Universiteit Leiden

ICT in Business and the Public Sector

Prescriptive Process Optimization Using Large Language Models: A Preliminary Investigation

Name: Eric Manintveld
Student-no: s2277352

Date: 23/08/2024

1st supervisor: Dr. M. Hilbert
2nd supervisor: Prof.dr.ir. J.M.W. Visser
Company supervisor: A. Maldonado

MASTER'S THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract
Prescriptive Process Optimization (PPO), a subfield of process mining, focuses on providing actionable and timely recommendations to improve processes. While current research explores various prescriptive methods to determine when to trigger an action, the specific actions to be triggered often remain unclear. Previous work proposed recommendation engines with predetermined actions. This research investigates the viability of large language models (LLMs), specifically the GPT-3.5 Turbo model, for generating tailored recommendations. We propose a system that integrates existing process mining techniques with an LLM. To map the process model and identify optimal intervention points, we use the Split Miner algorithm, a gradient-boosted tree model-based alarm system, and the PM4Py package. Process details obtained using these techniques are used to form prompts for the LLM, which then generates timely and actionable recommendations. Unlike previous approaches, using an LLM eliminates the need for predefined actions, enabling the system to handle unexpected situations and thereby increasing its flexibility. Survey results indicate the PPO-system, in its current state, would not provide significant value to companies. While the system can generate relevant recommendations, their applicability varies. Some recommendations are suitable for day-to-day operations, while others are only appropriate for tactical process management. The performance of the system was perceived to be mostly neutral. With these results, the newly discovered research directions and limitations of this research in mind, there seems to be potential for LLMs in a PPO context. Therefore, this research aims to lay the groundwork for further exploration of LLMs in the process mining field.

Acknowledgements
Writing this thesis was a great learning experience. While the performance of the PPO-system did not fully meet my initial ambitions, I am proud of the progress and outcomes I achieved. This would not have been possible without the continuous support and guidance of my supervisors.

I would like to express my gratitude to Dr. Marc Hilbert for his constructive feedback and structured approach throughout this process. Your insights and guidance have been very helpful for writing this thesis.

Thank you, Ana Maldonado, for your amazing feedback and for pushing me to think critically, as well as for brainstorming with me to come up with solutions.

I also want to thank Prof. dr. ir. Joost Visser for his great feedback as my second supervisor.

Finally, I want to thank Sooyeon and Chat-GPT for proofreading my thesis. (Chat-GPT[1] was used to check for grammar errors and improve textual flow.)

[1] https://chat.openai.com/

Contents

1 Introduction

Creating tailored recommendations for process improvements enables organizations to optimize their business operations. Ideally, these recommendations should be in easy-to-understand natural language, ensuring actions can be quickly executed by the right stakeholders. Prescribing recommendations in this manner is the primary focus of the prescriptive process optimization (PPO) subfield within process mining [Chapela-Campa and Dumas, 2023].

The PPO field concerns two main problems: finding the right timing for intervention, and generating good actionable recommendations. Most existing studies look into the first problem [Teinemaa et al., 2018], finding the right timing. In recent years a plethora of systems were developed that can accurately predict the right timing for an intervention. However, far fewer studies looked into generating recommendations [Park et al., 2023]. This research gap was noted by [Kubrak et al., 2022]. The few solutions that did get proposed, require a large amount of domain knowledge and cost a lot of effort to set up for a specific process. In other words, they are expensive to set up and not flexible.

Meanwhile we have witnessed the rise of large language models (LLMs), such as OpenAI's ChatGPT. These models are trained on large amounts of text data. Their main strengths are their flexibility and output in natural language. LLMs have demonstrated their applicability across a wide range of problems, and their natural language input and output have made them accessible to a broad audience.

Therefore, this research explores the viability of LLMs as recommendation engines in a PPO context. By integrating LLMs with existing process mining techniques, we aim to develop a system that can automatically generate timely and actionable recommendations to help negate undesirable outcomes in business processes. This system is called the PPO-system. Furthermore, this research attempts to answer the question: "How can large language models be leveraged in Prescriptive Process Optimization to generate recommendations that help negate undesirable outcomes in business processes?"

We evaluate the performance of the system through a survey. The results of the survey are both qualitatively and quantitatively analyzed. During the qualitative analysis we highlight several generated recommendations. The quantitative analysis looks at the average responses over all recommendations.

Looking ahead, the rapid advancement of LLMs, such as the development of GPT-4 and GPT-4o, offers exciting opportunities for future research. Evaluating the capabilities of these newer models and comparing their performance with GPT-3.5 will provide valuable insights into their potential for generating tailored recommendations for process optimization.

In summary, this research aims to explore the potential of LLMs in generating actionable recommendations within a PPO context. By integrating LLMs with process mining techniques, we aim to develop a system that enhances process execution by providing timely and relevant recommendations. The findings of this research highlight both the opportunities and challenges in this emerging field, paving the way for future advancements and practical applications.

This thesis begins by explaining the current state of research in Section 2. This provides an understanding of existing technologies and highlights the research gap. Section 3 presents the research question based on this research gap. Section 4 covers the technical aspects regarding the PPO-system. The system's output is then analyzed in Section 5 through both qualitative and quantitative methods. Following this, Section 6 interprets the results and outlines future research directions. Finally, Section 7 summarizes our findings and answers the research question.

2 Related Work

2.1 Process Mining

Due to the growing availability of data regarding business processes, process mining emerged within the data science field. The sub-field focuses on using event data or event logs to discover, monitor, and improve processes [van der Aalst, 2011]. Previously, process models were typically created manually, often through methods such as conducting interviews. However, this approach often missed parts of the process and was skewed by interviewees giving socially acceptable answers. Therefore, process mining uses trails left by people, machines, and software called *event* logs. And unlike classical data mining techniques like classification, clustering, and regression, process mining focuses on end-to-end processes, rather than analyzing a specific step in the overall process. The process models produced by process mining techniques are used in analysis like simulation, verification or digital twins [van der Aalst, 2012]. More recent research has expanded into combining process mining with the predictive capabilities of machine learning techniques. This allows analysts to predict whether the outcome of a process will be positive or negative [Chapela-Campa and Dumas, 2023]. Helping them to make interventions at runtime [Kubrak et al., 2022, Teinemaa et al., 2019].

2.1.1 Process Mining Pyramid

The augmented business process management (BPM) pyramid, as seen in figure 1, is an overview of the current situation in the research field of process mining. Within the pyramid, each layer builds upon the previous one. Naturally these layers are also roughly in chronological order when it comes to development. The two lower layers, Descriptive Process Analytics and Predictive Process Analytics, are fields of research that have matured and are established. The top two layers, Prescriptive Process Optimization and Augmented Process Execution, have been accelerated with the widespread adoption of AI and machine learning. The different layers all contain techniques that can be grouped into two use cases: tactical use cases, where the goal is to inform managers to help with decision-making, and operational use cases, that provide information and trigger actions in day-to-day operations [Chapela-Campa and Dumas, 2023]. This research focuses on the latter.

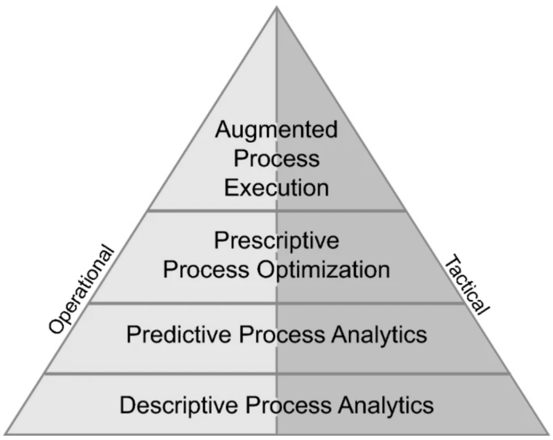


Figure 1: Augmented business process management pyramid [Chapela-Campa and Dumas, 2023].

Descriptive Process Analytics
The base layer of the pyramid is the Descriptive Process Analytics layer. In this layers the fundamentals of process mining are found. The main goal of the techniques in this layer, is to show the current state of business processes. These insights are used by business stakeholders (managers, architects and other experts) to make decisions in an attempt to improve performance. Early research suggested three basic types of process mining: automated process discovery, conformance checking, and performance mining [van der Aalst, 2012, Dumas, 2011, Nguyen et al., 2018, Taymouri et al., 2021]. The insights gathered using the techniques are often presented to business stakeholders using a dashboard.

Predictive Process Analytics
Techniques in the Descriptive Process Analytics layer enable business stakeholders to detect and investigate issues or implement improvements in process. However, they are not capable of detecting issues before they occur, or predict the future state of a process. This is the focus of the Predictive Process Analytics layer. The main goal of this layer is building predictive models capable of predicting the future state of a process [Chapela-Campa and Dumas, 2023]. With the rise of machine learning over the past decade a multitude of different approaches were presented [van Dongen et al., 2008, Di Francescomarino et al., 2016, Mehdiyev et al., 2020, Tax et al., 2017, Van der Aalst et al., 2011]. These approaches can be split up into two categories: case-level and process level. At case-level the goal is to predict details about a specific case, such as the outcome, time until completion or predicting the next action in the case. At process-level the goal is to predict the performance of a set of cases [Chapela-Campa and Dumas, 2023].

Prescriptive Process Optimization
Process mining techniques in the bottom two layers mostly focus on mapping processes and what-if simulations. In the past years, these techniques have been adopted by industries. Therefore, the focus of research has shifted towards automatic process improvement [Park and van der Aalst, 2020]. To automatically add value to the predictions made in the bottom two layers, automatic actions should follow. Therefore, determining the timing and type of these actions is the main focus of the third layer, the Prescriptive Process Optimization (PPO) layer. The PPO-system developed in this research is an example of a system in this layer. The techniques in the PPO layer turn predictions into timely actions. This area of research within the process mining field is undergoing rapid development. Due to the cutting-edge nature of this research, it is hard to find proposed solutions that are verified in real-world settings. A generic architecture of systems found in this layer can be seen in Figure 2.

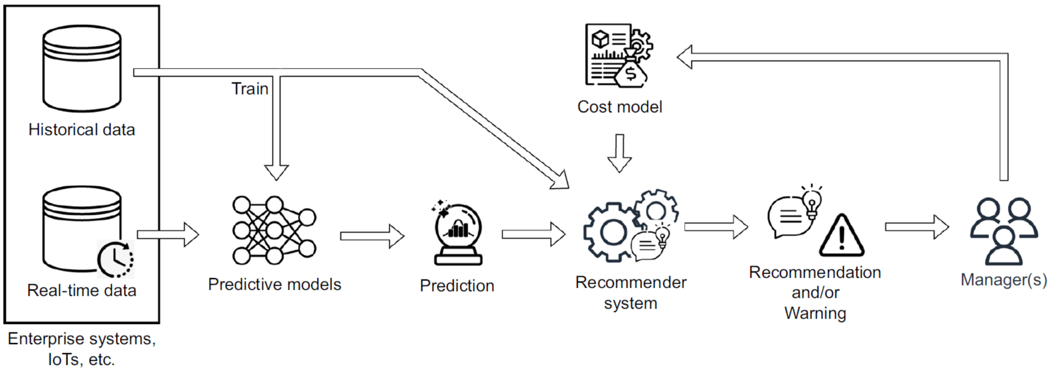


Figure 2: Generic architecture of prescriptive process optimization systems [Chapela-Campa and Dumas, 2023].

Solutions in this layer often present some kind of recommender system that advises on the best course of action. These recommender system consist of two distinct parts. One part is responsible for determining the correct timing of an intervention, like the alarm system by [Teinemaa et al., 2018]. While the other part determines the appropriate action to be taken, like the general framework for Action-Oriented Process Mining [Park and van der Aalst, 2020]. This framework generates recommendations based on a constraint function. However, in these systems, predefined actions are required, meaning that unexpected issues can still arise and remain unsolved.

Augmented Process Execution

Augmented Process Execution, the top layer in the pyramid, takes automating process optimization a step further. In the PPO layer, a human operator receives suggestions in order to improve a process. This operator can still choose to accept or ignore the suggestion. In the Augmented Process Execution layer however, the machine is in control. The human operator acts as a supervisor and can give suggestions or make interventions. To make sure the system operates within reason, it is only allowed to operate within a predefined set of restrictions [Chapela-Campa and Dumas, 2023].

2.1.2 Research Gap

A literature review by [Kubrak et al., 2022] reveals a research gap. Techniques have been proposed that focus on identifying the right timing for an intervention, such as the alarm system by [Teinemaa et al., 2018]. However, there is limited research on generating actionable recommendations.

2.2 Process Visualization

Visualization of business processes has long been a interesting topic for many businesses [Rinderle et al., 2006]. Simple processes can be easily understood, but when processes become larger and more complex this becomes difficult. Therefore, different techniques and standards have been proposed to visualize processes using process models. This section will discuss the various aspects of visualizing a process.

2.2.1 Event Logs

Events logs are at the base of process mining and process visualization. They contain the information that is displayed in process models. Event logs are datasets containing data from processes and are extracted from information systems. They describe a collection of events. Each event has an activity, a timestamp and an associated process. Often they also contain additional information about the event. This information can include things like the resource executing the activity or information like the size of an order [van der Aalst, 2012]. There are two main types of event logs: traditional and object-centric event logs [Berti and Qafari, 2023]. This research will use only traditional event logs, as the available and suitable datasets we found are in this format.

Traditional Event Logs

In traditional event logs each event is tied to a specific case. A case is a unique execution or instance of the process. All the different instances of events within a specific case are called a *trace*. Every event is characterised by its activity and a timestamp [Berti and Qafari, 2023]. Often there are also objects or items associated with processes, a specific customer for example [Adams et al., 2022]. Table 1 shows an example of a traditional event log.

Case ID	Activity	Timestamp
Case 1	Activity A	2024-02-25 10:20:00
Case 1	Activity B	2024-02-25 14:50:00
Case 1	Activity C	2024-02-26 09:15:00
Case 1	Activity D	2024-02-27 15:40:00
Case 2	Activity A	2024-02-24 08:30:00
Case 2	Activity C	2024-02-25 16:45:00

Table 1: A simple example of a traditional event log.

XES-standard

Traditional event logs are usually stored in the Extensible Event Stream (XES) standard format [Gunther and Verbeek, 2014]. The purpose of this format is to make sharing event logs between information systems easier [Acampora et al., 2017]. The XES-standard is a structured way of storing traditional event logs using XML. The biggest element in this structure is the log element. This element gives context regarding the specific process that is described by the event log. A log consists of traces. A trace is a collection of events. A trace describes one instance or execution of the process. For example, it could represent the process associated with a request for one building permit. The trace follows this request from the moment it comes into the system, until it has been accepted or denied. The smallest element within XES, is the event. An event represents an action or activity that has taken place. Therefore, it often includes an activity name and a timestamp. It is also possible to capture more data relating to the specific activity. For example, the value of a traffic fine or the cost of a purchase. Figure 3 shows an example of an event log. The event log is from a dataset of a traffic fine management system [de Leoni and Mannhardt, 2015].

```xml
<trace>
        <string key="concept:name" value="A29028"/>
        <event>
                <float key="amount" value="38.0"/>
                <string key="org:resource" value="537"/>
                <string key="dismissal" value="NIL"/>
                <string key="concept:name" value="Create Fine"/>
                <string key="vehicleClass" value="A"/>
                <float key="totalPaymentAmount" value="0.0"/>
                <string key="lifecycle:transition" value="complete"/>
                <date key="time:timestamp" value="2009-04-30T00:00:00+02:00"/>
                <int key="article" value="157"/>
                <int key="points" value="0"/>
        </event>
        <event>
                <string key="concept:name" value="Send Fine"/>
                <string key="lifecycle:transition" value="complete"/>
                <float key="expense" value="13.5"/>
                <date key="time:timestamp" value="2009-07-28T00:00:00+02:00"/>
        </event>
        <event>
                <string key="notificationType" value="P"/>
                <string key="concept:name" value="Insert Fine Notification"/>
                <string key="lifecycle:transition" value="complete"/>
                <string key="lastSent" value="P"/>
                <date key="time:timestamp" value="2009-08-03T00:00:00+02:00"/>
        </event>
        <event>
                <float key="amount" value="77.5"/>
                <string key="concept:name" value="Add penalty"/>
                <string key="lifecycle:transition" value="complete"/>
                <date key="time:timestamp" value="2009-10-02T00:00:00+02:00"/>
        </event>
        <event>
                <string key="concept:name" value="Send for Credit Collection"/>
                <string key="lifecycle:transition" value="complete"/>
                <date key="time:timestamp" value="2012-03-26T00:00:00+02:00"/>
        </event>
</trace>
```

Figure 3: XES trace from traffic fine management system.

2.2.2 Process Models

Process models are an important tool in process mining. They visualize the order in which activities occur and their relationship to each other. These models are based on the idea that activities in a process have a structured order, meaning that the completion of a process proceeds the start of another [Berti and Qafari, 2023]. There are multiple approaches to visualize a process model, such as Directly-Follows Graphs (DFGs) [Van Der Aalst, 2019], Petri nets [Van der Aalst, 1998], and Business Process Model and Notation (BPMN) models [Dijkman et al., 2011]. These models make processes understandable to humans and serve as abstractions that machines can use to analyze processes.

Directly-follows Graph

One of the simplest ways to capture processes is using DFGs. They visualize the sequence of activities in a process by showing how activities follow each other. In DFGs, nodes represent activities, and directed edges between nodes indicate that one activity directly follows another [Van Der Aalst, 2019]. Numbers along the edges note the frequency of the path, showing how many times one activity follows another. Due to their simplicity, DFGs struggle to represent more complex process behaviors, such as parallelism or loops.

Business Process Model and Notation

Business Process Model and Notation (BPMN) is a framework for business process modeling, using graphical symbols. This notation system, rooted in flowcharting methodologies, helps to represent intricate business processes. BPMN provides a notation that is intuitive for business stakeholders while retaining the capacity to express complex process structures [Von Rosing et al., 2015]. Figure 4 shows part of a BPMN model that was mined from the traffic fine management system dataset.

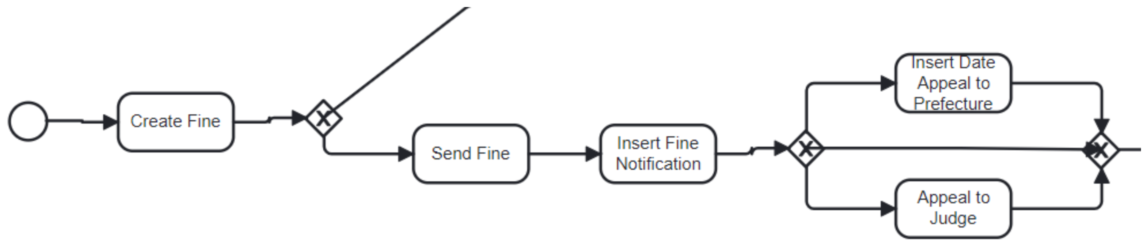


Figure 4: Part of a BPMN model mined from traffic fine management system dataset using the split miner algorithm.

A BPMN model begins with a start node and is composed of various elements, such as activities and gateways. These gateways enable BPMN models to represent parallelism and loops, allowing for modeling of complex process flows. The most common type of activity in a BPMN model is the 'task' activity, represented by a solid, rounded rectangle containing a description of the task. For the purposes of this research, we focused exclusively on task activities. Tasks can include 'markers,' which indicate special characteristics such as loops. These markers are displayed as small symbols within the task rectangle. In this research, the only marker encountered during the generation of BPMN models was the loop marker. This marker indicated that the task is repeated until a certain criteria or condition is met.

The ends or outcomes of a process flow in a BPMN model are represented by circles with a bold outline. These "end events" may include symbols inside to denote different types of endings, such as normal completion or termination. The presence of multiple end events allows the BPMN model to depict different possible outcomes of a process.

