, and event data must be extracted numerous times (*for different views*). To address this issue, many ways have been presented. The research published in [31] includes fundamental notations as well as a sound discovery methodology for modeling BP awareness and providing methods for classifying event data.

The conformity checking and enhancement techniques are the second application of PM approaches. Recently, there has been a lot of works done regarding the creation of declarative specification-based compliance testing tools. These requirements work well for describing highly variable processes. However, letting multi-perspective requirements remains an outstanding problem in the context of declarative models and conformance verification. This means that declarative models used for conformance verification should include viewpoints on data and time in addition to the control flow to explain the behavior of the process. To close the gap of the conformance checking methodologies, the works exposed in [119–121] suggest efficient approaches and techniques. At the end, we can state that some PM approaches [122–124] are used based on a BP-graph model to enhance the actual discovered model with further performance features extracted from event-log data.

- **PM based DL approaches:** A panoply of algorithms has been proposed in this context and they vary from supervised, unsupervised or semi-supervised to RL based approaches.

The supervised algorithms ranging from RNN, LSTM, the encoder of the transformers, Artificial Neural-Network (ANN), CNNs...etc. Generally, the goal of the supervised algorithms centred toward the predictive and recommendation as well as anomaly detection systems. BINet is a neural network architecture that is proposed by the authors in [135] for real-time multivariate anomaly identification in the gathering of BP event logs. Both the BP's control flow and the associated data are handled by BINet's architecture. Additionally, the authors specify an heuristic approach for automatically setting the threshold of an anomaly detection algorithm. Basing on the proposed heuristic, they demonstrate that BINet can be used to detect anomalies in an event logs, not only for a case level, but also on event attributes level. Furthermore, the authors in [129], prove that the CNNs architectures outperform RNNs when modeling tasks' sequence in order to predict the next activity of a running execution trace. While the previous algorithm employ the row form of the event-log as input of their supervised DL algorithm, Using graph convolutional neural networks (GCNNs), the authors of [33] suggest

a method for training DL-based models that converts an input event log into an accurate Petri-net.

The second type of DL approaches consists of the unsupervised algorithms, which range from the transformers as an encoder-decoder to the generative models. These algorithms aim to discover the BP model, and to check its conformity with the real-life event-log data-set. The authors in [136] provide a novel technique for identifying and evaluating anomalies that arise during a BP execution by utilizing auto-encoders. This approach may be trained on a noisy data-set that contains certain abnormalities beforehand and doesn't rely on any prior information about the procedure. Additionally, in [137], the authors provide a method for filtering anomalous event data in an online PM environment using a variational auto-encoder. This approach helps to improve the outcomes of PM approaches, which in turn improves BP management. Predictive process monitoring, which seeks to forecast the future features of an ongoing process case, such as case conclusion or remaining timestamp, is another application topic for unsupervised algorithms. In [130], the authors provide a unique adversarial training framework to tackle the problem of DL models' inadequate quantity of event-log data-set, which leads to generalization issues. The foundation of this system is a Generative Adversarial Network (GAN) modified for sequential temporal input. To predict the process specification basing on running process instances, the authors in [131] develop a mechanism to predict future events by using previously observed event log data and which are complete. When solving this next event prediction problem, a fully attention-based transformer was employed. As the aim of these unsupervised algorithms is to discover the BP model, the authors in [127] propose an unsupervised algorithm by combining the control-flow and data perspectives under a single roof by extending inductive process discovery.

The semi-supervised algorithms [128, 134, 143] is the third type of the DL algorithms. They combine the supervised algorithms with the unsupervised ones. The operating principle of this family of algorithms consists to generate the non-defined data with the unsupervised algorithms and then to validate them with the labeled data of the supervised algorithms.

Finally, to a real-time mining of the BP model, RL based techniques are employed to optimize the discovered BP models [140, 141], also to predict next activities [133], as well as to detect anomalous behavior of long-term and short-term [138].

4.3.5 The evaluation data

It's obvious that the discovered BP model must be validated. To this end, having an event-log data-set that serve for testing and evaluation the resulting model is inevitable. This data-set can either be synthetic (*artificial*) or real-life event log data. Synthetic event log data is generated by simulating a BP under controlled constraints. This technique allows generating collections of data-sets that can be used for testing and evaluation purposes, as well as understanding how a BP behaves under certain conditions. However, it cannot be utilized to validate the strategy at hand on its own.

On the other hand, using real life event logs data seems more effective and most of the identified works are based on collecting actual BP executions [27–31, 33, 118, 121, 122, 124–134, 136–138, 140–143]. This approach allows for the validation of a BP under real-world conditions and provides a more accurate representation of the actual performance of the BP.

As using both synthetic and real-life event log data can provide a more comprehensive dimension for validating an extracted BP, the authors in [119, 120, 135] profits of this double advantage and propose approaches for evaluating the BP model under controlled conditions, as well as real-world conditions. Such approaches allow, also, comparing the obtained results from both data-sets, which can help to identify any discrepancies or anomalies.

4.4 Synthesis of the selected works analysis

Table 4.1 summarizes the analysis of the identified research works based on the suggested aspects. Each row represents the selected work, alongside the columns indicate the proposed criteria..

In summary, the reviewed studies highlight the diverse methodologies and assumptions applied in process analysis, emphasizing the role of event logs, process models, and various computational techniques in extracting meaningful insights and BP decisions. The following are the findings that were reached:

- The studies leverage different types of input data, ranging from raw event logs to complex workflow models, and generate outputs such as process models, anomaly detection, predictive analytics, and optimization strategies.
- The underlying assumptions vary significantly, addressing challenges like model complexity, behavioral patterns, and stochastic behavior.

- The choice of algorithms spans conventional process mining techniques and machine learning-based methods, reflecting the evolution of computational approaches in the field.
- Evaluation across real-life and synthetic datasets ensures the robustness and applicability of these methods, demonstrating their relevance in real-world scenarios.

Overall, the landscape of process analysis continues to evolve, integrating diverse data sources, analytical frameworks, and evaluation metrics to enhance the accuracy and efficiency of business process management.

TABLE 4.1: Overview of the BP based conventional PM and PM based DL approaches

Study	Input data	Output	The Assumption	Type of algorithm	Evaluation
[27]	Event-Log	BP Model	Intention-oriented Modeling	Conventional PM	Real-life Event-Log
[119]	Event-Log + BP Model	Precision / Fitness measures	Multi-perspective declarative process model	Conventional PM	Synthetic + Real-Life Event-Log
[122]	Petri-net Model	BPMN Model	BPMN modeling elements(sub-processes, cancellations, ..etc.)	Conventional PM	Real-life Event-Log
[125]	Event-Log + Complex BP graph	Optimized BP graph	- Raw models mined from such logs often suffer from the problem of inaccuracy and high complexity - The activities on the same cluster are densely connected	Conventional PM	Real-life Event-Log
[29]	Event-Log	BP Model	Apply subgroup discovery in the PM domain to generate actionable insights like patterns in deviating case	Conventional PM	Real-life Event-Log
[128]	Event-Log	BP behavioral patterns	Behavioral patterns can reduce the complexity of the BP models	PM based DL	Real-life Event-Log
[28]	Event-Log	BP Model	Small fraction of a log could have already led to a high-quality model	Conventional PM	Real-life Event-Log

[120]	Event-Log + BP Model	Precision / Fitness measures	There is no need to spend unnecessary time to compute the exact fitness value	Conventional PM	Synthetic + Real-Life Event-Log
[30]	Event-Log	BP Model	The BP executions suffer from the complexity and dynamics as well as the heterogeneous of its data	Conventional PM	Real-life Event-Log
[135]	Event-Log	List of anomalies	The anomalies in an event logs exist on both a case, and also on event attribute level	PM based DL	Real-life + Synthetic anomalies
[136]	Event-Log	List of anomalies	Generally the anomalies occurred during the execution of the process	PM based DL	Real-life Event-Log
[126]	Event-Log	- Trace clustering - Process model comparison	Embedding vectors can be used for automated inference tasks such as trace clustering, etc	PM based DL	Real-life Event-Log
[31]	Event-Log	BP Model	Object-oriented Modeling	Conventional PM	Real-life Event-Log
[123]	The Workflow BP Model	The Workflow BP Model	Most organizations face substantial differences between the process specifications and the actual run-time behavior	Conventional PM	Real-life Event-Log
[129]	Event-Log	Next predicted activity of a trace	CNNs architectures can outperform RNNs on tasks for sequence modeling	PM based DL	Real-life Event-Log

[140]	Event-Log	Optimized Process Models	Optimize BPs in an environment which includes some degree of uncertainty	PM based DL	Real-life Event-Log
[124]	Event-Log	List of anomalies	Monitoring and detecting anomalies in the enormous log	Conventional PM	Real-life Event-Log
[130]	Event-Log	predictive process	The insufficient training data cause the generalization issue	PM based DL	Real-life Event-Log
[143]	Event-Log	Topic Modeling	Textual reviews, feedback, and support data has driven enterprises to seek and research engagement patterns	PM based DL	Real-life Event-Log
[142]	Event-Log + Petri-net	Generalized Stochastic Petri-net	The stochastic behavior is highly relevant information	Conventional PM	Real-life Event-Log
[121]	Event-Log + Petri-net BP Model	Precision / Fitness measures	Mixed-paradigm process allow capturing complex behavior in a compact way	Conventional PM	Real-life Event-Log
[137]	Event-Log	List of anomalies	An increasing data-centric analysis of processes by using PM techniques has a direct impact on BPM	PM based DL	Real-life Event-Log
[131]	Event-Log	predictive process	Predict events based on previously completed event-log data	PM based DL	Real-life Event-Log

[132]	Event-Log	predictive process	Predicting future characteristics of a running process using event logs	PM based DL	Real-life Event-Log
[141]	Event-Log	An optimal resource allocation policy	Modeling of a multi-process environment with different process-based rewards	PM based DL	Real-life Event-Log
[133]	Event-Log	predictive process	The collaborative and transversal activities in the unstructured BP	PM based DL	Real-life Event-Log
[127]	Event-Log	The Workflow BP Model	Combining the control-flow and data perspectives under a single roof by extending inductive process discovery	PM based DL	Real-life Event-Log
[134]	Event-Log	A prediction BP model	Predicting the final outcome of an ongoing process instance is a key problem in many real-life contexts	PM based DL	Real-life Event-Log
[138]	Event-Log	Long-term and short-term anomalous behavior	- Supervised and unsupervised learning techniques are impracticable because of the miss of labeled BP anomaly data and prior knowledge of true anomalies	PM based DL	Real-life Event-Log

[118]	Event-Log	The Workflow BP Model	- The way in which the BP events are coordinated is not necessary a unique pattern	Conventional PM	Real-life Event-Log
[33]	Event-Log	The Petri-net BP Model	How a given behavioral description of an event-log could be translated into a process model?	PM based DL	Real-life Event-Log
[139]	Event-Log	The Interpretation and recommendation of the process	paves the way for continuous innovation at the intersection of PM and AI, promising to revolutionize the way businesses analyze and optimize their processes.	PM based DL	Real-life Event-Log

To emphasize the research deficit of the current works, the following part will show the road-map and research gap of the existing works.

4.5 Road-map and research-gap

The state-of-the-art analysis reveals a significant lack of the techniques and methods dedicated to enforce decision-making in the BP area. Furthermore, the existing approaches have not yet benefit from the recent AI advances. To address this challenge, in what follows we draw up our road-map that will enable us to address the existing gaps and inadequacies. This road-map will guide us in defining our scientific contributions in the field. Mainly, our contributions will focus on two complementary dimensions. (i) Discovering BP models and enhancing their representation to be suitable for AI techniques deployment in the BP decisions. (ii) using chat-bots techniques to facilitate Decision-making by utilizing natural languages.

These two aspects are deeply discussed in what follows.

4.5.1 Discovering and enhancing BP models for decision making

Nowadays, the research literature demonstrates that KG models combined with Caps-net has significant advantages for DSS [95, 144–147]. Although, a few researchers have attempted to profit from the KG in the BP field, using caps-net is still very limited.

As illustrated in Fig. 4.1, this section first describes related works in the BP field, then it summarizes relevant works on decision making using KG and Caps-net. Finally, it discusses the rare studies that combine the two perspectives.

4.5.1.1 BP mining field

Depending on the BP goals, these works fall into three categories: *BP model discovering*, *BP model optimization* and *decision-making*. These topics are addressed in the following subsections.

- **BP-model discovery** Basing on the analysis 4.3 subsection, two distinguished types of algorithms (*explained in the subsection 4.3.4*) are widely utilized to mine BP models (*explained in the subsection 4.3.2*). This panoply of work has been carried out to allow the extraction of an *abstract model* of a BP from its $\mathcal{EL}$. The discovered models, ranging from *formal to visual representations*, are critical to

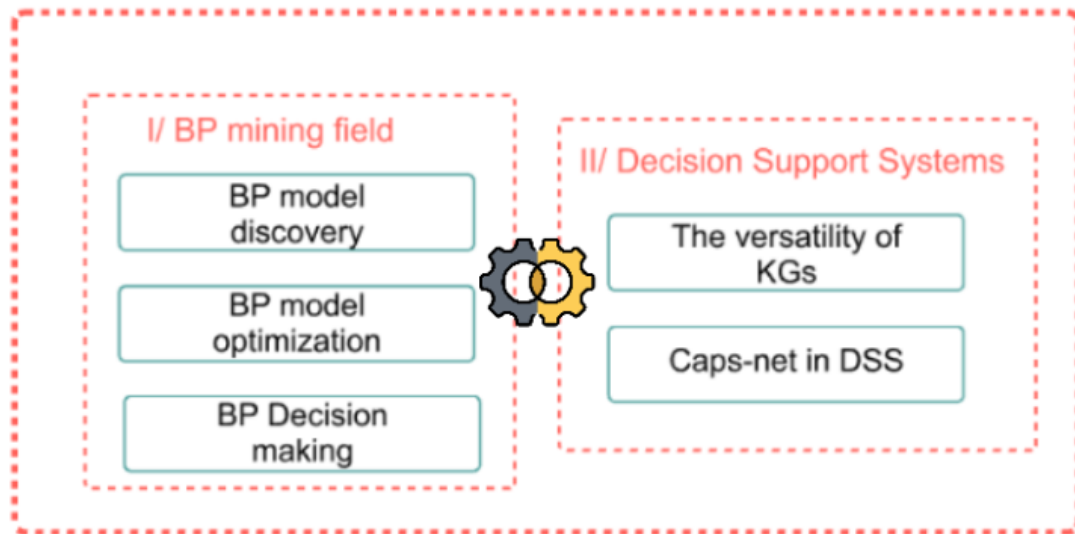


FIGURE 4.1: Problem statement of using Enhanced BP models for decision making in the BP area

support enterprises decision making and to ensure BP integration and cooperation. Unfortunately, these BP mining techniques are extremely complex and difficult for users to comprehend and to manage. In this context, and in order to conduct an efficient mining of BP manageable models, Representation Learning (RL) based techniques are employed to optimize the discovered BP models [140, 141].

To this end, remarkable efforts have been made by the field's research community to formalize a range of methods for optimizing the discovered BP-models. The following subsection expose the BP-model optimization techniques.

- **BP-model optimization** To remain competitive in today's rapidly changing business environment, the optimization of BP models and their stream-less executions provide contemporary businesses many benefits such as: *reducing costs, boosting productivity, enhancing customer satisfaction, and improving the overall performances*. To this end, the goals expected by the BP-model optimization methods are often achieved through process automation and their continuous improvement. Furthermore, the spectacular advancements in recent technologies, such as ML and DL techniques, as well as data analytics have strongly impacted the BP optimization research works.

The analysis of the encountered works explained in 4.3.4, it's worth noting that the more or less optimized models obtained by the previous methods remain limited for automatic decision-making. In fact, in various scenarios, end users are forced to make their decisions manually, or in a semi-automated manner. To overcome this

limitation, a third type of approaches which aim to strengthen decision-making automatically are presented in the following subsection.

- **BP decision-making** According to the preceding analysis explained in the section 4.3, browsing a straightforward BP model to be used in different decision-making contexts, is actually an issue to be addressed by the researcher community in the area. Indeed, instead of building several separate models whose usefulness remains limited to a specific aspect (*recommendation, classification, anomaly detection, ...*), the ideal is to concentrate the design efforts on the development of a rich intrinsic kernel which expresses all the semantics conveyed by the $\mathcal{EL}$ (e.g. *KG-BP*).

The related works of the DSS independently of the BP field are presented in the next subsection.

4.5.1.2 Decision support systems outside the BP field

Nowadays, the combination of the KG and caps-net is gaining great importance in the development of decision-making systems. This section begins by illustrating the importance and the versatility of KGs usage, then continues by presenting caps-nets architecture.

- **The versatility of KGs** Nowadays, a considerable resurgence of interest for using KGs is observed in both research and socio-economic fields. In fact, a large panoply of application domains, including social networks analysis, routing and optimization, data representation as well as a large variety of other application fields benefit from KG models. This enthusiasm for using KGs is justified by the rich theory underlying such abstract data structure (*See [148] for more details*). Further, KGs have strongly contributed to the success of the DL models based-graphs. KG are versatile and can be used in a large variety of domains.

As an illustration, in the e-learning domain the quality of the provided courses is extremely important. Hence, a panoply of KG based applications are geared at facilitating teaching and learning. For example, given the significance of course allocation tasks in universities, the authors in [149] developed a knowledge graph-based course management system for automatic course allocation. They created a course knowledge network including items such as courses, readers, course books, and authors to help students find appropriate courses. The authors in [150] presented KnowEdu, an educational knowledge graph generation system that generates knowledge graphs for school-based learning and teaching.

Furthermore, other scientific KG-based recommender systems, such as citation recommendation, cooperation suggestion, and reviewer recommendation, are proposed by the authors in [151]. For example, [152] developed a KG-based reviewer assignment system to ensure accurate match between reviewers and publications. They combined KGs with recommendation rules to create a rule engine for the recommendation process.

Another very interesting application area of KGs is social networks. The fast rise of social media platforms such as Facebook and Twitter, online social networks have permeated human life and provide several benefits such as social connection creation and convenient information acquisition [153]. Various social KGs are designed and used to examine vital information from a social network. These KGs are often built from people's social media postings and behaviors, and they are used in a variety of applications for various purposes [154].

In this regard, the capsule neural network is one of the most common architectures that uses the KG to make decisions. The following sections detail Caps-net's related works.

- **Caps-net in DSS** The initial appearance of capsule" neural networks [155] have attracted a lot of research interest due to their potential to address Convolution Neural Network (CNNs') representational limitations, which include the loss of information caused by pooling operations.

Basically, capsule networks capture the part-whole interactions due to the transformation matrices between various capsules. Using MNIST image data-set [156], capsules neural network has produced encouraging results when combined with routing-by-agreement [95, 157]. Numerous application domains, on the other hand, began to make advances based on the coupling of caps-nets with other types of neural nets (*e.g.*, *Recurrent Neural Nets (RNN)*, *transformer mechanism*, *..etc*). The following are some notable works.

- **RNN combined with caps-net:** The authors in [144] propose RNN-Capsules for sentiment analysis. The work [145] proposes a bidirectional Gated Recurrent Units (GRU) for text categorization in conjunction with the suggested compositional coding technique. The attention model was employed in the capsule network for aspect-level sentiment analysis based on the RNN [158].
- **CNN combined with caps-net:** Sabour et al. [95] used a convolutional layer and a primary capsule layer in their model, much like in image classification. To do multi-task text classification, MCapsNet combined the convolutional network, primary capsule layer, and representation layer [146]. Unlike

MCapsNet, in [147] paralleled three such architectures with filter windows of 3, 4, and 5 in the convolutional layer.

Since the graph demonstrates the efficacy of DGRL approaches on the one hand, and the routing mechanism shows the success of capsule networks on the other, a new revolution known as capsule neural networks is established based on graph the routing mechanism between capsules. The study [34], is the first to approach the capsule as a node in a graph, with graph routing being used to explore the relation between capsules in the same layer. Along with the attention mechanism, graph routing addresses the drawback of routing that is heavily reliant on lower level attributes.

As shown in figure 4.1, despite the vitality of KGs and caps-net, they are rarely employed in the BP area. The following subsection highlights a number of significant research on the KG through Caps-nets that satisfy the topic of BP.

4.5.1.3 Deploying KG through Caps-nets in BP field

This subsection aims to explain the research gap and the novelty of our suggested model in the BP field

- **Research gap** Although existing works have attempted to leverage KGs to enhance decision-making systems in BP [159–161], their application in BP mining remains limited. Traditional BP mining techniques often struggle to effectively represent and process the intricate and dynamic behaviors captured in EL. KGs, with their structured representation of relationships and data semantics, offer a more robust way to model the complexity of BP logic. However, these conventional KG-based approaches have limitations in capturing hierarchical, spatial, and contextual dependencies within BP workflows.

Capsule Networks (Caps-Nets) and their graph-based extension, G-Caps-NN, address these gaps by providing a novel way to learn and preserve the intricate dependencies and latent features of BP activities. Unlike traditional methods that rely on simple embeddings or flat models, G-Caps-NN dynamically encodes hierarchical relationships and complex patterns in BP workflows, making it more adaptable to variations in EL. This allows for the creation of a Knowledge Graph for Business Processes (KG-BP) that captures the complete behavior of BP logic. By doing so, the KG-BP serves as a foundation for automatic and informed decision-making for business stakeholders, transforming static BP models into adaptive engines for real-time insights and optimization.

To be specific, KGs enhance BP mining by providing a semantic structure to represent BP activities and relationships, while Caps-Nets enable the identification of multi-dimensional dependencies, such as sequence, hierarchy, and context within the BP data. G-Caps-NN further refines this by propagating feature values across nodes in the graph, resulting in enriched event embeddings and a comprehensive BP model. This ensures more accurate and context-aware predictions, empowering businesses to optimize processes and make proactive decisions.

- **The novelty of G-Caps-NN** The proposed DSS4BP framework significantly improves upon conventional BP mining techniques by integrating the structured representation of KGs with the adaptive and hierarchical learning capabilities of G-Caps-NN. Unlike traditional approaches, which often struggle to handle large-scale and heterogeneous data, DSS4BP offers a unified and scalable solution for capturing complex BP behaviors. Its ability to generalize BP logic while addressing real-time decision-making challenges ensures more accurate predictions, better performance, and enhanced usability for business stakeholders—all without overwhelming the user with overly technical complexities.

In this circumstance, *the chapter 5 suggests a novel G-Caps-NN architecture that discover the KG-BP. This KG-BP aims to encompass the complete behavior of the BP logic that is conveyed in the $\mathcal{EL}$. Therefore, the target representation will serve as an automatic BP's decision-making engine for the business stakeholders.*

To tackle the issue of expressing the behavior of the BP contained in an $\mathcal{EL}$, we propose the development of a DSS4BP system based on a G-caps-NN approach. We use a KG representation learning technique that prioritizes embedding knowledge conveyed by the execution traces of the considered BP in a KG basing on G-caps-NN architecture.

4.5.2 Deploying chat-bot techniques for BP decisions

To improve the performance of DSS, the junction of NLP and decision-making processes has benefits from a substantial attention in recent research literature by integrating more innovative approaches. The advent of advanced LMs, particularly, those based on transformers architecture, has propelled this ambition to new horizons consolidated with more possibilities.

In what follows, we review research works having contributed to the improvement of the monitoring tasks related to decision-making in business processes. We examine the most relevant works having tackled the issue of incorporating AI techniques in DSS related to BPs. More specifically, we focus on the integration of chatbot fine-tuned with GPT-3.5-turbo in the DSS context and the induced aspects.

4.5.2.1 Smart BPs decision making

Recent approaches trying to incorporate process mining and DL techniques are gaining increasing attention. In fact, the BP management fields has benefits from a panoply of approaches and techniques that involve making decisions based on specific business rules and resources constraints within a process. Such approaches aim to improve decision-making processes by leveraging large amount of data and various management rules suitable for advanced analytic. By exploiting and analyzing the large amounts of data generated by BPs, such approaches can uncover hidden patterns and insights that may not be apparent through traditional methods.

The data analysis process applies ML algorithms and decision-based approaches. Thus, it can identify patterns [162] and measure the variation between the discovered model and the original data [120, 121]. Other works focuses on making predictions about future outcomes [129–134] and detecting anomalous behavior [124, 135–138]. Finally, the studies conducted by Van der Aalst et al. (2016) have long been instrumental. This work on process mining, consists in extracting insights from event logs, and discovering process models aligns with the goal of enhancing decision-making through a data-driven approach.

The deployment of the previous approaches can lead to more accurate decision-making and better overall performance of the BPs management. Overall, decisive-based approaches on BPs have the potential to bring significant benefits to organizations, including better decision-making, increasing efficiency, and improving competitiveness.

