



universität
ulm

Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie
Institut für Datenbanken
und Informationssyste-
me (DBIS)

Identifikation von DSGVO-kritischen Aktivitäten in Business Prozessen mittels Large Language Models

Abschlussarbeit an der Universität Ulm

Vorgelegt von:

Merten Dieckmann
merten.dieckmann@uni-ulm.de
1058340

Gutachter:

Prof. Dr. Manfred Reichert
Prof. Dr. Rüdiger Pryss

Betreuer:

Magdalena von Schwerin

2025

Fassung 26. Oktober 2025

© 2025 Merten Dieckmann

Satz: PDF-L^AT_EX 2_ε

Zusammenfassung

In der EU ist die Einhaltung der Datenschutz-Grundverordnung (DSGVO) in Geschäftsprozessen essenziell, da bei Nichteinhaltung hohe Bußgelder drohen können. Jedoch ist eine manuelle Prüfung von Prozessmodellen in der Praxis aufwendig und fehleranfällig. Diese Arbeit untersucht daher, wie Large Language Models (LLMs) DSGVO-kritische Aktivitäten in BPMN-Prozessmodellen automatisiert identifizieren können, da sie großes Potenzial im Umgang mit komplexen Texten und Zusammenhängen zeigen. Hierzu wird (1) eine Klassifizierungspipeline mit Zero-Shot-Prompting und strikt strukturiertem JSON-Output, ergänzt um *id*-Validierung-/Vervollständigung und einem automatischen Retry-Mechanismus, (2) ein Labeling-Tool für gelabelte BPMN-Testfälle sowie (3) ein Evaluationsframework mit deklarativer Konfiguration, standardisierter HTTP-Schnittstelle und Frontend für reproduzierbare, vergleichbare Experimente entwickelt. Diese Infrastruktur ermöglicht belastbare Modellvergleiche über mehrere Domänen und Sprachen hinweg.

In einer empirischen Studie werden 13 LLMs (Europäische vs. Internationale, Offene vs. Proprietäre, Große vs. Kleine) auf 25 Testfälle über fünf Wiederholungen evaluiert. Daraufhin werden die Mittelwerte und Standardabweichungen für Precision, Recall, F1-Score und Accuracy berechnet. Neun von dreizehn Modellen erreichen einen F1-Score von $\geq 0,80$. Spitzenreiter sind Qwen3-235B-A22B-Thinking-2507 (F1-Score = 0,874, Recall = 0,932), GPT-0SS-20B (F1-Score = 0,866, Recall = 0,918) und DeepSeek-R1-Distill-Qwen-14B (F1-Score = 0,848, Precision = 0,829). Unter den EU-Modellen stechen Mistral Medium 3.1 und Mistral-Large-Instruct-2411 mit F1-Scores von 0,843 bzw. 0,823 heraus.

Die Ergebnisse zeigen zudem unterschiedliche Trade-offs: GPT-4o erzielt die höchste Precision (0,892), verfehlt jedoch mit einem Recall von 0,762 die Mindestanforderung für ein Recall-orientiertes Screening. Gemma-3-27B-it erreicht umgekehrt einen sehr hohen Recall (0,916) bei niedriger Precision (0,687). Insgesamt ist die Varianz über Seeds für die meisten Modelle gering ($SD_{F1\text{-Score}} \leq 0,02$).

Fehleranalysen zeigen vor allem False Positives bei fehlendem Kontext (z. B. nicht explizit modellierte Anonymisierung von Daten) und False Negatives, wenn personenbezogene Datenflüsse über mehrere Schritte nicht sicher erkannt werden.

Insgesamt eignen sich aktuelle LLMs gut für ein automatisiertes, Recall-orientiertes Vorscreening von BPMN-Prozessen, jedoch bleibt eine nachgelagerte fachliche Prüfung erforderlich.

Inhaltsverzeichnis

Abkürzungen	vi
1 Einleitung	1
1.1 Problemstellung	2
1.2 Zielsetzung und Beiträge	3
1.3 Aufbau der Arbeit	4
2 Hintergrund und verwandte Arbeiten	5
2.1 Datenschutzgrundverordnung (DSGVO)	5
2.2 Business Process Model and Notation (BPMN)	6
2.3 Large Language Models (LLMs)	9
2.4 Verwandte Arbeiten	11
3 Problemdefinition und Zielkriterien	16
3.1 Aufgabenstellung	17
3.2 Qualitätsziele	18
3.3 Scope und Annahmen	21
3.4 Experimentdesign	22
4 Design und Implementierung der Klassifizierungspipeline	24
4.1 BPMN Preprocessing	25
4.2 Prompt Engineering	26
4.3 Validierung der Ausgabe	30
4.4 API-Design	32
4.5 Webapp-Sandbox	35
5 Labeling und Datensätze	38
5.1 Labeling-Tool	38

Inhaltsverzeichnis

5.2 Quellen und Eigenschaften der Datensätze	40
5.3 Labeling-Guide	41
6 Evaluationsframework	44
6.1 Use-Cases und Anforderungen	44
6.2 Konfiguration einer Evaluierung	47
6.3 Architektur und Komponenten	49
6.4 Evaluationsergebnisse	50
6.5 Frontend	51
7 Modellauswahl	58
7.1 Kriterien	58
7.2 Modellvorstellung	60
8 Versuchsaufbau und Durchführung	65
8.1 Vergleichbarkeit	66
8.2 Konfigurationen	66
8.3 Durchführung	68
9 Ergebnisse	70
9.1 Zusammenfassung der Ergebnisse	70
9.2 Analyse nach Modellkategorien	74
9.3 Robustheit	77
9.4 Fallstudien	79
9.5 Beantwortung der Forschungsfragen	82
10 Diskussion	85
10.1 Einordnung und Interpretation	85
10.2 Modelle im Vergleich	86
10.3 Robustheit	87
10.4 Fehlerbilder und Grenzen	88
11 Fazit	90
Literatur	92
Quelltexte	101

Abkürzungen

BPM	Business Process Management
BPMN	Business Process Model and Notation
DSGVO	Datenschutz-Grundverordnung
EU	Europäische Union
FN	False Negative
FP	False Positive
KI	Künstliche Intelligenz
LLM	Large Language Model
LM	Language Model
MoE	Mixture-of-Experts
OSI	Open Source Initiative
RAG	Retrieval Augmented Generation
TN	True Negative
TP	True Positive

1 Einleitung

Geschäftsprozesse sind in nahezu allen Organisationen allgegenwärtig und bilden die Grundlage für effiziente Abläufe. Zugleich ist in Europa durch die Datenschutz-Grundverordnung (DSGVO) der Datenschutz zu einem zentralen regulatorischen Aspekt geworden [14, 18]. Unternehmen müssen sicherstellen, dass in ihren Prozessen personenbezogene Daten rechtskonform verarbeitet werden; andernfalls drohen Strafen von bis zu 20 Millionen Euro oder 4% des gesamten weltweit erzielten Jahresumsatzes [18].

Die Überprüfung von Prozessen auf Konformität in Bezug auf Datenschutz ist jedoch zeit- und kostenintensiv [55, 83]. Besonders in großen Organisationen mit hunderten parallel laufenden Prozessen ist eine manuelle Analyse kaum praktikabel und zudem fehleranfällig. Fehlerhafte Untererkennungen datenschutzkritischer Aktivitäten, sogenannte False Negatives (FN), können weitreichende Folgen haben - von Reputationsschäden bis hin zu hohen Bußgeldern [55].

Vor diesem Hintergrund rücken Large Language Models (LLMs) als aufstrebende Technologie im Bereich Künstliche Intelligenz (KI) in den Fokus. Sie sind darauf trainiert, natürliche Sprache auch in langen und komplexen Texten zu verstehen, Zusammenhänge über große Kontexte hinweg zu erkennen und Anweisungen zu befolgen. Damit erscheinen LLMs als vielversprechender Ansatz für das automatisierte Screening von Prozessmodellen. Erste Arbeiten belegen dieses Potenzial, etwa bei der Identifikation datenschutzrelevanter Verarbeitungstätigkeiten oder in der Analyse von Datenschutzerklärungen [15, 68].

Besonders interessant sind in diesem Kontext europäische Open-Source-Modelle wie die von Mistral [4]. Sie sind zum einen frei verfügbar und transparent, zum anderen wurden sie bislang kaum im Hinblick auf DSGVO-bezogene Aufgaben evaluiert. Es fehlen belastbare, reproduzierbare empirische Vergleiche, die eine fundierte Bewertung dieser Modelle erlauben würden [79].

1.1 Problemstellung

Trotz der genannten Potenziale fehlt es bisher an standardisierten, reproduzierbaren Vergleichen verschiedener Modelle für die konkrete Aufgabe, Aktivitäten in Geschäftsprozessen nach „kritisch“ und „unkritisch“ zu klassifizieren. Erste Ansätze, wie z. B. der von Nake et al. [55], zeigen, dass ML-Ansätze grundsätzlich in der Lage sind DSGVO-kritische Aktivitäten in textuellen Prozessbeschreibungen zu erkennen; dennoch existieren keine einheitlichen Benchmarks, die einen systematischen Vergleich unterschiedlicher LLMs erlauben.

Auch von Schwerin et al. [79] heben hervor, dass trotz großer Fortschritte im Einsatz von LLMs für juristische Aufgaben bislang erhebliche Lücken in der Evaluation für Compliance-spezifische Anwendungen bestehen und geeignete DSGVO-spezifische Benchmarks fehlen. Somit mangelt es derzeit an einer belastbaren empirischen Grundlage, um Modelle zuverlässig und vergleichbar zu bewerten.

Besonders interessant ist die Frage, wie sich Open-Source-Modelle - insbesondere mit Ursprung aus der Europäische Union (EU) - im Vergleich zu internationalem außerhalb der EU entwickelten Modellen schlagen und welche Trade-offs dabei entstehen [79]. Diese Perspektive ist nicht nur aus Leistungs-, sondern auch aus Transparenz- und Regulierungsgründen relevant.

Eine zusätzliche Herausforderung ergibt sich aus der Natur von Business Process Model and Notation (BPMN)-Modellen: Typischerweise konzentrieren sie sich auf den Kontrollfluss und vernachlässigen die Datenebene. Datenobjekte werden oftmals gar nicht explizit modelliert oder nur implizit in den Aktivitäten referenziert. Dadurch ist die Datennutzung von Aktivitäten nicht direkt erkennbar und muss aus textuellen Beschreibungen und dem Kontext erschlossen werden [77]. Das erschwert die automatische Identifikation von DSGVO-kritischen Aktivitäten, da Algorithmen personenbezogene Datenflüsse zunächst indirekt und über den Kontext ableiten müssen.

1.2 Zielsetzung und Beiträge

Ziel der Arbeit ist es, einen methodischen Beitrag zur automatisierten Identifikation von DSGVO-kritischen Aktivitäten in Geschäftsprozessen zu leisten. Hierfür werden folgende Beiträge angestrebt:

- Entwicklung einer Klassifizierungspipeline für Geschäftsprozesse, die Aktivitäten binär in datenschutzkritisch oder unkritisch einordnet.
- Konzeption und Umsetzung eines Evaluationsframeworks, das reproduzierbare Vergleiche verschiedener LLMs und Algorithmen über eine einheitliche Schnittstelle ermöglicht.
- Entwicklung einer Labeling-Software zur Erstellung und Annotation von Datensätzen für das Evaluationsframework.
- Aufbau eines repräsentativen Datensatzes aus gelabelten BPMN-Prozessen, inklusive klar definierter Labeling-Kriterien.
- Bereitstellung überprüfbarer empirischer Befunde, inklusive Code, Konfigurationen der Experimente und Seeds, um Nachvollziehbarkeit und Reproduzierbarkeit zu gewährleisten.

Auf dieser Grundlage ergibt sich die zentrale Forschungsfrage dieser Arbeit:

FF1 Wie zuverlässig identifizieren LLMs DSGVO-kritische Aktivitäten in BPMN-Prozessmodellen?

Um diese Frage differenziert beantworten zu können, werden außerdem folgende Unterfragen betrachtet:

UF1 Wie gut schneiden europäische Modelle im Vergleich zu internationalen Modellen ab?

UF2 Wie unterscheiden sich große und kleine Modelle in ihrer Leistungsfähigkeit?

UF3 Welche Open-Source-Modelle (insbesondere aus der EU) erzielen die besten Ergebnisse?

UF4 Wie gut schneiden Open-Source-Modelle gegenüber kommerziellen Modellen wie GPT-4o ab?

Für ein initiales Screening reicht, wie in [55], eine binäre Klassifikation (kritisch vs. unkritisch). Eine tiefergehende rechtliche Prüfung kann in einem nachfolgenden Schritt durchgeführt werden und ist nicht Bestandteil dieser Arbeit.

1.3 Aufbau der Arbeit

Die Arbeit ist wie folgt gegliedert: Kapitel 2 gibt einen Überblick über den theoretischen Hintergrund, die DSGVO und BPMN sowie eine Einführung in LLMs und verwandte Arbeiten. Kapitel 3 beschreibt den Rahmen der Entwicklung der Klassierungspipeline, des Evaluationsframeworks und der Experimente. Kapitel 4 stellt den entwickelten Algorithmus zur Klassifikation von BPMN-Modellen und dessen einheitliche Schnittstelle vor. Kapitel 5 präsentiert die Labeling-Software und erläutert die Erstellung der Datensätze. Anschließend werden in Kapitel 6 die Architektur und der Funktionsumfang der Evaluationspipeline beschrieben. Kapitel 7 zeigt, wie die Auswahl der LLMs erfolgte. Kapitel 8 erläutert den Versuchsaufbau und die Durchführung der Experimente, Kapitel 9 stellt die Ergebnisse vor und Kapitel 10 diskutiert diese im Kontext der Forschungsfragen. Zum Schluss fasst Kapitel 11 die Arbeit zusammen und gibt einen Ausblick auf mögliche zukünftige Forschungsthemen.

2 Hintergrund und verwandte Arbeiten

2.1 Datenschutzgrundverordnung (DSGVO)

Die europäische Datenschutz-Grundverordnung (DSGVO) [18] bildet den zentralen rechtlichen Rahmen für den Schutz personenbezogener Daten in der EU. Sie gilt seit dem 25. Mai 2018. Durch die DSGVO werden Betroffenenrechte gestärkt und Verantwortliche zu technischen und organisatorischen Maßnahmen verpflichtet, wie z. B. *Datenschutz durch Technikgestaltung* und *datenschutzfreundliche Voreinstellungen* (Art. 25 DSGVO) [19].

Definitionen

Im Folgenden werden zentrale Begriffe der DSGVO erläutert, die für das Verständnis dieser Arbeit relevant sind:

- **Personenbezogene Daten** sind alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person beziehen (Art. 4 Abs. 1 DSGVO) [18]. Eine Person ist identifizierbar, wenn sie direkt oder indirekt bestimmbar ist (z. B. anhand des Namens, einer Kennnummer, von Standortdaten, einer Online-Kennung).
- **Verarbeitung** bezeichnet jeden mit personenbezogenen Daten vorgenommenen Vorgang (Art. 4 Abs. 2 DSGVO). Sie umfasst insbesondere das **Erheben, Speichern, Verwenden/Nutzen, Offenlegen durch Übermittlung** sowie das **Löschen/Vernichten** [18].

- Im BPMN-Kontext sind alle Aktivitäten als **datenschutzkritisch** zu betrachten, die solche Verarbeitungshandlungen an personenbezogenen Daten vornehmen oder auslösen (z. B. Abruf aus einer Kundendatenbank, Übergabe an externe Stellen).

