



**HAL**  
open science

# Optimiser la gestion des e-mails : Une approche basée sur les processus métier

Ralph Bou Nader

► **To cite this version:**

Ralph Bou Nader. Optimiser la gestion des e-mails : Une approche basée sur les processus métier. Computer Science [cs]. Institut Polytechnique de Paris, 2024. English. NNT : 2024IPPAS017 . tel-04796453

**HAL Id: tel-04796453**

**<https://theses.hal.science/tel-04796453v1>**

Submitted on 21 Nov 2024

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

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



INSTITUT  
POLYTECHNIQUE  
DE PARIS

NNT : 2024IPPAS014

Thèse de doctorat



# Enhancing Email Management Efficiency: A Business Process Mining Approach

Thèse de doctorat de l'Institut Polytechnique de Paris  
préparée à Télécom SudParis

École doctorale n°626 Institut Polytechnique de Paris (ED IP Paris)  
Spécialité de doctorat: Informatique

Thèse présentée et soutenue à Palaiseau, le 30/10/2024, par

**Ralph BOU NADER**

Composition du Jury :

M. Mustapha LEBBAH Professeur des Universités, Université Paris-Saclay, France	Rapporteur
Mme. Salima BENBERNOU Professeur des Universités, Université Paris Descartes, France	Rapporteuse
M. Bruno DEFUDE Professeur des Universités, Télécom SudParis, France	Examineur
Mme. Nour FACI Maître de conférences, Université Claude Bernard, France	Examinatrice
M. Walid GAALOUL Professeur des Universités, Télécom SudParis, France	Directeur de thèse
M. Yehia TAHER Maître de conférences, Université de Versailles, France	Co-encadrant de thèse



# Acknowledgments

*I am deeply grateful to God for allowing me to achieve what initially seemed impossible and for giving me the strength to persevere.* The path to developing a thesis is far from easy. Yet, thanks to the advice, help, and inspiration of many people, I have never regretted embarking on this difficult journey.

At the forefront of my gratitude is my supervisor, **Prof. Walid Gaaloul**. His unwavering confidence in my abilities and his meticulous advice played an absolutely central role in shaping the trajectory of this thesis. Beyond the role of simple supervisor, *Professor Gaaloul's mentorship has been truly enlightening.* He taught me to aspire to excellence, to deepen my research, and to approach research with a spirit of discernment and analysis. His patience, dedication, and deep insight not only shaped the outcome of this work but also fundamentally shaped my entire academic identity. I sincerely thank you, **Professor Gaaloul**, for the immeasurable contributions you have made.

I would also like to express my sincere thanks to all the members of the jury. *Your invaluable expertise, rigorous evaluations, and constructive comments have greatly enriched the depth and quality of this work.*

A special and dedicated thank you is reserved to **Dr. Yehia Taher, Dr. Nour Assy, Dr. Marwa Elleuch, Dr. Ikram Garfatta** and **Prof. Boualem Benatallah** for their valuable collaboration. The exchange of wisdom, innovative ideas, and enriching experiences that we collectively shared contributed significantly to the depth and substance of this research.

*My deepest gratitude resonates at the heart of my support system: my family.* To my father, **Abdo Bou Nader**, I owe an immense debt of gratitude for being a pillar of strength and wisdom in my life. His advice and sacrifices paved the path I travel today, and his unwavering confidence in my potential has always inspired me to strive for excellence. I am forever grateful for the love and support he gave me.

To my mother, **Léna el Yahchouchy**, whose love and caring presence define my life, I am infinitely grateful for her constant encouragement and boundless care. Her grace and the lessons she has imparted to me are deeply appreciated and guide me through every challenge I face.

I am so grateful for the understanding and friendship of my sisters, **Alma, Elsa, and Anna-Maria Bou Nader**, who have been my mentors and companions. You are an inspiration to me. Your everlasting confidence in me and your unceasing support have greatly influenced who I am now.

Last but not least, my friends, especially **Dr. Marie-Rita Hojeij and Mr. Nidal Khater**, deserve my sincere gratitude for their unwavering support and companionship. Your

encouragement, empathy, and shared moments of laughter inject a sense of joy into this difficult journey. *Your unwavering presence constantly motivates me to overcome obstacles and continue striving for excellence.*

Each of you has uniquely contributed to this achievement, and I am deeply grateful for the merit you bring to my life and work.



# Contents

<b>Table of abbreviations and acronyms</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Research Context . . . . .	1
1.2 Research Challenges . . . . .	6
1.3 Research Questions . . . . .	7
1.4 Thesis Objectives and Principles . . . . .	10
1.5 Thesis Contributions . . . . .	11
1.6 Thesis Outline . . . . .	14
<b>2 State Of The Art</b>	<b>17</b>
2.1 Introduction . . . . .	17
2.2 Conformance Checking in business process Mining . . . . .	17
2.3 Examining Email Recommendation and Process Prediction . . . . .	22
2.4 Conclusion . . . . .	36
<b>3 Multi-Perspective Conformance Checking For Email-driven Processes</b>	<b>39</b>
3.1 Introduction . . . . .	39
3.2 The Proposed Approach Overview . . . . .	40
3.3 Proof of Concept . . . . .	53
3.4 Experiments and Validation . . . . .	58
3.5 Conclusion . . . . .	61
<b>4 Predictive Process Approach for E-mail Response Recommendations</b>	<b>63</b>
4.1 Introduction . . . . .	63
4.2 The Proposed Approach Overview . . . . .	64
4.3 Experiments and Validation . . . . .	81
4.4 Conclusion . . . . .	86
<b>5 Conclusion &amp; Perspectives</b>	<b>89</b>
5.1 Contributions . . . . .	89

---

5.2 Perspectives . . . . .	91
<b>A A Novel Approach For Unsupervised Anomaly Detection In Time Series</b>	<b>93</b>
A.1 Introduction . . . . .	94
A.2 Related Work . . . . .	95
A.3 The Proposed Approach . . . . .	99
A.4 Experiments and Validation . . . . .	105
A.5 Conclusion . . . . .	110
<b>B Résumé Etendu</b>	<b>113</b>
B.1 Contexte et problématique de la recherche . . . . .	113
B.2 Objectifs et Contributions de la Thèse . . . . .	114
<b>Bibliographie</b>	<b>125</b>



# List of Figures

1.1	The Business Process Management Life-cycle . . . . .	2
1.2	Emails retrieved from Enron data-set . . . . .	5
1.3	Email Request for Report Review . . . . .	6
1.4	Overview of the Proposed Framework Components . . . . .	13
2.1	Example of Interview Setting Emails . . . . .	33
2.2	Email Correspondence: Message from David . . . . .	34
2.3	Recommended Email Response Candidate: Leveraging Our Approach . . . . .	35
2.4	Generated Email Response Candidate: GPT-4 Model Application . . . . .	35
3.1	Approach overview . . . . .	41
3.2	Emails retrieved from Enron data-set . . . . .	42
3.3	A partial view of the Email Process model . . . . .	48
3.4	Example of an Event Log Extract . . . . .	50
3.5	Real email retrieved from Enron dataset for planning trading positions . . . . .	50
3.6	Urgent Email: Alice’s Notification Regarding Deal Priority . . . . .	54
3.7	Bob’s First Email Response . . . . .	54
3.8	Bob’s Second Email Response . . . . .	55
3.9	A partial view of the <i>Email Process Model</i> Illustrating Gas Deal Management Processes . . . . .	56
4.1	The proposed approach overview . . . . .	65
4.2	Email main body . . . . .	66
4.3	Example of an Event Log Extract . . . . .	67
4.4	Sequence of events example extracted from two emails in the same thread . . . . .	68
4.5	David’s Email Correspondence with Julie . . . . .	71
4.6	Predicted sub-sequence of events . . . . .	72
4.7	Sentence Recommendation for BP Context . . . . .	74
4.8	Network graph of selected employees based on similarity in writing styles . . . . .	78
4.9	Proposed Email Response Template Tailored for Julie’s Communication Style in Reply to David . . . . .	81

4.10 Comparing Metrics for the Top 30 Frequent Event Log Classes in Experiments 1 and 2 . . . . .	83
A.1 TBD diagram components . . . . .	100
A.2 AVL tree constructed out of the data points in Table A.1 . . . . .	102

# List of Tables

2.1	Comparison of Conformance Checking Techniques . . . . .	21
2.2	Process Prediction Techniques . . . . .	30
2.3	Email Recommendation Techniques . . . . .	31
3.1	Overview of the results . . . . .	59
3.2	Precision and Recall Metrics for Non-Conformance Types Detected in Enron Event Logs . . . . .	59
3.3	Impact of API-Aided Revisions on Email Draft Quality and Efficiency . . . . .	61
4.1	Comparing Evaluation Results: Our Approach vs. Fine-tuned GPT-3 vs. Fine-tuned GPT-4 Models . . . . .	84
4.2	Comparison Between Automated and Manual Email Approaches . . . . .	86
A.1	Subset of the first q buffered data from IoT temperature monitor device . . . . .	101
A.2	Example showing the result of TBD after processing 4 incoming IoT data . . . . .	104
A.3	Comparative study of TBD with existing approaches . . . . .	106
A.4	Calculated anomaly score as a heat map against both IoT incoming data and the timestamps for six approaches . . . . .	107
A.5	Accuracy of TBD against existing approaches computed on instances where TBD detects a point anomaly followed by a collective anomaly. Instances detected as anomalous (point or collective) are considered as TP . . . . .	108
A.6	Performance metrics computed on instances where TBD detects a point anomaly followed by a collective anomaly. Point anomaly is considered as TP, collective anomaly is considered as TN . . . . .	108
A.7	Comparative table showing the efficiency of TBD on top of the AE model . . . . .	109



# Table of notations

<b>AE</b>	Auto-Encoders
<b>API</b>	Application Programming Interface
<b>ATS</b>	Automated Transition System
<b>AVL</b>	Adelson-Velsky and Landis
<b>AWS</b>	Amazon Web Services
<b><math>\mathcal{BD}</math></b>	<i>The set of business data</i>
<b>BDR</b>	Buffered Data Retrieval
<b>BERT</b>	Bidirectional Encoder Representations from Transformers
<b>BF</b>	Balancing Factor
<b>BP</b>	Business Process
<b>BPM</b>	Business Process Management
<b>BPMS</b>	Business Process Management Systems
<b>BPTT</b>	Back-propagation through time
<b>BST</b>	Binary Search Tree
<b>CA</b>	Collective Anomaly
<b>COF</b>	Connectivity-based Outlier Factor
<b>DBSCAN</b>	Density-Based Spatial Clustering of Applications with Noise
<b>DL</b>	Deep Learning
<b>DNNs</b>	Deep Feed-forward Neural Networks
<b>EPM</b>	Email process model
<b>GANs</b>	Generative Adversarial Networks
<b>GELU</b>	Gaussian Error Linear Unit
<b>GPR</b>	Gaussian Process Regression
<b>GPT</b>	Generative Pre-trained Transformer
<b>HDBSCAN</b>	Hierarchical Density-Based Spatial Clustering of Applications with Noise

<b>HPStream</b>	High-dimensional Projected Stream
<b>HTTP</b>	Hypertext Transfer Protocol
<b>INFLO</b>	Improving Influenced Outlierness
<b>IoT</b>	Internet of Things
<b>IS</b>	Information Systems
<b>JSON</b>	JavaScript Object Notation
<b>KNN</b>	k-nearest neighbors
<b>LDBSCAN</b>	Local Density-based Spatial Clustering of Applications with Noise
<b>LOF</b>	Local Outlier Factor
<b>LOP</b>	Local Outlier Probabilities
<b>LSTM</b>	Long Short-Term Memory
<b>MP-Declare</b>	Multi-Perspective Declare
<b>NER</b>	Named Entity Recognition
<b>NLP</b>	Natural Language Processing
<b>OCEL</b>	Object-centric event logs
<b>OPTICS</b>	Ordering Points to Identify the Clustering Structure
<b>PA</b>	Point Anomaly
<b>PAIS</b>	Process-Aware Information Systems
<b>PCA</b>	Probable Collective Anomaly
<b>PR</b>	PageRank
<b>REST</b>	Representational State Transfer
<b>RNN</b>	Recurrent Neural Network
<b>RPF</b>	Repetitive Processing Function
<b>SD</b>	Standard Deviation
<b>SVM</b>	Support Vector Machines
<b>TBD</b>	Track Before Detect
<b>TF-IDF</b>	Term Frequency Inverse Document Frequency
<b>UMAP</b>	Uniform Manifold Approximation and Projection
<b>Yake</b>	Yet Another Keyword Extractor

# Introduction

---

## Contents

---

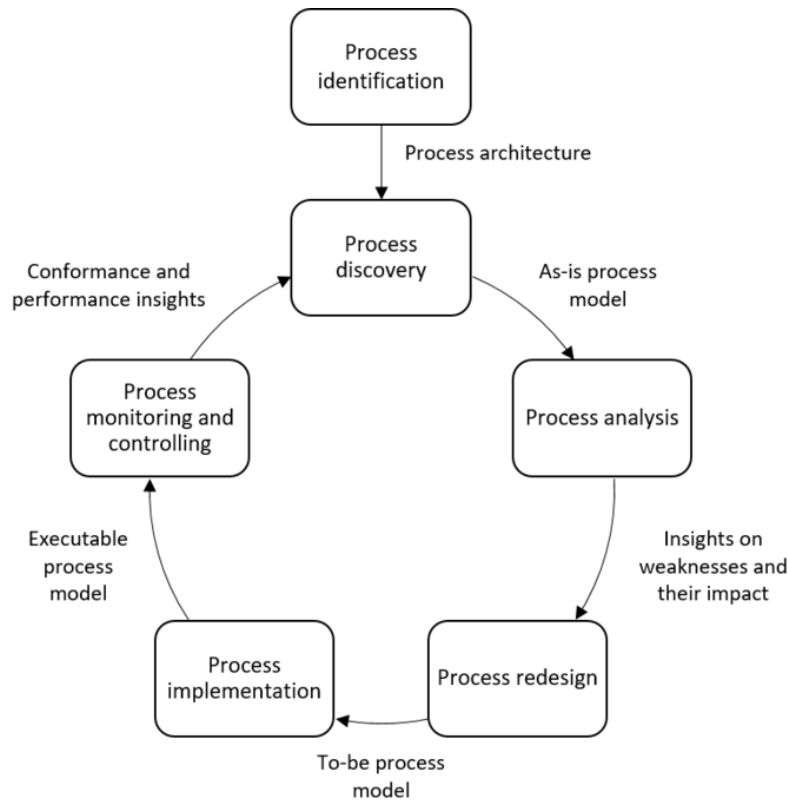
<b>1.1</b>	<b>Research Context</b>	<b>1</b>
1.1.1	Motivating Example	4
<b>1.2</b>	<b>Research Challenges</b>	<b>6</b>
<b>1.3</b>	<b>Research Questions</b>	<b>7</b>
1.3.1	How effectively do current conformance checking techniques perform when applied to email-driven processes?	8
1.3.2	How can predictive techniques be utilized to recommend specific process-oriented emails?	9
<b>1.4</b>	<b>Thesis Objectives and Principles</b>	<b>10</b>
<b>1.5</b>	<b>Thesis Contributions</b>	<b>11</b>
<b>1.6</b>	<b>Thesis Outline</b>	<b>14</b>

---

## 1.1 Research Context

Business processes [101] refer to the series of interconnected activities, tasks, and steps that are performed within an organization to achieve a specific goal or outcome. Their purpose is to optimize productivity, improve efficiency, and simplify processes. Businesses in a variety of sectors and roles, from manufacturing and supply chain management to customer service and finance, might have very varied business processes. In order to provide value to customers and stakeholders, they frequently entail the coordination of people, resources, and technology.

The field of Business Process Management (BPM) [95, 108, 3] has emerged to ensure continuous improvement in business processes. In order to improve organizational performance, business processes must be managed and optimized throughout the BPM life-cycle ???. This life-cycle normally consists of six important phases. The first phase is process discovery, where the current state of a specific business process is modeled (as-is model) to achieve a common understanding among stakeholders. Information is gathered through document analysis, interviews, and observation, while research focuses on the suitability of modeling languages and the impact of different visual elements. The next phase, process analysis, uses the as-is model to identify issues and their root causes, employing qualitative and quantitative techniques.



**Figure 1.1:** The Business Process Management Life-cycle

This is followed by process redesign, where a future state (to-be model) is developed using various methods. During process implementation, the to-be model is operationalized through change management and IT development, supported by process-aware information systems. Finally, process monitoring and controlling involve tracking the process execution against the to-be model, using performance dashboards and process mining, providing insights for continuous improvement and restarting the lifecycle if necessary.

Business Process (BP) mining, as introduced by Aalst et al. [2], is an integral part of the BPM lifecycle that analyzes and improves business processes by examining event logs and transaction data. It involves extracting insights from these data sources to understand how processes are actually executed in practice, as opposed to how they are designed to be executed. Process mining techniques can help identify bottlenecks, inefficiencies, and deviations from the intended process flow. By analyzing the sequence of activities and dependencies within a process, organizations can uncover opportunities for optimization and automation.

Within this framework, two pivotal components are conformance checking [24] and process prediction [57]. In the context of BP mining, conformance checking is an important component since it evaluates how well the process is actually carried out in comparison to the model that is expected. Through the identification of deviations or inconsistencies between the actual and projected behavior, it offers important insights into how well the current business processes are working. Organizations can make sure that their procedures follow established models and



standards by using conformance checking. By pointing out places where the actual execution deviates from the planned behavior, it enables quick corrective action and promotes continual development.

Process prediction is another essential component of BP mining that works in tandem with conformance checking. The objective is to predict the future behavior of a specific process by utilizing machine learning algorithms and sophisticated analytical approaches on the basis of past data. Predicting how tasks will be performed in a business process and how they will depend on one another is the prediction. Anticipating any deviations from the intended flow is the ultimate goal since it allows firms to optimize their operations and deal with concerns proactively.

When combined, process prediction and conformance checking enable businesses to improve the efficiency, flexibility, and transparency of their business processes. By harnessing insights from historical data, these techniques enable businesses to make informed decisions, mitigate risks, and stay responsive in today's dynamic and competitive business environment. The future of BPM practices will be significantly shaped by the integration of these components as the area of BP mining continues to develop.

The growing influence of BP mining has opened up diverse opportunities across various domains. In this dynamic landscape, a promising avenue emerges for seamlessly integrating process prediction and conformance checking techniques into the realm of email communication. Specifically, this integration targets email-driven processes, which are generally BP fragments executed using emailing systems rather than traditional BPM systems. These email-driven processes are not traditionally supported by Process-Aware Information Systems (PAIS) [35], which are systems designed to manage and execute business processes. PAIS enable BPM by providing the necessary tools and infrastructure to automate and control these processes based on defined models. Unlike structured processes managed by PAIS, email-driven processes lack formalized models and execution control, making it difficult to apply conventional BPM techniques.

To understand why conventional process models are insufficient for handling email-driven processes and why a more appropriate model is necessary, it is important to distinguish these processes from traditional business processes. Email-driven processes are characterized by a series of activities that represent the execution of tasks. Traditional business processes typically have a clear and explicit temporal control flow, where activities are executed in a precise order. In contrast, email-driven processes do not follow this structure. Instead, the sequence of activities in email processes is determined by both their appearance within individual emails and the chronological order of email threads.

In traditional business processes, the temporal order of activities is explicit and easily identifiable, facilitating the modeling and monitoring of process flows. However, in email-driven processes, the control flow is not temporal but is based on the order of activities as they appear in emails and threads. This order reflects how users perceive and execute activities within the context of email communications. Each email and thread can be seen as a projection of the business process instance, where the sequence of activities is captured by

the structure of the email content and the thread’s conversation.

Moreover, email-driven processes often consist of fragments of larger business processes. These fragments are grouped to achieve specific business objectives. In line with the traditional definition of a business process as a set of activities aiming to achieve a business goal, email-driven processes can similarly be viewed as a collection of activities working towards a business goal or executing a part of a business process. However, identifying the control flow in emails is inherently more complex due to the lack of explicit temporal ordering, as previously explained.

To address these challenges, our research develops methodologies tailored for the unique characteristics of email-driven processes, ensuring they can benefit from the same level of process management and optimization as those supported by PAIS.

### 1.1.1 Motivating Example

This section presents a motivating example to illustrate the importance of conformance checking and prediction in email-driven business processes. Consider the recruitment process, where all communications with candidates occur via email—whether sending/forwarding resumes, planning interviews, or informing candidates about hiring decisions. In other contexts, emails may be used to handle specific events during business processes. Figure 1.2 includes an example of real emails retrieved from the Enron database<sup>1</sup>. It shows a set of interactions between employees outside a gas trading system for handling a flow gas event (i.e., a gas volume that exceeds a certain threshold).

The emails are related to the same gas meter (Meter 5192) and the same trading deal (deal 454057). They belong to two conversations (with subjects ‘Flow w/no nom’ and ‘Dec 00’) and are sorted in ascending order according to their timestamps. The emails report how employees act when a meter detects a gas flowing. The first and third emails show how an employee notifies his manager once this event occurs to request the execution of some activities (e.g., extend the associated deal or create a new one in email *email*<sub>3</sub>). The second and fourth emails report the activity carried out by the manager to cover the flowing event of the gas meter (roll or extend the associated deal as indicated in *email*<sub>2</sub> and *email*<sub>4</sub> respectively). Figure 1.2 illustrates a case where a trading BP part related to managing gas deals is supported by emails inside Enron company. It also shows an example of an email (i.e., *email*<sub>3</sub>) summarizing employee expertise when handling some events/exceptions (through the requested activities).

Conformance checking, in the context of email-driven processes, offers significant advantages by ensuring the accuracy and completeness of exchanged text. To illustrate, let’s examine into a scenario involving an employee named Bob, tasked with sending emails about upcoming workshops to colleagues. There was an instance where Bob unintentionally omitted crucial information from an email before dispatching it. In such situations, conformance checking plays a pivotal role in maintaining the quality and integrity of the communication

---

<sup>1</sup><http://www.cs.cmu.edu/~enron>

<p>Message-ID: &lt;1552589.1075853972210.JavaMail.evans@thyme&gt; <b>email<sub>1</sub></b>  Date: Fri, 10 Nov 2000 06:17:00 -0800 (PST)  From: aimee.lannou@enron.com  To: daren.farmer@enron.com  cc:  Subject: Flow w/ no nom</p> <p>Meter 1601 last deal 412219 for 10/00 flowed 11/9</p> <p>Meter 5192 last deal 454057 for 10/00. flowed 11/3-4</p>	<p>Message-ID: &lt;3140966.1075854206364.JavaMail.evans@thyme&gt; <b>email<sub>3</sub></b>  Date: Tue, 9 Jan 2001 04:43:00 -0800 (PST)  From: aimee.lannou@enron.com  To: daren.farmer@enron.com  cc: edward.terry@enron.com  Subject: Dec 00</p> <p>Daren - meter 5192 flowed 8 dth on 12/19, 33 dth on 12/20 and 2 dth on 12/29. The last deal for this meter was 454057 in Nov 00. Could you please extend this deal for these 3 days or create a new one? Please let me know.  AL</p>
<p>Message-ID: &lt;29717536.1075854150388.JavaMail.evans@thyme&gt; <b>email<sub>2</sub></b>  Date: Wed, 15 Nov 2000 02:24:00 -0800 (PST)  From: daren.farmer@enron.com  To: aimee.lannou@enron.com  cc:  Subject: Re: Flow w/ no nom</p> <p>Rolled deal 454057 to cover flow at mtr 5192.  d</p> <p>Aimee Lannou 11/10/2000 02:17 PM  To: Daren J Farmer/HOU/ECT@ECT  cc:  Subject: Flow w/ no nom</p> <p>Meter 1601 last deal 412219 for 10/00 flowed 11/9  Meter 5192 last deal 454057 for 10/00. flowed 11/3-4</p>	<p>Message-ID: &lt;26296505.1075854337364.JavaMail.evans@thyme&gt; <b>email<sub>4</sub></b>  Date: Tue, 9 Jan 2001 06:48:00 -0800 (PST)  From: daren.farmer@enron.com  To: aimee.lannou@enron.com  cc:  Subject: Re: Dec 00</p> <p>I extended 454057 for the month of December.  D  Aimee Lannou 01/09/2001 12:43 PM  To: Daren J Farmer/HOU/ECT@ECT  cc: Edward Terry/HOU/ECT@ECT  Subject: Dec 00</p> <p>Daren - meter 5192 flowed 8 dth on 12/19, 33 dth on 12/20 and 2 dth on 12/29. The last deal for this meter was 454057 in Nov 00. Could you please extend this deal for these 3 days or create a new one? Please let me know.  AL</p>

Figure 1.2: Emails retrieved from Enron data-set

process.

Before an email is sent, the conformance checking mechanism evaluates its content against the prescribed process model. This assessment aims to detect any omissions or discrepancies, such as the one in Bob's case. In this scenario, the conformance checking system identifies the missing information and promptly brings it to Bob's attention. This timely intervention offers Bob an opportunity to rectify the mistake before the email reaches his colleagues.

Through this proactive conformance checking practice, all of Bob's colleagues ultimately receive comprehensive and accurate information regarding the upcoming workshops. Consequently, they can plan their schedules efficiently and actively participate in the events, contributing to a smoother and more effective workflow.

On the other hand, integrating process prediction into email communication offers even greater benefits. Predictive capabilities extend beyond merely identifying future BP activities through email correspondence. This integration envisions a system that suggests not only the appropriate emails to send but also enhances the textual content of these communications.

For instance, consider another employee, Alice, who is responsible for managing customer support emails. By analyzing historical email data within the BP framework, predictive models can discern patterns and provide tailored recommendations. Imagine a scenario where Alice receives a customer email reporting an issue with a product. The predictive model, having analyzed similar past interactions, can suggest a draft response for Alice. This draft email might include activities related to an apology, a request for additional information, and potential troubleshooting steps.

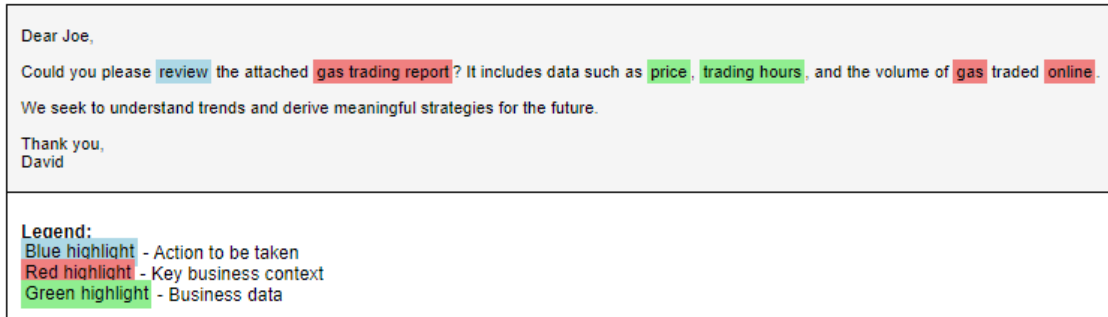
Furthermore, the predictive system can recommend follow-up actions based on the historical resolution of similar issues. For example, if previous cases show that offering a discount or

expedited shipping resolved similar complaints effectively, the system can suggest including such offers in Alice’s response. This ensures that Alice’s communication is not only prompt but also aligned with successful resolution strategies.

Thus, the combination of conformance checking and predictive analytics can significantly improve the management of email-driven business processes. This approach recommends suitable responses to incoming emails, saving time and ensuring timely communication, while also ensuring the accuracy and completeness of information in manually composed emails. These techniques enhance efficiency in information exchange, minimize errors, and facilitate agile responses to evolving process requirements. However, challenges arise when attempting to apply these methods to business processes conducted through unconventional systems, particularly email-driven processes, which form the primary focus of our research.

## 1.2 Research Challenges

A significant obstacle arises from the distinctive structure of event logs derived from email systems, as pointed out by Elleuch et al. [41]. In contrast to classical information systems, events in email systems often come with additional attributes. This unique structure presents a challenge in the application of conventional process prediction and conformance checking methods, necessitating customized solutions to address the specific characteristics of email-driven processes.



**Figure 1.3:** Email Request for Report Review

For instance, the goal of an event occurring in an email is identified not only by the explicitly mentioned activity name but also by the interlocutor’s speech act. As an example, in the email depicted in Figure 3.5, where David asks Joe, “*Could you please review the attached gas trading report?*” David’s speech act is a request. He is specifically asking Joe to review the report. Thus, the activity is the *review of the attached gas trading report*.

Furthermore, participants in business processes, such as David, often provide additional context by incorporating pertinent business data (**BD**) into their discussions. This data can be categorized into two groups. The first group comprises data utilized and generated during the activity, such as ‘*price*’ and ‘*trading hours*’. The second type encompasses expressions or terms that provide additional context to the business scenario, such as ‘*gas*’ to specify the

type of energy being traded and 'online' to indicate the method of the trading operation.

The structural characteristics of event logs from email systems are often overlooked in traditional approaches, but they are crucial in email-based BP mining. Introducing additional attributes to the event log can complicate matters, leading to complexities such as:

- **Complexity of Analysis:** Analyzing these new attributes demands advanced algorithms and specialized tools, especially when seeking to understand the underlying connections between these attributes and their effects on process prediction and conformance checking.
- **Integration with Existing Business Process Approach and Emailing System:** Integrating these additional attributes into the existing business process approach and emailing system might pose its own challenges. Such integration could necessitate substantial customization or even a complete overhaul of the current systems, resulting in both time and cost implications.

The context of this thesis revolves around addressing the challenges associated with mining business processes from *email-driven processes*. It explores the unique characteristics of email event logs and emphasizes how important they are to email-driven processes. The aim of the thesis is to facilitate the efficient implementation of process prediction and conformance checking in the context of process-oriented emails by comprehending and tackling these intricacies.

Moving forward, we will discuss the research problems related to this context in Section 1.3. Subsequently, we will outline our thesis objectives, principles, and contributions in Sections 1.4 and 1.5. Finally, in Sections 1.6, we will present the structure of the thesis, providing a clear roadmap for the logical flow of ideas and findings throughout this report.

## 1.3 Research Questions

In our pursuit of applying process prediction within the context of *email-driven processes*, we encountered a multitude of challenges that pushed the boundaries of our comprehension. However, the complexity of the situation escalated even further when we opted to implement conformance checking as a part of our approach. These challenges can essentially be distilled and categorized around two pivotal research questions that laid the foundation for our study:

Our first research question, (*Q1*) "***How effectively do current conformance checking techniques perform when applied to email-driven processes?***", is addressed in Section 1.3.1. In this section, we scrutinize the challenges of applying conformance checking techniques within the realm of emails, exploring how they fit and function in this specific context.

Our second research question, (*Q2*) "*How can predictive techniques be utilized to recommend specific process-oriented emails?*", is explored in Section 1.3.2. Here, we discuss the constraints and limitations encountered when incorporating existing process prediction techniques to recommend process-oriented emails.

### 1.3.1 How effectively do current conformance checking techniques perform when applied to email-driven processes?

Several methods have been proposed for conformance checking in business processes [37, 24, 110, 48], such as alignment-based methods [16, 17] that utilize event logs and process models to find the optimal alignment between expected and actual behavior. While these methods have shown promising results, they struggle to identify discrepancies that go beyond the perspective of the sequential flow of activities. To overcome this limitation, some approaches have addressed conformance checking for multi-perspective processes, considering attributes like time, resources, and other data attributes [33, 49, 115, 76]. An example of this innovation is a technique that employs fuzzy logic to evaluate compliance from both structural and temporal perspectives [114].

However, these existing methods fall short in detecting event discrepancies from a business context perspective. They typically assume that the business context of an event is predefined based on the process it belongs to [37], which means they don't account for the possibility that an event within the same process instance might deviate and correspond to a different business context. This limitation is particularly evident in scenarios such as email conversations, where the topic can frequently shift. For example, an email thread might start discussing a project update (one business context) and then shift to budget discussions (another business context) within the same conversation. Current conformance checking methods would struggle to identify such shifts and correctly attribute the events to their respective business contexts, leading to potential inaccuracies in detecting discrepancies.

Additionally, many current approaches [28, 27, 75] mainly focus on calculating conformity metrics, assuming that event attribute values are either categorical or numerical. They often overlook the complexities in cases like email events, where attributes may include words that provide essential context. Understanding these attributes requires a closer examination of the context in which they appear, ensuring alignment with ongoing discussions within email conversations.

Given these limitations, there is a clear need for a more comprehensive and context-aware approach to effectively detect event discrepancies. Therefore, we must develop an approach for multi-perspective conformance checking in email-driven processes. This new direction aims to bridge the existing gaps and provide a more robust approach to evaluating conformance in complex business environments, such as email communication. To achieve this goal, we need to address the following sub-questions:

- Q1-1: How can we design a conformance checking method that takes into account both

the structural and contextual perspectives of the email events?

- Q1-2: What are the common patterns of discrepancies that can occur in the business context of email events, and how can they be accurately detected?
- Q1-3: How can we handle non-categorical or non-numerical attribute values in conformance checking methods, particularly those related to email events?
- Q1-4: How can we validate and measure the accuracy, reliability, and performance of the newly proposed context-aware conformance checking method?

### 1.3.2 How can predictive techniques be utilized to recommend specific process-oriented emails?

In the context of emails, the term "*prediction*" in general refers to the application of specific algorithms aimed at suggesting various fields for email responses [102, 88, 50, 72]. These fields include the sender's identity, recipients, attached files, subject lines, and even the email main body itself.

The focus on recommending the main body of email messages has evolved into a highly intriguing area, prompting numerous advancements in the field. Various research studies, such as those conducted by Yang et al. [111] and Zhang et al. [113], have explored this domain. The approach outlined in [111] was grounded in collaborative filtering, a technique that exploits user behaviors and preferences to offer suggestions. By analyzing users' historical email interactions and patterns, the algorithm identifies common phrases and structures that can be harnessed to compose effective email responses. This process assists users in crafting relevant and contextually fitting replies, ultimately enhancing their efficiency in email communication.

The work by Zhang et al. [113] took this further by adopting a deep learning approach. Their model utilized a combination of user and email features, including user interaction history, email metadata, and contextual information, to develop a predictive framework. This sophisticated framework was capable of comprehending the nuanced requirements of individual recipients, enabling the production of a set of recommended email bodies tailored specifically to each person.

Nevertheless, the primary focus of these existing works was to enhance email management without giving much thought to the context of business processes. Among those that combined email management with the concept of business processes, they were mostly limited to the stage of BP discovery from email logs [66, 62, 26]. In certain cases, the focus shifted towards categorizing incoming emails into different business process activities [88].

It's important to highlight that process prediction in the context of emails should encompass more than just identifying activities; it should extend to suggesting specific emails that assist BP actors in executing their activities. Acknowledging the limitations inherent in the prevailing methodologies, we must develop a process-activity-aware email response recommendation system that goes beyond the scope of traditional email management approaches.

Essentially, our system recommends email response templates based on predicted BP knowledge. This knowledge pertains to the set of activities to be conveyed in the email responses, the intention behind expressing them in the email (i.e., speech act), and the manipulated business data. Our approach strives to integrate email management with business processes by utilizing advanced techniques in natural language processing and machine learning. To realize this, we need to address several sub-questions:

- Q2-1: How can we effectively leverage the event log from previously exchanged emails to predict future BP knowledge that will be expressed in email responses?
- Q2-2: What are the most suitable machine learning algorithms or predictive models that can be employed to forecast this future BP knowledge?
- Q2-3: What types of event attributes should be considered when predicting process-specific email responses?
- Q2-4: How can we personalize the recommended email responses not only based on the process activity but also on the preferences and communication styles of individual participants?
- Q2-5: How can we validate the effectiveness and performance of the process-activity-aware email response recommendation system to ensure it provides meaningful and valuable email responses in response to received emails?

## 1.4 Thesis Objectives and Principles

Given the research problems described earlier, the primary objectives of this thesis can be summarized as follows:

- **Objective 1:** Implement Multi-Perspective Conformance Checking For Email-driven Processes.
  - This objective involves proposing a process model based on sequential and contextual constraints specified by a data analyst/expert.
  - Additionally, it entails implementing an algorithm to compare process instances with the process model, thereby identifying fulfilling and violating events based on sequential and contextual constraints.
- **Objective 2:** Develop a Process-Activity-Aware Email Response Recommendation System.
  - This involves leveraging a structured event log to predict future BP knowledge. It includes the prediction of the set of activities to be expressed in the email response, the intention of expressing them in the email, as well as the manipulated business data.



- Furthermore, the system aims to recommend email response body templates based on the predicted activities and historical textual contents related to the predicted BP knowledge.

To achieve these objectives, we consider the following principles:

- **Principle 1: Context Sensitivity:** In both the process-activity-aware email response recommendation system and the multi-perspective conformance checking approach, understanding the context of the emails within the business process is crucial. This principle highlights how crucial it is to evaluate and comprehend the business context of email exchanges in order to make sure that the suggested responses and conformance checks are pertinent and acceptable.
- **Principle 2: Interdisciplinarity:** This idea emphasizes how important it is to build bridges between several fields, including business management, machine learning, natural language processing, and process mining. Our study strives to provide a more comprehensive and nuanced approach to email-based business operations by utilizing ideas and approaches from these many domains.
- **Principle 3: Consistency:** Coherent and unified systems require a consistent approach to both email response prediction and conformance checking. In order to ensure consistency in results and interpretations, this concept calls for the creation of standardized techniques and algorithms that can be applied with confidence across various contexts and datasets.
- **Principle 4: Automation:** Creating models and systems that require minimal human intervention is the aim of this approach, which focuses on automating the processes of email recommendation and conformance checking. Because automation increases efficiency and reduces the chance of human error, it improves accuracy and smoothness.
- **Principle 5: Accessibility and Integration:** Solutions must to be easy to use and adaptable enough to fit into a range of corporate settings. This approach focuses on designing tools that are easily integrated, easily available, and robust.

It is noteworthy that the proposed work in this thesis needs validation through a public dataset of emails to enable comparison with related studies (i) and evaluation through different experiments on real datasets (ii). Furthermore, the implementation, experiments, and results should be detailed.

## 1.5 Thesis Contributions

To effectively achieve the stated objectives and overcome the outlined research challenges, we have devised a range of algorithms for analyzing email-driven processes, forming a robust and

comprehensive framework. Figure 1.4 offers a comprehensive visual representation of the proposed framework components. This sophisticated framework functions with dual capabilities. Firstly, the framework can conduct a thorough evaluation of written emails, ensuring adherence to predefined conformance standards before the actual sending process. Secondly, it can analyze incoming emails and intelligently recommend appropriate response templates. This not only enhances efficiency but also streamlines communication processes. By incorporating these advanced features, our approach optimizes email response generation and establishes a crucial layer of quality control to maintain communication integrity. The significance of our work lies in these multifaceted contributions, combining innovation, efficiency, and reliability to address the complex landscape of email communication in a nuanced manner.