4.5.2.2 Fine-tuned generative models for decision making

Brown et al. [163] presented a seminal work on language models as few-shot learners, revealing the extraordinary capacity of models like GPT-3.5-turbo to generalize knowledge from limited examples. This finding laid the foundation for subsequent studies exploring the applicability of such models in complex decision scenarios. In parallel, fine-tuning, as discussed by OpenAI [164], emerged as a crucial mechanism for tailoring these large-scale language models to specific domains, providing an avenue for leveraging their inherent language understanding capabilities in targeted contexts, such as decision-making. Furthermore, the work of Devlin et al. [165] introduces the influential language model BERT (*Bidirectional Encoder Representations from Transformers*) which emphasizes bidirectional contextual embedding. While our focus lies on GPT-3.5-turbo, the comparative exploration of numerous pre-trained models contributes to a nuanced understanding of the strengths and limitations within the landscape of decision-making

models. Understanding the architectural underpinnings of these models is essential for comprehending their capabilities and limitations, especially in the context of decision-making complexity.

4.5.2.3 Fine-tuned generative models in BP decision making

The transformer models presented in [166, 167] are considered as the pioneering architecture and the cornerstone of the generative models. By integrating fine-tuned generative models into BPs decision-making scenarios, the authors in [168, 169] aim to bridge the gap between traditional decision-making approaches and cutting-edge AI technologies. Building upon foundational works in decision theory and AI, the authors suggest a novel approach which leverage chat-bots empowered by fine-tuned generative models. These advanced language models are able to understand and generate human-like text, allowing to provide sophisticated insights, recommendations, and support within BP decision contexts.

The insufficiency of these two works that they are only proposed ecosystems, without a real implementation in BP decision making area. As we delve into the synthesis of chat-bot and generative models (e.g. GPT-3.5-turbo) within the realm of BPs decision-making, the previous works lay the groundwork for our exploration. By building upon existing knowledge, we aim to contribute to the evolving landscape of DSS and bringing forth novel insights based on an approach that leverages advanced LM in the area of BPs.

To this end, we suggest in the next chapter 6 a framework describing a chat-bot that will be used for supporting decision-making related to BPs

4.6 Conclusion

Nowadays, the multidisciplinary nature of the BP mining field, combined with the emergence of a large panoply of BP mining methods, a wide confusion has appeared in the community. This interference leading to potential difficulties for protocol managers to select and adopt the most appropriate BP mining approach for a particular operational scenario.

This chapter provides a clear qualitative analysis and perceptive overview of BP mining techniques. Thus, the existing works are categorized in five distinguished classes. Further potential research perspectives aiming to enhance BP management systems are identified. In this context, the two chapters target to:

- **The chapter 5** aims to build a new DL approach to discover the semantic BP model of the behavior hidden in the event-log data traces. This chapter suggests a novel G-Caps-NN architecture that discover the KG-BP. This KG-BP aims to encompass the complete behavior of the BP logic that is conveyed in the EL. Therefore, the target representation will serve as an automatic BP's decision-making engine for the business stakeholders. To tackle the issue of expressing the behavior of the BP contained in an EL, we propose the development of a DSS4BP system based on a G-caps-NN approach. We use a KG representation learning technique that prioritizes embedding knowledge conveyed by the execution traces of the considered BP in a KG basing on G-caps-NN architecture.
- **The chapter 6** aims to build a new framework based on a GPT-3.5-turbo. This framework describing a chat-bot that will be used for supporting decision-making related to BPs.

Part II: Contributions

Chapter 5

Towards a BP-DSS through KG Capsule Network

5.1 Introduction

In this chapter, we introduce a novel Decision Support framework for BP mining named (DSS4BP) and the main contributions of the suggested system are the following.

- Using Knowledge Graph to model BP event log data. The resulting graph KG-BP will serve as a foundation for making more informed decisions.
- This KG-BP is discovered by a new DL architecture, named a Graph Capsule Neural Network (G-Caps-NN) and inspired by caps-GNN [34]. Such structure is able to learn the embedded event characteristics of the BP and their correlation degree. These highlighted features provide crucial knowledge for automatically learning a robust KG-BP properties.
- The conducted experiments, on both virtual and real event log data, exhibits the effectiveness and the efficiency of the suggested modeling in the areas of optimization and prediction tasks.

5.2 Development of DSS4BP framework

Nowadays, the AI and ML and in particularity DL enable computers to predict patterns, evaluate accuracy, and continually optimize the process. In 2021, 57% of businesses using AI and ML used the technology to improve customer experience [170]. The adoption of AI techniques is caused by the following cues [171]:

- 65% of companies who are planning to adopt machine learning say the technology helps businesses in decision-making.
- 74% of respondents consider ML and AI to be a game changer, indicating it had the potential to transform their job and industry.

The most popular representation in recent years is the graph representation as it is one of the most important steps in the success of the DL algorithms for DSS. As shown in the Fig. 5.1 [22], Graph Neural Network (GNN) consistently ranked in the top 3 keywords year over year [22]. Further, to examine the versatility of GNN by looking at their impact across different domains of application [22], between them the BP area and computer science for the DSS.

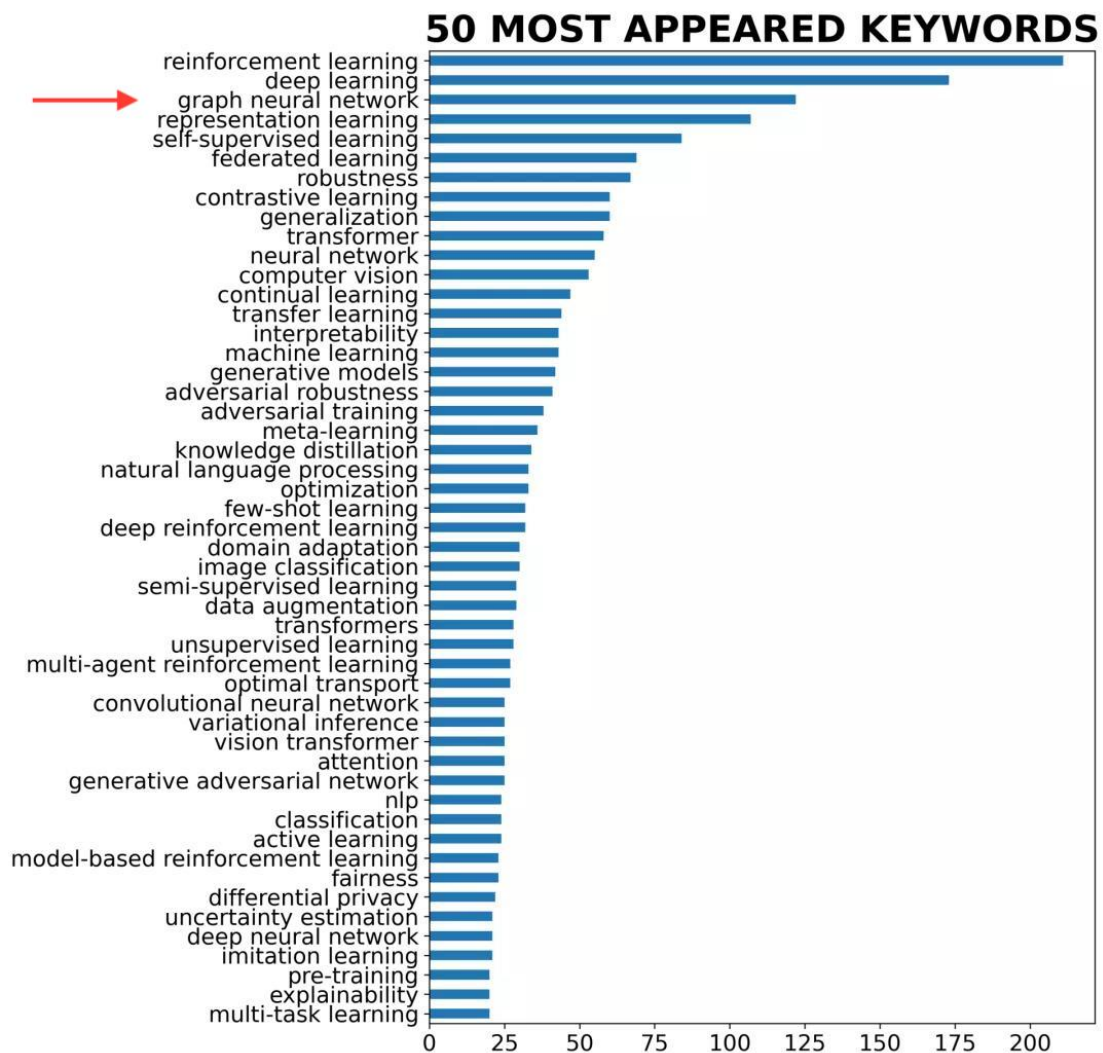


FIGURE 5.1: Graph Neural Networks in the AI Research Landscape [22]

In this context, this section starts first by exposing the KG-BP assumption 5.2.1, then it presents the event-log specification. After that, it illustrates the fundamental principles

underlying the KG-BP model, and terminates by exposing the development of a DL model that uncover the target KG-BP requirements.

5.2.1 The KG-BP's assumption

Often, the behavior of an $\mathcal{EL}$ is modeled by a *transition system* [172] with a number of *states and transitions between them*. However, the assumption made when defining a behavior on an $\mathcal{EL}$ is the major factor in determining what *the states and transitions* express. As a consequence, the two complementary aspects of the homogeneous and asymmetrical behaviors of the $\mathcal{EL}$ data [173] serve as the foundation for our formalized assumption.

Learning a DL model to discover a KG-BP models provide an added-value that allows elaborating an appropriate business decision. Such KG model will acts as a guideline that serves for providing a solid reference for various business decisions activities, such as BP prediction, recommendation systems, classification and clustering.

The authors in [174–176] argue that: "*The conventional process-mining techniques frequently assume that, within any organization, all executions of a particular process are distinguished by a homogeneous behavior, which can be easily compared. This presumption is frequently not true in reality because the same process may have multiple variations (asymmetrical behavior), even within the same organization.*"

Indeed, neither of the two prior assumptions, which are rooted on the homogeneous and asymmetrical process, is wrong. As a sense, they are complementary assumptions, where the same process within the same organization or in another company can be similar or distinct. The process's events, on the other hand, are characterized with the same features describing various values of different attributes, as stored in the executions' traces.

As an illustration, consider the example for managing the bank loans process of different branches. The existence of different ways to invoke this BP is mainly a possible behavior reflecting various company business logic. The key concept here consists to extract an abstract model, represented as a KG, and that can perform the loan process regardless of the branch, not to create the entire set of behaviors.

The first step in modeling this assumption is to comprehend the event-log data described in the following subsection.

5.2.2 Event-Log specification

In the context of BPM environments, an $\mathcal{EL}$ data refers to the set of historical execution traces recorded in the enterprises information systems and which reflect the history of the invoked tasks during a particular BP progression.

TABLE 5.1: A sample retrieved from an event log of the bank loan system

Traces	Activities		Performance features				
Trace ID	Label	Code	Start date	Start time	Duration	Cost	...
1	Specify the loan amount	A	18/1/2022	8:30	30min	20	...
	Choose the loan type	B	18/2/2022	9:30	20min	40	...
	Revise loan conditions	C	19/2/2022	8:30	60min	5	...
	Validate the loan	D	20/2/2022	8:30	25min	10	...
2	Specify the loan amount	A	30/3/2022	10:00	15min	2	...
	Revise loan conditions	C	30/3/2022	11:25	5min	30	...
	Validate the loan	D	30/3/2022	11:30	19min	4	...
3	Specify the loan amount	A	18/1/2022	8:30	5min	20	...
	Run the numbers	D	18/1/2022	8:35	45min	1	...
	Revise loan conditions	C	20/1/2022	8:35	10min	5	...
	Validate the loan	D	20/1/2022	8:55	30min	30	...

Further to the first simple example of the $\mathcal{EL}$ explained in the table 1.1 of the chapter 1, let's consider the following **table 5.1** which depicts an $\mathcal{EL}$ excerpt of a bank data expressing the loan procedure execution. We notice that events with the same trace ID are belonging to the same execution trace. The previous $\mathcal{EL}$ may include a very huge number of traces and each of which constitutes a separate execution case of the considered BP.

In this context, extracting the general semantic of the process hidden behind the execution traces of an $\mathcal{EL}$ is the prerequisite inevitable for elaborating the KG-BP graph formalization.

We introduce bellow, the necessary definitions and notations related to execution traces.

- Let $\mathcal{P}$ be a particular BP and $\mathcal{EL}$ its event log data.
- Let $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$ be the universe of execution traces in $\mathcal{EL}$, and let $\mathcal{E} = \{e_1, e_2, \dots, e_m\}$ be the set of events (*activities*) in $\mathcal{EL}$.
- The set of events in a trace t_i are executed in chronological order over $\mathcal{P}$ to express the historical activities' sequences achieved by different customers or stakeholders. Each trace t_i contains a set of events $\{e_{i1}, e_{i2}, \dots, e_{im}\}$, each of which is designated by e_{ij} , and it denotes the event item $e_j \in \mathcal{E}$ executed within the trace $t_i \in \mathcal{T}$.

TABLE 5.2: The Notations used throughout the paper with examples

Notations	Descriptions	Examples
$\mathcal{P}$	A particular BP	The BP of <i>loan bank system</i>
$\mathcal{EL}$	The event-log data	The table 5.1 is <i>an event-log data</i> of BP named $\mathcal{P}$
$\mathcal{T} = \{t_1, t_2, \dots, t_n\}$	The set of traces of the $\mathcal{EL}$ data	The three traces of $\mathcal{EL}$ data = $\{TraceID = 1, 2, 3\}$
$\mathcal{E} = \{e_1, e_2, \dots, e_m\}$	The set of events contained in the $\mathcal{EL}$ data	The four events of $\mathcal{EL}$ data = $\{A, B, C, D\}$
$t_i = \{e_{i1}, e_{i2}, \dots, e_{im}\}$	The set of events executed through the trace t_i	- The execution of the trace 1 is: $t_1 = \{e_{11}, e_{12}, e_{13}, e_{14}\} = \{A, B, C, D\}$ - The execution of the trace 2 is: $t_2 = \{e_{21}, e_{22}, e_{23}\} = \{A, C, D\}$ - The execution of the trace 3 is: $t_3 = \{e_{31}, e_{32}, e_{33}, e_{34}\} = \{A, D, C, D\}$
$\mathcal{F}(t_i) = \{\mathcal{F}(e_{i1}), \mathcal{F}(e_{i2}), \dots, \mathcal{F}(e_{im})\}$	The feature vector of the trace t_i	- The feature vector of execution trace 1 is: $\mathcal{F}(t_1) = \{\mathcal{F}(e_{11}), \mathcal{F}(e_{12}), \mathcal{F}(e_{13}), \mathcal{F}(e_{14})\}$ $\mathcal{F}(t_1) = \{\mathcal{F}(A), \mathcal{F}(B), \mathcal{F}(C), \mathcal{F}(D)\}$ - The feature vector of the execution trace 2 is: $\mathcal{F}(t_2) = \{\mathcal{F}(e_{21}), \mathcal{F}(e_{22}), \mathcal{F}(e_{23})\}$ $\mathcal{F}(t_2) = \{\mathcal{F}(A), \mathcal{F}(C), \mathcal{F}(D)\}$ - The feature vector of the execution trace 3 is: $\mathcal{F}(t_3) = \{\mathcal{F}(e_{31}), \mathcal{F}(e_{32}), \mathcal{F}(e_{33}), \mathcal{F}(e_{34})\}$ $\mathcal{F}(t_3) = \{\mathcal{F}(A), \mathcal{F}(D), \mathcal{F}(C), \mathcal{F}(D)\}$
$\mathcal{F}(e_{ij}) = [\mathcal{F}_{date}(e_{ij}), \mathcal{F}_{time}(e_{ij}), \mathcal{F}_{dur}(e_{ij}), \mathcal{F}_{cost}(e_{ij})]$	Represents the feature vector of each event j in the trace i	- The feature values of event 1 in the trace 1 are: $\mathcal{F}(e_{11}) = \mathcal{F}(A) = [18/01/2022, 08 : 30, 30, 20]$ - The feature values of event 2 in the trace 1 are: $\mathcal{F}(e_{12}) = \mathcal{F}(B) = [18/02/2022, 09 : 30, 20, 60]$ - The feature values of event 3 in the trace 1 are: $\mathcal{F}(e_{13}) = \mathcal{F}(C) = [19/02/2022, 08 : 30, 60, 5]$ - The feature value of event 4 in the trace 1 are: $\mathcal{F}(e_{14}) = \mathcal{F}(D) = [18/01/2022, 08 : 30, 25, 10]$

- We associate to each event e_{ij} a feature vector denoted with $\mathcal{F}(e_{ij})$, which represents the relevant properties of the event e_{ij} . Hence, the trace t_i will be characterized by a feature vector, denoted by $\mathcal{F}(t_i)$ that is composed of a set of rows expressing the features of each event, i.e.; $\mathcal{F}(t_i) = \{\mathcal{F}(e_{i1}), \mathcal{F}(e_{i2}), \dots, \mathcal{F}(e_{im})\}$.
- The feature vector $\mathcal{F}(e_{ij})$ specify the set of the performance attributes, such as *start time, start date, duration, cost and so on*. We designate by $\mathcal{F}_a(e_{ij})$ the value of the feature a in the event e_{ij} of the trace t_i . For example, considering the previous four attributes, the following formulations are obtained: $\mathcal{F}_{date}(e_{ij})$, $\mathcal{F}_{time}(e_{ij})$, $\mathcal{F}_{dur}(e_{ij})$ and $\mathcal{F}_{cost}(e_{ij})$. Thus, the feature vector of the event j of the trace i is expressed by: $\mathcal{F}(e_{ij}) = [\mathcal{F}_{date}(e_{ij}), \mathcal{F}_{time}(e_{ij}), \mathcal{F}_{dur}(e_{ij}), \mathcal{F}_{cost}(e_{ij})]$.

The **table 5.2** illustrates the preceding specifications throughout the *loan banking* system.

5.2.3 Specification of KG-BP model

The KG-BP architecture is articulated around two basic concepts: *the semantic nodes* and *the semantic transitions* which are defined bellow.

- **Semantic node:** A semantic node (SN) refers to an event's unified features of a particular event which is executed over a set of execution traces $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$. Hence, the SN corresponds to a vector, denoted with $\mathcal{F}(e_{*j})$ (for $j \in [1...m]$), and it represents the vector expressing the mean features' values of the considered event belonging to the set of traces (for $i \in [1...n]$).

By considering the four previous performance parameters, a SN is expressed with $[\mathcal{F}_{date}(e_{*j}), \mathcal{F}_{time}(e_{*j}), \mathcal{F}_{dur}(e_{*j}), \mathcal{F}_{cost}(e_{*j})]$.

For example, consider the event A of table 5.1 and its corresponding features in the three respective traces.

- $\mathcal{F}(e_{11}) = \mathcal{F}(A) = [18/01/2022, 08:30, 30, 20]$
- $\mathcal{F}(e_{21}) = \mathcal{F}(A) = [30/03/2022, 10:00, 15, 2]$
- $\mathcal{F}(e_{31}) = \mathcal{F}(A) = [18/01/2022, 08:30, 5, 20]$

In this case, $\mathcal{F}(e_{*1})$, also noted $\mathcal{F}(A_*)$ is equal to the vector $[22/02/2022, 09:20, 16.66, 14]$.

As shown in the figure 5.2, the feature vector $\mathcal{F}(A_*)$ in the KG-BP represents the mean accomplishment of the event A in the whole traces t_j ($j \in [1...3]$) of the $\mathcal{EL}$ data.

The KG-BP to be constructed must optimize the distance between the different nodes of the graph and the specified semantic node. Thus, the primary goal of the G-Caps-NN architecture, which is exhibited in the next subsection 5.2.4, is to reduce the error or distance to the event's mean vectors $\mathcal{F}(e_{*j})$ that will serve as the neuron capsules of the proposed architecture.

- **Semantic transition:** Semantic Transitions (ST) in the KG-BP characterize the weight reflecting the strength for moving from a semantic node SN $\mathcal{F}(e_{*j})$ to another one $\mathcal{F}(e_{*j'})$. Such transitions are noted $\mathcal{W}_{\mathcal{F}(e_{*j}).\mathcal{F}(e_{*j'})}$ and they ensure switching between semantic nodes of the graph, while progressing in accordance with the supported business logic.

In the KG-BP, the set of semantic nodes ST represents the weighted matrix between the different G-Caps-NN capsules, where each element $\mathcal{W}_{F(e_{*j}).F(e_{*j'})}$ is a pondered value. As a sample illustration, the mean accomplishment of the event A in the whole $\mathcal{EL}$ of the bank loan of table 5.1 is $\mathcal{F}(A_*)=[22/02/2022, 09:20, 16.66, 14]$ and the consistency value to transit to the event B is around 33 %. Thus, $\mathcal{W}_{A,B}$ is equal to 33%.

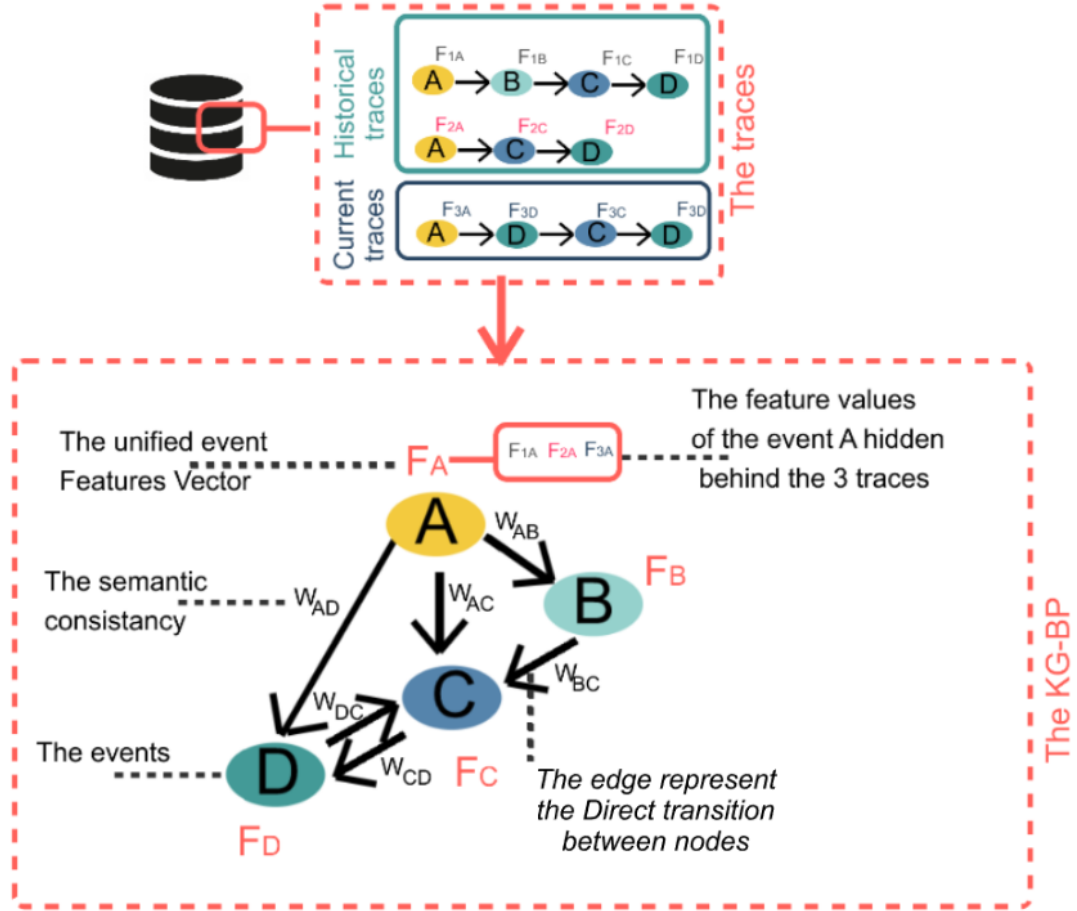


FIGURE 5.2: The KG-BP representing the loan BP

As it's observed in the figure 5.2, modeling SN and ST is able to capture the unified execution of the process, through the overall executions traces of the $\mathcal{EL}$ data.

In fact, building such a KG basing only on real execution traces is a hard task. Consequently, using AI techniques, specifically DL methods, is a very promising avenue. In fact, a significant step during the KG-BP construction process consists to learn graph parameters through the G-Caps-NN algorithms. Further, it is worth emphasizing that the resulting KG is essential for achieving BP mining which will serves as a decision-support tool. This KG-BP helps the decision makers through the analysis of the *consistency weights between the event features*, while guiding them to identify the reasons that cause

particular bottlenecks, overload and delay during the executions of BP activities. Thus, the conceived KG-BP helps managers operating more informed decision regarding the process optimization.

The next subsection covers the G-Caps-NN layers based on the customized graph routing mechanism that is used to discover the KG-BP model.

5.2.4 Graph capsule neural net discover KG-BP

The capsule networks [157] and their underlying routing mechanism are recognized as the mostly deployed models for image processing, and to conduct a coherent image detection, many capsules were necessary. On the other hand, in the text mining area, analyzing the intra-relationship between capsules is an inevitable step, and the authors in [34] present the first study on intra-relationship learning for text data routing.