Abgrenzung: Risiko-Screening vs. Rechtsberatung

Die in dieser Arbeit eingesetzten Klassifizierungsverfahren dienen einem *automatisierten Risiko-Vorscreening* von Prozessaktivitäten. Sie ersetzen keine individuelle Rechtsprüfung im Einzelfall. Insbesondere in Deutschland ist die Erbringung konkreter Rechtsdienstleistungen Personen mit entsprechender Befugnis vorbehalten [11]. Die Ergebnisse sind daher als Entscheidungshilfe zu verstehen und bedürfen - insbesondere bei Grenzfällen - der Bewertung durch qualifizierte Experten.

2.2 Business Process Model and Notation (BPMN)

BPMN ist ein Standard zur Modellierung von Geschäftsprozessen. Die Notation wurde entwickelt, um eine einheitliche Notation bereitzustellen, die sowohl von Geschäftsanalysten als auch von technischen Entwicklern verstanden wird. BPMN-Modelle bestehen aus verschiedenen Elementen wie Aktivitäten, Ereignissen, Gateways und Verbindungen, die zusammen den Ablauf eines Geschäftsprozesses darstellen [58].

Relevante BPMN-Elemente

Für die Identifikation von DSGVO-kritischen Aktivitäten sind insbesondere folgende Elemente relevant, da sie Hinweise auf den Umgang mit (personenbezogenen) Daten geben. Sie sind ebenfalls in Abbildung 2.1 dargestellt:

- **Aktivitäten** bilden die auszuführenden Arbeitsschritte eines Prozesses ab. Sie können Ein- und Ausgaben sowie Datenabhängigkeiten definieren [58]. Durch ihren Namen oder Kontext können Rückschlüsse auf die Verarbeitung personenbezogener Daten gezogen werden.

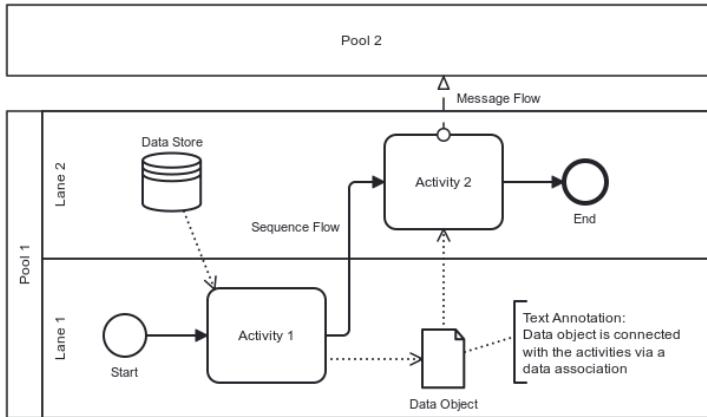


Abbildung 2.1: Die relevanten BPMN-Elemente in Beziehungen zueinander.

- **Sequenzflüsse** verbinden Aktivitäten, Ereignisse und Gateways und zeigen die Reihenfolge der Ausführung im Prozess an [58]. Mit ihrer Hilfe kann eine einzelne Aktivität im Kontext des gesamten Prozesses betrachtet werden, indem der Pfad zu und von der Aktivität verfolgt wird.
- **Datenobjekte und Datenspeicher** repräsentieren flüchtige oder persistente Daten, die im Prozess von z. B. Aktivitäten genutzt oder geschrieben werden können [58]. Sie können auch personenbezogene Daten enthalten.
- **Datenassoziationen** (Eingangs- und Ausgangsassoziationen) verbinden Aktivitäten mit Datenobjekten und Datenspeichern und zeigen so Ein- und Ausgaben explizit an [58]. Sie sind ein wichtiges Signal für die Verarbeitung personenbezogener Daten, da sie den direkten Bezug einer Aktivität zu bestimmten Daten verdeutlichen, z. B. Lesezugriff auf eine Kundendatenbank.
- **Pools** modellieren Organisationseinheiten oder Prozessbeteiligte, während **Lanes** Verantwortlichkeiten innerhalb eines Pools darstellen. Innerhalb eines Pools befinden sich die Aktivitäten und anderen Elemente des Prozesses [58]. Die Rollen und Verantwortlichkeiten, die durch Pools und Lanes dargestellt werden, können für die Bewertung der Datenverarbeitung relevant sein.
- **Nachrichtenflüsse** stellen den Austausch von Nachrichten zwischen verschiedenen Pools dar [58]. Sie können auf eine Übermittlung personenbezogener Daten an Dritte hinweisen (z. B. Transfer von Kundendaten an einen externen Dienstleister).

- **Textannotationen und Assoziationen** dienen dazu, zusätzliche Informationen zu Prozessmodellen hinzuzufügen, die nicht durch die standardmäßigen BPMN-Elemente abgedeckt sind [58]. Sie können genutzt werden, um die Art der Datenverarbeitung zu präzisieren, z. B. „enthält E-Mail-Adresse“.

BPMN-XML

BPMN-Modelle werden in einer XML-Serialisierung, der BPMN 2.0 XML, gespeichert [58]. Diese Darstellung enthält alle relevanten Strukturinformationen, wie Elementtypen, Namen, Beziehungen, Zuordnungen, Positionen der Elemente, und wird von vielen Prozess-Engines und Modellierungswerkzeugen wie Camunda [22] und BPMN.io [21] unterstützt. Für diese Arbeit dient BPMN-XML als Eingabeformat der Klassifizierungspipeline (siehe Kapitel 4).

Jedes BPMN-Element hat ein eindeutiges `id`-Attribut, das als eindeutige Referenzierung dient [58]. Diese stabile `id` ist für die Klassifizierungspipeline wichtig, da sie eine stabile Referenzierung der Aktivitäten und anderer Elemente ermöglicht. Dies ist insbesondere dann relevant, wenn die Ergebnisse der Klassifizierung auf die ursprünglichen Prozessmodelle zurückgeführt werden müssen.

Beispiel einer Datenassoziation als Datenschutzsignal

Abbildung 2.2 zeigt ein einfaches Beispiel, wie eine Datenassoziation die DSGVO-Relevanz einer Aktivität verdeutlichen kann. In Abbildung 2.2a ist die Aktivität „Review data“ ohne Datenassoziation dargestellt, was wenig über die Art der verarbeiteten Daten aussagt. In Abbildung 2.2b hingegen zeigt die eingehende Datenassoziation von einem Datenspeicher „Customer DB“, dass die Aktivität personenbezogene Daten verarbeitet. Dies macht die Aktivität als potenziell datenschutzkritisch erkennbar. Dieses Beispiel unterstreicht die Notwendigkeit, den gesamten Kontext einer Aktivität zu betrachten, um fundierte Rückschlüsse auf die Verarbeitung personenbezogener Daten ziehen zu können.

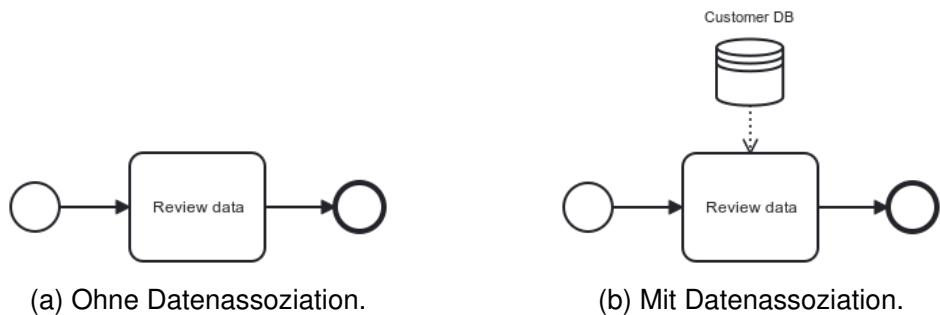


Abbildung 2.2: Beispiel einer Datenassoziation als Datenschutzsignal.

2.3 Large Language Models (LLMs)

LLMs sind große, vortrainierte Sprachmodelle, die auf der Transformer-Architektur basieren. Transformer, erstmals von Vaswani et al. [84] beschrieben, verarbeiten eine textuelle Eingabe nicht strikt sequenziell, sondern beachten alle Tokens einer Sequenz parallel. Über sogenannte Self-Attention gewichten sie, welche Token füreinander relevant sind. Als Token gelten Wörter oder Wortbestandteile, in die der Text vorab zerlegt wird. Dieser Attention-Mechanismus erfasst Abhängigkeiten über große Distanzen innerhalb der Sequenz und ermöglicht dadurch eine effiziente Kontextmodellierung, was das zentrale Prinzip moderner LLMs darstellt. Die Transformer-Architektur bildet heute das Fundament moderner Sprachmodelle wie der GPT-Familie von OpenAI [6, 46, 65], die durch ChatGPT breite Anwendung finden.

In chatbasierten Systemen wird das Verhalten des LLM über System- und User-Prompts gesteuert. Gutes Prompt Engineering kann die Leistung und Format-Treue der Ausgabe verbessern, ohne dass die Modellparameter verändert werden müssen [42]. Ein deutlicher Vorteil aktueller LLMs ist Zero-/Few-Shot Learning. Damit lassen sich Aufgaben allein über Instruktionen und wenige Beispiele lösen, ohne dass erneutes Training benötigt wird [10, 42]. Das ist besonders nützlich für Klassifikaionsaufgaben, bei denen nur wenige gelabelte Beispiele vorliegen, wie etwa die Identifikation von DSGVO-kritischen Aktivitäten in Prozessmodellen.

Um LLMs in automatisierten Pipelines zu integrieren, sind schema-konforme Ausgaben, wie ein gültiges JSON, unerlässlich. In der Praxis gibt es dafür drei Ansätze:

1. Klare Angaben über das Ausgabeformat im System- oder User-Prompt [42].

2. API-gestützte Mechanismen wie Function Calling oder Structured-Output/JSON-Mode mit Schemaüberprüfung [5, 61, 66].
3. Constrained Decoding, das die Generierung auf eine vorgegebene Grammatik beschränkt. Ein Beispiel ist PICARD: Bei jedem Generationsschritt des Language Model (LM) werden nur zulässige Tokens ausgewählt [78].

Typische Fehlerbilder bei der Nutzung von LLMs sind Halluzinationen. Diese sind plausibel wirkende, aber fehlerhafte Aussagen und Formatfehler, wie z. B. ungültiges JSON. In [38] wird argumentiert, dass Halluzinationen bereits beim Erstellen des LLM durch die Trainings- und Evaluationsmethoden begünstigt werden, die das Modell dazu bringen, eher zu raten als Unsicherheit zuzugeben. Das Raten bei Unsicherheit verbessert die Testergebnisse. Gegenmaßnahmen gegen Halluzinationen sind u.a. präzisere Prompts, Informationserweiterung des Prompts durch Retrieval Augmented Generation (RAG) und Self-Check/Retry-Strategien als Post-Processing Methoden nach der Generierung [37].

Die meisten großen LLMs werden von Unternehmen wie OpenAI, Google oder Anthropic entwickelt und als API-Dienste angeboten. In der Industrie zählt GPT-4o aktuell zu den am weit verbreitetsten Modellen [63]. Es ist ein multimodales Modell mit starken Text-, Bild- und Audiofähigkeiten. Proprietäre Modelle wie GPT-4o sind leistungsfähig, bringen jedoch mehrere Nachteile mit sich. Dazu zählen hohe Kosten und mangelnde Transparenz. Außerdem erfolgt die Datenverarbeitung serverseitig auf Infrastruktur der Anbieter, die sich teils außerhalb der EU befindet, wo die DSGVO nicht gilt. Für die Verarbeitung personenbezogener Daten innerhalb der EU ist das problematisch. Eine Übermittlung in Drittländer ist nur zulässig, wenn dort der Auftragsverarbeiter sämtliche Vorgaben aus Kapitel 5 (Art. 44-50) der DSGVO einhält [18].

Als Alternative zu proprietären Modellen steht eine wachsende Zahl frei verfügbarer Open-Source-LLMs zur Verfügung, die auch lokal betrieben werden können. Prominente Beispiele sind die Modelle von Mistral [4], Deepseek [3] und Qwen [70]. Der lokale Betrieb ermöglicht volle Kontrolle darüber, wo und wie Daten verarbeitet werden. Das erleichtert die Einhaltung datenschutzrechtlicher Anforderungen. Zudem bieten Open-Source-Modelle weitere Vorteile wie geringere Kosten und hohe Anpassbarkeit. In dieser Arbeit werden sowohl proprietäre als auch Open-Source-LLMs evaluiert (siehe Kapitel 7).

2.4 Verwandte Arbeiten

Dieses Kapitel bündelt Arbeiten zur automatisierten *Klassifikation datenschutzkritischer Aktivitäten* in Geschäftsprozessen und zur *Nutzung von LLMs* in Datenschutzaufgaben und Tätigkeiten im Business Process Management (BPM). Im Fokus stehen (1) frühe Klassifikations- und Modellierungsansätze, (2) die LLM-basierte Analyse von Richtlinientexten bis hin zu strukturierter Extraktion, (3) Qualitätssicherung, Prompting und Reproduzierbarkeit sowie (4) der Einsatz von LLMs im BPM-Lebenszyklus. Abschließend werden Forschungslücken abgeleitet.

Frühe Ansätze: Klassifikation und modellbasierte Kennzeichnung

Die Identifikation von Prozessschritten mit der Verarbeitung personenbezogener Daten ist eine Voraussetzung wirksamer DSGVO-Konformität, da nur so technische und organisatorische Maßnahmen gemäß Art. 32 Abs. 1 DSGVO (Vertraulichkeit, Integrität, Verfügbarkeit, Belastbarkeit) zielgerichtet festgelegt werden können [18]. Nake et al. [55] beschreiben einen ersten automatisierten Ansatz: Mit einem überwachten Verfahren (Lernen aus gelabelten Beispielen) klassifizieren sie Aktivitäten in *textuellen* Prozessbeschreibungen als DSGVO-kritisch bzw. unkritisch. Der Datensatz umfasst 37 Prozesse mit 509 Aktivitäten. In der stärksten Konfiguration werden ein F1-Score von 0,81 und ein Recall von 0,83 erreicht. Die Generalisierbarkeit bleibt aufgrund des kleinen, nicht repräsentativen Datensatzes begrenzt. Fehler entstehen u. a. durch zu wenige Trainingsbeispiele für bestimmte Merkmalswerte. Der Ansatz ist daher als Assistenz für Datenschutzbeauftragte zu verstehen, nicht als vollständige Automatisierung.

Komplementär dazu markieren Capodieci et al. [14] BPMN-Elemente mit DSGVO-Metadaten via *Tagged Values* (GDPR:legalbasis, GDPR:Duration, GDPR:risklevel, GDPR:ispersonaldataprocessing/GDPR:personaldata), sodass Datenschutzaspekte bereits vor der Implementierung der Geschäftsprozesse prüfbar werden. In eine ähnliche Richtung zielt die designorientierte Arbeit von Agostinelli et al. [2], die DSGVO-Anforderungen als wiederverwendbare Muster („Datenpanne“, „Einwilligung zur Nutzung der Daten“, „Recht auf Auskunft/Berichtigung“,

„Datenübertragbarkeit“, „Recht auf Vergessenwerden“) für eine transparente Einbettung in BPMN formalisiert.

LLMs für Policy-Analyse

Ciaramella et al. [15] nutzen BERT, RoBERTa und DistilBERT, um Sätze aus Online-Datenschutzerklärungen gezielt im Hinblick auf die Informationspflicht aus Art. 13 (2)(b) DSGVO (Hinweis auf Berichtigung/Lösung) zu klassifizieren. Die Ergebnisse sind *moderat* und zeigen, dass innerhalb einer Erklärung konforme und nicht konforme Passagen koexistieren. Daher schließen sie daraus, dass Konformität kein binäres Gesamteurteil auf Textebene ist.