Petri Net

Petri nets offer a method to visualize various types of processes, not limited to business processes. However, a subset known as Workflow nets is specifically designed to represent business processes. Petri nets generally consist out of four elements: places, transitions, arcs, and tokens.

Places represent possible states of the process [Petri and Reisig, 2008]. A token must be present in a place before the actions that follow it can be executed. In Petri nets, tokens represent the current state of a process. Every Petri net starts with a specific number of tokens in the initial place(s). Tokens move from one place to another through transitions. A transition consumes tokens from its input place(s) and produces tokens in its output place(s). Some Petri nets contain two special places, the source and the sink. These Petri nets belong to the subclass of Workflow nets [Van der Aalst, 1997]. These nets are used to model business processes. In a Workflow net, the source has no preceding activities, and the sink has no succeeding activities.

Transitions are the activities that consume incoming tokens and produce tokens in the subsequent place. In the context of process mining, these transitions often represent business processes. Arcs are the connections between places and transitions [Peterson, 1977]. They determine the flow of the process.

2.2.3 Split Miner

The Split Miner algorithm [Augusto et al., 2021] is a technique found in the Descriptive Process Analytics layer of the process mining pyramid. It focuses on automatically extracting a process model from event logs. Although the bottom layer of the pyramid is a matured research field, this approach was proposed relatively recently. It uses a novel approach that leverages the strengths of DFGs. The output of the algorithm is a process model in BPMN format. [Augusto et al., 2019] found that Split Miner outperformed other process model extraction methods based on F-score, while execution times were significantly faster. The algorithm works by analyzing event logs to identify frequent and significant splits in the process flow. A split occurs when a process diverges into multiple paths, representing different choices or options [Augusto et al., 2019]. Five main steps within the algorithm can be identified [Augusto et al., 2021]:

1. DFG and Loops Discovery
2. Concurrency Discovery
3. Filtering
4. Splits Discovery
5. Joins Discovery

These five steps are discussed in more detail in Section 4.2.4. The only input, apart from the event logs, is the concurrency threshold parameter. This parameter specifies the minimum percentage of overlap between the life cycles of two activities required to classify them as concurrent. The default value set by the developers of the algorithm is 0.05. The specific version of the algorithm used in this research is Split Miner 2.0 [Augusto et al., 2021].

2.2.4 PM4Py

With the emergence of the process mining sub-field, a wide range of tools have been developed to cater to process mining needs. However, many of these tools are expensive and/or restricted in customizability. Data science in Python is mostly openly available through open-source licenses, with a lot of advanced data science packages available, like pandas, numpy, scipy and sci-kit learn. Therefore, it is also a suitable environment for the machine learning based techniques found in the third and fourth layers of the BPM pyramid. The Python package PM4Py [Berti et al., 2019] provides access to process mining capabilities and offers extensive customizability. PM4Py bridges the gap between process mining and Python's data science environment. The package supports a wide range of process mining techniques, like discovery, conformance checking, and process enhancement.

Large Language Models and PM4Py

Recently, support for LLMs was added to the PM4Py package [Berti and Qafari, 2023]. This functionality mainly focuses on transforming traditional abstractions of processes, as described in Section 2.2.2, into a textual format. Figure 5 shows an example of a textual abstraction of a DFG. Individual cases can also be abstracted into a textual format.

If I have a Petri net:

places: [ent_node_05fa37dc-55ca-4bf5-b13d-4edefd8de021, ent_node_0d91f028-d522-43a2-9a51-dfba1c121bc7,

...

exi_node_d9e66cec-db0b-4708-99ca-50ca3fc6b152, exi_node_daad3d93-d7ea-4b8e-9162-b202f8c3777c, sink, source]
transitions: [(node_0a202bb8-6a8c-46cc-a0bc-0906e21a45f8, 'A_Denied'), (node_0d91f028-d522-43a2-9a51-dfba1c121bc7, 'A_Accepted'),

...
(sfl_node_f0679e6e-a5b0-4b97-92f6-367fe3aefe28, None), (sfl_node_f0a75256-86ad-4e10-a5dc-b9051d1035fe, None)]
arcs: [(node_0a202bb8-6a8c-46cc-a0bc-0906e21a45f8, 'A_Denied')->ent_node_e3a7a475-2b13-4d3f-b215-76fa62dcde21,

...

exi_node_daad3d93-d7ea-4b8e-9162-b202f8c3777c->(node_870cfe03-5032-4875-99a9-b91f5bcbaaf7, 'O_Accepted'), source->(node_245a6b4d-cfda-4c5f-966f-4d478d166b41, 'A_Create Application')]

initial marking: ['source:1']
final marking: ['sink:1']

Figure 5: A shortened example of a textual abstraction of a Petri net model created using PM4Py.

2.3 Alarm-based Prescriptive Process Monitoring

The alarm system proposed by [Teinemaa et al., 2018] operates within the PPO layer of the BPM pyramid. The system attempts to prevent undesirable case outcomes by alarming a human operator. A notable feature of this system is its integration of a cost model, which serves as a mechanism to prevent incessant triggering of alarms. This ensures that alerts are only raised when warranted by deviations or issues.

The system first extracts a process model from event logs. Then it checks running cases and uses machine learning to predict their outcomes. It either uses a Random Forest (RF) or a Gradient Boosted Tree (GBT) model. [Teinemaa et al., 2018] found that the GBT model performed slightly better in their evaluation. Therefore, we will focus on the GBT model.

For each event in a trace, the model predicts the probability the trace will have an undesirable outcome. If this probability is higher than a certain threshold the alarm is raised. The exact value of this threshold is determined during the training phase of the model using the cost model. The alarm system is built in Python. It takes a comma separated value (CSV) file as input, with each row representing an event. Figure 6 shows an high-level abstraction of the alarm system.

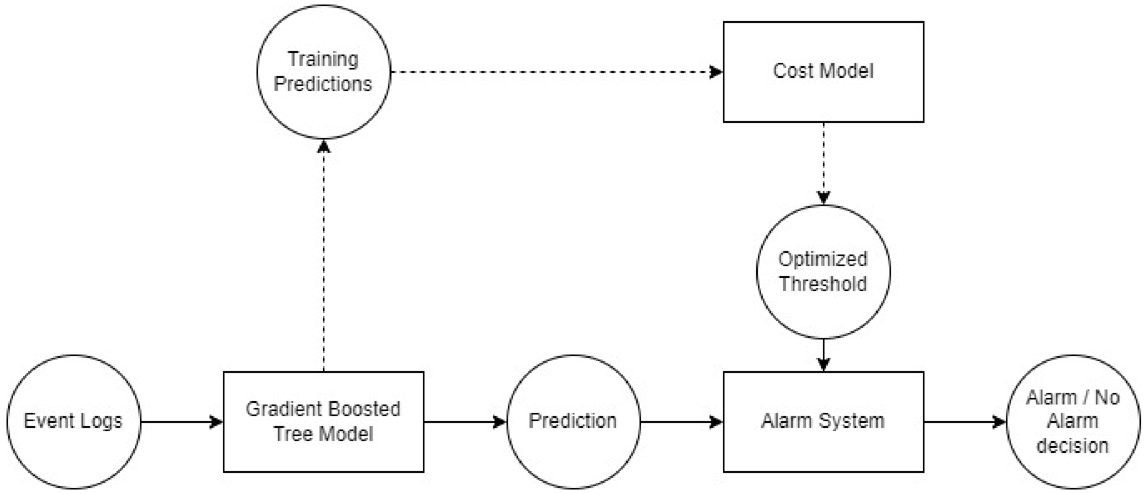


Figure 6: High-level abstraction of the alarm system. Training steps are depicted with a dotted line.

2.3.1 LightGBM

The alarm system uses LightGBM [GuolinKe et al., 2017] as its GBT implementation. Compared to other gradient boosting implementations such as XGBoost [Chen and Guestrin, 2016], LightGBM is meant to be lightweight and fast. It achieves this by using two techniques. Exclusive feature bundling (EFB), and gradient-based one side sampling (GOSS). During this research, the alarm system ran locally. Therefore, we chose to use LightGBM for efficient computing times.

Boosting
Boosting algorithms combine multiple intermediate machine learning models into an ensemble model. These intermediate models are also known as “weak learners” [Schapire, 2003]. The process begins by training an initial model on the entire dataset. Every time a new model is trained, the boosting algorithm analyzes its output, specifically focusing on the errors made by the previous models. Samples that were incorrectly classified by previous models are given more weight during the subsequent training run, thereby forcing the new model to focus on the harder-to-predict samples [Natekin and Knoll, 2013]. This iterative process continues until the desired number of weak learners is trained. The weak learners are then combined into a single ensemble model. The weight of each weak learner in the ensemble model is often based on the model’s performance during training.

Gradient Boosting
Gradient boosting is a specific type of boosting algorithm. It starts with a simple initial model based on some default value or rule, such as the average or median of the label variable. Every time new weak learner is trained, the output of the ensemble model is reviewed through a loss function. This loss function is arbitrary and is often based on

metrics such as root mean square error (RMSE) or the residuals of the samples [Natekin and Knoll, 2013]. Subsequently, a weak learner is trained to minimize the loss function. This model is then assigned a "learning rate", which determines the impact of the model on the final outcome. If the prediction of the initial model is p_1 and the prediction of the first weak learner is p_2, the prediction of the ensemble model consisting of p_1 and p_2 is $p_1 +$ (learning rate $\times$ p_2). The learning rate is a hyperparameter that is often set to 0.1 by default, but it can be adjusted based on the specific problem and model tuning. When applying gradient boosting to classification problems, the output of the model represents the probability that a sample belongs to a certain class.

Exclusive Feature Bundling
EFB [GuolinKe et al., 2017] takes advantage of the nature of exclusive features to improve performance. Exclusive features are features where for any given data sample, if one feature has a non-zero value, the other features must be zero. These exclusive features can naturally appear in a dataset, but are also often caused by encoding techniques, such as one-hot encoding. The alarm system uses the Pandas method `pd.get_dummies()` to encode the input data. Similarly to one-hot encoding, this method can produce exclusive features. By identifying and bundling these exclusive features, the total number of features can be decreased. This reduced number of features speeds up the training process and reduces memory usage.

Gradient-based One Side Sampling
GOSS [GuolinKe et al., 2017] uses the gradients of data samples to determine which data is most useful for training. The gradients are sorted from high to low, with higher gradients basically indicating larger errors in the prediction. The top twenty percent of data points with the highest gradients are selected, as they represent the samples with the greatest potential for improving the model. From the remaining eighty percent, a random ten percent is selected and combined with the high-gradient samples. The low-gradient samples not selected are excluded from the training data. This approach reduces the number of samples for the next training iteration. This speeds up training times, while maintaining the core principle of gradient boosting by prioritizing samples with high gradient values.

2.3.2 Training the Model and Threshold Optimization

The output of our LightGBM model is the probability that a process will result in a undesirable outcome. So, the final step in training the alarm system is determining the appropriate threshold for triggering the alarm. This threshold selection process is facilitated by the cost model, where the optimization of the threshold value is guided by ratios derived from the costs associated with undesirable outcomes and the costs of intervention. The resulting optimal threshold is then serialized as a *hyperopt.space eval* object and stored as a *.pickle* file. [Teinemaa et al., 2018] proposed three different ways of optimizing the threshold:

1. Varying the ratio between the cost of the undesired outcome and the cost of the intervention, while keeping other parameters of the cost model unchanged.

2. Varying both the ratio from the first method and the mitigation effectiveness of the intervention.

3. Varying two ratios: the cost of the undesired outcome and the cost of the intervention and the cost of the intervention and the cost of compensation.

2.4 Large Language Models (LLMs)

LLMs are a type of machine learning within the field of natural language processing [Berti and Qafari, 2023]. They are capable of producing human-like text by predicting what word is most likely to come next based on previously observed words. This allows them to accomplish a wide range of tasks with output in natural language. LLMs are neural networks [Krogh, 2008] that use the transformer architecture [Vaswani et al., 2017]. LLMs are usually trained on large amounts of data. This gives them access to information on a wide range of topics. At the same time this also means that the capability of an LLM to interpret input is limited by to training data. LLMs can be interacted with using natural language, making them accessible to a wide audience.

2.4.1 Chat-GPT

Chat-GPT is a transformer based large language model. It is being developed by the research laboratory OpenAI. The output of the model is in natural language [Singh et al., 2023]. Since its inception, there have been multiple different versions, ranging from GPT-1 until the current version of GPT-4. Currently, Chat-GPT 3.5 and 4o are openly available through the ChatGPT chatbot[2]. However, using the Chat-GPT API requires a subscription. For this research we got access to the `gpt-3.5-turbo-0125` API through an Azure[3] subscription provided by Avanade. The API connection used in this research was established through Azure OpenAI Studio.

2.4.2 Prompting

The way a question is asked often has a big influence over the answer to that question. This is also the case with large language models. Formulating the right prompts is crucial in order to get the desired output. Therefore, different prompting strategies have been proposed, such as *ask me anything* [Arora et al., 2022], *few-shot* [Logan IV et al., 2021], *zero-shot* [Reynolds and McDonell, 2021] and *least-to-most* [Zhou et al., 2022].

Least-to-most prompting breaks the problem down into smaller chunks [Zhou et al., 2022]. These smaller chunks are then solved one by one by the LLM. The *ask me anything* approach argues that combining multiple "less-than-perfect" prompts can lead to better results than attempting to engineer a singular "perfect" prompt [Arora et al., 2022]. In *few-shot* prompting, instead of training a model on a massive amount of labeled data for specific tasks, the model is provided with a few examples (or shots) of the task it needs to perform, along with a prompt or instruction [Logan IV et al., 2021]. *Zero-shot* prompting is a variant of *few-shot* prompting that takes the concept a step further. In *zero-shot* prompting, instead of providing the model with a few examples (or shots) of the task it needs to perform, the model is expected to perform the task without any explicit examples or training data [Reynolds and McDonell, 2021].

Another method for guiding the model is by providing instructions. We can specify the tasks for the model and outline the desired format of its responses. With ChatGPT-3.5, this is accomplished by using a *system message* [OpenAI, 2024]. The model does not produce an output in response to this message. Instead, it uses this contextual information when generating a response to a subsequent prompt.

[2]https://chatgpt.com/

[3]https://azure.microsoft.com/en-us

2.4.3 Temperature

In the context of LLMs, the temperature (t) parameter controls the randomness of the model's output [Renze and Guven, 2024]. When forming a response, an LLM is constantly determining what word has the highest likeliness to follow. $t = 0$, means that the model is deterministic and will choose the word with the highest likeliness to come next. $t > 0$, means that there is a randomness introduced. This makes the model more "creative" in its responses. The default temperature value for ChatGPT is 0.7 [Qualtir, 2024]. Recent research suggests that dynamically adjusting the temperature parameter during inference can be beneficial. This has led to the development of adaptive temperature sampling techniques [Zhu et al., 2024, Xie et al., 2024].

2.4.4 Evaluation

LLMs have only recently taken the spotlight. Therefore, there is no generally accepted framework or technique for evaluating the quality of their output. Based on a recent survey, Chang et al. identified three key points when it comes to evaluating LLMs [Chang et al., 2023]:

1. **What to evaluate?**
   First, it is important to understand what kind of task has to be evaluated, and what aspects of the outcome or what metrics to use. For example, the models ability to process natural language, ethics, trustworthiness and bias.

2. **Where to evaluate?**
   This concerns looking for the right data and benchmarks. In recent times, new benchmarks and datasets have become available. However, these are only focused on evaluating the LLM itself, and not its implementation in another system.

3. **How to evaluate?**
   When evaluating LLMs, there are two options: human evaluation and automatic evaluation. Automatic evaluation requires a ground truth, while human evaluation requires domain experts.

3 Research Question

Based on the research gap highlighted in Section 2.1.2, we present the following research question with corresponding hypotheses.

"How can large language models be leveraged in Prescriptive Process Optimization to generate recommendations that help negate undesirable outcomes in business processes?"

- **H1**: Large language models can interpret business processes well enough to generate relevant recommendations.
- **H2**: Large language models can add value as recommendation engines in a Prescriptive Process Optimization context.
- **H3**: The temperature setting in LLMs affects the performance of the LLM as a recommendation engine.

4 Method

To address the presented research gap, this study aims to leverage the knowledge embedded in existing LLMs. By integrating LLMs with recently proposed process mining techniques, we aim to develop a prescriptive process optimization system (PPO-system). The PPO-system is designed to effectively and timely identify problematic traces in event logs, which indicate inefficiencies or potential issues within a process execution. Once these problematic traces are identified, the system generates a fitting solution in natural language, making the recommendations easily understandable and actionable for human operators. The source code for this system is available on GitHub[4].

4.1 Theoretical Framework

The main deliverable of this research is a prototype, the PPO-system. Therefore we use the design science methodology. This methodology consists of two main parts: developing an artifact that provides improvement for stakeholder, and scientifically investigating the performance of this artifact [Wieringa, 2014]. In this case, the artifact is the PPO-system. The context of these main parts can be extended by a social context and a knowledge context, encapsulating every aspect of a design science project. The knowledge context contains everything that is already known in existing literature. This includes the techniques and knowledge discussed in Section 2, like the alarm system, PM4Py, and LLMs. The stakeholders of the project make up the social context. These stakeholders include organisations with business processes. They want to improve the efficiency of these processes, while keeping costs low. A graphical representation of this framework can be found in Figure 7. Alongside this design science framework, the augmented BPM pyramid, as presented in Section 2.1.1, will also be used as a framework in this research. The pyramid helps to put the different techniques found in the process mining research field into an ordered context.

[4]https://github.com/EricManintveld/PPO-system

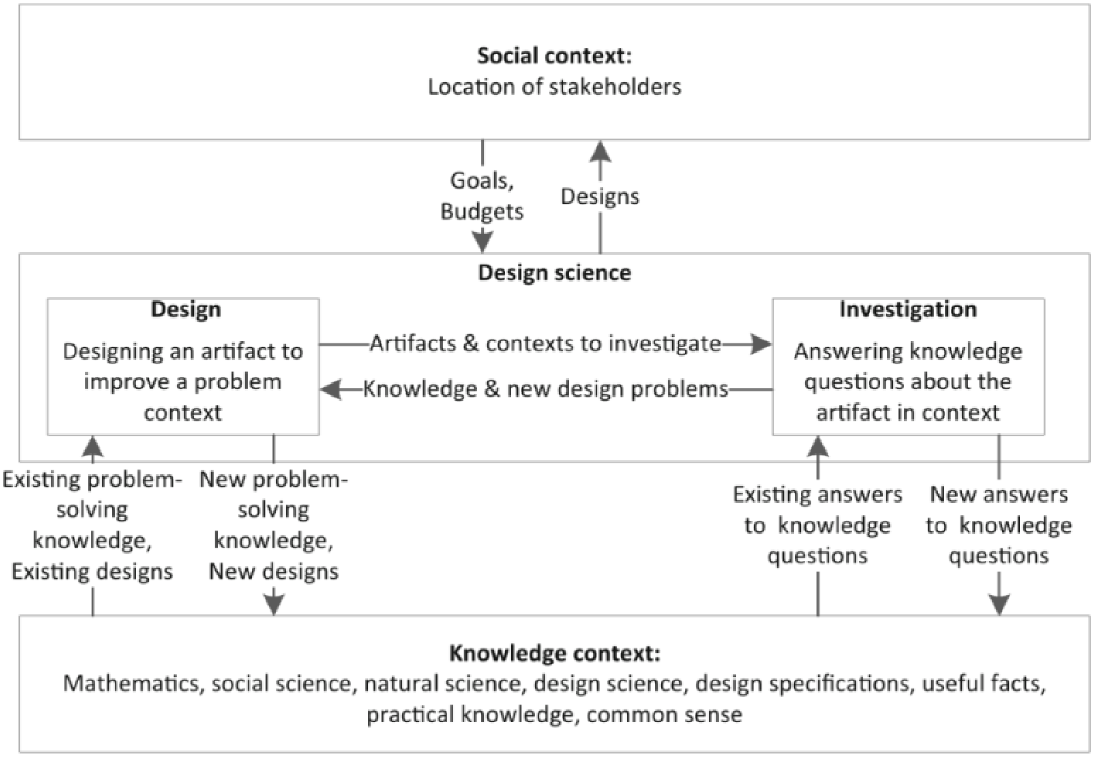


Figure 7: Design Science Framework proposed by Wieringa.