To the best of our modest knowledge, no study has been conducted to develop the concept of learning, both the unified event characteristics of a BP and their intra-relationship, basing on the routing mechanism. To overcome this deficiency, we propose a G-Caps-NN architecture to build the KG-BP reflecting the BP model conveyed in the set of execution traces.

The suggested G-Caps-NN architecture is inspired from the works described in [34, 177]. This model of G-Caps-NN is articulated around *two modules*, namely: the *Primary Capsule Module (PCM)* and the *Capsule Network Module (CNM)* (see figure 5.3). The PCM represents the embedding input vectors and CNM represents the layers of the DL architectures. The functioning and interaction between these layers are exposed bellow.

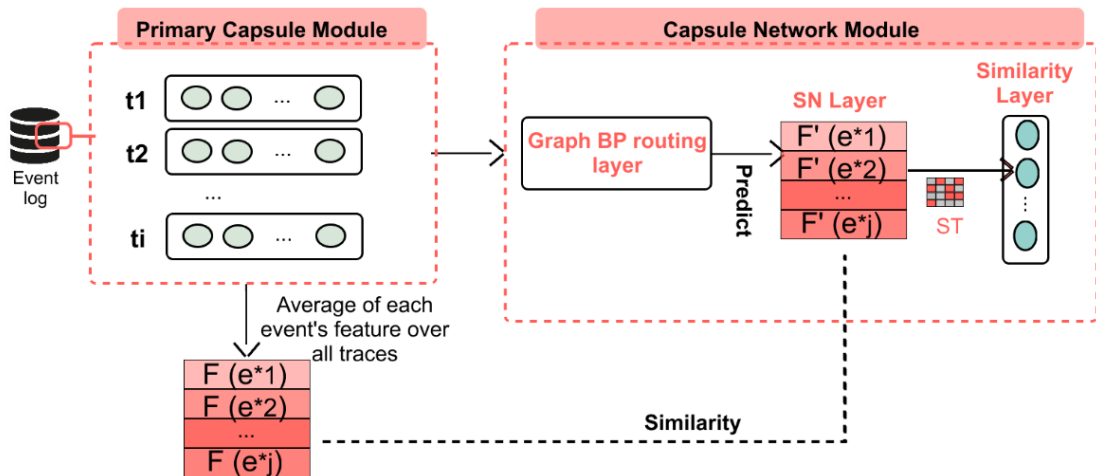


FIGURE 5.3: Graph capsule neural network architecture

5.2.4.1 Primary capsule module (PCM)

A trace $t_i \in \mathcal{EL}$ is composed of a set of events, i.e.; $t_i = \{e_{i1}, e_{i2}, \dots, e_{im}\}$. Each event e_{ij} is characterized by its specific features in the trace t_i , i.e.; $\mathcal{F}(t_i) = \{\mathcal{F}(e_{i1}), \mathcal{F}(e_{i2}), \dots, \mathcal{F}(e_{im})\}$. Hence, the handling of the corresponding values of each trace t_i allows feeding the input capsules in the PCM. For example, the three following capsules $\{F(t_1), F(t_2), F(t_3)\}$ of the loan bank $\mathcal{EL}$ constitute the first layer of the PCM module. The features values of each trace are described in the 6th and 7th row of the table 5.2.

At this stage, the CNM aims to discover KG-BP over the previous introduced set of input capsules. In this perspective, the SN and ST are the two parameters to be learned by the CNM module. The first one (SN) refers to an event's unified features of an event $\mathcal{F}(e_{*j})$ that is executed over a set of execution traces $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$. And the second (ST) expresses the strength and the consistency of the action to move from the SN denoted $F(e_{*j})$ to the $F(e_{*j'})$ one. The subsequent section goes into depth about how the CNM module assesses these two parameters.

5.2.4.2 Capsule network module (CNM)

As shown in the Fig.5.3, the CNM is composed of three capsule layers that aims to discover the KG-BP from an input $\mathcal{EL}$ data. These layers are described in what follows.

- **Graph routing layer (GRL)** As shown in the figure 5.4, the GRL is the first layers of the CNM that aims to predicts the unified event features $F'(e_{*j})$ from a set of input traces t_i . The two following stages are the pillars of our GRL.
 - Propagating the features of an input capsule e_{ij} by considering the weighted mean of its features vector ($\mathcal{F}(e_{ij})$) through a set of traces t_i , according to **equation (5.1)**.

$$\mathcal{F}'(e_{*j}) = \frac{\frac{1}{i} \sum (\mathcal{F}(e_{ij}) * W_{i,j}) * \text{Log}(d + 1))}{\text{Log}(d_j + 1)} \quad (5.1)$$

Where, $\mathcal{F}(e_{ij})$ is the feature of the event j at the set of traces t_i . $W_{i,j}$ represent the weight of the node j at the trace i , d describes the total degree of the whole KG, and d_j is the degree of the node j .

- The second step consists to apply the squash function formulated in **equation (5.2)**, in order to adjust the value of the unified vectors $F'(e_{*j})$.

$$F'(e_{*j}) = \frac{\|F'(e_{*j})\|^2 * F'(e_{*j})}{(1 + \|F'(e_{*j})\|^2) * \|F'(e_{*j})\|} \quad (5.2)$$

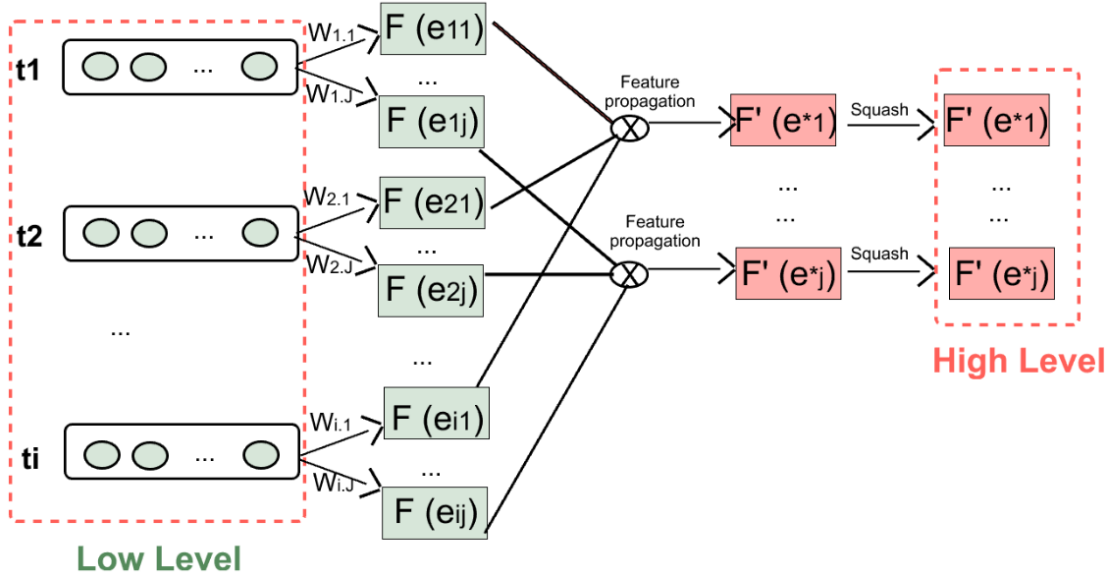


FIGURE 5.4: Graph BP routing layer

The resulting vector $F'(e_{*j})$ represents the unified (*global*) representation encapsulating the behavior of the whole $\mathcal{EL}$ data.

- **SN layer (SL)** represents the predicted unified event features $F'(e_{*j})$ obtained by the GRL,
- **Similarity layer (SML)** the next step, aims to evaluate the predicted values $F'(e_{*j})$ and their consistency with real values. To do that, the *Similarity layer* is composed of j events and the *ST* matrix is the weighted edges between the *SN* and *SML* layers.

This matrix aims to learn the similarity between the predicted features the event j noted $F'(e_{*j})$ and the target $F(e_{*j})$, that represents the mean of each event feature through a set of traces as expressed in the equation 5.3.

$$\mathcal{F}(e_{*j}) = \frac{1}{i} \sum_{i=1..n} (\mathcal{F}(e_{1j}), \mathcal{F}(e_{2j}), \dots, \mathcal{F}(e_{ij})) \quad (5.3)$$

Further, the *ST* matrix aims to calculate the pondered similarity between each predicted event $F'(e_{*j})$ and its j' adjacent events noted $F(e_{*j'})$.

At this point, our picks is that each node of the j top-level capsules of the SML layer assesses the high similarity instantiation between the predicted target $F'(e_{*j})$ with its target $F(e_{*j})$ and its neighboring events $F(e_{*j'})$. The margin loss converge training the output capsules to 1 since the best similarity is 100%.

To highlight the feasibility and the applicability of the proposed KG-BP architecture, in what follows we present and discuss some experimental results.

5.3 Experimental results

For evaluating the KG-BP learnt using the G-CAPS-NN to determine the automated BP decision, we start by explaining the $\mathcal{EL}$ data-set utilized to experiment the proposed architecture, then we describe the evaluation setup to be conducted.

5.3.1 BPI19 event-log description

We used the purchase data set provided by the 9th International Business Process Intelligence Challenge (*ICPM on 2019*) named BPI19 [178]. This challenge provides participants with a real-life data over 1,5 million events originated from a large multinational company operating from the Netherlands in the area of coatings and paints, and challenges them to analyze these data using whatever techniques available, focusing on one or more of the process owner's questions or proving other unique insights into the process(es) captured in the event log. Participants are asked to investigate the purchase order handling process for some of its 60 subsidiaries. In particular, the process owner has compliance questions.

5.3.2 The evaluation setup

We start the experimental process by examining the G-CAPS-NN model using *the purchase BPI19 event-log data* described in [178]. As this paper represents the first strive towards a BP-DSS through knowledge graph capsule neural net, we employed this BPI19 event-log since its complexity, consistency and high level of size. To understand the power of the size of the event-log data, four virtual event-logs of varying sizes are used by Splitting the complete BPI19 event log data, where the loss and accuracy are measured. Next, using both virtual and real data, the learned KG-BP graph efficiency is evaluated. The fields of optimization and prediction application serve as the foundation for the experimentation. Lastly, we describe and explore our proposal's limitations and future prospects.

The next subsections expose a thorough analysis, some experimental results and discussions of the proposed architecture.

5.3.3 The G-Caps-NN evaluation

This section aims to evaluate the G-Caps-NN architecture through the evaluation of the training stage through 3 performance metrics. Then the ablation analysis is discussed.

5.3.3.1 The evaluation of the G-CAPS-NN training

To investigate the robustness of the G-CAPS-NN model, we trained it with the complete BPI19 event-log and the four virtual ones, splitted from the BPI19, having different complexities degrees. As shown in the table 5.3, we conclude that:

- The accuracy of the G-CAPS-NN model trained in the complete BPI19 and the four virtual ones have lead to 97% accuracy in average, and for a 99% of accuracy at the complete BPI19 in particular. These findings demonstrate the effectiveness of the proposed model, where this learning results shows that the model has the ability to establish the relationship between the semantic BP nodes by aggregating its characteristics using the squash routing algorithm.
- The efficiency of the G-CAPS-NN is further demonstrated by training progressively $\mathcal{EL}$ s, as claimed by the authors in [28], Who assumed that relatively a fraction of a log could already build a high-quality BP model.

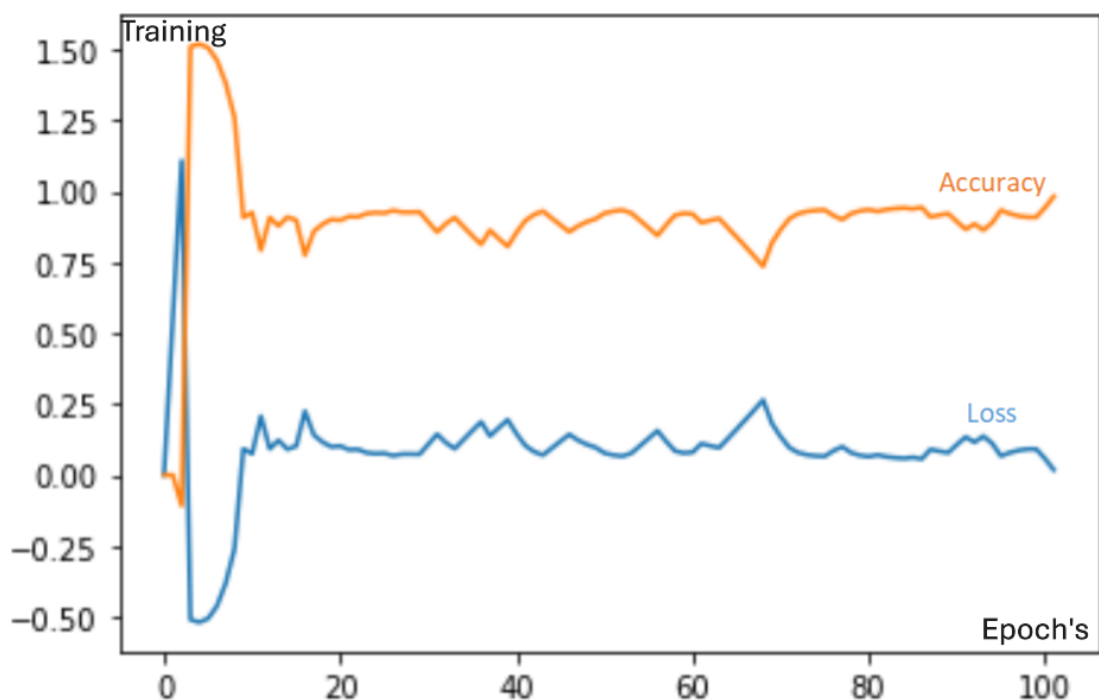


FIGURE 5.5: Training the G-CAPS-NN with the complete BPI19 $\mathcal{EL}$ data

The training of 100 epochs of the complete BPI19 $\mathcal{EL}$ is shown in the Fig.5.5. This training shows the effectiveness of incorporating KG behind the layers of our proposed G-Caps-NN layers in order to discover the KG-BP. As shown in the Fig.5.5, the reason why the accuracy starts from 1.5 and then converges to around 0.99 when embedding the G-Caps-NN by the KG-BP lies in the nature of the initial knowledge provided by

TABLE 5.3: The G-CAPS-NN Results through 04 metrics

Samples	sample 1	sample 2	sample 3	sample 4	Complete BPI19	Average
The size	22	194	1338	107429	1595922	340981
The accuracy	0.90800	0.98500	0.98800	0.98100	0.99870	0.97214
The loss [179]	0.09200	0.01500	0.01200	0.01900	0.00130	0.02786
The MSE [179]	0.023	0.01500	0.01200	0.018	0.044	0.044
The RMSE [179]	0.153	0.123	0.10	0.13	0.209	0.143

the KG and the goal of G-Caps-NN, which prioritizes generalization rather than strictly learning from existing event data. In the following we will explain the cues that the G-Caps-NN models starts the training from values higher than 1 and converges in 0.99:

- **Starting Accuracy 1.5:**

- The KG-BP embedding provides a highly structured representation of BP logic before the training begins. This is because the Knowledge Graph already encodes meaningful semantic relationships, dependencies, and patterns between the events in the BP workflows.
- As a result, G-Caps-NN does not start "from scratch." Instead, it begins training with an advantage: the model has prior knowledge that allows it to make fairly accurate predictions or representations even at the initial stages. This can lead to an initial performance metric (e.g., accuracy) that appears to surpass the typical bounds of models starting at random (accuracy closer to 0.5).

- **Convergence to 0.99:**

- As training progresses, the G-Caps-NN fine-tunes itself to improve its understanding of event generalization. The KG embedding assists the network in avoiding overfitting to specific event logs by focusing on broader BP logic and relationships.
- The model shifts from using static knowledge (from KG-BP) to learning nuanced patterns and exceptions in the data, which may cause the accuracy to stabilize closer to 0.99, reflecting real-world uncertainties and data imperfections. The slight drop from an inflated starting accuracy ensures the model does not over-rely on pre-learned patterns but adjusts to achieve generalization over unseen data.

- **Generalization vs. Training for Existing Events:**

- The primary goal of embedding the G-Caps-NN with KG-BP is to generalize BP logic rather than solely training on and memorizing the events in the dataset.

- Traditional models focus on fitting existing events, which can lead to high training accuracy but poor generalization on unseen data. In contrast, the KG-BP ensures the model learns a broad and adaptable BP structure while using the G-Caps-NN to refine event-level embeddings dynamically. This approach ensures the model performs consistently across diverse workflows rather than achieving perfect accuracy on a specific dataset.

According to the previous discussions, the following results are conducted:

- The suggested model may identify the $\mathcal{EL}$'s hidden behavior in the early training phases and subsequently optimize the model to provide the most effective KG-BP.
- Starting with an accuracy ≥ 1.0 and converging to 0.99 demonstrates that the pre-trained KG embedding provides a robust foundation, enabling the G-Caps-NN to skip redundant learning steps and focus on event abstraction and generalization.
- This ensures better scalability, transferability, and real-world applicability of the model, where the goal is not just fitting the data but building an adaptable decision-support framework for Business Process mining.

The results found in the training stage proves the efficiency of our G-Caps-NN and the KG-BP data structure. However, since the proposed framework contains several components ranging from the KG-BP, Capsnet and G-Caps-NN, the ablation analysis discussed in the following is essential.

5.3.3.2 The ablation analysis of the G-Caps-NN

Ablation analysis [180] is a systematic method used in machine learning to evaluate the contribution of different components of a model by selectively removing or modifying them. By comparing the model's performance with and without specific components, researchers can assess their impact on overall effectiveness. This technique helps identify redundant or essential parts of a model, guiding optimization and interpretability.

In this context, the ablation analysis of the G-CAPS-NN framework is crucial for evaluating the individual contributions of its core components, such as the Capsule Network structure, KG embeddings, and the integration of BP-specific features. To assess the significance of these elements, specific experiments are conducted by systematically removing or replacing them. For instance, the following steps are conducted:

TABLE 5.4: Performance Comparison of Different Model Configurations

Model Configuration	Error	Accuracy	MSE	RMSE
GNN Neural Nets trained on KG-BP	0.023	0.977	0.012	0.108
CapsNet without dynamic routine trained on KG-BP	0.105	0.895	0.052	0.229
G-CAPS-NN trained on raw EL data	0.029	0.971	0.015	0.121
G-Caps-NN trained on KG-BP	0.001	0.998	0.023	0.153

- Removing the KG embeddings and training the G-CAPS-NN on raw event log data:** The goal of this step is to demonstrate how the knowledge graph contributes to learning richer and more interconnected event representations. As shown in the table 5.4, when the KG component is removed, the accuracy (97.1%) is comparable to the GNN approach but slightly lower than the full G-Caps-NN model trained through KG-BP. The performance indicates that the Capsule Network alone can perform well but does not reach the same level of accuracy and error reduction as when KG-BP is included.
- Replacing the Capsule Network with a standard Graph Neural Network (GNN):** The goal allows us to evaluate the role of capsule structures in capturing complex dependencies and hierarchical relationships within BP data. As shown in the table 5.4, the integration of KG-BP with a GNN provides good accuracy (97.7%) and low error rates. However, the RMSE is slightly higher compared to the performance of the G-Caps-NN model. This shows that while GNNs can effectively utilize KG embeddings for decision-making, their performance falls short in capturing more complex dependencies or relationships within the BP logic.
- removing the dynamic routing mechanism within the Capsule Network:** reveals how critical this mechanism is for maintaining consistent performance under varying input distributions. As shown in the table 5.4, by replacing the GNN with a simple Capsule Network (without customization), accuracy drops significantly to 89.5%, with a notable increase in error (0.105) and RMSE (0.229). This suggests that while capsule networks have potential, they require proper customization and tuning (*such as the dynamic routing mechanism*) to achieve optimal performance in BP mining tasks.
- KG-BP with G-Caps-NN:** As shown in the table 5.4, the integration of KG-BP with the G-Caps-NN achieves the best performance across all metrics, with the highest accuracy (99.8%) and the lowest error (0.001). This highlights the synergy between the KG embeddings and the G-Caps-NN architecture, which allows for better generalization and the ability to capture complex dependencies in the BP logic.

From the above analysis the following conclusions are made:

- **Impact of KG-BP:** Including KG-BP significantly boosts performance, as seen when comparing configurations with and without KG-BP. This highlights the importance of using knowledge graph embeddings for BP decision-making.
- **Capsule Network Tuning:** A simple capsule network without a customized dynamic routine performs worse than GNNs, indicating the need for customization to unlock its full potential.
- **Superiority of G-Caps-NN:** The G-Caps-NN, with KG-BP, clearly outperforms other configurations, proving its capability to handle complex BP tasks and relationships more effectively than simpler models.

These findings confirm that each component plays a critical role, and the proposed KG-BP + G-Caps-NN framework achieves state-of-the-art performance through their combined strength.

However, as the results generated should be explained based on the study issues and the findings of the past studies, the next subsection aims to compare and discuss the superiority of the G-Caps-NN with the previous related works.

5.3.4 Comparison of the G-Caps-NN with the past studies' findings

To evaluate this measured efficiency of G-CAPS-NN, the table 5.5 illustrates the accuracy of the G-Caps-NN model with three similar works having employed the same complete BPI19 $\mathcal{EL}$ data. We choose [33] as the sole work done to discover the BP model through GNN. This comparison is based on the produced Output (**O**) and the accuracy value (**ACC**). The comparison of works in table 5.5 yielded the following results:

- The G-Caps-NN has high accuracy of 99% to discover the complete BP model comparing with the three other related works. The goal of the KG-BP is different from the discovered models in the related works. The KG-BP aims to be useful for any further decision, e.g. the next event prediction, the recommendation, anomaly detection ... etc.
- The G-Caps-NN and GNN are in the same category of graph approaches, where the suggested architecture has a high accuracy of 99% comparing to GNN of 95%. Furthermore, both split miner and the generative DL have an accuracy of 98%.

The squash routing that aim to discover the relationship between the event features is the one who give the proposed approach the power comparing to the three other methods.

The accuracy of the training phase and its comparison with three key studies validate the effectiveness of the G-Caps-NN as a first step in the BP domain. However, since the accuracy alone cannot be used to assess the efficacy of the suggested strategy, our experiment will continue rather than stopping with a discussion of these numerical findings.

TABLE 5.5: Comparison of G-Caps-NN accuracy with related works

Goal	BP model discovery											
Model	G-Caps-NN			GNN [33]			Split-miner[49]			Generative DL[181]		
Criteria	I	O	ACC	I	O	ACC	I	O	ACC	I	O	ACC
Value	$\mathcal{EL}$	KG-BP	0.99	$\mathcal{EL}$	Graph	0.95	$\mathcal{EL}$	BPMN	0.98	$\mathcal{EL}$	Process Model	0.98
Assumption	How a KG-BP of $\mathcal{EL}$ can be discovered by the G-Caps-NN as a foundation to further automate decision-making?			How a given behavioral description of an $\mathcal{EL}$ could be translated into a process model?			Aim to produce models that not poorly fit the $\mathcal{EL}$ (low fitness) or over-generalize it (low precision)			a Process Model is extracted via discovery PM techniques, and a DL model to generate timestamped event sequences		

In our opinion, it is more crucial to ensure the scalability of the suggested G-Caps-NN model by running the following research question via a number of validation metrics:

Is the KG-BP model semantically correct for making a further decisions?

To answer this question, the KG-BP optimization and event prediction perspectives are evaluated and discussed below through different metrics.

5.3.5 The KG-BP scalability through different performance metrics

To examine the scalability of the KG-BP model, we tested it in at least two application domains. Thus, the optimization findings of the KG-BP and its event prediction capabilities are experimented with in this subsection in an attempt to reach decision-making objectives.

5.3.5.1 KG-BP optimization

TABLE 5.6: The message under clusters

Message under clusters	Sample 1	Sample 2	Sample 3	Sample 4	Complete BPI19
Pearson metric [182]	72%	62%	79%	79%	67%
Spearman metric [182]	60%	51%	60%	59%	61%
Kandalltau metric [182]	57%	49%	58%	56%	63%
Cosinus similarity metric [183]	76%	68%	81%	80%	72%

Due to the great performance of k-means technique for dealing with clustering problems, across a variety of application areas [78], it's employed in this study to optimize and cluster the KG-BP feature nodes learned by the G-CAPS-NN. At this crucial stage, the similarity under-cluster and distance between clusters are measured and compared to check whether the clusters are semantically accurate.

We can plainly conclude from the above table findings that the suggested model is an acceptable starting for DL graph clustering. Even while the suggested model can reduce the graph nodes with clusters of features 66% similar on average (*See table. 5.6*), it can retain about 97% of the original graph flow (*general logic*) (*as shown in the Fig.5.6*).

Furthermore, as shown in the figure 5.6, we notice the strong relationship between the validation step's average accuracy and Global Message Flow (GMF). While the GMF shows that the created clusters may keep 97% of the original graph flow, the average accuracy of the model that aims to optimize this distance between the different clusters is almost the same. This demonstrates the effectiveness of the performance learning

results as well as the suggested squash learning function for generating KG-BP based on feature aggregation.