Neuere Arbeiten gehen darüber hinaus: Hooda et al. [25] stellen mit *PolicyLR* eine logikbasierte Repräsentation von Richtlinien vor und übersetzen Richtlinientexte mit einem zweistufigen LLM-Compiler (Übersetzung → Entailment-Prüfung) in atomare Formeln. Als Evaluationsbasis dient *ToS;DR* (Terms of Service; Didn't Read), eine von der Community betriebene Plattform, auf der Freiwillige Passagen aus Nutzungs- und Datenschutzerklärungen mit prägnanten *Cases* zu Datenpraktiken annotieren (z. B. „Sie können Ihren Inhalt von diesem Dienst löschen“). Der Compiler extrahiert solche *Cases* aus Richtlinientexten und prüft anschließend, ob sie logisch aus dem Text folgen (*Entailment*). Mit gemma2-27b erreicht der PolicyLR-Compiler eine Precision von 0,84 bei einem Recall von 0,88. Damit werden Compliance-Checks, Konsistenzprüfungen und Vergleiche von Datenschutzerklärungen auf Basis formaler Repräsentationen möglich.

Rodriguez et al. [75] optimieren Prompts, Parameter und die Kontextaufteilung (Chunking) für die feingranulare Extraktion von Erhebungs- und Weitergabepraktiken mit GPT-4 Turbo. Auf MAPP erreichen sie einen F1-Score von 0,935, und auf OPP-115 einen F1-Score von 0,93, eine Precision von 0,949, einen Recall von 0,912 und eine Accuracy von 0,904, während Llama-2-70B-Chat mit einem F1-Score von 0,882 leicht darunter liegt. MAPP ist eine von Rechtsexperten manuell annotierte Sammlung aus 64 App-Datenschutzerklärungen, die die Erhebung und Weitergabe personenbezogener Daten auf Absatzebene kennzeichnen. OPP-115 umfasst 115 manuell annotierte Datenschutzerklärungen mit einem zu MAPP ähnlichen Annotationsschema und bietet damit eine breitere Domänenabdeckung.

Qualitätssicherung, Prompting und Reproduzierbarkeit

Halluzinationen sind ein zentrales Problem bei LLMs für Datenschutzaufgaben und führen zu fehlerhaften Klassifikationen. Zur Reduktion von Halluzinationen koppeln Silva et al. [80] einen erklärenden LLM-Klassifikator mit einer Entailment-Prüfung. Entailment bezeichnet die Entscheidung, ob eine Aussage aus einem Text logisch folgt. Der Klassifikator liefert Label und textuelle Begründung, ein Filter prüft diese Begründung erneut, und ein Entailment-Verifikator lässt nur logisch gestützte Entscheidungen zu. Auf OPP-115 steigt der Macro-F1-Score auf 0,63 (+11,2%). Die zusätzliche Prüfstufe erhöht die Precision von 0,38 auf 0,61, senkt jedoch den Recall von 0,85 auf 0,61.

Neben der nachgelagerten Verifikation wirkt auch die Eingabe als Qualitätshebel: Von den Zero-/Few-Shot-Grundlagen [10, 42] über die RoPA-Generierung [68] zeigen zahlreiche Arbeiten, dass Beispielanzahl, Kontextaufbereitung und Modellwahl entscheidend sind. Von Schwerin et al. [79] zeigen im Datenschutzkontext jedoch auch, dass LLMs bereits eine sehr gute Grundperformance liefern und Few-Shot-Prompting die Qualität nur begrenzt steigert. Ein Few-Shot-Ansatz kann allerdings bei der Steuerung des Ausgabeformats helfen, ohne die Modelle neu trainieren zu müssen. Sie zeigen ebenfalls, dass kleinere, offene Modelle wie Qwen2-7B in bestimmten Bereichen größere proprietäre Modelle wie GPT-4 übertreffen können.

Reimers und Gurevych [72] haben untersucht, wie sich die nichtdeterministische Natur von LLMs auf die Ergebnisqualität auswirkt. Sie fanden heraus, dass die Abhängigkeit vom Seed-Wert zu statistisch signifikanten Unterschieden in der Performance führen kann. Diese Varianz kann dazu führen, dass ein modernes, leistungsfähiges Modell je nach Seed von sehr gut bis mittelmäßig abschneidet. Um das in Experimenten mit LLMs zu berücksichtigen, wird vorgeschlagen, Score-Verteilungen zu vergleichen, die auf mehreren Durchläufen mit unterschiedlichen Seeds basieren. Dadurch werden die Ergebnisse robuster, und das Risiko sinkt, dass ein Modell nur aufgrund eines günstigen Seeds gut oder aufgrund eines ungünstigen Seeds schlecht abschneidet.

LLMs im BPM-Lebenszyklus

Über Richtlinientexte hinaus skizzieren Vidgof et al. [85] zentrale Forschungsrichtungen für LLM-gestütztes BPM, darunter Best Practices, BPM-spezifische Datensätze und Leitlinien zu Prompting und Modellauswahl. Kourani et al. [39] vergleichen in einem Benchmark mit 20 Prozessen 16 LLMs zur Transformation von Prozessbeschreibungen in ausführbare Modelle. Claude-3.5-Sonnet erzielt die höchste durchschnittliche Qualitätsbewertung, während z. B. Mixtral-8x22B zurückfällt. Es wird ein positiver Zusammenhang zwischen Fehlerbehandlung und Modellqualität beobachtet und, dass durch Output-Optimierungstechniken schwächere Modelle spürbar verbessert werden können.

Daneben befassen sich Bernardi et al. [8] mit *dialogorientierter Unterstützung*: Gemeint ist, dass Mitarbeitende in normaler Sprache mit einem Assistenten über ihre Prozesse sprechen und sich bei Aufgaben helfen lassen. Technisch kommt ein Business-Process-LLM (BPILLM) zum Einsatz, das RAG mit feinabgestimmten LLaMA-2-Modellen kombiniert und, wenn genug Kontext vorliegt, präzise Auskünfte etwa zu Aktivitäten und Sequenzflüssen liefert.

Forschungslücken

Aus der Literatur ergeben sich mehrere offene Fragen, die die vorliegende Arbeit adressiert:

1. **Granularität und Domänenfokus.** Viele Studien fokussieren einzelne Artikel der DSGVO (z. B. Art. 13) oder allgemeine Privacy-Tasks. Eine systematische Klassifikation *kompletter* Geschäftsprozesse nach datenschutzrelevanten Aktivitäten ist selten. Zudem sind Datensätze klein und wenig repräsentativ [55].
2. **LLM-Anwendung in Geschäftsprozessen.** Während es Benchmarks für Prozessmodellierung gibt, fehlen reproduzierbare Benchmarks speziell für die Klassifikation datenschutzkritischer Prozessschritte. Positionsarbeiten wie Vidgof et al. [85] fordern BPM-spezifische Datensätze und Modelle. Öffentlich verfügbare, auf den europäischen Rechtsraum zugeschnittene Benchmarks sind rar.

3. **Erklärbarkeit und Halluzinationen.** LLMs erzeugen teils überzeugende, aber unzutreffende Ausgaben. Ansätze wie Silva et al. [80] und Hooda et al. [25] zeigen, dass Schlussfolgerungsprüfer oder formale Repräsentationen nötig sind, um Halluzinationen zu reduzieren.
4. **Datenschutz- und Sicherheitsbedenken.** Studien nutzen häufig geschlossene Modelle wie GPT-4, deren Einsatz aufgrund möglicher Datenübermittlungen in die USA datenschutzrechtlich problematisch sein kann. Offene Modelle wie Qwen2-7B liefern vergleichbare Ergebnisse [79] und sind für EU-Organisationen potenziell vorteilhaft.
5. **Prompt-Engineering-Leitlinien.** Obwohl mehrere Arbeiten den maßgeblichen Einfluss von Prompt-Gestaltung (z. B. Beispielanzahl, Kontexttrennung) belegen [68, 42], fehlen breit akzeptierte Leitfäden speziell für Datenschutz- und BPM-Kontexte.

Diese Lücken unterstreichen den Bedarf an umfassenden, reproduzierbaren Benchmarks sowie an robusten Methoden zur Klassifikation datenschutzkritischer Aktivitäten in Geschäftsprozessen unter Berücksichtigung der europäischen DSGVO. Das folgende Kapitel präzisiert die Problemdefinition und die Qualitätsziele der vorliegenden Arbeit.

3 Problemdefinition und Zielkriterien

Ausgehend von Kapitel 1 und den Grundlagen zu DSGVO, BPMN und LLMs führt dieses Kapitel die Kernaufgabe ein: die binäre Klassifikation einzelner BPMN-Aktivitäten in *kritisch* und *unkritisch* mithilfe von LLMs. Es definiert die Qualitätsziele und den fachlichen Geltungsbereich der Arbeit und beschreibt das Experimentdesign zur systematischen Beantwortung der Forschungsfragen. Damit legt es die Grundlage für die in Kapitel 4 dargestellte Klassifizierungspipeline sowie für die anschließenden Experimente und deren Auswertung.

Abbildung 3.1 zeigt einen Beispielprozess, der in den folgenden Abschnitten als laufendes Beispiel dient.

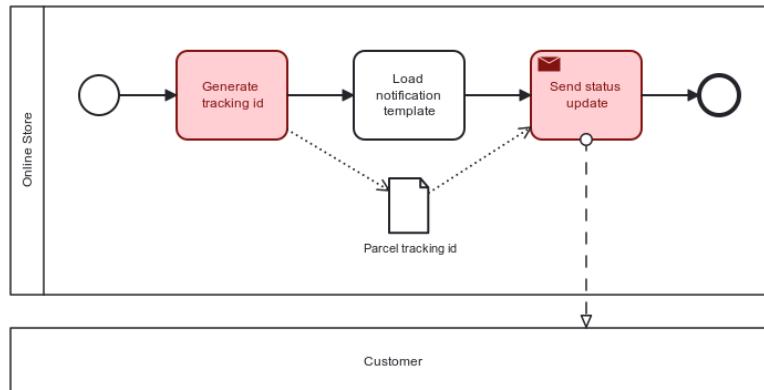


Abbildung 3.1: Beispielprozess zur Veranschaulichung der Aufgabenstellung.

Der Prozess modelliert den Versand eines Statusberichts eines Onlineshops an Kunden. Dabei werden personenbezogene Daten in den Aktivitäten „Generate tracking id“ und „Send status update“ verarbeitet, die die stabilen ids `Activity_generate` und `Activity_send` besitzen. Die Aktivität „Load notification template“, mit der id `Activity_template`, dient als Negativbeispiel.

3.1 Aufgabenstellung

Ziel der Arbeit ist eine *binäre Klassifikation* auf Ebene einzelner BPMN-Aktivitäten: Für jede Aktivität eines Eingabemodells im BPMN-XML-Format (Version 2.0.2) [58] soll entschieden werden, ob sie *kritisch* im Sinne des Datenschutzrechts ist oder nicht.

- **Eingabe** ist ein valides BPMN-XML mit stabilen id-Attributen je Aktivität [58].
- **Ausgabe** ist eine Menge von Aktivitäts-ids, die als *kritisch* klassifiziert wurden. Optional wird zusätzlich eine natürlichsprachige Begründung für einzelne Entscheidungen ausgegeben. Im Fall der Klassifizierungspipeline dieser Arbeit werden die Begründungen vom LLM generiert. Diese dienen ausschließlich der Nachvollziehbarkeit der gewählten Klassifizierungen, werden allerdings nicht in der Evaluation berücksichtigt.

Zur Veranschaulichung zeigt Listing 3.1 einen vereinfachten Auszug aus dem BPMN-XML des laufenden Beispiels. Die erwartete Ausgabe ist {Activity_generate, Activity_send}.

Listing 3.1: BPMN-XML-Auszug des laufenden Beispiels

```
1 <task id="Activity_generate" name="Generate tracking id"/>
2 <task id="Activity_template" name="Load notification template"/>
3 <task id="Activity_send" name="Send status update"/>
```

Begriffsbestimmung „kritisch“

Eine Aktivität gilt in dieser Arbeit als *kritisch*, wenn sie *personenbezogene Daten* verarbeitet. Personenbezogene Daten sind, nach Art. 4 Abs. 1 DSGVO [18], alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person beziehen. Gemäß Art. 4 Abs. 2 DSGVO [18] umfasst Verarbeitung jede mit personenbezogenen Daten vorgenommene Operation, wie z. B. Erheben, Speichern, Abrufen, Verwenden, Übermitteln und Löschen. Dies schließt auch die *Nutzung bereits vorhandener Daten* (z. B. Lesen/Abgleichen) ein. Am laufenden Beispiel bedeutet dies konkret: Activity_send ist kritisch, da beim Versand typischerweise personenbezogene Daten (z. B. E-Mail-Adresse) verarbeitet werden.

Activity_generate ist kritisch, weil die Tracking-id im Kontext des Kundenkontakts einer Person zugeordnet werden kann. Activity_template ist im Regelfall nicht kritisch, sofern lediglich eine generische Vorlage geladen wird und keine personenbezogenen Daten einfließen.

Diese Aufgabenstellung reiht sich in Arbeiten zur Kennzeichnung kritischer/unkritischer Tätigkeiten in Prozessartefakten ein und bildet die Referenz für die Qualitätsziele im nächsten Abschnitt. [55]

3.2 Qualitätsziele

Die Aufgabe der datenschutzrechtlichen Klassifikation von Prozessen ist risiko-sensitiv. Übersehene kritische Aktivitäten, auch FN genannt, bergen erhebliche Compliance-Risiken und können zu hohen Strafen nach der DSGVO führen. Beispielsweise erhielt Meta Platforms Ireland Limited (Meta IE) 2023 aufgrund von rechtswidriger Übermittlung von EU Nutzerdaten in die USA eine Geldbuße von 1,2 Milliarden Euro [1]. Auch Amazon wurde 2025 nach einem langjährigen Rechtsstreit wegen Datenschutzverstößen mit 746 Millionen Euro bestraft [16, 74]. Um derartige Strafen zu vermeiden, müssen kritische Aktivitäten zuverlässig identifiziert werden. Daher ist das **Hauptziel** der Klassifikation:

© Hauptziel

Maximaler Recall bei *minimalen FN* und zugleich *akzeptabler Precision*, damit der manuelle Prüfaufwand durch False Positives (FP) begrenzt bleibt.

Konfusionsmatrix und Metriken

Zur Bewertung des Hauptziels wird eine Konfusionsmatrix verwendet. Im vorliegenden binären Kontext entspricht die positive Klasse DSGVO-kritischen Aktivitäten. Die vier Felder der Konfusionsmatrix haben folgende Bedeutung [81]:

True Positives (TP) sind korrekt als kritisch erkannte Aktivitäten. Sie bilden den unmittelbaren *Nutzen* der Klassifikation.

FP sind fälschlich als kritisch markierte Aktivitäten. Sie erhöhen den manuellen Prüfaufwand, verursachen aber *keine* unmittelbaren Compliance-Risiken.

True Negatives (TN) sind korrekt als unkritisch eingestufte Aktivitäten und reduzieren den Gesamtaufwand.

FN sind übersehene kritische Aktivitäten. Sie sind besonders problematisch, da sie zu ausbleibender Risikobehandlung und potenziellen Bußgeldern führen können [55].

Aus diesen Größen leiten sich die Evaluationsmetriken ab. Relevante Metriken für eine aussagekräftige Evaluierung sind *Accuracy*, *Precision*, *Recall*, *F1-Score* sowie die Konfusionsmatrix-Zahlen (TP, FP, TN, FN) und die Anzahl korrekt/inkorrekt klassifizierter Testfälle. Technische Fehler wie z. B. Parsing-Fehler oder überschritene Token Limits werden separat ausgewiesen.

Diese Metriken sind in Information Retrieval und maschinellem Lernen seit langem etabliert und bilden den De-facto-Standard zur Bewertung von Klassifikatoren [43, 55, 81]. Für das hier betrachtete binäre Problem gelten:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Zielwerte

Vor dem Hintergrund der oben definierten Metriken und dem Hauptziel werden im Folgenden mithilfe von vergleichbarer Literatur realistische Zielkorridore abgeleitet.