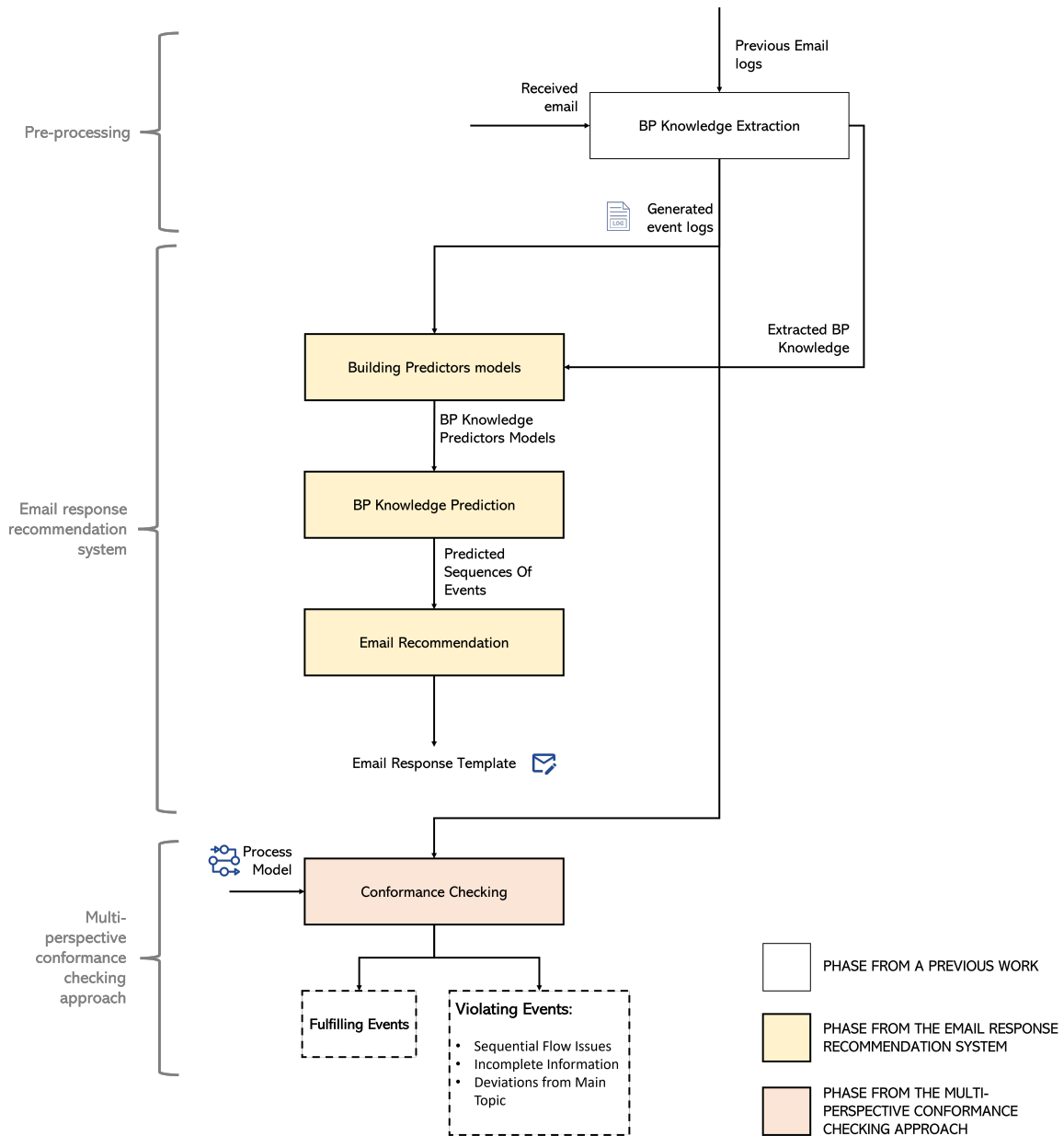
- **Multi-Perspective Conformance Checking Approach (Q1):** We present an approach for multi-perspective conformance checking in email-driven processes (**Principle 2**). The approach, as illustrated in Figure 1.4, takes as input a structured event log generated from an email log (**Principle 3**), along with a *process model*.

The latter is specified by a data analyst/expert and is built upon a set of constraints that pertain to both the *sequential* and *email contextual* aspects of the business process, highlighting the multi-perspective nature of our approach (**Q1-1**). Such constraints are guided by *constraint classes* that we have gleaned through the analysis of email datasets and that serve as templates for the analysts/experts to define their specific constraints (**Q1-2**). We further note that in this work, we propose an *Email Process Model* that we use for the conformance checking of events within the same email or thread. Consequently, the conformance checking algorithm proposed in our approach involves comparing the execution of a process instance (i.e., an event log instance) with the process model (**Q1-3**). This comparison helps identify two sets of events termed *fulfilling* and *violating* events. The former refers to events that occur within a process instance and align with the expected behaviors, while the latter refers to events that violate *sequential* and/or *email contextual* constraints. In addition, the conformity-checking algorithm is able to identify missing topics, highlighting deviations or inconsistencies in the context.

*Note: In the upcoming chapters, we will delve into how we can resolve all the principles by outlining specific methodologies and techniques that ensure conformance across various perspectives within email-driven processes, as well as within the context of our email response recommendation system.*

- **Email Response Recommendation System (Q2):** We have introduced a process-activity-aware email response recommendation system. Our main goal is to provide recommendations for appropriate email response templates based on incoming emails (**Principle 4**).

Our methodology consists of four distinct phases, depicted in Figure 1.4, each contributing to our primary objective. These phases are linked by arrows, some in blue indicating their role in proposing email response templates for received emails, and others in black indicating the preprocessing steps needed to generate the necessary models and inputs for suggesting the appropriate email response template.



**Figure 1.4:** Overview of the Proposed Framework Components

It's important to emphasize that the BP Knowledge extraction phase is based on previous work [42], forming the basis of this work and serving two purposes: preprocessing and email recommendation. In the preprocessing stage, upon receiving an email log—a chronological record of email communications containing sender and recipient addresses, timestamps, subject lines, and sometimes message content—the initial step involves creating an event log from previously exchanged emails. This event log, established during the BP Knowledge extraction phase, lays the groundwork for the Building Predictors models phase, where we develop two BP prediction models designed to anticipate the BP knowledge to be integrated into the email response (**Q2-1**). The first model takes

the sub-sequence of events appearing in a received email as input and predicts the possible next combinations of events that may appear in the email response. The events within the predicted sub-sequences can occur in multiple orders. Conversely, the second prediction model is used to forecast the order of the events following it in the same email. We utilized the Long Short-Term Memory (LSTM) architecture to train both models, known for its exceptional performance in handling sequential data tasks [99] (**Q2-2**).

However, in the email recommendation stage, the BP Knowledge extraction phase is invoked again to identify the instance of the received email. Subsequently, in the BP Knowledge Prediction phase, we employ the prediction models developed during the Building Predictors models phase to predict the relevant BP knowledge to be included in our email response (**Q2-3**).

Lastly, in the Email Recommendation phase, our approach suggests an email response template by analyzing the textual content related to the BP knowledge of the email response (**Principle 1**). The suggested email responses are recommended based on carefully defined criteria, including alignment with the business context, incorporation of predicted BP knowledge, and consistency with the author’s writing style. These criteria ensure that the responses are relevant, include specialized information, and maintain stylistic consistency, ultimately aiming to provide a more personalized and coherent communication experience for the recipient (**Q2-4**).

- **Streamlining Communication with Representational State Transfer (REST) Application Programming Interfaces (API) for Prediction and Compliance Methods:** In alignment with **Questions 1-5 and 2-4**, we have introduced a solution that utilizes RESTful APIs to establish a unified framework for email management in the context of BPM (**Principle 2 and 5**).

Finally, we validate all the introduced algorithmic solutions using real emails from the public Enron dataset (**Principles 3**). We publicly provide our experimental results to facilitate quantitative comparisons with related studies that utilize the same public dataset. This facilitates a more practical analysis for future research.

## 1.6 Thesis Outline

This thesis is structured as follows: Chapter 2 provides a comprehensive background on our research context, starting with conformance checking methods. It then explores the evolution of process prediction techniques and studies email recommendation systems. Finally, it discusses and compares these various approaches, emphasizing the unresolved research questions.

Chapters 3 and 4 constitute the core of our thesis, elaborating on our main contributions. Chapter 3 delves into an efficient approach for multi-perspective conformance checking for email-driven processes. Chapter 4 introduces a novel process-activity-aware email response

recommendation system. Finally, Chapter 5 concludes this thesis by summarizing the presented work and discussing possible perspectives.

We note that Appendix A marks a departure from the primary theme of the thesis. Driven by deep curiosity and drawing inspiration from the investigations in Chapter 4, in this appendix we studied the domain of anomaly detection in streaming data, particularly within the context of the Internet of Things (IoT). We therefore introduced a novel unsupervised method, termed "**Track Before Detect**" (TBD), specifically designed to detect anomalies in IoT time-series data.



# State Of The Art

---

## Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>17</b>
<b>2.2</b>	<b>Conformance Checking in business process Mining</b>	<b>17</b>
2.2.1	Traditional Conformance Checking Methods	18
2.2.2	Multi-Perspective Conformance Checking in Process Analysis	19
2.2.3	Synthesis & Discussion	21
<b>2.3</b>	<b>Examining Email Recommendation and Process Prediction</b>	<b>22</b>
2.3.1	Process Prediction from Execution Logs	22
2.3.2	Email Recommendations	25
2.3.3	Synthesis & Discussion	29
<b>2.4</b>	<b>Conclusion</b>	<b>36</b>

---

## 2.1 Introduction

In this chapter, we aim to review existing studies relevant to our research questions. We begin with an examination of methods for conformance checking in Section 2.2. Following this, in Section 2.3, we will conduct a comprehensive exploration of process prediction based on execution logs, which includes a detailed discussion on email recommendation.

## 2.2 Conformance Checking in business process Mining

Conformance checking plays a crucial role in BP mining, ensuring that executed processes align with predetermined models. Section 2.2.1 elucidates how traditional conformance checking methods have laid the groundwork for identifying deviations and inefficiencies in process execution. These methods range from rudimentary token replay techniques to more sophisticated fitness measurement methods.

The traditional approaches primarily focus on four categories of discrepancies: deviations, repetitions, omissions, and insertions. However, due to the escalating complexity of

contemporary processes, there is an increasing need to broaden the perspectives considered in conformance checking. To address this evolving need, Section 2.2.2 explores emerging multi-perspective conformance checking methodologies. These innovative approaches integrate multiple perspectives, including time, resources, and data attributes. They provide a more intricate view of process execution, enabling a deeper analysis. Importantly, these approaches allow the field to adapt more effectively to the growing complexity of modern operational processes.

In combination, traditional and multi-perspective conformance checking methods contribute to the continuous enhancement of process performance. This synergistic relationship reflects the evolving needs and challenges faced by various industries, underscoring the importance of ongoing advancement and adaptation in conformance checking. While traditional and multi-perspective methods have long been considered the cornerstone of process conformance checking, they do have limitations, particularly when applied to the context of emails. The shortcomings of these methods are extensively discussed in Section 2.2.3.

### 2.2.1 Traditional Conformance Checking Methods

Traditional conformance checking methods in BP mining encompass a collection of techniques developed to compare an observed event log with a predefined process model [37]. The objective is to uncover any disparities between the actual execution of a process and the anticipated process flow. Among the most widely used strategies within traditional conformance checking lies token replay [1]. In this approach, the event log is reenacted on the process model to ascertain if each event aligns with the anticipated sequence of activities. The process model is represented as a Petri net, and tokens are employed to simulate process execution. Every event in the log corresponds to a transition in the Petri net, and the tokens are adjusted accordingly. If the tokens can traverse the Petri net in a manner that matches the event sequence in the log, it signifies a high degree of conformance. However, if tokens encounter obstacles or require artificial creation, it indicates a deviation from the expected process model.

Another prevalent technique in traditional conformance checking is fitness measurement [5]. This approach quantifies the level of alignment between the event log and the process model. The fitness measure is computed based on the count of missing and remaining tokens after the event log is replayed on the process model. A high fitness value indicates that the process model can accurately reproduce the behavior observed in the log, indicating strong conformance. Conversely, a low fitness value suggests significant disparities between the model and the log.

Traditional conformance checking methods have primarily been dedicated to detecting four key types of disparities, as outlined in previous research [24, 110, 48]. The first type of disparity, known as deviations, denotes deviations in the sequence of activities from the standard process flow. Such deviations can arise due to diverse factors, including human errors, system malfunctions, or changes in the operational environment. For instance, an operator's error might lead to an abnormal sequence in a manufacturing assembly line process,



such as attaching part B before part A. Detecting and comprehending these deviations is crucial for organizations to enhance their processes and prevent future disparities.

The second disparity type, repetitions, refers to instances where the same activity is redundantly performed multiple times. Such repetitions can result in inefficiencies, increased costs, and process delays due to unnecessary resource consumption. For example, a document like an invoice being reviewed and approved by the same individual multiple times due to communication breakdowns within a team. Identifying and eliminating these repetitions enables organizations to streamline processes and eliminate redundant activities.

Omissions constitute the third form of disparity, occurring when vital activities are skipped. Such omissions can lead to severe consequences, including incomplete or incorrect process outcomes. These gaps often stem from human errors, system failures, or miscommunications. For instance, in a software development process, an omission might arise if the testing phase is omitted, potentially resulting in undiscovered software bugs, negatively impacting user experience, and posing security risks. Identifying omissions is crucial to ensure process integrity and correctness.

Finally, the fourth disparity type, insertions, involves adding superfluous activities to the process. These insertions can lead to inefficiencies by consuming resources without adding value and disrupting process flow. An instance of an insertion could occur in a hiring process, where an unnecessary additional review of an applicant's resume is introduced before the scheduled interview, causing delays and inefficiencies. Detecting and eliminating such insertions empowers organizations to enhance process efficiency and effectiveness.

However, traditional conformance checking methods have limitations. They predominantly focus on the sequential progression of activities and might not sufficiently capture the entire spectrum of intricate dependencies and process variations. For instance, they might not adeptly identify disparities related to activity timing, resource utilization, or achieved outcomes. To address these challenges and provide a more comprehensive perspective, the following section introduces the concept of Multi-Perspective Conformance Checking in Process Analysis, aiming to analyze processes from diverse angles and dimensions.

### 2.2.2 Multi-Perspective Conformance Checking in Process Analysis

In recent years, the field of conformance checking for multi-perspective processes has witnessed significant advancements. These progressions have transcended the confines of traditional process models, encompassing additional perspectives like time, resources, and other data attributes [33, 49, 115, 76]. This expansion has been motivated by the realization that these supplementary viewpoints can offer valuable insights into process execution and performance.

An eminent research domain within this field revolves around time-aware conformance checking [96]. This methodology centers on the temporal facets of events during process execution, aiming to identify anomalies in event durations, delays, and timeouts that might not be evident solely by considering the event sequence. This approach proves particularly

beneficial in time-sensitive processes such as manufacturing or logistics, where delays can exert a substantial impact on overall process efficiency.

Time-aware conformance checking typically entails aligning event timestamps with the process model, facilitating a direct comparison between anticipated and actual event timings. By identifying disparities between these two, it becomes feasible to identify process segments that deviate from expectations. Temporal reasoning techniques, encompassing interval-based comparisons and sequence-based comparisons, are employed in this approach, offering a more nuanced comprehension of process performance.

Another dimension integrated into conformance checking is resource-awareness [97, 21, 32]. This perspective concentrates on the allocation and utilization of resources throughout process execution, with the objective of identifying instances of resource under-utilization, over-utilization, or improper allocation. Resource-aware conformance checking can furnish valuable insights into process efficiency. For instance, consistent overload or frequent idleness of a resource during process execution might signify inefficiencies in resource allocation. Identifying these inefficiencies facilitates process adjustments to enhance resource utilization and overall process efficiency.

In resource-aware conformance checking, conformance is evaluated by aligning the event log with the process model, enabling a direct comparison between the anticipated and actual resource utilization. Discrepancies between the two can reveal segments of the process that deviate from expectations. Moreover, resource-aware conformance checking entails fitness assessment, a quantification of the agreement between the actual and expected process behaviors. Evaluating fitness quantifies the extent to which the process adheres to the expected model.

A noteworthy contribution to the multi-perspective conformance checking arena is the Multi-Perspective Declare (MP-Declare) approach introduced by Burattin et al. [20]. This approach amalgamates data, temporal, and control flow perspectives to formulate constraints using Metric First-Order Linear Temporal Logic. This amalgamation offers a more holistic outlook on process performance than any single perspective could deliver.

The MP-Declare approach facilitates efficient conformance checking over event logs by leveraging a constraint template-based algorithmic framework. This framework permits the creation of constraints that can be swiftly verified against the event log. A key advantage of the MP-Declare approach is its time complexity. The approach's time complexity is bounded from above by a quadratic function in the worst-case scenario, implying that the time required for conformance checking using the MP-Declare approach grows proportionally to the square of the input size. This attribute renders the approach viable for application with large event logs, a common occurrence in many real-world processes.

### 2.2.3 Synthesis & Discussion

Extensive research has explored conformance checking techniques for multi-perspective processes, focusing on structural and operational aspects while considering diverse data attributes. However, a significant gap exists in acknowledging the crucial contextual perspective of events. Current approaches have overlooked the analysis of the email business data attribute, which is essential for comprehending the intended meaning of messages in email-driven processes.

Table 2.1 presents a comparison of these conformance checking techniques used in process mining, detailing their characteristics across several attributes. The first column, *Technique*, lists the different methods being compared. The second column, *Description*, provides a brief explanation of how each technique operates. The third column, *Key Features*, highlights the primary capabilities and functionalities of each method. The fourth column, *Disparities Detected*, details the types of process discrepancies that each technique can identify. Finally, the fifth column, *Limitations*, outlines the potential drawbacks or challenges associated with each technique.

In this table, we can clearly see, for example, that MP-Declare integrates data, temporal, and control flow perspectives but fails to address the contextual nuances of email content. An email containing a request for approval might carry different business data, signifying urgency or priority, which could impact the process flow. Techniques like MP-Declare and traditional methods, such as Token Replay and Fitness Measurement, are not suitable for use in the context of email-driven processes due to their lack of consideration for the contextual attributes of the communication.

**Table 2.1:** Comparison of Conformance Checking Techniques

Technique	Description	Key Features	Disparities Detected	De-	Limitations
Token Replay	Reenacts event log on process model using tokens	Simulates process execution with Petri nets	Deviations, Repe- titions, Omissions, Insertions		Focuses on sequence of activities, may miss timing and resource issues
Fitness Measurement	Quantifies alignment between event log and process model	Measures missing and remaining tokens after replay	Deviations, Repe- titions, Omissions, Insertions		May not capture intricate dependencies and process variations
Time-Aware Conformance Checking	Analyzes temporal aspects of events	Identifies anomalies in event durations, delays, and timeouts	Temporal Anomalies		Requires accurate timestamps, may be complex to implement
Resource-Aware Conformance Checking	Focuses on resource allocation and utilization	Identifies resource under-utilization, over-utilization, improper allocation	Resource Utilization Anomalies		Requires detailed resource data, may overlook control flow issues
Multi-Perspective Declare (MP-Declare)	Integrates data, temporal, and control flow perspectives using constraints	Formulates constraints with Metric First-Order Linear Temporal Logic	Multiple perspectives including data, time, and control flow		Complexity in defining and verifying constraints, quadratic time complexity

On the other hand, some research has considered context in Natural Language Processing

(NLP) techniques [90, 40]. However, these approaches are limited for conformance checking in email-based processes as they ignore processes within these approaches. NLP methods predominantly rely on text analysis techniques and lack consideration for the process information (e.g., emails that belong to a process instance). Email process conformance requires rich abstractions to reason about control flow, data, and context perspectives.

To address these challenges, several criteria need to be resolved based on specific research questions:

- **C1: Integration of Structural and Contextual Perspectives.** This criterion is deduced from research sub-question Q1-1 (Chapter 1, Section 1.3.1).
- **C2: Detection of Discrepancy Patterns.** This criterion is derived from research sub-question Q1-2 (Chapter 1, Section 1.3.1).
- **C3: Handling of Complex Attribute Values.** This criterion is based on research sub-question Q1-3 (Chapter 1, Section 1.3.1).
- **C4: Validation and Measurement of Method Accuracy and Reliability.** This criterion is related to research sub-question Q1-4 (Chapter 1, Section 1.3.1).

In Chapter 3, we propose an approach to mitigate these limitations through multi-perspective conformance checking in email-driven processes. This approach involves analyzing a structured event log generated from email logs alongside a process model specified by a data analyst or expert. This process model is based on constraints related to the sequential and contextual aspects of business processes, reflecting the multi-perspective nature of the approach (**Criterion C1 and C2**). The proposed model is used to check the conformance of events within emails or threads, comparing event log traces to the *Email Process Model* to identify fulfilling and violating events (**Criterion C3 and C4**), ensuring accuracy and reliability.

## 2.3 Examining Email Recommendation and Process Prediction

We provide a thorough analysis of process prediction using execution logs in Section 2.3. We also explore the topic of email recommendation in Section 2.3.2.

### 2.3.1 Process Prediction from Execution Logs

By projecting the future states of ongoing process executions at runtime, process prediction techniques have been presented as a way to improve BPM. By examining past process executions, these forecasts are produced [64]. These strategies give organizations access to timely information so they can respond promptly to risks and take corrective action.

We will look at two different types of process prediction methods in the following sections. We will start by discussing Uni-dimensional Process Prediction Techniques (Section 2.3.1.1), which are used to predict particular business process features. Process prediction approaches based on multidimensional data (covered in Section 2.3.1.2) will be looked at next; these approaches take a wider variety of data into account when making predictions.

### 2.3.1.1 Uni-dimensional Process Prediction Techniques

In this section, we explore Uni-dimensional Process Prediction Techniques, which focus on predicting specific aspects of business processes by leveraging linear data. These techniques are designed to address singular, well-defined dimensions of business operations, such as resource allocation, process duration, risk prediction, and future process behaviors. A fundamental component of these techniques is machine learning. For instance, in a study by Camargo et al. [22], machine learning was used to forecast subsequent stages in a manufacturing process. Models were trained on past data to accurately predict the next phase of production, optimizing resource allocation and preemptively addressing bottlenecks.

Deep learning, a more sophisticated subset of machine learning, has also demonstrated potential in process prediction. Hinkka et al. [61] demonstrated that deep learning models could effectively classify process instances and predict future process behaviors, contributing valuable insights to the management of business processes.

Ensemble learning methods have found their place in process prediction as well. These techniques integrate different learning algorithms to reduce errors and enhance predictive performance, providing a robust alternative to conventional prediction methods.

Moreover, the realm of process prediction isn't confined to machine learning. Traditional statistical methods continue to hold relevance. Conforti et al. [29] used survival analysis, a statistical technique that focuses on the time until an event occurs, to predict the remaining duration of a process. This approach offers precise time estimates, optimizing resource allocation, enhancing customer experience, and making workflows more efficient.

As businesses strive to become more proactive, risk prediction is becoming an integral part of operations. Algorithms can identify patterns and trends that suggest potential risks before they fully materialize by analyzing past and real-time data. Logistic regression is one of the statistical methods used for risk prediction, as demonstrated by Raffaele et al. [29, 30]. It uses historical data and real-time inputs to identify potential risks before they become fully evident. By analyzing the association between predictor variables (like trends and patterns) and binary outcome variables (like the occurrence of a risk event), this method calculates the chance of a risk event.

Process prediction tasks have also demonstrated the effectiveness of Automated Transition Systems (ATS). ATS is a kind of model that represents several stages and transitions of a process and is used in process mining and predictive analytics. Expanding on the traditional ATS model, Aburomman et al. [4] incorporated features extracted from business logs, pro-

viding a richer context for the prediction task. The study then employed a linear regression technique to predict the remaining time of new traces in the BP.

### 2.3.1.2 Process Prediction Approaches Based on Multidimensional Data

In this section, we explore Process Prediction Approaches Based on Multidimensional Data, which utilize a diverse array of data variables to forecast future outcomes and behaviors in business processes. Multidimensional approaches go beyond the limitations of uni-dimensional methods by integrating various attributes such as timestamps, activity types, user roles, and resource allocations, among others. These methodologies leverage advanced machine learning techniques, particularly deep learning and hybrid models, to handle the complexity and richness of multidimensional data [99, 84].

For instance, Mehdiyev et al. [79] introduced a multi-stage deep learning approach to predict the next process event from completed activities of running process instances. The foundation of this approach lies in the utilization of execution log data from previously completed process instances, encompassing diverse attributes such as timestamps, activity types, user roles, and resource allocations. The model's architecture incorporates stacked auto-encoders and a supervised fine-tuning component. A key strength of this proposed approach is its adeptness in managing the intricacies of multidimensional data input. Notably, the paper also dedicates attention to two significant aspects: identifying appropriate hyper-parameters for the proposed method and managing the inherent imbalanced nature of business process event datasets.

On another innovative front, Ebrahim et al. [38] introduced an ensemble method that combines decision trees and neural networks to predict manufacturing process outputs. This method harnesses the robustness of Random Forests, a decision-tree-based model, in capturing feature interactions and handling missing values, alongside the versatility of Deep Feed-forward Neural Networks (DNNs) [10] in modeling complex relationships. The intermediary outputs provided by the decision trees act as inputs to the DNNs, forming a layered prediction mechanism that exploits the strengths of both approaches, resulting in a marked improvement in prediction accuracy.

Recent advancements in Generative Adversarial Networks (GANs), particularly in the manufacturing sector, cannot be overlooked [112, 15]. GANs consist of two neural networks—a generator and a discriminator—that work together to generate new synthetic instances of data that can pass as real data. This method, often used for generating new examples in datasets, has been adapted to predict possible future states in manufacturing processes based on current data.

Hybrid models, which combine different predictive models, have become a powerful tool in the field of prediction, achieving superior predictive performance. Their applications span diverse domains, including industrial robotics, animal nutrition, laser technology, and solar radiation forecasting [63, 106, 67]. In these applications, the hybrid model typically consists of a combination of a physics-based model and Gaussian Process Regression (GPR) [107].

The physics-based model captures well-established variable relationships within the system, while the GPR excels at accounting for subtle nonlinear relationships and interactions within the data, enhancing the predictive accuracy of the hybrid model.

Furthermore, Bayesian models [103, 47] have also gained considerable attention in multidimensional process prediction, offering a systematic way to incorporate prior knowledge into the model and adjust predictions based on the evidence provided by the data. One of the most prominent Bayesian models is the Bayesian Network, which has shown tremendous promise in handling high-dimensional, multivariate data.

Additionally, statistical modeling remains a key player in process prediction based on multidimensional data, especially when variables exhibit a well-defined linear relationship. Sarswatula et al. [92] leveraged multiple linear regression models to predict energy consumption in manufacturing processes, incorporating both direct and indirect influences on energy consumption. This approach underscores the critical role of statistical models in process prediction, particularly when dealing with large datasets with linear and well-defined relationships among variables.

## 2.3.2 Email Recommendations

The advent of technology has sparked a revolutionary change in communication, particularly due to the widespread use of emails. As the daily exchange of emails continues to rise, there is a growing demand for automated and intelligent email systems. Imagine receiving timely suggestions for answering queries or generating appropriate responses to emails. This level of assistance can streamline communication and save valuable time for users.

To explore these advancements further, we will focus on three major areas of email communication improvement. First, we will delve into the methodologies and approaches used in recommending email fields, which will be discussed in Section 2.3.2.1. Second, we will detail the methods employed in answering email questions, elaborated upon in Section 2.3.2.2. Lastly, we will explain the techniques used to suggest email responses, covered in Section 2.3.2.3.

While these developments are undoubtedly promising, it is essential to acknowledge that existing email recommendation methods have certain limitations. In Section 2.3.3, we will delve into these limitations and explore potential opportunities for further progress in email management.

### 2.3.2.1 Approaches for Recommending Email Fields

The use of natural language processing methods and machine learning algorithms is crucial to the development of email recommendation systems. Various email response fields, such as recipients, attachments, subjects, and named entities, are predicted with the use of these technologies [102, 88, 50, 72]. These methods' main goal is to improve users' overall email

experience by making email composition easier and decreasing the likelihood of mistakes.

One notable method has been suggested by Qadir et al. [88] and is based on machine learning algorithms. Using both the user's past data and the content of the email being composed, this approach aims to anticipate the most likely recipients of emails. A machine learning model is trained using the user's previous email patterns in order to accomplish this. The model considers variables including the frequency of emails sent to particular recipients, the email's language style, and the subjects that are typically covered. This trained model assists during the email composition process by suggesting the most likely recipients based on the email's content and context. As a result, this method not only streamlines email composition but also serves as a precaution against mistakes, such as sending an email to the wrong recipient.

One further important strategy is to suggest appropriate files for the email attachment. One such system was introduced by Dredze et al. [34], which notifies users when a file has to be attached to an email before sending it. Using natural language processing techniques, this system looks for textual clues in emails that could indicate the need for an attachment. For instance, the system will notify the user to add the required file before sending the email if it recognizes terms like *"in the attachment"*, *"find attached"*, or *"attached is"*.

McCallum et al. [77] provide evidence of the successful application of machine learning and natural language processing techniques in the field of email subject recommendations. Their method entails using email content analysis to forecast appropriate subject lines. They create succinct and informative subject lines by cherry-picking the most important information from the email body. The system can suggest subject lines such as *"Meeting Agenda for Discussion"* or *"Project Proposal Draft"* if the email addresses a *"meeting agenda"* or a *"project proposal"*. This is a really helpful feature for those who frequently struggle to come up with accurate and meaningful subject lines for their emails.

Ashequl et al. [88] have investigated the use of named entities in the email to suggest appropriate ones as another noteworthy strategy. Named entities are particular names that stand in for actual things in the real world, including individuals, groups, places, dates, amounts, and monetary values. Their approach can locate these items in the text and use Named Entity Recognition (NER), a subtask of information extraction, to improve the email's content. For example, the system might recommend that further information be added regarding a project that was addressed in the email or ask the user to clarify a date or time that was perhaps mentioned in passing. This process aids in making sure the email's content is thorough and efficiently communicates the required information.

### 2.3.2.2 Approaches for Recommending Answers to Email Questions

Email response automation systems have become increasingly essential in modern communication, designed to comprehend, interpret, and provide suitable responses to email inquiries. One such system, discussed by Arsovski et al. [14], employs natural language processing to determine whether an incoming email contains a query. The system scans the email's content,



analyzing linguistic structure, semantics, and context to identify questions. Once recognized as a query, the system classifies it into specific categories based on parameters such as subject matter, question complexity, or email tone.

After categorizing emails, Arsovski's system employs a semi-automated approach to generate appropriate responses. This is achieved through the use of the Artificial Intelligence Markup Language, enabling the system to utilize pre-built templates and rules for crafting responses. These templates, designed with numerous potential email queries in mind, offer a structured framework from which the system can derive replies. A significant advantage of this approach is its adaptability. As each user possesses unique needs, the system can adjust its responses based on individual requirements, thereby ensuring higher response accuracy and relevance.

In contrast to Arsovski's approach, Patel et al. [85] proposed a method that not only classifies users' queries but also incorporates an additional layer of processing. In this model, following query classification, the system doesn't merely generate a response from predefined templates. Instead, it strives to find the most analogous query within the same class as the user's query. This approach resembles an intricate matching game, with the system identifying patterns, semantics, and other pertinent factors to establish query similarity. By identifying a match within the same class, the system guarantees a response that is as precise and relevant as possible. However, the method introduced by Patel et al. [85] has its limitations. To function effectively, it necessitates a training dataset associating each class with pairs consisting of a query and its corresponding answer. The challenge lies in the fact that the system can solely respond to queries that have been previously defined and trained. If confronted with a new or unprecedented query, the system might struggle to provide an accurate response. Despite these limitations, this approach offers potential benefits by significantly reducing the time and effort involved in crafting individual responses, making it a valuable tool for managing high email volumes.

### 2.3.2.3 Approaches for Recommending Email Responses

Numerous methods have been developed to simplify the process of generating or suggesting email responses. The complexity of this task lies in deciding whether to reuse a pre-existing email or create an entirely new one. The choice between these options largely depends on the specific needs and circumstances of the situation at hand.

One of the pioneering efforts in this domain was made by Lapalme et al. [70], who introduced a technique based on the cosine similarity measure. This measure quantifies the similarity between the issues and solutions discussed in different emails. By utilizing this method, their system could identify the email that bears the highest similarity to a given situation and subsequently reuse its answer in the proposed response. However, this approach grappled with several limitations: the inherent semantic constraints of cosine similarity, struggles to capture nuanced contextual cues, and potential challenges with synonymy and data sparsity. Furthermore, domain-specific variations and the lack of temporal considerations posed

additional hurdles, potentially leading to inaccurate response reuse. Over-fitting, scalability concerns, and the possibility of misaligned user expectations also underscored the method's pitfalls, while the need for extensive pre-processing further complicated its implementation.

To address some of these issues, other approaches have leveraged the potential of supervised learning techniques for generating email responses. Supervised learning, a subset of machine learning, involves training a model on a labeled dataset, enabling the model to predict outcomes based on learned patterns. By utilizing supervised learning, these systems can draw from vast amounts of existing data to generate relevant and accurate responses.

A notable example of applying supervised learning to email response generation was demonstrated by the team at Google [105]. The Google team's innovative approach to email response generation using supervised learning incorporated several technical steps. Initially, they preprocessed the email data by tokenizing and cleaning the text, enabling efficient handling by the model. They employed the Long Short-Term Memory (LSTM) architecture, a type of Recurrent Neural Network (RNN) [78], to capture sequential dependencies and context within the emails. This architecture enabled the model to learn from paired email-response data, leveraging a sequence-to-sequence learning framework.

During training, the team employed a technique called "*teacher forcing*" in which correct previous tokens from the desired response were provided to the model as it generated the output sequence. This practice helped the model learn by minimizing errors that could accumulate during sequence generation, aiding in understanding how to structure coherent and relevant responses. However, in real-time scenarios without access to these predetermined correct tokens, the model might face challenges. Despite these limitations, their application showcased the power of supervised learning and deep learning in automating email responses.

In another instance, Parameswaran et al. [83] developed a sophisticated approach to automatically generating and suggesting short emails. The core of their method relies on a combination of advanced machine learning classification techniques, weighted keyword analysis, and similarity measurement algorithms. To begin, they utilized a substantial dataset of emails to train Support Vector Machines (SVM). This classifier is adept at recognizing patterns and context within email conversations, enabling it to categorize incoming emails based on their content and intent.

To enhance the specificity of the generated suggestions, the authors incorporated weighted keywords as a crucial component of their approach. These keywords are extracted from the text of incoming emails and assigned varying importance scores to capture the significance of different terms. Doing so allows the system to gain a deeper understanding of the topics at hand, leading to more accurate and relevant response recommendations. Additionally, similarity measurement techniques are employed to compare the content of incoming emails with a repository of previously seen email interactions. This repository acts as a knowledge base, allowing the system to identify historically effective responses to similar contexts.

Despite the remarkable progress mentioned earlier, it's important to note that these advancements might not consistently take the context of the conversation fully into account.

Overlooking context on occasion can result in suggestions that aren't entirely suitable for the current situation. This flaw is of utmost significance, as context plays a pivotal role in human communication, influencing both how we comprehend messages and how we convey them. A crucial feature of an optimal email response system would involve possessing a profound comprehension of context, encompassing all previous interactions and the overarching discourse objectives. By doing so, it would have the capacity to offer more refined and suitable recommendations. Achieving this level of competence demands a fusion of natural language processing and machine learning techniques, similar to the methodologies harnessed in the Generative Pre-trained Transformer (GPT) [52] architecture.

GPT represents a cutting-edge language model capable of generating human-like text, including email responses. The architecture of GPT is rooted in the Transformer model, employing self-attention mechanisms and deep learning techniques. The complexity and depth of GPT's architecture empower it to excel in intricate tasks involving natural language understanding and generation, making it suitable for crafting nuanced and context-aware email responses. To illustrate its prowess, let's delve into a scenario that involves an email exchange between a customer and a company representative. In this instance, the customer is seeking information about the product's availability. Leveraging its intricate architecture, GPT can adeptly formulate a response that precisely addresses the customer's query. The generated email response could unfold as follows:

*Dear [Customer's Name], Thank you for contacting us regarding the availability of the [Product Name]. We value your interest in our products. According to our most recent update, the [Product Name] is presently in stock and ready for purchase. Please be aware that availability may change due to demand, so we encourage you to place your order soon to secure the desired item. If you require further assistance or have additional inquiries concerning your purchase, please do not hesitate to inform us. Our dedicated team is here to assist you. Best regards, [Your Name]*

GPT's contextual comprehension and sophisticated language creation skills are demonstrated in this case. In addition to providing precise information on the product's availability, the response expresses gratitude for the customer's interest. Even while GPT models are quite strong, they frequently act as "black boxes," producing emails that are grammatically correct but occasionally lacking in meaning.

### 2.3.3 Synthesis & Discussion

Existing predictive models have shown success within their specific domains; however, when applied to predicting processes within the context of emails, they often face limitations. The primary challenge lies in the requirement of structured event logs as input for building prediction models. Table 2.2 underscores several critical aspects in this regard by comparing various process prediction techniques used in different domains, detailing their characteristics across several attributes. The first column, *Approach*, lists the different methods being compared. *Input*, the second column, specifies the type of data each approach uses. *Output*, the third

column, describes the type of predictions or results each technique aims to produce. The fourth column, *Method Used*, outlines the specific methodologies or algorithms employed in each approach.

**Table 2.2:** Process Prediction Techniques

Approach	Input	Output	Method Used
Predictive business process monitoring with LSTM neural networks [99]	Historical process executions	Future states of ongoing process executions	Machine learning
Manufacturing Causal Knowledge Discovery using a Modified Random Forest-based Predictive Model [38]	Past data from manufacturing processes	Next phase of production	Machine learning
Using convolutional neural networks for predictive process analytics [84]	Historical process data	Future process behaviors and resource allocation	Deep learning, Ensemble learning
Risk Prediction Models [104]	Historical and real-time data	Risk prediction	Logistic regression
A novel business process prediction model using a deep learning method [79]	Execution log data	Next process event prediction	Multistage deep learning
Predicting the Effect of Processing Parameters on Caliber-Rolled Mg Alloys through Machine Learning [112, 15]	Current data	Possible future states in manufacturing processes	Generative Adversarial Networks (GANs)
Hybrid Models (Industrial Robotics) [63]	Various data sources	Future outcomes and behaviors	Hybrid models combining physics-based models and Gaussian Process Regression (GPR)
Hybrid Models (Solar Radiation Forecasting) [106]	Various data sources	Future outcomes and behaviors	GPR
Hybrid Models (Laser Technology) [67]	Various data sources	Future outcomes and behaviors	GPR
Bayesian Models [79]	Multidimensional data	Predictions incorporating prior knowledge	Bayesian models
Modeling energy consumption using machine learning [92]	Large datasets	Predictions considering linear relationships	Multiple linear regression models

- **Structured Input Requirement:** Most existing models, such as those using machine learning and deep learning [92, 84], rely heavily on structured historical process executions and execution log data. This structured input allows these models to identify patterns and make accurate predictions. In contrast, email data is highly unstructured, making it difficult to apply these techniques directly.
- **Assumptions on Event Logs:** Traditional predictive models assume well-defined case identifiers and a clear event log structure. For instance, logistic regression for risk prediction [104] and linear regression on ATS models [92] rely on structured, timestamped events. However, emails often involve multiple events within the same timestamp, lacking the clear structure assumed by these models.

In contrast, studies that have primarily focused on utilizing recommendation techniques to enhance email management have largely overlooked the crucial integration of business processes, even though they are able to recommend email fields, answer email questions, and

suggest email responses, as shown in Table 2.3. For instance, *Activity Modeling in Email* [88] leverages historical email data and machine learning to recommend email recipients, enhancing email management by suggesting the most relevant contacts. Similarly, *Sorry, I Forgot the Attachment* [34] uses natural language processing on email text to alert users about necessary attachments, preventing common email errors. Another example is the *GPT-3 model* [52], which generates email responses based on email text and previous interactions, streamlining communication through advanced language processing.