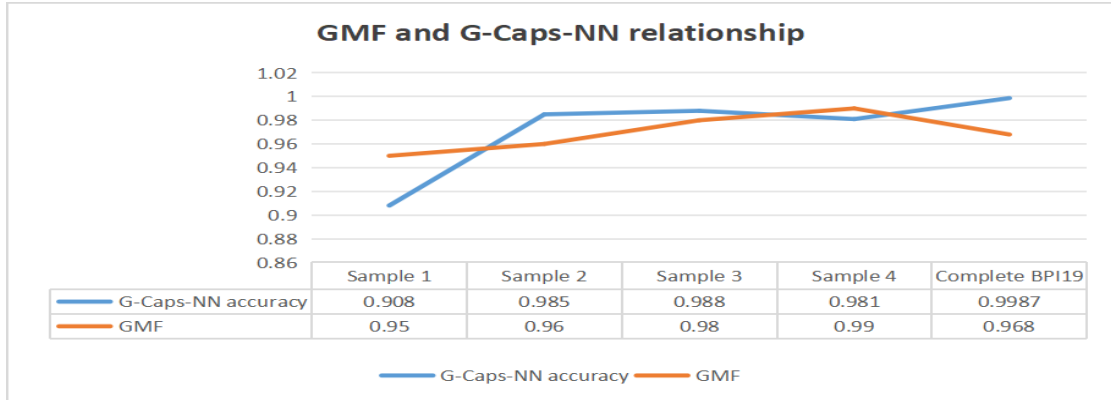


FIGURE 5.6: The global message flow and accuracy relationship

Once G-CAPS-NN's performance has been verified as being effective start, look for KG-BPs that can help the firms to reach the optimization objectives. Validating the same KG-BP using the prediction procedure is the next step detailed at the following subsection.

5.3.5.2 Event prediction through KG-BP

The event prediction evaluation is the second assessment that looks at how the final KG-BP behaves. The Decision Trees Regression (DTR) [184] and k-Nearest Neighbors method (KNN) [185] are selected to assess the prediction in order to achieve this goal. These two Machine Learning (ML) techniques were selected in order to estimate the KG-BP's effectiveness at prediction aspects using straightforward algorithms.

To this end, the learnt event features of the KG-BP are used by both algorithms to predict the next event. Table 5.7 demonstrates the remarkable efficacy of the KG-BP, as it can capture the whole conceptualization required to make a BP prediction even with a basic ML algorithm, whatever of the event-log complexity.

The next section emphasizes the outcomes of the proposed approach, the limitations, and pave the way for our future research perspectives.

5.3.6 Limitations of the G-Caps-NN

The Graph Capsule Neural Network (G-Caps-NN) is a powerful architecture for capturing hierarchical and spatial relationships within graph-structured data, such as in BP

TABLE 5.7: The prediction evaluation based KG-BP features

The $\mathcal{EL}$	Sample 1	Sample 2	Sample 3	Sample 4	The complete BPI19	Average
DTR accuracy	45.04%	48.3%	55%	70.09%	100%	63.68%
KNN accuracy	52%	55%	79.42%	80.77%	99%	73.23%

mining. However, like any deep learning model, it comes with certain limitations. Below are some key limitations of G-Caps-NN:

- The performance of G-Caps-NN is heavily dependent on hyper-parameters such as the number of capsules, the routing iterations, and the learning rate. Finding the optimal configuration is nontrivial and requires extensive experimentation.
- Although G-Caps-NN is designed to improve interpretability compared to traditional methods, understanding how individual capsules interact and contribute to specific decisions remains a challenge.
- The success of G-Caps-NN in BP mining depends on the quality and completeness of the Knowledge Graph (KG) used as input. Incomplete or noisy KGs can degrade model performance significantly.
- G-Caps-NN, with its complex architecture, has a high risk of over-fitting, especially when trained on small or imbalanced datasets.
- Implementing G-Caps-NN requires an understanding of both graph neural networks and capsule networks, which may not be straightforward for beginners.
- The quality and richness of the event logs used in BP mining significantly influence the KG-BP discovery process. Noisy or incomplete logs can hinder G-Caps-NN performance.

While G-Caps-NN is a promising approach for BP mining and decision-making, addressing these limitations is essential to maximize its potential. Future research could focus on optimizing computational efficiency, enhancing interpretability, and improving robustness to incomplete or noisy data.

5.4 Extended discussions and future perspectives

This section aims to discuss more the DSS4BP approach with further perspectives.

5.4.1 Real-world implementation of DSS4BP

DSS4BP presents a framework that integrates Knowledge Graphs (KG) and Capsule Neural Networks (CapsNet) to enhance business process decision-making. As shown in the figure 5.7, implementing this in a real-world business or company involves the following steps:

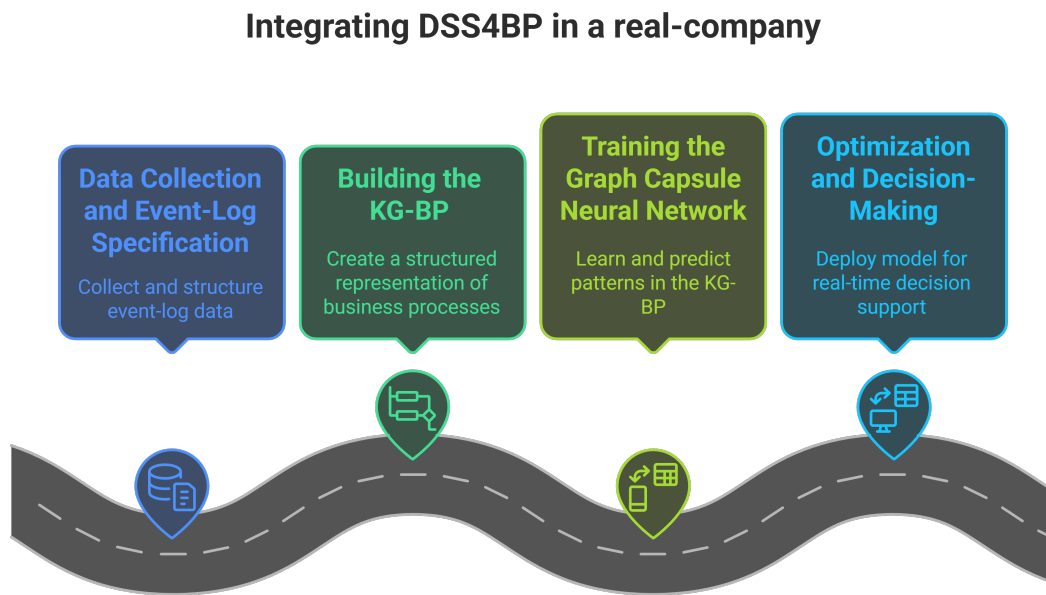


FIGURE 5.7: Integrating the DSS4BP in a real-company

1. **Data Collection and Event-Log Specification:** Before deploying an AI-driven decision-support system like KG-BP, businesses must collect and structure their event-log data. The event-log serves as the foundation for process analysis, optimization, and automation.
2. **Building the KG-BP:** A KG-BP is a structured representation of business processes where as explained in the subsection 5.2.3, the *nodes represent tasks, activities, and events, while edges represent dependencies and relationships*.
3. **Training the Graph Capsule Neural Network:** Once the KG-BP is built, the Graph Capsule Neural Network (G-Caps-NN) is used to learn and predict *the patterns (semantic nodes and transitions)* of the KG-BP substantial for further decision-making. The power of the G-Caps-NN is that the traditional neural networks often fail to capture hierarchical relationships in complex business processes. G-Caps-NN overcome this by:

- Recognizing dependencies between activities rather than treating them as independent events.
 - Handling process variations (*e.g., different ways of approving a loan*).
4. **Optimization and Decision-Making:** As shown in the table 5.8, after training the model, it can be deployed to support real-time decision-making.

TABLE 5.8: Business Functions Enabled by DSS4BP

Function	Description	Example
Process Optimization	Detect inefficiencies and automate workflow adjustments.	A retailer optimizes inventory management.
Event Prediction	Anticipate failures, risks, and bottlenecks.	A factory predicts machine failure before it happens.
Decision Support	AI-driven recommendations for human decision-makers.	A hospital predicts patient admission trends.

5.4.2 Advantages of DSS4BP compared to existing DSS systems

After experimenting and ensuring the efficiency of the DSS4BP system, this section aims to explain the cues of the superiority of the suggested approach comparing to the existing DSS systems. Where it introduces advanced AI-driven techniques, particularly leveraging KG and CapsNet, to enhance decision-making in BPs. When compared to traditional Decision Support Systems (DSS), BP4DSS offers several advantages:

5.4.2.1 Improved Process Awareness and Automation

As shown in the table 5.9, DSS4BP represents a significant advancement over traditional DSS by :

- Integrating event log analysis, AI-driven automation, and knowledge representation to enhance decision-making. Unlike traditional DSS, which relies on predefined rules and static data, DSS4BP continuously tracks real-time business activities using event logs and KGs, providing a more dynamic and adaptive decision-making framework.
- The AI-powered workflow automation in DSS4BP enables anomaly detection and process optimization, eliminating the need for constant manual intervention, which is a major limitation of traditional DSS.

TABLE 5.9: Comparison Between DSS4BP and Traditional DSS

Feature	DSS4BP	Traditional DSS
Process Awareness	Uses event logs knowledge graphs to track real-time business activities.	Relies mostly on predefined rules static data.
Automation	AI-driven workflow automation anomaly detection.	Requires manual intervention rule updates.
Contextual Insights	Provides deep context awareness using PM.	Limited to historical reporting dashboard.

- Moreover, DSS4BP enhances contextual understanding by leveraging process mining techniques, allowing businesses to uncover hidden patterns and dependencies in their workflows, whereas traditional DSS is typically restricted to historical reporting and static dashboards. This transformation makes DSS4BP a powerful tool for businesses looking to optimize their processes through intelligent decision support.

5.4.2.2 Enhanced Predictive Decision-Making

As shown in the table 5.10, DSS4BP significantly outperforms traditional DSS systems by:

- Integrating CapsNet and KG, which enhance predictive decision-making.
- Unlike traditional DSS, which relies on basic statistical models and predefined rule-based decision making, DSS4BP uses DL algorithms to capture complex dependencies between BPs. This enables higher accuracy in prediction, allowing organizations to anticipate failures, risks, and inefficiencies before they occur, rather than reacting after issues arise.
- Furthermore, DSS4BP offers advanced scenario planning, where AI-driven models simulate multiple future outcomes, helping businesses select the optimal course of action. Traditional DSS, in contrast, is highly based on static “if-then” simulations, which lack the ability to adapt dynamically to new data and evolving business conditions.
- The incorporation of graph-based AI models allows DSS4BP to continuously learn and refine its decision-making framework, making it a powerful tool for industries such as finance, healthcare, supply chain, and manufacturing.

TABLE 5.10: Comparison of DSS4BP and Traditional Predictive DSS

Feature	DSS4BP	Traditional DSS
Prediction Accuracy	Uses Caps-Net AI-driven predictions.	Uses basic statistical models for forecasting.
Event Prediction	Anticipates failures, risks, and process inefficiencies.	Mostly reactive, not proactive.
Scenario Planning	Simulates multiple future outcomes to suggest optimal decisions.	Limited by rule-based simulations.

- By bridging the gap between process automation and intelligent decision support, BP4DSS provides businesses with real-time, data-driven, and context-aware insights, ultimately improving operational efficiency, reducing costs, and enhancing strategic agility.

5.4.2.3 Integration with AI and Knowledge Representation

As shown in the table 5.11, DSS4BP represents a paradigm shift in decision making, utilizing GNN and capsule networks to provide deep learning-based insights into complex business processes. This DSS4BP significantly outperforms traditional DSS systems where:

1. Unlike traditional DSS, which relies mainly on predefined rule-based systems with limited adaptability, DSS4BP dynamically learns from real-time event logs, historical data, and structured business workflows.
2. The integration of KGs allows DSS4BP to model relationships and dependencies within business processes, ensuring context-aware decision-making that evolves with operational changes. This contrasts with traditional DSS, which typically uses structured relational databases that lack the ability to represent inter-dependencies effectively.
3. Furthermore, DSS4BP detects hidden patterns and anomalies in business operations through advanced data-driven insights, enabling predictive analytics, proactive risk mitigation, and continuous optimization. Traditional DSS, on the other hand, rely on static business rules and pre-configured reports, which often fail to capture the complexity and variability of modern business environments.
4. The ability of DSS4BP to self-learn, adapt, and provide real-time recommendations makes it a superior alternative to traditional DSS, improving efficiency, scalability, and precision of decisions in industries.

TABLE 5.11: Comparison between DSS4BP and Traditional DSS

Feature	DSS4BP	Traditional DSS
AI Integration	Uses GNN + Capsule Networks for DL-based decision-making.	Limited AI capabilities, mostly rule-based systems.
Knowledge Representation	Uses KG to link BPs and dependencies.	Uses structured relational databases with limited contextual understanding.
Data-Drive Insights	Analyzes large event logs, detects hidden patterns and relationships.	Mostly relies on pre-configured business rules and reports.

5.4.3 Future perspectives of DSS4BP

The KG-BP and G-Caps-NN framework presents an innovative approach to DSS4BP by combining KGs and capsnets. The future development of this model focuses on the three following areas:

5.4.3.1 Expanding data validation

The DSS4BP model has been tested on the BPI19 event-log dataset, which provides real-world event sequences from business processes. However, business processes vary significantly across industries such as finance, healthcare, supply chain, and manufacturing. The following future directions of the present contribution are:

1. **Industry-Specific Testing:** The model needs to be tested on datasets from multiple industries to ensure its robustness and generalization.
2. **Cross-Dataset Benchmarking:** Future research should compare DSS4BP against other existing business process mining models across datasets like BPI Challenge 2012, BPI Challenge 2015, and real-world event logs.
3. **Noise Handling and Data Cleaning:** Real-world event logs contain incomplete, inconsistent, or redundant information. Techniques such as data augmentation, imputation, and anomaly detection can be integrated to enhance preprocessing.

5.4.3.2 Extending DSS4BP applications

The DSS4BP model is designed for a decision-making in a BP based on historical event logs. The future directions of this context are the following:

1. **Anomaly Detection:** Businesses need to detect process deviations, fraudulent transactions, or compliance violations. By integrating anomaly detection techniques (*e.g.*, *Isolation Forest*, *Auto-encoders*), DSS4BP can help organizations prevent financial fraud, security breaches, or process failures.
2. **Fraud Identification:** In financial transactions and auditing, fraud detection requires advanced pattern recognition. DSS4BP can be extended to learn hidden relationships between financial activities and detect suspicious behavior in business process workflows.
3. **Recommendation Systems:** By analyzing event dependencies, DSS4BP can suggest optimized business workflows, making data-driven recommendations for process improvement.

5.4.3.3 Integration with Reinforcement Learning

The existing DSS4BP framework relies on historical data for predictions, meaning it cannot dynamically adapt or optimize based on real-time BP changes. RL will be integrated to allow adaptive decision-making, where the model:

1. learns optimal actions based on rewards.
2. learns from real-time business events and suggest optimal task scheduling or resource allocation.
3. can help businesses predict and mitigate risks dynamically, such as supply chain disruptions or failures.

5.5 Conclusion

This chapter presents the first contribution of this thesis named DSS4BP mining system to discover a KG-BP from event log data. The proposed system is based on the G-Caps-NN architecture, which is able to learn the unified event features of the BP and the relationship between them. Beyond the simple visualization of the BP model extracted from execution traces, the KG-BP proposed in this work constitutes an effective tool for the optimization and prediction decision-making support. The results found in the training stage prove the efficiency of our G-Caps-NN and the KG-BP data structure.

Chapter 6

BP-DSS3: A Business Processes Decision-Support Chatbot through the Language Model GPT-3.5-turbo

6.1 Introduction

The objective of this chapter consists to investigate the adoption of innovative language models to improve decision-making in the context of business processes. We exploit the NLP paradigm in the context of DSS related to BPs, and we develop a conversational chat-bot whose capabilities are fine-tuned with domains specific data. Thus, instead of dealing with complex graphical representations and formal models when analyzing BPs, resorting to the use of natural languages can significantly contribute to the understanding of how the BP works. Hence, relying on NLP allows facilitating the selection of the appropriate decision to be adopted at each progression stage of the BP execution. In such scenarios, BP managers can assess the extent to which understanding and mastering BP structures and functionalities can impact strategic decision making. In addition, by consolidating the BP ecosystem with effective decision support tools, it is possible to strengthen communication and collaboration activities between the different involved actors.

6.2 Motivations

This section exposes the motivations having lead to this work.

The contemporary business landscape is marked by unprecedented complexities and dynamic challenges, requiring organizations to permanently refine their decision-making processes [186]. In this context, exploiting AI techniques contributes significantly facing to the arising challenges. In fact, the fusion of AI and NLP holds immense promise for revolutionizing how decisions are made in diverse business domains [187]. More precisely, in this work we focus on exploring and leveraging the capabilities of chatbots (that boosts the business productivity from 2022, as shown in the figure 6.1 [23], where, it is estimated that by 2025, generative AI would account for over 30% [23] of the whole AI market and be worth around 60 dollars billion.)and we customize the GPT-3.5-turbo model [188] in order to build a new chatbot named "BP-DSS3" that empowers and supports decision-making within intricate BPs.

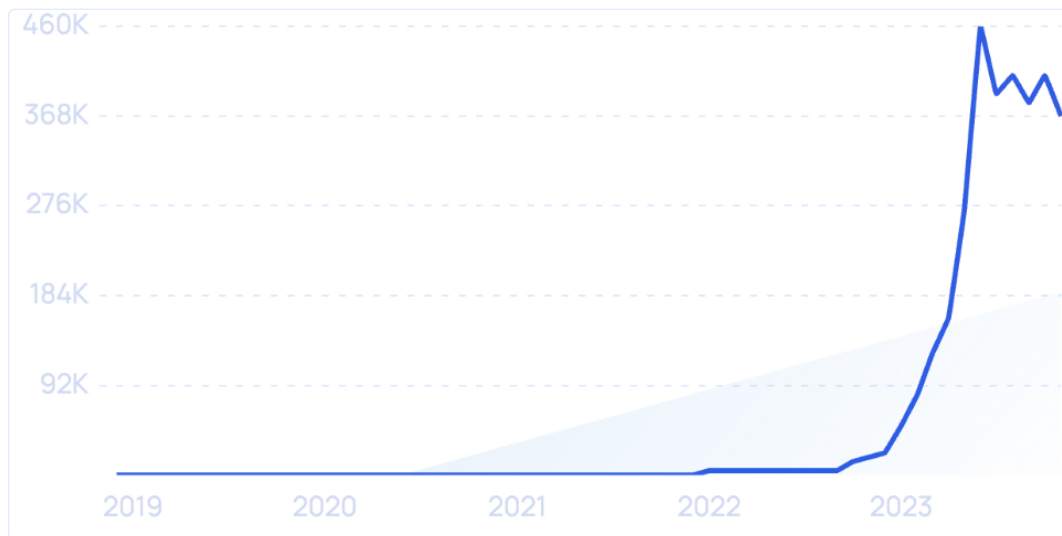


FIGURE 6.1: Search volume for “generative AI” has exploded in late 2022 [23]

The current work is motivated by the following reasons.

- **Handling dynamic evolution of BPs:** Modern companies evolve in open environments where the conducted BPs span across geographical boundaries [189]. Hence, markets are becoming global and more competitive [190]. To cope with the induced effects, firms have to be flexible and able to adapt to the ever changing conditions that occur in their environments at an increasing rate. As an immediate consequence, the intricacies of decision-making have grown exponentially [191], involving multifaceted considerations spanning strategic planning, resource

allocation, risk management, and operational efficiency. To deal with this complex landscape and to manage the associated constraints, there is a compelling need for intelligent systems that can comprehend, contextualize, and augment the decision-making prowess of human stakeholders. In this perspective, this current work attempts to consolidate DSS with AI techniques.

- **Deploying NLP in the DSS context:** As a sub-field of AI, NLP plays a vital role in DSS by enabling extraction, analysis, and interpretation of information from unstructured textual data. In the DSS context, NLP techniques are employed to process various types of text data, including documents, emails, customer's feedback, and social media posts to provide valuable insights and support decision-making processes.

As an illustration, one key application of NLP in DSS is sentiment analysis, where text data is analyzed to determine the sentiment or opinion expressed within it. This analysis can help businesses appreciate customer satisfaction, identify emerging trends, and assess public opinion about products, services, or brands. Hence, sentiment analysis results can inform strategic decisions related to marketing campaigns, product development, and customer service initiatives.

Another important use case of NLP in DSS is information extraction, where relevant information is extracted from text documents or sources to support decision-making processes. This may involve identifying key entities (*such as people, organizations, or locations*), extracting important events or facts, or summarizing vast amount of textual data into concise and actionable insights. Thus, information extraction techniques enable DSS to automatically process and analyze vast amounts of textual information, facilitating faster and more informed decision-making.

Furthermore, NLP techniques are utilized in DSS [192] for text classification and categorization tasks related to BPs, where text data is classified into predefined categories or topics. This can help organize and structure textual information, making it easier to navigate and analyze. Text classification algorithms can be applied to numerous domains, such as document management, content filtering, and risk assessment, to streamline decision-making processes and improve efficiency.

Overall, NLP enhances DSS by unlocking valuable insights from unstructured textual data, enabling organizations to make more informed decisions, anticipate trends, and respond effectively to changing business environments. By leveraging NLP techniques, DSS can harness the power of text data to drive strategic initiatives, optimize operations, and gain a competitive edge in today's data-driven

landscape.

- **Exploiting chatbots advantages for DSS:** Chatbots with their conversational AI capabilities, offer a valuable opportunity to enhance decision support systems[192]. By engaging users in natural language conversations, the model can comprehend the subtleties of decision points, consider diverse factors, and provide context-aware recommendations. The potential benefits range from improved decision accuracy to enhanced collaboration among decision-makers, marking a significant advancement of the landscape of intelligent decision support. The major benefits of using chatbots are the following:
 - **24/7 availability:** As shown in the figure 6.2 [23], a 2022 Gallup survey showed 56% of full-time US employees have jobs that can be done from home. That amounts to 70 million workers. Of those workers, 50% have a hybrid work schedule, 30% work fully remotely, and 20% work fully in-office. In this context, the chatbots enable businesses to provide round-the-clock support to users and customers. Further, it enhances accessibility and responsiveness, especially in global operations.
 - **Cost saving:** Automation of tasks leads to cost savings by reducing the need for human intervention. Thus, chatbots can handle a large volume of inquiries simultaneously.
 - **Scalability:** Chatbots are easily scalable to handle an increasing number of interactions with numerous users. Furthermore, they are adaptable to the growing needs of the business.
 - **Data analysis and insights:** Chatbots can collect and analyze data from interactions, providing valuable insights. They enable businesses to understand user behavior and preferences.
 - **Consistency:** Chatbots can ensure consistent responses and adherence to business rules and policies. Consequently, they reduce the likelihood of human errors in communication.

Based by the aforementioned considerations, we aim to overcome the observed limitations and constraints of traditional DSS and offer decision-makers a sophisticated and perceptive ally by building an enhanced chatbot refined with GPT-3.5-turbo, and which serves as a powerful tool during BPs management.

Before exposing the suggested system, the next subsection discusses related works having addressed the issue of using chatbots in DSS environments.

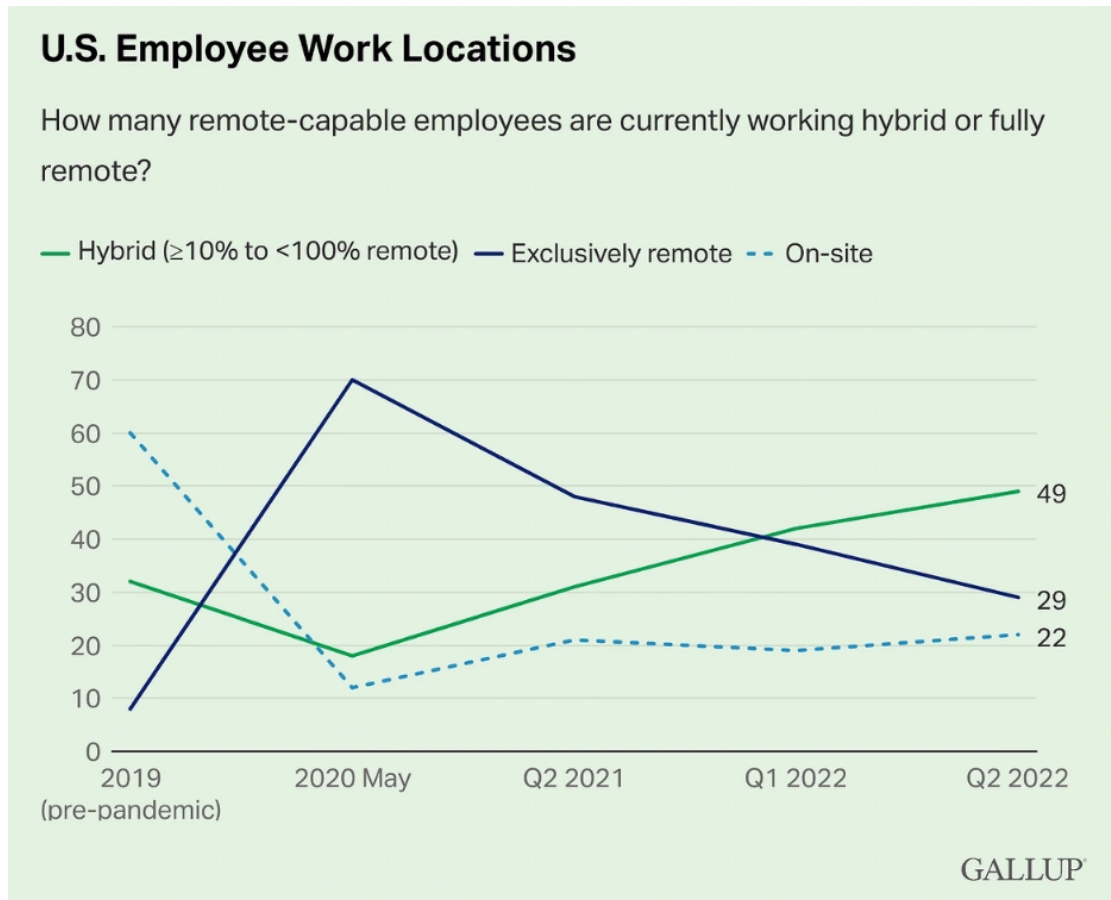


FIGURE 6.2: As the pandemic wanes, many workers are adopting a hybrid work schedule [23]

6.3 Development of the chatbot BP-DSS3

Generative language models, such as employed in Chat-GPT are designated to generate human-like text that express responses or outputs to users queries, basing on the data introduced in input. In the context of BPs, elaborating a decision-making framework consists of customizing the language model for particular use cases, by fine-tuning GPT-3.5-turbo [193].