Ähnliche Arbeiten, wie von Nake et al. [55], zeigen Referenzwerte von einem maximalen *Recall* $\approx 0,83$ und *F1-Score* $\approx 0,81$ bei der Identifikation DSGVO-kritischer Aufgaben in Prozessbeschreibungen. Jüngere DSGVO-nahe LLM-Studien berichten von *Precision/Recall* im hohen 0,8x bis 0,9x-Bereich [25] und F1-Scores von $\approx 0,68$ bis zu $\approx 0,79$ [79].

Basierend darauf werden folgende Zielkorridore als *pragmatische Abnahmekriterien* gesetzt:

- **Recall** soll ein Mindestniveau von $\geq 0,80$ erreichen und ein *angestrebter* Bereich ist $\geq 0,85$.
- **Precision** soll $\geq 0,75$ als Untergrenze zur Begrenzung des Prüfaufwands erreichen.
- **F1-Score** soll $\geq 0,80$ erreichen.

Im Kontext des laufenden Beispiels bedeutet dies u. a.: Eine Strategie „alles ist kritisch“ liefert zwar Recall = 1,0, unterschreitet mit Precision $\approx 0,67$ jedoch das Ziel. Stattdessen ist daher eine ausgewogene Erkennung gefordert, die Activity_template korrekt als unkritisch belässt.

Nake et al. [55] zeigen, dass selbst ein *Recall* von 0,83 für kritische Aufgaben ohne menschliche Nachkontrolle nicht ausreicht, da die Strafen für Nichteinhaltung der DSGVO sehr hoch sind. Viel mehr eignet sich ein System mit diesem Recall-Wert für *assistierte* Prüfungen, bei denen die Ergebnisse durch qualifizierte Experten validiert werden. Für ein Screening von Geschäftsprozessen, wie es in dieser Arbeit angestrebt wird, sind die genannten Zielwerte daher als realistisch und praxisrelevant einzuschätzen.

Zusammenfassend fixieren die Zielwerte die angestrebte Performance. Im nächsten Abschnitt wird dargelegt, dass aufgrund der nicht-deterministischen Natur von LLMs die Ergebnistabilität über wiederholte Läufe berücksichtigt werden muss.

Stabilität über Wiederholungen

Da LLMs nicht-deterministisch sind, ist das Berichten eines einzelnen Leistungswertes nicht ausreichend für den Vergleich von Modellen. Studien wie von Reimers et al. [72] zeigen, dass die Abhängigkeit vom Seed-Wert der LLMs zu statistisch signifikanten Unterschieden in der Performance führen kann. Diese Varianz kann dazu führen, dass ein modernes, leistungsfähiges Modell von sehr gut bis mittelmäßig abschneidet. Stattdessen wird vorgeschlagen, Score-Verteilungen zu vergleichen, die auf mehreren Durchläufen basieren. Dadurch wird das Risiko reduziert, dass ein Modell nur aufgrund eines günstigen Seeds gut oder aufgrund eines ungünstigen Seeds schlecht abschneidet. Im laufenden Beispiel ist „Generate tracking id“ grenzwertig und wird in der Praxis von Modellen gelegentlich fälschlich als *nicht-kritisch*

markiert (FN) - Wiederholungen und das Berichten von Mittelwert \pm Standardabweichung (SD) erfassen diese Instabilität.

In dieser Arbeit werden daher die Ergebnisse auf Basis von Wiederholungen berichtet. Es wird der Mittelwert \pm Standardabweichung je Metrik angegeben. Modellvergleiche basieren am Ende auf diesen Verteilungen und nicht auf Einzelfällen.

3.3 Scope und Annahmen

Dieser Abschnitt definiert Geltungsbereich, Annahmen und Risiken des Ansatzes. Dadurch wird eine klare Einordnung der Ergebnisse und ihrer Reproduzierbarkeit ermöglicht.

Die folgenden Punkte definieren den Geltungsbereich der Arbeit:

- Klassifiziert werden ausschließlich *Aktivitäten*. Dafür wird sinnvoller Kontext berücksichtigt, wie Labels, Pools/Lanes (z. B. „Onlineshop“, „Kunde“), Message Flows sowie vorhandene Datenobjekte (z. B. „Paket Tracking-id“).
- Labels und Artefakte liegen in Deutsch und Englisch vor.
- Es handelt sich um ein *Screening*, nicht um eine Rechtsprüfung. Kritisch klassifizierte Aktivitäten sind anschließend juristisch zu prüfen.

Zusätzlich sind die folgenden Annahmen und potenziellen Risiken für die Interpretation der Ergebnisse relevant:

- Bei fehlenden Datenobjekten oder mehrdeutigen Labels kann sich die Einschätzung verschlechtern. Das ist ein bekanntes Problem in ähnlichen Studien [55].
- Optional generierte LLM-Begründungen sind als *Hilfetexte* zu verstehen, um die Entscheidung des LLMs besser einordnen zu können, bilden aber nicht zwingend die tatsächlichen Entscheidungsgründe des Modells ab.
- Ungültiges BPMN-XML oder Laufzeitfehler werden als „technischer Fehler“ erfasst und nicht in die Metrikzählung eingerechnet. Sie werden separat berichtet.

3.4 Experimentdesign

Das gesamte Kapitel definierte die binäre Klassifikation von BPMN-Aktivitäten als kritisch/unkritisch mit Fokus auf maximalen Recall bei akzeptabler Precision und legte Qualitätsziele, Metriken, Geltungsbereich sowie Annahmen fest. Darauf aufbauend beschreibt dieser Abschnitt das Experimentdesign, mit dem LLMs fair und reproduzierbar verglichen werden, um die Forschungsfrage **FF1** sowie die Unterfragen **UF1-UF4** zu beantworten. Die konkrete Ausgestaltung und Durchführung der Experimente wird in Kapitel 8 erläutert. Im Folgenden werden die wesentlichen Aspekte des Experimentdesigns beschrieben:

Ziel Ziel ist ein transparenter Vergleich mehrerer LLMs, die alle dieselbe Klassierungspipeline durchlaufen. Sie wird in Kapitel 4 daher so entworfen, dass sich das LLM austauschen lässt. Die Auswahl der im Evaluationsframework aus Kapitel 6 zu nutzenden LLMs erfolgt zur Laufzeit anhand übergebener Identifikationsparameter (z. B. Modellname, Basis-URL/Endpunkt).

Vergleichsgegenstand Die Experimente werden über eine deklarative Konfiguration definiert, siehe Kapitel 6.2. Sie legt fest, welche Modelle, Datensätze und weitere Parameter zum Einsatz kommen. Je nach Auswahl werden mehrere Modelle und Modellvarianten parallel im Evaluationsframework ausgeführt, darunter Open-Source und kommerzielle Modelle. Die deklarative Konfiguration sorgt für Portabilität und Wiederholbarkeit.

Datenbasis Als Datenbasis dienen die im Labeling-Tool erzeugten, gelabelten Testdatensätze, siehe Kapitel 5. Ein Testdatensatz enthält mehrere gelabelte Testfälle. Ein Testfall umfasst ein BPMN-Prozessmodell mit Labels, die Aktivitäten als DSGVO-kritisch markieren. Die Auswahl der Datensätze für ein Experiment erfolgt in der Evaluierungskonfiguration, und das Laden der Testfälle erfolgt während der Laufzeit. Die Datensätze sollten idealerweise unterschiedliche Eigenschaften abdecken, damit die Forschungsfrage und die Unterfragen möglichst umfassend beantwortet werden. Unterschiede können sich etwa in der Domäne, der Größe der Prozesse, den eingesetzten Sprachen oder den verwendeten BPMN-Elementen zeigen.

Metriken und Erfolgskriterium Ausgewertet werden die in Abschnitt 3.2 beschriebenen Metriken: Accuracy, Precision, Recall und F1. Zusätzlich werden die

Kennzahlen der Konfusionsmatrix betrachtet: TP, FP, TN, FN. Ein Testfall gilt als *bestanden*, wenn die vom Modell als kritisch ausgegebenen Aktivitäten exakt den gelabelten kritischen Aktivitäten entsprechen. Technische Fehler werden separat ausgewiesen.

Auf Basis dieser Definitionen erfolgt die Durchführung der Experimente in folgenden Schritten:

1. **Konfiguration laden.** Die Konfiguration mit Modellen, Datensätzen und optionalem seed wird geladen.
2. **Ausführung.** Für jedes Modell werden alle ausgewählten Testfälle durch die Klassifizierungspipeline verarbeitet. Pro Testfall werden TP, FP, FN, TN sowie der Status „bestanden“ oder „nicht bestanden“ berechnet.
3. **Stabilität.** Die Läufe erfolgen mit niedriger temperature¹ und festem seed, sofern das jeweilige LLM dies unterstützt. Um die Nicht-Deterministik moderner LLMs abzubilden, werden die Experimente mehrfach mit unterschiedlichen Seeds wiederholt. Die Ergebnisse werden über die Läufe gemittelt.
4. **Bericht.** Aggregierte Kennzahlen pro Modell, wie Konfusionsmatrix, die genannten Metriken sowie die Bestehensraten werden ausgegeben. Metadaten wie verwendete Modelle, Datensätze und Seeds werden dokumentiert.

Dieses Kapitel definiert, *was* verglichen wird: Modelle, Datensätze und Metriken. Es beschreibt zudem, *wie* der Vergleich erfolgt. Kapitel 8 dokumentiert später die praktische Umsetzung mit konkreten Modellen, exakten Parameterwerten, Seeds sowie den vollständigen genutzten Konfigurationen. Im nächsten Kapitel folgt das Design und die Implementierung der Klassifizierungspipeline, die für den Vergleich der LLMs verwendet wird.

¹Die temperature steuert die Zufälligkeit der Textgenerierung bei LLMs. Niedrige Werte liefern stabilere Antworten, hohe Werte vielfältigere, jedoch weniger verlässliche [57].

4 Design und Implementierung der Klassifizierungspipeline

Dieses Kapitel beschreibt die Pipeline zur Klassifikation DSGVO-kritischer Aktivitäten in BPMN-Prozessen. Ausgehend von der in Kapitel 3.1 formulierten Aufgabenstellung wird der gesamte Weg von der Eingabe eines *BPMN-XML* über die Vorverarbeitung, das Prompt Engineering bis hin zur strukturierten, schema-konformen Ausgabe aufgezeigt. Außerdem wird ein HTTP-basiertes API-Design vorgestellt, das die Einbindung in weitere Werkzeuge und das Evaluationsframework ermöglicht. Der Prozessfluss der Klassifizierungspipeline ist in Abbildung 4.1 dargestellt und wird in diesem Kapitel im Detail erläutert.

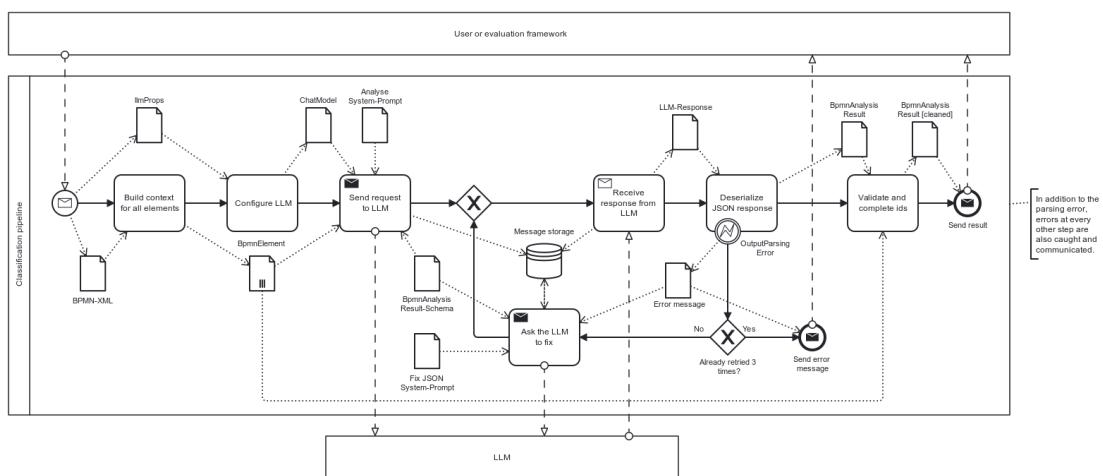


Abbildung 4.1: BPMN-Diagramm der Klassifizierungspipeline.

Die Klassifizierungspipeline soll eine binäre Entscheidung auf Ebene einzelner BPMN-Aktivitäten treffen: Für jede Aktivität eines Eingabemodells wird bestimmt, ob sie *kritisch* im Sinne der DSGVO ist. Die Pipeline ist so konzipiert, dass sie mit Modellen aus gängigen Modellierungswerkzeugen kompatibel ist. Dadurch kann sie

in bestehende Modellierungswerzeuge wie Camunda Modeler [22] integriert werden, um einen praktischen Einsatz in realen Prozessmodellierungs-Workflows zu ermöglichen.

4.1 BPMN Preprocessing

Ziel der Vorverarbeitung (Preprocessing) ist es, für jedes Flow-Element einen *strukturierten Kontext* zu erzeugen. Dieser Kontext umfasst die eigenen Attribute, wie `id`, `name` und `documentation`, sowie die Beziehungen zu anderen Elementen im BPMN-Diagramm. Dazu gehören vorangehende und nachfolgende Flow-Elemente, Datenobjekte, assoziierte Elemente sowie Informationen über den Pool und die Lane, in denen sich das Element befindet. Das Parsen des BPMN-XML erfolgt mit der *Camunda BPMN Model API*, die das XML in ein Objektmodell überführt [12, 13]. Auf dieser Basis werden die relevanten Informationen extrahiert und in der Datenklasse `BpmnElement` strukturiert abgelegt. Die Datenklasse ist in Listing 4.1 zu sehen.

Listing 4.1: Interne BPMN-Repräsentation je Flow-Element.

```

1  data class BpmnElement(
2      val type: String,
3      val id: String,
4      val name: String? = null,
5      val documentation: String? = null,
6      val poolName: String? = null,
7      val laneName: String? = null,
8      val outgoingFlowElementIds: List<String> = emptyList(),
9      val incomingFlowElementIds: List<String> = emptyList(),
10     val outgoingMessageFlowsToElementIds: List<String> = emptyList(),
11     val incomingMessageFlowsFromElementIds: List<String> = emptyList(),
12     val incomingDataFromElementIds: List<String> = emptyList(),
13     val outgoingDataToElementIds: List<String> = emptyList(),
14     val associatedElementIds: List<String> = emptyList()
15 )

```

Dadurch entsteht für jedes Flow-Element ein umfassender Kontext, der später im Prompt genutzt wird, um dem LLM alle notwendigen Informationen strukturiert bereitzustellen. Außerdem werden durch das Format Tokens eingespart, da irrelevante Informationen, wie die Positionen der Elemente im XML, weggelassen werden. In Abbildung 4.1 ist dieser Schritt über die Aktivität „Build context for all elements“ dargestellt.

4.2 Prompt Engineering

Eine robuste Klassifikation hängt maßgeblich von sorgfältig gestalteten Prompts ab. Ziel ist es, das LLM mit klaren Anweisungen, einem konsistenten Bewertungsschema und präzisen Formatvorgaben so zu steuern, dass es die Klassifizierung zuverlässig löst und strukturierte Ausgaben liefert. Im Folgenden werden zunächst die deklarative Orchestrierung der Kommunikation mit dem LLM mithilfe von LangChain4j und anschließend die Prompt-Konzeption beschrieben.