4.2 PPO-system Architecture

The architecture of the PPO-system is based on the general prescriptive process optimization architecture presented in Section 2.1.1. The alarm system covers the *predictive models*, *prediction* and *cost model* parts of the architecture. The novelty of this research concerns the *recommender system* and *recommendation and/or warning* elements.

Figure 8 shows a high-level overview of the PPO-system. This overview illustrates the architecture of the system using various shapes to denote different elements. The input of the system is indicated by a slanted square. Rectangles represent processes, while rectangles with a wavy bottom edge indicate intermediate outputs. Final outputs or end states are depicted by rounded rectangles. There is a decision point in the flowchart, shown as a circle with a cross. Additionally, the overview includes a merge point where the outputs of two processes combine into a single intermediate output, represented by an upside-down triangle. The overview shows the first initial execution of the system. The process model abstraction does not get regenerated with every execution, since we assume the process does not change. In the following section, we go into more detail for each part of the proposed architecture.

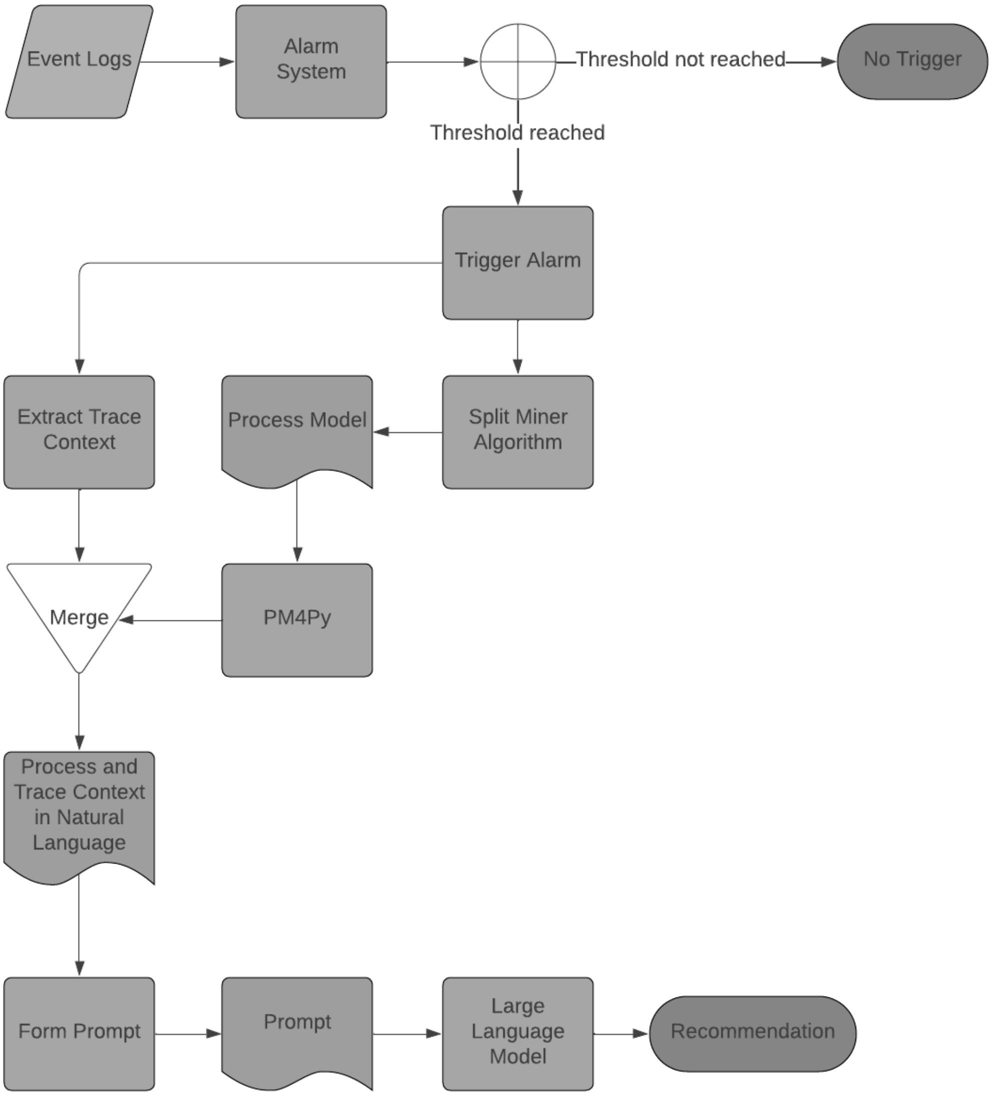


Figure 8: Proposed design of prescriptive process optimization (PPO) system.

4.2.1 Event Logs

The context of this research is process mining. Therefore we use event logs extracted from business processes as the data input for the PPO-system. These event logs are openly available for research and are often used in previous process mining related works. For this research we have selected two different event log datasets. The Road Traffic Fine Management Process dataset for use during development, and the BPI Challenge 2017 dataset for evaluation. Descriptive statistics of these datasets are shown in Table 2.

Name	**Purpose**	**Logged Events**	**Distinct Events**
Road Traffic Fine Management Process	Development	561470	11
BPI Challenge 2017	Evaluation	1202267	26

Table 2: Descriptive statistics of the datasets.

BPI Challenge 2017 Dataset
The *BPI Challenge 2017* [van Dongen, 2017] dataset describes a loan application process within a Dutch financial institution. The dataset includes all applications submitted through an online system in 2016, along with their subsequent events tracked until February 2017. It also provides information on executed activities, the resources that executed these activities, timestamps, reasons for the applications, loan amounts, applicant credit scores, and the amounts of money offered. We used this dataset during the evaluation of the PPO-system. It captures a complex, yet understandable, process. Both datasets are stored in the XES format. To ensure the system's generalizability, we develop and test it on one dataset and validate it on another. The process described by the dataset is understandable to a wide audience, increasing the pool of potential experts for evaluating the system.

Preprocessing
Before using the datasets, we perform a preprocessing step. The XES dataset is converted into CSV format using PM4Py and Pandas. This is necessary since the alarm system only accepts event logs in CSV format. Next, the dataset is divided into training and validation sets. Eighty percent of the traces are reserved for training, while the remaining twenty percent is used for evaluating the system. Labels for the training set are then generated, as they are needed to train the alarm system model. During the labeling process, we categorize the data into three distinct categories. Positive labels, which should trigger the alarm and therefore represent undesirable outcomes. Negative labels, which represent the desirable outcomes, and finally, there are unknown labels. Unknown labels are applied if it is unclear if the process execution will finish in a desirable or undesirable outcome. In other words: the process execution has not yet concluded. Below an overview of the different labels in the BPI Challenge 2017 dataset:

- **Positive labels**: The undesirable outcomes.
  - *O_Refused*: The customer has refused the loan offer.
  - *O_Cancelled*: The financial institution cancelled the loan application process.
- **Negative labels**: The desirable outcome.
  - *O_Accepted*: The financial institution send a final loan offer and the customer accepted this offer.
- **Unknown labels**: Neither a positive nor a negative label exists in the trace. The trace was not yet completed.

Figure 9 shows the distribution of the labels. The dataset contains 12178 traces with a negative label, 19239 traces with a positive label and 92 traces with an unknown label.

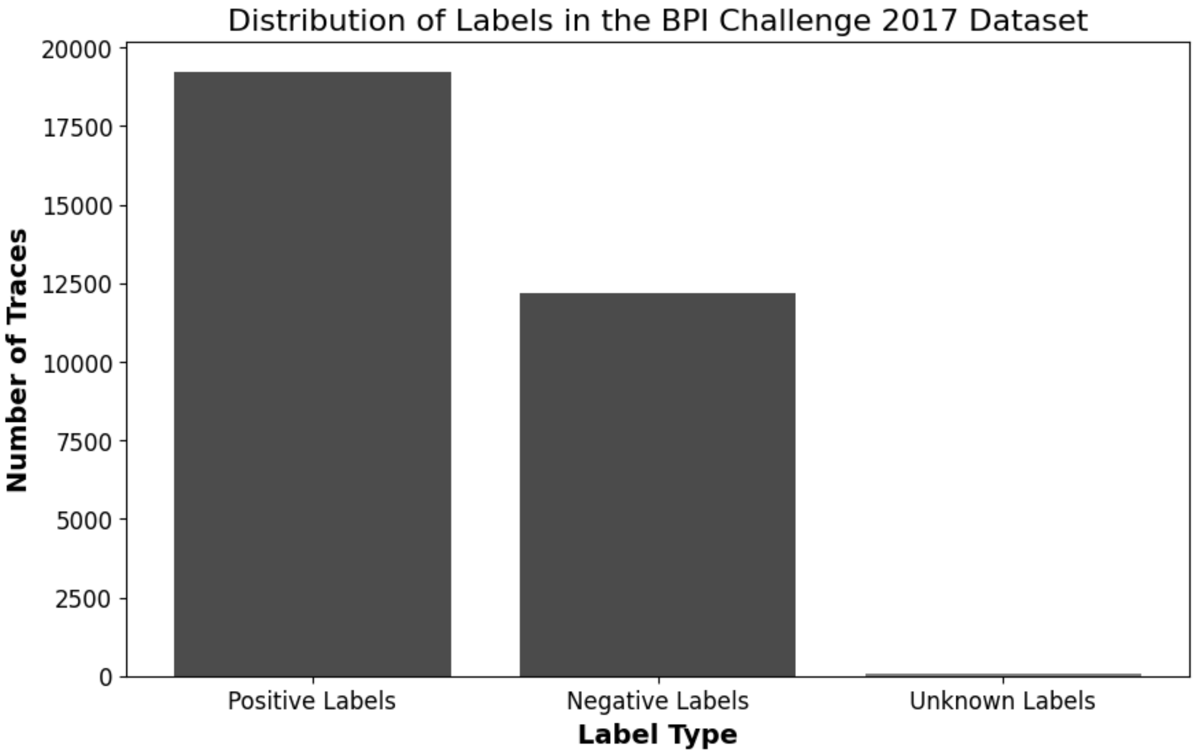


Figure 9: Distribution of labels in the BPI Challenge 2017 dataset.

After applying the labels, all traces labeled with *unknown* are omitted. They provide no information about the ending state of the process, making them useless for training the alarm system model. We omitted a total of 92 traces from the BPI Challenge 2017 dataset.

Next, the trace is truncated to remove the perfect predictor that is present at the end of each completed trace. To simulate the randomness of real-time data a random number of events is omitted. Let l be the length or number of events in a trace. The number of events removed from the end of the trace (r) is determined by the following conditions:

- If $l > 10$, then $5 < r < 9$
- If $6 < 1 < 10$, then $5 < r < l - 1$
- If $l < 6$, the trace is omitted from the dataset.

Traces with fewer than six events are excluded from the dataset. Analyzing the initial part of the process, as depicted in Figure 10, shows the first five activities follow a relatively linear path. Due to this linearity and predetermined sequence of actions, there are limited opportunities for intervention or deviations. Additionally, traces declined early in the process offer no further opportunities for action. We assume these offers are unserious or unreasonable. So the only decision is an immediate decline. Therefore, these traces do not provide useful information. The Jupyter Notebook containing the preprocessing steps can be found in Appendix A, along with a subset of the preprocessed dataset in Appendix B.

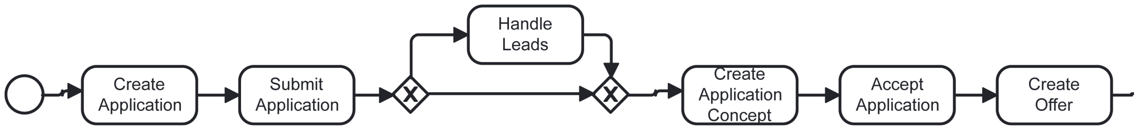


Figure 10: The start of the process captured in the BPI Challenge 2017 dataset.

4.2.2 Alarm System

Previous research found that the alarm system performs better when using the GBT model over the RF model [Teinemaa et al., 2018]. Therefore, the PPO-system uses the GBT model. To avoid overfitting during hyperparameter optimization and model training the alarm system automatically splits up the dataset. The system uses 64 percent of the data for training, 16 percent for optimizing the alarming threshold and twenty percent for evaluation [Teinemaa et al., 2018].

Threshold Optimization
For optimizing the threshold we use a script provided with the alarm system. This particular script optimizes the threshold by exploring different ratios between the cost of an undesired outcome and the cost of an intervention, keeping other cost model parameters unchanged. This is the least complex of the the three provided scripts. Since we are mainly interested in the recommendations given by the LLM, we determined that the added complexity of the remaining two scripts provides no additional value for this research.

This script provides optimized thresholds for different ratios of the *average cost of intervention* versus the *average cost of an undesired outcome*. Determining what ratio the system should use requires domain knowledge about these two costs. Due to the unavailability of this knowledge for the current research and dataset, we have adopted a ratio of five to one between the average cost of an undesired outcome and the average cost of intervention. This relatively high ratio ensures that the alarm will be triggered by a wide range of problems. This allows for a comprehensive analysis of the LLM's potential as a recommendation engine across various scenarios.

Alarm System Implementation
To implement the alarm system into the PPO-system, we had to make various adjustments. In its original state the alarm system is not ready for implementation in a practical system. It was intended for academical purposes. However, in a real-life situation data points can be registered at any time. In the PPO-system, the alarm system is able to generate a prediction for a new event when it is registered. Figure 11 shows a UML activity diagram highlighting the implementation of the alarm system in the PPO-system.

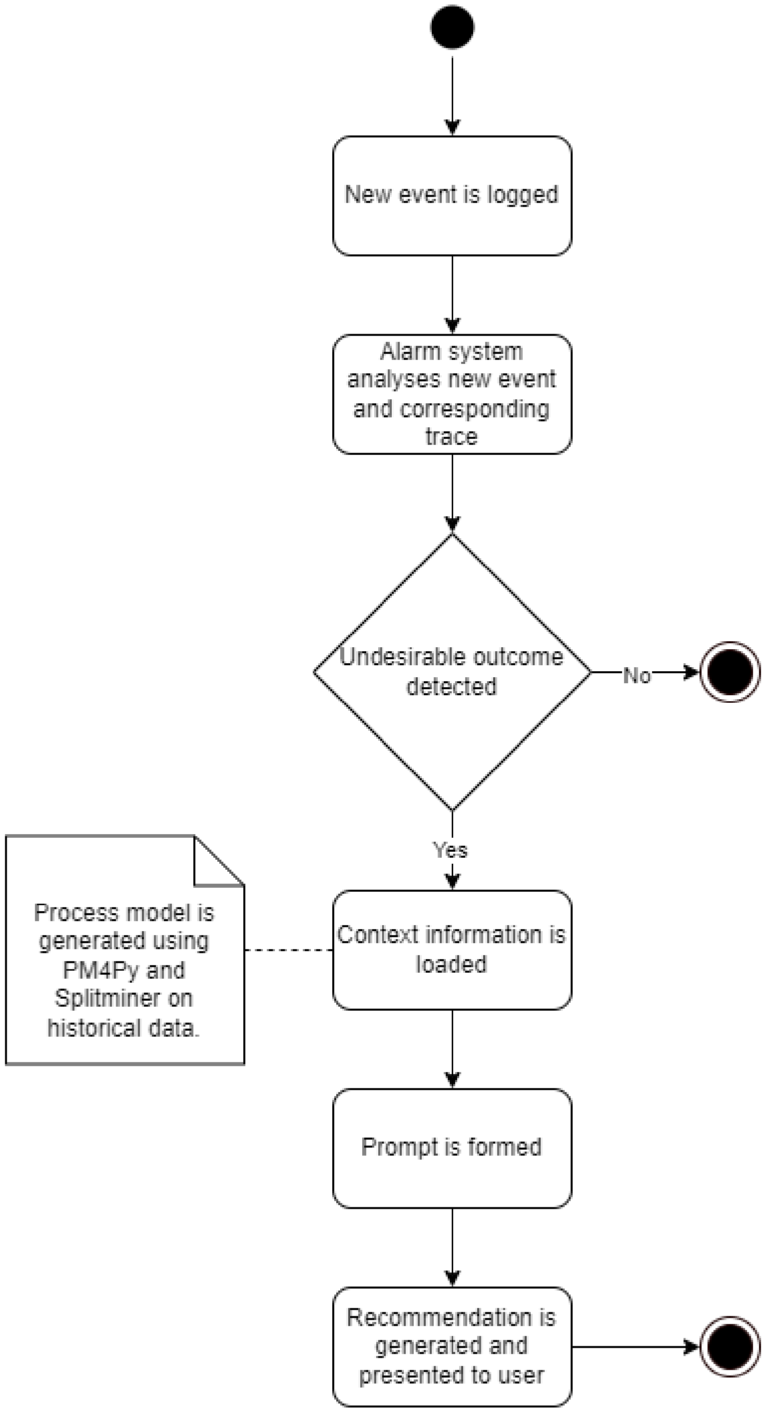


Figure 11: UML activity diagram highlighting the implementation of the alarm system in the PPO-system.

4.2.3 Extract Trace Context

To generate a specific recommendation that fits the situation, the PPO-system system requires information about the trace that triggered the alarm system. After triggering the alarm, all events leading up to this moment in the responsible trace are formatted in natural language. This is done by consecutively listing the different event connected by arrows.

4.2.4 Split Miner

The PPO-system uses the Split Miner algorithm to extract a process model from the event logs. This process model helps the LLM understand the process. The input to the Split Miner algorithm consists of event logs in XES format. This process of discovering the process model is executed only once, as we assume that the process does not change over time. This assumption is based on the relatively short one-year timespan during which the events in the dataset were recorded. However, in practice processes change, so the process model has to be regenerated periodically. We leave investigating the optimal frequency for regenerating the model to future research. The only hyperparameter that can be tweaked in the split miner algorithm is the concurrency threshold parameter. For the purposes of this research we set this to the default value of 0.05. As discussed in Section 2.2.3, the Split Miner algorithm consists of five main steps. We will now take a closer look at each of these five steps.

DFG and Loops Discovery

The first step of the Split Miner algorithm is transforming the provided event log into a DFG graph [Augusto et al., 2021]. Within the context of the Split Miner algorithm the definitions for an event log, a directly-follows relation and a DFG graph are given by Definitions 4.1, 4.2 and 4.3. These definitions, along with Definition 4.4 have been cited from previous research on the Split Miner algorithm [Augusto et al., 2019, Augusto et al., 2021]. By applying Definitions 4.1, 4.2 and 4.3 we obtain a DFG graph representing the process captured in the provided event logs.

Definition 4.1 (Event Log)**.** Given a set of events E, an event log L is a multiset of traces, where a trace $t \in L$ is a sequence of events $t = \langle e_1, e_2, ..., e_k \rangle$, with $e_i \in E, 1 \leq i \leq k$. Each event $e \in E$ is a tuple $e = (l, p, t)$, where $l \in A$ is the process activity the event refers to, retrieved with the notation e^l; $p \in \{start, end\}$ is the state of the life-cycle of activity l, retrieved with the notation e^p; and t is the timestamp of the event, retrieved with the notation e^t. [Augusto et al., 2021]

This definition states that an event log is a collection of traces. Where each trace is a sequence of events. Where each event has an associated activity, state, and timestamp.