To this end, this section defines the problem being addressed and presents its formal specification, then the different components and stages reflecting the architecture and functionalities of the conceived framework are deeply explained.

6.3.1 Problem statement and specification

The goal of this work consists to adapt the existing GPT-3.5-Turbo model to perform decision-making tasks, such as BP documentation, anomalies detection, or recommendation systems. This goal can be achieved by exploiting BP-specific datasets, through a pragmatic analysis of their content and links. The target model starts learning the domain's specific concepts, patterns, and decision-making logic, as well as the applied business rules. After that, the fine-tuning of GPT-3.5-Turbo on BP data involves adjusting the model's pre-trained parameters using a domain-specific dataset. The previous steps allow improving the model's ability to understand and generate text related to BPs, while minimizing the loss on data related to BP-specific tasks.

The Fig.6.3 bellow depicts the problem and its parameters.

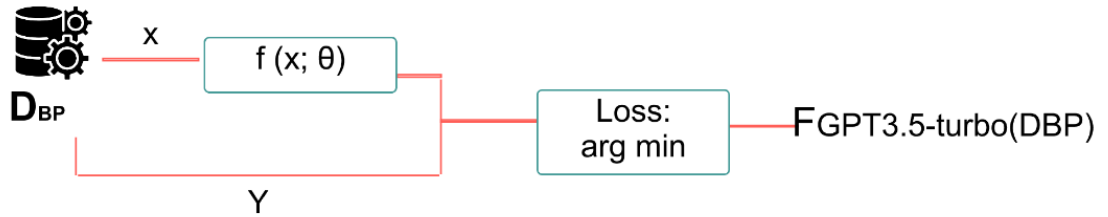


FIGURE 6.3: Illustration of the BP decision-making problem using chatbots technology

Where the used notations are explained below.

- Let $\mathcal{F}_{\text{GPT3.5-turbo}}(\mathcal{D}_{\text{BP}})$ be the fine-tuned GPT-3.5-Turbo model on BP tasks, where:
 $\mathcal{D}_{\text{BP}}$ designates the domain-specific BP dataset used for the fine-tuning stage,
- θ corresponds to the model's parameters used in GPT-3.5-Turbo,
- Consider that $f(x; \theta)$ expresses the function producing the output of GPT-3.5-Turbo, given the input x (*contextual BP data*) and the parameters θ ,
- Now assume that $\mathcal{L}$ is the loss function capturing the error between the model's output and the true label y (*ground truth for BP tasks*),
- This previous loss is captured using $\mathbb{E}_{(x,y) \in \mathcal{D}_{\text{BP}}}$, which denotes the expectation over the dataset $\mathcal{D}_{\text{BP}}$ containing the input-output pairs (x, y) specific to the BP domain.

Basing on the previous specifications, the formal definition of the decision-making problem using chatbots is formalized with the equation (6.1) bellow.

$$\mathcal{F}_{\text{GPT3.5-turbo}}(\mathcal{D}_{\text{BP}}) = \arg \min_{\theta} \mathbb{E}_{(x,y) \in \mathcal{D}_{\text{BP}}} \mathcal{L}(f(x; \theta), y) \quad (6.1)$$

Where $\arg \min$ denotes the habitual mathematical function for calculating the loss between the input and the output values.

This iterative fine-tuning process minimizes the loss function $\mathcal{L}$ and ensures that the target model evolves to a level that it becomes highly specialized for tasks related to the BPs domain.

6.3.2 Development of the Chatbot BP-DSS3

This sub-section describes the different modules of the developed Chatbot BP-DSS3 and the progress of the processing flow when handling user requests. As shown in Fig. 6.4, BP-DSS3 operates in three main progressive stages leading to decision-making related to BPs. In what follows, these stages are deeply explained and illustrated.

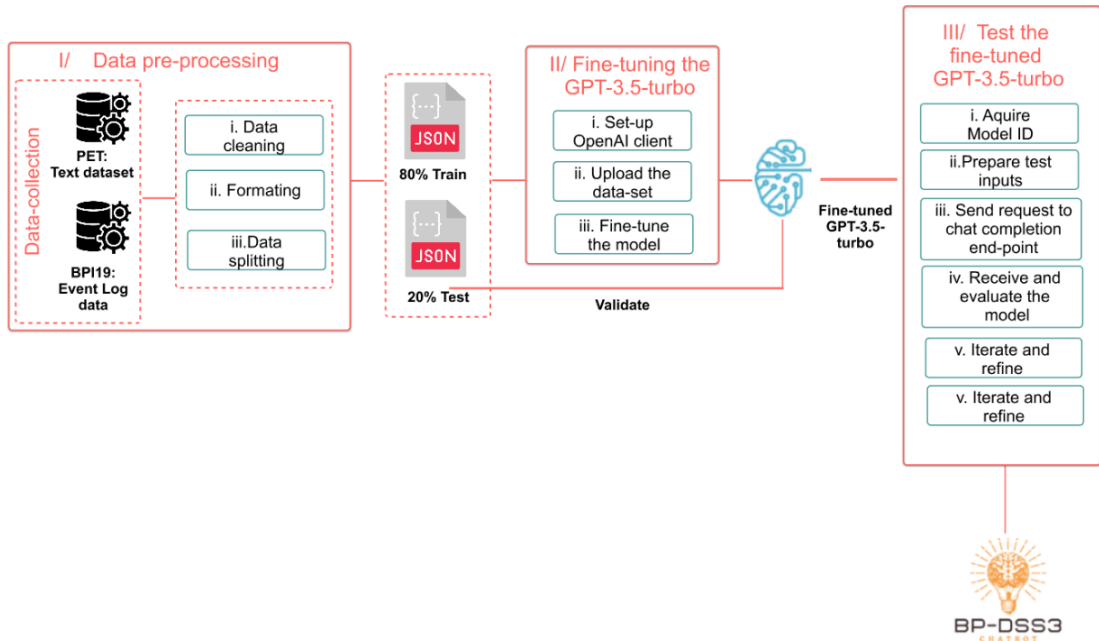


FIGURE 6.4: Architecture and functionalities of the Developed chatbot BP-DSS3

6.3.2.1 Pre-processing stage

Data pre-processing is a critical step to ensure that the employed data-set is consistent, appropriately formatted and ready for the fine-tuning process. The pre-processing stage is articulated around the four following steps detailed bellow.

1. **Data collection:** The first step toward an accurate fine-tuning process involves acquiring qualitative and quantitative input datasets. To this end, the annotated dataset, named PET [194] is employed. This textual corpora targets the extraction of BPs specifications from natural languages' texts and unstructured data, expressing processes' descriptions. Further, the widespread event-log data BPI19, capturing various activities performed by a real-life loan application process was intensively exploited. This second dataset is provided by the 9th International Business Process Intelligence Challenge (*ICPM on 2019*)[178]. Its content consist of a huge event log data reflecting a real-life data over 1,5 million events originating from a Netherlands large multinational company operating in the area of coatings and paints.

The Fig. 6.5 bellow depicts an excerpt of BPI19 dataset employed to fine-tune the GPT3 model.

```
{
  "messages": [
    {
      "role": "system",
      "content": "A bot aim to answer the questions about BPI Challenge 2019."
    },
    {
      "role": "user",
      "content": "What BPI challenge 2019?"
    },
    {
      "role": "assistant",
      "content": "This data originated from a large multinational company operating from The Netherlands in the area of coa"
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "A bot aim to answer the questions about BPI Challenge 2019."
    },
    {
      "role": "user",
      "content": "What is the reference of BPI challenge 2019?"
    },
    {
      "role": "assistant",
      "content": "https://icpmconference.org/2019/icpm-2019/contests-challenges/bpi-challenge-2019/"
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "A bot aim to answer the questions about BPI Challenge 2019."
    },
    {
      "role": "user",
      "content": "3-way matching, invoice after goods receipt."
    },
    {
      "role": "assistant",
      "content": "For these items, the value of the goods receipt message should be matched against the value of an invo"
    }
  ]
},
{
  "messages": [
    {
      "role": "system",
      "content": "A bot aim to answer the questions about BPI Challenge 2019."
    },
    {
      "role": "user",
      "content": "3-way matching, invoice before goods receipt."
    },
    {
      "role": "assistant",
      "content": "Purchase Items that do require a goods receipt message, while they do not require GR-based invoicing (i"
    }
  ]
}
```

FIGURE 6.5: A formatted excerpt of the BPI19 dataset

2. **Data cleaning:** Data cleaning is a fundamental operation for preparing the dataset to be exploited during the fine-tuning process of GPT-3.5-turbo model. Indeed, cleaning the data helps ensuring that the model receives and manages high-quality input and, consequently, it can generate meaningful and accurate outputs. The two activities involved during data cleaning are the following.
 - For the PET text data, data cleaning consists in removing special characters, punctuation, and unnecessary white-spaces of the tokens.
 - Identifying and handling missing values for the BPI19 event log dataset. This might involve imputing missing values, removing the columns or records with incomplete data.
3. **Formatting:** To start the fine-tuning process, the cleaned dataset was prepared for training the developed model. In this perspective, a set of conversations scenarios similar to those that could be asked during real-time conversation sessions was elaborated. Each example in the formatted dataset should be a conversation in the same format as OpenAI's chat completions API. Specifically, conversations' messages are structured as objects with the following attributes.

- **Role:** Can be either "system", "user", or "assistant". First, a system message is included at the beginning to set the behavior and context for the assistant. This message helps guide the model's responses. Second, user messages are structured in a conversation format, including both user and assistant turns. The use of the role labels aims to differentiate between user and assistant messages.

In this context, the role can take three types of messages in a conversation: system, user, and assistant.

- The system message comes first and sets the tone for how the assistant should behave.
- The user messages correspond to the user questions and queries.
- The assistant messages express the responses provided by the assistant agent.

These labels help keep track of who's saying what in the conversation.

- **Content:** A string containing the current message originating either from the user or the assistant.

4. **Data splitting:** The resulting output of the formatting step are stored in a JSON format [195], due to its simplicity and portability. The produced files are splitted into two distinguished parts. The first one contains 80% of data and will be used for the training process, while the remaining 20% of data will serve during tests of the system.

Once the pre-processing step is achieved, data becomes ready to be used for the fine-tuning stage. In what follows, the fine-tuning process is deeply tackled.

6.3.2.2 Fine-tuning stage

To develop the chatbot BP-DSS3, the GPT-3.5-turbo model was used and we were inspired from the work [196]. Thus, by using the OpenAI API with the TypeScript (TS) to fine-tune the developed chatbot, the focus was made on the iterative analysis of the system's responses corresponding the users' queries. Hence, the model was trained to understand nuanced decision contexts across different business domains.

Below, the pivotal steps of the fine-tuning stage using TS with the OpenAI GPT-3.5-turbo API are depicted.

- **i) Setup OpenAI Client:** To setup the OpenAI client for GPT-3.5-turbo, an OpenAI API key is used to authenticate the user requests to the OpenAI API. As

illustrated in Fig. 6.6, this action provides specific endpoints for the fine-tuning process.

```
import fs from 'fs'
import OpenAI from 'openai'

import { FineTuningJobEvent } from 'openai/resources/fine-tuning'
import 'dotenv/config'

// Gets the API Key from the environment variable `OPENAI_API_KEY`
const client = new OpenAI({
  apiKey: process.env.OPENAI_API_KEY
})
```

FIGURE 6.6: The setup of OpenAI API client code of the GPT3.5-turbo

- **ii) Uploading the dataset:** As already stated in § 6.3.2.1, fine-tuning a model typically involves handling a dataset that is in the format ("role", "content"). The code shown in Fig. 6.7 below allows uploading the dataset file.
- **iii) Fine-tuning process:** The process is triggered by submitting requests to the OpenAI fine-tuning API. It provides the user the dataset and specify the task or role for which he wants to fine-tune the model. Further, the OpenAI API will guide him through the steps needed for fine-tuning. The code in the Fig. 6.8 illustrates this fine-tuning process.

Once the model has been adequately fine-tuned, it becomes exploitable and its capabilities to address decision-making concerns are examined in the following.

6.3.2.3 Testing the fine-tuned model

The testing procedure, typically, involves formulating and submitting the users' requests to the chat completion endpoint with the specific ID of the fine-tuned model. The rightmost part of the Fig. 6.4 enumerate the different activities to be conducted during the test procedure. Hereafter the considered activities are deeply explained.

1. **Acquire model ID:** Consists in obtaining the unique identifier (ID) assigned to the fine-tuned GPT-3.5-turbo model (*e.g. in this case the ID is ft:gpt-3.5-turbo-0613:university-of-8may::8fChoSKr*). This ID is necessary for accessing and utilizing the model for inference tasks related to decision-making in the BP context.

```
async function main() {
  console.log(`Uploading file`)

  let file = await client.files.create({
    file: fs.createReadStream('./scripts/data.jsonl'),
    purpose: 'fine-tune'
  })
  console.log(`Uploaded file with ID: `, file)

  console.log('-----')

  console.log(`Waiting for file to be processed`)
  while (true) {
    file = await client.files.retrieve(file.id)
    console.log(`File status: ${file.status}`)

    if (file.status === 'processed') {
      break
    } else {
      await new Promise(resolve => setTimeout(resolve, 1000))
    }
  }
}
```

FIGURE 6.7: The code for uploading the dataset

2. **Prepare test inputs:** An adequate set of test inputs representing different conversational scenarios is prepared and ready to be introduced as input of the developed chatbot BP-DSS3. These inputs concern decision-making on BPs and should cover a diverse range of contexts, use-case scenarios and potential user queries or prompts.
3. **Send requests to chat completion endpoint:** This task utilizes the OpenAI API (*or any other appropriate interface*) to send requests to the chat completion endpoint of BP-DSS3. Each request must include the ID of the concerned fine-tuned model along with the related test inputs.

The three previous steps (*i*, *ii* and *iii*) are shown in the third box (*numbered III*) of the Fig. 6.4 and their implementation is illustrated in Fig. 6.9. At this development stage, the evaluation and refinement of the resulting responses of the fine-tuned GPT-3.5-turbo model constitute the goal of the three following steps.

```

console.log(`Starting fine-tuning`)
let fineTune = await client.fineTuning.jobs.create({
  model: 'gpt-3.5-turbo',
  training_file: file.id
})
console.log(`Fine-tuning ID: ${fineTune.id}`)

console.log('-----')

console.log(`Track fine-tuning progress:`)
console.log("status is ", fineTune.status)
const events: Record<string, FineTuningJobEvent> = {}

while (fineTune.status == 'validating_files' || fineTune.status == 'running' ) {
  fineTune = await client.fineTuning.jobs.retrieve(fineTune.id)
  console.log(`${fineTune.status}`)

  const { data } = await client.fineTuning.jobs.listEvents(fineTune.id, {
    limit: 100
  })
  for (const event of data.reverse()) {
    if (event.id in events) continue
    events[event.id] = event
    const timestamp = new Date(event.created_at * 1000)
    console.log(`- ${timestamp.toLocaleTimeString()}: ${event.message}`)
  }

  await new Promise(resolve => setTimeout(resolve, 5000))
}

```

FIGURE 6.8: The code for fine-tuning process of the GPT3.5-turbo model

4. **Receive and evaluate the model responses:** Upon receiving the requests, the chat completion endpoint processes the introduced inputs. As a response to the submitted queries, the model generates the corresponding answers based on the provided context and its decision-making capabilities. After that, the generated responses are evaluated by the fine-tuned model accordingly to a set of predefined criteria and metrics. This evaluation consists in assessing the accuracy of the made decisions, the relevance of the responses to the considered context, and the overall coherence and quality of the dialogue between the user and the chatbot BP-DSS3.
5. **Iterate and refine:** Based on the results obtained during the evaluation phase, potential situations needing more improvements can be identified and analyzed. This may include refining the fine-tuning process, incorporating feedback, adjusting model parameters, or incorporating additional training data to address any shortcomings observed during the previous evaluation phase.
6. **Repeat testing as needed:** Here, the testing process is iterated and the necessary adjustments are made in order to enhance the model abilities, while retesting it to validate the operated improvements. This iterative cycle is repeated until the model achieves satisfactory performance according to the desired criteria.

Although the previous steps can effectively test the output sequences provided by the fine-tuned GPT-3.5-turbo model for decision-making, it's important to assess that the

```
const config = new Configuration({
  apiKey: process.env.OPENAI_API_KEYYY,
});

const openai = new OpenAIApi(config);

export const runtime = "edge";

export async function POST(req: Request) {
  const { messages } = await req.json();

  const response = await openai.createChatCompletion({
    model: 'ft:gpt-3.5-turbo-0613:university-of-8may::8fChoSKr',
    max_tokens: 20,
    stream: true,
    messages: messages,
  });

  const stream = OpenAIStream(response, {});
  openai.createFineTune

  return new StreamingTextResponse(stream);
}
```

FIGURE 6.9: The code for utilizing the fine-tuned GPT-3.5-turbo model

developed chatbot BP-DSS3 also meets the standards of performances. In this perspective, the experiments of the fine-tuned model will allow assessing this important aspect. The next section is dedicated to the discussion of the experimental results.

6.4 Experimental Results

This section aims to experiment the applicability of the proposed chatbot and to assess its performance for supporting decision-making in the context of BPs. The first goal consists to measure and validate the quality of the training stage of the proposed BP-DSS3 model. To this end, the efficiency of the trained model when handling the datasets used during the fine-tuning process is examined. After that, the abilities of the developed chatbot is experimented with two real-world scenarios *i.e.*; *alignment with organizational objectives and risk management and contingency planning*) and the obtained results are illustrated and discussed. Finally, the potential challenges and issues related to the

large adoption of chatbots in DSS environments are discussed in order to mark out new research perspectives.

Before going into further experimental details, we recall that the employed datasets during the training stage are: (i) the PET dataset which is used to extract BP models from natural languages, and (ii) the BPI19 event-log data containing real-word BPs execution. For more details about the used fine-tuning datasets, see the previous first item (*i*) data collection) in § 6.3.2.1).

The following subsection aims to validate the decision-making capabilities of the proposed system through the training stage.

6.4.1 Capabilities of BP-DSS3 for decision-making

These experiments aim to showcase the abilities of the developed chatbot BP-DSS3 for supporting decision-making tasks during its training on the pre-processed two datasets PET and BPI19. To this end, the chatbot is trained with the two previous datasets (*i.e.*; *PET* and *BPI19*) and for each one the training loss is estimated.

The training results of the model for more than 100 epochs with the two datasets are shown in Fig. 6.10 and Fig. 6.11, respectively.

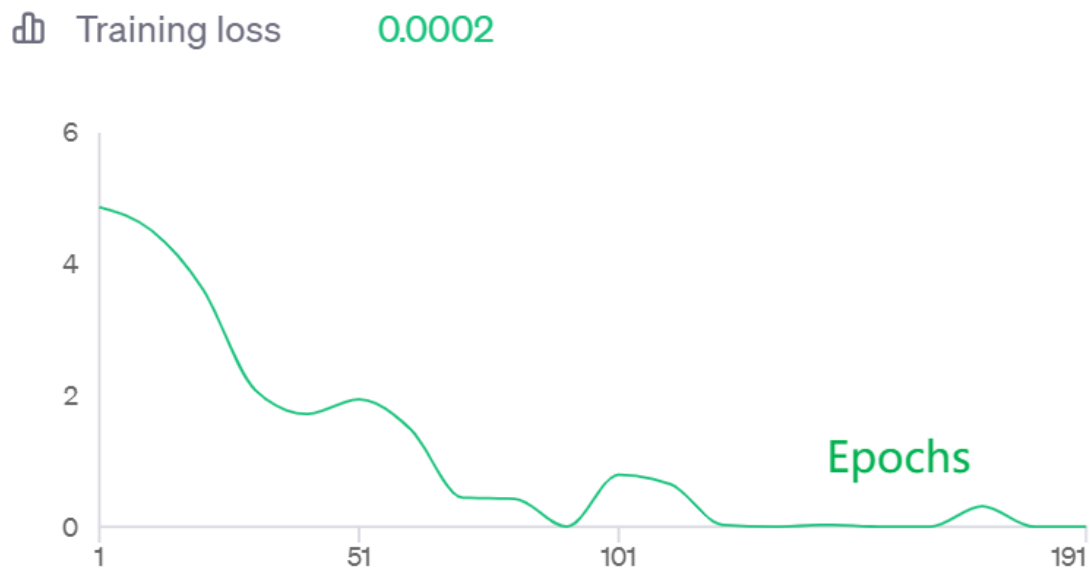


FIGURE 6.10: The GPT-3.5-turbo trained with PET dataset

The interpretation of the two figures brings out the two following conclusions.

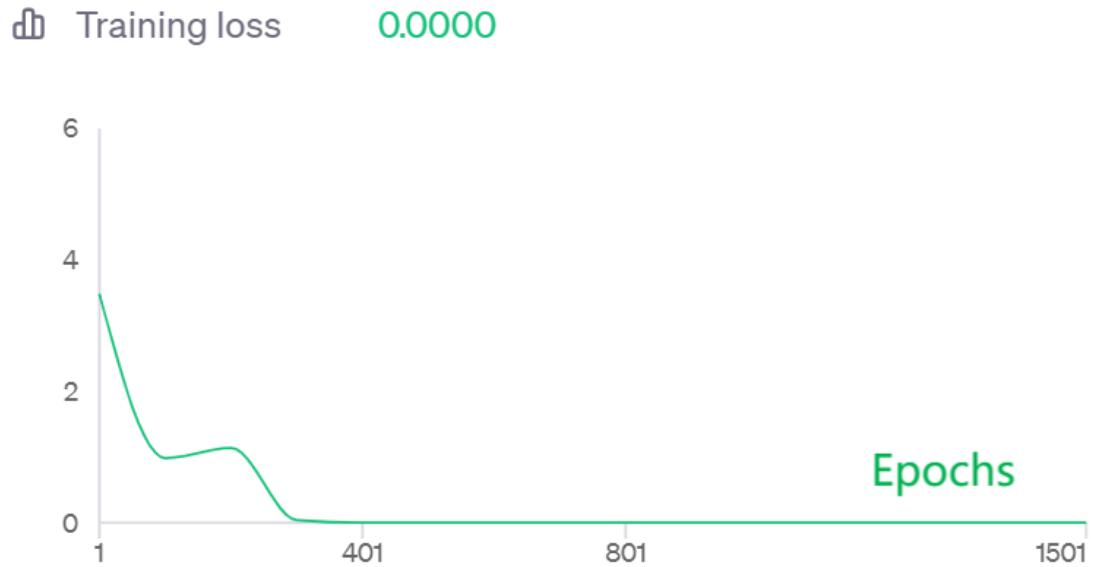


FIGURE 6.11: The GPT-3.5-turbo trained with BPT19 dataset

1. The accuracy of the training phase of both datasets is higher than 99% (*0.9998 for the PET dataset and 100% for BPI19 dataset*). During the fine-tuning of GPT-3.5-turbo on a specific BP dataset, the model undergoes a rapid training process due to the sophisticated pre-trained architecture that already encompasses a vast amount of generalized knowledge. The model does not require the same degree of training complexity as traditional neural networks which act from scratch. As a result, the fine-tuning stage is relatively fast, depending on the dataset size and the specific domain-related adjustments. This efficiency allows the model to adapt quickly to new contexts, while preserving the high accuracy of its predictions. Even during rapid training phases, GPT-3.5-turbo can achieve near-perfect adaptation to the specific domain without prolonged training epochs.
2. Further, such training results prove the efficiency of the proposed framework and its capabilities to learn patterns and associations between input messages and appropriate responses. This fine-tuning of GPT-3.5-Turbo on BP datasets like BPI19 and PET uncovers a rich array of patterns, from sequential workflows to conditional decision-making and exception handling. These patterns help the model to become highly contextualized and customized for BP-related tasks, leading to more accurate predictions, better anomaly detection, and a deeper understanding of process variations. By understanding the previous patterns, organizations can use the model to automate decision-making (*e.g. identifying inefficiencies or areas*) for improvement in their BPs.