LangChain4j: deklarative Orchestrierung

Zur Reduktion von Boilerplate und für konsistente Prompts wird *LangChain4j* [41] benutzt. Mit den *AI Services* werden Interaktionen mit dem LLM als Java/Kotlin-Interface *deklarativ* beschrieben. Zur Laufzeit erzeugt LangChain4j einen Proxy, der den System-Prompt injiziert, den User-Prompt aus den Methodenparametern generiert und die LLM-Antwort in den passenden Rückgabetyp deserialisiert [69]. Beim Erstellen des AI Service werden ein *ChatModel*, die Systemnachricht und die Interface-Methoden konfiguriert. Ein *ChatModel* ist die spezifische Implementierung der Chat-Completion-Schnittstelle eines LLM von Langchain4j [40]. Die Methodenparameter der Interface-Methoden repräsentieren die Nutzereingabe. Der Rückgabetyp der Methoden definiert die erwartete Antwortstruktur des LLM. Optional kann jeder Interface-Methode noch ein eigener User-Prompt zugewiesen werden, der bei Laufzeit mit den übergebenen Parametern gefüllt wird [69].

Die Kommunikation mit dem LLM erfolgt damit über einfache Funktionsaufrufe, während LangChain4j Prompt-Erzeugung, Parameterbindung sowie die Deserialisie-

rung der Antwort übernimmt [69]. So kann im Code ohne zusätzlichen Aufwand direkt mit typisierten Objekten gearbeitet werden.

Für die Klassifikation *DSGVO-kritisch vs. unkritisch* wird ein **Zero-Shot**-Ansatz verwendet. Das LLM erhält im System-Prompt eine präzise Instruktion mit Kriterien und illustrativen Beispielfällen, was als kritisch gilt. Es sind jedoch keine Beispiele mit konkreten Ein- und Ausgabe-paaren pro Prozess enthalten. Zero-Shot reduziert den Pflegeaufwand und nutzt die In-Context-Fähigkeiten moderner Modelle, nur über Instruktionen zu generalisieren [10, 42]. Wie genau die Prompts aufgebaut sind, wird im Folgenden beschrieben.

System-Prompt

Der System-Prompt definiert das Verhalten des LLM, zusätzlichen Kontext und das gewünschte AusgabefORMAT. Der vollständige System-Prompt befindet sich im Anhang, siehe Listing 1. Im Kern legt der genutzte System-Prompt Folgendes fest:

1. **Rolle und Auftrag des Modells.** Das Modell agiert als Experte für das Analysieren von BPMN-Modellen auf DSGVO-konformität und prüft sämtliche Aktivitäten eines Prozesses auf Datenschutzrelevanz. Jede Aktivität wird berücksichtigt und die Entscheidung erfolgt auf Basis sämtlicher verfügbarer Kontextinformationen wie Name, Beschreibung, Annotationen sowie Daten- und Nachrichtenassoziationen.
2. **Rechtliche Definitionen nach DSGVO.** Der System-Prompt erläutert die Begriffe „personenbezogene Daten“ und „Verarbeitung“ gemäß Art. 4 DSGVO. Beispiele für personenbezogene Daten umfassen Identifikatoren, Kontakt- und Zahlungsdaten, Beschäftigungsdaten, Gesundheitsdaten, biometrische Merkmale, Standortinformationen und Online-Kennungen. Verarbeitung umfasst Erheben, Speichern, Abrufen, Verwenden, Übermitteln, Ausrichten, Kombinieren, Einschränken, Löschen und Vernichten.
3. **Indikatoren für Kritikalität.** Der System-Prompt enthält typische Auslöser für Datenschutzrelevanz wie Datenerfassung und Dateneingabe, Anlage und Aktualisierung von Datensätzen, Übermittlung oder Offenlegung an andere Systeme oder Dritte, Zahlungen und Finanztransaktionen und noch mehr. Diese

Indikatoren sind mit Beispielen angereichert und dienen als *Entscheidungshelper* für das Modell.

4. **Abgrenzung durch Negativbeispiele.** Der System-Prompt grenzt unkritische Fälle klar ab. Rein administrative oder logistische Schritte ohne Personenbezug werden nicht als kritisch gewertet. Ebenso gilt dies für Fälle in denen anonymisierte Daten verwendet werden und keine Identifikation einer Person mehr möglich ist.
5. **Erwartetes AusgabefORMAT.** Die Antwort erfolgt als strukturierte JSON-Ausgabe mit einer Liste relevanter Aktivitäten. Für jede Aktivität wird die `id` und eine Begründung in natürlicher Sprache ausgegeben. Es werden ausschließlich Aktivitäten zurückgegeben, die nach den Kriterien als datenschutzrelevant eingestuft wurden.

Die Kombination dieser Elemente im System-Prompt stellt sicher, dass das LLM die Aufgabe versteht, die relevanten Kriterien kennt und die Ausgabe in einem maschinenlesbaren Format liefert. So entsteht die Basis für eine zuverlässige Klassifikation. Zu einer Anfrage an ein LLM gehört außerdem stets ein User-Prompt, der die eigentliche Nutzereingabe enthält. Dessen Aufbau wird im nächsten Abschnitt beschrieben.

User-Prompt

Der User-Prompt über gibt dem LLM die konkreten Eingabedaten einer Anfrage. Während der System-Prompt Regeln, Ziele und Ausgabevorgaben festlegt, liefert der User-Prompt die Fall- bzw. Kontextinformationen, auf die diese Regeln angewendet werden.

Der User-Prompt wird mithilfe der Daten aus der Vorverarbeitung aus Abschnitt 4.1 erzeugt und enthält eine Liste von `BpmnElement`-Objekten, siehe Listing 4.1. Die Interaktion mit dem LLM erfolgt deklarativ über *LangChain4j*. Dafür wird die Liste der `BpmnElement`-Objekte als Methodenparameter mit der Annotation `@UserMessage` an die Interface-Methode übergeben und dort automatisch in den User-Prompt eingebettet.

Zur Laufzeit serialisiert *LangChain4j* die `BpmnElement`-Liste zu einem JSON-Array und stellt sie als User-Prompt bereit. Der zuvor konfigurierte System-Prompt wird

bei einer Anfrage an das LLM automatisch dem User-Prompt vorangestellt. Auf diese Weise wendet das LLM die im System-Prompt definierten Kriterien auf die im User-Prompt gelieferten Informationen zum BPMN-Prozessmodell an. Dadurch wird jede Aktivität des Prozesses genauso wie im System-Prompt beschrieben klassifiziert. In Abbildung 4.1 findet dieser Schritt in der Aktivität „Send request to LLM“ statt.

Zusammenfassend setzt der System-Prompt typischerweise das Regelwerk, und der User-Prompt liefert die konkreten Eingabedaten. Besonders in mehrstufigen Dialogen mit dem LLM spielt dieses Muster eine größere Rolle, da der System-Prompt konstant bleibt, während der User-Prompt je nach Anfrage variiert. Im vorliegenden Szenario, wo immer nur genau eine Anfrage pro Prozessmodell gestellt wird, fällt der Unterschied weniger ins Gewicht, als würden sämtliche Vorgaben direkt im User-Prompt stehen. Die Trennung erhöht dennoch die Nachvollziehbarkeit, sorgt für klare Rollen und erleichtert die Wiederverwendung.

Im folgenden Abschnitt wird beschrieben, wie auf dieser Basis strukturierte Ausgaben erzeugt werden, damit im Code direkt mit typisierten Objekten weitergearbeitet werden kann.

Strukturierte Ausgaben mit LangChain4j

Im Fall der Klassifikation wird ein `BpmnAnalysisResult` als Antwort erwartet, also eine Liste von Elementen mit Paaren aus `id`, `reason` und `isRelevant`. Siehe 2 für die vollständige Definition der Datenklasse. Langchain4j inferiert auf Basis des Rückgabetyps der Interface-Methode ein JSON-Schema und fügt dieses automatisch dem User-Prompt zusammen mit der Aufforderung, die Antwort in diesem JSON-Format zu liefern hinzu [69]. Durch die explizite Angabe des gewünschten JSON-Formats im Prompt wird die Format-Treue der Antwort, also die Wahrscheinlichkeit, dass die Antwort tatsächlich dem gewünschten Schema entspricht, erhöht [42].

Einige LLMs unterstützen darüber hinaus die Möglichkeit, das Antwortformat API-seitig zu erzwingen. Das ist beispielsweise bei Mistral und OpenAI der Fall [5, 66]. Falls das LLM die `response_format` Funktionalität unterstützt, setzt LangChain4j dies zusätzlich auf das gewünschte Schema und erzwingt so das Ziel-JSON API-

seitig [69]. Fehlt diese Fähigkeit, greift ausschließlich die Prompt-basierte Schemaanweisung.

Das vom LLM gelieferte JSON deserialisiert *LangChain4j* anschließend automatisch zu einem `BpmnAnalysisResult`. So kann im Code direkt mit einem typsicheren Objekt weitergearbeitet werden. In Abbildung 4.1 ist dieser Prozess über die Aktivitäten „Receive response from LLM“ und „Deserialize JSON response“ dargestellt.

4.3 Validierung der Ausgabe

Zusätzlich zu den in Kapitel 4.2 beschriebenen Maßnahmen stellt die Pipeline mehrere Validierungs- und Korrekturschritte bereit, die in Abbildung 4.1 direkt auf „Deserialize JSON response“ folgen. Diese Schritte dienen dazu, die Qualität und Korrektheit der Ausgabe des LLM zu gewährleisten. Im Folgenden werden die einzelnen Validierungsmechanismen erläutert.

Schema-Parsing und Retry-Mechanismus

Die vom LLM zurückgelieferte Antwort wird zunächst von *Langchain4j* zu einem `BpmnAnalysisResult` deserialisiert. Entspricht die Struktur dabei nicht dem erwarteten JSON-Schema, löst *Langchain4j* eine `OutputParsingException` aus. In diesem Fall greift der in Abbildung 4.1 ab dem Boundary-Error-Event „OutputParsingError“ dargestellte Retry-Mechanismus. Dabei wird bis zu dreimal die ursprüngliche Anfrage erneut gesendet, ergänzt um die Parser-Fehlermeldung sowie eine explizite Anweisung, die Ausgabe exakt gemäß Schema zu formatieren. So bleiben sowohl der Kontext der ursprünglichen Anfrage als auch die Information über den aufgetretenen Fehler erhalten, damit das LLM die Ausgabe entsprechend anpassen kann. Schlagen alle drei Versuche fehl, wird der Fehler an die aufrufende Schnittstelle zurückgegeben und die Klassifizierung gilt als fehlgeschlagen.

Ein zusammenfassender Log-Auszug des Retry-Mechanismus findet sich in Listing 5. Er zeigt exemplarisch, dass zunächst der boolesche Wert `isRelevant` fehlt und im zweiten Versuch korrekt ergänzt wird.

Relevanz-Filterung

Nach erfolgreichem Parsing werden alle Elemente mit `isRelevant = false` entfernt. Dieser Schritt geschieht bereits im Konstruktor der Datenklasse `BpmnAnalysisResult` automatisch. Dieser Mechanismus adressiert modellseitige *Überklassifizierungen*, bei denen das LLM fälschlicherweise `ids` von Aktivitäten ausgibt, obwohl sie nicht DSGVO-kritisch sind. Es wird sichergestellt, dass nur Aktivitäten, die als kritisch klassifiziert wurden, in der finalen Ausgabe verbleiben.

Ohne das `isRelevant`-Flag hat das LLM in der Praxis des Öfteren Aktivitäten als kritisch ausgegeben, deren Begründung jedoch ausdrücklich darlegte, *warum sie nicht kritisch seien*. Das Modell erkannte die Unkritikalität also korrekt, hielt sich aber nicht strikt an die Vorgabe, ausschließlich `ids` kritischer Aktivitäten in die Antwort aufzunehmen. Als pragmatische Absicherung wurde daher das boolesche `isRelevant`-Flag eingeführt. Das LLM muss zusätzlich neben der Ausgabe der `ids` auch explizit angeben, ob die jeweilige Aktivität kritisch ist oder nicht. In der Summe reduziert diese Filterung die Anzahl widersprüchlicher Ausgaben.

`isRelevant` dient ausschließlich einer internen Validierung und wird in der finalen Ausgabe der Klassifizierungspipeline nicht berücksichtigt.

id-Validierung und -Vervollständigung

In der Praxis liefert das LLM mitunter unvollständige oder fehlerhafte `id`-Werte, die im Prozess nicht existieren. Zur Erhöhung der Robustheit werden die vom LLM ausgegebenen `ids` daher gegen die tatsächlich im Prozess vorhandenen Aktivitäts-`ids` geprüft und - wenn möglich - automatisch vervollständigt. Der Ablauf ist:

1. Ermittlung der Grundmenge aller gültigen Aktivitäts-`ids` aus der `BpmnElement`-Liste, die beim Preprocessing erstellt wurde.
2. Für jede vom LLM gelieferte `id` wird ein Präfix-Match gegen die gültigen `ids` durchgeführt. Ist die ausgegebene `id` *Präfix genau einer* gültigen `id`, wird sie durch diese vollständige `id` ersetzt.
3. Bleibt das Präfix-Match ohne eindeutiges Ergebnis, folgt ein Substring-Match: Ist die ausgegebene `id` *Teilstring genau einer* gültigen `id`, wird entsprechend vervollständigt.

4. Liefert weder Präfix- noch Substring-Match eine eindeutige Übereinstimmung, gilt die ausgegebene `id` als ungültig und wird aus der finalen Ausgabe entfernt.
5. Abschließend werden Duplikate entfernt, sodass jede kritische Aktivität höchstens einmal in der Ausgabe erscheint.

Gibt das LLM beispielsweise die `id Activity_1` aus, existiert im Prozess jedoch nur `Activity_12345`, wird die Ausgabe automatisch auf die korrekte `id` vervollständigt. Existieren hingegen sowohl `Activity_123` als auch `Activity_124` im Prozess, bleibt die Ausgabe unvollständig und wird entfernt, da keine eindeutige Zuordnung möglich ist.

Dieser Schritt fängt typische LLM-Ausgabefehler ab - etwa Halluzinationen oder abgeschnittene Bezeichner - und stellt die Konsistenz mit dem Eingabemodell sicher. Im Diagramm 4.1 ist er als „Validate and complete ids“ dargestellt. Ein fokussierter Code-Auszug findet sich in Listing 3.

Nach der Validierung besteht `BpmnAnalysisResult` nur noch aus Aktivitäten, die vom LLM als kritisch eingestuft wurden (`isRelevant = true`) und deren `ids` im Prozess existieren.

Da es nun möglich ist, BPMN-Prozesse vorzuverarbeiten, zu klassifizieren und die Ausgabe zu validieren und zu beheben, folgt als nächster Schritt die Definition einer Schnittstelle zum Aufruf der Pipeline. Das nächste Kapitel beschreibt dafür das API-Design.

4.4 API-Design

Dieses Kapitel beschreibt das API-Design der Klassifizierungspipeline, die zur Erkennung DSGVO-kritischer Elemente in BPMN-Modellen dient. Das Ziel ist es, eine standardisierte Schnittstelle zu definieren, die (1) die Einbindung in bestehende Werkzeuge und das Evaluationsframework vereinfacht, (2) die Austauschbarkeit unterschiedlicher Klassifizierungsalgorithmen - insbesondere im Evaluationsframework - ermöglicht, um verschiedene Ansätze der Klassifizierung vergleichen zu können, und (3) Erweiterbarkeit fördert, sodass zukünftige Arbeiten die Schnitt-

stelle wiederverwenden können, um ihre eigenen Klassifizierungsalgorithmen zu integrieren.

HTTP-Endpunkt

Die Klassifizierungspipeline ist über einen standardisierten HTTP-Endpunkt nutzbar, dessen Struktur und die klar definierten JSON-Schemas eine einfache Integration in bestehende Werkzeuge sowie das Evaluationsframework ermöglichen. Der POST-Endpunkt akzeptiert `multipart/form-data` mit den folgenden Teilen:

bpmnFile (Pflicht) Eine BPMN-2.0-XML-Datei (`.bpmn` oder `text/xml`), die den zu analysierenden Prozess beinhaltet.

llmProps (Optional) Ein JSON-Objekt zur Überschreibung von LLM-Eigenschaften zur Laufzeit. Siehe Listing 4.2 für das JSON-Schema. Wird nichts angegeben, nutzt die Pipeline Standardwerte.