Definition 4.2 (Directly-Follows Relation)**.** Given an event log L and two process activities $a_x, a_y \in A$, the relation $a_x \rightarrow_r a_y$ holds iff $\exists \langle e_1, e_2, \ldots, e_k \rangle \in L \mid e_i^l = a_x \wedge e_j^l = a_y \wedge e_i^p = \text{end} \wedge e_j^p = \text{start} \wedge 1 \leq i < j \leq k \wedge \nexists n \in (i, j) \mid e_n^p = \text{end}$. [Augusto et al., 2021]

In other words, there is a directly-follows relationship between the two activities a_x and a_y, if the activity a_x ended before activity a_y started. And there are no activities that ended between the end of activity a_x and the start of activity a_y.

Definition 4.3 (Directly-Follows Graph)**.** Given an event log L, its DFG is a directed graph $G = (N, E)$, where N is the non-empty set of nodes, where each node represents a unique activity $a \in L$ and there exists a bijective function $\lambda : N \mapsto A$ such that $\lambda(n)$ retrieves the activity n refers to; and E is the set of edges capturing the directly-follows relations of the activities observed in $L, E = \{(n.m) \in N \times N | \lambda(n) \rightarrow \lambda(m)\}$. [Augusto et al., 2021]

So, a DFG is a directed graph consisting of a collection of nodes and edges. Where each node represents a unique activity. Meaning the same activity cannot appear twice in the same DFG. The edges indicate the relationship between these activities like they are described in the event log.

Concurrency Discovery
The next step in the algorithm is determining which events happen concurrently. This notion of concurrency is influenced by the arbitrary value $\varepsilon \in [0, 1]$. This value ε is the concurrency threshold. In the context of the Split Miner algorithm, we consider two activities $a_x, a_y \in A$ concurrent if the following statement holds:

$$2 \times \frac{|a_x \asymp a_y|}{|a_x| + |a_y|} \leq \varepsilon$$

where $|a_x \asymp a_y|$ denotes the number of times a_x and a_y were observed to overlap. And $|a_x|$ and $|a_y|$ represent the number of times the activities were observed in the event log L. The edges between the points that were found to be executed concurrently are removed from the set of edges E [Augusto et al., 2021].

Filtering
The filtering step ensures that the generated DFG satisfies the three properties listed below. It is important that the DFG satisfies these properties to ensure a smooth transformation into a BPMN model later. The three properties are:

1. Each node in the DFG must be part of a path that connects the start node to the end node.

2. For each node, the path it's on is the path having the maximum capacity. In the context of Split Miner, capacity of a path is determined by the frequency of the least frequent edge of the path.

3. The number of edges in the DFG must be minimal.

These properties aim to maximize the fitness and precision of the model, while also ensuring there are no deadlocks. To achieve these three properties within the generated DFG, Split Miner uses a modified version of the Dijkstra algorithm. [Augusto et al., 2019]

Splits Discovery
The last two steps of the algorithm focus on transforming the DFG into a BPMN model. To understand these steps its first understand the precise definition of a BPMN model as seen in Definition 4.4.

Definition 4.4 (BPMN Model)**.** A BPMN model is a connected graph $M = (i, o, T, G, E_m)$, where i is the start event, o is the end event, T is a non-empty set of tasks, $G = G^+ \cup G^\times \cup G^\circ$ is the union of the set of AND gateways (G^+), the set of XOR gateways ($G^\times$) and the set of OR gateways (G°), and $E_m \subseteq (T \cup G \cup \{i\} \times (T \cup G \cup \{o\})$ is the set of edges. Further, given $g \in G$, g is a split gateway if it has more than one outgoing edge. [Augusto et al., 2019]

Before defining the splits, the algorithm defines the tasks T and edges E. The set of tasks T is defined by the set of nodes present in the generated DFG plus a start and end event. The set of edges E_m is defined by the set of edges of the DFG plus the two new edges that connect the first and last nodes of the DFG to the start and end events in the BPMN respectively.

A split gateway is a point in a process model where a single flow diverges into multiple paths. Split Miner identifies these split gateways by using the concurrency relations found earlier. The key idea is that if we know which activities occur after a split, we can determine the type of split. This is done by checking whether its successors are concurrent or mutually exclusive. For instance, consider an XOR gateway followed by activities A and B. If the list of activities that can run concurrently with A is the same as the list for B, it indicates that A and B are mutually exclusive, meaning only one of them can occur at a time. On the other hand, if the concurrency lists for activities C and D include each other, it suggests that C and D can run concurrently. This implies that the split was an AND split, allowing parallel execution of C and D.

Joins Discovery
A BPMN model always ends with a single end event. So, if there are splits, there should also be joins. Finding the joins is done by identifying which tasks have multiple incoming edges. To find these tasks, Split Miner uses the Refined Process Structure Tree (RPST) [Polyvyanyy et al., 2011]. This is a tree representation of the BPMN model, in which each node represents a single-entry single-exit (SESE) fragment of the model. So, the root node contains the entire model, because a BPMN model always has single start and end node. The children of this root node contain the next largest SESE fragment, etcetera. Split Miner uses the RPST by going through its tree from leaves to root (bottom-up). For each node it analyzes the SESE fragment, and where a task is found with two or more incoming edges, a join gateway is created. Subsequently, all the incoming edges are redirected to this join gateway. This leaves us with the completed BPMN model.

4.2.5 PM4Py

After obtaining the process model, it is translated into natural language for use as input for the LLM. PM4Py provides functionality to translate various process models into natural language. The output of the Split Miner algorithm is in BPMN format. However, PM4Py does not currently support the abstraction of BPMN models into textual format. This is a limitation, because BPMN includes additional information that could improve the LLM's output.

Given this constraint, we convert the process model into Petri net format, which PM4Py can effectively handle. Although BPMN has its advantages, Petri nets are the closest available alternative that maintains the structural integrity of the process model, while also being available in PM4Py. Compared to DFGs, Petri nets are more similar to BPMN because both Petri nets and BPMN models are designed to represent processes. While DFGs are used for analyzing processes.

```
 If I have a Petri net:

places:  [ ent_node_05fa37dc-55ca-4bf5-b13d-4edefd8de021,
ent_node_0d91f028-d522-43a2-9a51-dfba1c121bc7,
ent_node_1204fd05-2462-430c-b057-7dcd5e46cad7,
ent_node_1aa47a2d-5667-484f-99a3-cb3fc8d09153,
ent_node_1ff1b219-aff8-4a06-91ae-4fa3e32d18ca,
ent_node_29bfdc9f-316a-4f0e-b852-3ac63dda22f6,
ent_node_53908bf8-f90a-47f7-bfb8-f1e4a2634a83,
ent_node_59d0620c-a507-4894-986a-e18791f57cbf,
ent_node_5a93b378-a252-4233-84a8-b90066afec25,
ent_node_80fda2e8-dd16-4fba-abf6-db9cf30eda69,
ent_node_8a5c0cd6-456c-4308-9f63-b9cf52a52829,
ent_node_8ce53d1f-0972-4c95-b66d-6af9c73952f7,
ent_node_a190fb86-d463-429f-89b0-9a3892e63ebc,
ent_node_c6e9dcd2-c3cf-4547-bcf2-a695bc81e8ee,
ent_node_e3a7a475-2b13-4d3f-b215-76fa62dcde21,
ent_node_e5176d81-33ff-44e0-9365-eb5ae04db8a9,
ent_node_f388e83e-8494-464d-ba74-ef751b041049,
exi_node_237a48db-f9e9-4026-8207-81b9336d5e2b,
exi_node_8826e160-977d-48bd-a5fa-f72c17836daa,
exi_node_8e22a760-b969-4cef-a8a2-4686cb514d87,
exi_node_90f55cd1-008e-4b39-8fbc-0ec44320efa1,
exi_node_94513847-a879-4e49-8062-7a3f7309ee22,
exi_node_b3138561-fff8-4f94-90bc-d7367564e21f,
exi_node_d20111b2-5661-48b1-b955-b56305e9a85e,
exi_node_d9e66cec-db0b-4708-99ca-50ca3fc6b152,
exi_node_daad3d93-d7ea-4b8e-9162-b202f8c3777c, sink, source ]
```

Figure 12: First part of the abstraction for the BPI Challenge 2017 dataset. The definition of the places.

Places

Figure 12 contains the definitions for the places found in the Petri net. A fundamental difference between BPMN models and Petri nets is the Petri net's use of tokens and places. Even though these elements are not present in our BPMN model, adding them does not fundamentally change the meaning of the depicted process. Finally, we identify two special places: the source and the sink. This indicates that the generated abstraction represents a Workflow net.

```
 transitions:  [ (node_0a202bb8-6a8c-46cc-a0bc-0906e21a45f8,
'A_Denied'), (node_0d91f028-d522-43a2-9a51-dfba1c121bc7,
'A_Accepted'), (node_1f3e4937-8d72-444e-b543-2ec26d6e2720,
'O_Returned'), (node_1ff1b219-aff8-4a06-91ae-4fa3e32d18ca,
None), (node_245a6b4d-cfda-4c5f-966f-4d478d166b41, 'A_Create
Application'), (node_29bfdc9f-316a-4f0e-b852-3ac63dda22f6,
'O_Cancelled'), (node_2be9bd94-01c0-48f6-909c-1763a1c0bef7,
'O_Sent (online only)'), (node_2f89e937-d841-4857-9179-16ebe79bc1c8,
'O_Sent (mail and online)'), (node_36c14719-4df8-44b9-80d8-195e909226ae,
'A_Incomplete'), (node_4cc590f6-e6bc-4f17-985c-c2a2afdecf3e,
'W_Complete application'), (node_53908bf8-f90a-47f7-bfb8-f1e4a2634a83,
'A_Pending'), (node_59d0620c-a507-4894-986a-e18791f57cbf, None),

...

(node_e3a7a475-2b13-4d3f-b215-76fa62dcde21, 'O_Refused'),
(node_e5176d81-33ff-44e0-9365-eb5ae04db8a9, None),
(node_f388e83e-8494-464d-ba74-ef751b041049, None),
(node_fa7bb3a8-10d5-4824-b505-e2362c6902d1, 'W_Validate
application'), (sfl_node_24d8cefb-7d74-4199-a078-5fe6e0f10333,
None), (sfl_node_655a7462-0b18-4673-9722-9f5811a511d8,
None), (sfl_node_99bc46ec-1d92-434e-873f-f5a14f4da62d,
None), (sfl_node_c1f3d63e-972d-4ac8-8c9c-c957733a03a7,
None), (sfl_node_d3288ab0-3b77-4e6d-ab72-4e4a7e6b8d2a,
None), (sfl_node_f0679e6e-a5b0-4b97-92f6-367fe3aefe28, None),
(sfl_node_f0a75256-86ad-4e10-a5dc-b9051d1035fe, None) ]
```

Figure 13: Second part of the abstraction for the BPI Challenge 2017 dataset. The definition of the transitions.

Transitions

Figure 13, contains the definitions for the transitions present in the Petri net. The transitions correspond to the activities or tasks from the BPMN model. In the abstraction we find several nodes with the activity name “None”. These nodes are artifacts created during the conversion from BPMN to Petri net. In Petri nets, two places cannot follow each other. There must always be a transition between two places. However, in BPMN models, a split gateway and a join gateway can directly follow each other. If we directly convert such a model to a Petri net, it would be invalid because it would lack a transition between two places. To ensure a valid Petri net, new nodes are created during the conversion process in such cases. Figure 14 shows a situation in which these artifacts are generated. Although these artifacts do not prevent the system from understanding the process, the noise they create likely negatively impacts the system's performance. This is a downside of converting the BPMN model into a Petri net to generate the abstraction.

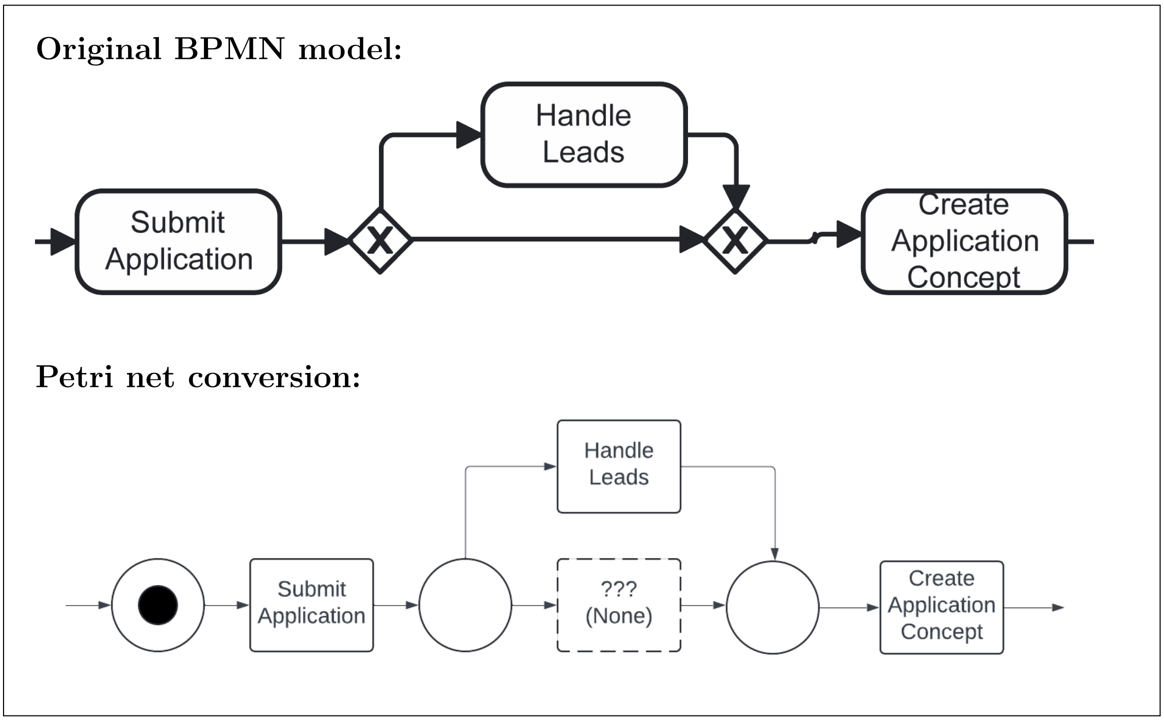


Figure 14: Example situation that generates artifact.

Arcs

Figure 15 shows the third and final part of the textual abstraction, the arcs. Additionally, the initial marking and final marking are defined. The initial and final marking are assigned to the source and sink nodes respectively. Both the initial and the final marking are assigned a single token. This means that the Petri net starts with one token in the source place and is completed when one token is present in the sink place.

```
 arcs:  [ (node_0a202bb8-6a8c-46cc-a0bc-0906e21a45f8,
'A_Denied')->ent_node_e3a7a475-2b13-4d3f-b215-76fa62dcde21,
(node_0d91f028-d522-43a2-9a51-dfba1c121bc7,
'A_Accepted')->ent_node_c6e9dcd2-c3cf-4547-bcf2-a695bc81e8ee,
(node_1f3e4937-8d72-444e-b543-2ec26d6e2720,
'O_Returned')->exi_node_94513847-a879-4e49-8062-7a3f7309ee22,
(node_1ff1b219-aff8-4a06-91ae-4fa3e32d18ca,
None)->ent_node_e5176d81-33ff-44e0-9365-eb5ae04db8a9,
(node_245a6b4d-cfda-4c5f-966f-4d478d166b41, 'A_Create
Application')->exi_node_90f55cd1-008e-4b39-8fbc-0ec44320efa1,
(node_29bfdc9f-316a-4f0e-b852-3ac63dda22f6,
'O_Cancelled')->ent_node_e5176d81-33ff-44e0-9365-eb5ae04db8a9,
(node_2be9bd94-01c0-48f6-909c-1763a1c0bef7, 'O_Sent (online
only)')->ent_node_5a93b378-a252-4233-84a8-b90066afec25,
(node_2f89e937-d841-4857-9179-16ebe79bc1c8, 'O_Sent (mail
and online)')->exi_node_b3138561-fff8-4f94-90bc-d7367564e21f,
(node_36c14719-4df8-44b9-80d8-195e909226ae,
'A_Incomplete')->exi_node_237a48db-f9e9-4026-8207-81b9336d5e2b,
(node_4cc590f6-e6bc-4f17-985c-c2a2afdecf3e, 'W_Complete
application')->ent_node_1aa47a2d-5667-484f-99a3-cb3fc8d09153,
(node_53908bf8-f90a-47f7-bfb8-f1e4a2634a83,
'A_Pending')->ent_node_1ff1b219-aff8-4a06-91ae-4fa3e32d18ca,
(node_59d0620c-a507-4894-986a-e18791f57cbf,
None)->ent_node_a190fb86-d463-429f-89b0-9a3892e63ebc,
(node_5a93b378-a252-4233-84a8-b90066afec25,
None)->ent_node_1204fd05-2462-430c-b057-7dcd5e46cad7,
(node_6c72e339-030d-4c0a-bfc2-32821af0d5b6,
'A_Cancelled')->ent_node_29bfdc9f-316a-4f0e-b852-3ac63dda22f6,
(node_73638630-7169-4ebf-85e1-8352bdac84dd,
'A_Validating')->exi_node_daad3d93-d7ea-4b8e-9162-b202f8c3777c,
(node_80fda2e8-dd16-4fba-abf6-db9cf30eda69,
None)->ent_node_1204fd05-2462-430c-b057-7dcd5e46cad7,
(node_86a1cb0f-d7c4-4666-a830-9e5830548618, 'A_Complete')->

...

(node_870cfe03-5032-4875-99a9-b91f5bcbaaf7, 'O_Accepted'),
source->(node_245a6b4d-cfda-4c5f-966f-4d478d166b41, 'A_Create
Application') ]

initial marking:  ['source:1']
final marking:  ['sink:1']
```

Figure 15: Third part of the abstraction for the BPI Challenge 2017 dataset. The definition of the arcs and initial and final markings.

4.2.6 Prompting and Large Language Model

This section addresses the final component of the PPO-system architecture. This includes the implementation of the LLM and the specific prompting strategies used. The input to this stage consists of the contextual information gathered in the preceding steps. While the output is a timely and actionable recommendation that aims to negate undesirable outcomes.

Large Language Model

In this research, we used the pre-trained `gpt-3.5-turbo-0125` model by OpenAI. This model is currently openly available and is widely recognized for its robustness. Using an openly available model also enhances the reproducibility of this research. The PPO-system implements the ChatGPT model through Azure OpenAI.

Prompting Strategy

The prompting strategy is highly influenced by the data and inputs available for the system. The PPO-system uses a structured approach to breaking down the context into manageable chunks. Initially, the system interprets the overall process model to get an understanding of the workflow. Subsequently, it focuses on the individual events flagged by the alarm system for intervention. This methodology is inspired by the least-to-most prompting technique [Zhou et al., 2022], which incrementally provides more information to guide the model's responses.

Additionally, the PPO system uses a few-shot prompting approach, where it includes one example response to guide the LLM. While our goal was to use zero-shot learning to maintain the system's generalizability and adaptability to various scenarios, this approach did not yield satisfactory results during testing. The LLM did not understand its role and instead tried to analyze and provide an overview of the process. Therefore, the few-shot method was adopted to improve the relevance of the recommendations. The instructions provided to the LLM consist of three main components:

- The **system message**, that informs the model of its tasks and goals.
- The **example prompt** and **example answer**.
- The **prompt** for which the model should form a response.

Figure 16 shows the system message. This message informs the model that its role is to assist process executors in finding actionable recommendations for day-to-day operations. Positioning the system as an operational use case (day-to-day operations). Additionally, the model is briefed on the desirable and undesirable outcomes of the process. Finally, it is provided with the previously extracted process model.