**Table 2.3:** Email Recommendation Techniques

Approach	Input	Output	Method Used
Activity modeling in email [88]	Historical email data	Email recipient recommendations	Machine learning
Sorry, I Forgot the Attachment: Email Attachment Prediction [34]	Email text	Alerts for necessary attachments	Natural Language Processing
Topic and role discovery in social networks with experiments on enron and academic email [77]	Email content	Suitable email subject lines	Machine learning and Natural Language Processing
Named Entity Recognition (NER) [88]	Email text	Named entity recommendations	Named Entity Recognition (NER)
An approach to email categorization and response generation [14]	Email content	Query classification	Natural Language Processing
Customized Automated Email Response Bot Using Machine Learning and Robotic Process Automation [85]	Historical email interactions	Query-response matching	Supervised learning
Mercure: Towards an automatic e-mail follow-up system [70]	Past emails	Suggested email responses	Cosine similarity
Case-Based Reasoning: A recent theory for problem-solving and learning in computers and people [105]	Paired email-response data	Generated email responses	Supervised learning
Automatic email response suggestion for support departments within a university [83]	Substantial email dataset	Short email generation	Support Vector Machines (SVM)
GPT-3: Its nature, scope, limits, and consequences [52]	Email text and previous interactions	Generated email responses	Generative Pre-trained Transformer (GPT) model

Recent research efforts, exemplified by the pioneering work of Chambers et al. [26],

have taken a new direction in this domain. Their work introduces innovative methodologies that explore the intricate relationship between email content and the underlying business processes. Chambers and his team employed advanced text mining techniques to meticulously extract and categorize business processes from emails. This categorization involves identifying key activities, the roles responsible for these activities, the temporal sequencing of tasks, interactions among stakeholders, and indicators of process evolution.

Another notable contribution to this field originates from [66], who devised a methodology using machine learning to discern patterns and correlations among various business activities mentioned in emails. Their work provided valuable insights into the structure and progression of these activities.

Despite the significant progress made in these research endeavors, it is crucial to recognize that their scope has primarily centered around the exploration and categorization of business processes contained within email communications. However, as we mentioned previously, the application of predictive models to process-oriented emails should not solely revolve around identifying upcoming BP activities conducted through emails. It should also encompass the task of suggesting relevant emails that facilitate BP actors in carrying out these activities, with a particular focus on the textual content present in their email responses.

To tackle these challenges, various criteria must be addressed according to the specific research questions:

- **C1: Effectively leveraging event logs from previously exchanged emails.** This criterion is deduced from research sub-question Q2-1 (Chapter 1, Section 1.3.2).
- **C2: Employing suitable machine learning algorithms or predictive models.** This criterion is deduced from the research sub-question Q2-2 (Chapter 1, Section 1.3.2).
- **C3: Considering relevant event attributes.** This criterion addresses sub-question Q2-3 (Chapter 1, Section 1.3.2).
- **C4: Personalizing recommended email responses.** This criterion is derived from research sub-question Q2-4 (Chapter 1, Section 1.3.2).
- **C5: Validating the effectiveness and performance of the recommendation system.** This criterion is deduced from the research sub-question Q2-5 (Chapter 1, Section 1.3.2).

To illustrate the practical implications of these criteria, consider the internal hiring process in large organizations. This process follows a systematic approach where each job offer must be meticulously crafted, reviewed, and signed by relevant staff members before publication. Once published, the organization receives numerous applications, each confirmed via email. For selected candidates, interviews are scheduled, confirmed, and conducted, ultimately leading to the retention and hiring of one or more candidates. These activities predominantly rely on email communication, as job offers are shared, applications are submitted, and interviews are scheduled through this medium.

Given the scale of large organizations, with dozens of new positions regularly opening, the described process is applied to each job position. Handling dozens of applications per position, the hiring staff must send individual confirmation emails, review applications, and send follow-up emails. These repetitive and tedious activities can lead to inefficiencies without automation and predictive assistance.

Confirmation and interview scheduling emails often contain the same body content, with minor differences such as the applicant's name and interview time. For example, Fig. 2.1 illustrates two interview-setting emails from the Enron dataset, showing high similarity in content with variations in specific details like applicant names, position titles, and interview times.

<p><i>Good morning <b>Yongcho</b>:</i></p> <p><i>The Enron Corp. Research Group would like to conduct a telephone interview with you at your convenience. This would be for a <b>full-time position</b> with the Research Group.</i></p> <p><i>Please let me know your availability <b>Monday, May 1 or Thursday, May</b></i></p> <p><i>The persons who will be interviewing you are:</i></p> <p><i>Vince Kaminski <b>Managing Director</b> Stinson Gibner <b>Vice President</b> Osman Sezgen <b>Manager</b></i></p> <p><i>I look forward to hearing from you. Administrative Coordinator</i></p>	<p><i>Good morning <b>Amy</b>:</i></p> <p><i>The Enron Corp. Research Group would like to conduct a telephone interview with you at your convenience. This will be as a <b>"summer Intern"</b> with the Research Group.</i></p> <p><i>Please let me know your availability on <b>Monday, May 1st or Thursday; May 4th.</b></i></p> <p><i>The persons who will be interviewing you are:</i></p> <p><i>Vince Kaminski <b>Managing Director</b> Stinson Gibner <b>Vice President</b> Osman Sezgen <b>Manager</b></i></p> <p><i>I look forward to hearing from you. Thank you and have a great day Administrative Coordinator</i></p>
<i>Interview Setting Email 1</i>	<i>Interview Setting Email 2</i>

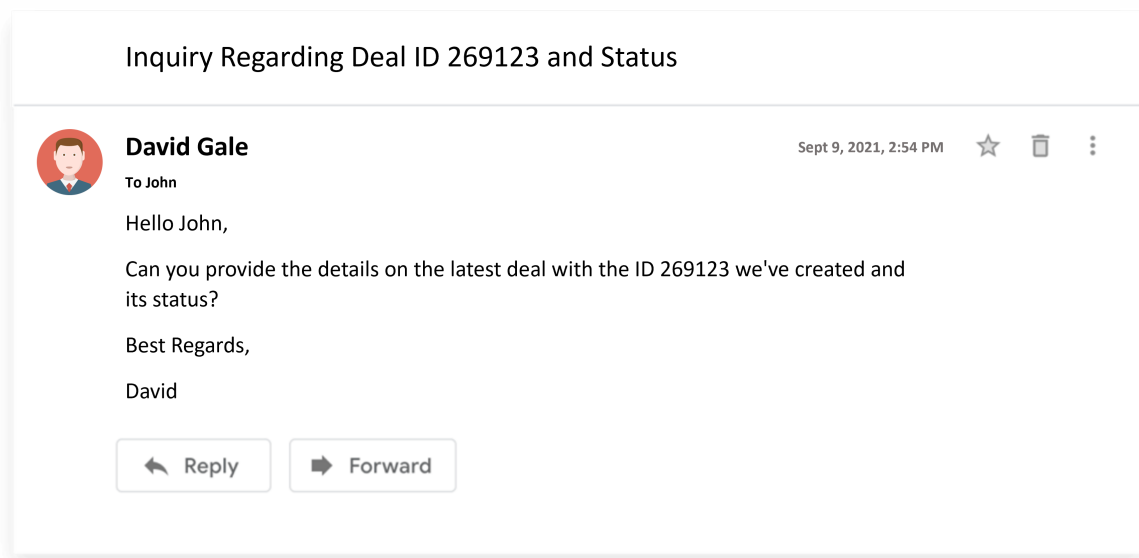
**Figure 2.1:** Example of Interview Setting Emails

Despite minor differences, manually sending each email for every application is time-consuming and error-prone, leading to delays in the hiring process. Therefore, we explore two approaches: traditional deep learning models, such as Generative Pre-trained Transformer (GPT) models, and a specialized process-aware email recommendation system that we aim to develop.

- **GPT Models:** While GPT models are powerful, they often function as black boxes, generating grammatically correct emails that can sometimes be contextually inadequate. This limitation highlights the need for models that can be better understood and controlled to align with specific business process activities.
- **Process-Aware Email Recommendation System:** This approach aims to aid employees in composing efficient and accurate emails. Upon receiving an email, the system would analyze its content, extracting relevant Business Process knowledge (Criterion C1). It would decipher the intended activity (e.g., processing an application), the purpose of communication, pertinent Business Data, and determine if a response is required. Based on this analysis, the system would predict the next BP activities (Criteria C2 and C3) and generate an appropriate email response template (Criterion C4). This

system would also consider the context and individual communication styles, ensuring personalized and contextually relevant responses (Criteria C4 and C5).

To highlight the potential of a Process-Aware Email Recommendation method compared to the GPT model, we provide a use case that underscores these distinctions. In this case study, our objective is to determine the best possible email response to an email sent by David. Subsequently, we contrast the recommendation of the Process-Aware Email Recommendation method with the response produced by the GPT-4 model.




**Figure 2.2:** Email Correspondence: Message from David

David's email, as illustrated in Figure 2.2, is addressed to John, asking for an update on a deal with the ID number 269123. This email primarily centers around the activity "create a deal" linked with Business Data that consists of terms like "created", "deal" and "269123" along with the act of requesting information.

To draft potential email responses, we utilize both approaches. The Process-Aware system would recommend an email response template known as the *recommended-email*, as depicted in Figure 2.3. This email would incorporate the predicted Business Process knowledge that should be included in the response, based on the received email, in this case, from David. Conversely, employing the GPT-4 model results in a distinct email, termed the *generated-email*, illustrated in Figure 2.4. Armed with David's email and the corresponding response options, we proceed to conduct a comparison. Our evaluation will focus on assessing the coverage of relevant Business Process knowledge and the logical flow of ideas.

- **Coverage of Relevant Business Process Knowledge:** The *recommended-email* would provide more complete coverage of the pertinent Business Process knowledge than the *generated-email*.




 **Edit Template**

Insert Token

**Subject:** Confirmation of Deal Entry: ID {deal\_id}

**Message:**

 INSERT TOKEN SIGNATURES

Dear {recipient\_name} ,


The deal with the ID {deal\_id} has been entered into our system. The details of the deal have been noted.

Regards,

{sender\_name}

**Save**

**Figure 2.3:** Recommended Email Response Candidate: Leveraging Our Approach

 **ChatGPT**

**Subject:** Re: Inquiry Regarding Deal 269123 Status




Hi [Recipient],

Thank you for your email and your interest in our deals. Regarding Deal 269123, I'd be happy to provide you with the latest updates and details.

Currently, Deal 269123 is in the final stages of negotiation. We are working closely with the involved parties to finalize the terms and conditions. Once everything is confirmed, I will promptly update you on the status and any further developments.

If you have any specific questions or require additional information, please feel free to let me know. I'm here to assist you.

Best regards,  
[Your Name]

**Figure 2.4:** Generated Email Response Candidate: GPT-4 Model Application

- In the original email, David requested both the "details" and "current status" of the deal. The *recommended-email* would acknowledge both, mentioning that the deal has been entered and its details noted. The *generated-email* lacks clarity in conveying the status of Deal 269123. It briefly mentions that it's in the final negotiation stages but doesn't clearly state whether it has been completed or what the current status is. This ambiguity might leave the recipient confused. Also, the response doesn't directly address the recipient's inquiry about the status of Deal 269123. Instead, it provides vague information about the deal being in the final stages of negotiation without confirming its completion or providing a concrete update on its status.
- **Logical Flow of Ideas:** The *recommended-email* would demonstrate a better logical flow of ideas compared to the *generated-email*.
  - David initially asked for details on the latest deal with ID 269123 and its current status. The *recommended-email* would address this by referencing the specific deal with ID *deal\_id* and its entry into the system. It would further assure David that the deal's details have been documented. However, the *generated-email* lacks coherence and clarity in addressing the recipient's inquiry about the status of Deal 269123. It fails to provide a direct response to the question, instead offering vague information about the deal being in the final negotiation stages.

In conclusion, the *recommended-email* is the better option because of its coherent structure, thorough description of the deal's conditions, and consistency with the content of the initial email. Thus, in Chapter 4, we present a process-aware email recommendation system that takes a generated event log from previously sent emails as input and predicts future BP knowledge. This includes predicting the set of activities to be expressed in the email response, the intention behind expressing them, and the manipulated business data. Additionally, we provide an email response body template recommendation based on the predicted activities and historical textual contents related to the predicted BP knowledge.

## 2.4 Conclusion

In this chapter, we have systematically reviewed existing studies and methodologies pertinent to our research domains, primarily focusing on conformance checking in BP mining and process prediction based on execution logs, as well as email recommendation systems.

We began by examining traditional conformance checking methods, highlighting their role in ensuring alignment between executed processes and predefined models. These methods, although foundational, exhibit limitations when applied to complex, modern processes. The evolution towards multi-perspective conformance checking was then discussed, emphasizing its ability to integrate additional dimensions such as time, resources, and data attributes to offer a more comprehensive analysis of process execution.

---

The chapter also explored process prediction techniques, categorizing them into unidimensional and multidimensional approaches. We noted the significant advancements in machine learning and deep learning, which have enhanced the accuracy and applicability of these techniques in various operational contexts. However, we identified gaps, particularly in handling the unstructured nature of email data within these predictive models.

In the domain of email recommendation, we detailed various methodologies for recommending email fields, generating responses, and answering email queries. Despite the progress in this area, current systems often fail to integrate the broader context of business processes, leading to suboptimal performance in email-driven processes.

In summary, while traditional and emerging methods in conformance checking, process prediction, and email recommendations have laid substantial groundwork, they present limitations, especially in the context of email communications. The necessity for an integrated approach that considers both process and context perspectives is evident. In the subsequent chapter, we will explore our first contribution: an approach for conducting conformance checking within the context of emails.



# Multi-Perspective Conformance Checking For Email-driven Processes

---

## Contents

<b>3.1</b>	<b>Introduction</b>	<b>39</b>
<b>3.2</b>	<b>The Proposed Approach Overview</b>	<b>40</b>
3.2.1	The Email Process Model	41
3.2.2	The Conformance Checking Algorithm	51
<b>3.3</b>	<b>Proof of Concept</b>	<b>53</b>
<b>3.4</b>	<b>Experiments and Validation</b>	<b>58</b>
3.4.1	Detection of Non-Conformance in Enron Email Logs	58
3.4.2	Use Case Study	59
<b>3.5</b>	<b>Conclusion</b>	<b>61</b>

---

## 3.1 Introduction

Conformance checking, a vital component of process mining, involves comparing actual process behavior with expected behavior [91]. While various conformance checking techniques have been developed for business processes executed in Business Process Management Systems (BPMS), they are not specifically tailored for email-driven processes—those that unfold within email systems rather than within BPMS, as discussed in Chapter 2. These methods fall short in identifying non-conformance in events from a complex business context perspective, such as emails. They typically assume that the business context of an event is predefined based on the process to which it belongs [37]. However, they fail to account for the possibility that an event in the same process instance could deviate and be related to another process. This limitation is particularly evident when dealing with emails or conversations, where topics often shift abruptly. Therefore, it is crucial to distinguish between email-driven processes and traditional ones when reasoning about conformance checking of processes. In fact, while traditional business processes have a clear temporal control flow in which one activity follows

another in a well-defined sequence, email-driven processes do not exhibit this characteristic. The order of the activities involved in such processes is instead determined by their appearance within the emails and the sequencing of these emails within the threads of conversations. This order is reflective of how users perceive the execution of activities. Adding the fact that email-driven processes usually represent fragments of business processes as opposed to complete ones, it is challenging to identify the control flow in such processes as it is not explicitly temporal but rather based on the structure of email conversations. Consequently, traditional process models are not capable of accommodating the unique characteristics of email-driven processes.

To address these limitations, this chapter proposes an efficient approach for multi-perspective conformance checking in the context of email-driven processes. Specifically, we focus on two perspectives: (i) the sequential flow of events and (ii) the contextual perspective of the events. Section 3.2 will explore the details of our proposed approach, outlining the key steps and techniques used for conformance checking on email-driven processes, forming the foundation for understanding our methodology. Section 3.3 presents a proof of concept. Section 3.4 presents the evaluation of our approach through various experiments and analyses, demonstrating its effectiveness and efficiency. Finally, Section 3.5 will conclude the chapter by summarizing our findings and discussing some limitations.

## 3.2 The Proposed Approach Overview

The approach, as depicted in Figure 3.1, receives as input a structured event log generated from an email log using the methodology proposed by Elleuch et al. [43], in conjunction with a *process model*. The latter is specified by a data analyst/expert and is built upon a set of constraints that pertain to both the *sequential* and *email contextual* aspects of the business process, highlighting the multi-perspective nature of our approach. Such constraints are guided by *constraint classes* that we have gleaned through the analysis of email datasets and that serve as templates for the analysts/experts to define their specific constraints.

Using the constraints defined by the experts, we proposed a process model, the *Email Process Model*, which we use for conformance checking of events within the same email or thread. Consequently, the conformance checking algorithm proposed in our approach involves comparing the execution of a process instance (i.e., an event log instance) with the process model. This comparison helps identify two sets of events termed *fulfilling* and *violating* events. The former refers to events that occur within a process instance and align with the expected behaviors, while the latter refers to events that violate *sequential* and/or *email contextual* constraints. In addition, the conformity-checking algorithm is able to identify missing topics, highlighting deviations or inconsistencies in the context.

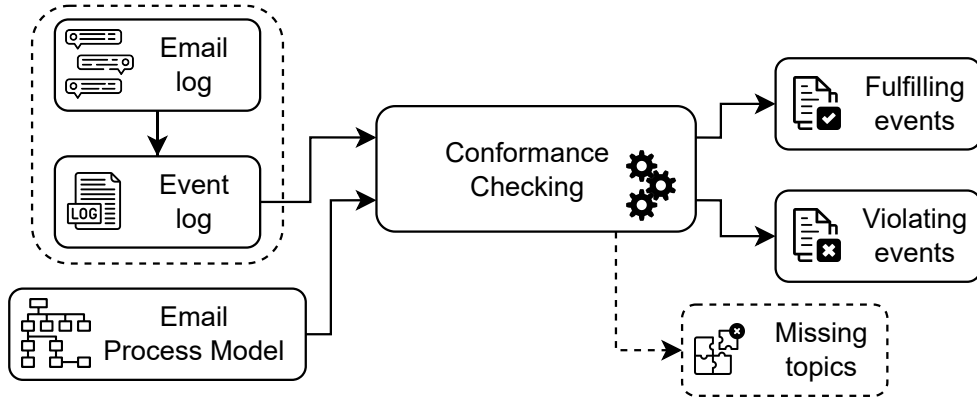


Figure 3.1: Approach overview

### 3.2.1 The Email Process Model

In the context of conformance checking techniques, a reference model is a representation of the expected behaviour of the running process. It serves as the standard against which actual process executions are compared to assess their adherence to predefined expectations. The representation of such a model can take several forms including imperative and declarative representations. In this work, we propose a declarative approach that gives the designer the possibility to define a reference model that we call *Email Process Model* by means of expressing constraints on the activities involved in the email process. Such a declarative approach prioritizes the definitions of desired outcomes over the explicit specification of the steps to achieve them. In order to express the constraints, our approach provides the designer with a set of *constraint classes* that we have extracted and defined following an analysis of email datasets. These constraint classes would serve as templates for the designer to express the actual constraints and they cover both the sequential and contextual perspectives of email-driven processes, and therefore provide a partial view on their respective BPs. In the following, we start by providing essential email-related definitions in section 3.2.1.1, followed by our proposed constraint classes in section 3.2.1.3.

#### 3.2.1.1 Email-Related Concepts

An email log can be viewed as a set of *emails*, each characterized by a set of attributes as outlined in Definition 3.1.

**Definition 3.1** (Email). *Let  $\mathcal{EM}$  be the set of all emails. An email  $em \in \mathcal{EM}$  is a tuple  $\langle ID, timestamp, from, to, sbj, body \rangle$  where:*

- *ID refers to its unique identifier,*
- *timestamp refers to its sending time,*

<p>Message-ID: &lt;1552589.1075853972210.JavaMail.evans@thyme&gt; <b>E1</b>  Date: Fri, 10 Nov 2000 06:17:00 -0800 (PST)  From: aimee.lannou@enron.com  To: daren.farmer@enron.com  cc:  Subject: Flow w/ no nom</p> <p>Meter 1601 last deal 412219 for 10/00 flowed 11/9</p> <p>Meter 5192 last deal 454057 for 10/00. flowed 11/3-4</p>	<p>Message-ID: &lt;3140966.1075854206364.JavaMail.evans@thyme&gt; <b>E3</b>  Date: Tue, 9 Jan 2001 04:43:00 -0800 (PST)  From: aimee.lannou@enron.com  To: daren.farmer@enron.com  cc: edward.terry@enron.com  Subject: Dec 00</p> <p>Daren - meter 5192 flowed 8 dth on 12/19, 33 dth on 12/20 and 2 dth on 12/29. The last deal for this meter was 454057 in Nov 00. Could you please extend this deal for these 3 days or create a new one? Please let me know.  AL</p>
<p>Message-ID: &lt;29717536.1075854150388.JavaMail.evans@thyme&gt; <b>E2</b>  Date: Wed, 15 Nov 2000 02:24:00 -0800 (PST)  From: daren.farmer@enron.com  To: aimee.lannou@enron.com  cc:  Subject: Re: Flow w/ no nom</p> <p>Rolled deal 454057 to cover flow at mtr 5192.</p> <p>d</p> <p>Aimee Lannou 11/10/2000 02:17 PM  To: Daren J Farmer/HOU/ECT@ECT  cc:  Subject: Flow w/ no nom</p> <p>Meter 1601 last deal 412219 for 10/00 flowed 11/9</p> <p>Meter 5192 last deal 454057 for 10/00. flowed 11/3-4</p>	<p>Message-ID: &lt;26296505.1075854337364.JavaMail.evans@thyme&gt; <b>E4</b>  Date: Tue, 9 Jan 2001 06:48:00 -0800 (PST)  From: daren.farmer@enron.com  To: aimee.lannou@enron.com  cc:  Subject: Re: Dec 00</p> <p>I extended 454057 for the month of December.  D  Aimee Lannou 01/09/2001 12:43 PM  To: Daren J Farmer/HOU/ECT@ECT  cc: Edward Terry/HOU/ECT@ECT  Subject: Dec 00</p> <p>Daren - meter 5192 flowed 8 dth on 12/19, 33 dth on 12/20 and 2 dth on 12/29. The last deal for this meter was 454057 in Nov 00. Could you please extend this deal for these 3 days or create a new one? Please let me know.  AL</p>

**Figure 3.2:** Emails retrieved from Enron data-set

- *from* refers to its expediter,
- *to* refers to the list of its receivers,
- *subj* refers to its subject,
- *body* refers to its textual content

This definition reflects what an email is from the point of view of users. It is however worth noting that in our work, emails are seen as a representation of the execution of a process.

Emails in an email log can be linked through *reply* and *forward* relations. More precisely, an email  $em_j$  can be a response (in the form of reply or forward) to at most another anterior email  $em_i$  (and we note  $em_i \mathcal{R} em_j$ ) and/or be replied to or forwarded in zero or more posterior emails. These relations evoke the notion of *conversations*. An email conversation can therefore be defined as follows:

**Definition 3.2** (Email Conversation). *Let  $\mathcal{CV}$  be the set of all email conversations. An email conversation  $conv \in \mathcal{CV}$  is a set of  $l$  emails  $\{em_i | \forall i \in [1, l], em_i \in \mathcal{EM}\}$  such that, if  $l > 1$  then  $\forall em_i \in conv, \exists em_j \in conv \setminus \{em_i\}$  such that  $(em_i \mathcal{R} em_j) \vee (em_j \mathcal{R} em_i)$ .*

In other words, an email conversation is a set of emails pairwise linked by a response (i.e., reply or forward) relation. For instance, in Fig. 3.2,  $[email_1, email_2]$  and  $[email_3, email_4]$  form two different conversations respectively.



### 3.2.1.2 Email-based Process

Emails pertaining to an email-driven process would express the occurrence of *activities* that are likely to belong to a BP. An activity is formally defined as follows (Definition 3.3):

**Definition 3.3** (Activity). *Let  $\mathcal{A}$  be the set of all activities and  $\mathcal{BD}$  the set of all business data. An activity  $act \in \mathcal{A}$  is defined as a tuple  $\langle AN, \mathcal{BD} \rangle$  such that:*

- *AN is the name of the activity that reflects its main goal;*
- *$\mathcal{BD} \subseteq \mathcal{BD}$  is a set of business data used and generated during activity execution*

An activity is defined as a composition of two components (activity name and business data). Taking the example of the activity creating trading deals; (1) the activity name (*AN*) is 'create deal' and (2) the set  $\{ 'deal price', 'deal identifier' \}$  is included in its business data (*BD*).

*Speech acts* are used to further express the reason of inclusion of an activity in an email, and is defined as follows:

**Definition 3.4.** *Let  $\mathcal{SA} = \{Request, Intention, Information, Request for information\}$  be the set of considered speech acts. Hence, a speech act  $SA \in \mathcal{SA}$  can be a:*

- *Request act: the sender requests that the recipient(s) carry out an activity.*
- *Intention act: the sender expresses a desire for future participation in the activity (by themselves or others).*
- *Information act: the sender uses the email to provide information about activity execution status (whether it has been executed or not or is currently being executed).*

For example, the activity that would be deduced from "I would like to request an interview with the candidate to further discuss their qualifications and suitability for the position" would be associated a speech act Request. Whereas the speech act in "Kate, my assistant, will participate in the telephone interview" would be Intention. The example "The telephone interview has been scheduled for tomorrow at 10 AM" provides information regarding the scheduled time for the activity *conducting an interview* and therefore would have a speech act Information.

To be able to represent an email-driven process model, just like for traditional process models, designers would have to identify the set of activities involved in their models. In our case, these activities would be present in the emails which represent the execution traces of email-driven process model. We also note that an email can involve multiple activities of the model and that an activity can appear in multiple emails. In fact, when an activity occurs in an email we talk about an activity *instance* and we define it as follows:

**Definition 3.5** (Activity Instance). Let  $\mathcal{O}_{act}$  be the set of occurrences of an activity  $act = \langle AN, \mathcal{BD} \rangle \in \mathcal{A}$ . An activity instance  $act_{occ} \in \mathcal{O}_{act}$  is a tuple  $\langle AN_{occ}, \mathcal{BD}_{occ}, em \rangle$  such that:

- $AN_{occ} = AN$ , is the name of the activity corresponding to the occurrence.
- $\mathcal{BD}_{occ} \subseteq \mathcal{O}_{\mathcal{BD}}$ , is the set of business data occurrences related to the activity instance.
- $em \in \mathcal{EM}$ , is the email in which the activity instance occurs.
- $th \in \mathcal{TH}$ , is the email thread in which the activity instance occurs (see Definition 3.6 for threads).

In a traditional business model, the notion of activity instance is directly linked to a process instance. However, in the case of an email-driven process, the notion of process instance is not explicitly apparent, but is rather expressed through the basic structures of an emailing system, namely the emails and email threads.

In fact, conversations in an email log can be grouped into *threads* based on their common *relevant information values* (such as specific  $\mathcal{BD}$ ) and email addresses, in order to approximate the concept of a *trace* (in an event log). In fact, the relevant information values that group conversations into threads approximate BP instance identifiers. These instance identifiers act as links between various events and the sequences of a particular BP execution. An email thread is therefore a set of email conversations that must have in common at least one relevant information value (i.e.,  $\mathcal{BD}$ ) and one interlocutor.

In the following, we denote by  $\mathcal{AD}$  the set of possible values for an email address, by  $\mathcal{TO}$  the set of possible values for an email receiver addresses and by  $\mathcal{O}_{bd}$  the set of possible values of a business data  $bd \in \mathcal{BD}$ .

Let  $f_{RI} : \mathcal{CV} \rightarrow 2^{\mathcal{O}_{\mathcal{BD}}}$  be a function that returns the relevant information values related to a conversation, and let  $f_{interloc} : \mathcal{CV} \rightarrow 2^{\mathcal{AD}}$  be a function that returns the set of interlocutors in a conversation.

**Definition 3.6** (Email Thread). Let  $\mathcal{TH}$  be the set of all email threads. An email thread  $th \in \mathcal{TH}$  is a set of  $n$  conversations  $\{cv_i | \forall i \in [1, n], cv_i \in \mathcal{CV}\}$  such that:  $(\bigcap_{i \in [1, n]} f_{RI}(cv_i)) \neq \emptyset \wedge (\bigcap_{i \in [1, n]} f_{interloc}(cv_i)) \neq \emptyset$

Figure 3.2 illustrates a real example of an email thread retrieved from the Enron dataset. The thread is composed of four emails belonging to two conversations. It reports interactions between employees in the context of a trading gas instance. This instance is identified by the relevant information value '454057' of the associated deal number.

### 3.2.1.3 Definition of Constraint Classes

Constraint classes can be seen as *relations* between different elements of an email where the *domains* and *ranges* can vary depending on the constraint class. In other words, each

constraint class  $C_i$  can be defined as a relation:  $C_i \subseteq X \times Y$  where  $X$  and  $Y$  will be later precised for each  $C_i$ . We will be using the notation  $(ref \xrightarrow{C_i} targ)$  to denote that the element  $ref$  is in relation to the element  $targ$  according to the constraint class  $C_i$ .

**Sequential constraints** These constraints establish a structured and organized flow of activities, enabling effective communication. The following provides an overview of the established sequential constraint classes that we incorporate into the definition of our process model. We note that for these constraint classes the relations are defined as:  $C_i \subseteq \mathcal{A} \times \mathcal{A}$  ( $i \in 1, 2$ ).

- **C1: Activity Sequencing Constraints in Emails** This constraint class expresses restrictions on the order of activities within the same email, focusing on the position of their occurrences and their associated speech acts, which enhances communication efficiency and comprehension among recipients. This class is defined as follows: Let  $\mathcal{A}_{em}$  denote the set of activities in an email  $em$ . The notation  $(act_i \rightarrow act_j)$  where  $act_i, act_j \in \mathcal{A}_{em}$  signifies that the activity  $act_i$  precedes the activity  $act_j$  within the email  $em$ . The constraint  $C_1$  is therefore formulated as follows:  $(act_i \xrightarrow{C_1} act_j)$ .

**Example:** The designer can define a constraint  $(flow\ gas \xrightarrow{C_1} extend\ deal)$  to express that an activity **flow gas** in an email precedes an activity **extend deal** within the same email. This can be highlighted in Email 1, as shown in Figure 3.2.

- **C2: Activity Sequencing Constraints in Threads** These constraints indicate which activities could be mentioned in subsequent emails of a thread as a response, either as activities to be performed in the future or as activities that have already been performed. Such constraints facilitate coherent and contextually relevant responses, streamlining communication and fostering a more organized exchange. Let  $\mathcal{A}_{th}$  be the set of activities in the thread  $th$ , and let  $\mathcal{A}_{th/cv}$  be the set of activities of this thread in the conversation  $cv \in th$ . The constraint  $C_2$  is therefore defined as follows:  $(act_1 \xrightarrow{C_2} act_2)$  with  $act_1, act_2 \in \mathcal{A}_{th}$  such that  $\exists em_1, em_2$  such that  $act_1 \in \mathcal{A}_{em_1} \wedge act_2 \in \mathcal{A}_{em_2} \wedge (em_1 \mathcal{R} em_2)$ .

**Example:** Given the definition of  $C_2$ , a designer might establish that the activity **create deal** is followed by the activity **extend deal**, i.e.,  $(create\ deal \xrightarrow{C_2} extend\ deal)$ . These activities belong to two different emails within the same thread, as exemplified in Email 1 and Email 2 shown in Figure 3.2.

**Contextual constraints** The importance of contextual constraints in conformance checking, especially in the context of emails, cannot be overstated. These constraints play a crucial role in ensuring adherence to the intended purpose of the conversation. In the following, we will define a set of contextual constraint classes that will be utilized in the construction of the process model. These classes will be of varying formats that will be precised for each one.

- **C3: Activity &  $\mathcal{BD}$  relationship constraints** These constraints represent the relationship between the activities and their associated  $\mathcal{BD}$  within threads. They facilitate the identification of incomplete information by delineating the interdependencies between activities and their corresponding business data within threads, ensuring a comprehensive overview of the ongoing processes. This constraints class is defined as a relation  $C_3 \subseteq \mathcal{A} \times 2^{\mathcal{BD}}$  and we note  $(act \xrightarrow{C_3} FB)$  where  $act = \langle AN, \mathcal{BD} \rangle$  and  $FB \subseteq \mathcal{BD}$ .

**Example:** The designer might define the following the constraint:  $(\text{create deal} \xrightarrow{C_3} \{\text{revenue, create, deal, ...}\})$ . In other words, the constraint specifies that the activity `create deal` is always used with a subset of the  $\mathcal{BD}$  specified in the constraint.

- **C4: Activity &  $\mathcal{BD}$  Topic Relationship Constraints**  $\mathcal{BD}$  can be interconnected, forming a cluster of words representing a subject, commonly referred to as a *topic*. The constraints class that we present here represents the correlation between the activities and their most significant  $\mathcal{BD}$  topics within threads, offering a clearer understanding of their primary focus, thereby reducing deviation from the main discussion thread. Let  $\mathcal{TP}$  be the set of all possible topics. This constraint class is therefore defined as a relation  $C_4 \subseteq \mathcal{A} \times 2^{\mathcal{TP} \times \mathbb{R}}$  and expressed using the format:  $(act \xrightarrow{C_4} TP)$  where  $act \in \mathcal{A}$  and  $TP \in 2^{\mathcal{TP} \times \mathbb{R}}$  to denote that an activity  $act$  is linked to a set of topics with their respective probability score (or ranking) according to constraint  $C_4$ .

**Example:** The designer might define the following constraint:  $(\text{extend deal} \xrightarrow{C_4} \{\text{Business Growth}, 0.8\})$ . This means that the activity "extend deal" should always be accompanied by a business data related to the topic `Business Growth` with a score of 0.8. Here, the score reflects the probability of the relevance of the business data to the topic of `Business Growth`, with a score of 0.8 indicating a high likelihood of correlation.

- **C5: Activity &  $\mathcal{BD}$  frequency relationship Constraints** These constraints represent the correlation between an activity mentioned in an email and the presence of its related  $\mathcal{BD}$ . This constraint class is therefore defined as a relation  $C_5 \subseteq \mathcal{A} \times 2^{\mathcal{BD} \times \mathbb{R}}$  and expressed using the format:  $(act \xrightarrow{C_5} BDF)$  where  $act \in \mathcal{A}$  and  $BDF \in 2^{\mathcal{BD} \times \mathbb{R}}$ .

**Example:** The designer might define the following constraint:  $(\text{flow gas} \xrightarrow{C_5} \{('flow', 0.84), ('meter', 0.71)\})$ .

- **C6: Activity &  $\mathcal{BD}$  Pairs Relation Constraints** These constraints represent connections between activities and business data within threads. This constraint class is defined as a relation  $C_6 \subseteq \mathcal{A}^2 \times 2^{\mathcal{BD}}$  and expressed using the format:  $((act_1, act_2) \xrightarrow{C_6} FD)$  where  $act_1 = \langle AN_1, BD_1 \rangle, act_2 = \langle AN_2, BD_2 \rangle$  such that  $(act_1, act_2) \in C_1 \cup C_2$  and  $FD \subseteq BD_1 \cup BD_2$ .

**Example:** The designer might define the following constraint:  $((\text{flow deal}, \text{extend deal}) \xrightarrow{C_6} FD)$  where  $FD = \text{flow deal}.BD \cup \text{extend deal}.BD$ . In this example, the constraint captures a semantic relationship between the activities `flow deal` and `extend deal`, indicating that they are expected to be closely associated in terms of business data content. The specific terms in the example, such as `gas`, `system`, `price` serve as illustrative examples of the kind of business data that might be expected in the associated sets. The constraint is generic and implies a high likelihood of co-occurrence of specific business data when these two activities are observed in sequence.

Having defined both the sequential and contextual constraints classes, we can now represent the process model to be fed to the conformance checking algorithm. The process model is represented as directed graph to accurately capture the process behavior from two perspectives. In the graph representation, each directed edge connecting a parent and a child node corresponds to a reference event and a target event with the same sequential and contextual constraint. This approach ensures activities that can be taken as responses to a specific event are linked to the same reference/parent node. It allows us to establish the expected order, dependencies, and context between activities based on their sequential and contextual constraints. In the following definition, we denote by  $\mathcal{A}_C$  the set of all activities used in the user-defined constraints.

**Definition 3.7** (Email Process Model). *A Email Process Model is a directed graph  $EPM = (V, E)$  where  $V \subseteq (\mathcal{A} \times 2^{\mathcal{BD}} \times 2^{\mathcal{TP}})$  and  $E \subseteq (V \times V)$ :*

- $V = \{v_{act} | v_{act} = (act, \mathcal{BD}, TP) \forall act \in \mathcal{A}_C\}$  with:
  - $\mathcal{BD} = (\bigcap_{act_t \in \mathcal{A}_C} C_6((act, act_t)) \cap C_3(act) \cap C_5(act))$
  - $TP = \{tp | \exists r \in \mathbb{R} \wedge (tp, r) \in C_4(act)\}$
- $E = \{e_i | e_i = (v_{act_1}, v_{act_2}), \forall v_{act_1}, v_{act_2} \in V \text{ such that } (act_1, act_2) \in C_1 \cup C_2\}$

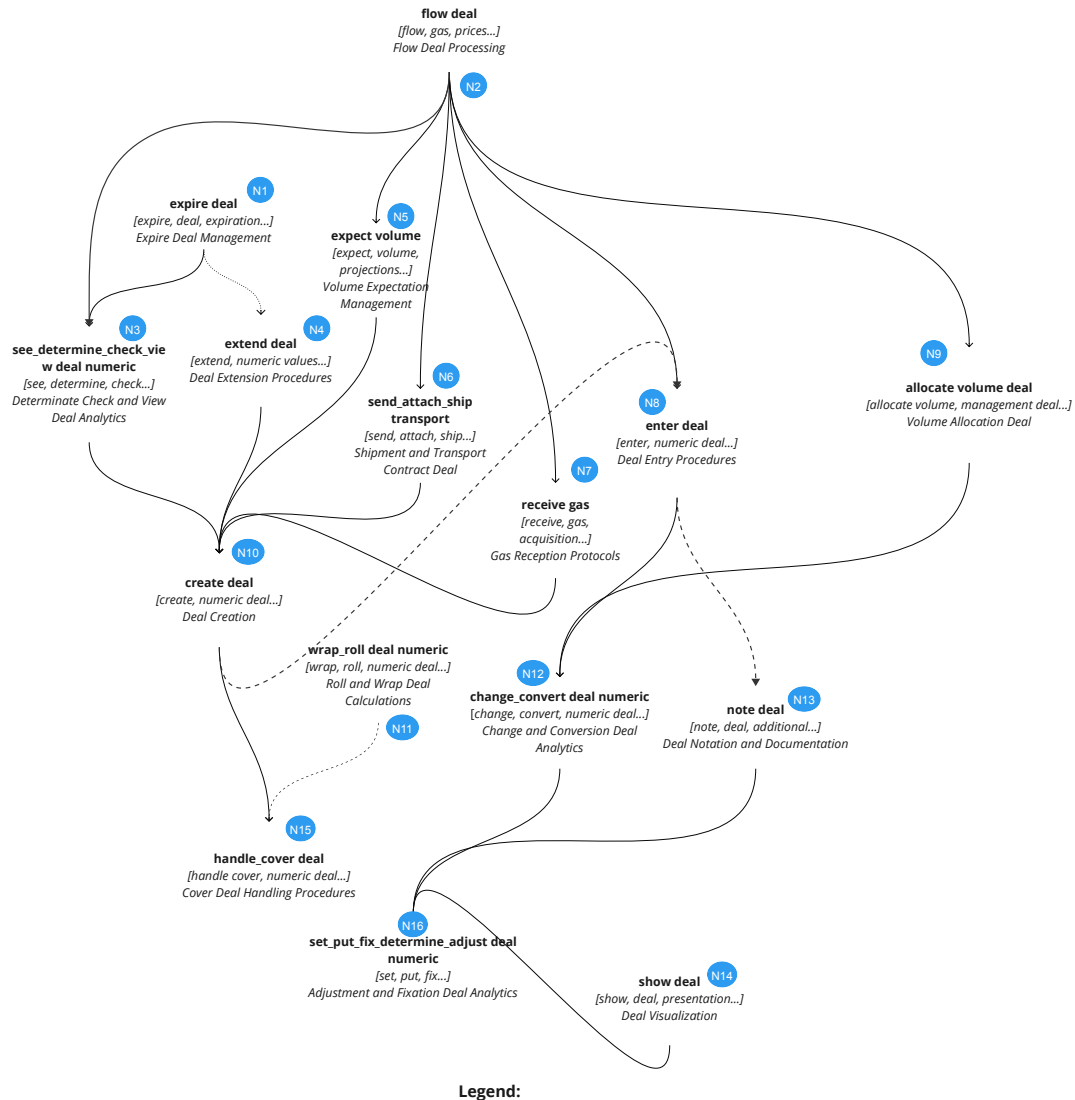


Figure 3.3: A partial view of the Email Process model

Figure 3.3 depicts a partial view of the *Email Process Model*'s directed graph. This

partial view focuses on activities related to managing gas deals, such as creating or receiving a deal. In the provided Email Process Model fragment, the node labeled **N10** highlights the defined constraints as follows: the activity **create deal** represents *act* in bold, the business data *[create, numeric deal...]* corresponds to **BD**, and the topic name *Deal Creation* corresponds to *TP*. These elements form the  $v_{act} = (act, \mathbf{BD}, TP)$ . An edge example is the directed connection from **N10 (create deal)** to **N11 (wrap, roll deal numeric)**, representing  $e_i = (v_{act1}, v_{act2})$ , where  $(act_1, act_2) \in C_1 \cup C_2$ . This edge indicates a relationship between the activities based on conditions specified in the definition.