The following subsection describes a more practical use of the developed chatbot BP-DSS3 and tests its effectiveness for managing handling two real-world scenarios.

6.4.2 Case studies

The goal of deploying the developed chatbot for further case studies aims to experiment its applicability and its efficiency for handling real-world BPs. Afterwards, the obtained results are compared with those provided by chat-GPT before the fine-tuning stage. More precisely, the experimental objectives of this stage are the following.

1. Whatever the accuracy [197] of BP-DSS3 is very satisfying, this parameter alone may not fully capture the chatbot's decision-making capabilities. Thus, resorting to the usage of real-world scenarios can consolidate its performances' features during its effective deployment.
2. Generally, over-fitting is a common concern during the training phase of ML models. In this context, the over-fitting issue does not pose significant difficulties, even with high training accuracies (*99% or more*). This observation is largely justified by the GPT-3.5's ability to generalize knowledge effectively, due to its massive pre-training on diverse datasets. As a consequence, the experimental results that are based on testing real cases will examine the absence of the over-fitting phenomena.

The developed chatbot BP-DSS3 was experimented in the field of strategic planning. Strategic planning ensures that BP decision-making aligns with the overall objectives and missions of the organization. This involves identifying key strategic priorities, such as increasing profitability, improving customer satisfaction, or expanding market share, and developing BPs that support these priorities. By aligning decision-making with strategic objectives, organizations can optimize their resources and focus their efforts on activities that drive long-term success.

For experimental requirements, a set of BPs aimed at setting goals have been identified and consisely expressed. Than, long-term strategies, as well as the outlining needed actions to achieve those goals are formulated and implemented. After that, the developed chatbot BP-DSS3 is deployed for assisting users in decision-making tasks. For space reasons, the experimental results are illustrated basing only on two important BPs related to strategic planning. The first BP concerns the Alignment with Organizational Objectives (AOO), while the second one focuses on Risk Management and Contingency Planning (RMCP).

For each BP, relevant requests have been formulated, and submitted to both chatbots, i.e.; chatGPT and BP-DSS3. Finally, the output sequences are analyzed and compared.

Hereafter, the conversations scenarios generated during dialogue sessions between users and the two chatbots are depicted and discussed.

6.4.2.1 Scenario 1: Alignment with organizational objectives (AOO)

As observed in Fig. 6.12, Fig. 6.13 and Fig. 6.14, the answers returned by the general chat-GPT for AOO concerns are general and very vague. Consequently, they can't satisfy the decision-making needs in an effective way. However, after the fine-tuning

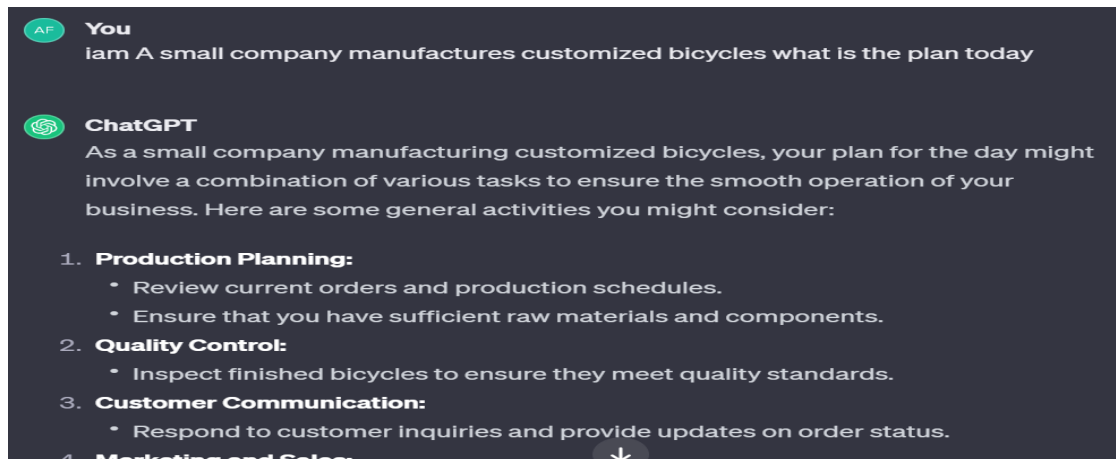


FIGURE 6.12: Responses of the chat-GPT to questions of the AOO BP (part 01)

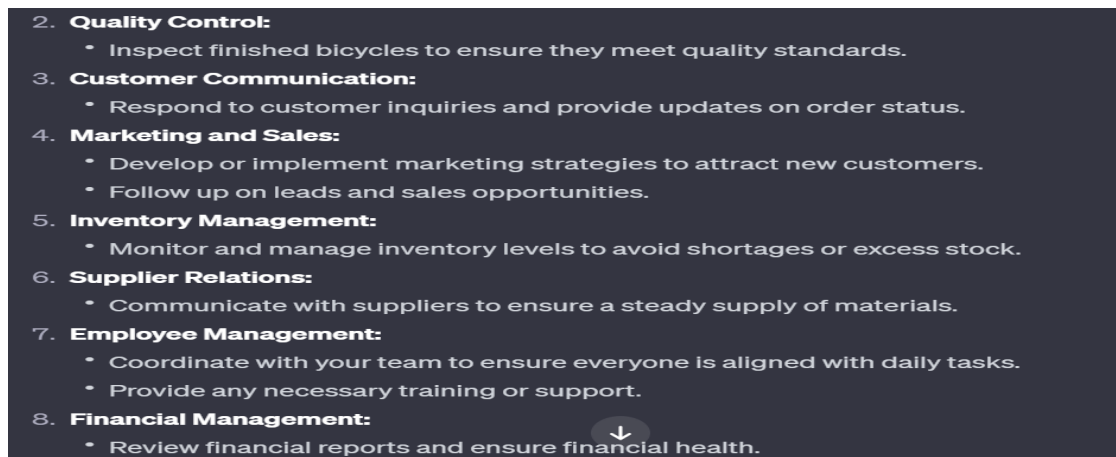


FIGURE 6.13: Responses of the chat-GPT to questions of the AOO BP (part 02)

process, the developed chatbot BP-DSS3 provides more accurate answers that can be exploited immediately by firms managers. As illustrated in Figures 6.15 and 6.16, the exact business objectives are concisely considered by the developed BP-DSS3 chatbot, which supports the managers making informed decisions during their daily activities. In fact, this chatbot can facilitate strategy planning by simulating different business



FIGURE 6.14: Responses of the chat-GPT to questions of the AOO BP (part 03)

scenarios, relying on historical knowledge of market trends. This capability helps business leaders anticipate potential challenges related to threats and opportunities, and consequently develop robust strategies.

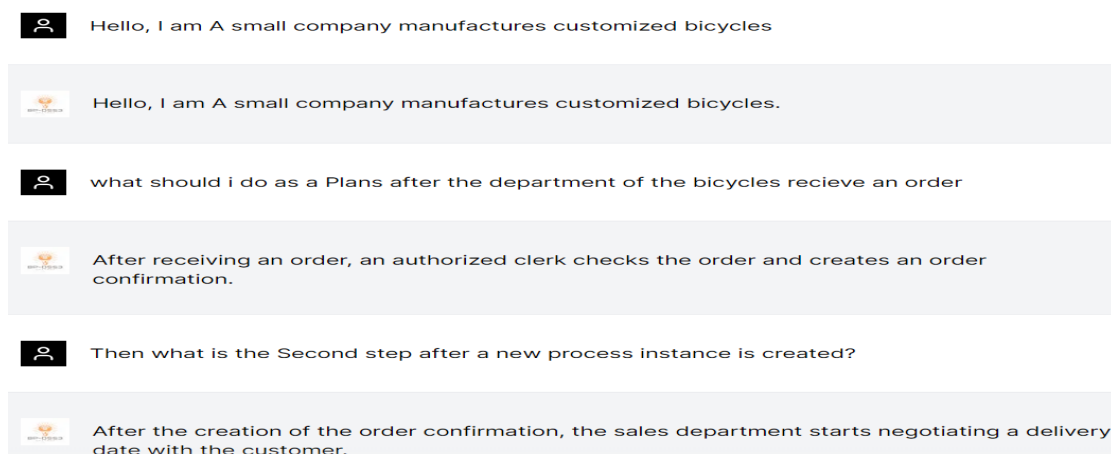


FIGURE 6.15: Response of the BP-DSS3 chatbot for the AOO business process (part 01)

The following case study is dedicated to the evaluation of the second BP that expresses strategic planning related to Risk Management and Contingency Planning (RMCP).

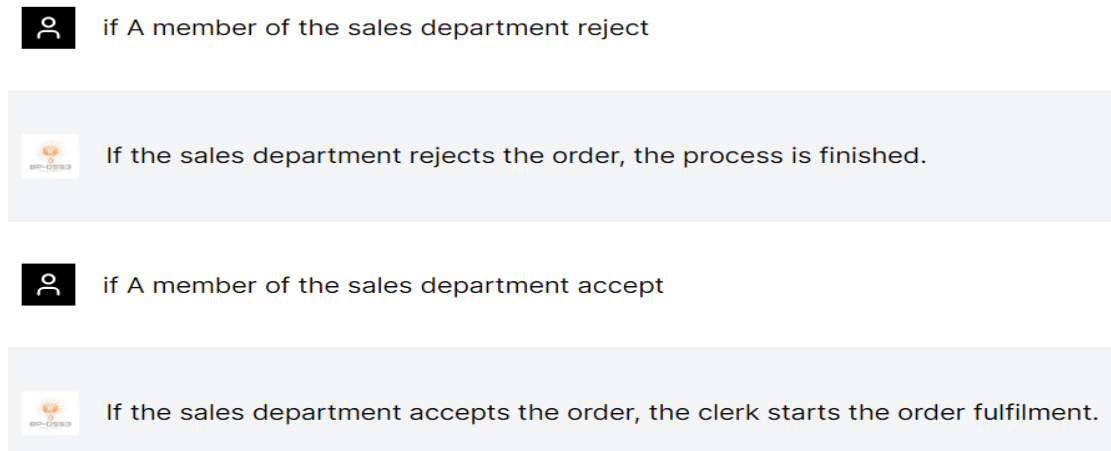


FIGURE 6.16: Response of the BP-DSS3 chatbot for the AOO business process (part 02)

6.4.2.2 Scenario 2: Risk management and contingency planning (RMCP)

Effective strategic planning involves identifying and attenuating risks associated with BP decision-making. This includes assessing potential risks and uncertainties, such as market volatility, regulatory changes, or technology disruptions, and developing contingency plans to address such challenges. By proactively managing risks, organizations can minimize the impact of unforeseen events on their operations and maintain business continuity.

The response of the general chat-GPT to the particular concern related to *RMCP*: "Could I have a loss of money today ?" is shown in Fig. 6.17. It's observed that chat-GPT is enable to handle such a question and it advises the user to consult a financial expert. However, the same previous issue was effectively handled and resolved with

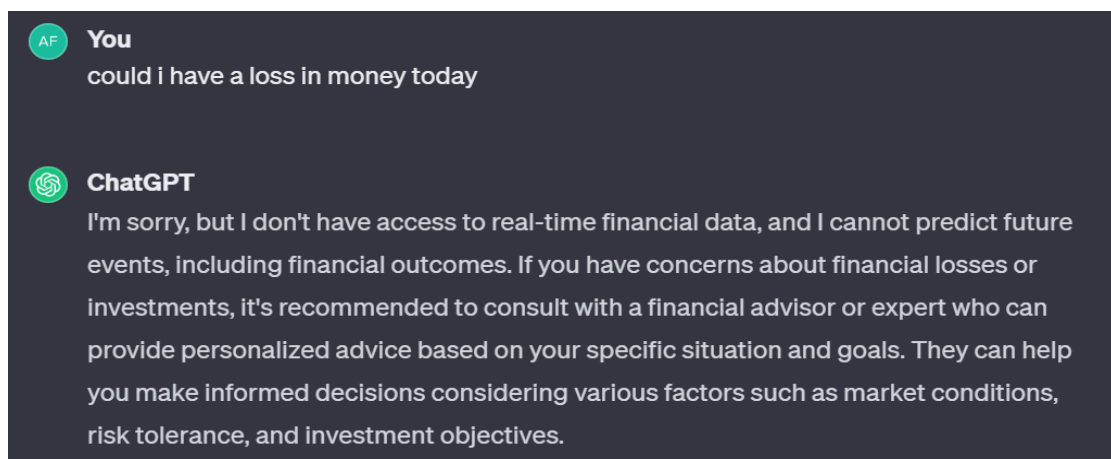


FIGURE 6.17: Response of the general chat-GPT to suggest the possible future risks

the developed BP-DSS3, which can identify the potential risks and their impact on the

company success and efficiency. As illustrated in the Fig. 6.18 and Fig. 6.19, the chatbot BP-DSS3 exhibits remarkable skills in handling firms potential risks and for addressing contingency plans.

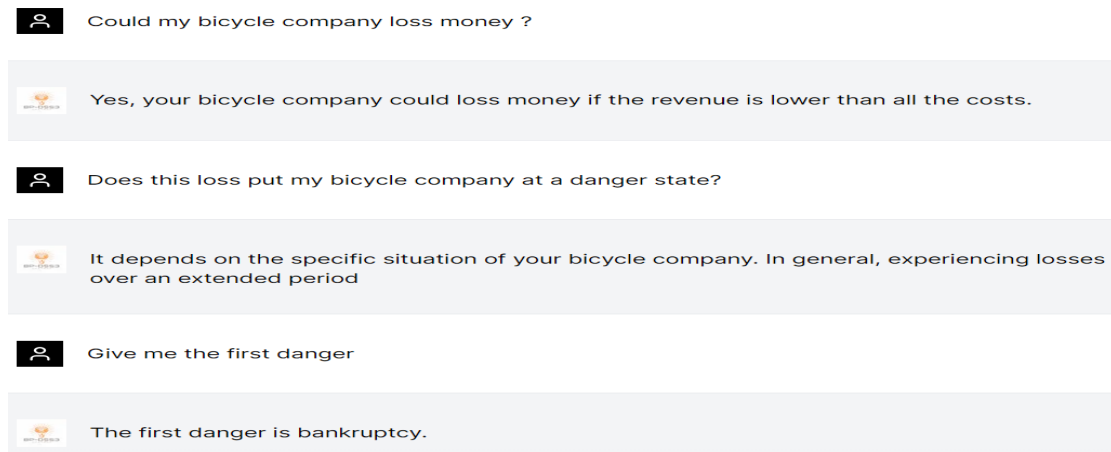


FIGURE 6.18: Response of the BP-DSS3 chatbot for predicting future risks (part 01)

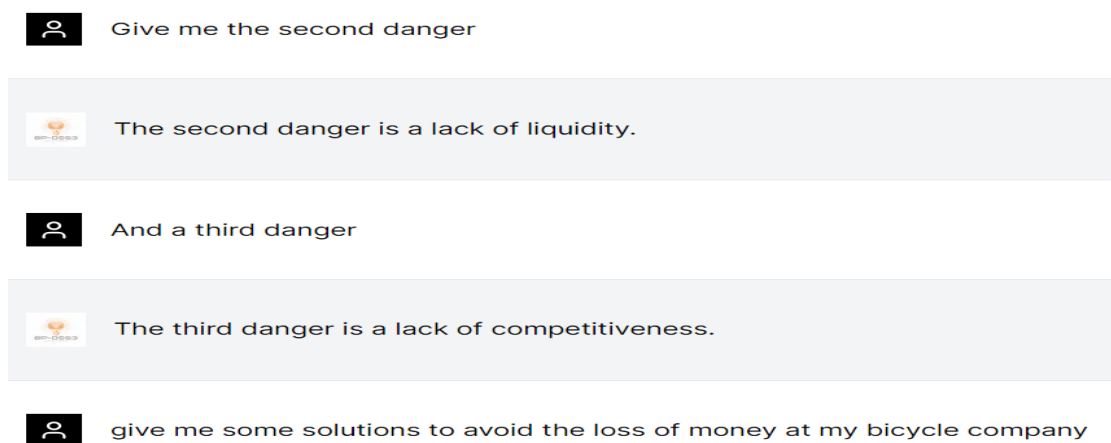


FIGURE 6.19: Response of the BP-DSS3 chatbot for predicting future risks (part 02)

In summary, the proposed BP-DSS3 provides more detailed and tailored replies to the target domain concerns, comparing with the general chat-GPT.

To assess a model performances, such as BP-DSS3, a set of known metrics is used in order to compare the model's predictions to actual labels. In this regard, the previous test case studies are evaluated in the following subsection through four performance metrics, namely: (*precision* [198], *recall* [198], *f1.score* [198] and *accuracy* [197]).

6.4.2.3 Performance metrics evaluation

In the general context of a chat-bot based decision-making system, such as BP-DSS3, the performance score must indicate the abilities of the model to effectively perform

TABLE 6.1: The confusion matrix of BP-DSS3 and Chat-GPT

BP-DSS3	Predicted			Chat-GPT	Predicted		
		True	False			True	False
Actual	True	8 (TP)	2 (FN)	Actual	True	1 (TP)	9 (FN)
	False	0 (FP)	2 (TN)		False	2 (FP)	0 (TN)

the users' requests. Although the developed system can be deployed and exploited for decision-making purposes, it would be appropriate to analyse its real performances in order to further improve its capabilities. In this perspective, the confusion matrix [199] serves as a diagnostic useful tool which highlights the distribution of correct and confused predictions generated by the model. The set of evaluation values contained in this matrix allows stakeholders appreciate the capabilities and weakness of the proposed model. Furthermore, the conventional metrics, such as precision, recall, and F1-score can provide deeper insights regarding the model's reasoning abilities. In this regard, the chatbot's responses resulting from the prior two case studies (*i.e.*; *AOO* and *RMCP*) were exploited to evaluate the BP-DSS3's performances.

As shown in the table 6.1, the accuracy matrix values for the BP-DSS3 model are measured and compared with those of Chat-GPT in order to assess the predicted values against actual values for the two used datasets.

It is important to clarify that in the particular context of BP decision-making, the explicit signification of the values of the different metrics contained in table 6.1 are the following.

- **Actual**

- **True:** Expresses the questions derived from the test cases contained in (*BPI19* or *PET*).
- **False:** corresponds to the questions provided outside of the dataset's test cases (*PET* or *BPI19*)

- **Performance metrics**

- **TP:** Represents the number of correctly detected responses from the test cases contained in (*BPI19* or *PET*)
- **TN:** Represents the number of correctly detected responses of the questions provided outside of the dataset's test cases (*PET* or *BPI19*).
- **FP (*Type I error*):** Expresses the number of responses detected by the chatbot as correct for the questions provided outside of the dataset's test cases (*PET* or *BPI19*).

TABLE 6.2: A performance evaluation of the test case studies

Metrics	Precision	Recall	F1.Score	Accuracy
Chat-GPT	0.33	0.1	0.09	0.08
BP-DSS3	1	0.8	0.89	0.83

- **FN (*Type II error*)**: Expresses the number of undetected responses by the chatbot for the questions provided inside of the dataset’s test cases (*PET or BPI19*).

The analysis of the results displayed in the previous table allows establishing the following interpretations of the performances achieved by the chatbots BP-DSS3’s and Chat-GPT.

- **True Positive (TP)**: BP-DSS3 correctly predicted 8 positive instances (*i.e.; it successfully identified 8 correct results*), while Chat-GPT correctly predicted only 1 positive instance (*i.e.; it recognized just 1 correct positive result*).
- **True Negative (TN)**: BP-DSS3 correctly identified 2 negative instances, while Chat-GPT did not correctly identify any negative instances.
- **False Positive (FP)**: BP-DSS3 did not make any false positive errors (*i.e.; it didn’t mistakenly detect any negative instances as positive*), while Chat-GPT detects 2 cases.
- **False Negative (FN)**: BP-DSS3 failed to predict 2 positive instances (*i.e.; it misclassified 2 positive cases as negative*), while Chat-GPT misclassified 9 positive instances as negative, showing a significant number of missed positive predictions.

Basing on the previous interpretations, it appears clearly that the developed chatbot BP-DSS3 outperforms Chat-GPT, particularly, in correctly identifying positive cases. In fact, Chat-GPT missed a large number of positive instances (9 **false negatives**) and also incorrectly flagged 2 negative instances as positive, demonstrating areas where its performances could be improved.

The advanced exploitation of the results obtained in table 6.1 allows using more accurate evaluation metrics. In fact, four performance metrics (*precision, recall, f1.score and accuracy*) are calculated to understand the strengths and the weaknesses of the chat-based system. These action helps assessing the overall effectiveness of each system in handling test case studies.

The table 6.2 summarizes the used four performance measures and their interpretation is the following.

- **Precision :** The BP-DSS3 have a precision of 1, which indicates that when it predicted something as positive, it was always correct (*no false positives*). However, Chat-GPT had a low precision, meaning only 33% of the instances it predicted as positive were actually correct.
- **Recall:** Chat-GPT's recall is 0.1, meaning it identified only 10% of the actual positive cases. It missed a substantial number of positive cases (high false negatives). On the other hand, BP-DSS3's recall is 0.8, meaning it successfully identified 80% of the positive cases, performing much better than Chat-GPT in capturing true positives.
- **F1-Score:** Chat-GPT's F1-score is 0.15, indicating very poor performance due to its low recall. This means that Chat-GPT struggles with both correctly identifying and predicting positive cases. But, BP-DSS3's F1-score is 0.89, meaning it performs much better in balancing both precision and recall. It effectively identifies most of the positive cases with few errors.
- **Accuracy:** Chat-GPT's accuracy is 0.08, meaning it correctly classified only 8% of all instances. This highlights Chat-GPT's low overall performance in predicting the correct outcomes. In the contrary, BP-DSS3's accuracy is 0.83, meaning it correctly classified 83% of all instances, which is much higher and indicates stronger overall performance.

From the above interpretation, the BP-DSS3 is clearly the superior system in this evaluation, achieving high performance in identifying and correctly predicting the results of the test cases, whereas Chat-GPT suffers from low recall and overall accuracy. Overall, BP-DSS3 demonstrates high performance in BP decision-making, while Chat-GPT requires substantial improvements to be considered effective in this area. In this context, the following conclusions are achieved:

- BP-DSS3 significantly outperforms Chat-GPT in all key performance metrics (*Precision, Recall, F1-Score, and Accuracy*).
- BP-DSS3's perfect precision (1.0) and high recall (0.8) suggest it is highly reliable in making correct predictions with very few errors, while Chat-GPT struggles with both false positives and false negatives.
- Chat-GPT's low recall (0.1) and F1-Score (0.09) indicate that it misses a large number of positive cases and fails to achieve a good balance between precision and recall.

After validating the proposed BP-DSS3 model, in the next subsection we will discuss and highlight the system's limitations and challenges.

6.4.3 Limitations of BP-DSS3

The previous experiments involving real-BP datasets proves that the BP-DSS3 model's adaptability are very satisfactory. In fact, the training step reinforced with the fine-tuning mechanism allow the conceived chatbot to avoid memorizing specific data points, while retaining the ability to generalize well across unseen previous data. Hence, it provides robust results even with fast training cycles. Thus, the experiment confirms the model's resilience against over-fitting, regardless of the high accuracy achieved during the fine-tuning stage.

After having illustrated the performance of the proposed chatbot and its abilities to handle decision-support concerns, in what follows the limitations and challenges inherent to BP-DSS3 chatbots' are discussed.

1. **Model explainability:** The explainability of chatbots, and AI generative models in general, pertains to the ability to understand and interpret the decisions made by the model. This allows the chatbot gaining the user confidence and helps him make good and fast decision, and furthermore preserve the transparency. In fact, while the chatbot can generate responses that are contextually relevant and coherent, understanding how and why it arrives at a particular response can be challenging. This lack of transparency can be problematic, especially in sensitive or high-stakes applications where users need to trust the reasoning behind the model's decisions.
2. **Monitoring and maintenance:** Regular monitoring and updating are needed to ensure that the chatbot performs as expected and generates pertinent results. Thus, the utilization and maintenance of the chatbot for a long time is required to handle refreshed information and features.
3. **Handling complex scenarios:** In numerous scenarios, the chatbots are limited in handling highly complex or ambiguous situations. In this context, human intervention may be necessary for operating particular complex decision-making activities.
4. **User experience insights and judgements:** User judgment plays a crucial role in fine-tuning GPT-3.5-turbo for chatbot applications in the BP decision-making field. Such user contribution helps refining the model's responses to ensure

that they are contextually relevant and aligned with the organization's objectives. By incorporating user feedback, the fine-tuning process becomes more adaptive, leading to improved accuracy, trustworthiness, and decision quality in BP scenarios over a long time utilization.