Listing 4.2: JSON-Schema der `llmProps`.

```
1  {
2      "$schema": "https://json-schema.org/draft/2020-12/schema",
3      "title": "LlmProps",
4      "type": "object",
5      "properties": {
6          "baseUrl": {"type": "string"},
7          "modelName": { "type": "string" },
8          "apiKey": { "type": "string" },
9          "timeoutSeconds": { "type": "number" },
10         "seed": { "type": "number" },
11         "temperature": { "type": "number" },
12         "topP": { "type": "number" }
13     },
14     "required": []
15 }
```

Die `llmProps` erlauben das Überschreiben von LLM-Eigenschaften zur Laufzeit. Dadurch können unterschiedliche Modelle mit demselben Klassifizierungsalgorithmus flexibel getestet und verglichen werden, ohne die Anwendung neu starten zu

müssen. Dieses Design wurde gewählt, um die Experimente wie in Kapitel 3.4 beschrieben flexibel durchführen zu können.

Die Antwort des Endpunkts hat den Medientyp `application/json`. Sie enthält eine Liste der als DSGVO-kritisch klassifizierten Elemente. Für jedes Element können optional eine Begründung der Klassifikation sowie ein Elementname zur besseren Lesbarkeit angegeben werden. Zusätzlich kann die Anzahl der Versuche ausgegeben werden, die zur erfolgreichen Validierung der Antwort nötig waren (z. B. nach einem Schema-Validierungsfehler mit anschließendem Retry-Mechanismus). Diese Informationen nutzt das Evaluationsframework später, um die Robustheit der eingesetzten LLMs sichtbar zu machen. Das JSON-Schema der Antwort ist in Listing 4.3 dargestellt.

Listing 4.3: JSON-Schema der API-Antwort.

```
1 {
2     "$schema": "https://json-schema.org/draft/2020-12/schema",
3     "title": "BpmnAnalysisResult",
4     "type": "object",
5     "properties": {
6         "criticalElements": {
7             "type": "array",
8             "items": {
9                 "type": "object",
10                "properties": {
11                    "id": { "type": "string" },
12                    "name": { "type": "string" },
13                    "reason": { "type": "string" }
14                },
15                "required": ["id"]
16            }
17        },
18        "amountOfRetries": { "type": "number" }
19    },
20    "required": ["criticalElements"]
21 }
```

Im nächsten Kapitel wird die Webapp-Sandbox beschrieben, die als Beispielanwendung dient, um die Klassifizierungspipeline intuitiv nutzen zu können. Sie verwendet das hier beschriebene API, um die Klassifizierung durchzuführen.

4.5 Webapp-Sandbox

Zur interaktiven Nutzung der Klassifizierung wurde eine *Sandbox* in Form einer Webapp entwickelt. Sie verbindet einen vollwertigen BPMN-Editor auf Basis von BPMN.js [20] mit der in Kapitel 4.4 beschriebenen HTTP-Schnittstelle und macht die Analyse damit intuitiv bedienbar. In der Sandbox können BPMN-Modelle erstellt, verändert, exportiert und importiert sowie auf Datenschutzrelevanz analysiert werden. Als kritisch klassifizierte Aktivitäten werden nach der Analyse direkt im Editor farblich hervorgehoben, wie in Abbildung 4.2 zu sehen ist.

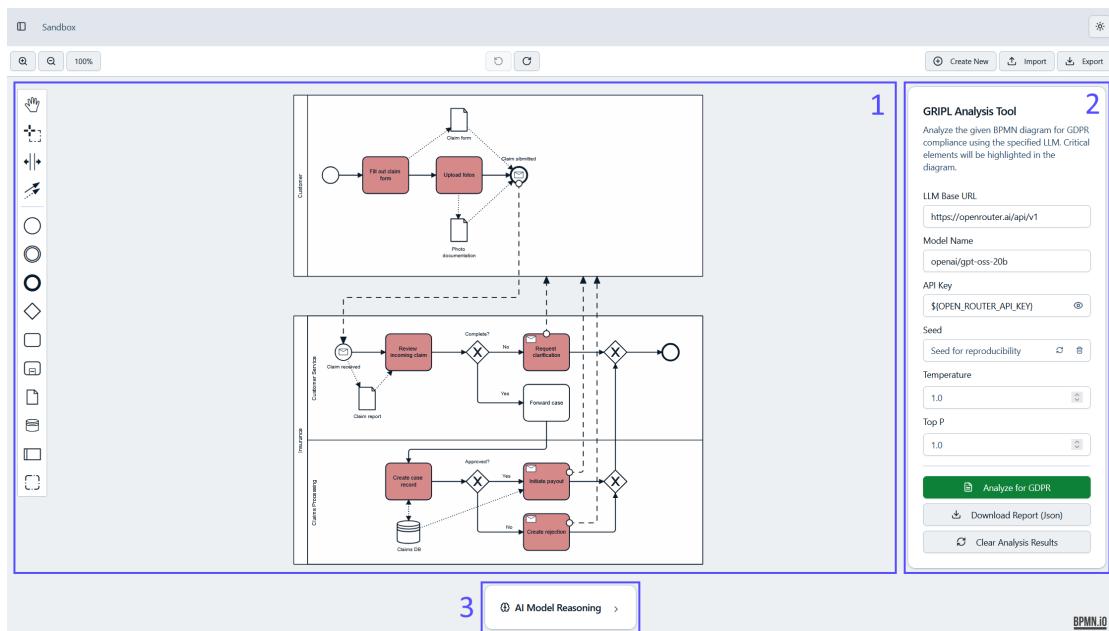
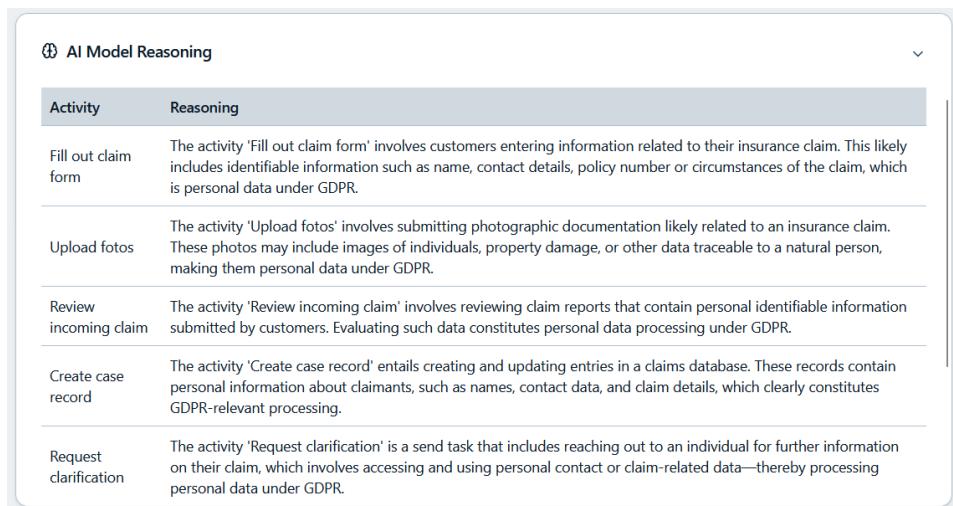


Abbildung 4.2: Sandbox im Frontend mit hervorgehobenen kritischen Aktivitäten nach Analyse.

Die Webapp richtet sich primär an (1) Forschende und Studierende, die die Klassifizierungspipeline explorativ testen möchten, (2) Praktiker aus BPM und Datenschutz (z. B. Prozessverantwortliche), die ein niedrigschwelliges Risiko-Screening einzel-

ner Modelle benötigen, und (3) interessierte Dritte, die die in dieser Arbeit beschriebenen Ergebnisse reproduzieren möchten. Die Webapp schließt die Lücke zwischen der reinen API und typischen Modellierungswerkzeugen. Sie erlaubt ein Prototyping direkt am Prozessmodell, vermeidet lokale Installationshürden und dient als leichtgewichtige Demonstrationsoberfläche für Live-Analysen und Modellvergleiche. Die in Abbildung 4.2 markierten Bereiche strukturieren die Bedienoberfläche:

1. **BPMN-Editor** (linke Hauptfläche): Vollwertiger Editor zum Modellieren, Importieren und Exportieren von BPMN-Prozessen. Nach einer Analyse werden als kritisch eingestufte Aktivitäten unmittelbar im Diagramm farblich markiert.
2. **Analyse-Panel** (rechte Seitenleiste): Konfiguration der LLM-Eigenschaften und Start der Analyse. Neben Modell, Basis-URL und API-Schlüssel können u. a. seed, temperature und topP gesetzt werden. Diese Parameter sind identisch zu den in Kapitel 4.4 beschriebenen `llmProps` und werden beim Starten der Analyse in die API-Anfrage überführt. Über Schaltflächen lassen sich die Ergebnisse als JSON herunterladen oder vorherige Markierungen löschen.
3. **LLM-Begründungen** (untere, aufklappbare Karte): Begründungen des LLM zu jeder als kritisch erkannten Aktivität. Die Karte kann ein- und ausgeklappt werden. Eine geöffnete Ansicht ist in Abbildung 4.3 dargestellt.



Activity	Reasoning
Fill out claim form	The activity 'Fill out claim form' involves customers entering information related to their insurance claim. This likely includes identifiable information such as name, contact details, policy number or circumstances of the claim, which is personal data under GDPR.
Upload fotos	The activity 'Upload fotos' involves submitting photographic documentation likely related to an insurance claim. These photos may include images of individuals, property damage, or other data traceable to a natural person, making them personal data under GDPR.
Review incoming claim	The activity 'Review incoming claim' involves reviewing claim reports that contain personal identifiable information submitted by customers. Evaluating such data constitutes personal data processing under GDPR.
Create case record	The activity 'Create case record' entails creating and updating entries in a claims database. These records contain personal information about claimants, such as names, contact data, and claim details, which clearly constitutes GDPR-relevant processing.
Request clarification	The activity 'Request clarification' is a send task that includes reaching out to an individual for further information on their claim, which involves accessing and using personal contact or claim-related data—thereby processing personal data under GDPR.

Abbildung 4.3: Exemplarische Begründungen der Klassifikation durch das LLM.

4 Design und Implementierung der Klassifizierungspipeline

Obwohl diese Arbeit auf Deutsch verfasst ist, verwendet die Webapp in der Benutzeroberfläche Englisch. Ebenso ist das LLM in der Klassifikationspipeline über den System-Prompt auf Englisch eingestellt, sodass die generierten Antworten auf Englisch sind. Dies erhöht die Wiederverwendbarkeit für eine internationale Nutzerschaft. Die BPMN-Modelle selbst können hingegen sowohl auf Deutsch als auch auf Englisch modelliert und analysiert werden.

5 Labeling und Datensätze

Für die Evaluation der Klassifikation ist es erforderlich, zunächst geeignete Testdatensätze mit Annotationen bereitzustellen. Ein Datensatz umfasst mehrere Testfälle, die jeweils aus einem BPMN-Prozessmodell bestehen. Standardisierte Datensätze schaffen einheitliche Prüfbedingungen und ermöglichen objektive Leistungsvergleiche. Die in dieser Arbeit verwendeten Testfälle decken ein breites Spektrum ab - von kundenorientierten Service- und Bestellprozessen (E-Commerce) über fachliche Abläufe im Versicherungs- und Gesundheitswesen bis hin zu technischen Szenarien (Smart-Home/IoT). Ergänzend sind typische betriebliche Querschnittsprozesse (z. B. Finanzen, Logistik, HR) sowie lehrnahe Prozesse aus universitären Übungsaufgaben enthalten. Darüber hinaus umfasst der Datensatz bewusst kleine, gezielt reduzierte Testfälle, um unterschiedliche Modellkomplexitäten und Randfälle abzubilden. Die Modelle liegen sowohl in deutscher als auch in englischer Sprache vor.

5.1 Labeling-Tool

Um die Erstellung und Verwaltung von gelabelten BPMN-Prozessmodellen zu erleichtern, wurde eine Webapp entwickelt. Mit dieser können BPMN-Testfälle erstellt, bearbeitet und Aktivitäten mit Labels versehen werden. Wichtige Funktionen des Labeling-Tools sind dabei: (1) das Anlegen und Verwalten von Datensätzen, (2) die Erstellung beliebig vieler Testfälle pro Datensatz, (3) die direkte Bearbeitung von BPMN-Modellen im Browser mittels BPMN.io [21], (4) ein Labeling-Modus, in dem Aktivitäten als DSGVO-kritisch markiert und optional mit einer Begründung versehen werden können, sowie (5) die persistente Speicherung der annotierten Testfälle in einer Datenbank zur späteren Nutzung im Evaluationsframework (siehe Kapitel 6).

5 Labeling und Datensätze

Abbildung 5.1 zeigt den Labeling-Editor im Labeling-Modus. Die in der Abbildung nummerierten Bereiche strukturieren die Oberfläche und das Zusammenspiel der Funktionen:

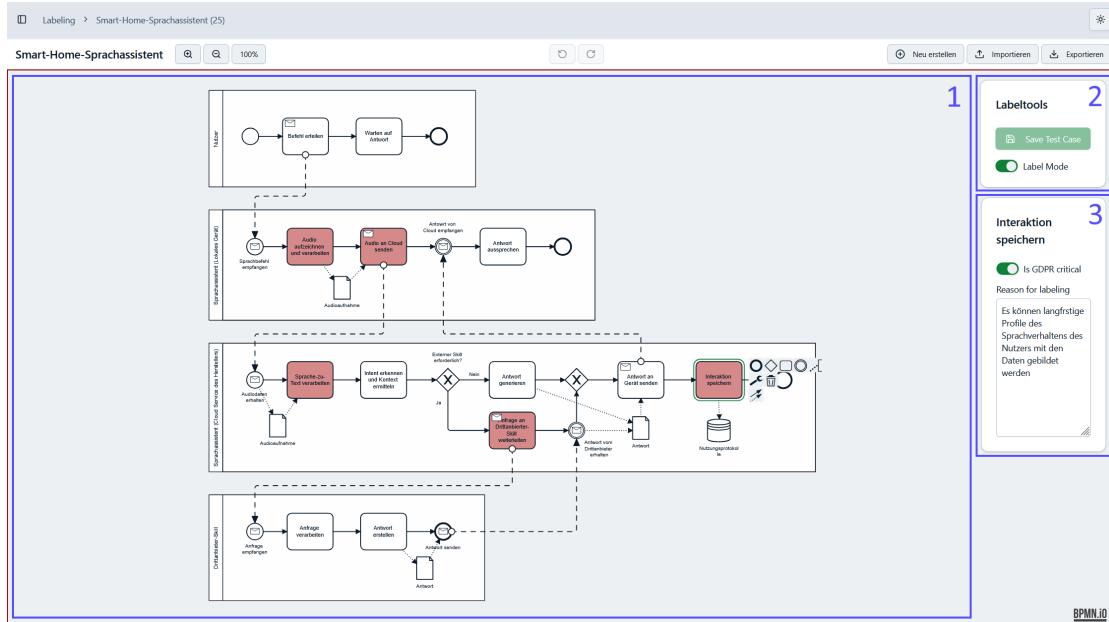


Abbildung 5.1: Labeling-Editor im Labeling-Modus mit exemplarischem Modell.