### 3.2.1.4 Email-based Process Log

Now that we have presented our proposed Email Process Model, we shift our focus onto the second input of our conformance checking approach, namely the structured event log. The latter will be used to compare against the defined process model, enabling the algorithm to identify deviations and ensure that the processes adhere to expected standards and workflows. Elements such as activities and business data, previously defined, can be found as attributes within an event in the event log. This structured event log can be obtained using the work of Elleuch et al. [43] and defined as in Definition 3.8. We note that this definition for a structured event log could easily be integrated into existing standards, such as Object-centric event logs (OCEL) [56], as an extension that would allow the representation of additional email-specific concepts included in our definition.

**Definition 3.8** (Event Log). *Let  $\mathcal{E}$  be the set of all events. An event log is a set of events:  $Log = \{Ev_i, \forall i \in [1, n]\}$  where each event  $Ev_i$  is characterized by the tuple  $\langle Act_o, SA, At_{ind}, I_{values}, em, Th_{id} \rangle$ , where:*

- $Ev_{ID}$  is the event's unique identifier.
- $Act_o \in \mathcal{O}_A$  is the occurred activity.
- $SA \in \mathcal{SA}$  is the speech act of  $Act_o$ .
- $At_{ind}$  is the set of textual indices concerning the performers of  $Act_o$ .
- $I_{values} \subset Act_o.\mathbf{BD}_{occ}$  is the set of related relevant information values.
- $em \in \mathcal{EM}$  is the email where the activity occurred.
- $Th_{id}$  is the set of thread IDs to which the event belongs.

Event	Act <sub>o</sub>			SA	At <sub>Ind</sub>	I <sub>values</sub>	em					Th <sub>id</sub>
Ev_ID	AN <sub>occ</sub>	BD <sub>occ</sub>	BC <sub>occ</sub>				ID	times tamp	Sender	Recipients	Con vids	
e <sub>1</sub>	{'flow deal', 'flowed', 'deal', [7,4]}	{{('deal numeric', ['deal', '412219'], [4,5]), (('meter numeric', ['meter', '1601'], [2,3]), ('deal numeric', ['deal', '45057'], [11,12]), (('meter numeric', ['meter', '5192'], [9,10]))	{{('meter deal', ['meter', deal'], [2,4])}	'information'	[]	{'deal numeric_deal 412219'; 'meter numeric_mete r 1601'; 'deal numeric_deal 45057'; 'meter numeric_mete r5192'}	<'155258 9.107585 3972210. JavaMail .evans@ thyme>	'Fri, 10 Nov 2000 06:17: 00 - 0800 (PST)'	'aimee.la nnou@en ron.com'	['daren.farmer @enron.com']		
e <sub>2</sub>	{'flow deal', 'flowed', 'deal', [14,11]}		{}	'information'	[]						'c1'	[Th <sub>1</sub> ]
e <sub>3</sub>	{'roll deal', ['rolled', 'deal'], [2,3]}	{{('deal numeric', ['deal', '45057'], [3,4]), (('meter numeric', ['mtr', '5192'], [7,8]))	{}	'information'	[]	{'deal numeric_deal 45057'; 'meter numeric_mete r5192'}	<'297175 36.10758 5415038 8.JavaM ail.evans @thyme >	'Wed, 15 Nov 2000 02:24: 00 - 0800 (PST)'	'daren.far mer@enr on.com'	['aimee.lanno u@enron.com ']		
e <sub>4</sub>	{'cover flow', 'cover', 'flow'], [3,4]}			'intention'	[]							
e <sub>5</sub>	{'open short position', ['short', '100', 'mws'], [1, 2, 3]}			'information'	['we']							
e <sub>6</sub>	{'display deal', 'shows', '100', 'mws'], [27, 28, 29]}	{{('numeric mw', ['numeric', '100'], [2,3]), (('hourDend numeric numeric', ['hour ending', '7', '22'], [5,6]), ('orgname', ['Ercot Asset'], [4]), (('orgname', ['El Paso'], [20]), (('orgname', ['Empower'], [26]), ('numeric mw', ['numeric', '100'], [28,29]), (('orgname', ['Ercot'], [30]), (('orgname', ['bal-day'], [38]))		'information'	['enpower']							
e <sub>7</sub>	{'cut deal', ['cut', 'deal'], [33, 31]}		{}	'intention'	[]		<'235351 50.10758 5236920 0.JavaM ail.evans @thyme >	'Wed, 26 Sep 2001 10:19: 15 - 0700 (PDT)'	'm..forne y@enron. com'	['joe.capasso @enron.com', 'l.day@enron .com', 'joe.errigo@e nron.com', 'alexander.mc elreath@enro n.com', 'jeffrey.miller @enron.com', 'steve.olinde @enron.com', 'eric.saibi@en ron.com']	'c3'	[Th <sub>3</sub> ]
e <sub>8</sub>	{'replace deal', 'replace', 'deal', [34, 31]}			'intention'	[]							
e <sub>9</sub>	{'purchase power', 'power', 'purchased'], [36, 35]}			'information'	[]							

Figure 3.4: Example of an Event Log Extract

Figure 3.4 depicts an excerpt of the event log that would be generated from an email log including  $email_1$  and  $email_2$  shown in Figure 3.2 and the email shown in Figure 3.5.

Message-ID: <23535150.1075852369200.JavaMail.evans@thyme>  
Date: Wed, 26 Sep 2001 10:19:15 -0700 (PDT)  
From: m..forney@enron.com  
To: joe.capasso@enron.com, l.day@enron.com, joe.errigo@enron.com,  
alexander.mcelreath@enron.com, jeffrey.miller@enron.com,  
steve.olinde@enron.com, eric.saibi@enron.com  
cc:  
Subject: Position for tomorrow

We are short 100 mw's in the Ercot Asset book for hours ending 7-22.  
We want to remain that way, unless the balancing energy market goes  
haywire. We would need to delete our SC counterparty EL Paso and input  
a replacement QSE in the portal if this is filled.  
Empower shows 100 mw's coming from Ercot imbalance.  
This deal would need to be cut and replaced with the power that was  
purchased for bal-day, etc.

Thanks, JMF

Figure 3.5: Real email retrieved from Enron dataset for planning trading positions

In the work of Elleuch et al. [41] the authors utilized unsupervised learning techniques to discover activities within email bodies. The approach involved identifying frequent patterns of words shared by activity expressions, even when variations such as synonymous words



(e.g., the expressions '*change deal*' and '*convert deal*' refer to the same activity while using the synonymous words: '*change*' and '*convert*') or different word orders (e.g., the positions of appearance of the words '*extend*' and '*deal*' are switched in these expressions '*the deal was extended*' and '*I extended the gas deal*'). Additionally, the two words are nearly successive in the first expression while separated by the word '*gas*' in the second expression) are present. By analyzing emails on a per-employee basis, common patterns related to activity components are captured. Subsequently, similar activities across different employees are grouped together based on measures of word synonymity and activity context.

### 3.2.2 The Conformance Checking Algorithm

In this section, we focus on the algorithm designed to ensure smooth alignment between the execution of a process instance and the process model. This algorithm is crucial in evaluating the degree to which events conform to the anticipated behavior within threads or emails. In our case, the execution of a process instance can be seen as a trace from the event log, which is a sequence of events.

The algorithm shown in Algorithm 1 takes as inputs a sequence of events and the *email process model* (EPM), returning a comprehensive list of fulfilling and/or violating events. Additionally, it provides information on any missing topics for each event, if applicable. The algorithm begins by setting up a current node based on the starting point of the event sequence (line 1). It then visits each event in the sequence, evaluating the similarity between the **BD** of the current node in the EPM and its occurrence in the event of the sequence by employing cosine similarity (line 13). Upon calculating the similarity and identifying a match between the next activity in the sequence of events and one of the activities of the next nodes in the *email process model*, the algorithm determines the type of the event by calling the *detect\_event\_type* function (line 14). The *detect\_event\_type* function (detailed in Algorithm 2) takes the next activity, similarity, current node, and event as inputs, and it returns the event type and any missing topics. If the similarity score is above a specified threshold and the next activity is valid, the event is classified as fulfilling and appended to a set of fulfilling events. Conversely, if the similarity score is below the threshold or there is no match found for the next activity, the event is added to a set of violating events (lines 15  $\rightarrow$  19, Algorithm 1).

We have identified three types of violating events that warrant our attention to enhance communication effectiveness. The first type is *Sequential Flow Issues*, detected when the algorithm identifies a mismatch between the expected and actual subsequent activities in the sequence of events. Specifically, the algorithm checks if the *next activity* aligns with the expected activity, and if it does not, the event is classified as a sequential flow issue (line 6 in *detect\_event\_type*). The second type is *Incomplete/Incorrect Information*, where the algorithm detects instances in which essential details are missing or incorrect. This involves comparing the **BD** of the current event with those of the expected event (line 8 in *detect\_event\_type*). The third type is *Deviation from the Main Topic*, aiming to identify instances where the conversation veers off course, potentially leading to miscommunication. This is determined by examining if the **BD** topics of the event contains elements unrelated to

the expected topics derived from the current node (line 10 in *detect\_event\_type*). Additionally, the Algorithm 1 identifies missing topics by comparing the detected topics of the current event with the expected topics in the current node from the *email process model* (line 20 → 22).

---

**Algorithm 1** Conformance Checking
 

---

**INPUT:**  $S, EPM$ 
**OUTPUT:**  $fulfilling\_events \leftarrow \emptyset, violating\_events \leftarrow \emptyset, missing\_topics\_list$ 

```

1:  $current\_node \leftarrow findStartingPoint(EPM, S[0].Act_o)$  {Find the starting point in the
   EPM based on the first event's activity}
2:  $fulfilling\_events \leftarrow []$  {Initialize the list of fulfilling events}
3:  $violating\_events \leftarrow []$  {Initialize the list of violating events}
4:  $missing\_topics\_list \leftarrow []$  {Initialize the list of missing topics}
5: for  $i \leftarrow 0$  to  $len(S) - 1$  do {Iterate over each event in the sequence}
6:    $event \leftarrow S[i]$  {Get the current event}
7:    $next\_nodes \leftarrow successors(current\_node)$  {Find successors of the current node in
   EPM}
8:   if  $i + 1 < len(S)$  then {Check if there is a next event in the sequence}
9:      $next\_activity \leftarrow findOccurrences(S[i + 1].Act_o, next\_nodes)$  {Find occurrences of
   the next activity in successors}
10:  else
11:     $next\_activity \leftarrow null$  {If no next event, set next activity to null}
12:  end if
13:   $similarity \leftarrow CosineSimilarity(BD[current\_node], BD[event])$  {Compute the similar-
   ity between current node and event based on their BD vectors}
14:   $(event\_type, missing\_topics) \leftarrow detect\_event\_type(next\_activity,$ 
    $similarity, current\_node, event)$  {Detect the type of the event and compute missing
   topics}
15:  if  $event\_type = 'fulfilling'$  then
16:    Append  $event$  to  $fulfilling\_events$  {If event is fulfilling, add it to fulfilling events}
17:  else
18:    Append  $(event, event\_type)$  to  $violating\_events$  {If event is violating, add it to
   violating events with the reason}
19:  end if
20:   $detected\_topics \leftarrow detect\_topics(event)$  {Call function to detect topics of the event}
21:   $missing\_topics \leftarrow expected\_topics \setminus detected\_topics$  {Find missing topics by sub-
   tracting detected topics from expected topics}
22:  Append  $(event, missing\_topics)$  to  $missing\_topics\_list$  {Add event and missing
   topics to missing topics list}
23:   $current\_node \leftarrow next\_activity$  {Move to the next activity for the next iteration}
24: end for
25: return  $fulfilling\_events, violating\_events, missing\_topics\_list$  {Return the lists of
   fulfilling events, violating events, and missing topics}

```

---

**Algorithm 2** Detect Event Type**INPUT:** *next\_activity*, *similarity*, *current\_node*, *event***OUTPUT:** *event\_type*, *missing\_topics*


---

```

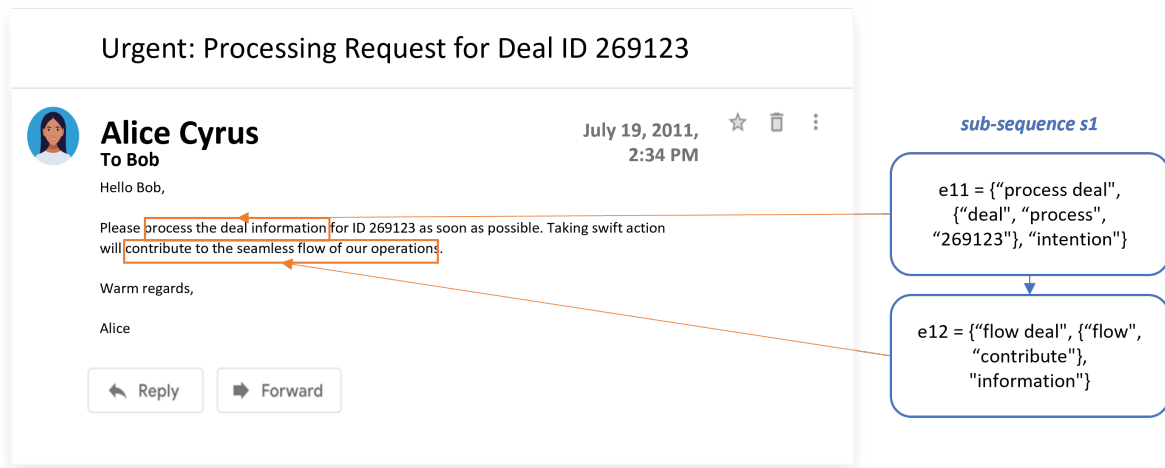
1: missing_topics  $\leftarrow$  BD_topics(current_node) – BD_topics(event) {Compute missing
   topics by subtracting event topics from current node topics}
2: if next_activity  $\neq$  null and similarity > threshold then {Check if next activity is valid
   and similarity is above threshold}
3:   event_type  $\leftarrow$  'fulfilling' {Event is fulfilling if conditions are met}
4: else
5:   if next_activity = null then {Check if next activity is null}
6:     event_type  $\leftarrow$  'Sequential Flow Issue' {Event is violating due to 'Sequential Flow
   Issue'}
7:   else if missing_topics  $\neq$   $\emptyset$  then {Check if there are missing topics}
8:     event_type  $\leftarrow$  'Deviation from the Main Topic' {Event is violating due to 'Incorrect
   Topic'}
9:   else
10:    event_type  $\leftarrow$  'Incomplete/Incorrect Information' {Event is violating due to Incom-
   plete/Incorrect Information}
11:  end if
12: end if
13: return event_type, missing_topics {Return the event type and missing topics}

```

---

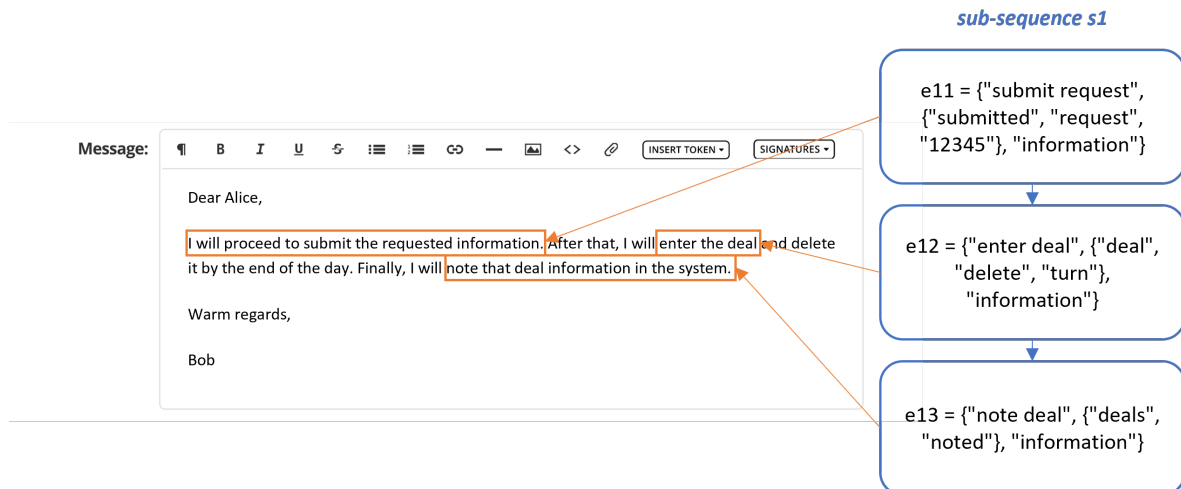
### 3.3 Proof of Concept

This section explores the practical application of our newly proposed methodology, providing insight into its real-world implementation. Our analysis focuses on the assistance offered to Bob, who faces a situation requiring a prompt response to an email from his superior, Alice. In Alice's email to Bob shown in Figure 3.6, she outlines two key actions she wants Bob to take. Initially, she requests that Bob processes the deal information, an action referred to as event  $e_{11}$ , described as (“**process deal**”, [“**process**”, “**deal**”, “**intention**”). Following that, she emphasizes the importance of taking swift action to ensure the seamless flow of operations, represented by event  $e_{22}$ , defined as (“**flow deal** ”, [“**flow**”, “**operations**”, “**intention**”).



**Figure 3.6:** Urgent Email: Alice’s Notification Regarding Deal Priority

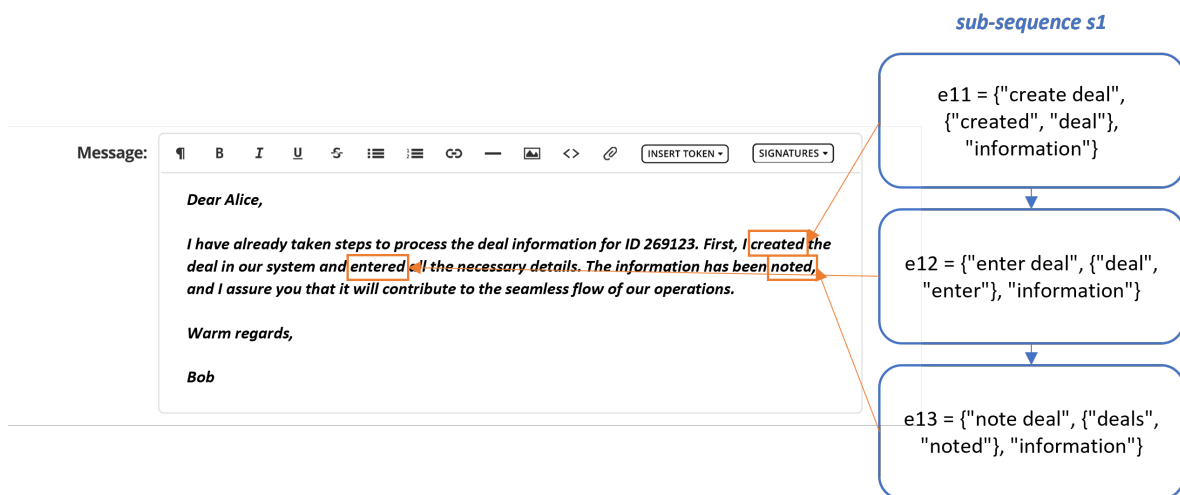
Bob has proactively drafted two potential email responses but grapples with uncertainty about the most suitable option for the given scenario. To gain clarity, let’s delve deeper into the drafted emails.



**Figure 3.7:** Bob’s First Email Response

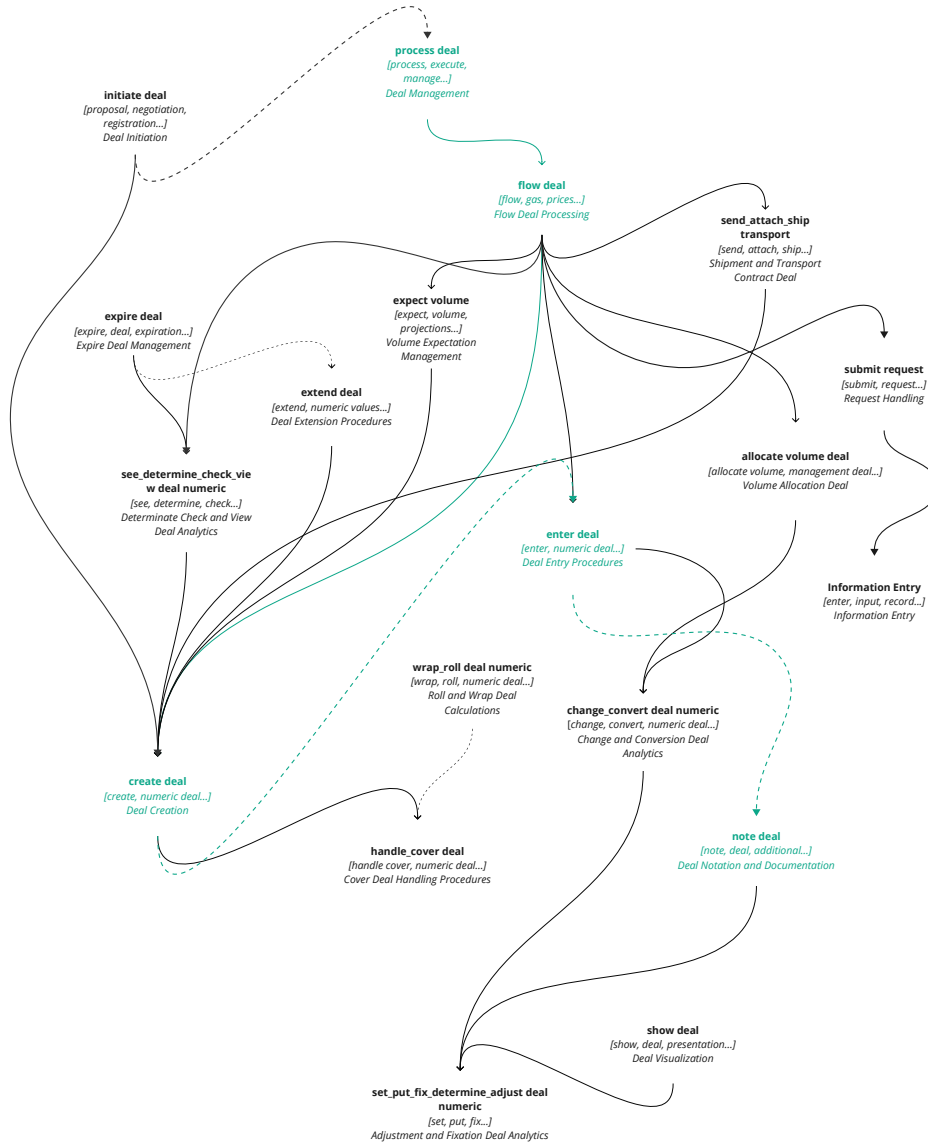
In Bob’s *initial email*, as depicted in Figure 3.7, he outlines a series of steps he plans to take. Initially, he will provide the requested information—an action referred to as event  $e_{11}$ , described as (“**submit request**”, [‘submitted’, ‘request’, ‘12345’], “**information**”). Following that, Bob intends to finalize the deal, a progression represented by event  $e_{12}$ , defined as (“**enter deal**”, [‘deal’, ‘complete’, ‘process’], “**information**”). He commits to ensuring its deletion by the end of the day. Finally, he notifies Alice that he will record the relevant deal information in the system. This process is encapsulated by event  $e_{13}$ , characterized as (“**note deal**”, [‘deals’, ‘recorded’], “**information**”).

In the *second email*, as depicted in Figure 3.8, Bob informs Alice that he has processed the deal information for ID 269123. This was achieved through a series of sequential steps. Initially, he initiated the creation of the deal within their system, as indicated by the event  $e_{11}$ , described as (“**create deal**”, [‘**created**’, ‘**deal**’], “**information**”). Subsequently, he inputted all the necessary details, a process denoted by the event  $e_{12}$ , defined as (“**enter deal**”, [‘**deal**’, ‘**enter**’], “**information**”). Finally, he recorded the completion of the deal within their system, an action represented by the event  $e_{13}$ , defined as (“**note deal**”, [‘**deals**’, ‘**noted**’], “**information**”). Bob assures Alice that these actions will contribute to the seamless flow of their operations.



**Figure 3.8:** Bob’s Second Email Response

Our goal is to identify the most suitable response from the provided emails to Alice’s initial email, utilizing conformance checking. To conduct conformance checking effectively, we require access to two critical components: the real-time process execution instance (in our case, the sequence of events illustrating the email exchange between Alice and Bob) and the *Email Process Model* depicting the expected behaviors of a process within a thread.



### Legend:

- **Activity Name (act):** Activities are displayed in bold. ( $\mathcal{A}_C$ )
- **Business Data (BD):** Business data are presented in italics and within [array brackets]. ( $BD = (\bigcap_{act_i \in \mathcal{A}_C} C_6((act, act_t)) \cap C_3(act) \cap C_5(act))$ )
- **Topic Name (TP):** Topic names are also presented in italics. ( $TP = \{tp \mid \exists r \in \mathbb{R} \wedge (tp, r) \in C_4(act)\}$ )
- **Edge (e):** Directed edges between activities are shown as arrows. ( $E = \{e_i \mid e_i = (v_{act_1}, v_{act_2}), \forall v_{act_1}, v_{act_2} \in V \text{ such that } (act_1, act_2) \in C_1 \cup C_2\}$ )
  - **Dot Line ( $C_1$ ):** Edges with a dot line represent connections mostly appearing in the constraint  $C_1$ .
  - **Disconnected Dot Line ( $C_2$ ):** Edges with a disconnected dot line represent connections mostly appearing in the constraint  $C_2$ .

**Figure 3.9:** A partial view of the *Email Process Model* illustrating Gas Deal Management Processes

Figure 3.9 illustrates a partial view of the *Email Process Model*, focusing on processes related to gas deal management, the central topic of Alice’s correspondence. The trace, representing the anticipated exchange between Alice and Bob, is highlighted in green. With the knowledge that the sub-sequence of events in Alice’s email conforms to the *Email Process Model*, and having both the sub-sequences of events associated with Bob’s drafted responses, our next step involves verifying the adherence of these sub-sequences using our proposed algorithm.

**Examining Bob’s First Email Response** In this section, we will assess Bob’s first email response using our conformance checking algorithm to analyze the sequence of events within the email interactions between Alice and Bob. The algorithm begins by identifying a suitable starting node within the graph that corresponds to the initial event in the sequence, which, in this case, is *submit request*. Once the matching node is determined, the algorithm proceeds with a systematic classification process, considering both the sequential and contextual perspectives of events. Starting with the *submit request* event, the algorithm verifies whether subsequent activities align with the expected activities. According to the *Email Process Model*, the event following *submit request* should be *send information*, not *enter deal*. To assess the second perspective of the event, we examine the **BD** associated with the events. Here, we check how closely the **BD** of the current event aligns with those associated with the expected event. In this case, although the similarity is above the threshold (0.8), the next activity is incorrect, resulting in the classification of the event as a **violating event**, specifically a **sequential issue type**. Continuing the analysis of the remaining events in the sequence, the algorithm follows the same procedures outlined in the conformance checking algorithm. Upon scrutinizing each event, the algorithm detects another instance of violation within Bob’s sub-sequence, specifically the events labeled as *enter deal*. These violations arise from the usage of unrelated terms within the associated **BD**, such as *delete*, introducing an action that was not explicitly requested or mentioned by Alice in her initial email. This discrepancy is classified as **Incomplete/Incorrect Information**. Alice’s communication focused solely on processing deal information for **ID 269123**, without any indication of deleting the deal. Therefore, assuming the action of deletion without additional context or clarification from Alice may confuse or concern her. Consequently, since 2 out of 3 events were classified as **violating events**, this email response is deemed inadequate due to a lack of coherence.

**Examining Bob’s Second Email Response** Through the algorithmic analysis, Bob’s actions are categorized as *fulfilling*, indicating alignment with expected events. Bob initiates by addressing the creation of the deal, which correlates with the *create deal* event. Subsequently, he delves into specifics about the deal, aligning with the *enter deal* event. Finally, Bob notes down the deal’s information, corresponding to the *note deal* event. These actions mirror the expected sequence of events, highlighting Bob’s meticulous approach in addressing Alice’s concerns. Consequently, as all events were classified as *fulfilling events*, this email response is considered adequate. Bob’s clear and sequential communication underscores his comprehension of both the urgency and importance of Alice’s email, demonstrating his intention to keep her well-informed. By detailing each stage of processing the deal’s details, Bob

effectively communicates to Alice that he has handled her concerns with great care.

When we applied conformance checking to Bob’s email responses, distinct results emerged. The first email exhibited multiple discrepancies, diverging from the expected sequence and introducing unrelated information. In contrast, the second email perfectly aligned with Alice’s specifications. This discrepancy highlights the importance of our methodology, as it aids in pinpointing communication flaws and guiding individuals to craft responses in line with the original sender’s expectations.

## 3.4 Experiments and Validation

Introducing a novel contribution that has remained unexplored until now presents a challenge in comparing our work to existing related research. This challenge primarily arises due to the distinctive structure of the email event log in our case, which deviates from the conventional format commonly studied in the literature.

To address this challenge and provide a comprehensive assessment, we present in Section 3.4.1 a detailed analysis of our approach’s capability to detect non-conformance within email-driven processes. In Section 3.4.2, we demonstrate the practical application and benefits of our proposed approach with a use case involving the utilization of a RESTful API endpoint. The approach, developed in Python, is readily available on GitHub <sup>1</sup>, where some of the experimental results can also be found.

### 3.4.1 Detection of Non-Conformance in Enron Email Logs

In typical conformance checking, algorithms handle conventional processes with nearly 100% accuracy, meaning that they can almost perfectly determine whether an event conforms or not. However, in email-based processes, the unstructured nature of emails and the variability of business data across different emails make it challenging to consistently achieve such high accuracy. To effectively demonstrate the performance of our approach, we resorted to an expert who can provide insights and validation. In the following, we conducted an experiment using two event logs extracted from the Enron dataset, which consists of internal emails related to the company’s business operations. The first event log focuses on financial transactions, budgets, and forecasts, while the second log centers around discussions related to the energy industry and market trends. We first applied our proposed algorithm to identify violating and fulfilling event types within emails or threads in each log. Subsequently, we asked experts to classify the events manually. We then compared the non-conformance types detected by our approach with those identified by the experts to assess the accuracy and reliability of our method. This comparison is crucial for validating our algorithm’s effectiveness in handling email-based processes. The results of our algorithm’s classifications and the

---

<sup>1</sup><https://github.com/ralphbn1995/Multi-Perspective-Conformance-Checking-For-Email-driven-Processes.git>



expert assessments are presented in Table 3.1. Additionally, Table 3.2 presents the calculated precision and recall for each type of non-conformance detected, based on the combined results from the two event logs extracted from the Enron dataset.

**Table 3.1:** Overview of the results

Non-Conformity Type	Approach Classification		Expert Classification	
	First Log	Second Log	First Log	Second Log
Sequential Flow Issue	40 Events	39 Events	50 Events	45 Events
Incomplete Information	70 Events	58 Events	75 Events	65 Events
Deviation from Main Topic	55 Events	38 Events	60 Events	45 Events

**Table 3.2:** Precision and Recall Metrics for Non-Conformance Types Detected in Enron Event Logs

Non-Conformance Type	Precision	Recall
Sequential Flow Issues	0.88	0.92
Incomplete Information	0.91	0.87
Deviations from Main Topic	0.83	0.81

In terms of Sequential Flow Issues, the expert classifications identified 50 events in the first log and 45 events in the second log, while the algorithm detected 40 events in the first log and 39 in the second log, highlighting the algorithm’s proficiency in handling sequencing challenges with a precision of 0.88 and a recall of 0.92, indicating its effectiveness in ensuring the correct order of events, crucial for maintaining the logical flow of communication. For Incomplete Information, experts classified 75 events in the first log and 65 events in the second log, while the algorithm identified 70 and 58 events, respectively, demonstrating its capability to detect missing details that could hinder effective communication; the precision of 0.91 and recall of 0.87 underscore the algorithm’s ability to identify and address gaps in information, enhancing the completeness and clarity of communications. Regarding Deviation from Main Topic, expert classifications recorded 60 events in the first log and 45 in the second log, while the algorithm found 55 events in the first log and 38 in the second log, suggesting the algorithm’s effectiveness in identifying when conversations stray from the main topic, as reflected by a precision of 0.83 and recall of 0.81, thus helping to maintain focus and relevance in the communication.

Overall, the results suggest that the proposed algorithm adeptly identifies non-conformities within both event logs, closely aligning with expert classifications. This alignment underscores the algorithm’s potential to enhance communication effectiveness by systematically addressing specific non-conformance types, thereby contributing to the overall quality and coherence of communication within the Enron dataset’s email event logs.

### 3.4.2 Use Case Study

This section presents a use case of the proposed approach in this chapter through the utilization of RESTful API [86] endpoint. Our RESTful API endpoint is meticulously designed

with a dual-focus approach involving two key decisions. Firstly, we exclusively employ Hypertext Transfer Protocol (HTTP) [51] POST requests for all endpoint interactions. This ensures secure data exchange and facilitates the confidential transmission of large data payloads. Secondly, we have standardized on JSON as the data interchange format due to its lightweight structure and human-readable nature, which promotes seamless communication across diverse systems, languages, and platforms. The */email-compliance-verification* endpoint verifies the logical flow of ideas within an email. Users submit POST requests with an email as input, and the approach developed in this chapter examines the content to ensure coherence and logical progression of thoughts. The response, provided in JSON format, indicates adherence to established logical patterns, thereby aiding in maintaining communication quality and coherence.

Thirty-three participants from diverse backgrounds, including data scientists, software engineering students, Ph.D. students, and developers, were involved. All participants had a solid understanding of process and data analysis, with varying levels of familiarity with REST APIs, ranging from extensive knowledge to basic understanding. Participants received identical email samples and were tasked with manually composing email replies. They were then instructed to submit their drafts through the */email-compliance-verification* endpoint to validate the logical coherence of ideas within their manually composed email responses.

On average, the API flagged  $I_{avg}$  issues per email, with  $I_{avg}$  representing the average number of issues identified. These issues ranged from abrupt topic transitions to vague argumentation lines. The precise feedback provided empowered participants to promptly identify areas requiring improvement. Upon receiving feedback from the API, participants demonstrated swift responsiveness. The majority managed to revise and enhance their email drafts within an average time of 3 minutes. This indicates that the API effectively expedites the revision process, showcasing its practical utility.

To verify the improvement in the speed of the revision process achieved by incorporating the */email-compliance-verification* endpoint compared to the traditional manual revision process, we calculated the efficiency gain (%EG). The formula for calculating the efficiency gain is given by:

$$\%EG = \frac{R_{\text{manual}} - R_{\text{API}}}{R_{\text{manual}}} \times 100\%$$

Here,  $R_{\text{manual}}$  represents the average time taken for manual revisions, and  $R_{\text{API}}$  represents the average time taken for revisions assisted by the API.

The efficiency gain is calculated by taking the difference between the average manual revision time and the average API-assisted revision time, dividing it by the average manual revision time, and then expressing this as a percentage. A positive %EG indicates that the API-assisted revisions were faster than manual revisions, reflecting a gain in efficiency. A higher %EG suggests a greater improvement in efficiency due to the API.

The table labeled as Table 3.3 provides a summary of the results gathered from that use

**Table 3.3:** Impact of API-Aided Revisions on Email Draft Quality and Efficiency

Participants	Number of Issues Identified ( $I_{avg}$ )	Average Revision Time (min)	Revision Efficiency (%EG)	Gain
33	3.5	3	15%	

case, including several key attributes: the number of participants representing the individuals involved in the study; the average number of issues identified per email, indicating the typical count of logical or coherence problems detected; the average revision time reflecting the mean time taken by participants to revise their emails after receiving API feedback; and the efficiency gain. Participants utilized the API for self-assessment and quality assurance, resulting in an average identification of 3.5 issues per email, ranging from abrupt topic transitions to vague argumentation lines. Remarkably, the average revision time of 3 minutes showcases participants' swift responsiveness to the API's feedback, signifying its role in expediting the revision process. The efficiency gain (%EG) of 15% further emphasizes the API's contribution to a more efficient revision workflow. These results collectively illustrate that the */email-compliance-verification* endpoint is instrumental in not only identifying and addressing issues promptly but also in elevating the overall quality and efficiency of manual email composition.

### 3.5 Conclusion

In this chapter, we achieved our first objective and answered the second research question (Q1) raised in the thesis problematic, as detailed in Chapter 1, Section 1.3.1. The central question explored was: **How well do current conformance checking techniques perform when applied to email-based business processes?** To address this multi-faceted question, we considered several sub-questions:

- **Q1-1:** How can we design a conformance checking method that takes into account both the structural and contextual perspectives of the email events?
- **Q1-2:** What are the common patterns of discrepancies that can occur in the business context of email events, and how can they be accurately detected?
- **Q1-3:** How can we handle non-categorical or non-numerical attribute values in conformance checking methods, particularly those related to email events?
- **Q1-4:** How can we validate and measure the accuracy, reliability, and performance of the newly proposed context-aware conformance checking method?

In response to these questions, we have devised an approach for multi-perspective conformance checking within the context of email communication, consisting of two main phases:

1. *Model Construction Phase:* We constructed an *Email Process Model* for conformance checking of events within individual emails and threads based on the constraints defined by the experts, which address questions **Q1-1** and **Q1-2**.
2. *Conformance Checking Phase:* We introduced a conformance checking algorithm to answer question **Q1-3**, which entails comparing the execution of a process instance (i.e., an event log instance) with one or both of the process models. This comparison aids in identifying two sets of events referred to as *fulfilling* and *violating* events.