Additionally, the following ethical considerations of the BP-DSS3 Chatbot are considered as potential issues and challenges to be addressed by different stakeholders in the field.

1. **Potential biases persist:** Algorithmic biases and limitations in language understanding further exacerbate the explainability issue, resulting often in misinterpretations or inappropriate responses. Moreover, feedback loops can reinforce and amplify existing biases over time. Addressing this challenge requires a holistic approach, including diverse and representative training data, bias detection and attenuation techniques, human oversight and intervention, as well as transparency measures to ensure efficient chatbot interactions.
2. **Data privacy and security:** Handling sensitive information requires robust and adequate security policies. Thus, compliance with data protection regulations is crucial.
3. **Transparency:** Maintaining transparency about the capabilities and limitations of the chatbot. Users should be aware that they are interacting with an AI system and not a human, which helps set appropriate expectations regarding the quality and reliability of the advice provided.
4. **Accountability:** Establishing clear lines of accountability for decisions made based on the chatbot's recommendations. Organizations should ensure that there is a human oversight mechanism to review and validate critical decisions to avoid disasters leading to loss of resources and human lives due to the systematic use of automatic responses.
5. **User Autonomy:** Respecting user autonomy by encouraging individuals to make their own decisions rather than overly relying on the chatbot's recommendations. Providing options for users to seek human assistance when needed can foster a more balanced decision-making process.

In this circumstance, the chatbot BP-DSS3 can be maintained and updated over time by periodically fine-tuning GPT-3.5-turbo with new business process data, ensuring that the model adapts to evolving decision-making needs and trends. Continuous monitoring and incorporating feedback loops from real-world interactions can further enhance its accuracy and relevance in BP decision-making.

6.5 The process of deploying BP-DSS3 in real-world

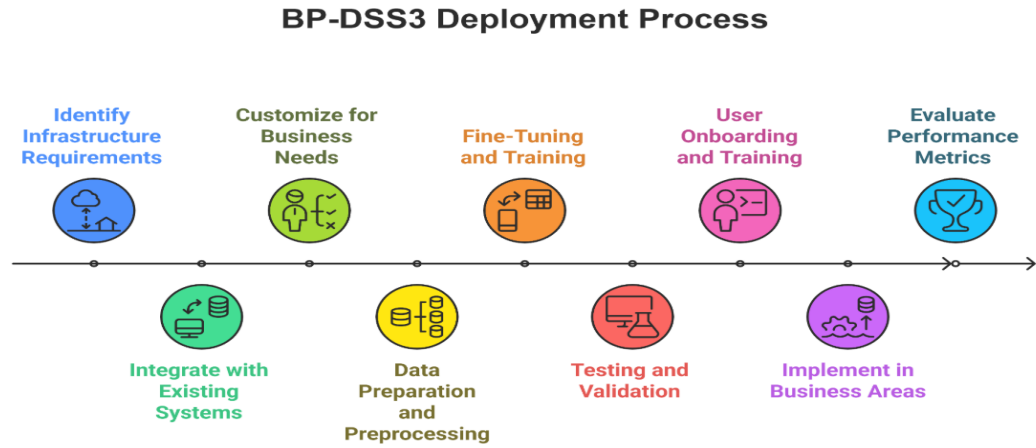


FIGURE 6.20: Deploying the BP-DSS3 in a real-company

The figure 6.20 illustrates the BP-DSS3 Deployment Process, presenting a structured approach to implementing the model in a real-world business environment. The process follows a sequence of steps, each representing a crucial phase in deployment.

1. The first step in the process is *identifying infrastructure requirements*. This phase ensures that the organization has the necessary computing power, storage, and network capabilities to support the BP-DSS3 system. The decision between cloud-based or on-premises deployment [] plays a significant role in shaping the system's scalability and security. Organizations must assess their existing infrastructure to determine compatibility and identify gaps that need to be addressed.
2. Following this, the system must be *integrated with existing business applications and information technologies systems*. This integration enables seamless data flow between BP-DSS3 and enterprise solutions. The ability to interact with existing databases and automation tools is crucial for enhancing operational efficiency. Businesses need to establish robust data pipelines to ensure the continuous exchange of information between BP-DSS3 and other company systems.
3. Once integration is established, customization for business needs becomes a priority. BP-DSS3 must be tailored to meet the specific operational requirements of an organization. This includes modifying algorithms, defining business rules, and adjusting data models to align with the company's goals. Customization ensures that the decision support system is not a generic tool but one that fits the organization's workflows, policies, and strategic objectives.

4. **Data preparation and preprocessing** form the foundation of the system's accuracy and efficiency. This stage involves collecting relevant data, cleaning inconsistencies, and structuring information to ensure it is usable. High-quality data is essential for the system to generate meaningful insights. Feature engineering techniques may be applied to enhance the dataset and improve decision-making capabilities.
5. **Fine-tuning and training of the BP-DSS3** system are essential to optimize its performance. This phase involves selecting the most appropriate machine learning models and adjusting their parameters to improve accuracy. Training the system with historical and real-time data enables it to recognize patterns and make informed decisions. Continuous monitoring during this stage helps refine the model to adapt to the specific needs of the organization.
6. Once the system is trained, **rigorous testing and validation** are conducted to ensure reliability. This includes running simulations, comparing outputs with expected results, and validating predictions against real-world scenarios. Pilot deployments within a controlled environment allow businesses to identify potential errors and fine-tune the system before full-scale implementation. Ensuring the system's robustness through extensive testing reduces the risk of operational disruptions.
7. The next step involves **user on-boarding and training**, which is critical for successful adoption. Employees must be educated on how to use BP-DSS3 effectively to support their decision-making processes. Training programs, workshops, and user manuals help staff understand the system's functionalities. Establishing a support mechanism during this stage ensures that users can seek assistance when needed, fostering confidence in the new system.
8. After successful on-boarding, the **BP-DSS3 model is implemented in business** areas where it is expected to provide value. This may include departments such as finance, operations, marketing, or customer service. The system's deployment in real-world scenarios allows organizations to leverage its predictive and prescriptive analytics capabilities to improve efficiency and drive better business outcomes.
9. The final phase of the deployment process is evaluating **performance metrics**. Organizations must measure the system's effectiveness using key performance indicators such as accuracy, recall, and precision. By continuously assessing the system's performance, businesses can identify areas for improvement and refine the model further. Performance evaluation ensures that BP-DSS3 remains relevant and continues to support data-driven decision-making in the long run.

This structured deployment process ensures that BP-DSS3 is not just a theoretical framework but a practical tool that can be integrated into real business environments. By following these steps, organizations can maximize the potential of BP-DSS3, improve decision-making, and enhance operational efficiency.

6.6 Conclusion

Effective and efficient decision-making is at the heart of successful business activities. The integration of advanced technologies, such as NLP, offers novel opportunities to enhance decision support systems. This chapter presents the second contribution of the thesis by fine-tuning the GPT-3.5 language model to develop a specialized chat-bot BP-DSS3 that can perform decision-making support in the context of business processes management. After the training step on adequate data, the BP-DSS3 is deployed for handling real-world scenarios. The first experimental results demonstrate the efficiency of BP-DSS3 compared to the general chat-GPT for decision-making tasks related to BPs.

The following issues outline the future research directions. **(i)** connecting the BP-DSS3 with existing business databases and systems in order to validate the achieved finding with more case studies. **(ii)** Including a self-learning feature process in the chat-bot to adjust and adapt the system capabilities over time through iterative training and refinement procedures. **(iii)** Further, the ultimate objective is to include the BPs modeling in the BP-DSS3, which will enable users to access the company's general process abstract model whenever needed. **(iv)** Moreover, we plan to integrate the user judgment in the conceived system. **(v)** Finally, expanding future work to consider scalability, adaptability to various industries, and integration with other business intelligence tools would strengthen the development potential of chat-bots.

General Conclusion

In this thesis, the decision-support systems related to BPs are studied and enhanced with advanced AI technologies. Novel methods for improving decisions are explored and rich knowledge representations, combined with efficient software tools are suggested. With a focus on deploying AI techniques, this thesis constitutes a significant contribution in the field. It makes use of developments in machine learning to address problems at the nexus of Knowledge representation and NLP. In order to fully utilize KGs and effectively reflect managers decisions, the thesis underlines the importance of effective knowledge presentation consolidated with AI techniques.

The three following major contributions are presented in the thesis:

1. First, knowledge representation of BPs models is enriched. In this perspective, abstract specifications of BP models are specified as KG-BP, which provide a structured and semantically rich abstraction of BP specifications. This representation captures intricate relationships among BP entities, activities, and constraints, facilitating advanced reasoning and predictive analytics.
2. Additionally, the second contribution aims to integrate AI techniques such as Graph Neural Networks (GNNs) and Capsule Networks to strengthen knowledge propagation and contextual understanding of this KG-BP representation. This enriched representation supports decision-making, process optimization, and conformance checking by uncovering hidden dependencies and enabling dynamic adaptability in BP management. Experimental results prove the effectiveness of the G-Caps-NN to take a BP decision based on the KG-BP representation.
3. Third, a novel chat-bot is developed and implemented as an essential tool for decision-making in the context of BP management. This chat-bot leverages advanced NLP techniques and is fine-tuned using domain-specific BP data to enhance its accuracy and relevance. The fine-tuning process involves training the chat-bot on structured and unstructured BP datasets, allowing it to provide context-aware

recommendations, process optimization insights, and predictive analytics. Experimental results demonstrate the chat-bot's effectiveness in improving decision support, achieving high accuracy in aligning business objectives, risk management, and contingency planning. Through real-world deployment, the chat-bot exhibits enhanced adaptability, significantly outperforming generic AI-driven assistants in BP-related queries and decision-making scenarios.

Future perspectives

The findings of this thesis open up new perspectives for further exploration and investigation. In fact, the obtained results, encourage researchers to explore the untapped potential of KGs and their applications in various domains. Particularly, in the BP and Decision-support fields.

The integration of BP-DSS3 and KG-BP provides a robust ecosystem for BP management, where BP-DSS3 excels in real-time decision support through conversational AI, and KG-BP, driven by G-CAPS-NN, offers predictive insights and a deeper understanding of process dynamics. Together, these frameworks empower organizations to automate decision-making, improve efficiency, and adapt to evolving business challenges with greater agility.

Future research directions could focus on further enhancing the scalability and real-time adaptability of these systems, integrating them with dynamic, multi-source data streams, and exploring their applications in broader industries. Furthermore, as shown in the Fig.6.21 and based on the aforementioned findings, BP-DSS3 and KG-BP have the potential to redefine the landscape of business process management and decision-making. In this context, the combination is significant enough to create a new technology called Chat-bot-G-Caps-NN as a BP-DSS.

In this sense, one approach for combining those major architectures is to utilize H-caps-GNN as the chat-bot's kernel model rather of GPT3.5-turbo.

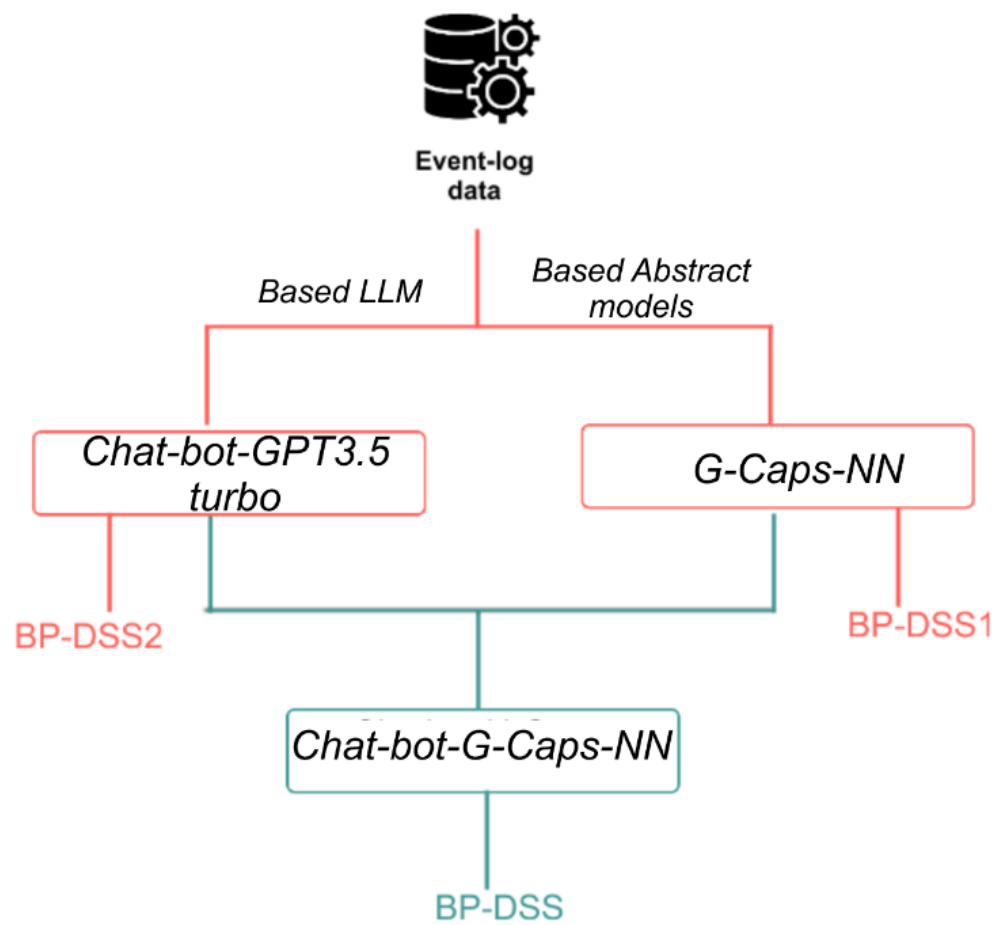


FIGURE 6.21: The Chatbot-G-Caps-NN architecture for BP decision making

Bibliography

- [1] Written By Haisam Abdel Malak. Bpm lifecycle: 5 stages to business process excellence. *theecmconsultant*, October 15, 2024.
- [2] Albert Fleischmann, Werner Schmidt, and Christian Stary. (re-) justifying bpm: A quest for the interaction turn reviewing subject-oriented bpm. In *2013 IEEE 15th CBI*, pages 228–233. IEEE, 2013.
- [3] MV Manoj Kumar, Likewin Thomas, and B Annappa. Capturing the sudden concept drift in process mining. *Proceedings of the Algorithms & Theories for the Analysis of Event Data (ATAED’15), Brussels, Belgium*, pages 22–23, 2015.
- [4] Kosha Mehta. Decision support system. *WallStreetMojo*.
- [5] Gergana Petkova. What is the data, information, knowledge, wisdom (dikw) pyramid? <https://www.ontotext.com/knowledgehub/fundamentals/dikw-pyramid/>, 2021.
- [6] Sayantini. What is knowledge representation in ai? techniques you need to know. <https://www.edureka.co/blog/knowledge-representation-in-ai/>, Dec 05,2024.
- [7] John Stegeman. What is a knowledge graph? <https://neo4j.com/blog/what-is-knowledge-graph/>, 2024.
- [8] Isaac Kargar. Graphs to graph neural networks: From fundamentals to applications — part 2b: Knowledge graphs. <https://kargarisaac.medium.com/graphs-to-graph-neural-networks-from-fundamentals-to-applications-part-2b-knowledge-graphs-841f21dca7c3>, 2023.
- [9] Samadrita Ghosh. Knowledge graphs — what, why, and how. <https://samadritaghosh.medium.com/knowledge-graphs-what-why-and-how-84f920316ca5>, 2022.
- [10] Martin Grandjean. Knowledge is a network perspective on archival and encyclopedic organization. *Les Cahiers du Numérique*, 10(3):37–54, 2014.

- [11] Ajitesh Kumar. *Difference between Machine Learning Traditional Software*. <https://vitalflux.com/difference-between-machine-learning-traditional-software/>, October 30, 2019.
- [12] SAGAR SHARMA. What the hell is perceptron? In *Toward datascience*. <https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53>, Sep 9, 2017.
- [13] Layers in artificial neural networks (ann). In *Geeks for Geeks*. <https://www.geeksforgeeks.org/layers-in-artificial-neural-networks-ann/>, 19 Jul, 2024.
- [14] Rukshan Pramoditha. Overview of a neural network’s learning process. In *Toward datascience*. <https://medium.com/data-science-365/overview-of-a-neural-networks-learning-process-61690a502fa>, Feb 1, 2022.
- [15] Deepanshi. Beginners guide to artificial neural network. In *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/05/beginners-guide-to-artificial-neural-network/>, 31 May, 2021.
- [16] Arne Wolfewicz. Deep learning vs. machine learning – what’s the difference? In *levity*. <https://levity.ai/blog/difference-machine-learning-deep-learning>, September, 2024.
- [17] Generative and discriminative models. In *LearnOpenCV Team*. <https://learnopencv.com/generative-and-discriminative-models/>, May 10, 2021.
- [18] B. Chen and L. Guangquan et al. Dgrl: text classification with deep graph residual learning. In *ICADMA*, pages 83–97, 2020.
- [19] Gabriele Corso, Hannes Stark, Stefanie Jegelka, Tommi Jaakkola, and Regina Barzilay. Graph neural networks. *Nature Reviews Methods Primers*, 4(1):17, 2024.
- [20] Language models, explained: How gpt and other models work. *Data science*, 2023.
- [21] Tian Ma, Wanwan Wang, and Yu Chen. Attention is all you need: An interpretable transformer-based asset allocation approach. *International Review of Financial Analysis*, 90:102876, 2023.
- [22] Marco Ramponi. Ai trends in 2024: Graph neural networks. *AssemblyAI*, Feb, 2024.
- [23] Josh Howarth. 9 top business trends (2024–2025). *ExplodingTopics*, May 2024.

- [24] Lakshmi. Activation functions neural networks: A quick complete guide. In *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/>, 08 Aug, 2024.
- [25] Daniel Deutch and Tova Milo. Querying structural and behavioral properties of business processes. pages 169–185, 09 2007. ISBN 978-3-540-75986-7. doi: 10.1007/978-3-540-75987-4_12.
- [26] Wil M. P. van der Aalst. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 3642193447.
- [27] Ghazaleh Khodabandelou and al. A novel approach to process mining: Intentional process models discovery. In *2014 IEEE Eighth International Conference on RCIS*, pages 1–12, 2014.
- [28] Martin Bauer and et al. Senderovich. How much event data is enough? a statistical framework for process discovery. In *AISE: 30th International Conference, Tallinn, Estonia, June 11-15, 2018, Proceedings 30*, pages 239–256. Springer, 2018.
- [29] Fani Sani.al. Subgroup discovery in process mining. In *Business Information Systems: 20th International Conference, BIS 2017, Poznan, Poland, June 28–30, 2017, Proceedings 20*, pages 237–252. Springer, 2017.
- [30] Wacharawan Intayoad and Till Becker. Applying process mining in manufacturing and logistic for large transaction data. In *AIL Conference LDIC 2018, Bremen, Germany*, pages 378–388. Springer, 2018.
- [31] Wil MP van der Aalst. Object-centric process mining: Dealing with divergence and convergence in event data. In *SEAFM: 17th International Conference, SEFM 2019, Oslo, Norway, September 18–20, 2019, Proceedings 17*, pages 3–25. Springer, 2019.
- [32] Mohit Sewak. *Deep Reinforcement Learning*. Springer, 2019.
- [33] Dominique Sommers, Vlado Menkovski, and Dirk Fahland. Supervised learning of process discovery techniques using graph neural networks. *Information Systems*, page 102209, 2023.
- [34] Yang Li, Wei Zhao, Erik Cambria, Suhang Wang, and Steffen Eger. Graph routing between capsules. *Neural Networks*, 143:345–354, 2021.
- [35] Bayu Adhi Tama, Marco Comuzzi, and Jonghyeon Ko. An empirical investigation of different classifiers, encoding, and ensemble schemes for next event prediction using business process event logs. *ACM Transactions on TIST*, 2020.

- [36] C.Affi and A.KHEBIZI. A road-map for mining business process models via artificial intelligence techniques. *IJAM*, 5, 2022.
- [37] Johannes Tenschert and al. Enterprise system capabilities for organizational change in the bpm life cycle. In *BPM Workshops*, pages 70–82. Springer, 2020.
- [38] Alireza Alibakhshi and Erfan Hassannayebi. Towards an enhanced next activity prediction using attention based neural networks. *Cluster Computing*, 28(1):1–25, 2025.
- [39] Ivan Compagnucci, Flavio Corradini, Fabrizio Fornari, and Barbara Re. A study on the usage of the bpmn notation for designing process collaboration, choreography, and conversation models. *Business & Information Systems Engineering*, 66(1): 43–66, 2024.
- [40] Simone Agostinelli and al. Process mining meets model learning: Discovering deterministic finite state automata from event logs for business process analysis. *IS*, 114, 2023.
- [41] AK Alves .al De Medeiros. An outlook on semantic business process mining and monitoring. In *On the MMIS OTM 2007*, pages 1244–1255. Springer, 2007.
- [42] Zineb Lamghari, Maryam Radgui, and al. Defining business process improvement metrics based on bpm life cycle and process mining techniques. *IJBPM*, 9(2): 107–133, 2019.
- [43] Christopher van Dun, Linda Moder, and al. Processgan: Supporting the creation of business process improvement ideas through generative machine learning. *DSS*, 165, 2023.
- [44] Belén Ramos Gutiérrez. Improving data preparation for the application of process mining. 2023.
- [45] Güzin Özdağoglu and al. Identifying and prioritising process portfolio for sustaining an effective business process management lifecycle. *JMCDA*, 2023.
- [46] Maxim Vidgof, Stefan Bachhofner, and Jan Mendling. Large language models for business process management: Opportunities and challenges. *arXiv preprint arXiv:2304.04309*, 2023.
- [47] Aleksandra Revina and Ünal Aksu. An approach for analyzing bp execution complexity based textual data and event log. *IS*, 114, 2023.
- [48] André Filipe Domingos Gomes and al. Comparative analysis of process mining algorithms in process discover. In *NTDTTEAI: The DITTET Collection 1*. Springer, 2022.

- [49] Adriano Augusto, Josep Carmona, and Eric Verbeek. Advanced process discovery techniques. In *Process Mining Handbook*, pages 76–107. Springer, 2022.
- [50] Lars Reinkemeyer. Status and future of process mining: from process discovery to process execution. In *Process Mining Handbook*, pages 405–415. Springer, 2022.
- [51] Chiara Di Francescomarino and al. Genetic algorithms for hyperparameter optimization in predictive business process monitoring. *Information Systems*, 74: 67–83, 2018.
- [52] Marwan Hassani, Sebastiaan J van Zelst, and Wil MP van der Aalst. On the application of sequential pattern mining primitives to process discovery: Overview, outlook and opportunity identification. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 9(6):e1315, 2019.
- [53] John Saint, Yizhou Fan, Shaveen Singh, Dragan Gasevic, and Abelardo Pardo. Using process mining to analyse self-regulated learning: a systematic analysis of four algorithms. In *LAK21: 11th international learning analytics and knowledge conference*, pages 333–343, 2021.
- [54] Marco Pegoraro, Merih Seran Uysal, and Wil MP van der Aalst. Conformance checking over uncertain event data. *Information Systems*, 102:101810, 2021.
- [55] Alessandro Berti and Wil MP van der Aalst. A novel token-based replay technique to speed up conformance checking and process enhancement. In *Transactions on Petri Nets and Other Models of Concurrency XV*, pages 1–26. Springer, 2021.
- [56] Roberto de Lima-Hernandez and Maarten Vergauwen. A generative and entropy-based registration approach for the reassembly of ancient inscriptions. *Remote Sensing*, 14(1):6, 2021.
- [57] Emmanuel Helm and Ferdinand Paster. First steps towards process mining in distributed health information systems. *International Journal of Electronics and Telecommunications*, 61:137–142, 2015.
- [58] Panagiotis Drakoulogkonas and Dimitris Apostolou. On the selection of process mining tools. *Electronics*, 10(4):451, 2021.
- [59] Sonia Deshmukh, Shikha Gupta, and Naveen Kumar. Ga-prom: A genetic algorithm for discovery of complete process models from unbalanced logs. In *International Conference on Big Data Analytics*, pages 209–218. Springer, 2021.
- [60] Wil MP van der Aalst. Process mining: a 360 degree overview. In *Process Mining Handbook*, pages 3–34. Springer, 2022.