1. **BPMN-Editor** (linke Hauptfläche): Hier werden Prozessmodelle erstellt, importiert, bearbeitet und angezeigt. Im Labeling-Modus ist die Modellierung bewusst gesperrt. Die Elemente können dann nur ausgewählt werden, um sie zu labeln. Als kritisch gelabelte Aktivitäten werden im Modell farblich hervorgehoben und sind damit sofort visuell erkennbar.
2. **Label-Tools** (rechte Seitenleiste oben): Über dieses Panel wird zwischen Editier- und Labeling-Modus gewechselt und ein Testfall gespeichert.
3. **Label-Panel der Aktivität** (rechte Seitenleiste unten): Ist eine Aktivität ausgewählt, kann sie hier als „DSGVO-kritisch“ markiert werden. Zusätzlich lässt sich optional eine natürlichsprachige Begründung hinterlegen. Diese Begründung dient ausschließlich der Dokumentation und Nachvollziehbarkeit und wird nicht in der Evaluierung berücksichtigt.

In der Übersicht der Datensätze aus Abbildung 5.2 sind alle angelegten Datensätze und zugehörigen Testfälle aufgelistet. Von hier aus können neue Datensätze und Testfälle erstellt sowie bestehende bearbeitet werden.

5 Labeling und Datensätze

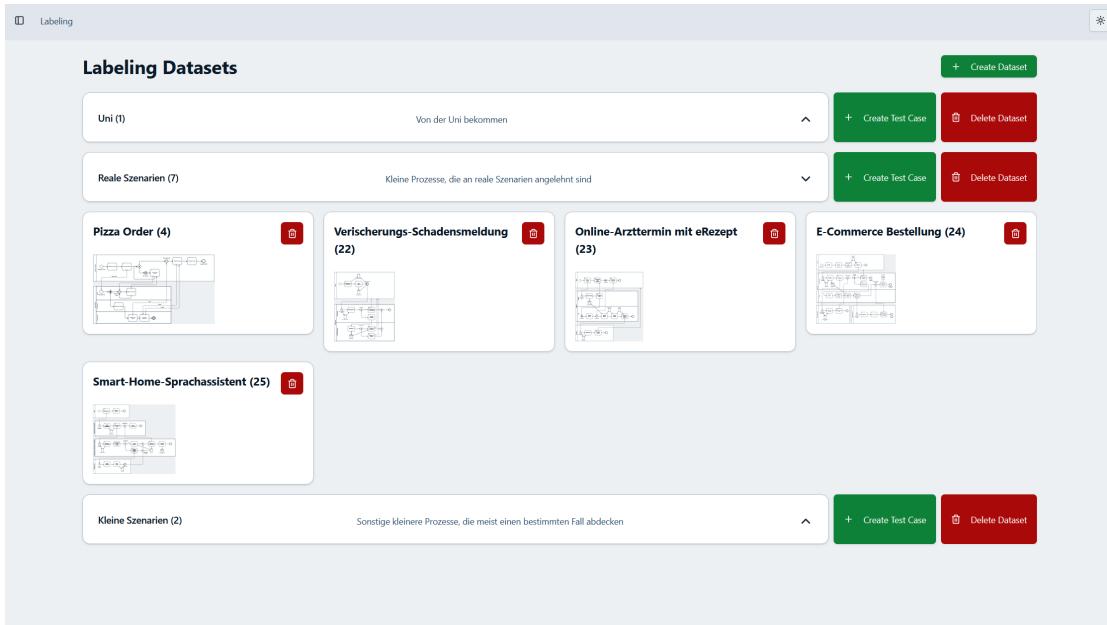


Abbildung 5.2: Übersicht der Datensätze im Labeling-Tool.

5.2 Quellen und Eigenschaften der Datensätze

Für die Evaluation wurden drei Gruppen von BPMN-Datensätzen eingesetzt:

1. Prozesse, die von der Universität Ulm bereitgestellt wurden (z. B. Lehrbeispiele aus Übungsaufgaben).
2. Mittelgroße Praxisbeispiele aus verschiedenen Domänen. Diese Prozesse beinhalten Elemente wie Pools, Lanes, Datenobjekte und Gateways.
3. Kleine, reduzierte Testfälle mit maximal fünf Aktivitäten und wenigen weiteren Elementen (z. B. einfacher Sequenzfluss ohne Pools).

Die Praxisbeispiele (Gruppe 2) decken typische *geschäftliche Domänen* ab, darunter kundenorientierte Service- und Bestellprozesse (E-Commerce), fachliche Abläufe im Versicherungs- und Gesundheitswesen sowie technische Smart-Home/IoT-Kontexte. Die universitären Beispiele (Gruppe 1) stammen aus lehrnahen Übungsaufgaben und stellen Domänen wie Finanzen, Logistik und HR dar. Die kleinen Testfälle (Gruppe 3) sind bewusst minimal gehalten, um Randfälle und unterschiedliche Modellkomplexitäten abzudecken.

Diese bewusst heterogene Auswahl über Domänen, Sprachen und Komplexitätsstufen erhöht die Aussagekraft der Evaluation. In der Literatur wird betont, dass eine erhöhte Datensatzvielfalt die Robustheit der Bewertung steigert und einseitige Ergebnisse vermeidet [9]. Tabelle 5.1 zeigt die Eckdaten der Datensätze.

Tabelle 5.1: Eckdaten der verwendeten Datensätze.

	Uni-Prozesse	Praxisbeispiele	Kleine Testfälle
Testfälle gesamt	5	5	14
Testfälle (DE)	0	3	15
Testfälle (EN)	5	2	0
$\bar{\Omega}$ Aktivitäten \pm SD ¹	$13,4 \pm 2,6$	$11,6 \pm 4,2$	$3,9 \pm 1,4$
$\bar{\Omega}$ Aktivitäten (kritisch) \pm SD	$8,6 \pm 3,6$	$6,6 \pm 1,9$	$2,1 \pm 1,5$
$\bar{\Omega}$ Datenobjekte \pm SD	$1,4 \pm 1,9$	$3,6 \pm 2,1$	$0,4 \pm 0,7$
$\bar{\Omega}$ Datenassoziationen \pm SD	$2,4 \pm 3,3$	7 ± 4	$0,7 \pm 1,2$
$\bar{\Omega}$ Ereignisse \pm SD	$21 \pm 13,8$	$8,2 \pm 2,8$	2 ± 0
$\bar{\Omega}$ Gateways \pm SD	$13 \pm 7,6$	$1,8 \pm 1,5$	0 ± 0
$\bar{\Omega}$ Pools \pm SD	$3,4 \pm 1,1$	3 ± 1	$0,4 \pm 0,6$
$\bar{\Omega}$ Lanes \pm SD ²	3 ± 1	$4 \pm 0,7$	$0,3 \pm 0,5$
$\bar{\Omega}$ Nachrichtenflüsse \pm SD	$9,4 \pm 5,3$	$5,2 \pm 0,8$	$0,1 \pm 0,3$
$\bar{\Omega}$ Annotationen \pm SD	$1 \pm 1,7$	0 ± 0	0 ± 0

¹ SD = Standardabweichung s der jeweiligen Anzahl pro Testfall.

² Blackbox-Pools ohne Lanes wurden nicht mitgezählt, daher kann der Durchschnittswert der Lanes geringer ausfallen als der, der Pools.

5.3 Labeling-Guide

Nachfolgend wird beschrieben, nach welchen Richtlinien die Daten für die Klassifizierung DSGVO-kritischer Aktivitäten gelabelt wurden.

Die Aktivitäten in den Testfällen sollen mit dem Label „kritisch“ versehen werden, wenn sie potenziell personenbezogene Daten verarbeiten und somit im Sinne der DSGVO relevant sein könnten. Die wichtigsten Begriffe der DSGVO wurden bereits in Abschnitt 2.1 definiert.

Beim Labeln einer Aktivität können Grenzfälle auftreten - etwa, wenn kein Datenobjekt vorhanden ist, der Name aber auf Datenverarbeitung hindeutet (z. B. „Verträge

5 Labeling und Datensätze

archivieren“). Solche Verträge können personenbezogen sein (z. B. Arbeitsverträge) oder rein geschäftlich zwischen Unternehmen. In diesen Fällen wird zunächst der Kontext geprüft: Gibt es Hinweise auf personenbezogene Daten, z. B. über Pools/Lanes oder angrenzende Aktivitäten im Prozess? Fehlen eindeutige Hinweise, wird die Aktivität als unkritisch gelabelt. Deutet der Kontext hingegen auf die Verarbeitung personenbezogener Daten hin, z. B. durch einen Prozessnamen wie „Mitarbeiterverwaltung“ oder vorangehende Aktivitäten wie „Mitarbeiterdaten erfassen“, erhält die Aktivität das Label kritisch. Im Zweifel wird kritisch gelabelt, um eine höhere Sensitivität zu gewährleisten.

Tabelle 5.2 listet beispielhaft einige Aktivitäten mit ihrer Klassifikation und der zugehörigen Begründung auf.

Tabelle 5.2: Beispielhafte Aktivitäten und Label.

Aktivität	Kritisch?	Kommentar
Lieferadresse eingeben	Ja	Name, Anschrift des Kunden werden aufgenommen.
Rückfrage an Kunden senden	Ja	Kontaktinformationen werden verwendet.
Fall anlegen	Ja	Aktivität befindet sich im Kundenservice-Kontext, personenbezogene Daten wahrscheinlich.
Sprache zu Text verarbeiten	Ja	Im Kontext eines Sprachassistenten werden biometrische Daten des Nutzers verarbeitet.
Produkt versenden	Nein*	Logistik und Sachvorgänge sind nicht per se datenschutzkritisch, solange keine neue Datenverarbeitung, wie ein Labeldruck stattfindet.
Systemprotokoll auslesen	Ja	Im Kontext einer technischen Wartung können personenbezogene Daten (z. B. Nutzer-ids) enthalten sein.
Logdaten archivieren (anonym)	Nein	Keine personenbezogenen Daten enthalten.
Gerät kalibrieren	Nein	Im Kontext einer technischen Wartung werden keine personenbezogenen Daten verarbeitet.

6 Evaluationsframework

Nachdem nun Daten gelabelt werden können und der Testdatensatz für diese Arbeit erstellt wurde, wird in diesem Kapitel das Evaluationsframework vorgestellt. Das Framework nutzt die in Kapitel 4 entwickelte Klassifizierungspipeline, um verschiedene LLMs anhand gelabelter Testdaten systematisch, reproduzierbar und transparent zu vergleichen. Leitendes Gestaltungsprinzip ist die Entkopplung: Modelle, Klassifizierungsendpunkte und Testdaten werden zur Laufzeit über eine deklarative Konfiguration und eine standardisierte HTTP-Schnittstelle, siehe Abschnitt 4.4, angebunden. Dadurch sind sie austauschbar und erweiterbar, ohne Codeänderungen vornehmen zu müssen, zum Beispiel durch die Einbindung eines neuen Modells mit Endpunkt, Modellname und Parametern wie temperature oder topP sowie durch zusätzliche Datensätze. So wird ein fairer und reproduzierbarer Vergleich unterschiedlicher Modelle unter identischen Rahmenbedingungen ermöglicht.

6.1 Use-Cases und Anforderungen

Das Evaluationsframework richtet sich an Forschende und Entwickler, die LLMs und Klassifizierungsalgorithmen für die Identifikation DSGVO-kritischer BPMN-Aktivitäten auswerten und miteinander vergleichen möchten. Es bietet eine einheitliche Ausführungs- und Auswertungsumgebung mit klar definierten Schnittstellen und standardisierten Berichten. In diesem Kapitel werden die Use-Cases und funktionalen Anforderungen des Evaluationsframeworks beschrieben.

Use-Cases

Die wichtigsten Anwendungsfälle des Evaluationsframeworks sind:

- **Benchmarking von LLMs.** Systematischer Vergleich mehrerer LLMs auf denselben Datensätzen, mit identischem Algorithmus und identischen Parametern.
- **A/B-Vergleich von Algorithmen.** Gegenüberstellung verschiedener Klassifizierungspipelines, mit z. B. alternativen Prompts oder anderem Preprocessing, über eine standardisierte HTTP-Schnittstelle, die in Kapitel 4.4 definiert ist.
- **Explorative Analyse.** Detaillierte Einsicht pro Modell und Testfall (inklusive Begründungen und Visualisierungen), um Fehlklassifikationen gezielt zu untersuchen.
- **Berichterstellung.** Die Ergebnisse lassen sich als JSON oder Markdown exportieren und später wieder importieren, um sie erneut untersuchen zu können. Sie eignen sich zudem für die Publikation. Die Diagramme werden automatisch erzeugt und stehen ebenfalls zum Download bereit.

In dieser Arbeit werden keine A/B-Vergleiche unterschiedlicher Klassifizierungsalgorithmen durchgeführt, sondern lediglich verschiedene LLMs mit demselben Algorithmus verglichen. Dies ist eine bewusste Einschränkung des Untersuchungsrahmens, um die Analyse auf die Leistungsfähigkeit der LLMs zu fokussieren. Die in 4.4 definierte Schnittstelle erlaubt es jedoch, in zukünftigen Arbeiten alternative Klassifizierungsalgorithmen mit geringem Aufwand einzubinden.

Funktionale Anforderungen

In der folgenden Tabelle sind die funktionalen Anforderungen an das Evaluationsframework aufgelistet, die notwendig sind, um die definierten Use-Cases zu erfüllen:

FA01 — Nutzen gelabelter Testdatensätze

Beschreibung:	Das Framework kann die gelabelten Testdatensätze benutzen, die mit dem Labeling-Tool aus 5.1 erstellt worden sind.
Abhängigkeiten:	-

FA02 — Vergleichbarkeit von Modellen und Algorithmen

Beschreibung: Das Framework erlaubt den direkten Vergleich verschiedener LLMs sowie unterschiedlicher Klassifizierungsalgorithmen anhand gelabelter Testdaten. Die Anbindung an Klassifizierungsalgorithmen erfolgt über die in Kapitel 4.4 definierte, standardisierte HTTP-Schnittstelle.

Abhängigkeiten: FA01

FA03 — Deklarative Konfiguration

Beschreibung: Ein Evaluationslauf ist vollständig über eine YAML-Datei konfigurierbar. Dazu zählen Modelle, Klassifizierungsendpunkte, Testdatensätze und Seed. Experimente werden dadurch portabel und wiederholbar.

Abhängigkeiten: FA02

FA04 — Detaillierte Ergebnisaufbereitung

Beschreibung: Das Framework gibt Ergebnisse auf zwei Ebenen aus.

1. Pro Testfall und pro Modell: Status („bestanden“/„nicht bestanden“), klassifizierte Elemente mit Begründungen, TP/FP/FN/TN und eine Visualisierung der Klassifikation im BPMN-Prozess.
2. Pro Modell als Summe über alle Testfälle: Accuracy, Precision, Recall, F1-Score und die Konfusionsmatrix.

Zusätzlich protokolliert das Framework Metadaten der Evaluation, z. B. Endpunkt, verwendete Modelle und den Seed.

Abhängigkeiten: FA02

☰ FA05 — Frontend

Beschreibung: Für eine einfache Bedienung und Ansicht der Ergebnisse bietet das Evaluationsframework ein Frontend an.

Abhängigkeiten: FA02, FA03, FA04

☰ FA06 — Visualisierung und Berichte der Gesamtresultate

Beschreibung: Kennzahlen werden als Side-by-Side-Diagramme und tabellarisch dargestellt. Zusätzlich stehen Export/Import der Ergebnisse als JSON sowie ein Markdown-Report zur Verfügung.

Abhängigkeiten: FA05

6.2 Konfiguration einer Evaluierung

Die funktionale Anforderung FA03 fordert, dass Evaluationsläufe deklarativ konfiguriert werden können. Das Framework unterstützt dies auf zwei Wegen: Erstens bietet die Weboberfläche, die in 6.5 gezeigt wird, die Möglichkeit, Evaluationsläufe interaktiv zu konfigurieren und zu starten. Zweitens lässt sich eine Evaluierung über eine YAML-Datei beschreiben, die entweder in der Weboberfläche hochgeladen oder per CLI an das Evaluationsframework übergeben wird. Auf diese Weise werden Reproduzierbarkeit und Versionierung der Evaluationsläufe sichergestellt. Listing 6.1 zeigt ein Beispiel für eine solche YAML-Konfiguration. Ein ausführliches JSON-Schema ist im Anhang (Listing 4) zu finden.