We conducted experiments to answer question **Q1-4** using a public dataset from Enron, presenting promising results. Furthermore, we have openly shared our findings, a contribution absent in related studies, rendering direct comparisons unfeasible. Nevertheless, we acknowledge potential limitations at two levels:

1. *Enhancing User Interaction:* Our current model lacks support for interactive feedback, which could prove invaluable for users seeking to comprehend discrepancies in real-time. Incorporating an interactive dashboard or visualization could enhance the intuitive understanding of results.
2. *Expanding Universality:* Given the global nature of business, emails in non-English languages may be prevalent. Our method presently does not accommodate multilingual content, potentially limiting its universality.

In the upcoming chapter, we will introduce our proposed process-activity-aware email response recommendation approach.

# Predictive Process Approach for E-mail Response Recommendations

---

## Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>63</b>
<b>4.2</b>	<b>The Proposed Approach Overview</b>	<b>64</b>
4.2.1	First segment: BP Knowledge Extraction and Prediction	66
4.2.2	Second segment: Response Template Recommendation	72
<b>4.3</b>	<b>Experiments and Validation</b>	<b>81</b>
4.3.1	Precision of Predicted BP Knowledge in Email Responses	81
4.3.2	Effectiveness and Coherence of the Textual Content in Predicted Emails	83
4.3.3	Use Case Study	84
<b>4.4</b>	<b>Conclusion</b>	<b>86</b>

---

## 4.1 Introduction

Prediction is a critical field in process mining. As mentioned earlier, to make process predictions, it is necessary to have access to the traces of the executions of business processes. These traces can be found in the logs of information systems used by business actors during process execution, including email systems.

In the context of emails, prediction refers to recommending a set of email response fields, including the email sender, recipients, attached files, main body, and more, in response to a received email. In cases specifically focused on recommending email main body content, machine learning algorithms are employed to analyze patterns and identify keywords or phrases. For example, Fang et al. [111] proposed a collaborative filtering-based email content recommendation system, while Wang et al. [113] proposed a deep learning-based approach that uses a combination of user and email features to build a predictive model that recommends email main bodies tailored to individual recipients. Nevertheless, the main focus of these existing works was to enhance email management without giving much thought to the context of business processes. For those that combined email management with the notion of

BP, they were mostly limited to the stage of BP discovery from email logs [66] or at most classifying incoming emails into BP activities [88].

However, as stated in Chapter 2, making predictions in the context of process-oriented emails is not only limited to identifying future BP activities to be performed through emails but also requires recommending the emails that enable BP actors to perform these activities, mainly the textual content of their email response bodies. By analyzing historical email data in the context of BP, predictive models can generate recommendations for the content of future emails, helping business actors produce emails more efficiently and effectively. This can lead to improved communication and collaboration, reduced errors, and increased productivity.

Within this context, this chapter introduces a process-activity-aware email response recommendation system that takes the generated event log from the previous work as input and predicts future BP knowledge. This knowledge pertains to the set of activities to be conveyed in the email responses and the manipulated business data. Additionally, we provide an email response body template recommendation based on the predicted activities and historical textual contents related to the predicted BP knowledge.

The structure of this chapter is outlined as follows: Section 4.2 delves into the specifics of the proposed approach, elucidating the methodology employed for recommending an email response body template. Section 4.3 outlines the evaluation of the proposed approach, showcasing its performance and impact. Finally, in Section 4.4, we conclude the chapter by summarizing the key contributions and providing reflections on potential future directions and applications of this work.

## 4.2 The Proposed Approach Overview

This section will provide a detailed analysis of our suggested solution, including the methods and approaches we intend to use to address the problems that have been discovered. *Figure 4.1* provides an extensive graphic depiction of our methodology. Our primary objective is to provide email response recommendations while considering the activities executed through the received email. The figure emphasizes shaded gray areas, representing the work of Elleuch et al. [42] that has laid the foundation upon which this chapter is built.

Our approach is structured into four distinct phases, divided into two segments. In the first segment, entitled *BP Knowledge Extraction and Prediction*, phases 1, 2, and 3 serve to predict the BP knowledge to be included in the email response. In the second segment, the *Response Template Recommendation*, phase 4 uses the predicted BP knowledge to recommend a response template. These phases are connected by arrows, with some in **blue** representing their role in suggesting an email response template for received emails, and others in *black* indicating the preprocessing steps necessary to generate the required models and inputs for recommending the appropriate email response template.

It is important to note that the BP Knowledge extraction phase serves a dual purpose:

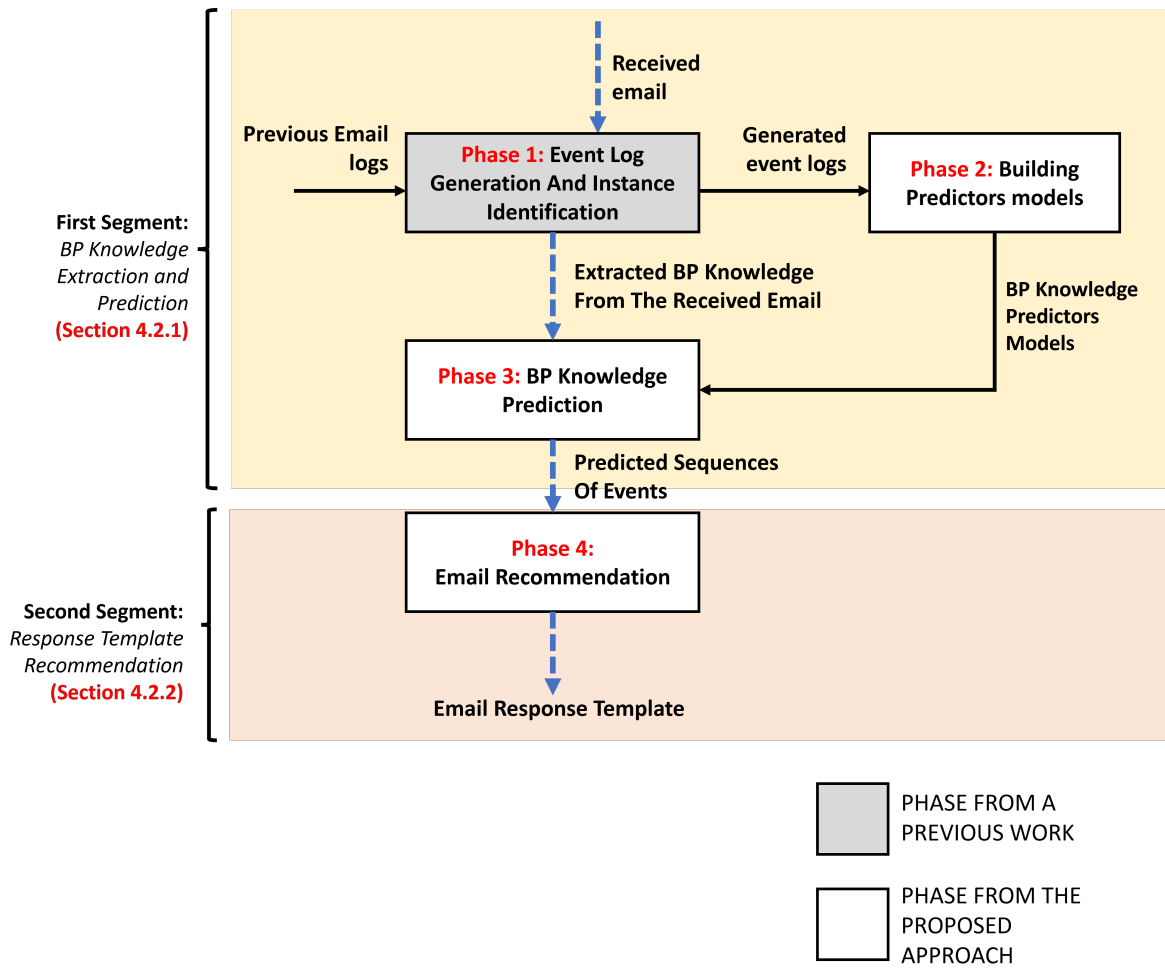


Figure 4.1: The proposed approach overview

preprocessing and recommending emails. In the preprocessing step, when an email log—a chronological record of email communications, including sender and recipient addresses, timestamps, subject lines, and sometimes message content—is received, the initial phase involves generating an event log from previously exchanged emails (as detailed in section 4.2.1).

To achieve this, we utilized the work of Elleuch et al. [42], who proposed an approach to transform unstructured email logs into structured event logs before mining them for discovering BP from multiple perspectives. The authors introduced several algorithmic solutions for: (i) unsupervised learning activities based on discovering frequent patterns of words from emails, (ii) discovering activity occurrences in emails for capturing event attributes, and (iii) discovering speech acts of activity occurrences for recognizing the sender’s purpose of including activities in emails.

The event log, created during the BP knowledge extraction phase, serves as the foundation for the building predictors models phase, where we develop BP prediction models designed to predict the BP knowledge to be integrated into the email response (elaborated in section

4.2.1). We recall that the BP knowledge includes the prediction of the set of activities to be expressed in the email response, the intention of expressing them in the email, as well as the manipulated business data.

However, when recommending an email response template for a received email, the BP Knowledge extraction phase is again invoked to identify the instance of the received email. Subsequently, the BP Knowledge Prediction phase predicts the BP knowledge relevant to the predicted response email (explained in section 4.2.1). Finally, in the Email Recommendation phase, our approach recommends an email response template by analyzing the textual content related to the BP knowledge of the email response (detailed in section 4.2.2).

#### 4.2.1 First segment: BP Knowledge Extraction and Prediction

##### BP Knowledge extraction - Phase 1

This phase serves a dual purpose. Firstly, it involves creating an event log from an existing email log, which serves as the basis for training and developing prediction models. Secondly, when the system receives an email, it can identify the specific instance of that email. To achieve this, we employ the approach developed by Elleuch et al. [42], which has been detailed in Chapter 3, Section 3.2.1. This approach is entirely unsupervised and relies on pattern discovery to extract valuable business process knowledge from emails. Figure 4.3 depicts an excerpt from the event log extracted from the main body of the email shown in Figure 4.2. We have highlighted the expressions where the events  $e_5$ ,  $e_6$ ,  $e_7$ ,  $e_8$  and  $e_9$  occurred in the email, as explained in the legend of Figure 4.2.

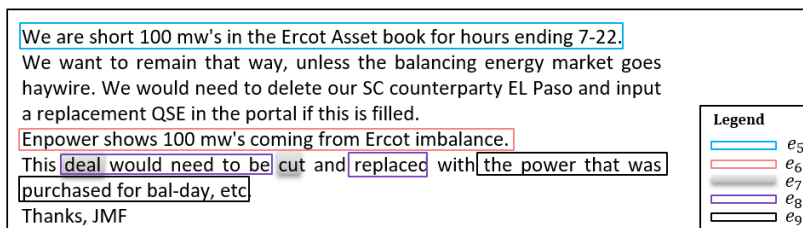


Figure 4.2: Email main body

##### Building Predictors models - Phase 2

The primary objective of this phase is to train prediction models tailored to the task of forecasting relevant BP knowledge that can be seamlessly integrated into email responses. To achieve this, we leveraged the information extracted from the generated event log, enabling us to discern the sequential relationships that exist among events. We transformed every thread within our event log into a well-structured sequence of events. In the subsequent sections, we provide a formal definition of what constitutes a sequence of events.



Event	$Act_o$			SA	$At_{Ind}$	$I_{values}$	em					$Th_{id}$
Ev_ID	$AN_{occ}$	$BD_{occ}$	$BC_{occ}$				ID	times tamp	Sender	Recipients	Con vids	
$e_5$	('open short position', ['short', '100', 'mws'], [1, 2, 3])			'information'	['we']							
$e_6$	{'display deal', ['shows', '100', 'mws'], [27, 28, 29]}	{{'numeric mw', ['numeric', '100'], [2, 3]}, {'hour0end numeric numeric', ['hour ending', '7', '22'], [5, 6]}, {'orgname', ['Ercot Asset'], [4]}, {'orgname', ['El Paso'], [20]}, {'orgname', ['Empower'], [26]}, {'numeric mw', ['numeric', '100'], [28, 29]}, {'orgname', ['Ercot'], [30]}, {'orgname', ['bal-day'], [38]}}		'information'	['enpower']							
$e_7$	('cut deal', ['cut', 'deal'], [33, 31])		{}	'intention'	[]	[]	'<23535150.1075852369200.javaMail.evans@thyme>'	'Wed, 26 Sep 2001 10:19:15 -0700 (PDT)'	'm.forney@enron.com'	['joe.capasso@enron.com', 'l.day@enron.com', 'joe.rrigo@enron.com', 'alexander.mcelreath@enron.com', 'jeffrey.miller@enron.com', 'steve.olinde@enron.com', 'eric.saibi@enron.com']	'C3'	['Th <sub>3</sub> ']
$e_8$	('replace deal', ['replace', 'deal'], [34, 31])			'intention'	[]							
$e_9$	('purchase power', ['power', 'purchased'], [36, 35])			'information'	[]							

Figure 4.3: Example of an Event Log Extract

**Definition 4.1. Sequence of Events** A sequence of email events is defined as  $S = s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_n$  where:

1.  $s_i$  is a sub-sequence of events such that  $s_i = e_{i1} \rightarrow e_{i2} \rightarrow \dots \rightarrow e_{ij}$  refers to an ordered list of events belonging to the same email  $i$ . Each event is denoted by a single variable  $e_{ij}$ , where the index  $i$  refers to the email index in which the event belongs and  $j$  indicates the  $j$ -th event of the sequence. Also,  $e_{ij} \rightarrow e_{i,j+1}$  means that in an email  $e_i$ , the event  $e_{ij}$  appears before  $e_{i,j+1}$ ;
2.  $s_i \rightarrow s_{i+1}$  means that the sub-sequence  $s_i$  (appearing in email  $i$ ) precedes the sub-sequence  $s_{i+1}$  (appearing in email  $i + 1$ ) in terms of sending time;
3. The index  $n$  (such that  $n \geq 1$ ) refers to the number of sequentially sent emails within a part of an email thread.

Consider the example of emails depicted in Figure 4.4. The sub-sequence of events extracted from email 1 is denoted by  $s1$ . It contains three sequential events based on their order of appearance and it is represented as follows:  $s1 = e_{11} \mapsto e_{12} \mapsto e_{13}$ . Similarly, the sub-sequence of events extracted from email 2 is denoted by  $s2$  and contains a single event  $e_{21}$ . Email 1 is received before email 2 in the same thread. Thus, the sub-sequence  $s1$  precedes the sub-sequence  $s2$  in the sequence of events and is represented as  $s1 \mapsto s2$ .

In this study, two prediction models were developed and trained. The first model takes as input the sub-sequence of events appearing in a received email and predicts the possible next combinations of relevant BP knowledge that may appear in the email response. The BP knowledge within the predicted sub-sequences from the first model can have multiple orders of appearance. The second prediction model is used to predict the order of the BP knowledge following it in the same email. In the subsequent sections, we will refer to the first prediction model as the "**next-bp-knowledge**" prediction model, and to the second prediction model as the "**sub-sequence**" prediction model.

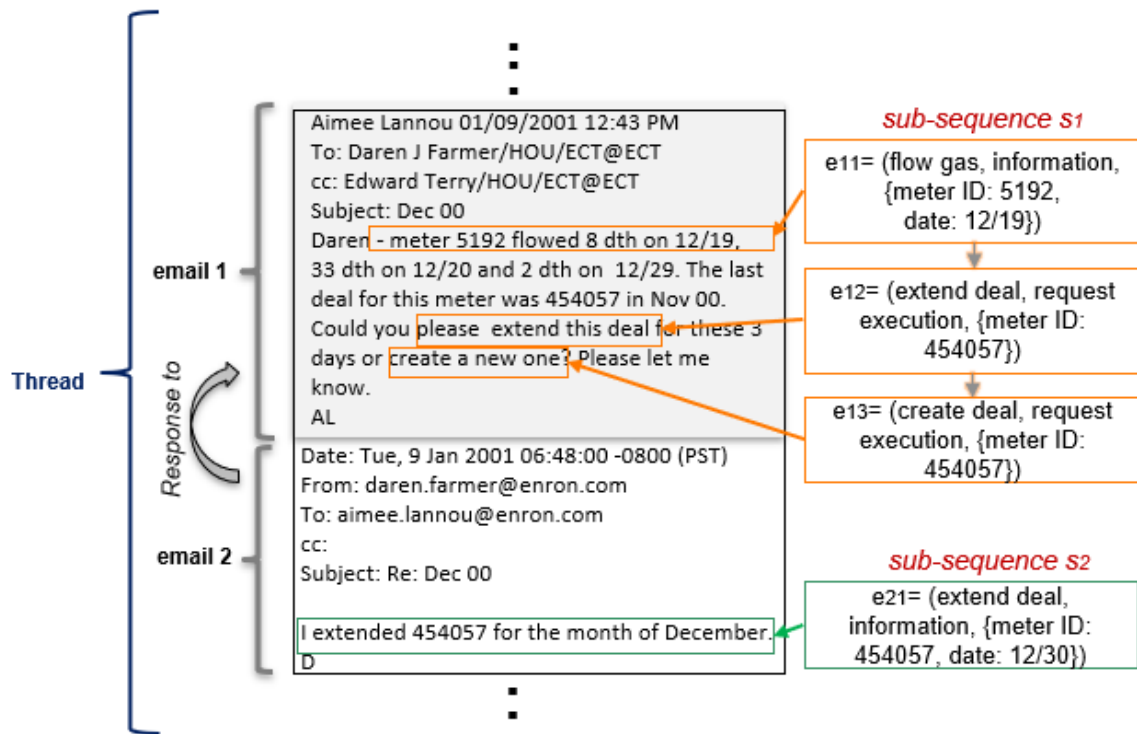


Figure 4.4: Sequence of events example extracted from two emails in the same thread

Both models were trained using the Long Short-Term Memory (LSTM) architecture. A particular kind of Recurrent Neural Network (RNN) renowned for its remarkable ability to process sequential data jobs is the LSTM. It is perfect for jobs where the order and temporal dependencies of data are crucial, like time series analysis, speech recognition, and natural language processing. It also excels at catching patterns and anticipating sequences.

During the training process, generated sub-sequences of events extracted from email threads were utilized. The "**next-bp-knowledge**" model focused on learning dependencies between sub-sequences of events within the same email threads, while the "**sub-sequence**" model learned from dependencies between BP knowledge's belonging to the same sub-sequences within the emails. To effectively capture dependencies and patterns from these sequential data inputs in each model, the LSTM cell utilizes three crucial components: the input gate, the forget gate, and the output gate. These components are equipped with sigmoid activation functions, acting as adaptive switches that allow the model to control the flow of information at each time step.

When processing a sub-sequence of events, each BP knowledge is represented as a data point at a specific order, and the LSTM processes these BP knowledge's sequentially, considering their order and time dependencies. Several computations take place at every time step: In order to enable the model to focus on relevant characteristics while filtering out noise, the input gate decides how much of the current BP knowledge information should be incorporated into the current memory cell state; The LSTM can capture long-term dependencies because the forget gate regulates how much data from the previous time step should be kept in the

memory cell state; The candidate memory update, which is derived from the current input and the prior hidden state, represents new data that might be stored in the memory cell state; By combining data from the input gate and the candidate memory update through element-wise multiplication and subtraction of the forget gate, the state of the memory cell is updated; the output gate determines how much of the current memory cell state should be exposed to the current hidden state; and finally, the hidden state is calculated based on the memory cell state and the output gate, capturing the relevant information from the current memory cell state.

To enhance computational efficiency and manage memory consumption, we have implemented truncated back-propagation in our LSTM network. This technique involves breaking the input sequence into smaller sub-sequences of a specified length ( $K$ ) and unfolding the network for a fixed number of time steps. Back-propagation through time (BPTT) is then employed on each sub-sequence, enabling the computation of gradients and subsequent updating of model parameters based on the accumulated information from past time steps during the training process. By applying BPTT to these sub-sequences, we ensure smoother and more efficient training of our model while maintaining its recurrent nature. Once all sub-sequences have been processed, the accumulated gradients are then utilized to update the model parameters effectively. Given a sub-sequence of events with  $T$  time steps and a fixed truncation length  $K$  (where  $1 \leq K \leq T$ ), the truncated back-propagation process can be summarized in three main steps: First, during the forward pass, for each sub-sequence with  $K$  time steps from the original sequence, initialize the hidden state and memory cell state, and then perform forward pass computations for each time step  $t$  from 1 to  $K$ , producing predictions and updating the hidden and memory cell states. Second, calculate the loss function based on the predictions and target outputs for each time step within the sub-sequence. Third, apply BPTT for each sub-sequence by initializing the gradients of the model parameters to zero and then performing backward pass computations from  $t = K$  to 1, updating the gradients based on the loss function and the dependencies between the model parameters, predictions, and hidden states.

To enhance the performance of our LSTM models, we conducted a comprehensive hyper-parameter tuning process. We fine-tuned critical hyper-parameters, including the learning rate, batch size, number of LSTM layers, and dropout rates.

### BP Knowledge Prediction - Phase 3

In this phase, we utilize the prediction models developed during the second phase to forecast the relevant BP knowledge to include in our email response. Our objective is to expand traditional email recommendations, typically confined to BP discovery within process-oriented emails (discussed in Chapter 2, Section 2.3). The process involves inputting the extracted sub-sequence of events from the third phase into the "*next-bp-knowledge*" prediction model. This model returns a list of BP knowledge combinations. Each combination is assigned a confidence value. This value is determined by the average length of the intersection between the BP knowledge in the combination and the BP knowledge found in previous emails. The

confidence value for each combination  $C_k$  is calculated using the formula:

$$\text{conf}(C_k) = \frac{1}{\text{total\_num\_combinations}} \sum_{\text{previous\_email}} \text{len}(\text{intersection}(C_k, \text{previous\_events}))$$

Here,  $\text{total\_num\_combinations}$  is the total number of combinations from the prediction model across all previous emails.

We select the BP knowledge combination  $C_{\text{best}}$  with the highest confidence value:

$$C_{\text{best}} = \text{argmax}(\text{conf}(C_k)) \quad \text{for } k \in \{1, 2, \dots, k\}$$

Consider a sub-sequence of events  $s_1 = [e_{11}, e_{12}, \dots, e_{ij}]$ , where each  $e_{ij}$  represents an individual BP knowledge. The “*next-bp-knowledge*” prediction model takes  $s_1$  as input and returns a list of BP knowledge combinations  $C = [C_1, C_2, \dots, C_k]$ . Each combination  $C_k$  is assigned a confidence value  $\text{conf}(C_k)$  based on the overlap with BP knowledge from previous emails. After selecting  $C_{\text{best}}$ , we process each BP knowledge within it using the “*sub-sequence*” prediction model. This model predicts the ordered BP knowledge that should follow in the email. For each BP knowledge, the *sub-sequence* prediction model provides a list of ordered BP knowledge combinations  $O_i$ , each with a confidence value. The confidence value for each ordered combination  $O_i$  is calculated similarly to before

$$\text{conf}(O_i) = \frac{1}{\text{total\_num\_ordered\_combinations}} \sum_{\text{previous\_email}} \text{len}(\text{intersection}(O_i, \text{previous\_ordered\_events}))$$

Here,  $\text{total\_num\_ordered\_combinations}$  is the total number of ordered combinations from the model across all previous emails. We then select the final ordered sub-sequence of events  $O_{\text{best}}$  with the highest confidence value:

$$O_{\text{best}} = \text{argmax}(\text{conf}(O_i)) \quad \text{for } i \in \{1, 2, \dots, m\}$$

This selected  $O_{\text{best}}$  represents the predicted BP knowledge to include in the email response.

To illustrate Phases 1, 2, and 3, we will provide a running example. Consider the following scenario, which involves writing a response to an email sent by David. Within this email (illustrated in Figure 4.5), David communicates with Julie about Deal 235670, associated with Teco Gas Processing, which expired on 12/22. Notably, he also mentions a successful sale on 02/01, suggesting the potential for an extension for this deal with Teco Gas Processing. Given these developments, he solicits Julie’s perspective on whether to proceed with an extension and make adjustments to the sale using the Unify system, stressing the urgency of her feedback.

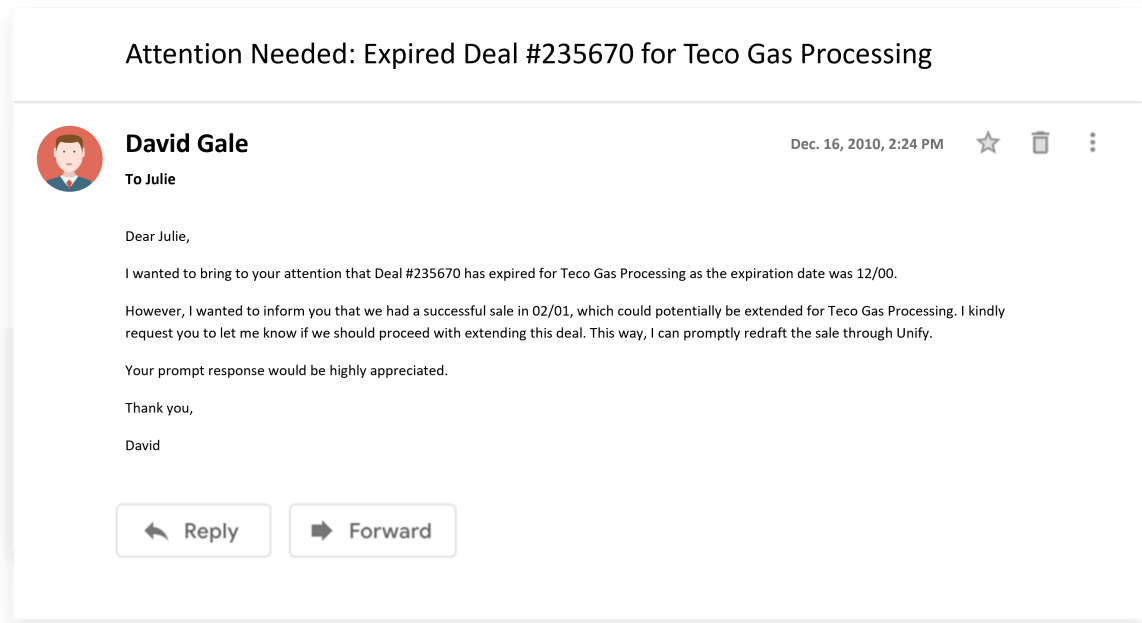


Figure 4.5: David's Email Correspondence with Julie

To effectively assist Julie in crafting a response to the email received from David, let's apply a phased approach as outlined below. The initial phase involves the identification of specific BP knowledge that directly pertains to the content of David's email (**Phase 1**). Within the email, two notable events stand out: the "**Expired Deal**" and the "**Extended Deal**".

The "**Expired Deal**" event is clearly delineated in a specific sentence wherein David communicates, *"Deal 235670 for Teco Gas Processing has expired as of 12/00"*. This statement serves as an informative declaration, indicating the termination of the mentioned deal. The associated **BD** relevant to this event include *"deal"* and *"expired"*.

Similarly, the "**Extended Deal**" event is discernible from another sentence where David provides details and solicits a response: *"However, I wanted to inform you that we had a successful sale in 02/01, which could potentially be extended for Teco Gas Processing"*. In this instance, the speech act performed is a *"request for information"*, and the corresponding **BD** associated with this event encompass *"deal"* and *"extended"*.

Once the events contained in the received email have been identified, forming the sub-sequence of events from the received email, the next step involves forecasting relevant BP knowledge that can be seamlessly integrated into the email response, marking **Phase 2 and 3** of our approach. To do so, the sub-sequence of events is input into the *"next-bp-knowledge"* prediction model, which generates several potential BP knowledge combinations, each assigned a confidence value  $\text{conf}(C_k)$  or  $\text{conf}(O_i)$ . Figure 4.6 illustrates the BP knowledge combinations represented as sub-sequences of events predicted by our models.

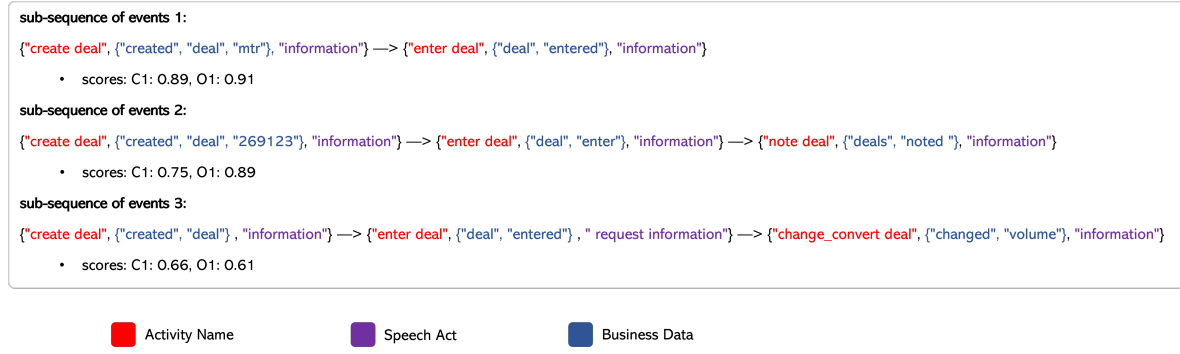


Figure 4.6: Predicted sub-sequence of events

- **Sub-sequence of Events 1:** This sub-sequence involves creating a deal and subsequently entering the deal details, boasting the highest combination score. Both the speech acts and **BD** align with the situation presented in the original email, making it the most appropriate response option. The high confidence score indicates that this sequence is highly likely to occur and aligns well with the speech acts and **BD** expectations in the context of the original email. This suggests that this BP knowledge is the most appropriate to be included in the response to the email received by David.
- **Sub-sequence of Events 2:** Although this sequence has a reasonable confidence score, it is lower than the first sub-sequence. This indicates that while noting a deal is a relevant activity, it is less commonly executed by BP actors immediately after entering a deal. Thus, it is not considered the primary response option, even though it remains significant.
- **Sub-sequence of Events 3:** This sub-sequence has the lowest confidence score, indicating that it is the least likely to align with the situation presented in the original email. The activity "change convert deal" seems unrelated to the primary topic of the email, and the sequence of speech acts and **BD** does not fit the context well. Therefore, this sub-sequence is considered irrelevant for guiding the subsequent steps in response to the email.

## 4.2.2 Second segment: Response Template Recommendation

### Email Recommendation - Phase 4

Finally, we have reached the last phase, which involves recommending an email response template based on the textual content related to the BP knowledge of the email response.

Let's define  $W$  as the set of words in emails. A formal definition (Definition 4.2) of an email response template is presented as follows:

**Definition 4.2 (Email Response Template).** *An email response template  $T$  represents a sequence of sentences  $T = p_1 \mapsto p_2 \mapsto \dots \mapsto p_n$ , where:*

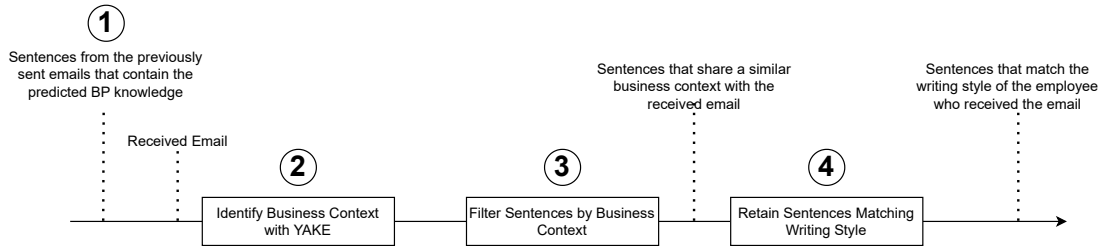
1.  $p_i$  is a sequence of terms within the same sentence,  $p_i = t_{i1} \mapsto t_{i2} \mapsto \dots \mapsto t_{ij}$ , such that:
  - (a) Each term is denoted by a single variable  $t_{ij}$ , where the index  $i$  refers to the sentence index, and the index  $j$  refers to the  $j^{\text{th}}$  term in the sentence.
  - (b)  $t_{ij} \in \mathcal{W}$  for all  $i, j$ .
  - (c)  $t_{ij} \mapsto t_{i,j+1}$  means that  $t_{ij}$  occurs before  $t_{i,j+1}$ .
  - (d) Each term  $t_{ij}$  could be of two types: an unchangeable word or a modifiable word. A modifiable word refers to an entity tag (e.g., numeric value, organization, person's name, localization) that an employee can replace with a list of business data values.
2.  $p_i \mapsto p_{i+1}$  means that the sentence  $p_i$  appears before the sentence  $p_{i+1}$ .
3. The index  $n$  (where  $n \geq 1$ ) refers to the number of sequential sentences in the email template  $T$ .

We proceed with the underlying assumption that the suggested email responses are recommended based on the carefully delineated criteria outlined below:

- **Alignment with Business Context:** The responses must share a similar business context with the received email. This implies that the suggested replies should be pertinent to the subject topics found in the original email. For instance, if the original email concerns a marketing strategy, the suggested responses must also concentrate on marketing-related matters, rather than diverging into unrelated domains like Information Technology support.
- **Inclusion of Predicted BP Knowledge:** The suggested responses should not merely align with the business context but should also encompass predicted BP knowledge. This refers to the integration of accurate, current information related to the specific business operations, practices, or workflows mentioned in the original email. By ensuring that the response incorporates this type of specialized information, the email can effectively address the recipient's needs or queries.
- **Consistency with the Author's Writing Style:** It's imperative that the suggested email responses are composed in the same writing style as the author. This entails emulating the language, syntax, and overall approach found in the original email. Here, keeping stylistic consistency is key to giving the recipient a seamless experience. The suggested replies should have the same tone as the original email, regardless of whether it was formal, informal, technical, or conversational. This will give the appearance that the original sender sent them.

By adhering to these criteria, the suggested email responses aim to provide more personalized, relevant, and coherent communication that aligns with the context. Consequently, for each BP Knowledge within the chosen list of predicted BP Knowledge from **Phase 3**, we

select the most appropriate sentence and concatenate this sentence to form the content of the recommended email response. As highlighted in Figure 4.7, this process involves identifying the business context, filtering sentences by this context, and retaining those that match the employee’s writing style. In the following, we detail each step taken to select the most suitable sentence for each BP Knowledge to construct the email response template.



**Figure 4.7:** Sentence Recommendation for BP Context

1. We retrieve all email sentences from previously sent emails that contain the predicted BP knowledge.
2. We identify the business context of the received email and the retrieved email sentences using Yet Another Keyword Extractor (Yake). Yake, as introduced by Campos et al. [23], employs a sophisticated methodology that taps into the inherent semantic and syntactic patterns contained within sentences. This enables it to extract pertinent keywords and phrases specific to the business domain.

The initial phase of *Yake* involves preprocessing the sentences from the received email and the retrieved emails. This preparatory step encompasses processes such as tokenization, part-of-speech tagging, as well as the elimination of stop words and punctuation. By segmenting the sentences into individual tokens and discerning their grammatical functions, *Yake* is able to attain a more profound comprehension of the linguistic structure, thereby facilitating more effective extraction of context.

Following the completion of preprocessing, *Yake* adopts a hybrid strategy that amalgamates unsupervised and supervised learning techniques. The unsupervised methods come into play during the initial keyword extraction phase, during which the algorithm identifies frequently occurring and statistically noteworthy terms in the sentences. Subsequently, *Yake* harnesses the power of supervised learning to rank these potential keywords based on their pertinence to the business domain. The supervised models are fine-tuned using annotated datasets that have been meticulously curated to encompass a diverse spectrum of business topics and contexts.

The procedure for contextual ranking within *Yake* hinges on the usage of Term Frequency Inverse Document Frequency (TF-IDF), a sophisticated scoring mechanism that takes into consideration various linguistic attributes and the semantic interconnectedness between words. By taking these elements into account, *Yake* ascertains the significance of a given keyword or phrase with respect to the overarching business context portrayed by the sentences.



Consider again the email in Figure 4.5. Therefore, Yake identifies and ranks the following keywords based on their pertinence to the business domain:

- *Expired Deal and Expiration Date*: The email from David Gale mentions that Deal #235670 for Teco Gas Processing has expired as the expiration date was 12/00. This highlights the need to address the expired deal to ensure continuity and avoid any disruptions in operations.
  - *Successful Sale and Deal Extension*: David also informs Julie of a successful sale in 02/01, suggesting the potential to extend the deal for Teco Gas Processing. This implies there is a recent positive performance that can be leveraged to renegotiate or extend the terms of the expired deal.
  - *Teco Gas Processing and Redraft Sale*: The email specifically mentions Teco Gas Processing and the need to promptly redraft the sale through Unify. This context indicates that Teco Gas Processing is a significant client, and there is a procedural step involved (using Unify) to formalize any extensions or new agreements.
3. We refine the retrieved sentences by selecting those that closely match the business context of the received email. This process entails several interconnected steps that synergize seamlessly.

Initially, we utilize contextual embeddings generated by training a transformer-based model with an extensive collection of business-related texts. This methodology enables our system to comprehend sentence meanings deeply within the realm of business communication. These embeddings not only capture word meanings but also consider how words integrate into the sentence's structure.

We use a sentence-to-vector technique, combining an attention mechanism to assign different weights to words in a phrase based on their relevance, to bridge the gap between these embeddings and the email content. Subsequently, we concentrate on filtering and retaining sentences that closely align with the associated business context. We utilize cosine similarity to gauge the resemblance between the vector of a retrieved sentence and those of sentences within the email, setting a threshold determined through empirical testing and validation. High cosine similarity scores highlight sentences that share a comparable business context.

Acknowledging that relying solely on cosine similarity may not always yield optimal results, we introduce an additional layer of filtering using supervised learning. We train a classifier that considers features such as sentence length, specific keywords, and other attributes relevant to the unique business context. This additional step ensures that the retained sentences are not only contextually similar but genuinely relevant to the specific context of the business email. This layered approach creates a synergistic method for effectively preserving sentences that align with the business context of the received email.

4. Finally, we retain sentences that match the writing style of the employee who received the email. To accomplish this, we used a method called stylometric analysis. This technique enables us to identify the unique writing styles of different authors in individual

sentences. Our approach involved several stages, including data preparation, feature extraction, and style modeling.

---

**Algorithm 3** Stylometric Analysis
 

---

**Input:** *emailSentences*

**Output:** *matchedEmployee*

```

1: preparedData  $\leftarrow$  {}
2: features  $\leftarrow$  []
3: clusters  $\leftarrow$  []
   {Step 1: Clean and preprocess each sentence in the email}
4: for sentence in emailSentences do
5:   cleanedSentence  $\leftarrow$  CleanAndProcess(sentence)
6:   preparedData.append(cleanedSentence)
7: end for
   {Step 2: Extract stylometric features from the cleaned sentences}
8: for data in preparedData do
9:   featureVector  $\leftarrow$  ExtractFeatures(data)
10:  features.append(featureVector)
11: end for
   {Step 3: Initialize cluster centers for k-means clustering}
12: kCenters  $\leftarrow$  InitializeRandomCenters(k)
   {Step 4: Iteratively assign features to nearest cluster center and update centers}
13: repeat
14:   for feature in features do
15:     AssignToNearestCenter(feature, kCenters)
16:   end for
17:   for center in kCenters do
18:     UpdateCenter(center, features)
19:   end for
20: until HasConverged(kCenters)
   {Step 5: Group features into clusters based on final centers}
21: clusters  $\leftarrow$  GroupByCenter(features, kCenters)
   {Step 6: Profile employee based on the clustered features}
22: for cluster in clusters do
23:   matchedEmployee  $\leftarrow$  ProfileEmployee(cluster)
24: end for

```

---

The first step in our methodology involved preparing labeled data. We accumulated a substantial collection of sentences authored by the same individual to ensure a well-rounded representation. This step required significant manual effort to gather and label the sentences accurately. Domain experts were involved in this process to ensure the quality and reliability of the labels. These experts reviewed and annotated the sentences, associating them with the respective authors. This extensive manual labeling process was crucial to creating a high-quality training dataset (lines 1  $\rightarrow$  3).