- [61] Leni Sagita Riantini Supriadi, Low Sui Pheng, Leni Sagita Riantini Supriadi, and Low Sui Pheng. Knowledge based decision support system (kdbss). *Business Continuity Management in Construction*, pages 155–174, 2018.
- [62] Joey G Fernando and Myelinda Baldeovar. Decision support system: Overview, different types and elements. *Technoarete Trans. Intell. Data Min. Knowl. Discov.(TTIDMKD)*, 2:13–18, 2022.
- [63] Marko Bohanec, Gennaro Tartarisco, Flavia Marino, Giovanni Pioggia, Paolo Emilio Puddu, Michele Salvatore Schiariti, Anneleen Baert, Sofie Pardaens, Els Clays, Aljoša Vodopija, et al. Heartman dss: A decision support system for self-management of congestive heart failure. *Expert Systems with Applications*, 186:115688, 2021.
- [64] M. Stierle, S. Weinzierl, M. Harl, and M. Matzner. A technique for determining relevance scores of process activities using graph-based neural networks. *DSS*, 144: 113511, 2021.
- [65] Mohamed Khalifa, Mona Albadawy, and Usman Iqbal. Advancing clinical decision support: the role of artificial intelligence across six domains. *Computer Methods and Programs in Biomedicine Update*, page 100142, 2024.
- [66] Asma Mejri, Sonia Ayachi Ghannouchi, and Ricardo Martinho. Flexibility-based selection of paradigms and tools for bp modelling and execution. *International Journal of Business Process Integration and Management*, 11(3):165–178, 2024.
- [67] Alaa Khalaf Hamoud, Kamil Hussein Marwah, Zahraa Alhilfi, and Rabab Hassan Sabr. Implementing data-driven decision support system based on independent educational data mart. *International Journal of Electrical and Computer Engineering*, 11(6):5301, 2021.
- [68] Rayan Alshamrani, Ashrf Althbiti, Yara Alshamrani, Fatimah Alkomah, and Xiaogang Ma. Model-driven decision making in multiple sclerosis research: existing works and latest trends. *Patterns*, 1(8), 2020.
- [69] Kiyohito Fujishima. Knowledge-driven automated service composition as a method for developing decision support systems. *International Journal for Applied Information Management*, 2(1):44–49, 2022.
- [70] Kai Sauerwald, Eda Ismail-Tsaous, and Nina Thorwart. A report on sequential kr-approaches as cognitive logic. In *Proceedings of the joint Workshop on Knowledge Diversity and Cognitive Aspects of KR,(KoDis/CAKR) co-located with the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR2024)*, CEUR-WS. org, 2024.

- [71] Erik Blasch, Robert Cruise, Alexander Aved, Uttam Majumder, and Todd Rovito. Methods of ai for multimodal sensing and action for complex situations. *AI Magazine*, 40(4):50–65, 2019.
- [72] Qinghua Huang and Yongzhen Wang. Knowledge pyramid: A novel hierarchical reasoning structure for generalized knowledge augmentation and inference. *arXiv preprint arXiv:2401.09070*, 2024.
- [73] Anshuman Singh. Knowledge representation in artificial intelligence (ai). <https://www.appliedaicourse.com/blog/knowledge-representation-in-artificial-intelligence/>, December 12, 2024.
- [74] Francesco Berto. *Topics of thought: The logic of knowledge, belief, imagination*. Oxford University Press, 2022.
- [75] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. Do transformers really perform badly for graph representation? *Advances in neural information processing systems*, 34:28877–28888, 2021.
- [76] Barry De Ville. Decision trees. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(6):448–455, 2013.
- [77] Matthias Schonlau. The naive bayes classifier. In *Applied Statistical Learning: With Case Studies in Stata*, pages 143–160. Springer, 2023.
- [78] Mohiuddin Ahmed and al. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, 9(8):1295, 2020.
- [79] What is an expert system?
- [80] A.K. Jain, Jianchang Mao, and K.M. Mohiuddin. Artificial neural networks: a tutorial. *Computer*, 29(3):31–44, 1996. doi: 10.1109/2.485891.
- [81] Su-Hyun Han, Ko Woon Kim, SangYun Kim, and Young Chul Youn. Artificial neural network: understanding the basic concepts without mathematics. *Dementia and neurocognitive disorders*, 17(3):83–89, 2018.
- [82] Benjamin R Mitchell. Overview of advanced neural network architectures. In *Artificial Intelligence in Pathology*, pages 43–58. Elsevier, 2025.
- [83] AG Farizawani, M Puteh, Y Marina, and A Rivaie. A review of artificial neural network learning rule based on multiple variant of conjugate gradient approaches. In *Journal of Physics: Conference Series*, volume 1529, page 022040. IOP Publishing, 2020.

- [84] Katarzyna Janocha and Wojciech Marian Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.
- [85] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.
- [86] Iqbal H Sarker. Ai-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems. *SN Computer Science*, 3(2):158, 2022.
- [87] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN computer science*, 2(6):420, 2021.
- [88] Nitin Liladhar Rane, Mallikarjuna Paramesha, Saurabh P Choudhary, and Jayesh Rane. Machine learning and deep learning for big data analytics: A review of methods and applications. *Partners Universal International Innovation Journal*, 2(3):172–197, 2024.
- [89] Jigna Bhupendra Prajapati, Savan Patel, Richesh Gaurav, Dhvanil Nileshkumar Prajapati, and Kavita Saini. Machine and deep learning (ml/dl) algorithms, frameworks, and libraries. In *Applying Drone Technologies and Robotics for Agricultural Sustainability*, pages 155–172. IGI Global, 2023.
- [90] Abbas Yazdinejad, Ali Dehghantanha, Reza M Parizi, and Gregory Epiphaniou. An optimized fuzzy deep learning model for data classification based on nsga-ii. *Neurocomputing*, 522:116–128, 2023.
- [91] Hanchi Liu, Xin Ma, Yining Yu, Liang Wang, and Lin Hao. Application of deep learning-based object detection techniques in fish aquaculture: a review. *Journal of Marine Science and Engineering*, 11(4):867, 2023.
- [92] Niri Rania, Hassan Douzi, Lucas Yves, and Treuillet Sylvie. Semantic segmentation of diabetic foot ulcer images: dealing with small dataset in dl approaches. In *Image and Signal Processing: 9th International Conference, ICISP 2020, Marrakesh, Morocco, June 4–6, 2020, Proceedings 9*, pages 162–169. Springer, 2020.
- [93] Maryam Dialameh, Ali Hamzeh, Hossein Rahmani, Safoura Dialameh, and Hyock Ju Kwon. Dl-reg: A deep learning regularization technique using linear regression. *Expert Systems with Applications*, 247:123182, 2024.

- [94] Siming Yan, Min Bai, Weifeng Chen, Xiong Zhou, Qixing Huang, and Li Erran Li. Vigor: Improving visual grounding of large vision language models with fine-grained reward modeling. In *European Conference on Computer Vision*, pages 37–53. Springer, 2025.
- [95] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *ANIPS*, 30, 2017.
- [96] Mu Chen, Zaojian Dai, Yong Li, Juling Zhang, Sheng Wang, Jian Zhou, and Fan Wu. Mobile terminal security protection method based on community detection algorithms and graph matching networks. In *International Conference on Artificial Intelligence and Security*, pages 529–542. Springer, 2022.
- [97] Mohsen Saffari, Mahdi Khodayar, Mohammad E Khodayar, and Mohammad Shahidehpour. Behind-the-meter load and pv disaggregation via deep spatiotemporal graph generative sparse coding with capsule network. *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [98] Zhang Xinyi and Lihui Chen. Capsule graph neural network. In *International conference on learning representations*, 2018.
- [99] Junyi Li, Siqing Li, Wayne Xin Zhao, Gaole He, Zhicheng Wei, Nicholas Jing Yuan, and Ji-Rong Wen. Knowledge-enhanced personalized review generation with capsule graph neural network. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 735–744, 2020.
- [100] Giorgio Franceschelli and Mirco Musolesi. Copyright in generative deep learning. *Data & Policy*, 4:e17, 2022.
- [101] Avyay Casheekar, Archit Lahiri, Kanishk Rath, Kaushik Sanjay Prabhakar, and Kathiravan Srinivasan. A contemporary review on chatbots, ai-powered virtual conversational agents, chatgpt: Applications, open challenges and future research directions. *Computer Science Review*, 52:100632, 2024.
- [102] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 3, 2022.
- [103] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.
- [104] Sandra Kublik and Shubham Saboo. *GPT-3*. " O'Reilly Media, Inc.", 2022.

- [105] Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. *Advances in Neural Information Processing Systems*, 36, 2024.
- [106] Mahdi Hajiali, Jorge Ramón Fonseca Cacho, and Kazem Taghva. Generating correction candidates for ocr errors using bert language model and fasttext sub-word embeddings. In *Intelligent Computing: Proceedings of the 2021 Computing Conference, Volume 1*, pages 1045–1053. Springer, 2022.
- [107] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pages 841–852, 2022.
- [108] Eleni Adamopoulou and Lefteris Moussiades. An overview of chatbot technology. In *IFIP international conference on artificial intelligence applications and innovations*, pages 373–383. Springer, 2020.
- [109] Hio Nam Io and Chang Boon Lee. Understanding the adoption of chatbot: A case study of siri. In *Advances in Information and Communication Networks: Proceedings of the 2018 Future of Information and Communication Conference (FICC), Vol. 1*, pages 632–643. Springer, 2019.
- [110] Sanket Sanjay Sadavarte and Eliane Bodanese. Pregnancy companion chatbot using alexa and amazon web services. In *2019 IEEE Pune Section International Conference (PuneCon)*, pages 1–5. IEEE, 2019.
- [111] Chetan Bulla, Chinmay Parushetti, Akshata Teli, Samiksha Aski, and Sachin Koppad. A review of ai based medical assistant chatbot. *Res Appl Web Dev Des*, 3(2):1–14, 2020.
- [112] MD Romael Haque and Sabirat Rubya. An overview of chatbot-based mobile mental health apps: insights from app description and user reviews. *JMIR mHealth and uHealth*, 11(1):e44838, 2023.
- [113] Iryna Pentina, Tyler Hancock, and Tianling Xie. Exploring relationship development with social chatbots: A mixed-method study of replika. *Computers in Human Behavior*, 140:107600, 2023.
- [114] Jianyang Deng and Yijia Lin. The benefits and challenges of chatgpt: An overview. *Frontiers in Computing and Intelligent Systems*, 2(2):81–83, 2022.
- [115] Hayrol Azril Mohamed Shaffril, Samsul Farid Samsuddin, and Asnarulkhadi Abu Samah. The abc of systematic literature review: the basic methodological guidance for beginners. *Quality & Quantity*, 55:1319–1346, 2021.

- [116] Chaima Afifi, Ali Khebizi, and Khaled Halimi. A systematic survey of the business process mining-based approaches. *International Journal of Business Process Integration and Management*, 11(4):314–331, 2024.
- [117] Michael Gusenbauer. Google scholar to overshadow them all? comparing the sizes of 12 academic search engines and bibliographic databases. *Scientometrics*, 118(1):177–214, 2019.
- [118] Di.al. Do you behave always the same? a process mining approach. In *ICPM 2022 International Workshops*, pages 5–17, 2023.
- [119] Andrea Burattin and al. Conformance checking based on multi-perspective declarative process models. *Expert systems with applications*, 65, 2016.
- [120] Wai Lam Jonathan Lee and et al.s Verbeek. Recomposing conformance: Closing the circle on decomposed alignment-based conformance checking in process mining. *Information Sciences*, 466:55–91, 2018.
- [121] van Dongen et al. Conformance checking of mixed-paradigm process models. *IS*, 102, 2021.
- [122] Anna A Kalenkova, van der Aalst, and al. Pm using bpmn: relating event logs and process models. In *Proceedings of the ACM/IEEE 19th ICMDELS*, pages 123–123, 2016.
- [123] Bemthuis.al. An agent-based process mining architecture for emergent behavior analysis. In *2019 IEEE 23rd EDOCW*, pages 54–64. IEEE, 2019.
- [124] Riyanarto Sarno, Fernandes Sinaga, and al. Anomaly detection in business processes using process mining and fuzzy association rule learning. *JBD*, 7:1–19, 2020.
- [125] Sun.al. A graph and trace clustering-based approach for abstracting mined business process models. In *ICEIS*, pages 63–74, 2016.
- [126] De Koninck et al. act2vec, trace2vec, log2vec, and model2vec: Representation learning for business processes. In *BPM: 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9–14, 2018, Proceedings 16*, pages 305–321. Springer, 2018.
- [127] Shraga.al. Process discovery with context-aware process trees. *IS*, 106, 2022.
- [128] Xixi Lu and et al. Fahland. Semi-supervised log pattern detection and exploration using event concurrence and contextual information. In *OTM 2017 Conferences: Proceedings, Part I*, pages 154–174. Springer, 2017.

- [129] Nicola Di Mauro and al. Activity prediction of business process instances with inception cnn models. In *XVIIIth International Conference of the Italian AAI*, pages 348–361. Springer, 2019.
- [130] Taymouri.al. Predictive business process monitoring via generative adversarial nets: the case of next event prediction. In *18th ICBPM 2020*, pages 237–256, 2020.
- [131] Moon.al. Pop-on: Prediction of process using one-way language model based on nlp approach. *Applied Sciences*, 11, 2021.
- [132] Bukhsh.al. Processtransformer: Predictive bp monitoring with transformer network. *arXiv preprint arXiv:2104.00721*, 2021.
- [133] Samia Chehbi Gamoura. Predictive reinforcement learning algorithm for unstructured bp optimisation: case of human resources process. *IJSTDS*, 1, 2021.
- [134] Folino.al. Semi-supervised discovery of dnn-based outcome predictors from scarcely-labeled process logs. *Business & ISE*, pages 1–21, 2022.
- [135] Timo Nolle, Alexander Seeliger, and et al. Binet: multivariate business process anomaly detection using deep learning. In *16th ICBPM*, pages 271–287, 2018.
- [136] Timo Nolle, Stefan Luetzgen, Alexander Seeliger, and Max Mühlhäuser. Analyzing business process anomalies using autoencoders. *Machine Learning*, 107:1875–1893, 2018.
- [137] Krajsic.al. Variational autoencoder for anomaly detection in event data in online process mining. In *ICEIS (1)*, pages 567–574, 2021.
- [138] Elaziz.al. Deep reinforcement learning for data-efficient weakly supervised business process anomaly detection. *Journal of Big Data*, 10(1), 2023.
- [139] Mehrdad Agha Mohammad Ali Kermani, Hamid Reza Seddighi, and Mehrdad Maghsoudi. Revolutionizing process mining: A novel architecture for chatgpt integration and enhanced user experience through optimized prompt engineering. *arXiv preprint arXiv:2405.10689*, 2024.
- [140] Johan Silvander. Business process optimization with reinforcement learning. In *Business Modeling and Software Design: 9th International Symposium*, pages 203–212, 2019.
- [141] Kamil Żbikowski and al. Deep reinforcement learning for resource allocation in bp. In *ICPM 2022*, pages 177–189, 2023.

- [142] Adam et al. Burke. Stochastic process discovery by weight estimation. In *ICPM 2020 International Workshops*, pages 260–272. Springer, 2021.
- [143] Xin Deng and al. Semi-supervised learning approach to discover enterprise user insights from feedback and support. *arXiv preprint arXiv:2007.09303*, 2020.
- [144] Jingjing Gong, Xipeng Qiu, and al. Information aggregation via dynamic routing for sequence encoding. *arXiv preprint arXiv:1806.01501*, 2018.
- [145] Hao Ren and Hong Lu. Compositional coding capsule network with k-means routing for text classification. *PRL*, 160:1–8, 2022.
- [146] Liqiang Xiao, Zhang, and al. Mcapsnet: Capsule network for text with multi-task learning. In *Proceedings of the 2018 conference on EMNLP*, pages 4565–4574, 2018.
- [147] Wei Zhao, Jianbo Ye, Min Yang, and al. Investigating capsule networks with dynamic routing for text classification. *arXiv preprint arXiv:1804.00538*, 2018.
- [148] WL. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159, 2020.
- [149] Ismail Aliyu, A Kana, and Salisu Aliyu. Development of knowledge graph for university courses management. *IJEME*, page 1, 2020.
- [150] Penghe Chen, Yu Lu, Vincent W Zheng, Xiyang Chen, and Boda Yang. Knowedu: A system to construct knowledge graph for education. *Ieee Access*, 6:31553–31563, 2018.
- [151] Bilin Shao, Xiaojun Li, and Genqing Bian. A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *ESA*, 165:113764, 2021.
- [152] Yaoguang Yong, Zheng Yao, and Yawei Zhao. A framework for reviewer recommendation based on knowledge graph and rules matching. In *2021 IEEE ICICSE*, pages 199–203, 2021.
- [153] Zhifei Li, Hai Liu, Zhaoli Zhang, Tingting Liu, and Neal N Xiong. Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE TNNLS*, 33(8):3961–3973, 2021.
- [154] Jin Xu, Shuo Yu, Ke Sun, Jing Ren, Ivan Lee, Shirui Pan, and Feng Xia. Multivariate relations aggregation learning in social networks. In *Proceedings of the ACM/IEEE JCDL in 2020*, pages 77–86, 2020.

- [155] Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. Transforming auto-encoders. In *ANNML-ICANN 2011: 21st ICANN, Espoo, Finland*, pages 44–51. Springer, 2011.
- [156] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.
- [157] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *ICLR*, 2018.
- [158] Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. Aspect-level sentiment analysis using as-capsules. In *The WWWC*, pages 2033–2044, 2019.
- [159] Anne Füßl, Volker Nissen, and Stefan Horst Heringklee. Knowledge graph-based explainable artificial intelligence for business process analysis. *IJSC*, 17(02):173–197, 2023.
- [160] Irlán Grangel-González. *A knowledge graph based integration approach for industry 4.0*. PhD thesis, Universitäts-und Landesbibliothek Bonn, 2019.
- [161] Erik Aumayr, MingXue Wang, and Anne-Marie Bosneag. Probabilistic knowledge-graph based workflow recommender for network management automation. In *2019 IEEE 20th WoWMoM*, pages 1–7. IEEE, 2019.
- [162] Bilal Jan, Haleem Farman, Murad Khan, Muhammad Imran, Ihtesham Ul Islam, Awais Ahmad, Shaukat Ali, and Gwanggil Jeon. Deep learning in big data analytics: a comparative study. *Computers & Electrical Engineering*, 75:275–287, 2019.
- [163] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, and Prafulla Dhariwal. 333 arvind neelakantan, pranav shyam, girish sastry, amanda askell, et al. language models are 334 few-shot learners. *arXiv preprint arXiv:2005.14165*, 335, 2020.
- [164] Ömer Aydın and Enis Karaarslan. Openai chatgpt generated literature review: Digital twin in healthcare. *Available at SSRN 4308687*, 2022.
- [165] S Shreyashree, Pramod Sunagar, S Rajarajeswari, and Anita Kanavalli. A literature review on bidirectional encoder representations from transformers. *Inventive Computation and Information Technologies: Proceedings of ICICIT 2021*, pages 305–320, 2022.
- [166] Abiodun Finbarrs Oketunji. Evaluating the efficacy of hybrid deep learning models in distinguishing ai-generated text.

- [167] Hao Zhang, Yu-N Cheah, Osamah Mohammed Alyasiri, and Jieyu An. Exploring aspect-based sentiment quadruple extraction with implicit aspects, opinions, and chatgpt: a comprehensive survey. *Artificial Intelligence Review*, 57(2):17, 2024.
- [168] Amin Beheshti and al Yang. Processgpt: transforming business process management with generative artificial intelligence. In *2023 IEEE ICWS*, pages 731–739, 2023.
- [169] Kiran Busch, Alexander Rochlitz, Diana Sola, and Henrik Leopold. Just tell me: Prompt engineering in business process management. In *International Conference on Business Process Modeling, Development and Support*, pages 3–11. Springer, 2023.
- [170] Josh Howarth. 57 new artificial intelligence statistics (nov 2024). *ExplodingTopics*, NOvember 2024.
- [171] Cem Dilmegani. 45 statistics, facts forecasts on machine learning [2023]. *AIMultiple research*, Feb 14, 2024.
- [172] V. Aalst, M. Wil, M. Schonenberg, and M. Song. Time prediction based on process mining. *IS*, 36(2):450–475, 2011.
- [173] D. Murillas, E. Reijers, A. Hajo, V. Aalst, and M. Wil. Case notion discovery and recommendation: automated event log building on databases. *KIS*, 62(7): 2539–2575, 2020.
- [174] A. Bolt, M. Leoni, and V. Aalst. A visual approach to spot statistically-significant differences in event logs based on process metrics. In *ICAISE*, pages 151–166, 2016.
- [175] C.Afifi, A.Khebizi, and K.Halimi. The squash behavior graph routing between the business process event capsules. In *14th ICCCNT*. IEEE, 2023.
- [176] C.Afifi, A.Khebizi, and K.Halimi. Extracting and exploiting the behavior business process graph through transition-centric event-log data. In *2023 ICAIA*, pages 1–6. IEEE, 2023.
- [177] Jie LUO, Chengwan HE, and Hongwei LUO. Brsyn-caps: Chinese text classification using capsule network based on bert and dependency syntax. *IEICE TIS*, 107(2):212–219, 2024.
- [178] van Dongen. Dataset bpi challenge 2019. 4tu. centre for research data. <https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1>, 2019.

- [179] Timothy O Hodson. Root mean square error (rmse) or mean absolute error (mae): When to use them or not. *Geoscientific Model Development Discussions*, 2022: 1–10, 2022.
- [180] Akinori Mitani, Abigail Huang, Subhashini Venugopalan, Greg S Corrado, Lily Peng, Dale R Webster, Naama Hammel, Yun Liu, and Avinash V Varadarajan. Detection of anaemia from retinal fundus images via deep learning. *Nature biomedical engineering*, 4(1):18–27, 2020.
- [181] Manuel Camargo, Marlon Dumas, and Oscar González-Rojas. Learning accurate business process simulation models from event logs via automated process discovery and deep learning. In *ICAISE*, pages 55–71. Springer, 2022.
- [182] Kishore Babu Dasari and Nagaraju Devarakonda. Tcp/udp-based exploitation ddos attacks detection using ai classification algorithms with common uncorrelated feature subset selected by pearson, spearman and kendall correlation methods. *Revue d’Intelligence Artificielle*, 36(1):61–71, 2022.
- [183] Muhammad Sajid, Khuram Ali Khan, Jaroslav Frnda, and Atiqe Ur Rahman. A novel multi-attribute decision-making method for supplier selection in the health care industry using cosine similarity measures of single-valued neutrosophic cubic hypersoft sets. *IEEE Access*, 2025.
- [184] Ling Zhu and al. An intelligent boosting and decision-tree-regression-based score prediction (bdtr-sp) method in the reform of tertiary education teaching. *Information*, 14(6), 2023.
- [185] Nimish Ukey, Zhengyi Yang, and al. Survey on exact knn queries over high-dimensional data space. *Sensors*, 23(2):629, 2023.
- [186] Abdulaziz Aldoseri, Khalifa Al-Khalifa, and Abdel Magid Hamouda. Ai-powered innovation in digital transformation: Key pillars and industry impact. 2024.
- [187] CV Suresh Babu and PM Akshara. Revolutionizing conversational ai: unleashing the power of chatgpt-based applications in generative ai and natural language processing. In *Advanced applications of generative AI and natural language processing models*, pages 228–248. IGI Global, 2024.
- [188] Korhonen Marko. Applying generative ai and large language models in business applications. 2023.
- [189] Ali Khebizi, Hassina Seridi-Bouchelaghem, Boualem Benatallah, and Farouk Toumani. A declarative language to support dynamic evolution of web service

- business protocols. *Serv. Oriented Comput. Appl.*, 11(2):163–181, 2017. doi: 10.1007/S11761-016-0204-7. URL <https://doi.org/10.1007/s11761-016-0204-7>.
- [190] Jennifer Clapp. Concentration and crises: exploring the deep roots of vulnerability in the global industrial food system. *The Journal of Peasant Studies*, 50(1):1–25, 2023.
- [191] Ferdoush Saleheen, Md Mamun Habib, and Abdulmenaf Sejdini. Supply chain collaboration during disruption through sales & operations planning: A study in a manufacturing industry. *International Supply Chain Technology Journal*, 10(1), 2024.
- [192] Md Al-Amin, Mohammad Shazed Ali, Abdus Salam, Arif Khan, Ashraf Ali, Ahsan Ullah, Md Nur Alam, and Shamsul Kabir Chowdhury. History of generative artificial intelligence (ai) chatbots: past, present, and future development. *arXiv preprint arXiv:2402.05122*, 2024.
- [193] C.Affif, A.Khebizi, and k.Halimi. Bp-dss3: a business processes decision-support chatbot through the language model gpt-3.5-turbo’. *Int. J. Intelligent Engineering Informatics*, 2025. doi: 10.1504/IJIEI.2025.10071287.
- [194] Patrizio Bellan, Han van der Aa, Mauro Dragoni, Chiara Ghidini, and Simone Paolo Ponzetto. Pet: an annotated dataset for process extraction from natural language text tasks. In *ICBPM*, pages 315–321, 2022.
- [195] Ben Smith. *Beginning JSON*. Apress, 2015.
- [196] Steven Tey. Shooketh, 2023. URL <https://github.com/steven-tey/shooketh?tab=readme-ov-file>.
- [197] Gaël Varoquaux and Olivier Colliot. Evaluating machine learning models and their diagnostic value. *Machine learning for brain disorders*, pages 601–630, 2023.
- [198] Christopher D Manning. Introduction to information retrieval, 2008.
- [199] James T Townsend. Theoretical analysis of an alphabetic confusion matrix. *Perception & Psychophysics*, 9:40–50, 1971.