Assistant is an intelligent chatbot designed to help executors of a process find an actionable recommendation to improve the outcome of the process.
The recommendation should be applied during day-to-day process executions.
The process executor does not have the authority to make large changes to the overall structure of the process.
The executor is able to contact the customer.
The executor can only intervene manually and does not have the ability to automate parts of the process.
The desired outcome is: O_Accepted. In as few steps as possible.
The undesired outcomes are: O_Cancelled and O_Refused.
The process in question is described by the following Petri net:
...
`Petri net abstraction`
...

Figure 16: System message used for BPI Challenge 2017 dataset.

Figure 17 shows the example prompt and example answer. The model is told that a trace is likely to end in an undesirable outcome and is asked to provide a actionable recommendation. The example answer demonstrates a possible recommendation for the given trace. This helps the model understand the type of response that is expected. The third part of the instruction, the actual prompt, is the same as the prompt in Figure 17, except the event trace is replaced with the trace that triggered the alarm system.

Prompt:
Without intervention, the following active process trace will end in a negative outcome:
Created: A_Create_Application -> statechange: A_Submitted -> Created: W_Handle leads -> Deleted: W_Handle leads -> statechange: A_Concept -> statechange: A_Accepted -> Created: O_Create Offer -> statechange: O_Created -> statechange: O_Sent (online only)
Please give me an actionable recommendation to improve the outcome of this process.

Answer:
Send a reminder e-mail to the customer to inform them about the current status of the application. So they do not forget to respond to the previous mail.

Figure 17: Example prompt and example answer for BPI Challenge 2017 dataset.

4.3 Evaluation

This section discusses our method for evaluating the PPO-system. Given the absence of real-time data, we used a simulation approach to mimic real-life scenarios. Additionally, we conducted a survey to measure the perceived quality of recommendations generated by the system.

4.3.1 Simulation

Due to the lack of real-time data availability for this research, we simulate a real-life scenario to evaluate the system. In a real-world setting, the system would check for necessary interventions each time a new event is registered. To mimic this, we select a random trace from the evaluation dataset and sequentially present its events to the system, one by one, along with all preceding events. Each time a new partial trace of events is provided to the system, predictions are generated for each event in the trace. If any prediction exceeds the threshold value, the alarm is triggered. Pseudocode for this simulation is shown in Algorithm 1.

Algorithm 1 Pseudocode for simulation algorithm.

```
for trace in random_traces do
    events_analyzed = 0
    for event in trace_events do        ▷ Get predictions for all previously analyzed events
plus the next one.
        predictions = get_predictions(event, events_analyzed + 1)
        if predictions.last_event_prediction ≥ threshold then
            raise_alarm()
        end if
    end for
end for
```

4.3.2 Survey

For evaluating LLMs there are two options, human evaluation or automatic evaluation. Automatic evaluation requires prior knowledge on the desired outcome. In other words, we need to know what the right interventions are. Unfortunately, this information is not available for this dataset. Therefore, human evaluation remains as the most fitting approach for this research. However, for this approach domain experts are needed. The BPI Challenge 2017 dataset, captures a relatively simple and understandable process. This gives us a large pool of potential evaluators. Therefore, we will use this dataset.

Human Evaluation
To assess the effectiveness of the system's recommendations, we conducted a survey in which participants were asked to rate the provided suggestions. The recommendations included in the survey represent the first ten results generated by the system during the evaluation phase and were therefore selected randomly. The full survey can be found in Appendix C.

Temperature
To evaluate if the temperature (t) parameter has an impact on the perceived quality of the recommendations, half of the recommendations in the survey were generated using $t = 0.3$, while the other half were generated using $t = 0.7$. The default temperature setting used by ChatGPT is 0.7. Since $t \in [0, 1]$, a temperature of 0.3 serves as a complementary value to 0.7. This complementary relationship ($1 - 0.7 = 0.3$) was the reason for the choice of these parameter values. We opted to only evaluate these two temperature settings to limit the size of the survey. A longer survey might discourage participants to take part, decreasing the potential sample size.

Structure
The survey begins with a brief explanation of the extracted process model. Respondents are then asked to evaluate ten recommendations based on six metrics. Each metric is rated on a Likert scale ranging from *Very Low* to *Very High*. The six metrics are:

- **Actionability in day-to-day operations**: Can a worker implement this recommendation in a day-to-day situation? High scoring recommendations on this metric focus on immediate, routine tasks. They ensure that the recommendations can be quickly adopted without extensive changes or strategic planning. With the goal of improving everyday efficiency and effectiveness.

- **Usability in tactical operations**: Is the recommendation useful for improving the process from a process design view? High scoring recommendations on this metric are concerned with broader, process improvements that support strategic objectives. These recommendations require more planning and resource allocation and are aimed at long-term process optimization.

- **Relevance**: Is the recommendation applicable to the presented situation? Relevant recommendations are contextual appropriate, which increases the likelihood of adoption and success.

- **Expected improvement to process execution**: Does the recommendation improve the performance of the process? Good recommendations have a high potential impact.

- **Specificness**: Does the recommendation specifically describe what has to be done? Specific recommendation have high clarity, therefore reducing implementation errors.

- **Proportionality**: Are the required resources associated with executing the recommendation proportional to the presented situation? Proportionate recommendations balance benefits with resource requirements, ensuring feasible and sustainable improvements.

Use Case
The metrics *Actionability in day-to-day operations* and *Usability in tactical operations* are designed to evaluate the system's recommendations within the framework of the BPM pyramid, which distinguishes between operational and tactical use cases. The primary goal of this research is to develop a system that provides actionable recommendations suitable for day-to-day operational use. This means generating solutions that can be readily implemented to address immediate and routine issues within business processes.

However, through our observations, we discovered that the system's recommendations do not always align with this operational focus. Some recommendations appear to be more suited for tactical considerations, which involve broader, strategic changes to the process rather than quick fixes. Certain problems may inherently require more complex solutions that extend beyond simple operational adjustments and necessitate modifications to the overall process structure. By measuring both operational and tactical performance across the different temperatures, we can investigate the impact of the temperature setting on the system's recommendation behavior. By examining how different temperature settings affect the bias towards operational or tactical recommendations, we can optimize the system to better fit the intended use case. This allows for fine-tuning the system in the future.

Performance
The remaining metrics are designed to measure the performance of the system, ensuring that the recommendations generated by the LLM are effective and actionable within a process improvement context. Good recommendations should be *relevant*, to address the problem at hand. They should *improve the execution of the process*. They should be *specific*, so even an untrained worker can execute them. Finally, they should be *proportional*. The potential gain of executing a recommendation should outweigh its potential cost.

5 Results

In this section we analyze the outputs generated by the PPO-system. Examining the system's ability to identify problematic traces and generate corresponding recommendations in natural language. To understand the analysis it is important to understand the process for which the recommendations were generated. So, we will explain the process captured in the BPI Challenge 2017 dataset. By qualitatively analyzing the outputs, we aim to provide an understanding on how the system operates and the nature of its recommendations. Then, we will present the results of the survey to assess the perceived quality of the recommendations. By analyzing these results, we can evaluate the effectiveness of the system, highlight its strengths, and identify areas that need improvement.

5.1 The BPI 2017 Process

The dataset describes a loan application process within a Dutch financial institution. To make understanding the model easier, we have divided it into three distinct phases, pictured in Figure 18. The BPMN model in this figure was generated by the PPO-system.

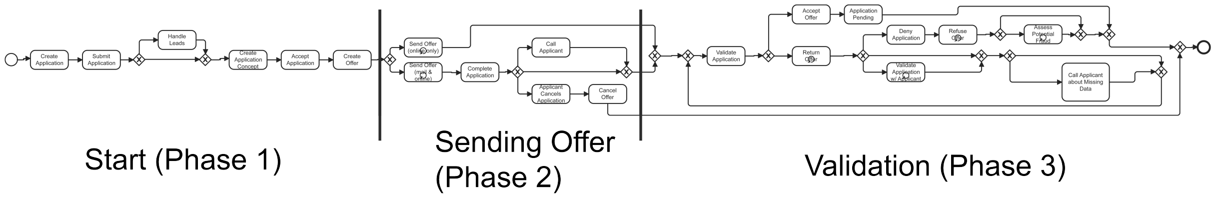


Figure 18: The BPMN process model split up into three distinct phases.

The Start of the Process (Phase 1)
Each process starts with the creation of a loan application, as depicted in Figure 19. Once the application is submitted, an employee of the financial institution may contact the customer for additional information or clarification. This step ensures that all necessary details are accurate and complete before proceeding. Following this interaction, a concept application is generated within the institution. This preliminary application undergoes internal review and adjustments to meet the institution's criteria. Once the concept application is finalized and approved, the formal loan application can be accepted. Subsequently, an offer is created and prepared to be sent to the customer.

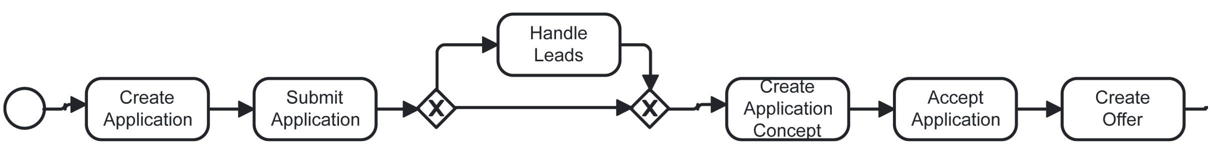


Figure 19: The starting phase of the process.

Sending the Offer to the Customer (Phase 2)

The offer can be sent to the customer through two methods: online only, or both mail and online. When the offer is received through both mail and online, the customer has the option to cancel the offer, which is considered one of the undesirable outcomes in the process. This phase of the process, including the methods of delivery and the potential for cancellation, is depicted in Figure 20.

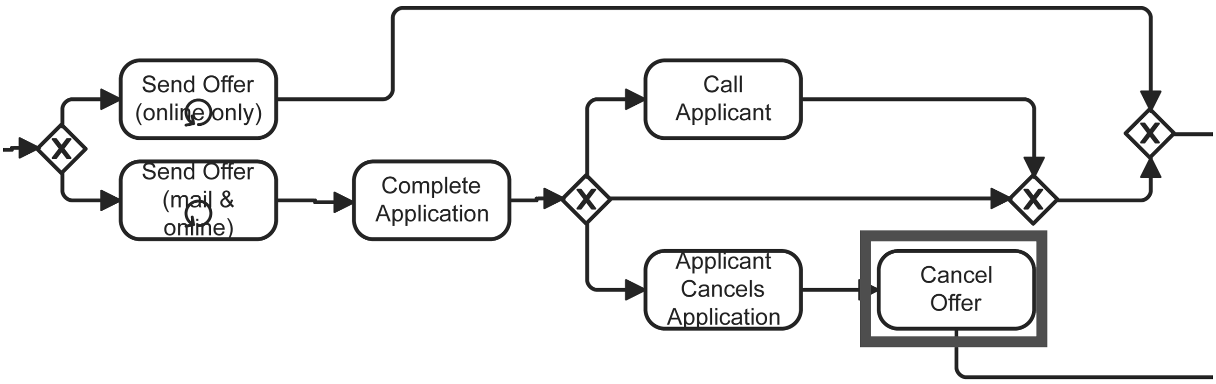


Figure 20: The second phase of the process, sending the offer to the customer.

Figure 20 reveals a potential issue in the model generation: the option for the customer to decline the offer is only present when the offer is sent via both mail and online. Upon generating the model, there was likely no available data in the training dataset showing instances where the offer was sent exclusively online and subsequently declined by the customer. This lack of data highlights a limitation in the model's representation of the process, suggesting that it might not fully capture all possible customer interactions and outcomes. During real-world implementation, the model should ideally be updated manually after thorough investigation. However, we lack knowledge of the actual process and cannot investigate the real flow. Therefore, we have chosen to leave this potential error in the model. This also avoids overfitting to this specific dataset, leading to more general results.

Validating the Offer (Phase 3)

The final part of the process, depicted in Figure 21, encompasses the validation and finalization of the application process. After the finalized application is validated there are two options: the institution accepts the offer and the loan is granted (desirable) or the offer is returned. After the offer is returned the institution either completely denies the application (undesirable) or attempts to update the application so it can be accepted later. If the institution decides to update the application, the customer is contacted to obtain any missing data. With the missing data obtained, the institution decides again to either refuse or accept the application.

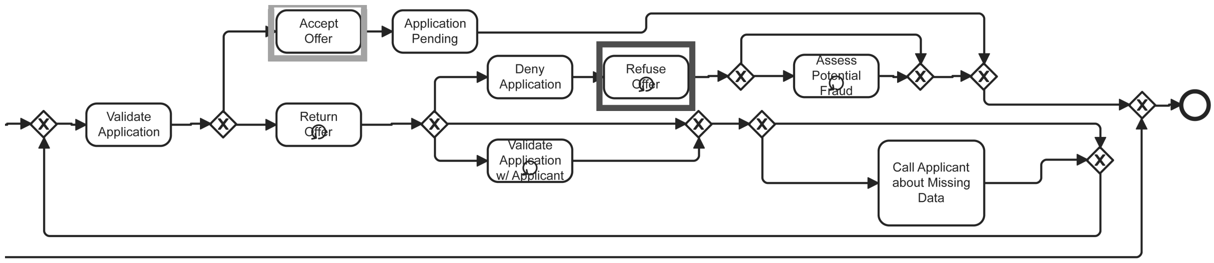


Figure 21: The third phase of the process, validating the offer.

5.2 Qualitative Analysis

Our goal was to create a system suited for generating day-to-day recommendations. Therefore, we analyze which metrics in the survey most contribute to positively perceived day-to-day recommendations. Additionally, we aim to identify areas that need improvement. We differentiate between use case metrics and performance metrics, as discussed in Section 4.3.2. The use case metrics evaluate the recommendations based on their applicability to either operational or tactical use cases, while performance metrics assess the overall performance of the system. We present several example outputs, highlighting recommendations that exhibit high response saturation in the use case metrics “Actionability in day-to-day operations” and “Usability in tactical operations”. We will also analyze possible correlations between the use case metrics and the performance metrics. By understanding what drives this saturation, we can inform future few-shot prompting strategies and refine the system for better operational effectiveness. Allowing for improvement of the PPO-system.

5.2.1 Actionability in Day-to-day Operations

Figure 22 shows a recommendation with a high positive saturation for the metric ”Actionability in day-to-day operations.” Meaning responses are concentrated around the *High* and *Very High* ratings. Furthermore, there are no negative responses for this metric. Another metric that stands out in this example is the “Specificness” metric. Intuitively, this makes sense. A good day-to-day recommendation should be specific, so it can be implemented quickly, correctly, and effectively. This might explain the positive saturation found in the “Actionability in day-to-day operations” metric. Analyzing the recommendation, it seems to align with the survey responses. The recommendation clearly states what a worker should do: send a confirmation email to the customer. These findings suggests a correlation between the metrics “Actionability in day-to-day operations” and “Specificness”. However, Figure 23, which also displays a recommendation with high positive saturation for the day-to-day metric, does not suggest this as strongly, as there are negative ratings present.

Context:
An offer was created and sent via mail and online. After this the trace deviated from the known process model. First a worker calls the customer. Then the application is completed. The system detected the deviation and gives the following recommendation.

Recommendation:
Send a confirmation email to the customer once the offer is created to make sure they received it and are aware of the next steps in the process. This will ensure that they remain engaged and reduce the likelihood of the application being forgotten or overlooked.

Survey Response:

	Very Low	**Low**	**Neutral**	**High**	**Very High**
Actionability day-to-day	0%	0%	20%	20%	60%
Specificness	0%	10%	10%	40%	40%

Figure 22: Recommendation with high positive response saturation in day-to-day operations.

Context:
An offer is made, which is sent to the customer via mail and online. The application is completed and enters the validation stage, where it is returned. After this the process execution deviates from the known process model, and a new offer is created. The system detects this anomaly and gives the following recommendation.

Recommendation:
It seems that there is a problem with the offer creation process, as it is being cancelled and recreated multiple times. One possible recommendation is to review the criteria used to create the offers and ensure that they are accurate and complete. Additionally, it may be helpful to involve a supervisor or manager in the offer creation process to provide oversight and ensure that the process is running smoothly. Finally, it may be helpful to streamline the process by automating some of the steps, such as sending reminders to incomplete files or validating applications.

Survey Response:

	Very Low	**Low**	**Neutral**	**High**	**Very High**
Actionability day-to-day	0%	0%	10%	90%	0%
Specificness	10%	30%	10%	50%	0%

Figure 23: Second recommendation with high positive response saturation in day-to-day operations.

Correlation with Specificness
Using the Spearman's rank correlation coefficient [Spearman, 1961], we can calculate the correlation between variables measured in a Likert scale. Figure 24 shows the results of this calculation for the variables "Actionability in day-to-day operations" and "Specificness".

Correlations

			Daytoday	Specificness
Spearman's rho	Daytoday	Correlation Coefficient	1.000	.488**
		Sig. (2-tailed)	.	<,001
		N	99	99
	Specificness	Correlation Coefficient	.488**	1.000
		Sig. (2-tailed)	<,001	.
		N	99	99

**. Correlation is significant at the 0.01 level (2-tailed).

Figure 24: Results of Spearman's rank correlation coefficient for "Actionability in day-to-day operations" and "Specificness".

- H_0: There is no positive correlation between the metrics "Usability in day-to-day operations" and "Specificness"
- H_A: There is a positive correlation between the two metrics.

Based on the correlation coefficient of 0.488 we observe a positive correlation between the two metrics. However, this correlation is weak. In this context $n = 99$, since we have 10 respondents who all rated 10 recommendations. (One rating is missing.) The correlation is statistically significant, since the p-value is smaller than 0,01. So, we reject our null hypothesis. We conclude that the specificness of a recommendations can be a weak predictor for its usefulness in day-to-day operations.

The analysis revealed no statistically significant correlation between the metrics "Usability in tactical operations" and "Specificness", as shown in Figure 25. This is possibly explained by the nature of tactical recommendations, which are usually broader and more abstract. Therefore, their value does not primarily depend on specificness. Furthermore, the absence of a negative correlation can also be understood in this context. While it might be expected that more abstract recommendations would lack specific details, the nature of tactical recommendations allows for a range of specificity levels without impacting their overall usability. Tactical recommendations can still be valuable and actionable even if they vary in their level of detail. Their effectiveness is more closely tied to their alignment with strategic objectives rather than the granularity of the actions. Therefore, the variability in the specificness of tactical recommendations does not detract from their overall usability, leading to the absence of a negative correlation as well.

Correlations

			Specificness	Tactical
Spearman's rho	Specificness	Correlation Coefficient	1.000	.025
		Sig. (2-tailed)	.	.810
		N	99	98
	Tactical	Correlation Coefficient	.025	1.000
		Sig. (2-tailed)	.810	.
		N	98	98

Figure 25: Results of Spearman's rank correlation coefficient for "Usability in tactical operations" and "Specificness".

5.2.2 Usability in Tactical Operations

Figure 26 presents a recommendation with a high positive response saturation on the metric "Usability in tactical operations".

Context:
A worker calls the customer to retrieve the missing data. After this the offer is revalidated. Again, data is missing. So, it seems like the first attempt to validate the customer and application data was unsuccessful.

Recommendation:
It seems that there are several steps in the process where the application is delayed or incomplete. One recommendation would be to implement a system to track the progress of the application and send reminders to the customer if there are any delays or missing information. This could help ensure that the application is completed in a timely manner and reduce the likelihood of negative outcomes. Additionally, it may be helpful to review the process to identify any bottlenecks or areas for improvement.

Survey Response:

	Very Low	Low	Neutral	High	Very High
Usability Tactical	0%	0%	0%	70%	30%

Figure 26: Recommendation with high positive response saturation on metric "Usability in tactical operations".