Following this, the gathered sentences underwent meticulous cleaning and pre-processing, which involved the removal of extraneous elements like special characters and

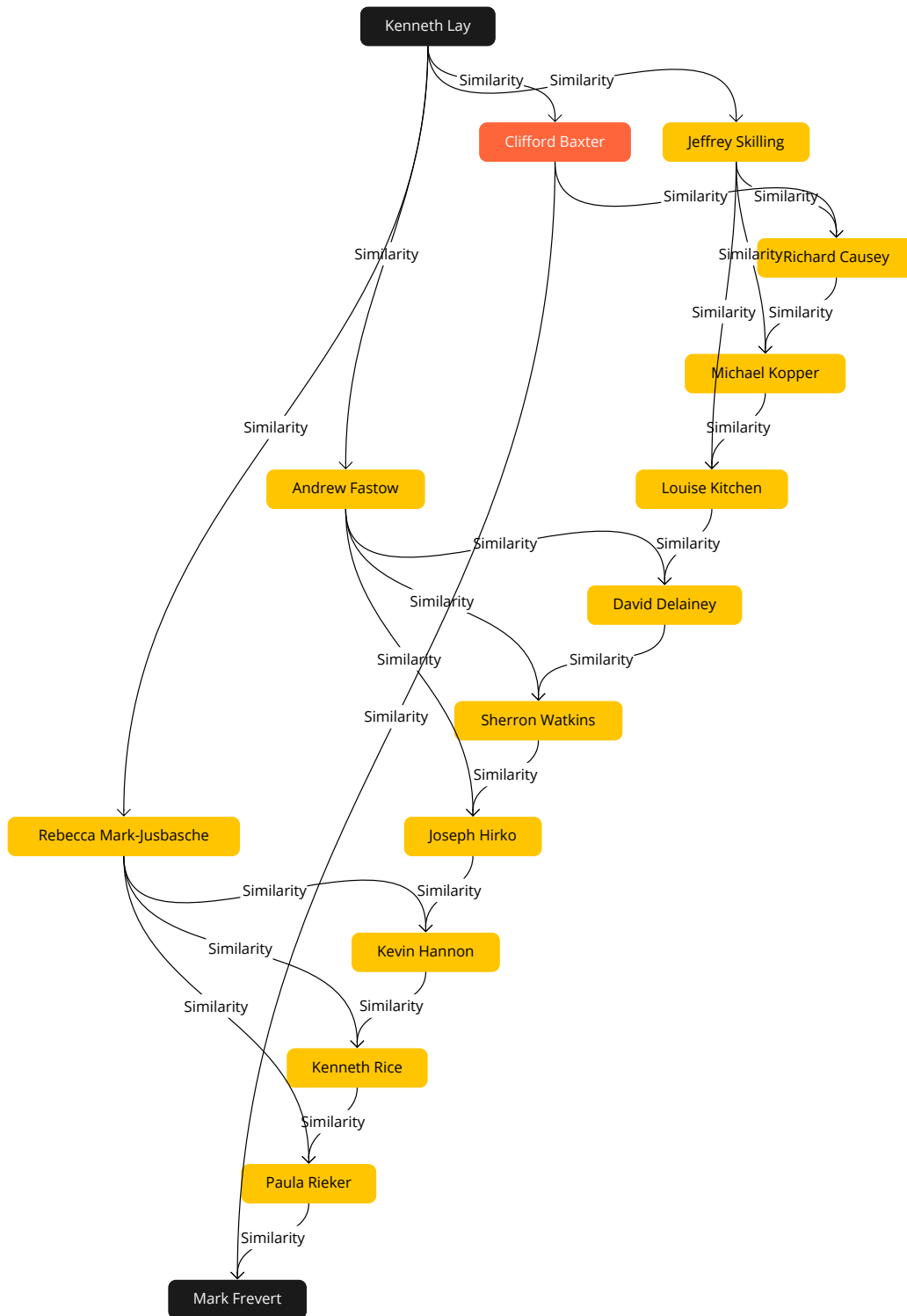
capitalization. The cleaning process included normalizing text to a consistent format, such as converting all characters to lowercase and removing punctuation. Tokenization was applied to split sentences into individual words or tokens, which facilitated more precise analysis in subsequent stages (lines 4  $\rightarrow$  7).

Transitioning to the feature extraction phase, the stylometric analysis leveraged a range of linguistic attributes to identify the distinctive traits of an author's writing style (lines 8  $\rightarrow$  11). The three primary categories for these characteristics were lexical, syntactic, and structural [81, 39, 68]. Metrics like word frequency, vocabulary richness, and n-grams (sequences of n words) are examples of lexical qualities. For instance, vocabulary richness assessed the variety of original terms employed by an author, whereas word frequency examined the frequency of particular words inside a sentence. Sentence length, grammar patterns, and part-of-speech tagging were among the topics covered by syntactic characteristics. Part-of-speech tagging helped to determine the author's syntactic preferences by identifying the grammatical components (nouns, verbs, adjectives, etc.) in a sentence. Variations in features like sentence and paragraph lengths, punctuation use, and the use of particular stylistic devices like alliteration or metaphors were all taken into account when determining structural qualities. These structural elements helped to capture the overall formatting and stylistic choices made by an author.

Having acquired the relevant features from the text, we proceeded to the style modeling phase. At this stage, we harnessed machine learning techniques to precisely quantify the similarities and differences between sentences, based on the extracted features. In our methodology, we opted for the widely used clustering algorithm "k-means". This iterative algorithm effectively groups sentences into a predefined number of clusters by minimizing the variance within each cluster. The process involved several steps: initializing cluster centers randomly, assigning sentences to the nearest cluster center based on feature similarity, and updating the cluster centers by calculating the mean of assigned sentences. This iterative process continued until the cluster centers converged, resulting in well-defined clusters of sentences exhibiting similar stylistic characteristics. This process was entirely automated, utilizing the extracted features to drive the clustering process without further manual intervention (lines 12  $\rightarrow$  21).

Subsequently, we engaged in employee profiling by meticulously studying the labeled data within each cluster, aiming to identify the predominant employee associated with each cluster (lines 22  $\rightarrow$  24). By mapping employees to clusters, we could discern distinctive writing styles connected to each employee, offering valuable insights into their individual linguistic traits. This analytical approach empowered us to select and retain sentences aligning with the writing style of the particular employee who had received the email. The final profiling step was semi-automated, involving algorithmic analysis followed by expert verification to ensure accuracy.

In Figure 4.8, a network graph is presented that effectively illustrates the interconnected relationships among 15 specific employees. Within the graph, individual nodes correspond to distinct employees, while the links interconnecting them symbolize the similarity in writing style between two employees.



**Figure 4.8:** Network graph of selected employees based on similarity in writing styles

We use the Stanza Python library for natural language analysis [89] to detect and categorize named entities in sentences. This includes identifying dates, person names, locations, organizations, quantities, and other entities, as detailed in algorithm 4 (lines 6  $\rightarrow$  12). After recognizing these named entities, we replace them with appropriate tags (lines 15  $\rightarrow$  18). These tags serve as editable placeholders that employees can subsequently replace with real business data.

---

**Algorithm 4** Named Entity Tagging and Replacement
 

---

**INPUT:** sentences

**OUTPUT:** taggedSentences {Sentences with replaced named entities}

```

1: namedEntities  $\leftarrow$  {} {To store identified named entities.}
2: tags  $\leftarrow$  {} {To store corresponding tags for named entities.}
3: taggedSentences  $\leftarrow$  [] {Will hold sentences with replaced named entities.}
4: for sentence in sentences do
5:   entities, sentenceWithTags  $\leftarrow$  IdentifyEntities(sentence) {Uses Stanza to identify
   entities and tag the sentence.}
6:   for entity in entities do
7:     if entity not in namedEntities then
8:       tag  $\leftarrow$  GenerateTag(entity) {Creates a new tag for the entity.}
9:       namedEntities[entity]  $\leftarrow$  tag
10:      tags[tag]  $\leftarrow$  entity
11:     end if
12:   end for
13:   taggedSentences.append(sentenceWithTags)
14: end for
15: for sentence in taggedSentences do
16:   for tag, entity in namedEntities do
17:     sentence  $\leftarrow$  ReplaceTagWithEntity(sentence, tag, entity) {Substitutes tags with
   actual entities.}
18:   end for
19: end for
20: return taggedSentences

```

---

Finally, as delineated in Algorithm 5, this phase concludes with two primary steps: (i) selecting the most suitable sentence corresponding to each event in the predicted event sequence (lines 2  $\rightarrow$  5), and (ii) concatenating these sentences in the order they appear (lines 6).

**Algorithm 5** Event Selection and Sentence Concatenation**INPUT:** taggedSentences, predictedEvents**OUTPUT:** finalEmailResponse {Concatenated sentences for email response}

---

```

1: eventSentences ← [] {To store selected sentences for each event.}
2: for each event in predictedEvents do
3:   selectedSentence ← SelectSentence(event, taggedSentences) {Selects the most appropriate sentence for the event.}
4:   eventSentences.append(selectedSentence)
5: end for
6: finalEmailResponse ← ConcatenateSentences(eventSentences) {Concatenates selected sentences for the email response.}
7: return finalEmailResponse

```

---

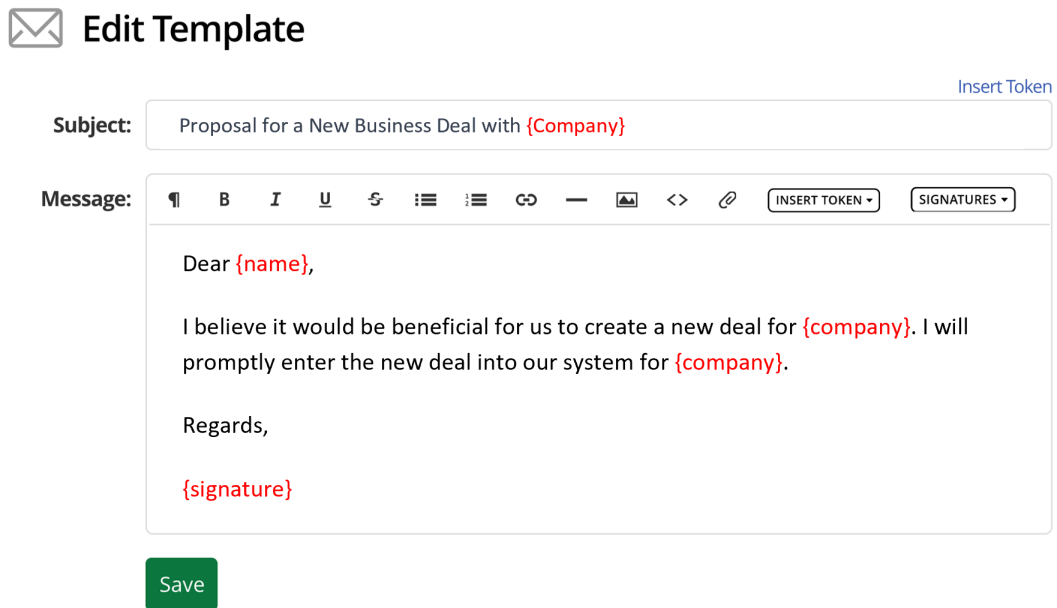
Referring back to our running example, let's apply **Phase 4** to the selected BP knowledge from **Phase 3** to recommend an email response template for the email received by David. In this process, we retrieve and analyze email sentences containing the selected BP knowledge from **Phase 3** to recommend a personalized and contextually relevant email response. We recall that for each BP knowledge, the proposed approach ensures that the selected sentences match both the business context of the received email and the writing style of the sender, in this case David. This ultimately constructs a coherent and suitable email response template.

For instance, the first BP knowledge is "*create deal*." The selected sentence for this knowledge is, "*I believe it would be advantageous for us to initiate a new deal with [Company].*" For the second BP knowledge, "*enter deal*," the chosen sentence is, "*I will promptly enter the new deal into our system for [Company].*"

These selected sentences are closely tied to the context provided in the email from David Gale, which highlights several key points:

- The email notes that Deal #235670 for Teco Gas Processing expired in December 2000. In the first sentence, this is addressed by highlighting the need to create a new deal to replace the expired one.
- The email emphasizes the significance of Teco Gas Processing and the necessity to promptly redraft the sale using Unify, a procedural step to formalize any new agreements. This second sentence underscores the importance of promptly formalizing the new agreement, in line with the procedural requirements mentioned in the email.

These sentences are then concatenated to form the final email response template, as illustrated in Figure 4.9. The phrases enclosed in curly brackets are customizable, allowing the user to personalize the email with specific details related to the recipient and the subject matter. For instance, the term *company* will be dynamically replaced with the company name.



**Edit Template**

Subject: Proposal for a New Business Deal with {Company}

Message:

Dear {name},

I believe it would be beneficial for us to create a new deal for {company}. I will promptly enter the new deal into our system for {company}.

Regards,

{signature}

Save

**Figure 4.9:** Proposed Email Response Template Tailored for Julie’s Communication Style in Reply to David

## 4.3 Experiments and Validation

The effectiveness of the proposed approach has been assessed using real emails acquired from the public Enron dataset<sup>1</sup>. These emails were either sent or received by Enron employees engaged in online energy trading. Instead of comparing our work to existing studies in the same field, which have significant limitations compared to the breadth and innovation of our approach, we evaluate our approach’s performance based on two aspects: (i) the precision of predicting BP knowledge in email replies (Section 4.3.1), and (ii) the relevance of the textual content in the suggested emails (Section 4.3.2). To demonstrate the practical application and benefits of our proposed approach, we present in Section 4.3.3 a detailed use case involving the utilization of RESTful API endpoint for email response recommendation. The approach, developed in Python, is readily available on GitHub<sup>2</sup>, where some of the experimental results can also be found.

### 4.3.1 Precision of Predicted BP Knowledge in Email Responses

For this evaluation, we will explain the process used in the event log generation phase to produce the event log. Additionally, we will demonstrate how the inclusion of BP knowledge in the sequences of events used for training the prediction models can affect the precision of the predicted BP knowledge.

<sup>1</sup><https://www.cs.cmu.edu/enron/>

<sup>2</sup><https://github.com/ralphbn1995/Predictive-process-approach-for-email-response-recommendations.git>

To generate the event log, we used 8200 emails from the Enron dataset and fed them to the event log generator developed in our previous work. The extracted event log contained 1340 sequences of events and 1865 sub-sequences of events, with 80% of the obtained sequence of events used to train the first prediction model and 80% of the obtained sub-sequences used to train the second prediction model. Each model had 128 neurons in the hidden layer and was trained for 100 epochs on the training dataset with a constant learning rate of 0.02. We selected 128 neurons in the hidden layer to balance model complexity and performance, capturing dependencies and patterns in the email threads without overfitting. Extensive experiments and cross-validation showed that this configuration provided the best trade-off between computational efficiency and predictive accuracy. Comparative analysis of models with varying neurons (from 64 to 256) revealed that the 128-neuron model consistently outperformed others in terms of precision and recall, justifying its selection for our final implementation.

We conducted two experiments:

- **Experiment 1:** We include only the activities in the sequences of events.
- **Experiment 2:** We include all the BP knowledge that we extracted previously. We represented a sequence of events as a set of linked activities, speech acts, and business data.

After training the LSTM model in each experiment, we apply the confusion matrix to the testing data. The confusion matrix serves as a performance measurement for predicting the accuracy of machine learning classification problems. Additionally, the confusion matrix proves to be useful in calculating precision and recall. In our case, the classes represent the activity names of the events found within the sequences of events.

Figure 4.10 illustrates the calculated metrics corresponding to the **30 most common classes**. These metrics are presented using bar charts for both *Experiment 1* and *Experiment 2*. In these charts, each class is depicted by a cluster of bars, where each individual bar represents a specific metric (*True Positive (TP)*, *True Negative (TN)*, *False Positive (FP)*, *False Negative (FN)*). The height of each bar reflects the frequency or count of the respective metric for that particular class.

Afterwards, we calculated the average precision and recall for each class. Our results show a comparison of average precision and recall for two datasets, '*Experiment one*' and '*Experiment two*'. **The findings demonstrate** that incorporating BP knowledge into the event sequences significantly enhances the LSTM model's prediction **accuracy** and **recall**. Specifically, with BP knowledge, the *average precision* increased from 0.76 to 0.85, and the *average recall* improved from 0.73 to 0.82. **This improvement is attributed to the integration of activities, speech acts, and business data**, which provides a richer context that aids the model in understanding and predicting complex connections within the sequences.



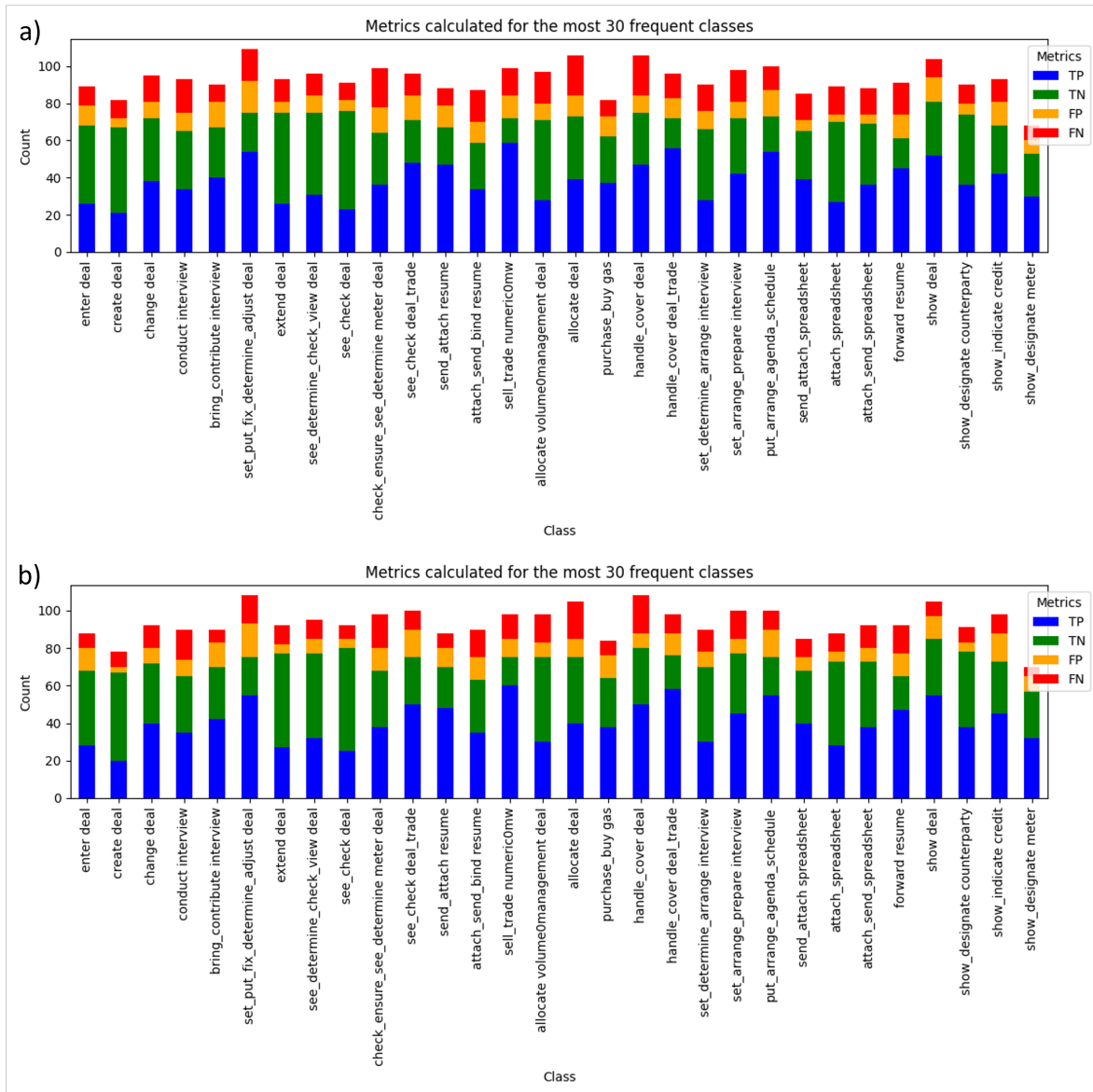


Figure 4.10: Comparing Metrics for the Top 30 Frequent Event Log Classes in Experiments 1 and 2

### 4.3.2 Effectiveness and Coherence of the Textual Content in Predicted Emails

To evaluate the effectiveness of our approach in generating **relevant** and **coherent** email responses compared to fine-tuned *GPT-3* and *GPT-4* models, focusing on the inclusion and order of events within the responses, we conducted a study. Our study utilized a custom dataset comprising 820 pairs of *Enron emails*, where each pair contained an original email and its corresponding response. This dataset was split into 85% for training and 15% for validation and testing to ensure a robust evaluation of the models. We fine-tuned *GPT-3* and *GPT-4* models using this dataset. The preprocessing steps included *tokenization* and

*normalization*, with careful selection of hyper-parameters: a learning rate of  $1 \times 10^{-5}$ , a batch size of 8, and training over 3 epochs using the **Adam optimizer**.

We conducted an analysis involving 35 email exchanges, each exchange including a received email and its response. We therefore extracted the **BP knowledge** from both the received emails and their responses. Using our approach and the fine-tuned *GPT-3* and *GPT-4* models, we then generated recommended responses to each of the received emails. From the generated email responses, we extracted the **BP knowledge** and then compared the **BP knowledge** from the generated responses with those from the email exchanges’ responses. To assess the quality of the generated responses, we used two key comparison metrics: **precision** and **recall**. *Precision* measures the number of relevant events included in the response divided by the total number of events in the response, helping us understand the accuracy of the generated responses. *Recall* measures the ability to retrieve and present relevant information in a logical and coherent manner. Our approach outperformed the fine-tuned *GPT-3* and *GPT-4* models in both precision and recall, as shown in Table 4.1. The precision of our approach was 0.9175, indicating a higher ability to include relevant events in the response emails compared to *GPT-3* (0.8789) and *GPT-4* (0.8915). This high precision demonstrated our model’s effectiveness in capturing and incorporating all pertinent details from the original email into the generated response. Our approach also achieved a recall of 0.9011, surpassing the recall of *GPT-3* (0.8517) and *GPT-4* (0.8723). This ensured that the sequence of events in the response was consistent with the expected order, enhancing the **logical flow** and **coherence** of the generated emails.

**Table 4.1:** Comparing Evaluation Results: Our Approach vs. Fine-tuned GPT-3 vs. Fine-tuned GPT-4 Models

Model	Events Included (Precision)	Order Accuracy (Recall)
Fine-tuned GPT-3 model	0.8789	0.8517
Fine-tuned GPT-4 model	0.8915	0.8723
Our Approach	<b>0.9175</b>	<b>0.9011</b>

### 4.3.3 Use Case Study

The goal of this use case is to evaluate the efficiency and quality, and user satisfaction of email response recommendations using our approach. To achieve this, we used a RESTful API endpoint to enable the seamless integration of our approach. Our RESTful API endpoint is meticulously designed with the same dual-focus approach discussed in Chapter 3, Section 3.4.2. The */response-template-suggestion* endpoint leverages our Predictive Process Approach for Email Response Recommendations to suggest appropriate email response templates. Upon receiving a POST request containing the email’s main body, it analyzes the content and provides a suitable response template in JSON format. This feature assists users in composing professional and contextually appropriate email responses, thereby enhancing communication efficiency and consistency.

Thirty-three participants from diverse backgrounds, including data scientists, software

engineering students, Ph.D. students, and developers, were involved. All participants had a solid understanding of process and data analysis, with varying levels of familiarity with REST APIs, ranging from extensive knowledge to basic understanding. They were randomly assigned to either an *API-assisted group*, utilizing APIs to compose or edit emails, or a *control group*, employing a manual approach. Both groups received identical email samples, ensuring consistent task complexity.

- ***API-Assisted Group: Semi-Automated Approach***

In this testing scenario, participants engaged with the */response-template-suggestion* endpoint, aiming to streamline their email response generation process. The total time invested in crafting responses, denoted as  $T_{\text{total}}$ , was measured, where:

$$T_{\text{total}} = T_{\text{API}} + T_{\text{customization}}$$

Here,  $T_{\text{API}}$  represents the time taken to invoke the API, encompassing the duration from initiating the API request to receiving the suggested templates. A shorter  $T_{\text{API}}$  implies a more responsive API, contributing positively to the overall efficiency of the response generation process. On the other hand,  $T_{\text{customization}}$  reflects the time participants spent tailoring the suggested templates to meet their specific needs. A higher  $T_{\text{customization}}$  might indicate that initial template suggestions required significant adjustments, potentially impacting the ease of customization.

Alongside response time metrics, a user satisfaction survey involving 30 participants was conducted. Each participant contributed a satisfaction score on a scale from 1 to 10. This survey likely encompassed aspects such as the relevance of the API's suggested templates to participants' specific scenarios, the ease of use, and overall satisfaction with the response recommendation process. A higher score suggests a more positive user experience and satisfaction with the API's performance.

- ***Control Group: A Manual Approach***

In the control group, participants were tasked with manually composing email replies without the assistance of the */response-template-suggestion* endpoint. This required careful reading of received emails and crafting responses based solely on individual judgment and writing skills. The performance metrics for this group mirrored those applied to the API-assisted group, enabling a direct comparison.

The manual approach exhibited a notable delay in time efficiency. Participants in this group took an average of  $T_{\text{manual}} = 5$  minutes to complete their email replies. In contrast, their API-assisted counterparts completed the same task in an average time of  $T_{\text{API}} = 2.2$  minutes.

The control group scored an average of  $S_{\text{manual}} = 7$  out of 10 in user satisfaction. This rating was noticeably lower than the API-assisted group's satisfaction score. Participants in the manual group often found the process to be more tedious and time-consuming, adversely impacting their overall satisfaction.

The Average Time Reduction (%TR) is a crucial metric in evaluating the efficiency of automated email response generation compared to manual methods. It quantifies the percentage improvement in response time achieved by utilizing an API for generating email responses. The formula given by

$$\%TR = \frac{T_{\text{control}} - T_{\text{API}}}{T_{\text{control}}} \times 100\%$$

where  $T_{\text{control}}$  and  $T_{\text{API}}$  are the average response times in the control and API-assisted groups, respectively.

This formula determines the difference between the average response time for the automated group, where participants use the API for response production, and the manual group, where participants create emails without automated aid. The result is then expressed as a percentage of the average response time for the manual group. A positive %TR indicates that the automated method is more time-efficient, reflecting a reduction in the time it takes to generate email responses compared to manual efforts. Conversely, a negative %TR would imply that the manual method is faster. This metric provides valuable insights into the practical efficiency gains offered by automated email response generation.

Table 4.2 presents a comprehensive overview of the outcomes from the experimental evaluation that involved the use of an API-assisted approach versus a manual approach for generating email responses. The results from the table clearly demonstrate the superior performance of the API-assisted approach in comparison to the manual method for generating email responses. The *Avg Response Time* values reveal that, on average, participants in the API-assisted group were able to compose responses more quickly than their counterparts in the manual group. This efficiency is further emphasized by the *Avg Time Reduction (%TR)* column, which shows a notable percentage reduction in response time for the API-assisted group. Additionally, the *User Satisfaction Score* column highlights that participants using the API reported higher levels of satisfaction, suggesting that the API's suggested templates were deemed more relevant and effective by the users.

**Table 4.2:** Comparison Between Automated and Manual Email Approaches

Participants	Avg Response Time (s)	Avg Time Reduction (%TR)	User Satisfaction Score
33	3764	18.5	8.9

In summary, these findings underscore the efficacy and efficiency of leveraging the automated API for email response recommendation. Participants using the API, in other words, our developed approach, completed their tasks in less than half the time taken by those crafting emails manually.

## 4.4 Conclusion

In this chapter, we have achieved our second objective and provided an answer to the first research question (Q2) raised in the thesis problem. This is elaborated in Chapter 1, Section

1.3.2. The main inquiry we delved into was: **Can predictive techniques be utilized to recommend specific process-oriented emails?** To tackle this multi-faceted query, we have considered a range of sub-questions:

- **Q2-1:** *How can we effectively leverage the event log from previously exchanged emails to predict future BP knowledge that will be expressed in email responses?*
- **Q2-2:** *What are the most suitable machine learning algorithms or predictive models that can be employed to forecast this future BP knowledge?*
- **Q2-3:** *What types of event attributes should be considered when predicting process-specific email responses?*
- **Q2-4:** *How can we personalize the recommended email responses not only based on the process activity but also on the preferences and communication styles of individual participants?*
- **Q2-5:** *How can we validate the effectiveness and performance of the process-activity-aware email response recommendation system to ensure it provides meaningful and valuable email responses in response to received emails?*

In response to these questions, we have developed a process-activity-aware email response recommendation system composed of five phases:

1. Generating an event log from past emails,
2. Constructing business process-oriented prediction models using the event log to guide email responses addressing questions **Q2-1** and **Q2-2**,
3. Identifying activities and instances from incoming emails concerning question **Q2-3**,
4. Predicting relevant BP knowledge for the response email pertaining to question **Q2-4**, and
5. Recommending an email response template by analyzing textual content associated with the BP knowledge of the response.

Regarding question **Q2-5**, we conducted experiments using a public dataset from Enron and demonstrated promising results. Furthermore, we have publicly shared our findings, which, to our knowledge, are absent in related studies (that's why comparison with them was not feasible when evaluating our proposals).

We recognize that certain limitations may arise at distinct levels:

1. **Technical Implementation Level:** Technical constraints and challenges pertaining to the integration with existing platforms, real-time responsiveness, scalability, and privacy and security concerns are pivotal factors to address at this stage.

2. **Personalization Level:** While the system endeavors to accommodate individual communication styles, achieving genuine personalization necessitates the incorporation of extensive datasets comprising diverse individual communication patterns.
3. **Ethical Concerns Level:** Predicting and recommending email responses raises concerns regarding user privacy and data confidentiality. The implementation of such systems in real-world settings would demand stringent guidelines and ethical considerations.

In the next and the final chapter, we discuss potential perspectives and improvements of the overall work presented in this report.

# Conclusion & Perspectives

---

## Contents

---

<b>5.1 Contributions . . . . .</b>	<b>89</b>
<b>5.2 Perspectives . . . . .</b>	<b>91</b>
5.2.1 Enhance Predictive Models and Integrate Advanced NLP Techniques . . .	91
5.2.2 Implement Real-Time Conformance Checking and Explore Cross-Platform Applicability . . . . .	91
5.2.3 Address User Interface, Security, and Scalability Considerations . . . . .	92

---

In this chapter, we first summarize our contributions in this thesis. Then, we discuss our future research directions.

## 5.1 Contributions

This thesis addressed the complexities of BPM in email-driven processes, focusing on multi-perspective conformance checking and a process-activity-aware email response recommendation system. Traditional BPM systems often overlook the informal and unstructured nature of email communications, which are crucial for many business activities. Our work aimed to bridge this gap by implementing innovative methodologies tailored for email-driven processes.

Our contributions in the realm of multi-perspective conformance checking are notable. We proposed process models based on sequential and contextual constraints specified by a data analyst/expert, addressing both structural and contextual perspectives of email events. We developed algorithms to identify fulfilling and violating events, ensuring that email-driven processes adhere to a predefined model. This approach allowed for a more comprehensive evaluation of email processes by considering both the sequence of activities and the business context in which they occur.

The process-activity-aware email response recommendation system represents another significant advancement. By leveraging structured event logs, we predicted future BP knowledge, including the sequence of activities, their intent, and relevant business data. This system recommends email response templates based on predicted BP knowledge, enhancing the relevance and efficiency of email communications. Our methodology consists of several

phases, including BP knowledge extraction, building predictor models, and recommending email responses. These phases work together to ensure that the recommended responses are contextually appropriate and align with the predicted future activities.

To validate our approach, we utilized emails retrieved from the public Enron dataset and conducted a series of experiments covering various phases, parts, and steps of our overall framework. We shared the results to facilitate quantitative comparisons with future studies using the same dataset, a feature absent in existing approaches. Additionally, by integrating RESTful APIs, we streamlined communication with prediction and compliance methods, enhancing the accessibility and usability of our proposed solutions. This integration ensures that our methods can be easily adopted and implemented in various business environments, facilitating the management and optimization of email-driven processes.

The design principles we presented in the introduction (Chapter 1, Section 1.4) have been respected:

- **Context Sensitivity:** We emphasized analyzing and interpreting the business context of email conversations to ensure appropriate and relevant recommendations and conformance checks.
- **Interdisciplinarity:** We bridged the gap between various domains, including process mining, natural language processing, machine learning, and business management. By leveraging insights and methodologies from these fields, we provided a comprehensive approach to email-based business processes.
- **Consistency:** We ensured a uniform approach in predicting email responses and checking conformance. This involved developing standardized methods and algorithms that could be reliably applied across different scenarios and datasets, maintaining consistency in results and interpretations.
- **Automation:** We emphasized the automation of the recommendation and conformance checking processes. This aimed to develop models and systems that function with minimal human intervention, enhancing efficiency and reducing the likelihood of human error, thereby providing a more accurate and seamless experience.
- **Accessibility and Integration:** We designed solutions to be user-friendly and versatile enough for integration into various business environments. This principle emphasized creating tools that were robust yet easily accessible and integrable, achieved with the developed APIs.

In conclusion, this thesis provides a comprehensive framework for addressing the challenges of BPM in email-driven processes. Our multi-perspective conformance checking approach and process-activity-aware email response recommendation system offer robust solutions for improving the management and efficiency of email communications in business processes. The validation of our methods using real emails from the public Enron dataset demonstrates their practical applicability and effectiveness. This work not only contributes to the academic



field of BPM but also offers valuable tools and methodologies for businesses to optimize their email-driven processes.

## 5.2 Perspectives

In future work, we intend to: (i) Enhance predictive models and integrate advanced NLP techniques (Section 5.2.1), (ii) Implement real-time conformance checking and explore cross-platform applicability (Section 5.2.2), (iii) Address user interface, security, and scalability considerations (Section 5.2.3).

### 5.2.1 Enhance Predictive Models and Integrate Advanced NLP Techniques

This perspective includes two main axes. The first axis involves enhancing the predictive accuracy of email response recommendations by leveraging more advanced machine learning techniques, such as transformer-based models. These models have the potential to include user-specific behavioral patterns and preferences to improve personalization and relevance, as well as better capture the context and subtleties of email exchanges.

Advanced NLP techniques, like sentiment analysis and emotion identification, can be integrated to provide greater insights into the urgency and emotional tone of emails in the second axis. This would allow for more nuanced and context-aware response recommendations. Additionally, exploring NLP applications for summarizing lengthy email threads and extracting key action items could further streamline email management processes.

### 5.2.2 Implement Real-Time Conformance Checking and Explore Cross-Platform Applicability

To further automate and improve the system, additional research questions need to be explored:

1. **Real-Time Conformance Checking:** Developing real-time conformance checking systems that operate dynamically as emails are drafted and sent. This involves creating lightweight, efficient algorithms capable of performing quick checks without disrupting the user experience. Continuous learning mechanisms within the conformance checking system could allow it to adapt to evolving business processes and user behaviors over time.
2. **Cross-Platform Applicability:** Extending the methodologies to other communication platforms such as instant messaging, project management tools, and social media. This would provide a more comprehensive view of business process management across various channels and uncover new insights into communication patterns and process flows, enabling more holistic process optimization strategies.

The developed APIs can also be designed for integration with a variety of other platforms beyond email clients. By creating flexible and adaptable APIs, the solutions can be employed in diverse systems, enabling seamless interoperability and expanding the utility of the developed techniques across different domains and applications.

### 5.2.3 Address User Interface, Security, and Scalability Considerations

Future studies should focus on three main axes to address practical integration requirements:

1. **User Interface and Experience Enhancements:** Improving the user interface and experience for the email management system is crucial for adoption and usability. Research could explore designing more intuitive and user-friendly interfaces that seamlessly integrate predictive and conformance checking features. Conducting user studies to gather feedback on system usability and effectiveness could inform iterative improvements, ensuring the tool meets user needs effectively.
2. **Security and Privacy Considerations:** Ensuring the security and privacy of email data is paramount. This involves investigating how to store discovered business process knowledge at the email client level and securely share validated data to a central subsystem. Solutions could include implementing a hashing process and exploring the salting concept to add a layer of security to the hashing process, thereby protecting sensitive information.
3. **Scalability and Performance Optimization:** As email volumes continue to grow, ensuring the scalability and performance of the system is essential. Research could focus on optimizing algorithms and infrastructure to handle large-scale email datasets efficiently. Investigating distributed computing and cloud-based solutions could support scalability and provide robust performance under heavy workloads.

By addressing these areas, future research can build on the foundations laid in this thesis, driving further innovation and enhancing the practical utility of business process mining techniques in email management.

# A Novel Approach For Unsupervised Anomaly Detection In Time Series

---

## Contents

---

<b>A.1 Introduction</b> . . . . .	<b>94</b>
<b>A.2 Related Work</b> . . . . .	<b>95</b>
A.2.1 Statistical-based methods . . . . .	95
A.2.2 Proximity-based methods . . . . .	96
A.2.3 Clustering-based methods . . . . .	97
A.2.4 Deep learning-based methods . . . . .	98
<b>A.3 The Proposed Approach</b> . . . . .	<b>99</b>
A.3.1 Handlers: Buffered Data Retrieval . . . . .	100
A.3.2 Constructors: Adelson-Velsky and Landis Tree Constructor . . . . .	101
A.3.3 Analytics: Anomaly Score Calculation . . . . .	102
A.3.4 Detectors: Anomaly Type Detector . . . . .	104
A.3.5 Monitors: Environmental Changes Detector . . . . .	105
<b>A.4 Experiments and Validation</b> . . . . .	<b>105</b>
A.4.1 Model Performance Evaluation . . . . .	106
A.4.2 TBD as Pre-processing Engine . . . . .	109
<b>A.5 Conclusion</b> . . . . .	<b>110</b>

---

In this chapter, we are diverging from our primary focus, which has mainly revolved around BP Mining. Specifically, we have concentrated on process prediction and conformance checking in the context of email communication. Although the forthcoming content might appear tangential to our main thesis, it represents a curious exploration that has piqued our interest. Our attention now turns to anomaly detection within the dynamic realm of streaming data, with a specific emphasis on the context of the *Internet of Things* (IoT).

## A.1 Introduction

Anomaly detection is a sub-field of data mining that has attracted more and more attention with the advent of IoT systems [80, 31]. Several definitions of the anomaly, often referred to as outlier, can be found in the literature. Hawkins defines an outlier as an observation that deviates considerably from the rest of the other observations as if it were generated by a different process [60]. As for [36], they argue that anomaly detection involves modeling what is normal in order to find out what is not. [7] distinguishes between an outlier and an anomaly. The degree of aberrance helps differentiate noises from anomalies.

Anomaly detection improves data quality by removing or replacing the abnormal data. In other cases, the anomalies reflect an event and provide useful new knowledge. For example, the detection of anomalies can prevent material damage and therefore encourage predictive maintenance in industry. It finds application in several other areas such as health, cybersecurity, finance, natural disaster prediction, and many other areas.

Data exists in many forms: static data, data flows, structured and unstructured data, etc. Each type of data is relevant in one or more areas. The multitude of data types and their different characteristics mean that there are different methods for detecting anomalies, each of which is effective in a particular area, with a given purpose. These methods generally use a decision threshold to isolate anomalies based on different techniques such as classification, clustering, regression, nearest neighbors, and statistical tools.

As part of our study, we are interested in outlier detection methods for handling data streams, especially for IoT time-series data. The majority of current anomaly detection methods (e.g., [100, 58, 94]) are very specific to the individual use case and require in-depth knowledge of the method as well as the situation to which it is being applied. IoT as a rapidly expanding field offers plenty of opportunities for this type of data analysis to be implemented. However, due to the nature of IoT, many challenges are raised. The IoTs are often used in real-life settings, hence, one should take into consideration factors like environmental changes (e.g., variation in the occurrences of some climatic factors) and resources limitations.

Currently, many anomaly detection methods have difficulties detecting anomalies in streaming data in an automatic manner. Most of the available approaches are window-based [44, 74, 54] which often face the problem of reflecting the actual distribution of the data since these techniques only focus on a fixed data size to detect anomalies. In addition, most mechanisms depend on thresholds that need to be manually updated whenever environmental changes occur. Others are designed to use all features of the data which are not always applicable in a streaming context such as IoT.

This chapter presents Track Before Detect (TBD), a novel approach that helps in detecting anomalies in IoT time-series data. The proposed approach overcomes the aforementioned limitations by automatically differentiating between anomalous behavior and environmental changes. An environmental change is a change or disturbance in the environment, most often caused by human influences or natural ecological processes (e.g., transitioning from

Spring to Summer). An anomalous behavior, on the other hand, is defined as a pattern that does not conform to the regular behavior of an IoT sensor device. Anomalous behavior may be manifested differently under different environmental conditions. For instance, a 40 degrees temperature is considered normal during hot seasons but anomalous during cold seasons. Differentiating between anomalous and environmental changes is therefore crucial for increasing the accuracy of anomaly detection systems. Once environmental changes are detected, TBD can automatically adapt without the need to manually set different anomaly thresholds for different contexts.

In addition, TBD can be used as a pre-processing engine for unsupervised deep learning-based models to enhance their performance. To the best of our knowledge, TBD is the first approach which is capable of differentiating between anomalous behavior and environmental changes in time series data in an unsupervised setting and without affecting the running system. The proposed pipeline is flexible and can be easily adapted for different use cases and domains. The rest of this chapter is structured as follows: Section A.2 provides an insightful overview of related work within the literature; Section A.3 introduces the intricate details of TBD, shedding light on its components and mechanisms; Section A.4 takes a closer look at the experiments conducted and the subsequent discourse on the achieved results; and Section A.5 brings the chapter to a close, offering concluding remarks and outlining prospective paths for future endeavors.

## A.2 Related Work

As outlined above, there are a number of different methods for detecting anomalies. Obviously, the choice of algorithm depends on the use case, the context, the type of data available, and many other parameters. Among these methods, we distinguish: statistical-based methods (Section A.2.1), proximity-based methods (Section A.2.2.1), clustering-based methods (Section A.2.3), and deep learning-based methods (Section A.2.4).

### A.2.1 Statistical-based methods

The statistical-based methods consist in developing flexible probabilistic statistical models that represent the distribution of the data sets tested such as Gaussian models [109] and regression models [6, 73]. The degree of anomaly of a particular object is evaluated against its conformity to the established model. Particularly, in [109], a Gaussian mixture model is proposed to represent the distribution of the tested data. Each object receives an anomaly score which characterizes its deviation from the model. A high score indicates a high probability that the object in question is an anomaly. These methods are very efficient, mathematically justified, and can reveal the meaning of the outliers found when a probabilistic method is given. However, the Internet of Things is often used in real-life settings, where there is often no previous sensor data distribution knowledge; hence these methods are not beneficial.

## A.2.2 Proximity-based methods

The Proximity-based methods determine for an observation  $o$  its  $k$ -nearest neighbors (KNN) by calculating the distance between all the observations in the data set. These methods require a preliminary calculation and, therefore, they are costly in execution time. There are two approaches based on nearest neighbors: the distance-based approach [12, 109] and the density-based approach [18].

### A.2.2.1 Distance-based approach

Distance-based anomaly detection methods operate on a fundamental premise: anomalies are observations that diverge significantly from other observations in a data set. This concept of "distance" or "disparity" between data points is pivotal and can be articulated through various metrics. The two most commonly employed metrics are the Euclidean and Manhattan distances. The Euclidean Distance, widely adopted in distance-based algorithms, measures the straight-line distance between two points. For two points  $P_1(x_1, y_1)$  and  $P_2(x_2, y_2)$  in a two-dimensional space, this distance is defined as:

$$d(P_1, P_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (\text{A.1})$$

This concept naturally extends to multi-dimensional spaces. On the other hand, the Manhattan Distance, often referred to as the L1 norm or taxicab distance, evaluates the distance between two points as the sum of the absolute differences of their respective coordinates. In a two-dimensional space, for points  $P_1(x_1, y_1)$  and  $P_2(x_2, y_2)$ , the formula becomes:

$$d(P_1, P_2) = |x_2 - x_1| + |y_2 - y_1| \quad (\text{A.2})$$

Building upon these distance metrics, various techniques have been developed for practical anomaly detection. One of the most prominent techniques is the KNN method, which computes the distance between a data point and its  $k$  nearest neighbors within a data set. A substantial average distance from these neighbors usually suggests that a data point might be an anomaly. An advanced version, the Weighted KNN, assigns varied weights to the neighbors based on their proximity, offering more nuanced anomaly detection capabilities.

When it comes to evaluating the distinctiveness of data points, anomalies often receive higher scores due to their considerable distance from the majority of data points. In contrast, typical observations, being closely packed, receive lower scores. Post the score assignment, anomalies can be isolated by organizing these scores in descending order and cherry-picking data points with the topmost scores.

Despite their versatility, distance-based anomaly detection methods come with their own set of advantages and limitations. A significant boon is their non-parametric nature, implying

they don't rely on any predetermined distribution for the data. Yet, a notable challenge arises when dealing with continuous data streams, where these methods can grapple due to the computational heaviness and the dynamic nature of data distributions.

### A.2.2.2 Density-based approach

Density-based methods are pivotal in understanding the neighborhood relationships within a data set. The local density of data points is often associated with their potential of being anomalies. This means that if a point's local density differs significantly from its neighbors, it's more likely to be an outlier.

The Local Outlier Factor (LOF) algorithm is a quintessential example of a density-based anomaly detection method [19]. The underlying concept of LOF takes its cues from the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm, which is adept at identifying dense clusters of points along with discerning outliers that do not belong to any cluster [45]. Within the scope of the LOF algorithm, an object's local density is estimated based on its distance to its KNNs. This is achieved by considering a set of distances from a data point to its respective KNNs and using it as a foundation to calculate its local density. Subsequently, the local densities of all the objects in the data set are juxtaposed to identify regions with consistent densities and, more importantly, the outlier points which have considerably lower local densities than their neighbors.

Furthermore, the domain of density-based anomaly detection hasn't limited itself to just the LOF. Several adaptations and extensions of the method have surfaced over the years. Variants such as Connectivity-based Outlier Factor (COF) [98], Local Outlier Probabilities (LOP) [69], and the incremental LOF [87] come with their unique twists, making them suitable for specific scenarios or challenges.

However, despite their robustness in anomaly detection, density-based methods present significant challenges in scalability and efficiency. Specifically, as new data is incorporated into the data set, the algorithm doesn't merely evaluate the local density of the new data point. It also recalculates the local densities of its neighbors. This cascading effect of recalculations, especially in large data sets, leads to substantial computational costs. Moreover, the non-updatability of the outlieriness measurements in these methods poses additional challenges, especially when working with data streams. In such dynamic environments, efficient updates and real-time processing are crucial, and the intricate nature of these density-based techniques might impede their efficacy.

### A.2.3 Clustering-based methods

The main purpose of clustering methods is to divide the data set into clusters containing the data that has similar behaviors. The key assumption is that normal observations belong to large, dense clusters, while anomalies do not belong to any cluster or belong to small isolated

clusters.

The k-means method initially used in [65] or a more robust version with respect to outliers, the k-medoid method [59], constitute classical methods of clustering applicable for the detection of anomalies which are based on the distance between observations relative to the variables of interest. However, these methods require initially setting the number of clusters and intrinsically only consider the distance between observations. It is then possible to rely on methods based on the density of observations such as DBSCAN proposed in [45] or Ordering Points to Identify the Clustering Structure (OPTICS) developed in [13]. Unlike DBSCAN, the OPTICS method can detect clusters of different densities. However, faced with the sometimes-difficult construction of clusters with the OPTICS method, other methods have emerged, such as Local Density-based Spatial Clustering of Applications with Noise (LDBSCAN) [46] which is based on the evaluation of a LOF.

Some of the aforementioned clustering algorithms work on data sets of fixed size, while others work on data streams in which new observations arrive periodically. In the first case, the clusters found fully represent the data set. In the second case, a group of clusters represents the data at a given moment and a method of devaluing the clusters as a function of time must be implemented. Examples of clustering techniques dealing with data streams are Clustream [8], High-dimensional Projected Stream (HPStream) [9] and STREAM [82].

Nevertheless, the objective of clustering-based methods is to only group the objects in any given data set. Thus, many researchers argue that clustering algorithms should not be considered as outlier detection methods. In addition, clustering-based methods are designed to use all the features of data in detecting outliers and the notions of outliers in the context of clustering are essentially binary in nature, without any quantitative indication as to how outlying each object is. It is desired in many applications that the outlines of the outliers can be quantified and ranked.

#### A.2.4 Deep learning-based methods

Deep learning (DL) has evolved as a fundamental tool in the field of anomaly detection [55, 25]. These DL methods can be broadly grouped into supervised and unsupervised categories. In the supervised context, the problem is translated into a binary classification task. Here, the primary objective is to categorize each data point as either an anomaly or not. This scenario's inherent challenge is that the anomalies constitute a minority class, often a meager fraction of the overall data set. While these methods sound intuitive, a significant impediment is their dependence on accurate labels for both the normal and anomalous instances. This kind of labeled data is often a luxury in real-world time series scenarios, which makes supervised techniques less feasible.

On the other hand, unsupervised methods appear more promising for anomaly detection. Here, the algorithms learn from the data holistically, without leveraging any prior labels. Essentially, the methods discern patterns and nuances from the data and attempt to identify outliers based on these learned patterns, without any predefined understanding of what



constitutes an anomaly. A popular unsupervised technique leveraged in this space is the Auto-Encoders (AE) [11].

The foundational concept of an AE comprises two components: an encoder and a decoder. The encoder's role is pivotal in data compression. It ingests data from a high-dimensional space ( $N$  dimensions, for instance) and condenses it into a lesser-dimensional space. This compressed representation captures the essential features of the data. Conversely, the decoder takes up the challenge of expanding this compressed data, trying to reconstruct it back to its original  $N$ -dimensional form. A key metric to gauge the success of this architecture is to compare the original data (input to the encoder) with the reconstructed data (output from the decoder). A smaller reconstruction error is indicative of the AE's efficiency in preserving the salient features during the compression and decompression processes.

Nevertheless, while AEs are both simplistic and potent in outlier detection, they are not without challenges. One pressing issue is their vulnerability to noisy training data. If the data used to train the AE has inherent noise or inaccuracies, the AE's performance in anomaly detection could suffer, as the model might misinterpret noise as essential features during the encoding process.

### A.3 The Proposed Approach

Given incoming IoT data streams as input, our proposed approach returns, for each newly captured data point, its anomaly score as well as the anomaly type. The anomaly score indicates how anomalous the incoming data point is compared to the set of data points collected within a specific time frame. If considered anomalous, our approach can detect the type of anomaly, which can be one of the following: i) point anomaly, ii) probable collective anomaly, and iii) collective anomaly. A point anomaly refers to an individual data instance that is anomalous. A probable collective anomaly is a warning flag indicating that the system may soon face a collective anomaly, which represents a group of data points that together exhibit anomalous behavior. Additionally, our approach can detect whether an environmental change has occurred and can automatically adapt to these changes without affecting either the running system or the anomaly detection process.

Figure A.1 provides a high-level view of the various components in the proposed solution. The functions used in each of the TBD components can be divided into three categories: the *Repetitive Processing Function* (RPFs), which are functions that run at regular intervals. The functions implemented in the monitor component are considered RPFs. The *On-Demand Functions*, such as those related to the constructor component, run instantly as and when needed. Finally, the *Step Functions* run successively and are applied to newly captured IoT data. The step functions include those used in the input, analytic, detector, and output components.

In the following subsections, the role of each component and how it fits into the overall architecture is described.

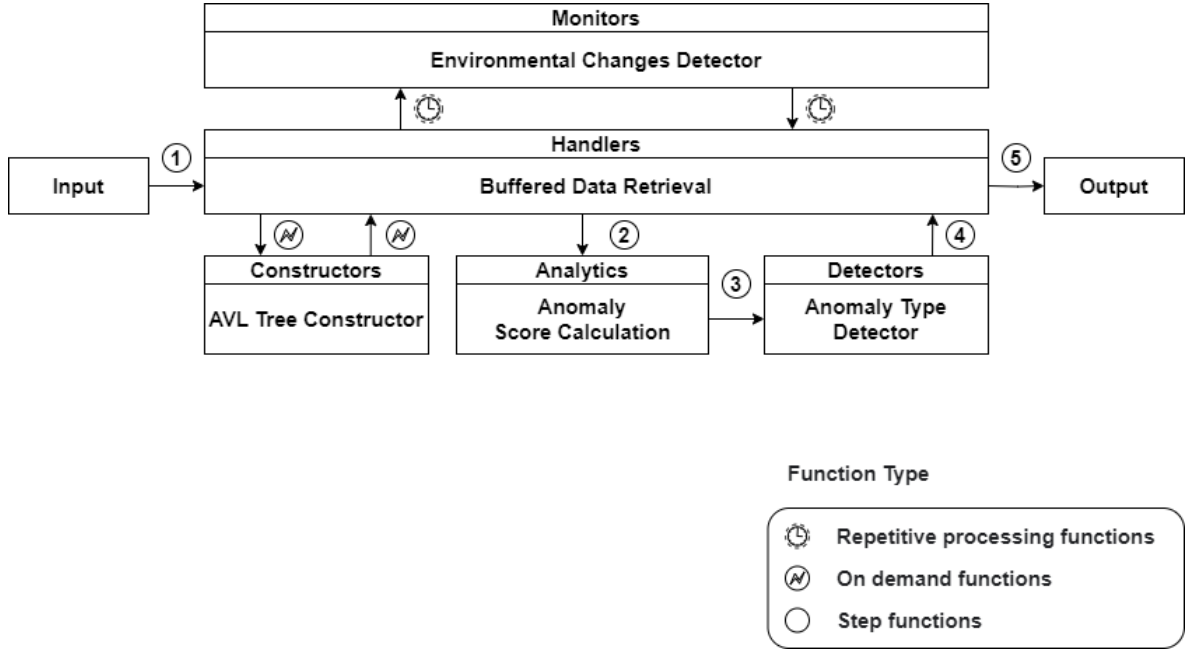


Figure A.1: TBD diagram components

### A.3.1 Handlers: Buffered Data Retrieval

Data handling is an essential aspect of IoT systems due to the sheer volume of data generated by IoT devices. Handlers, in this context, are mechanisms that assist in the management and processing of incoming IoT data. The principal benefit of employing handlers lies in facilitating efficient data exchange among various components of our approach. By leveraging handlers, one can reduce the frequency of function calls, thereby optimizing computational resources and improving overall system performance.

A salient feature of this handler is the use of a Buffered Data Retrieval (BDR) mechanism. At its core, the BDR allows for the temporary storage of incoming IoT data. The logic is based on a principle similar to batching: instead of continually collecting data from a single IoT sensor every time it is generated, the BDR mechanism buffers multiple data points and allows them to be retrieved collectively at a specified time. Formally, if  $f(t)$  represents the function that retrieves data at time  $t$ , without buffering,  $f(t)$  is called every instance. With buffering,  $f(t)$  may be called less frequently, say at  $t, t + \Delta t, t + 2\Delta t$ , etc., where  $\Delta t$  is the time interval between retrievals. The buffered data can be represented as an array or a list:  $B = [d_1, d_2, \dots, d_n]$ , where  $d_i$  is the data at the  $i$ -th interval.

The aforementioned BDR mechanism communicates with TBD. When TBD requests data, instead of directing the system to collect fresh data, it communicates with the BDR to retrieve the buffered data. These buffered data are then utilized in various TBD components.

### A.3.2 Constructors: Adelson-Velsky and Landis Tree Constructor

Efficiently storing and processing data from IoT devices presents significant challenges due to the vast amount and rapid rate of data generation. In response to this challenge, we utilize the Adelson-Velsky and Landis (AVL) tree [93] to manage the overwhelming influx of data. However, before delving into the specifics of the AVL tree, it's crucial to first understand its foundational structure: the Binary Search Tree (BST). Within a BST, each node may have up to two children, commonly known as the left and right child. A defining characteristic of a BST is that every node's left sub-tree contains elements smaller than the node itself, while the right sub-tree contains elements that are larger. This organization ensures that operations like searching, insertion, and deletion are executed efficiently.

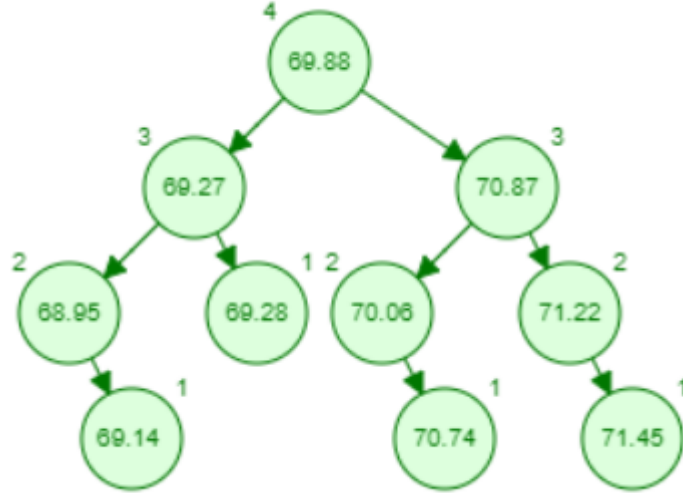
The AVL tree, a refined variation of the BST, introduces a self-balancing mechanism. In contrast to a typical BST, which can become skewed after multiple data insertions or deletions, the AVL tree maintains a balance by ensuring the height difference between the left and right sub-trees of any node is never more than one. This balancing act leads to quicker data access times. Each node within an AVL tree represents a specific data point, with nodes in our application representing individual IoT data points.

To provide a clearer understanding, we have prepared a sample set of buffered data generated by an IoT temperature monitor device. This data, and its representation in the AVL tree format, can be examined in Table A.1 and Figure A.2, respectively.

**Table A.1:** Subset of the first  $q$  buffered data from IoT temperature monitor device

Timestamp	IoT Data
2013-07-04 00:00	69.88
2013-07-04 01:00	71.22
2013-07-04 02:00	70.87
2013-07-04 03:00	68.95
2013-07-04 04:00	69.28
2013-07-04 05:00	70.06
2013-07-04 06:00	69.27
2013-07-04 07:00	69.14
2013-07-04 08:00	71.45
2013-07-04 09:00	70.74

The tree is initially constructed with the first arrived  $q$  data points, which are by default considered as the *normal behavior*. The construction of an AVL tree follows the same process as for BSTs in which each internal node represents a data point whose value is greater than all the data points' values in the node's left sub-tree and less than those in its right sub-tree. After each node insertion, the BF for each of the ancestors of the inserted node is calculated. The BF of a node  $X$  is defined as the height difference between its two child sub-trees as given in Equation A.3.



**Figure A.2:** AVL tree constructed out of the data points in Table A.1

$$BF(X) := Height(RightSubtree(X)) - Height(LeftSubtree(X)) \quad (A.3)$$

where *RightSubtree* and *LeftSubtree* are two functions that return the right and left child sub-trees of a node  $X$ . The BF of each node should be -1, 0, or 1. If the BF is less than -1 or greater than +1, the sub-tree rooted at this node is unbalanced, and a rotation is needed. The tree rotation is widely used in balanced trees in general because it allows for reducing the height of a tree by lowering the small sub-trees and raising the large ones, which makes it possible to re-balance the trees.

The obtained AVL tree represents the *normal behavior* to which the subsequent incoming data are compared as explained in Section A.3.3. The advantage of using AVL trees for detecting anomalous behavior is also explained in Section A.3.3. It is worth noting that the tree is updated only when an environmental change is detected (as explained in Section A.3.5). This allows us to maintain the representation of the normal behavior without being limited to a fixed window size.

### A.3.3 Analytics: Anomaly Score Calculation

As explained in the previous section, we use an AVL tree to store the normal behavior represented by the first arrived  $q$  data points. For each newly arrived data point, an anomaly score is computed. The anomaly score reflects the degree to which the new data is dispersed from the normal behavior in the tree. Dispersion is a statistical term that describes the size of the distribution of values expected for a particular variable. Dispersion can be measured by several different statistics, such as range, variance, and standard deviation. In our work,

we use the Standard Deviation (SD) [71], however, any other statistical measure can be used. The standard deviation is defined as the square root of the variance or, equivalently, as the root mean square of the deviations from the mean. Given a set of points  $S$ , the formula for SD is shown below where  $x_i \in S$  is a value in the data set,  $\mu$  is the mean of the data set, and  $N = |S|$  is the number of data points in the population.

$$\sigma = \sqrt{\frac{\sum(x_i - \mu)^2}{N}} \quad (\text{A.4})$$

The usage of an AVL tree allows us to compute the SD in a global and local way. Global SD is computed between the new data point and all points in the AVL tree (i.e., SD w.r.t the overall behavior). Local SD is computed between the new data point and the points in the right and left sub-trees of the root node. This allows us to accurately reflect the dispersion of the newly arrived data point from the normal behavior. The global and local SD are then used to compute an overall anomaly score that compares their closeness. In the following, we present some notations that allow us to formally define the global and local SD and the anomaly score.

**Definition A.1. (Global set  $S$ , local sets  $S_R$  and  $S_L$ )** Let  $T$  be an AVL tree. We denote by  $T.root$  the root node of  $T$ . We define  $traverse(T)$  as a function that does a traversal of an AVL tree and collects the nodes in a Set  $S$ . Let  $T_R$  and  $T_L$  be the right and left sub-trees returned by  $RightSubtree(T.root)$  and  $LeftSubtree(T.root)$ , respectively.  $traverse(T_R.root)$  collects the nodes of the right sub-tree  $T_R$  in a set denoted as  $S_R$  and  $traverse(T_L.root)$  collects the nodes of the left sub-tree  $T_L$  in a set denoted as  $S_L$ .

The traversal of the AVL tree can be done in different ways. The commonly used methods are: the in-order traversal where the left sub-tree is visited first, then the root, and later the right sub-tree. The pre-order traversal in which the root node is visited first, then the left sub-tree, and finally the right sub-tree. As for the post-order traversal, first we traverse the left sub-tree, then the right sub-tree, and finally the root node. In our work, we use the in-order traversal. Technically, the traversal returns a list (in which the elements are sorted). Since our approach does not require the returned elements to be sorted, we store them in a set.

In the illustrative scenario depicted in Figure A.2, the operation  $traverse(T)$  yields the set  $S$ , comprising the following elements: {68.95, 69.14, 69.27, 69.28, 69.88, 70.06, 70.74, 70.87, 71.22, 71.45}. Similarly, when applied to the root of the left sub-tree  $traverse(T_L.root)$ , it generates the subset  $S_L$  containing: {68.95, 69.14, 69.27, 69.28}. Conversely, for the right sub-tree's root  $traverse(T_R.root)$ , the resulting set  $S_R$  encompasses: {70.06, 70.74, 70.87, 71.22, 71.45}.

**Definition A.2. (Global  $\sigma_G$  and local  $\sigma_R, \sigma_L$  SDs)**

Let  $T$  be an AVL tree and  $\delta$  be a new incoming IoT data. The global SD, denoted as  $\sigma_G$ , is the SD computed over the data points  $S \cup \{\delta\}$ . Two local SDs are defined, right SD denoted

as  $\sigma_R$  and left SD denoted as  $\sigma_L$ .  $\sigma_R$  is computed over  $S_R \cup \{\delta\}$ , while  $\sigma_L$  is computed over  $S_L \cup \{\delta\}$ .

The anomaly score is computed by subtracting the result of  $\sigma_G$  from the summation of  $\sigma_L$  and  $\sigma_R$ .

$$anomalyScore = |\sigma_G - (\sigma_L + \sigma_R)| \quad (A.5)$$

The intuition behind the anomaly score defined in Equation A.5 is that normal behavior is manifested by a data point which has a low SD to at least one of the right and/or left sub-trees. This results in a global SD that is close to the sum of the local SD, and therefore a low anomaly score. On the other hand, anomalous behavior is manifested by a data point that has a very high SD in the global SD. Therefore, a high anomaly score is obtained. We consider as a standard threshold for anomalous data, any anomaly score that is greater than one.

**Table A.2:** Example showing the result of TBD after processing 4 incoming IoT data

Incoming IoT Data	Timestamp	Type	$\sigma_L$	$\sigma_R$	$\sigma_G$	Anomaly Score
63.3868	7/7/2013 20:00	No Anomaly Detected	0.8323	0.6296	1.7392	0.2771
73.9799	7/7/2013 21:00	Anomalous Data	4.0872	3.6284	9.9119	2.1961
60.6747	7/7/2013 22:00	Anomalous Data	1.0325	0.8644	3.8745	1.9775
62.4807	7/8/2013 0:00	No Anomaly Detected	1.0872	0.9284	2.9119	0.8961

Table A.2 illustrates an example of four IoT processed data points and their anomaly score computed according to Equation A.5. As shown in the table, when the anomaly score is greater than 1, the incoming data will be considered anomalous.

### A.3.4 Detectors: Anomaly Type Detector

If the anomaly score of the incoming data point  $\delta$  indicates an anomalous behavior (i.e., anomaly score  $> 1$ ), our approach detects whether the anomaly is one of the following three types: Point Anomaly (PA), Probable Collective Anomaly (PCA), and Collective Anomaly (CA).

**PA** refers to an individual data instance that is abnormally different from the rest of the data. An anomalous point is considered as PA if no previous anomalies have been detected for a  $k$  period of time. Once a PA is detected, a counter should run to count the occurrence of incoming PA data. The counter keeps increasing while the average time of arrival of PA data is close to each other (according to an average arrival time threshold  $t$ ). Otherwise, it is reset to 0.

**PCA** indicates that the running system may soon face a collective anomaly. If the counter of PA is greater than a user-specified threshold  $p$ , then the newly detected anomalous data should be classified as a PCA.

**CA** refers to a group of data points that are abnormally different from the rest of the data. After detecting a PCA, any successive anomalous data will lead to classifying all values between the first point detected as PCA to the last detected anomalous data as a CA.

### A.3.5 Monitors: Environmental Changes Detector

In order to differentiate between anomalous behavior and environmental changes, we implemented a background function that, for every period  $k_e$ , captures  $q$  incoming IoT data points. We then compute the SD between the captured  $q$  points and the data points in the AVL tree. If we get a high SD, this will be flagged as a warning. Hence, we need to check whether it is indeed an environmental change or simply an anomalous system behavior. To do so, we keep capturing different groups of  $q$  incoming data points within a specified time frame. If the data values in all captured groups are within the same range of values, we conclude that an environmental change has occurred and that the AVL tree needs to be updated to reflect this change. The update process requires first destroying the current tree and following the same steps mentioned in Section A.3.2 to the end, taking into consideration the newly captured data.

## A.4 Experiments and Validation

The approach has been implemented as a Python code<sup>1</sup>. The performance of TBD is evaluated using five real-world datasets<sup>2</sup>. For instance, one of these datasets includes a CPU usage dataset from a server in Amazon’s East Coast data-center. The dataset ends with a complete system failure resulting from a documented failure of Amazon Web Services (AWS) API servers. Another one is a temperature sensor dataset of an internal component of a large industrial machine in which successive anomalies led to a catastrophic failure of the machine. The primary reason for selecting these five datasets for assessment is the availability of a ground truth of anomaly labels, which are generally not available in publicly accessible streaming datasets.

We performed two experiments to evaluate our approach. In the first one (Section A.4.1), we compared the accuracy of our approach with existing state-of-the-art approaches. In the second experiment (Section A.4.2), we showed the accuracy gain obtained by using our approach as a pre-processing engine on top of an unsupervised deep learning anomaly detection method.

---

<sup>1</sup><http://www-inf.telecom-sudparis.eu/SIMBAD/tools/TrackBeforeDetect>

<sup>2</sup><https://www.numenta.com/resources/htm/numenta-anomaly-benchmark/>

### A.4.1 Model Performance Evaluation

To assess the accuracy of our approach, we relied on the following statistical measures: sensitivity, which refers to the ratio of correctly identified normal data to the total actual normal data; specificity, referring to the ratio of correctly classified anomalous data to the total actual anomalous data; precision, which refers to the correctly classified normal data out of the actual normal data; and accuracy, which helps determine how close the measurements are to the actual value.

For the evaluation, we took the most commonly used approach in each of the existing anomaly detection methods. Overall, we evaluated five different datasets on ten existing approaches. In Table A.3, we present the results for only one of the evaluated datasets: the ambient temperature in an office setting. Additional figures, tables, and information regarding the evaluated datasets can be found on our web page.

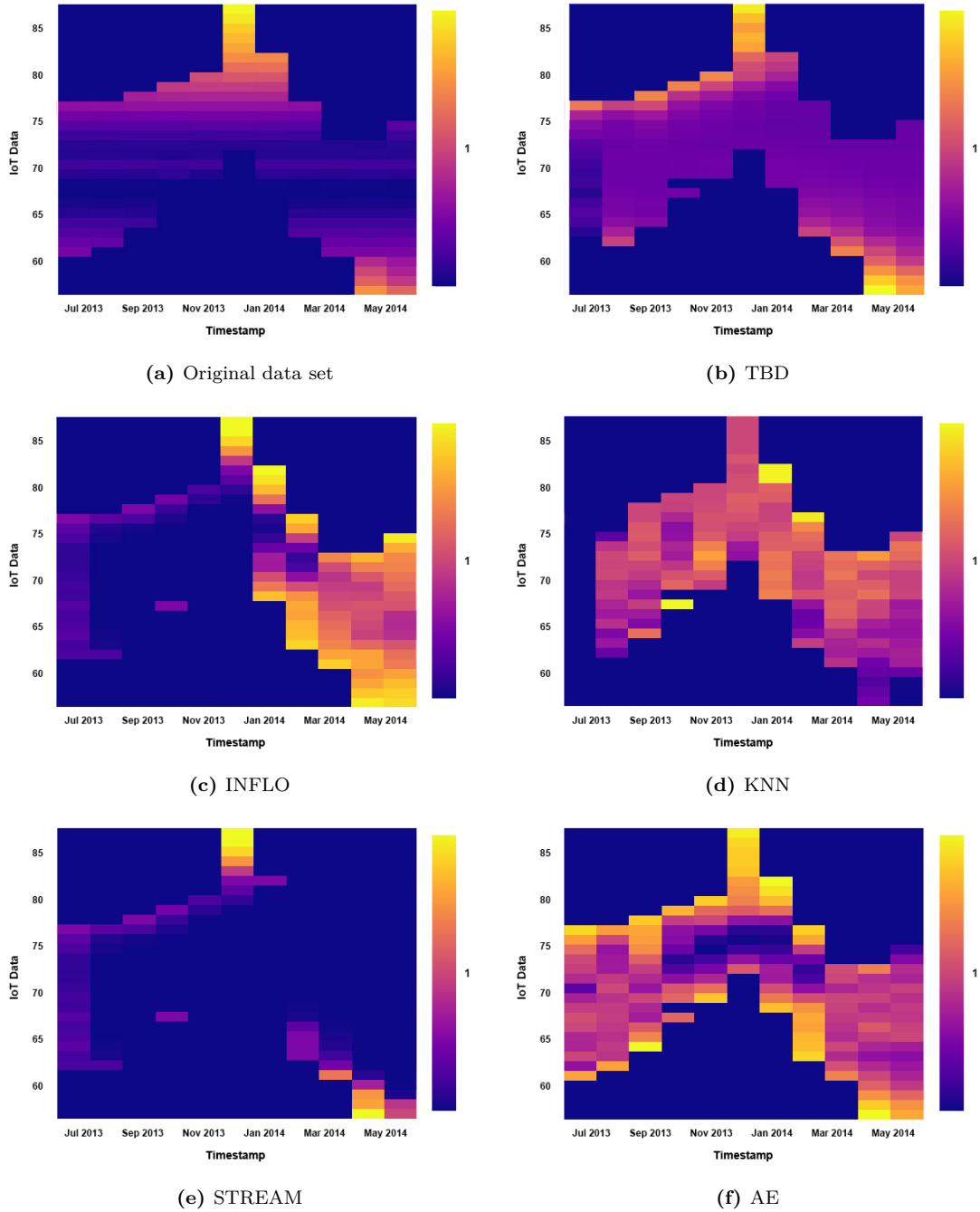
**Table A.3:** Comparative study of TBD with existing approaches

Method	Approach	Sensitivity	Specificity	Precision	Accuracy
Statistical-based	Gaussian Model	0.91	0.54	0.75	0.77
	Regression Model	0.89	0.50	0.72	0.73
Distance-based	Grid-ODF	0.82	0.51	0.59	0.66
	KNN	0.88	0.43	0.63	0.67
Density-based	LOF	0.79	0.49	0.59	0.64
	INFLO	0.86	0.34	0.43	0.53
	MDEF	0.77	0.49	0.57	0.62
Clustering-based	STREAM	0.94	0.58	0.77	0.79
	WAVECLUSTER	0.89	0.50	0.72	0.73
Unsupervised DAD	Autoencoders	0.96	0.61	0.77	0.81
Our approach	Track Before Detect	0.99	0.86	0.94	0.95

Overall, the study of individual results illustrates some interesting situations that occur in IoT real-time applications and how different methods behave. For totally sedentary work, a temperature ranging from 23 to 24 °C is strongly recommended in offices. But for a job that requires a minimum of physical activity, lowering the thermometer to 19 °C will be more appropriate. This preliminary study helps in setting the threshold that would help in detecting anomalous data. However, depending only on a threshold would not yield precise results, especially when environmental changes occur. For example, the temperature in offices could vary according to the season. Hence, an anomaly detection method should take into consideration the changes and readjust itself accordingly. The statistical, distance, and density-based methods could not adapt themselves to the environmental changes. Therefore, they continued to generate false anomalies for several days after the change had occurred. This false classification will affect the specificity of the concerned model as shown in Table A.3.

To closely inspect the false classification done by existing approaches, we plot the calculated anomaly score as a heat map against both the IoT incoming data and the timestamp as shown in Table A.4. We see that after January 2014 where a major change in the environment



**Table A.4:** Calculated anomaly score as a heat map against both IoT incoming data and the timestamps for six approaches

occurs, existing approaches such as KNN, INFLO and AE couldn't keep on correctly classifying incoming IoT data. The distance and density-based methods turn out to be a bad choice for anomaly detection due to their moderate results and very high time complexity. Even if the methods based on clustering have provided more or less satisfactory results, one cannot deny the manual effort required by an expert to manage such approaches since it only splits

observations into groups according to the similarity among them. In addition, these techniques have difficulty in detecting the projected outliers. Therefore, a system failure cannot be avoided. As for the AE, it performed well, but TBD outperformed this one.

Since TBD can differentiate between point and collective anomaly, we analyzed the behavior in the dataset where TBD detects successive point anomalies followed by successive collective anomalies. We filtered out all data instances that do not exhibit such behavior. In the first experiment, we analyzed the positive predictive value (i.e. precision) of TBD against existing approaches by considering every anomaly point detected correctly (whether point anomaly or collective anomaly) as TP. As shown in Table A.5, the accuracy of TBD in detecting collective anomaly behavior outperforms existing approaches. Since existing approaches do not explicitly differentiate between a point and a collective anomaly, they cannot accurately detect all point anomalies within a collective anomaly region.

**Table A.5:** Accuracy of TBD against existing approaches computed on instances where TBD detects a point anomaly followed by a collective anomaly. Instances detected as anomalous (point or collective) are considered as TP

	Gaussian	Regression	Grid-ODF	KNN	LOF	STREAM	INFLO	AE	TBD
<b>TP</b>	140	85	137	83	73	164	170	151	191
<b>FP</b>	66	121	69	123	133	42	36	55	15
<b>Precision</b>	0.679	0.412	0.665	0.402	0.354	0.796	0.825	0.733	0.927

To closely inspect the limitation of existing approaches in detecting collective anomaly behavior, we recomputed the performance metrics by considering every point anomaly detected correctly as  $TP_{PA}$ , every collective anomaly point detected correctly as  $TP_{CA}$  and every non-anomalous point detected correctly as  $TP_{NA}$ . For existing approaches, we label each point detected as anomalous with either point anomaly or collective anomaly according to the result of TBD on the same point. The results are illustrated in Table A.6. Overall, TBD is more accurate than all existing approaches as the computed accuracy values suggest. All existing approaches except TBD have a very low sensitivity. This means that existing approaches are generating many wrong predictions. More specifically, many points are predicted as normal while they are actually either point or collective anomaly points. This again demonstrates the limitation of existing approaches in detecting collective anomaly behavior.

**Table A.6:** Performance metrics computed on instances where TBD detects a point anomaly followed by a collective anomaly. Point anomaly is considered as TP, collective anomaly is considered as TN

Approach	Sensitivity	Precision	Accuracy
<b>Gaussian Model</b>	0.16	0.43	0.68
<b>Regression Model</b>	0.13	0.74	0.41
<b>Grid-ODF</b>	0.24	0.91	0.67
<b>KNN</b>	0.12	0.65	0.40
<b>LOF</b>	0.12	0.78	0.35
<b>STREAM</b>	0.21	0.30	0.80
<b>INFLO</b>	0.37	0.83	0.83
<b>AE</b>	0.25	0.70	0.73
<b>TBD</b>	0.64	0.78	0.93

As for precision, Grid-ODF and INFLO seem to be correctly detecting point anomaly and collective point anomaly. However, because of the very low sensitivity, they are missing many others. TBD on the other hand is making a good trade-off between precision and recall.

#### A.4.2 TBD as Pre-processing Engine

Recent research has revealed that few studies distinguish between noise and anomalies and investigate their interaction effects on detection results [116]. However, one cannot deny that the performance of unsupervised deep learning anomaly detection methods such as auto-encoders gets degraded due to noisy data.

Based on our observation, we found that the noisy data can be in fact eliminated from the processed dataset using our TBD approach. To do so, we defined the following formula that calculates a noisy data threshold.

$$\text{noisyDataThreshold} = \frac{\max(n_1, n_2, \dots, n_m)}{2} + 1 \quad (\text{A.6})$$

Where  $n_i$  denotes the calculated anomaly score of each of the processed IoT data by TBD. Therefore, all values that are greater than the obtained threshold will be considered as noise and will be removed.

After removing the noise, we fed the processed dataset to the deep learning model. We then used four fully connected layers with 14, 7, 7, and 29 neurons respectively. The first two layers are used for our encoder, the last two go for the decoder. Additionally, L1 regularization was used during training. After training our model, we then performed an evaluation of the results.

**Table A.7:** Comparative table showing the efficiency of TBD on top of the AE model

Approach	Sensitivity	Specificity	Precision	Accuracy
AE	0.96	0.61	0.77	0.81
TBD with AE	0.98	0.95	0.98	0.98

Table A.7 shows the result of a time series anomaly detection with AEs, before and after applying TBD on top of the deep learning model. Before applying TBD, the AE wrongly classified 1490 data points out of 7268. Nevertheless, after applying TBD on top of the AE, the performance has indeed enhanced as we can see in Table A.7, with only 143 out of 7268 being wrongly classified.