A qualitative analysis of the recommendation reveals the reasons for the high positive response saturation on the metric "Usability in tactical operations". The recommendation includes actions such as "implementing a system to track the progress of the application" and "reviewing the process", both of which are associated with strategy and tactical planning rather than tasks that a frontline worker can execute. This is noteworthy since the system message, as defined in Section 4.2.6, clearly states that "the recommendation should be applied during day-to-day process executions" and "the process executor can only intervene manually". Showing that the LLM can deviate significantly from its instructed tasks.

These recommendations, with positive saturation for the metric "Usability in Tactical Operations", pose a significant obstacle to the implementation of LLM-based recommendation engines in real-world situations, as they fail to align with the operational needs and capabilities of frontline employees. Therefore, they should ideally be automatically filtered out, before the system can be adopted in real-world scenarios.

5.2.3 Improvement Areas

The main goal of the PPO-system is to generate recommendations, that can be implemented in day-to-day operations and improve the execution of the process. Therefore, an important area for improvement is the specificness of the generated recommendations. As we have observed that recommendations rated higher in specificness also tend to rate higher in actionability for day-to-day operations. Furthermore, there are some recommendations that show a high positive saturation for use in tactical operations. Automatically filtering out these recommendations is crucial before the system can be implemented in the real world.

5.3 Survey Results

In this section, we present the survey results, focusing first on the system's overall performance before comparing results between $t = 0.3$ and $t = 0.7$. The results presented were obtained with a sample size of $n = 10$, where n is the number of respondents. The respondents are primarily IT consultants. While they are generally knowledge about processes and process improvement, they might not have specific knowledge about the financial industry. Table 3 displays the mean values for the different metrics. The first column shows the means for all recommendations combined. The next two columns present the results for recommendations generated with $t = 0.3$ and $t = 0.7$, respectively.

	Combined	$t = 0.3$	$t = 0.7$
Day-to-day	0.070707	0.380000	-0.244898
Tactical	0.469388	0.591837	0.346939
Relevance	0.646465	0.820000	0.469388
Improvement	0.414141	0.440000	0.387755
Specificness	-0.080808	-0.080000	-0.081633
Proportionality	0.292929	0.360000	0.224490

Table 3: Metric means.

To evaluate the results, the different categories on the Likert scale were converted into integers, with *Very Low* assigned a value of -2 and *Very High* a value of 2. To interpret the categories on the Likert scale, we assume that metrics rated as *High* perform well and can be implemented in real-world solutions, while *Very High* indicates that significant value is added to the execution of processes. Conversely, *Low* and *Very Low* ratings would negatively impact process execution. The *Neutral* category implies that no value is added or subtracted, suggesting that implementing such a system would offer no benefit to companies.

5.3.1 Combined temperatures

First, we analyze the overall means, that include both the results from $t = 0.3$ and $t = 0.7$. These results are presented in Figure 27. The means are generally concentrated around 0, or *Neutral*. Relevance is the only metric surpassing the 0.5 mark. Furthermore, all metrics, except specificness, are positive, meaning they are closer to *High* than *Low*. Specificness is the lowest scoring metric, indicating that the model performs poorly in delivering specific instructions. This is a significant obstacle for implementing the PPO-system in real-world scenarios, since we determined that specificness might be a weak predictor for the usability of recommendations in day-to-day operations. Enhancing the model's performance in this area could potentially be achieved by providing additional context about the company and detailing potential intervention strategies during few-shot learning. However, given that we are working with an academic dataset that lacks specific information about the company, this improvement is not feasible in this research. Furthermore, it is noteworthy that the perceived tactical applicability is higher than the perceived day-to-day applicability. This is particularly interesting since our prompting strategy was designed to position the system for operational use cases. However, according to the survey results, the system performed better in tactical use cases.

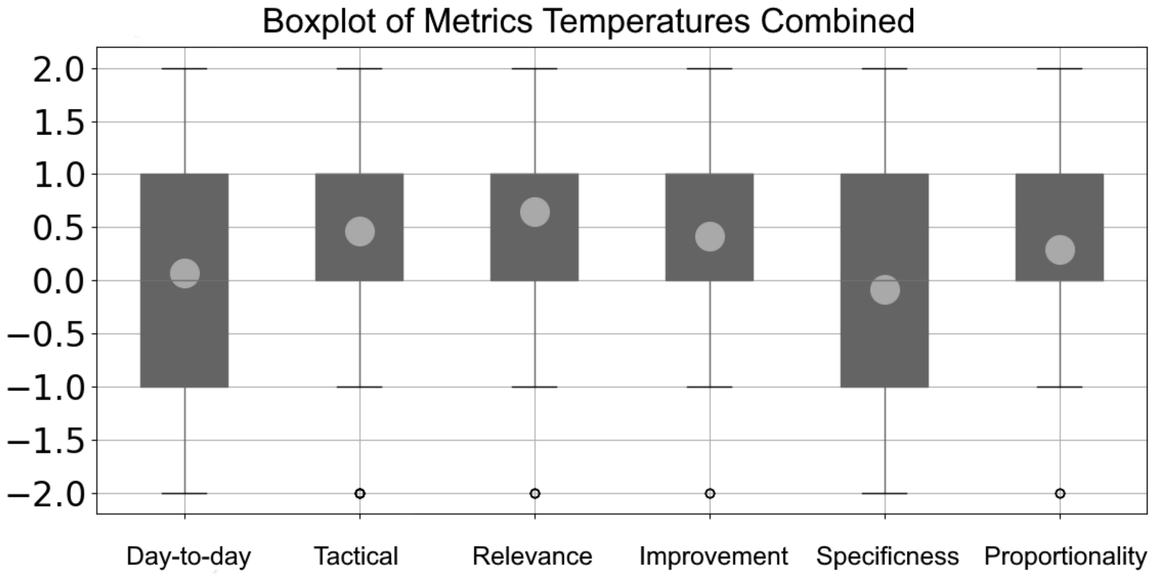


Figure 27: Combined survey results.

Overall the system exhibits a slightly positive perceived performance. This suggests there is potential for an LLM-based recommendation engine. However, none of the metrics reach an average score of *High*. So, currently, the PPO-system, as an LLM-based recommendation engine, would be unable to add value in a real-world application.

5.3.2 Comparing $t = 0.3$ and $t = 0.7$

Now we will analyze the differences in the perceived quality of the recommendations between the temperature settings $t = 0.3$ and $t = 0.7$. Results from the survey for $t = 0.3$ and $t = 0.7$ are found in Figures 28 and 29 respectively.

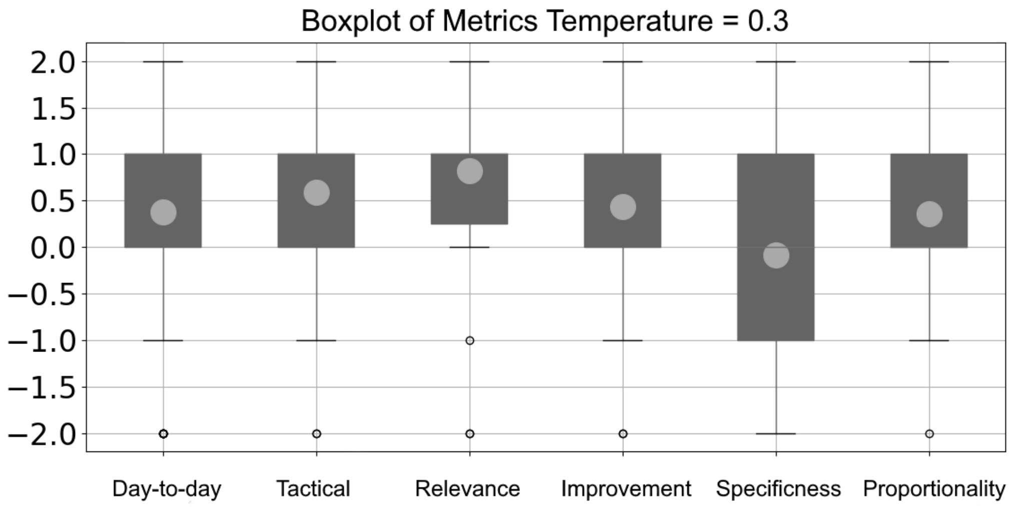


Figure 28: Survey results for recommendations where $t = 0.3$.

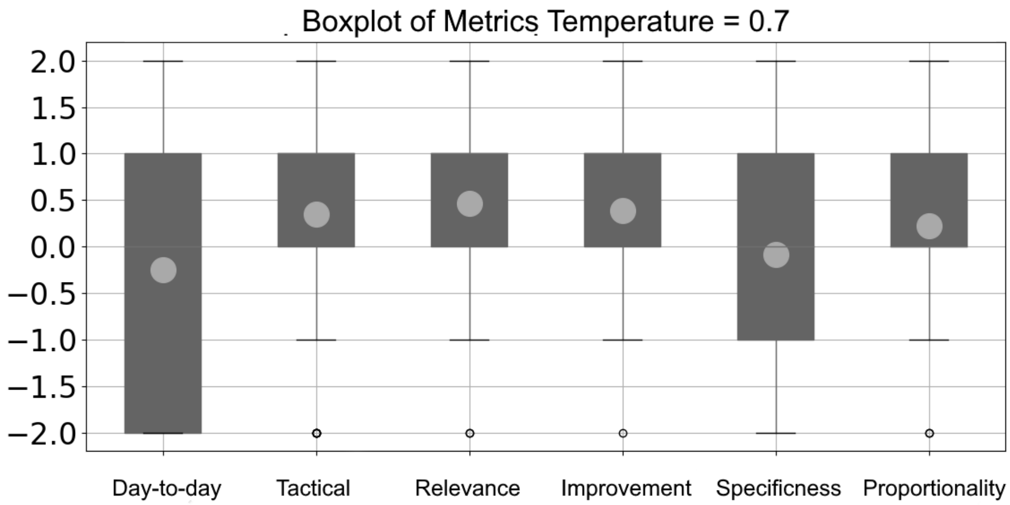


Figure 29: Survey results for recommendations where $t = 0.7$.

Operational vs Tactical
For $t = 0.3$, the "Day-to-day" metric scores 0.38, indicating that the recommendations are generally positive. This suggests that at $t = 0.3$, the system shows potential in its capability of routine decision-making and operational activities. However, 0.38 represents a mostly *Neutral* score. This means that the recommendations generated using $t = 0.3$ are also not capable of delivering value to organisations. The "Tactical" metric scores 0.59, suggesting that at $t = 0.3$ the system is better at providing recommendations for tactical use cases rather than operational use cases. The results for $t = 0.7$ show more variability and less effectiveness. The "Day-to-day" score of -0.24 indicates a negative impact, suggesting that the system's recommendations may hinder rather than help daily operations at this higher temperature setting. The"Tactical" score of 0.35 is also significantly lower than at $t = 0.3$, further indicating reduced reliability and effectiveness for both operational and tactical contexts.

Comparing the results between the two temperature settings reveals a big difference in the day-to-day applicability of the recommendations. Recommendations generated with $t = 0.3$ score on average 0.62 higher than ones generated with $t = 0.7$. This can be explained by the nature of desirable day-to-day recommendations. Which tend to be less complex and creative, since they need to be executable on short notice and with available resources. The increased"creativity" associated with higher temperature values might cause the recommendations to be less feasible for operational use cases. To confirm that there is an actual difference between the two means, we use an independent t-test. Since we are looking if there exists a difference between the two means, we use the two-sided independent t-test.

- H_0: The usability of the recommendations generated by the PPO-system in day-to-day operations is the same for both $t = 0.3$ and $t = 0.7$.
- H_A: The usability of the recommendations generated by the PPO system in day-to-day operations differs between $t = 0.3$ and $t = 0.7$.

The results in Tables 4 and 5 show the results of the test. We use a p-value of 0.05. Since $0.028 < 0.05$, we can reject the null hypothesis. Meaning it is likely that $t = 0.3$ generates better recommendations for operational use cases.

	temperature	*N*	*Mean*	*Std. Deviation*	*Std. Error Mean*
Daytoday	.3	50	.38	1.159	.164
	.7	49	-.24	1.601	.229

Table 4: Group Statistics

		F	*Sig.*	*t*	*df*	*Two-Sided p*
Daytoday	Equal variances assumed	14.205	$< .001$	2.228	97	.028

Table 5: Independent Samples Test

Performance Analysis

For $t = 0.3$, the "Relevance" metric scores highest with 0.82, indicating that the proposed solutions are generally aligned with the identified problems. The "Improvement" metric scores 0.44, showing a slight positive impact on process execution. "Specificness" and "Proportionality" score -0.08 and 0.36, respectively, highlighting that while the system provides action plans, there is room for improvement in balancing resource requirements with problem severity.

At $t = 0.7$, the "Relevance" score drops to 0.469388, and the "Improvement" metric slightly decreases to 0.387755. "Specificness" remains low at -0.081633, and "Proportionality" decreases to 0.22449. These results indicate that the higher temperature setting reduces the accuracy and overall quality of the recommendations, making them less effective for process optimization.

Consistency & Concentration

The results for $t = 0.3$ are more consistent and concentrated, with scores clustered around positive values. This concentration indicates that lower temperatures produce more deterministic and reliable recommendations, which is desirable in predictive process optimization. In contrast, the results for $t = 0.7$ are slightly more dispersed, reflecting greater variability and less reliability. Particularly for the "Relevance" metric, $t = 0.3$ shows a smaller spread, with the first quartile at 0 or *Neutral*, while for $t = 0.7$, the first quartile is at -1 or *Low*.

The analysis of the two different temperature settings reveals that a lower temperature value ($t = 0.3$) leads to more consistent, concentrated, and effective recommendations for prescriptive process optimization. Specifically, for the goal of enhancing day-to-day operations, $t = 0.3$ proves to be more effective.

6 Discussion

In this chapter, we analyze the results obtained from our research and discuss the results, limitations, and opportunities for future work. We begin by evaluating the results. This is followed by a discussion on the limitations of this research. Finally, we outline future research directions, including the potential of newer language models, real-life evaluations, and the benefits of dynamic temperature settings.

6.1 Connection to Related Work

Through a literature review by [Chapela-Campa and Dumas, 2023], we found that the field of process mining can be divided into layers. Our research specifically focuses on the PPO layer. This layer itself can be split into two steps. The first step involves finding the right timing for an intervention. The second step, generating an actionable recommendation, remained largely unsolved. Within this second step we can make another distinction. There are two types of use cases for the recommendations. Operational and tactical. Operational recommendations should focus on day-to-day operations, while tactical recommendations focus on improving processes from a birds eye view. Our research aimed to address this second step. Specifically with operational recommendations in mind. Additionally, with the rise of LLMs like ChatGPT, we sought to leverage these models. By combining efforts from different layers of process mining research with an LLM, we aimed to create a system that automatically generates timely and actionable recommendations to prevent negative outcomes in business processes. With the added benefit of generating the recommendations in natural language. Making it easy to understand for any worker.

6.2 Results Interpretation

In this section we will look at each of the hypotheses presented in Section 3. We will analyze if they can be accepted based on the results presented in Section 5.

H1: Large language models can interpret business processes well enough to generate relevant recommendations.

The results show that LLMs have some ability to interpret business processes to generate relevant recommendations. However, the survey responses average close to zero for most metrics, indicating that the results are inconclusive regarding this hypothesis. Although there are positive saturations for some of the recommendations in the "Relevance" metric, the overall neutrality of the responses suggests that LLMs may not yet interpret business processes effectively enough to consistently produce valuable recommendations. Therefore, while there is potential, H1 cannot be decisively accepted based on the current survey responses and relevance scores.

H2: Large language models can add value as recommendation engines in a Prescriptive Process Optimization context.

The evaluation of the survey results and analysis reveals that while LLMs show potential as recommendation engines in a prescriptive process optimization context, they are not yet capable of adding consistent value in the context of the PPO-system. The recommendations generated by the system were relevant and actionable in some cases, particularly in operational contexts, but the results are mixed and indicate areas for improvement. The variability in the responses and the presence of negative ratings in the survey suggest that the system's performance is not yet reliable enough to completely replace a human operator. However, the system could be valuable as an automation tool that assists human operators in tracking issues and offers a starting point for effective interventions. Additionally, in a real-life environment, the system would likely undergo iterative improvements. When experts are involved in this process, the system can serve as a tool to make their expertise accessible to all workers. By gathering expert responses to various situations, a more organization-specific prompting strategy can be developed. This approach embeds the experts' knowledge into the system, enabling less experienced workers to make informed interventions in real time. This would require expert knowledge and fine-tuning to a specific situation. However, the aim of this research is to propose a generalized system.

In conclusion, while LLMs demonstrate the potential to enhance prescriptive process optimization systems, the current capabilities of the generalized PPO-system are not sufficient to add consistent value. The results suggest that further development and fine-tuning are necessary to fully realize their potential in generating valuable, actionable recommendations for business process optimization. Therefore, we have to reject this hypothesis at the moment. However, with the right improvement, there is potential for LLMs to add value as a recommendation engine in a prescriptive process optimization context.

H3: The temperature setting in LLMs affects the performance of the LLM as a recommendation engine.

Our analysis confirms that lower temperature settings ($t = 0.3$) generally result in higher average ratings across all metrics. The survey results also showed that responses were slightly more concentrated, especially for the "Relevance" metric. Suggesting that a deterministic approach provided by lower temperatures leads to more uniform and predictable recommendations. This consistency is desirable in prescriptive process optimization, as it ensures that the recommended actions are reliable and can be depended upon to improve process execution. Therefore, H3 is accepted based on the average ratings.

6.3 Limitations

Now we will discuss the limitations of this research. Acknowledging these limitations, provides an understanding of the challenges faced during the research and highlights opportunities for future improvements.

6.3.1 Lack of Contextual Knowledge During Evaluation

The evaluation technique used in this research may not be the most effective or comprehensive for assessing the performance of the PPO-system. One key limitation was the restricted contextual knowledge available during evaluation. This constrained the ability to fully understand and interpret the system's outputs. Ideally, testing should be conducted in real-life situations where experts who are familiar with the process can provide insights. Without this expert input, the evaluation might miss important factors that influence process outcomes.

6.3.2 Limitations to Generalizability

The decision to omit traces shorter than six events from the dataset slightly limits the generalizability of the results. This cutoff was primarily influenced by the specific characteristics of the dataset used in this research. We found shorter traces less informative or relevant for our objectives. However, this exclusion may overlook scenarios where shorter processes play a significant role. Consequently, the findings of this research may not fully apply to datasets with a different distribution of trace lengths or to domains where shorter processes are prevalent. Including a broader range of trace lengths might enhance the generalizability of the results and provide a more comprehensive understanding of the system's applicability across various contexts.

6.3.3 Assumptions

This research assumes that workers can intervene at any point in the process. However, it is not clear which parts of the process are automated and thus prevent human intervention.

6.3.4 Alarm System Threshold Setting

The lack of domain knowledge about the dataset means the chosen threshold for the alarm system might not be realistic. We used a ratio of one to five between *average cost of intervention* versus *average cost of an undesired outcome*. This relatively high ratio, means that the alarm system gets triggered often. Secondly, the fixed threshold is perhaps not realistic to begin with. Since, in a real world scenario, different problems, will have different costs associated with them. Perhaps, groupings of issues would make the most sense in a real-world setting. For example: high, medium, and low priority cases, based on the probability and potential impact of an undesirable outcome.

6.3.5 Lack of Support for BPMN Abstraction

The lack of support in PM4Py for converting a BPMN into natural language is another limitation. Due to this restriction our BPMN model first has to be converted into a Petri net model. This removes the additional data captured in the BPMN model. Additionally, it also introduces artifacts in the resulting Petri net model. While these artifacts do not seem to negate the ability of the LLM to understand the process, the noise created by these artificats likely negatively impacts the performance.