## A.5 Conclusion

In this chapter, we have veered away from our primary discussion on BP Mining to introduce TBD—a groundbreaking method for identifying anomalies in IoT time-series data. The TBD approach introduces a novel way of handling IoT data streams by utilizing a Buffered Data Retrieval (BDR) mechanism and an Adelson-Velsky and Landis (AVL) tree for efficient data management. The BDR mechanism enhances data handling by temporarily storing incoming IoT data, allowing for collective retrieval, which optimizes computational resources. The AVL tree ensures quick data access times by maintaining a balanced structure, which is crucial for real-time anomaly detection.

Our evaluation on five real-world datasets has demonstrated TBD’s superior performance compared to existing state-of-the-art approaches. TBD exhibited higher sensitivity, specificity, precision, and accuracy, especially in distinguishing between point anomalies and collective anomalies. This differentiation is critical for applications that require precise anomaly detection to prevent system failures. Furthermore, we have shown that TBD can serve as a preprocessing engine for unsupervised deep learning models, significantly enhancing their performance by eliminating noisy data. This integration highlights TBD’s flexibility and its potential to improve existing anomaly detection frameworks.

Moving forward, future work for TBD will focus on several key areas to enhance its capabilities and address critical challenges in IoT systems. First, in the area of sensor failure detection, the challenge is that sensor malfunctions can cause significant disruptions, leading to inaccurate data analysis and system failures, as cited in [53]. The objective is to develop methods to accurately detect and isolate sensor failures using TBD’s anomaly detection capabilities. This will involve integrating TBD with advanced machine learning algorithms to improve the reliability of sensor data and ensure the robustness of IoT systems. Second, in terms of real-time adaptation and scalability, IoT systems generate vast amounts of data in real time, requiring scalable solutions that can adapt to changing conditions. The objective here is to enhance TBD to handle large-scale IoT deployments with minimal latency by implementing distributed processing frameworks and edge computing techniques to manage data locally and reduce the computational burden on central servers. Third, for integration with predictive maintenance, the challenge is that predictive maintenance relies on accurate anomaly detection to predict equipment failures and schedule maintenance proactively. The objective is to leverage TBD to improve the accuracy and reliability of predictive maintenance systems by combining it with predictive analytics to forecast potential failures and optimize maintenance schedules, thereby reducing downtime and maintenance costs. Fourth, in the area of anomaly explanation and visualization, understanding the context and cause of detected anomalies is crucial for effective decision-making. The objective is to develop tools to provide detailed explanations and visualizations of anomalies detected by TBD by implementing advanced visualization techniques and interpretable machine learning models to offer insights into the nature and impact of anomalies. Finally, for cross-domain applications, different domains have unique characteristics and requirements for anomaly detection. The objective is to adapt TBD for use in various domains such as healthcare, cybersecu-

urity, and environmental monitoring by customizing its algorithms and frameworks to address domain-specific challenges and enhance its applicability across different fields.



# Résumé Etendu

---

## B.1 Contexte et problématique de la recherche

L'évolution numérique au sein des entreprises transforme profondément leur fonctionnement. Au cœur de cette transformation se trouvent les **systèmes d'information (SI)**, qui visent à automatiser les tâches, à augmenter la productivité et à améliorer la prise de décision à tous les niveaux de l'organisation. Les SI sont des structures complexes comprenant du matériel, des logiciels, des données, des personnes et des processus. Parmi ces éléments, **les processus** sont essentiels pour obtenir les résultats escomptés.

Les *processus métiers* fournissent un cadre pour l'exécution des tâches, assurant ainsi clarté, contrôle et utilisation optimale des ressources. Il est crucial de suivre ces processus pour obtenir des certifications de qualité et une reconnaissance internationale. Cependant, les SI ne se contentent pas d'automatiser ces processus ; ils cherchent aussi à les gérer et à les améliorer en continu. Cela a donné naissance à la **Gestion des Processus Métiers (GPM)**, une approche globale centrée sur l'amélioration constante des processus organisationnels.

Dans le domaine de la GPM, certaines méthodes spécialisées comme l'*extraction des Processus Métiers (PM)* jouent un rôle clé. Cette méthode se compose principalement de deux aspects : la **vérification de conformité** et la **prédiction des processus**. La vérification de conformité consiste à comparer en temps réel les comportements des processus avec des modèles préétablis pour identifier les écarts et les erreurs. Cela est particulièrement utile dans le cadre des *courriels orientés processus*, où l'on peut vérifier si le contenu des courriels respecte les modèles de processus en termes de précision et d'exhaustivité.

La **prédiction des processus**, autre aspect crucial de la GPM, utilise des données historiques pour anticiper les comportements et performances futurs des processus métiers. Les données des journaux d'événements, qui enregistrent la séquence des activités dans un processus, sont analysées à l'aide de divers algorithmes et techniques d'extraction de processus.

Ces algorithmes permettent d'identifier des schémas et des tendances, qui servent de base à des modèles prédictifs utilisant des techniques comme les *arbres de décision*, les *modèles de Markov* et l'*apprentissage automatique*. Ces capacités offrent divers avantages : amélioration de la prise de décision, meilleure allocation des ressources et surveillance en temps réel, ce qui contribue à une meilleure satisfaction des clients.

Aujourd'hui, l'adoption de techniques prédictives s'étend à divers domaines, y compris les

**courriels orientés processus.** Dans ce contexte, les prédictions doivent pouvoir identifier non seulement les prochaines étapes possibles d'un processus, mais aussi suggérer des courriels spécifiques pour aider les acteurs du processus à accomplir ces étapes plus efficacement.

En intégrant la **vérification de conformité** et la **prédiction des processus** dans les systèmes de courrier électronique, les organisations peuvent créer un cadre de communication aligné sur les principes de l'extraction des Processus Métiers, favorisant ainsi un environnement flexible. Cependant, adapter ces techniques à la structure unique des courriels pose un défi en raison de leur composition distincte, comprenant des éléments spécifiques tels que les *actes de parole* et les *données métiers (DM)*.

## B.2 Objectifs et Contributions de la Thèse

Étant donné les problèmes décrits dans la section précédente, les objectifs principaux de cette thèse peuvent être résumés comme suit :

- **Objectif 1 :** Mettre en œuvre un *Contrôle de Conformité Multi-Perspectives* dans le contexte des emails.
- **Objectif 2 :** Développer un *Système de Recommandation de Réponses aux Emails Sensible aux Activités de Processus*.

Pour aborder efficacement ces objectifs, nous avons développé une variété d'algorithmes. L'importance de notre travail est encapsulée dans les contributions clés suivantes :

**Approche de Contrôle de Conformité Multi-Perspectives** Notre deuxième grande contribution implique une approche efficace pour assurer le contrôle de conformité à travers plusieurs perspectives au sein des contextes de courrier électronique. Cette approche garantit que les courriels électroniques individuels, ainsi que l'ensemble du fil, restent centrés sur le sujet.

Cette méthodologie comprend deux phases principales :

1. **Construction du Modèle :** Ici, nous construisons un modèle de processus en se basant sur les contraintes définies par un expert. Ce modèle combine deux types de contraintes :
  - *Contraintes Séquentielles :* Celles-ci spécifient l'ordre exact dans lequel les événements doivent se produire au sein des fils de courriels, garantissant une représentation fidèle de l'écoulement chronologique.
  - *Contraintes Contextuelles :* Celles-ci incorporent des détails contextuels liés à chaque événement, offrant une compréhension plus profonde des subtilités dans les interactions par courriel.



2. **Contrôle de Conformité** : Cette phase examine à quel point les échanges de courriels réels s'alignent avec notre modèle théorique. Nous utilisons un algorithme pour évaluer la conformité des événements dans les courriels individuels et dans les fils de courriels complets. Les événements sont classés soit comme *Événements Satisfaits*, qui adhèrent aux contraintes séquentielles et contextuelles, soit comme *Événements Violant*, qui s'écartent du comportement attendu.

**Système de Recommandation de Réponses aux Emails** Notre première grande contribution est la création d'un système de recommandation de réponses aux emails sensible aux activités de processus. Ce système est composé de quatre phases interconnectées :

1. **Élaboration d'un Modèle de Prédiction Orienté PM** : La première phase se concentre sur le développement de modèles prédictifs qui exploitent les journaux d'événements des échanges de courriels précédents.
2. **Identification des Activités et Instances** : Pendant cette phase, notre système scanne les emails reçus pour identifier les connaissances en PM existantes.
3. **Prédiction de la Connaissance en PM pour la Réponse** : À ce stade, le système prédit les connaissances en PM qui devraient être incluses dans la réponse email à venir.
4. **Recommandation de Modèles de Réponses** : Finalement, le système recommande un modèle de réponse par email qui s'aligne bien avec les activités et les connaissances en PM prédites.

Pour rendre ces méthodes prédictives et de conformité facilement accessibles, nous les avons intégrées en utilisant des *Interfaces de Programmation d'Applications RESTful*. Cette intégration offre un cadre unifié pour la gestion des courriels en GPM, simplifiant à la fois la mise en œuvre et l'utilisation.

En plus de nos contributions dans les domaines de la gestion des courriels et de la GPM, nous avons également exploré la détection d'anomalies dans les données en flux continu, en particulier dans le contexte de l'Internet des Objets (**IoT**). Poussés par une profonde curiosité, nous avons introduit une méthode non supervisée spécialement conçue pour détecter des anomalies dans les données séquentielles de l'IoT.



# Bibliography

- [1] Wil Van der Aalst, Arya Adriansyah, and Boudewijn Van Dongen. “Replaying history on process models for conformance checking and performance analysis”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2.2 (2012), pp. 182–192 (cit. on p. 18).
- [2] Wil M. P. van der Aalst. “Data science in Action”. In: *Retrieved from the Internet: URL: <https://www.coursera.org/lecture/process-mining/1-2-different-types-of-process-mining-G6am> [Retrieved on Dec. 27, 2018]* (2016) (cit. on p. 2).
- [3] Wil MP Van der Aalst. “Business process management: a comprehensive survey”. In: *International Scholarly Research Notices* 2013.1 (2013), p. 507984 (cit. on p. 1).
- [4] Ahmad Aburomman, Manuel Lama, and Alberto Bugarín. “A vector-based classification approach for remaining time prediction in business processes”. In: *IEEE Access* 7 (2019), pp. 128198–128212 (cit. on p. 23).
- [5] Arya Adriansyah, Boudewijn F. van Dongen, and Wil M. P. van der Aalst. “Conformance checking using cost-based fitness analysis”. In: *2011 IEEE 15th International Enterprise Distributed Object Computing Conference*. IEEE. 2011, pp. 55–64 (cit. on p. 18).
- [6] Charu C. Aggarwal. “On Abnormality Detection in Spuriously Populated Data Streams”. In: *Proceedings of the 2005 SIAM International Conference on Data Mining*, pp. 80–91 (cit. on p. 95).
- [7] Charu C Aggarwal and Charu C Aggarwal. “An introduction to outlier analysis”. In: Springer, 2017 (cit. on p. 94).
- [8] Charu C Aggarwal et al. “A framework for clustering evolving data streams”. In: (2003), pp. 81–92 (cit. on p. 98).
- [9] Charu C Aggarwal et al. “A framework for projected clustering of high dimensional data streams”. In: *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. 2004, pp. 852–863 (cit. on p. 98).
- [10] Charu C Aggarwal et al. *Neural networks and deep learning*. Vol. 10. 978. Springer, 2018 (cit. on p. 24).
- [11] Tsatsral Amarbayasgalan, Bilguun Jargalsaikhan, and Keun Ho Ryu. “Unsupervised novelty detection using deep autoencoders with density based clustering”. In: *Applied Sciences* 8.9 (2018), p. 1468 (cit. on p. 99).
- [12] Fabrizio Angiulli and Clara Pizzuti. “Fast Outlier Detection in High Dimensional Spaces”. In: *Principles of Data Mining and Knowledge Discovery*. Ed. by Tapio Elovmaa, Heikki Mannila, and Hannu Toivonen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 15–27 (cit. on p. 96).

- 
- [13] Mihael Ankerst et al. “OPTICS: Ordering Points to Identify the Clustering Structure”. In: *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*. SIGMOD '99. Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 1999, pp. 49–60 (cit. on p. 98).
- [14] Sasa Arsovski et al. “An approach to email categorization and response generation”. In: *Computer Science and Information Systems* 19.2 (2022), pp. 913–934 (cit. on pp. 26, 31).
- [15] Daniel Batrakhhanov et al. “Virtual sawing using generative adversarial networks”. In: *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*. IEEE. 2021, pp. 1–6 (cit. on pp. 24, 30).
- [16] Alessandro Berti and Wil M. P. 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*. Springer, 2021, pp. 1–26 (cit. on p. 8).
- [17] Alessandro Berti and Wil M. P. van der Aalst. “Reviving Token-based Replay: Increasing Speed While Improving Diagnostics”. In: *ATAED@ Petri Nets/ACSD 2371* (2019), pp. 87–103 (cit. on p. 8).
- [18] Markus M. Breunig et al. “LOF: Identifying Density-Based Local Outliers”. In: *SIGMOD Rec.* 29.2 (2000), pp. 93–104 (cit. on p. 96).
- [19] Markus M. Breunig et al. “LOF: Identifying Density-Based Local Outliers”. In: *SIGMOD Rec.* 29.2 (2000), pp. 93–104 (cit. on p. 97).
- [20] Andrea Burattin, Fabrizio M. Maggi, and Alessandro Sperduti. “Conformance checking based on multi-perspective declarative process models”. In: *Expert Systems with Applications* 65 (2016), pp. 194–211 (cit. on p. 20).
- [21] Cristina Cabanillas. “Process-and resource-aware information systems”. In: *2016 IEEE 20th international enterprise distributed object computing conference (EDOC)*. IEEE. 2016, pp. 1–10 (cit. on p. 20).
- [22] 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: (2022) (cit. on p. 23).
- [23] Ricardo Campos et al. “YAKE! Keyword extraction from single documents using multiple local features”. In: *Information Sciences* 509 (2020), pp. 257–289 (cit. on p. 74).
- [24] Josep Carmona et al. “Conformance checking”. In: *Switzerland: Springer* 56 (2018) (cit. on pp. 2, 8, 18).
- [25] Raghavendra Chalapathy and Sanjay Chawla. “Deep learning for anomaly detection: A survey”. In: *arXiv preprint arXiv:1901.03407* (2019) (cit. on p. 98).
- [26] Alexander J Chambers et al. “Automated business process discovery from unstructured natural-language documents”. In: *Business Process Management Workshops: BPM 2020 International Workshops, Seville, Spain, September 13-18, 2020, Revised Selected Papers 18*. Springer. 2020, pp. 232–243 (cit. on pp. 9, 31).

- 
- [27] Yiu-ming Cheung and Hong Jia. “A unified metric for categorical and numerical attributes in data clustering”. In: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. 2013, pp. 135–146 (cit. on p. 8).
- [28] Yiu-ming Cheung and Hong Jia. “Categorical-and-numerical-attribute data clustering based on a unified similarity metric without knowing cluster number”. In: *Pattern Recognition* 46.8 (2013), pp. 2228–2238 (cit. on p. 8).
- [29] Raffaele Conforti et al. “PRISM—a predictive risk monitoring approach for business processes”. In: *Business Process Management: 14th International Conference, BPM 2016, Rio de Janeiro, Brazil, September 18-22, 2016. Proceedings*. Springer. 2016, pp. 383–400 (cit. on p. 23).
- [30] Raffaele Conforti et al. “Supporting risk-informed decisions during business process execution”. In: *Advanced Information Systems Engineering: 25th International Conference, CAiSE 2013, Valencia, Spain, June 17-21, 2013. Proceedings*. Springer. 2013, pp. 116–132 (cit. on p. 23).
- [31] Alan Cook, Gözde Mısırlı, and Zhi Fan. “Anomaly Detection for IoT Time-Series Data: A Survey”. In: *IEEE Internet of Things Journal* 7.7 (2020), pp. 6481–6494 (cit. on p. 94).
- [32] Massimiliano De Leoni, Wil M. P. Van der Aalst, and Boudewijn F. Van Dongen. “Data-and resource-aware conformance checking of business processes”. In: *Business Information Systems: 15th International Conference, BIS 2012, Vilnius, Lithuania, May 21-23, 2012. Proceedings*. Springer. 2012, pp. 48–59 (cit. on p. 20).
- [33] Massimiliano De Leoni and Wil M. P. Van Der Aalst. “Aligning event logs and process models for multi-perspective conformance checking: An approach based on integer linear programming”. In: *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings*. Springer. 2013, pp. 113–129 (cit. on pp. 8, 19).
- [34] Mark Dredze, John Blitzer, and Fernando Pereira. “Sorry, I Forgot the Attachment: Email Attachment Prediction”. In: *CEAS*. 2006 (cit. on pp. 26, 31).
- [35] Marlon Dumas, Wil M Van der Aalst, and Arthur H Ter Hofstede. *Process-aware information systems: bridging people and software through process technology*. John Wiley & Sons, 2005 (cit. on p. 3).
- [36] Ted Dunning and Ellen Friedman. *Practical machine learning: a new look at anomaly detection*. " O'Reilly Media, Inc.", 2014 (cit. on p. 94).
- [37] Sebastian Dunzer et al. “Conformance checking: a state-of-the-art literature review”. In: *Proceedings of the 11th international conference on subject-oriented business process management*. 2019, pp. 1–10 (cit. on pp. 8, 18, 39).
- [38] Meshari Ebrahim. “Manufacturing Process Causal Knowledge Discovery using a Modified Random Forest-based Predictive Model”. In: (2020) (cit. on pp. 24, 30).
- [39] Maciej Eder, Jan Rybicki, and Mike Kestemont. “Stylometry with R: a package for computational text analysis”. In: *The R Journal* 8.1 (2016) (cit. on p. 77).

- [40] Gal Egozi and Rakesh M. Verma. “Phishing Email Detection Using Robust NLP Techniques”. In: *2018 IEEE Int. Conf. on Data Mining Workshops, ICDM Workshops, Singapore, November 17-20, 2018*. IEEE, 2018, pp. 7–12 (cit. on p. 22).
- [41] Marwa Elleuch. “Business process discovery from emails, a first step towards business process management in less structured information systems. (Découverte des processus métiers à partir des Emails, un premier pas vers la gestion des processus métiers dans des systèmes d’information moins structurés)”. PhD thesis. Polytechnic Institute of Paris, Palaiseau, France, 2021 (cit. on pp. 6, 50).
- [42] Marwa Elleuch et al. “Discovering Activities from Emails Based on Pattern Discovery Approach”. In: *Business Process Management Forum - BPM Forum 2020, Seville, Spain, September 13-18, 2020, Proceedings*. Ed. by Dirk Fahland et al. Vol. 392. Lecture Notes in Business Information Processing. Springer, 2020, pp. 88–104 (cit. on pp. 13, 64–66).
- [43] Marwa Elleuch et al. “Multi-perspective business process discovery from messaging systems: State-of-the art”. In: *Concurr. Comput. Pract. Exp.* 35.11 (2023) (cit. on pp. 40, 49).
- [44] Dina ElMenshawy and W. Helmy. “Detection techniques of data anomalies in IoT: A literature survey”. In: *International Journal of Civil Engineering and Technology* 9 (2018), pp. 794–807 (cit. on p. 94).
- [45] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231 (cit. on pp. 97, 98).
- [46] Martin Ester et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231 (cit. on p. 98).
- [47] Zhi-Ping Fan and Yang Liu. “A method for group decision-making based on multi-granularity uncertain linguistic information”. In: *Expert systems with Applications* 37.5 (2010), pp. 4000–4008 (cit. on p. 25).
- [48] Mohammadreza Fani Sani, Sebastiaan J. van Zelst, and Wil M. P. van der Aalst. “Conformance checking approximation using subset selection and edit distance”. In: *Advanced Information Systems Engineering: 32nd International Conference, CAiSE 2020, Grenoble, France, June 8-12, 2020, Proceedings*. Springer, 2020, pp. 234–251 (cit. on pp. 8, 18).
- [49] Paolo Felli et al. “CoCoMoT: conformance checking of multi-perspective processes via SMT”. In: *Business Process Management: 19th International Conference, BPM 2021, Rome, Italy, September 6-10, 2021, Proceedings*. Springer, 2021, pp. 217–234 (cit. on pp. 8, 19).
- [50] Yiwei Feng et al. “Reply using past replies—a deep learning-based e-mail client”. In: *Electronics* (2020) (cit. on pp. 9, 25).

- 
- [51] Roy Fielding et al. *RFC2616: Hypertext Transfer Protocol–HTTP/1.1*. 1999 (cit. on p. 60).
- [52] Luciano Floridi and Massimo Chiriatti. “GPT-3: Its nature, scope, limits, and consequences”. In: *Minds and Machines* 30 (2020), pp. 681–694 (cit. on pp. 29, 31).
- [53] Anuroop Gaddam et al. “Detecting Sensor Faults, Anomalies and Outliers in the Internet of Things: A Survey on the Challenges and Solutions”. In: *Electronics* 9.3 (2020) (cit. on p. 110).
- [54] Parag Gaikwad et al. “Anomaly detection for scientific workflow applications on networked clouds”. In: *2016 International Conference on High Performance Computing Simulation (HPCS)*. 2016, pp. 645–652 (cit. on p. 94).
- [55] John Cristian Borges Gamboa. “Deep learning for time-series analysis”. In: *arXiv preprint arXiv:1701.01887* (2017) (cit. on p. 98).
- [56] Anahita Farhang Ghahfarokhi et al. “OCEL: A standard for object-centric event logs”. In: *European Conference on Advances in Databases and Information Systems*. Springer. 2021, pp. 169–175 (cit. on p. 49).
- [57] Daniela Grigori et al. “Business process intelligence”. In: *Computers in industry* 53.3 (2004), pp. 321–343 (cit. on p. 2).
- [58] Riyaz Ahamed Ariyaluran Habeeb et al. “Real-time big data processing for anomaly detection: A Survey”. In: *International Journal of Information Management* 45 (2019), pp. 289–307 (cit. on p. 94).
- [59] Sandhya Harikumar and Surya PV. “K-Medoid Clustering for Heterogeneous DataSets”. In: *Procedia Computer Science* 70 (2015). Proceedings of the 4th International Conference on Eco-friendly Computing and Communication Systems, pp. 226–237 (cit. on p. 98).
- [60] Douglas M Hawkins. *Identification of outliers*. Vol. 11. Springer, 1980 (cit. on p. 94).
- [61] Markku Hinkka et al. “Classifying process instances using recurrent neural networks”. In: *Business Process Management Workshops: BPM 2018 International Workshops, Sydney, NSW, Australia, September 9-14, 2018, Revised Papers*. Springer. 2019, pp. 313–324 (cit. on p. 23).
- [62] Alaoui Ismaili. “Emails Analysis for Business Process Discovery”. In: *Algorithms & Theories for the Analysis of Event Data (ATAED’2019)* (2019), p. 54 (cit. on p. 9).
- [63] Sheng-Long Jiang, Xinyue Shen, and Zhong Zheng. “Gaussian process-based hybrid model for predicting oxygen consumption in the converter steelmaking process”. In: *Processes* 7.6 (2019), p. 352 (cit. on pp. 24, 30).
- [64] Andres Jimenez-Ramirez et al. “A method to improve the early stages of the robotic process automation lifecycle”. In: *Advanced Information Systems Engineering: 31st International Conference, CAiSE 2019, Rome, Italy, June 3-7, 2019, Proceedings*. Springer. 2019, pp. 446–461 (cit. on p. 22).
- [65] Xin Jin and Jiawei Han. “K-Means Clustering”. In: *Encyclopedia of Machine Learning*. Ed. by Claude Sammut and Geoffrey I. Webb. Boston, MA: Springer US, 2010, pp. 563–564 (cit. on p. 98).

- [66] Diana Jlailaty, Daniela Grigori, and Khalid Belhajjame. “Business process instances discovery from email logs”. In: *2017 IEEE International Conference on Services Computing (SCC)*. IEEE. 2017, pp. 19–26 (cit. on pp. 9, 32, 64).
- [67] Han-Shin Jo et al. “Path loss prediction based on machine learning techniques: Principal component analysis, artificial neural network, and Gaussian process”. In: *Sensors* 20.7 (2020), p. 1927 (cit. on pp. 24, 30).
- [68] Yong-Bin Kang, Anthony McCosker, and Jane Farmer. “Leveraging stylometry analysis to identify unique characteristics of peer support user groups in online mental health forums”. In: *Scientific Reports* 13.1 (2023), p. 22979 (cit. on p. 77).
- [69] Hans-Peter Kriegel et al. “LoOP: Local outlier probabilities”. In: 2009, pp. 1649–1652 (cit. on p. 97).
- [70] Guy Lapalme and Leila Kosseim. “Mercure: Towards an automatic e-mail follow-up system”. In: *IEEE Computational Intelligence Bulletin* 2.1 (2003), pp. 14–18 (cit. on pp. 27, 31).
- [71] Christophe Leys et al. “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median”. In: *Journal of Experimental Social Psychology* 49.4 (2013), pp. 764–766 (cit. on p. 103).
- [72] Qiuchi Li and Christina Lioma. “Template-based Contact Email Generation for Job Recommendation”. In: *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*. 2022, pp. 189–197 (cit. on pp. 9, 25).
- [73] Xiaolei Li and Jiawei Han. “Mining Approximate Top-k Subspace Anomalies in Multi-Dimensional Time-Series Data”. In: *Proceedings of the 33rd International Conference on Very Large Data Bases. VLDB ’07*. Vienna, Austria: VLDB Endowment, 2007, pp. 447–458 (cit. on p. 95).
- [74] Zhi Liang, M. A. C. Martell, and T. Nishimura. “A Personalized Approach for Detecting Unusual Sleep from Time Series Sleep-Tracking Data”. In: *2016 IEEE International Conference on Healthcare Informatics (ICHI)*. 2016, pp. 18–23 (cit. on p. 94).
- [75] Zupeng Liang et al. “An attribute-weighted isometric embedding method for categorical encoding on mixed data”. In: *Applied Intelligence* 53.22 (2023), pp. 26472–26496 (cit. on p. 8).
- [76] Felix Mannhardt et al. “Balanced multi-perspective checking of process conformance”. In: *Computing* 98 (2016), pp. 407–437 (cit. on pp. 8, 19).
- [77] Andrew McCallum, Xuerui Wang, and Andrés Corrada-Emmanuel. “Topic and role discovery in social networks with experiments on enron and academic email”. In: *Journal of Artificial Intelligence Research* 30 (2007), pp. 249–272 (cit. on pp. 26, 31).
- [78] Larry R Medsker, Lakhmi Jain, et al. “Recurrent neural networks”. In: *Design and Applications* 5.64-67 (2001), p. 2 (cit. on p. 28).
- [79] Nijat Mehdiyev, Joerg Evermann, and Peter Fettke. “A novel business process prediction model using a deep learning method”. In: *Business & information systems engineering* 62 (2020), pp. 143–157 (cit. on pp. 24, 30).



- [80] Mohsin Munir et al. “A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data”. In: 2019 (cit. on p. 94).
- [81] Tempestt Neal et al. “Surveying stylometry techniques and applications”. In: *ACM Computing Surveys (CSuR)* 50.6 (2017), pp. 1–36 (cit. on p. 77).
- [82] Liadan O’callaghan et al. “Streaming-data algorithms for high-quality clustering”. In: *Proceedings 18th international conference on data engineering*. IEEE, 2002, pp. 685–694 (cit. on p. 98).
- [83] Aditya Parameswaran et al. *Automatic email response suggestion for support departments within a university*. Tech. rep. PeerJ Preprints, 2018 (cit. on pp. 28, 31).
- [84] Vincenzo Pasquabisceglie et al. “Using convolutional neural networks for predictive process analytics”. In: *2019 international conference on process mining (ICPM)*. IEEE, 2019, pp. 129–136 (cit. on pp. 24, 30).
- [85] Maharsh Patel et al. “Customized Automated Email Response Bot Using Machine Learning and Robotic Process Automation”. In: *2nd International Conference on Advances in Science & Technology (ICAST)*. 2019 (cit. on pp. 27, 31).
- [86] Sanjay Patni. *Pro RESTful APIs*. Springer, 2017 (cit. on p. 59).
- [87] Dragoljub Pokrajac, Aleksandar Lazarevic, and Longin Jan Latecki. “Incremental Local Outlier Detection for Data Streams”. In: *2007 IEEE Symposium on Computational Intelligence and Data Mining* (2007), pp. 504–515 (cit. on p. 97).
- [88] Ashequl Qadir et al. “Activity modeling in email”. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2016, pp. 1452–1462 (cit. on pp. 9, 25, 26, 31, 64).
- [89] Peng Qi et al. “Stanza: A Python natural language processing toolkit for many human languages”. In: *arXiv preprint arXiv:2003.07082* (2020) (cit. on p. 79).
- [90] Md. Fazle Rabbi, Arifa I. Champa, and Minhaz F. Zibran. “Phishy? Detecting Phishing Emails Using ML and NLP”. In: *21st IEEE/ACIS Int. Conf. on Software Engineering Research, Management and Applications, SERA 2023, Orlando, FL, USA, May 23-25, 2023*. IEEE, 2023, pp. 77–83 (cit. on p. 22).
- [91] Anne Rozinat and Wil M. P. Van der Aalst. “Conformance checking of processes based on monitoring real behavior”. In: *Information Systems* 33.1 (2008), pp. 64–95 (cit. on p. 39).
- [92] Sai Aravind Sarswatula, Tanna Pugh, and Vittaldas Prabhu. “Modeling energy consumption using machine learning”. In: *Frontiers in Manufacturing Technology 2* (2022), p. 855208 (cit. on pp. 25, 30).
- [93] Robert Sedgewick. *Algorithms*. Addison-Wesley, 1983 (cit. on p. 101).
- [94] Martin Serrano and Amelie Gyrard. “A review of tools for IoT semantics and data streaming analytics”. In: vol. 6. 2016, pp. 139–163 (cit. on p. 94).
- [95] Mark S. Silver, M. Lynne Markus, and Cynthia Mathis Beath. “The information technology interaction model: A foundation for the MBA core course”. In: *MIS quarterly* (1995), pp. 361–390 (cit. on p. 1).

- [96] Florian Stertz, Juergen Mangler, and Stefanie Rinderle-Ma. “The role of time and data: online conformance checking in the manufacturing domain”. In: *arXiv preprint arXiv:2105.01454* (2021) (cit. on p. 19).
- [97] Elham Ramezani Taghiabadi et al. “Compliance checking of data-aware and resource-aware compliance requirements”. In: *On the Move to Meaningful Internet Systems: OTM 2014 Conferences: Confederated International Conferences: CoopIS, and ODBASE 2014, Amantea, Italy, October 27-31, 2014, Proceedings*. Springer. 2014, pp. 237–257 (cit. on p. 20).
- [98] Jian Tang et al. “Enhancing Effectiveness of Outlier Detections for Low Density Patterns”. In: *Proceedings of the 6th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. PAKDD '02. Berlin, Heidelberg: Springer-Verlag, 2002, pp. 535–548 (cit. on p. 97).
- [99] Niek Tax et al. “Predictive business process monitoring with LSTM neural networks”. In: *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017, Essen, Germany, June 12-16, 2017, Proceedings*. Springer. 2017, pp. 477–492 (cit. on pp. 14, 24, 30).
- [100] Maurras Ulbricht Togbe et al. “Etude comparative des méthodes de détection d’anomalies”. In: *Revue des Nouvelles Technologies de l’Information* (2020) (cit. on p. 94).
- [101] Wil Van Der Aalst. “Process mining: Overview and opportunities”. In: *ACM Transactions on Management Information Systems (TMIS)* 3.2 (2012), pp. 1–17 (cit. on p. 1).
- [102] Christophe Van Gysel et al. “Reply with: Proactive recommendation of email attachments”. In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 2017, pp. 327–336 (cit. on pp. 9, 25).
- [103] Venkat Venkatasubramanian et al. “A review of process fault detection and diagnosis: Part I: Quantitative model-based methods”. In: *Computers & chemical engineering* 27.3 (2003), pp. 293–311 (cit. on p. 25).
- [104] JP Verma and JP Verma. “Logistic regression: developing a model for risk analysis”. In: *Data Analysis in Management with SPSS Software* (2013), pp. 413–442 (cit. on p. 30).
- [105] Michael Gr. Voskoglou. “Case-Based Reasoning: A recent theory for problem-solving and learning in computers and people”. In: *The Open Knowledge Society. A Computer Science and Information Systems Manifesto: First World Summit on the Knowledge Society, WSKS 2008, Athens, Greece, September 24-26, 2008 Proceedings*. Springer. 2008, pp. 314–319 (cit. on pp. 28, 31).
- [106] Ying Wang et al. “Short-term solar power forecasting: A combined long short-term memory and gaussian process regression method”. In: *Sustainability* 13.7 (2021), p. 3665 (cit. on pp. 24, 30).
- [107] Andrew Gordon Wilson, David A Knowles, and Zoubin Ghahramani. “Gaussian process regression networks”. In: *arXiv preprint arXiv:1110.4411* (2011) (cit. on p. 24).

- 
- [108] Bastian Wurm et al. “Business process management and routine dynamics”. In: *Cambridge handbook of routine dynamics* (2021), pp. 513–524 (cit. on p. 1).
- [109] Kenji Yamanishi et al. “On-Line Unsupervised Outlier Detection Using Finite Mixtures with Discounting Learning Algorithms”. In: *Data Mining and Knowledge Discovery* 8 (2004), pp. 275–300 (cit. on pp. 95, 96).
- [110] Sen Yang et al. “An approach to automatic process deviation detection in a time-critical clinical process”. In: *Journal of Biomedical Informatics* 85 (2018), pp. 155–167 (cit. on pp. 8, 18).
- [111] Xiwang Yang et al. “A survey of collaborative filtering based social recommender systems”. In: *Computer Communications* 41 (2014), pp. 1–10 (cit. on pp. 9, 63).
- [112] Jinyeong Yu et al. “Predicting the Effect of Processing Parameters on Caliber-Rolled Mg Alloys through Machine Learning”. In: *Applied Sciences* 12.20 (2022), p. 10646 (cit. on pp. 24, 30).
- [113] Shuai Zhang et al. “Deep learning based recommender system: A survey and new perspectives”. In: *ACM Computing Surveys (CSUR)* 52.1 (2019), pp. 1–38 (cit. on pp. 9, 63).
- [114] Sicui Zhang et al. “Fuzzy multi-perspective conformance checking for business processes”. In: *Applied Soft Computing* 130 (2022), p. 109710 (cit. on p. 8).
- [115] Sicui Zhang et al. “Towards multi-perspective conformance checking with aggregation operations”. In: *Information Processing and Management of Uncertainty in Knowledge-Based Systems: 18th International Conference, IPMU 2020, Lisbon, Portugal, June 15-19, 2020, Proceedings, Part I*. Springer. 2020, pp. 215–229 (cit. on pp. 8, 19).
- [116] Zilong Zhao et al. “Robust anomaly detection on unreliable data”. In: *2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2019, pp. 630–637 (cit. on p. 109).



**Titre:** Optimiser la gestion des e-mails : Une approche basée sur les processus métier

**Mots clés:** Analyse des processus métier, Vérification de conformité, Prédiction des processus, Processus pilotés par courriel

**Résumé:** La gestion des processus métier (BPM) est cruciale pour toute organisation cherchant à améliorer constamment ses opérations. Cela implique plusieurs étapes : conception, modélisation, exécution, surveillance, optimisation et automatisation. Un élément central du BPM est l'analyse des processus, qui consiste à examiner les traces d'exécution pour identifier les inefficacités et les déviations par rapport aux processus prévus. Cette analyse se concentre particulièrement sur la prédiction des processus futurs et sur la vérification de leur conformité.

Dans cette thèse, nous nous penchons sur les défis spécifiques à l'analyse des processus métier lorsqu'ils sont pilotés par courriel. Il est essentiel de maîtriser ces pratiques pour rationaliser les opérations et maximiser la productivité. La vérification de conformité garantit que les processus réels respectent les modèles prédéfinis, assurant ainsi le respect des normes et standards. Par ailleurs, la prédiction des processus permet d'anticiper le comportement futur des opérations en se basant sur des données historiques, ce qui aide à optimiser l'utilisation des ressources et à gérer efficacement les charges de travail.

Appliquer ces techniques aux processus pilotés par courriel présente des défis uniques. En effet, ces processus manquent souvent des modèles formels trouvés dans les systèmes BPM traditionnels, ce qui nécessite des méthodologies adaptées. Les traces d'exécution dérivées des courriels ont une structure particulière, comprenant des attributs tels que les actes de parole des interlocuteurs et les données commerciales pertinentes. Cette complexité rend l'application des méthodes

standard de fouille des processus plus difficile. L'intégration de ces attributs dans les techniques existantes de BPM et les systèmes de courriel demande des algorithmes avancés et une personnalisation importante, d'autant plus que le contexte des communications par courriel est souvent dynamique.

Pour relever ces défis, cette thèse propose plusieurs objectifs. D'abord, mettre en place une vérification de conformité multi-aspects et concevoir un système de recommandation de réponse par courriel qui tient compte des activités du processus. Ensuite, il s'agit de concevoir un modèle de processus basé sur des contraintes séquentielles et contextuelles spécifiées par un analyste/expert en données. Il est également crucial de développer des algorithmes pour identifier les événements conformes et non conformes, d'utiliser les traces d'exécution pour prédire les connaissances des processus métier et de proposer des modèles de réponse par courrier électronique. Les principes directeurs de cette approche sont la sensibilité au contexte, l'interdisciplinarité, la cohérence, l'automatisation et l'intégration. L'une des contributions majeures de cette étude est le développement d'un logiciel complet pour l'analyse des processus pilotés par courriel. Ce programme combine la prédiction des processus et la vérification de conformité pour améliorer la communication par courriel. Il propose des modèles de réponse adaptés et évalue la conformité des courriels avant leur envoi. Pour valider ce logiciel, des données de courriels réels ont été utilisées, fournissant ainsi une base pratique pour des comparaisons et des recherches futures.

**Title:** Enhancing Email Management Efficiency: A Business Process Mining Approach

**Keywords:** Business Process Mining, Process Prediction, Conformance Checking, Email-Driven Processes

**Abstract:** Business Process Management (BPM) involves continuous improvement through stages such as design, modeling, execution, monitoring, optimization, and automation. A key aspect of BPM is Business Process (BP) mining, which analyzes event logs to identify process inefficiencies and deviations, focusing on process prediction and conformance checking. This thesis explores the challenges of BP mining within email-driven processes, which are essential for streamlining operations and maximizing productivity.

Conformance checking ensures that actual process execution aligns with predicted models, maintaining adherence to predefined standards. Process prediction forecasts future behavior based on historical data, aiding in resource optimization and workload management. Applying these techniques to email-driven processes presents unique challenges, as these processes lack the formal models found in traditional BPM systems and thus require tailored methodologies.

The unique structure of email-derived event logs, featuring attributes such as interlocutor speech acts and relevant business data, complicates the application of standard BP mining methods. Integrating these attributes into existing business process techniques and email

systems demands advanced algorithms and substantial customization, further complicated by the dynamic context of email communications.

To address these challenges, this thesis aims to implement multi-perspective conformance checking and develop a process-activity-aware email response recommendation system. This involves creating a process model based on sequential and contextual constraints specified by a data analyst/expert, developing algorithms to identify fulfilling and violating events, leveraging event logs to predict BP knowledge, and recommending email response templates. The guiding principles include context sensitivity, interdisciplinarity, consistency, automation, and integration.

The contributions of this research include a comprehensive framework for analyzing email-driven processes, combining process prediction and conformance checking to enhance email communication by suggesting appropriate response templates and evaluating emails for conformance before sending. Validation is achieved through real email datasets, providing a practical basis for comparison and future research.