6.4 Future Work

This section discusses potential future research directions. These directions focus on enhancing the performance and applicability of the PPO-system as well as more general directions regarding LLMs in process mining. We highlights technological advancements, evaluation methods, different use cases, drift detection, and dynamic temperature settings as key areas for exploration and development.

6.4.1 Leveraging Technological Advancements

Since the release of ChatGPT 3.5, more advanced models, such as GPT-4 and GPT-4o, have been developed. Concurrently, alternative implementations of LLMs, including Google Gemini and Meta's LLaMa, have emerged. Future research should investigate the capabilities of these newer models to determine if they offer improved performance and accuracy in generating tailored recommendations for process optimization. Alternatively, future work could also look into developping a LLM specifically designed for PPO tasks. It might be interesting to compare a model specifically made for PPO tasks to a more general LLM. Comparing the results of these models with those obtained using GPT-3.5 will provide valuable insights.

Converting the BPMN model directly into natural language could also improve the performance of the system. This allows the system to use the additional information contained in a BPMN model. Future research could either re-evaluate the system once BPMN support for PM4Py is introduced or develop a BPMN to natural language abstraction technique.

6.4.2 Alarm System Enhancement through Feature Engineering

Feature engineering is an aspect of machine learning that involves the creation, transformation, and selection of input variables to improve model performance. Since most of the focus of this research was invested in the LLM, we did not experiment with feature engineering. Therefore, incorporating feature engineering techniques could improve the predictive capabilities of the PPO-system, leading to more accurate and actionable recommendations.

Given our goal of creating a generalizable system, exploring automated feature engineering techniques presents a promising direction. Approaches such as AI-driven feature generation can automatically identify and construct features that may not be readily apparent through manual analysis, potentially uncovering hidden patterns and relationships within the data. The automation of this process also enhances its applicability as a general system, making it adaptable and applicable across various organizations and processes.

6.4.3 Different Sectors

The current research is based on a datasets from a financial organisation. Future work should consider expanding the diversity of datasets by including data from various industries, such as healthcare, retail, manufacturing, and technology. This will help assess the generalizability of the PPO-system. Additionally, exploring different sectors might help identify industry-specific challenges and opportunities.

6.4.4 Investigating Multimodal LLMs

This research focuses on text-based LLMs, but what if future research explored the potential of multimodal LLMs? These multimodal models can process and integrate data from various sources, such as text, images, and structured data. Integrating visual process flows with textual descriptions could enhance the model's understanding and provide more informed recommendations. These models might be able to capture nuances that single-modality models miss.

6.4.5 Exploring Ethical Considerations

An important area to explore is the ethical implications and potential biases of using LLMs in a PPO context. What if certain recommendations favor particular outcomes or stakeholders? Future research could delve into developing methods to audit, identify, and mitigate biases in LLM-generated recommendations. This includes exploring how biases might emerge from training data and how they could be addressed to ensure fair and unbiased outcomes.

Another topic related to ethical considerations is the explainability of the recommendation system. What if the system could generate not only recommendations but also transparent explanations of how it arrived at those suggestions? This could significantly improve user trust and adoption of the system, especially in high-stakes environments where understanding the reasoning behind decisions is crucial.

6.4.6 Enhancing Evaluation Methods

Conducting evaluations of the PPO-system in real-life environments, would improve the accuracy of the evaluation. This might also identify additional challenges and areas for improvement. Additionally, it will provide more context for the LLM to work with, potentially leading to more specific results. This real-life evaluation also makes involving domain experts easier. Allowing for the expansion of example prompts used during few-shot prompting, potentially leading to more specific results and better performance. This process could be implemented through a user feedback loop, where users are asked to rate and provide feedback on the recommendations after a set number of generated outputs.

The main limitation of our current evaluation method is the lack of a clear and objective way to determine the performance of an LLM in executing specific tasks, other than relying on survey data. This reliance on surveys introduces subjectivity and may not comprehensively capture the system's effectiveness. Future research should focus on developing robust evaluation frameworks tailored to assessing the performance and capabilities of LLMs. Specifically, frameworks designed for evaluating recommendations in prescriptive process optimization or process mining contexts would provide more precise and reliable insights.

6.4.7 Exploring Different Use Cases

The PPO-system performed better in tactical use cases despite being fine-tuned for operational use cases. This indicates that there might be more potential for the PPO-system in tactical use cases. Fine-tuning the system specifically for tactical use cases might show better performance.

6.4.8 Adapting in Real-time

Processes change over time, causing the predictions of the model to become less accurate over time. Future research should also investigate the optimal frequency for regenerating the model to maintain its accuracy and relevance over time.

Or a technique could be developed to adapt the system to process changes in real-time. Future research could explore the feasibility of integrating LLMs with real-time process monitoring tools, allowing for dynamic adjustments based on live data inputs. This approach could help in developing a more responsive and adaptive PPO-system that evolves alongside the processes it aims to optimize.

6.4.9 Explore Prompting Stages

Our findings indicate that a lower temperature setting generally provides more consistent and thus better results overall. However, an approach with a dynamic temperature might provide benefits. This approach would be similar to the approach by Zhu et. al [Zhu et al., 2024].

Exploring the use of dynamic temperature settings for different parts of the recommendation is therefore a valuable area for future research. For instance, different stages of generating a response could be considered. The first stage could involve understanding the situation and context, the second stage could focus on developing a general solution, and the third stage could define the specific tasks to be executed to achieve the solution. This staged approach with varying temperature settings could potentially enhance the overall performance of the system. Additionally, this staged approach might also provide benefits in other areas of the prompting process.

7 Conclusion

Our preliminary investigation presents promising results for LLMs as recommendation engines in PPO applications. This initial exploration into the topic uses relatively simple and unrefined methods. Despite this, the survey generally shows a slight positive perceived performance. However, we concluded that the PPO-system is currently unable to add significant value to the execution of business processes. Nevertheless, assuming that future research efforts and newer models will improve the performance of the PPO-system, LLMs will likely find a place in the PPO field as recommendation engines. Furthermore, our findings indicate that a lower temperature setting leads to more consistent results. Additionally, we identified multiple avenues for future research.

Currently, one the main issues is effectively setting up the PPO-system, since this requires domain knowledge. However, after it has been set up, it can serve as an efficient tool for transferring, storing, and using expertise. This means less trained personnel can operate at the same efficiency and performance level as a highly knowledgeable operator, effectively cloning the expertise of a single professional. This allows organizations to maintain high standards of efficiency and performance with fewer specialized staff.

To answer the research question: Yes, large language models can and be leveraged in process mining to determine the best actions to take when intervention is needed. However, more work is needed before our PPO-system can actually provide value to organisations. Further research and refinement are necessary to improve the system.

In the current technological landscape, it is not a question of if, but rather when, LLMs will be adapted into process mining. Therefore, this research highlights some critical areas that require focus for developing an LLM-based recommendation engine, such as improving prompting strategies to enhance specificness.

8 References

[Acampora et al., 2017] Acampora, G., Vitiello, A., Di Stefano, B., van der Aalst, W., Günther, C., and Verbeek, E. (2017). Ieee 1849tm: The xes standard. *IEEE Computational Intelligence Magazine*, pages 4–8.

[Adams et al., 2022] Adams, J. N., Schuster, D., Schmitz, S., Schuh, G., and van der Aalst, W. M. (2022). Defining cases and variants for object-centric event data. In *2022 4th International Conference on Process Mining (ICPM)*, pages 128–135. IEEE.

[Arora et al., 2022] Arora, S., Narayan, A., Chen, M. F., Orr, L., Guha, N., Bhatia, K., Chami, I., and Re, C. (2022). Ask me anything: A simple strategy for prompting language models. In *The Eleventh International Conference on Learning Representations*.

[Augusto et al., 2019] Augusto, A., Conforti, R., Dumas, M., La Rosa, M., and Polyvyanyy, A. (2019). Split miner: automated discovery of accurate and simple business process models from event logs. *Knowledge and Information Systems*, 59:251–284.

[Augusto et al., 2021] Augusto, A., Dumas, M., and La Rosa, M. (2021). Automated discovery of process models with true concurrency and inclusive choices. In *Process Mining Workshops: ICPM 2020 International Workshops, Padua, Italy, October 5–8, 2020, Revised Selected Papers 2*, pages 43–56. Springer.

[Berti and Qafari, 2023] Berti, A. and Qafari, M. S. (2023). Leveraging large language models (llms) for process mining (technical report). *arXiv preprint arXiv:2307.12701*.

[Berti et al., 2019] Berti, A., Van Zelst, S. J., and van der Aalst, W. (2019). Process mining for python (pm4py): bridging the gap between process-and data science. *arXiv preprint arXiv:1905.06169*.

[Chang et al., 2023] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., et al. (2023). A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

[Chapela-Campa and Dumas, 2023] Chapela-Campa, D. and Dumas, M. (2023). From process mining to augmented process execution. *Software and Systems Modeling*, pages 1–10.

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

[de Leoni and Mannhardt, 2015] de Leoni, M. and Mannhardt, F. (2015). Road Traffic Fine Management Process. https://doi.org/10.4121/uuid:270fd440-1057-4fb9-89a9-b699b47990f5. [Accessed 05-06-2024].

[Di Francescomarino et al., 2016] Di Francescomarino, C., Dumas, M., Maggi, F. M., and Teinemaa, I. (2016). Clustering-based predictive process monitoring. *IEEE transactions on services computing*, 12(6):896–909.

[Dijkman et al., 2011] Dijkman, R., Hofstetter, J., and Koehler, J. (2011). *Business Process Model and Notation*, volume 89. Springer.

[Dumas, 2011] Dumas, M. (2011). Consolidated management of business process variants. In *International Conference on Business Process Management*, pages 1–1. Springer.

[Gunther and Verbeek, 2014] Gunther, C. W. and Verbeek, H. (2014). Xes-standard definition.

[GuolinKe et al., 2017] GuolinKe, Q. M., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst*, 30:52.

[Krogh, 2008] Krogh, A. (2008). What are artificial neural networks? *Nature biotechnology*, 26(2):195–197.

[Kubrak et al., 2022] Kubrak, K., Milani, F., Nolte, A., and Dumas, M. (2022). Prescriptive process monitoring: Quo vadis? *PeerJ Computer Science*, 8:e1097.

[Logan IV et al., 2021] Logan IV, R. L., Balažević, I., Wallace, E., Petroni, F., Singh, S., and Riedel, S. (2021). Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*.

[Mehdiyev et al., 2020] Mehdiyev, N., Evermann, J., and Fettke, P. (2020). A novel business process prediction model using a deep learning method. *Business & information systems engineering*, 62:143–157.

[Natekin and Knoll, 2013] Natekin, A. and Knoll, A. (2013). Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21.

[Nguyen et al., 2018] Nguyen, H., Dumas, M., La Rosa, M., and ter Hofstede, A. H. (2018). Multi-perspective comparison of business process variants based on event logs. In *Conceptual Modeling: 37th International Conference, ER 2018, Xi'an, China, October 22–25, 2018, Proceedings 37*, pages 449–459. Springer.

[OpenAI, 2024] OpenAI (2024). https://platform.openai.com/docs/guides/text-generation/chat-completions-api. [Accessed 16-06-2024].

[Park et al., 2023] Park, G., Schuster, D., and van der Aalst, W. M. (2023). Pattern-based action engine: Generating process management actions using temporal patterns of process-centric problems. *Computers in Industry*, 153:104020.

[Park and van der Aalst, 2020] Park, G. and van der Aalst, W. M. (2020). A general framework for action-oriented process mining. In *Business Process Management Workshops: BPM 2020 International Workshops, Seville, Spain, September 13–18, 2020, Revised Selected Papers 18*, pages 206–218. Springer.

[Peterson, 1977] Peterson, J. L. (1977). Petri nets. *ACM Computing Surveys (CSUR)*, 9(3):223–252.

[Petri and Reisig, 2008] Petri, C. A. and Reisig, W. (2008). Petri net. *Scholarpedia*, 3(4):6477.

[Polyvyanyy et al., 2011] Polyvyanyy, A., Vanhatalo, J., and Völzer, H. (2011). Simplified computation and generalization of the refined process structure tree. In *Web Services and Formal Methods: 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers 7*, pages 25–41. Springer.

[Qualtir, 2024] Qualtir (2024). GPT Workspace — gpt.space. https://gpt.space/blog/how-to-use-openai-model-temperature-for-better-ai-chat-responses. [Accessed 13-06-2024].

[Renze and Guven, 2024] Renze, M. and Guven, E. (2024). The effect of sampling temperature on problem solving in large language models. *arXiv preprint arXiv:2402.05201.*

[Reynolds and McDonell, 2021] Reynolds, L. and McDonell, K. (2021). Prompt programming for large language models: Beyond the few-shot paradigm. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, New York, NY, USA. Association for Computing Machinery.

[Rinderle et al., 2006] Rinderle, S., Bobrik, R., Reichert, M., and Bauer, T. (2006). Businesss process visualization-use cases, challenges, solutions.

[Schapire, 2003] Schapire, R. E. (2003). The boosting approach to machine learning: An overview. *Nonlinear estimation and classification*, pages 149–171.

[Singh et al., 2023] Singh, S. K., Kumar, S., and Mehra, P. S. (2023). Chat gpt & google bard ai: A review. In *2023 International Conference on IoT, Communication and Automation Technology (ICICAT)*, pages 1–6. IEEE.

[Spearman, 1961] Spearman, C. (1961). The proof and measurement of association between two things.

[Tax et al., 2017] Tax, N., Verenich, I., La Rosa, M., and Dumas, M. (2017). 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 29*, pages 477–492. Springer.

[Taymouri et al., 2021] Taymouri, F., La Rosa, M., Dumas, M., and Maggi, F. M. (2021). Business process variant analysis: Survey and classification. *Knowledge-Based Systems*, 211:106557.

[Teinemaa et al., 2019] Teinemaa, I., Dumas, M., Rosa, M. L., and Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(2):1–57.

[Teinemaa et al., 2018] Teinemaa, I., Tax, N., de Leoni, M., Dumas, M., and Maggi, F. M. (2018). Alarm-based prescriptive process monitoring. In *Business Process Management Forum: BPM Forum 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings 16*, pages 91–107. Springer.

[van der Aalst, 2012] van der Aalst, W. (2012). Process mining: Overview and opportunities. *ACM Transactions on Management Information Systems (TMIS)*, 3(2):1–17.

[Van der Aalst, 1997] Van der Aalst, W. M. (1997). Verification of workflow nets. In *International conference on application and theory of petri nets*, pages 407–426. Springer.

[Van der Aalst, 1998] Van der Aalst, W. M. (1998). The application of petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66.

[van der Aalst, 2011] van der Aalst, W. M. (2011). *Process Mining: Discovery, Conformance and Enhancement of Business Processes.* Springer Berlin, Heidelberg.

[Van Der Aalst, 2019] Van Der Aalst, W. M. (2019). A practitioner's guide to process mining: Limitations of the directly-follows graph.

[Van der Aalst et al., 2011] Van der Aalst, W. M., Schonenberg, M. H., and Song, M. (2011). Time prediction based on process mining. *Information systems*, 36(2):450–475.

[van Dongen, 2017] van Dongen, B. (2017). BPI Challenge 2017. https://doi.org/10.4121/uuid:5f3067df-f10b-45da-b98b-86ae4c7a310b. [Accessed 05-06-2024].

[van Dongen et al., 2008] van Dongen, B. F., Crooy, R. A., and van der Aalst, W. M. (2008). Cycle time prediction: When will this case finally be finished? In *On the Move to Meaningful Internet Systems: OTM 2008: OTM 2008 Confederated International Conferences, CoopIS, DOA, GADA, IS, and ODBASE 2008, Monterrey, Mexico, November 9-14, 2008, Proceedings, Part I*, pages 319–336. Springer.

[Vaswani et al., 2017] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

[Von Rosing et al., 2015] Von Rosing, M., White, S., Cummins, F., and De Man, H. (2015). Business process model and notation-bpmn.

[Wieringa, 2014] Wieringa, R. J. (2014). *Design science methodology for information systems and software engineering*. Springer.

[Xie et al., 2024] Xie, J., Chen, A. S., Lee, Y., Mitchell, E., and Finn, C. (2024). Calibrating language models with adaptive temperature scaling. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

[Zhou et al., 2022] Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q., et al. (2022). Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

[Zhu et al., 2024] Zhu, Y., Li, J., Li, G., Zhao, Y., Jin, Z., and Mei, H. (2024). Hot or cold? adaptive temperature sampling for code generation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 437–445.

A Preprocessing Notebook

Importing dependencies and dataset

```
!pip install pm4py
import pm4py
import random
```

```
# Mount Drive
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
# Import XES logs
logs_dir = "/content/drive/MyDrive/BPI Challenge 2017.xes"
log = pm4py.read_xes(logs_dir)
```

```
# Export csv to save time on constantly importing and parsing the XES file.
log.to_csv("/content/drive/MyDrive/Thesis/bpi_challenge_2017.csv", index=False)
```

Finding the labels

Defining the values for the labels

```
pos_label_values = ["O_Refused", "O_Cancelled"] # Positive labels, the system should consider alarming
neg_label_values = ["O_Accepted"] # Negative labels, the desirable outcome

label_value_col = "concept:name" # The label in which the value for the label can be found

pos_label = 'deviant'
neg_label = 'regular'
unkwn_label = 'unknown'
```

```
# Reloading the data for debugging
import pandas as pd
log = pd.read_csv("/content/drive/MyDrive/Thesis/bpi_challenge_2017_short.csv", sep=';')
```

Applying the labels and truncating

```
# Create a list of labels corresponding to the dataset and add to the dataframe
labels = [] # List for storing the labels

# Get list of unique case id's
case_id_col = "case:concept:name"
case_ids = log[case_id_col].unique()

# For every unique case id, loop over the dataset and check the outcome
for case_id in case_ids:
  # Create a boolean mask representing a trace
  trace = log.loc[log[case_id_col] == case_id]

  # Check the last row in the trace for the pos or neg label value
  last_row = trace.iloc[-1]
  if last_row[label_value_col] in pos_label_values:
    # Add the label to the labels list.
    for event in range(len(trace.index)):
      # The number of labels added to the list is equal to the number of events in the trace
      labels.append(pos_label)


  elif last_row[label_value_col] in neg_label_values:
    for event in range(len(trace.index)):
      labels.append(neg_label)

  else:
    for event in range(len(trace.index)):
      labels.append(unkwn_label)

  # Now remove random number of events from the traces based on lenght
  # First determine trace length
```

```
    # First determine trace length
    trace_length = len(trace.index)
    # Now determine how much to remove
    if trace_length > 10:
      random_number = random.randint(5, 9)
    elif trace_length > 6:
      random_number = random.randint(5, trace_length-1)
    else:
      random_number = trace_length
    # Now drop
    # Find indexes of rows that need to be dropped
    rows_to_drop = trace.iloc[-random_number:]
    index_list = rows_to_drop.index.tolist()
    log.drop(index=index_list, inplace=True)
    # Now drop them from the labels list
    labels = labels[:len(labels)-random_number]

  # Now add the labels list as a labels column
  log['label'] = labels

  # Export with unknown labels
  log.to_csv("BPI_2017_labeled_w_unknown.csv", index=False)
```

Remove unknown rows

```
log.drop(log[log.label == unkwn_label].index, inplace=True)
```

Export

```
log.to_csv("BPI_2017_labeled.csv", index=False)
```

B Subset of Preprocessed Dataset

Due to formatting constraints the following columns are omitted from this preview: Action, org:resource, EventOrigin, lifecycle:transition, case:ApplicationType, FirstWithdrawalAmount, NumberOfTerms, Accepted, MonthlyCost, Selected, Creditscore, and OfferedAmount. Also note that the case:concept:name column was not used during training. This column indicates what trace a data sample belongs to.

concept:name	time:timestamp	case:LoanGoal	case:concept:name	case:RequestedAmount	OfferedAmount	label
A_Create Application	2016-01-01 09:51:	Existing loan takeover	Application_652823628	20000.0		regular
A_Submitted	2016-01-01 09:51:	Existing loan takeover	Application_652823628	20000.0		regular
W_Handle leads	2016-01-01 09:51:	Existing loan takeover	Application_652823628	20000.0		regular
W_Handle leads	2016-01-01 09:52:	Existing loan takeover	Application_652823628	20000.0		regular
W_Complete application	2016-01-01 09:52:	Existing loan takeover	Application_652823628	20000.0		regular
A_Concept	2016-01-01 09:52:	Existing loan takeover	Application_652823628	20000.0		regular
W_Complete application	2016-01-02 10:45:	Existing loan takeover	Application_652823628	20000.0		regular
W_Complete application	2016-01-02 10:49:	Existing loan takeover	Application_652823628	20000.0		regular
A_Accepted	2016-01-02 11:23:	Existing loan takeover	Application_652823628	20000.0		regular
O_Create Offer	2016-01-02 11:29:	Existing loan takeover	Application_652823628	20000.0	20000.0	regular
O_Created	2016-01-02 11:29:	Existing loan takeover	Application_652823628	20000.0		regular
O_Sent (mail and online)	2016-01-02 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
W_Complete application	2016-01-02 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-02 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-02 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
A_Complete	2016-01-02 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-02 11:32:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-06 09:26:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-06 09:27:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call after offers	2016-01-13 13:10:	Existing loan takeover	Application_652823628	20000.0		regular
W_Validate application	2016-01-13 13:10:	Existing loan takeover	Application_652823628	20000.0		regular
W_Validate application	2016-01-13 13:10:	Existing loan takeover	Application_652823628	20000.0		regular
A_Validating	2016-01-13 13:10:	Existing loan takeover	Application_652823628	20000.0		regular
O_Returned	2016-01-13 13:11:	Existing loan takeover	Application_652823628	20000.0		regular
W_Validate application	2016-01-13 13:15:	Existing loan takeover	Application_652823628	20000.0		regular
W_Validate application	2016-01-14 09:16:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call incomplete files	2016-01-14 09:16:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call incomplete files	2016-01-14 09:16:	Existing loan takeover	Application_652823628	20000.0		regular
A_Incomplete	2016-01-14 09:16:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call incomplete files	2016-01-14 09:17:	Existing loan takeover	Application_652823628	20000.0		regular

W_Call incomplete files	2016-01-14 11:27:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call incomplete files	2016-01-14 11:30:	Existing loan takeover	Application_652823628	20000.0		regular
W_Call incomplete files	2016-01-14 13:39:	Existing loan takeover	Application_652823628	20000.0		regular
W_Validate application	2016-01-14 13:39:	Existing loan takeover	Application_652823628	20000.0		regular
A_Create Application	2016-01-01 10:16:	Home improvement	Application_1691306052	10000.0		deviant
A_Submitted	2016-01-01 10:16:	Home improvement	Application_1691306052	10000.0		deviant
W_Handle leads	2016-01-01 10:16:	Home improvement	Application_1691306052	10000.0		deviant
W_Handle leads	2016-01-01 10:17:	Home improvement	Application_1691306052	10000.0		deviant
W_Complete application	2016-01-01 10:17:	Home improvement	Application_1691306052	10000.0		deviant
A_Concept	2016-01-01 10:17:	Home improvement	Application_1691306052	10000.0		deviant
W_Complete application	2016-01-02 10:50:	Home improvement	Application_1691306052	10000.0		deviant
W_Complete application	2016-01-02 10:51:	Home improvement	Application_1691306052	10000.0		deviant
W_Complete application	2016-01-02 10:53:	Home improvement	Application_1691306052	10000.0		deviant
A_Accepted	2016-01-02 10:59:	Home improvement	Application_1691306052	10000.0		deviant
O_Create Offer	2016-01-02 11:02:	Home improvement	Application_1691306052	10000.0	6000.0	deviant
O_Created	2016-01-02 11:02:	Home improvement	Application_1691306052	10000.0		deviant
O_Sent (mail and online)	2016-01-02 11:03:	Home improvement	Application_1691306052	10000.0		deviant
W_Complete application	2016-01-02 11:03:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-02 11:03:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-02 11:03:	Home improvement	Application_1691306052	10000.0		deviant
A_Complete	2016-01-02 11:03:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-02 11:09:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-06 09:03:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-06 09:04:	Home improvement	Application_1691306052	10000.0		deviant
W_Call after offers	2016-01-07 09:40:	Home improvement	Application_1691306052	10000.0		deviant
W_Validate application	2016-01-07 09:40:	Home improvement	Application_1691306052	10000.0		deviant
A_Create Application	2016-01-01 11:19:	Home improvement	Application_428409768	15000.0		deviant
A_Submitted	2016-01-01 11:19:	Home improvement	Application_428409768	15000.0		deviant
W_Handle leads	2016-01-01 11:19:	Home improvement	Application_428409768	15000.0		deviant
W_Handle leads	2016-01-01 11:20:	Home improvement	Application_428409768	15000.0		deviant
W_Complete application	2016-01-01 11:20:	Home improvement	Application_428409768	15000.0		deviant

C Evaluation Survey

Prescriptive Process Mining & Large Language Models Survey

This survey will take ~15-20 minutes to complete.

Goal
The aim is to investigate the viability of large language models (LLMs) in the context of process mining. More specifically, if LLMs can be used as an assistant to advise an operator regarding the next best step.

Process Mining
In process mining, business processes are mapped and analyzed using traces that are left behind in information systems, such as ERP or CRM systems. With the goal of optimizing processes while keeping the number of undesirable outcomes to a minimum. In my research, I combine 'traditional' machine learning, with LLMs. In order to answer the question: 'How do we prescribe the right action after finding the timing for an intervention?'

AI and Large language models
I have developed a system that combines the PM4Py package for process mining with a LLM. After the system detects an opportunity for an intervention, context information is collected and a prompt is formed. This prompt is then presented to an instance of Chat-GPT 3.5 turbo.

- A random selection of responses will be presented in this survey.
- The goal of this survey is to evaluate the recommendation given by the LLM.
- You will thus provide your feedback on whether or not the LLM recommendation is correct.

Metrics
You will be asked to rate the recommendations made by the large language model on different metrics:

- *Actionability in day-to-day operations:* Can a worker implement this recommendation in a day-to-day situation?
- *Usability in tactical operations:* Is the recommendation useful for improving the process from a process design view?
- *Relevance:* Is the recommendation applicable to the presented situation?
- *Expected improvement to process execution:* Does the recommendation improve the performance of the process?
- *Specificness:* Does the recommendation specifically describe what has to be done?
- *Proportionality:* Are the required resources associated with executing the recommendation proportional to the presented situation?

Participation is voluntary

Participation to the survey is completely voluntary and you are free to stop participating whenever you want. Responses are completely anonymous. The results of the survey will be presented in the final paper in an aggregated manner. The unaggregated data, will be deleted within 90 days of completing the survey.

Understanding the process

To be able to properly rate the outcomes of the LLM later, it is first important to get a rough understanding of the process in question. This process model describes a loan application process from within a financial institution. The process starts with an application for a loan by a customer. There are three possible outcomes:

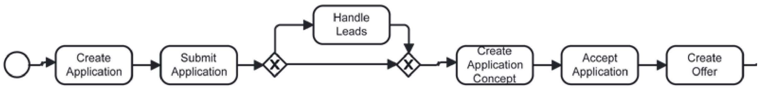
- The application either gets: accepted by both parties,
- refused by the financial institution,
- or cancelled by the customer.

For the purposes of my research I have identified the first outcome (accepted) as desirable, and the latter two (refused and cancelled) as undesirable.

Timestamps are left out. So, if the system identified a problem related to time (delays etc.), you can assume the system identified this correctly.

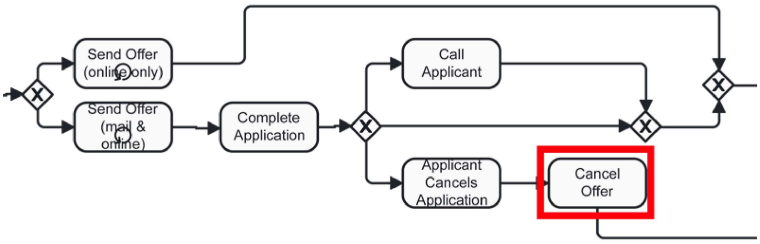
The start of the process:

Each process starts with the creation of a loan application. This application gets submitted, and there is the possibility that one of the employee of the financial institution contacts the customer. After this a concept application is created within the institution. When this concept is finalized, the application can be accepted and an offer to the customer can be created.



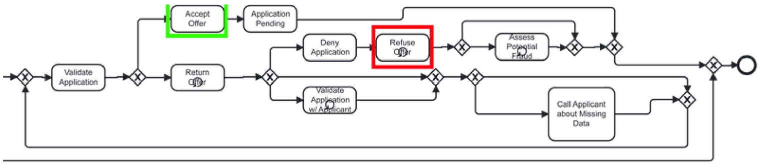
Sending the offer to the customer:

There are two methods when it comes to sending the offer to the customer: online only, and mail & online. When the customer receives the offer though mail and online, they have to option to cancel the offer, which is one of the undesirable outcomes.



Validating the offer:

The final part of the process encompasses the validation and finalization of the application process. After the finalized application is validated there are two options: the institution accepts the offer and the loan is granted (desirable) or the offer is returned. After the offer is returned the institution either completely denies the application (undesirable) or attempts to update the application so it can be accepted later. For this the customer is contacted to obtain any missing data. With the missing data obtained, the institution decides again to either refuse or accept the application.



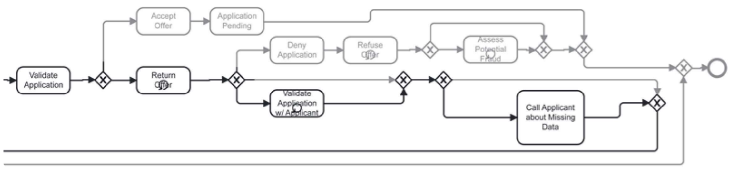
Responses

Now the LLM recommendations and their corresponding event traces will be presented. Each recommendation includes a short context alongside the relevant part of the process model. Based on the different metrics you will evaluate the response.

1

We pick up at the validation step of the process, where the offer is returned because information is missing. A worker calls the customer to retrieve the missing data. After this the offer is revalidated. Again, data is missing. So, it seems like the first attempt to validate the customer and application data was unsuccessful.

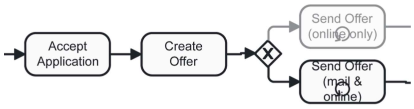
Recommendation:
It seems that there are several instances of incomplete applications in this process. To improve the outcome, you can add an automated email reminder to be sent to the customer when their application is incomplete, asking them to complete the application as soon as possible. Additionally, you can consider adding an automated follow-up email to be sent to the customer after a certain period of time if they have not completed their application yet. This will help ensure that the application is completed in a timely manner and reduce the likelihood of negative outcomes.



	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

2

After accepting the initial loan application, an offer was created and sent via mail and online. The system then detected it took too long until a next action was started.



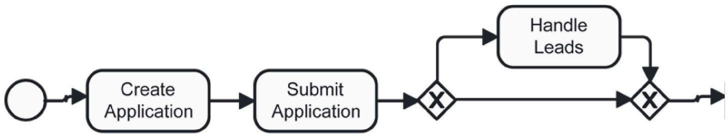
Recommendation:
It seems that the application review process is taking too long and causing delays. One actionable recommendation is to set a maximum time limit for each step of the process to ensure that each step is completed within a reasonable time frame. Another recommendation is to have an automated system that sends reminders to the responsible parties when a task is pending for too long.

	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	○	○	○	○	○
Usability in tactical operations	○	○	○	○	○
Relevance	○	○	○	○	○
Expected improvement to process execution	○	○	○	○	○
Specificness	○	○	○	○	○
Proportionality	○	○	○	○	○

3

A customer applied for a loan, an application was created and submitted. After this a worker called the customer. The system then detected it took too long until a next action was started.

Recommendation:
Assign a deadline for the lead handler to follow up with the customer. This will ensure that the lead handler does not forget to follow up with the customer and the application does not get stuck in the "W_Handle leads" state for too long.

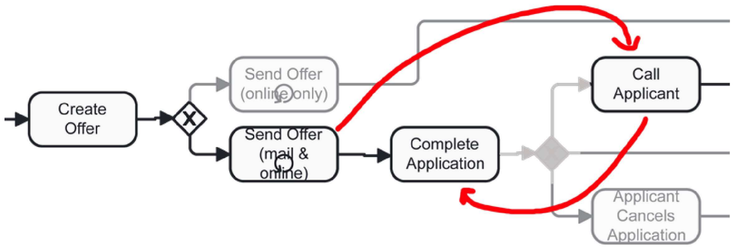


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

4

An offer was created and sent via mail and online. After this the trace deviated from the known process model. First a worker calls the customer. Then the application is completed. The system detected the deviation and gives the following recommendation.

Recommendation:
Send a confirmation email to the customer once the offer is created to make sure they received it and are aware of the next steps in the process. This will ensure that they remain engaged and reduce the likelihood of the application being forgotten or overlooked.

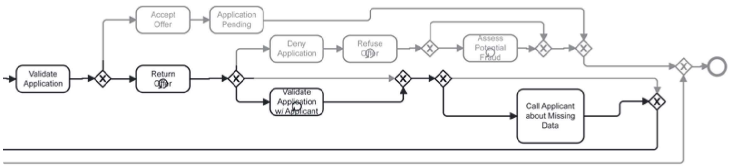


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

5

A worker calls the customer to retrieve the missing data. After this the offer is revalidated. Again, data is missing. So, it seems like the first attempt to validate the customer and application data was unsuccessful.

Recommendation:
It seems that there are several steps in the process where the application is delayed or incomplete. One recommendation would be to implement a system to track the progress of the application and send reminders to the customer if there are any delays or missing information. This could help ensure that the application is completed in a timely manner and reduce the likelihood of negative outcomes. Additionally, it may be helpful to review the process to identify any bottlenecks or areas for improvement.

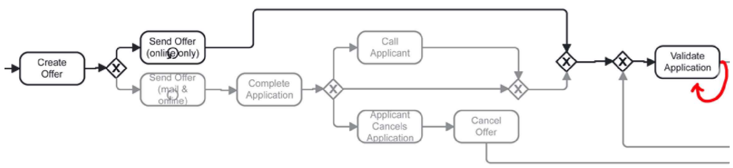


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

6

An offer is sent to the customer via online only. After this the action 'Validate Application' is executed multiple times. The system detects this anomaly and gives the following recommendation.

Recommendation:
One recommendation is to implement automated alerts for process stakeholders when a task has not been completed within a reasonable time frame. This can help to prevent delays and ensure that tasks are completed in a timely manner. Additionally, it may be helpful to review the process and identify any unnecessary steps that can be eliminated or streamlined to improve efficiency.

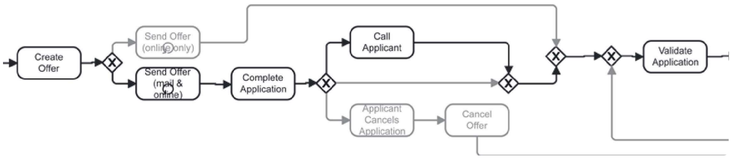


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

7

An offer is sent to the customer via mail and online. The application is completed and the customer is called. After this the application is validated. However, the system detected that this validation process is taking a long time.

Recommendation:
It seems that there is a delay in completing the application and handling leads. The recommendation is to optimize the lead handling process to reduce the time it takes to complete the application. Additionally, it may be helpful to send reminders to the customer to complete the application in a timely manner.

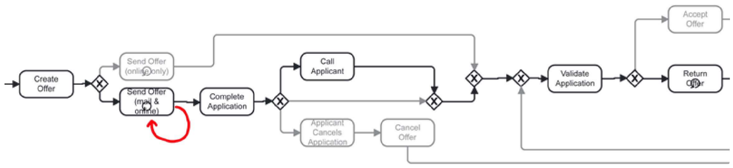


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	○	○	○	○	○
Usability in tactical operations	○	○	○	○	○
Relevance	○	○	○	○	○
Expected improvement to process execution	○	○	○	○	○
Specificness	○	○	○	○	○
Proportionality	○	○	○	○	○

8

An offer is sent to the customer via mail and online. After this the offer is sent again, which is unusual. Since normally the offer is sent only once. The application is completed, the customer is called and the application enters the validation stage, where the offer is returned.

Recommendation:
It seems like there are too many loops in the process, leading to a delay in the final outcome. One recommendation would be to streamline the process and reduce the number of loops. Another recommendation would be to implement automated reminders for incomplete applications or offers, so that the process can move forward more efficiently.

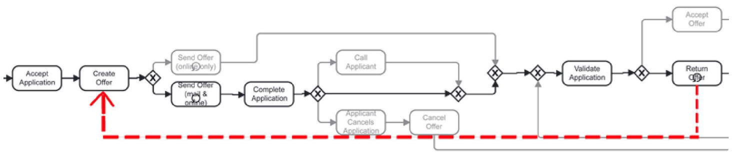


	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	○	○	○	○	○
Usability in tactical operations	○	○	○	○	○
Relevance	○	○	○	○	○
Expected improvement to process execution	○	○	○	○	○
Specificness	○	○	○	○	○
Proportionality	○	○	○	○	○

9

An offer is made, which is sent to the customer via mail and online. The application is completed and enters the validation stage, where it is returned. After this the process execution deviates from the known process model, and a new offer is created. The system detects this anomaly and gives the following recommendation.

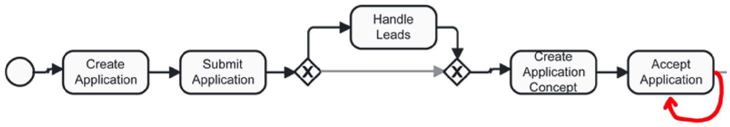
Recommendation:
It seems that there is a problem with the offer creation process, as it is being cancelled and recreated multiple times. One possible recommendation is to review the criteria used to create the offers and ensure that they are accurate and complete. Additionally, it may be helpful to involve a supervisor or manager in the offer creation process to provide oversight and ensure that the process is running smoothly. Finally, it may be helpful to streamline the process by automating some of the steps, such as sending reminders to incomplete files or validating applications.



	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	○	○	○	○	○
Usability in tactical operations	○	○	○	○	○
Relevance	○	○	○	○	○
Expected improvement to process execution	○	○	○	○	○
Specificness	○	○	○	○	○
Proportionality	○	○	○	○	○

A customer applied for a loan. An application was created and submitted. The customer was contacted and a application concept was created. After this the action 'Accept Application' is executed multiple times, which is unusual. Since, normally it is only executed once. The system detects this and gives the following recommendation.

Recommendation:
Automate the completion of the application process to reduce the likelihood of errors and increase efficiency. Additionally, have a review process in place to catch any errors or inconsistencies before the application is submitted.



	Very Low	Low	Neutral	High	Very High
Actionability in day-to-day operations	◯	◯	◯	◯	◯
Usability in tactical operations	◯	◯	◯	◯	◯
Relevance	◯	◯	◯	◯	◯
Expected improvement to process execution	◯	◯	◯	◯	◯
Specificness	◯	◯	◯	◯	◯
Proportionality	◯	◯	◯	◯	◯

This is the end of the survey.

Thank you very much for participating in this survey! Your answers will help me out a lot with completing my master thesis. I'm hoping to share the results of my work to you during a presentation at Avanade.