Die Evaluierungskonfiguration umfasst die folgenden Bausteine:

- `defaultEvaluationEndpoint` ist der Standardendpunkt für die Klassifizierung. Er wird verwendet, wenn für ein Modell kein eigener Endpunkt angegeben ist. Der Endpunkt muss die in Kapitel 4.4 beschriebene API-Spezifikation erfüllen und kann relativ (gegen die Basis-URL des Evaluationsframeworks) oder absolut (für einen externen Dienst) angegeben werden.

Listing 6.1: Beispiel einer Evaluierungskonfiguration in YAML.

```

1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering
2 maxConcurrent: 10
3 repetitions: 3
4 seed: 42
5 models:
6   - label: Mistral Medium 3.1
7     llmProps:
8       baseUrl: https://openrouter.ai/api/v1
9       modelName: mistralai/mistral-medium-3.1
10      apiKey: ${OPEN_ROUTER_API_KEY}
11      topP: 1
12   - label: Deepseek Chat v3.1
13     llmProps:
14       baseUrl: https://openrouter.ai/api/v1
15       modelName: deepseek/deepseek-chat-v3.1
16       apiKey: ${OPEN_ROUTER_API_KEY}
17       temperature: 0.1
18   - label: GPT oss 120b
19     llmProps:
20       baseUrl: https://openrouter.ai/api/v1
21       modelName: openai/gpt-oss-120b
22       apiKey: ${OPEN_ROUTER_API_KEY}
23 datasets:
24   - 2
25   - 7

```

- `maxConcurrent` gibt die maximale Anzahl parallel auszuführender Testfälle an. So lassen sich beispielsweise *Rate Limits*¹ der angebundenen LLMs einhalten, um technische Fehler in den Ergebnissen zu vermeiden.
- `repetitions` bestimmt, wie oft die Evaluierung pro Modell wiederholt wird. Die Ergebnisse werden später über alle Wiederholungen aggregiert (siehe Abschnitt 6.4).

¹Providerseitige Begrenzungen, etwa „Requests pro Minute“ oder maximale Parallelität. Bei Überschreitung antworten viele Anbieter mit HTTP 429 („Too Many Requests“). Zudem drohen strengere Drosselungen.

- seed legt einen Startwert (Seed) für reproduzierbare Evaluationsläufe fest. Auf Basis des Seeds und der Wiederholungsnummer wird für jede Wiederholung deterministisch ein eigener neuer Seed generiert, um unterschiedliche, aber reproduzierbare Ergebnisse zu erzielen. Er wird bei jedem Modell an die `llmProps` weitergereicht und bei der Kommunikation mit den LLMs verwendet, sofern diese einen Seed unterstützen.
- models enthält die zu evaluierenden Modelle. Jedes Modell besitzt ein label zur Identifikation und optional spezifische `llmProps`, um die Eigenschaften des verwendeten LLMs zu definieren. Diese sind identisch zu den in Kapitel 4.4 beschriebenen `llmProps`.
- datasets ist eine Liste von Datensatz-ids, die jeweils eine Menge von Testfällen beinhalten.

Wie im Schema in Listing 4 gezeigt, kann jedem Modell optional ein eigener `evaluationEndpoint` zugewiesen werden, der den in `defaultEvaluationEndpoint` definierten Standard überschreibt. Dadurch lassen sich unterschiedliche Klassifizierungsalgorithmen oder -versionen gezielt pro Modell vergleichen. Ist kein spezifischer Endpunkt angegeben, greift automatisch der Standardendpunkt.

API-Keys in den `llmProps` können optional als Umgebungsvariablen referenziert werden, wie im Beispiel in Listing 6.1 gezeigt. So lassen sich sensible Daten sicher handhaben, ohne sie direkt in der Konfigurationsdatei zu speichern. Die Umgebungsvariablen werden zur Laufzeit aufgelöst und müssen daher im Kontext der Anwendung verfügbar sein.

6.3 Architektur und Komponenten

Das Evaluationsframework ist modular aufgebaut und nutzt eine Pipeline-Architektur, um eine flexible und skalierbare Evaluierung zu ermöglichen - wie in FA02 gefordert. Die Architektur ist in Abbildung 6.1 dargestellt. Sie besteht aus mehreren Hauptkomponenten, die jeweils eine klar definierte Aufgabe erfüllen. Im Folgenden werden die Komponenten und ihr Zusammenspiel beschrieben.

Das Framework bietet zwei Einstiegspunkte zur Ausführung einer Evaluierung: Eine Evaluation kann mittels `EvaluationController` über einen HTTP-Request

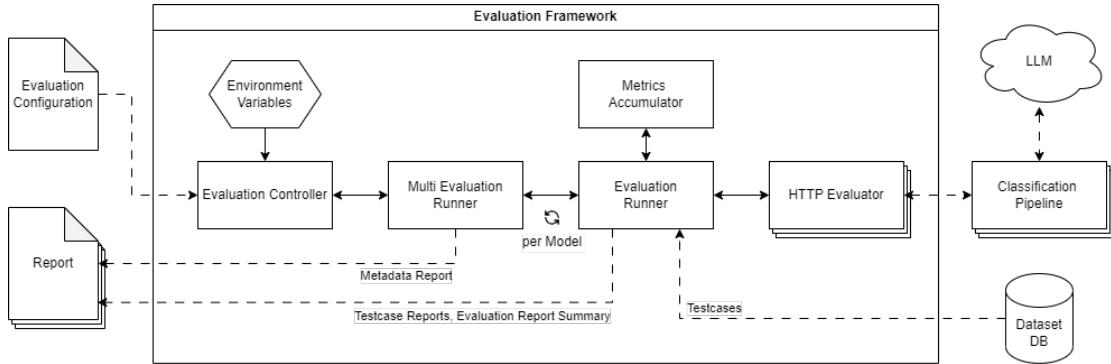


Abbildung 6.1: Architektur des Evaluationsframeworks.

oder über die Kommandozeile mit dem `EvaluationCommand` gestartet werden. Beide Einstiegspunkte akzeptieren die Konfiguration aus Kapitel 6.2, lösen ggf. Umgebungsvariablen auf und delegieren die Ausführung der Evaluation an den `MultiEvaluationRunner`.

Die Architektur trennt Zuständigkeiten strikt: Der `MultiEvaluationRunner` koordiniert die Modellläufe und steuert die Anzahl der *Wiederholungen* pro Modell. `EvaluationRunner` verarbeitet die einzelnen Testfälle innerhalb einer Wiederholung und sammelt Metriken. `HttpEvaluator` kommuniziert mit der Klassifizierungs-pipeline und der `Metrics Accumulator` aggregiert Ergebnisse Thread-sicher pro Modell über mehrere Testfälle.

6.4 Evaluationsergebnisse

Die im Folgenden beschriebenen Resultate werden während der Evaluierung laufend erzeugt und an das Frontend gestreamt. Anwender können damit sowohl Zwischenstände verfolgen als auch nach Abschluss detaillierte Analysen durchführen.

Für jeden Testfall eines Modells liegen vor: die von der Klassifizierungspipeline klassifizierten Aktivitäten mit optionalen Begründungen, die gelabelten erwarteten Aktivitäten, die Zählwerte für *TP*, *FP*, *FN* und *TN* sowie eine Bild-URL zur Visualisierung des BPMN-Modells mit hervorgehobenen Aktivitäten. Aus diesen Informationen lässt sich ableiten, ob der Testfall erfolgreich war. Ein Testfall gilt als erfolgreich, wenn die klassifizierten Aktivitäten exakt den erwarteten Aktivitäten entsprechen.

Technische Probleme, die während der Klassifizierung auftreten, werden ebenfalls erfasst, z. B. Parsing-Fehler, ungültiges BPMN, Token-Limit-Überschreitungen oder Zeitüberschreitungen.

Auf Modellebene stehen die Gesamtergebnisse über alle Testfälle zur Verfügung. Dazu gehören die aggregierten Kennzahlen *Precision*, *Accuracy*, *Recall* und *F1-Score* sowie eine Konfusionsmatrix mit den Gesamtwerten für *TP*, *FP*, *FN* und *TN*. Zusätzlich sind die Anzahlen der korrekt bzw. falsch klassifizierten sowie der technisch fehlgeschlagenen Testfälle aufgeführt.

Die Ergebnisse eines Modells über alle Testfälle werden zudem auch über alle Wiederholungen aggregiert. Das Framework berechnet pro Kennzahl den Mittelwert und die Standardabweichung. Dadurch lassen sich zufallsbedingte Schwankungen abfedern und robustere Aussagen treffen.

Abschließend sind die Metadaten der gesamten Evaluierung verfügbar. Dazu zählen die verwendeten Testdatensätze und Anzahl der Testfälle, die konfigurierten Modelle samt ihrer relevanten Parameter (u. a. Modellname, temperature, topP, ggf. eigener Endpunkt), der für die Reproduzierbarkeit verwendete Seed sowie ein Zeitstempel der Evaluierung. Zum unmittelbaren Vergleich werden die aggregierten Kennzahlen aller Modelle nebeneinander dargestellt. Alle Ergebnisse können über ein webbasiertes Frontend, das im nächsten Abschnitt beschrieben wird, eingesehen und im Detail analysiert werden.

6.5 Frontend

Das Frontend des Evaluationsframeworks setzt die Anforderungen FA05 und FA06 um. Es unterstützt die interaktive Konfiguration von Evaluierungen, die Live-Verfolgung des Fortschritts sowie die detaillierte Analyse der Ergebnisse bis auf Ebene einzelner Testfälle. Die Oberfläche ist so gestaltet, dass zentrale Kennzahlen wie *Accuracy*, *Precision*, *Recall* und *F1-Score*, die Konfusionsmatrix mit *TP*, *FP*, *TN*, *FN* sowie die Bestehensraten aller Modelle zunächst auf einen Blick erfasst und anschließend schrittweise vertieft werden können.

Konfigurationsansicht

Abbildung 6.2 zeigt das Formular zur Konfiguration einer Evaluierung. Sämtliche Parameter, die bereits aus der YAML-Konfiguration in Kapitel 6.2 bekannt sind, lassen sich hier setzen. Verfügbare Standardwerte, zum Beispiel der Endpunkt der in dieser Arbeit verwendeten Klassifizierungspipeline oder die in der Datenbank verfügbaren Datensätze, werden automatisch geladen.

The screenshot displays the configuration interface for an evaluation. It is divided into three main sections:

- Default Settings:** Contains fields for the default evaluation endpoint (using a preset), a preset endpoint (Preprocessing & Prompt Engineering Analysis), max concurrent LLM requests (10), number of repetitions (5), seed (24523833), and a note about seeds.
- Datasets:** A section titled "Select Datasets" showing three datasets: Kleine Szenarien, Reale Szenarien, and Uni.
- Models Configuration:** A table listing two models:

Model	Effective endpoint	Actions
Model 1	/gdpr/analysis/prompt-engineering	+ Add Model Duplicate Remove
Model 2	/gdpr/analysis/prompt-engineering	+ Add Model Duplicate Remove

 Each model row contains fields for Label (Gemma-3-128-it and Gemma-3-27b-it), API Key (\$OPEN_ROUTER_API_KEY), Endpoint (using default), Temperature (0,1), Top P (1), LLM Base URL (https://openrouter.ai/api/v1), LLM Model Name (google/gemma-3-12b-it and google/gemma-3-27b-it), and LLM Response Timeout (seconds) (240).

At the bottom, there are buttons for Download Markdown Report, Download JSON Report, Upload JSON Report, and Start Evaluation.

Abbildung 6.2: Formular zur Konfiguration einer Evaluation.

6 Evaluationsframework

YAML-Konfigurationen können importiert und exportiert werden, um sie zu speichern oder weiterzugeben. Unter dem Formular befinden sich Schaltflächen zum Starten der Evaluierung sowie zum Import und Export von JSON-Berichten. Auf diese Weise lassen sich Ergebnisse archivieren und später erneut laden, ohne die Evaluierung erneut ausführen zu müssen.

Gesamtübersicht

Nach dem Start der Evaluierung werden die Ergebnisse pro Modell inkrementell vom Backend übermittelt, im Frontend verarbeitet und in einer Gesamtübersicht wie in Abbildung 6.3 angezeigt. Dadurch können Teilergebnisse bereits untersucht werden, während die Evaluierung noch läuft. Die Gesamtübersicht bietet eine kompakte Zusammenfassung der Metadaten und der Kennzahlen aller Modelle über sämtliche Wiederholungen. So lassen sich die Modelle direkt miteinander vergleichen.

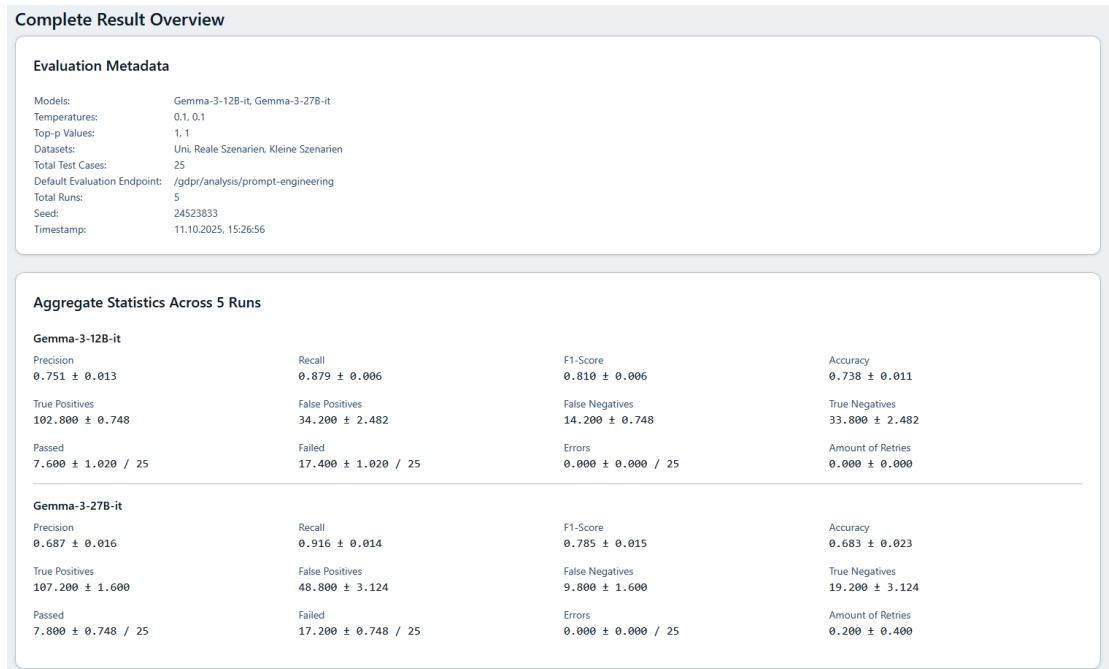


Abbildung 6.3: Gesamtübersicht einer Evaluierung mit aggregierten Metriken über alle Wiederholungen.

Ergebnisse pro Wiederholung

Die Übersicht der Ergebnisse pro Wiederholung ist in Abbildung 6.4 dargestellt. Sie ermöglicht den Vergleich der Modelle innerhalb eines konkreten Laufs. So lassen sich die Kennzahlen einzelner Wiederholungen untersuchen und die Streuung der Ergebnisse zwischen den Läufen beurteilen. Zusätzlich werden die Anzahl der technisch fehlgeschlagenen Testfälle sowie die aufgetretenen Retries bei der Kommunikation mit dem LLM angezeigt, um die Robustheit der Modelle bewerten zu können.

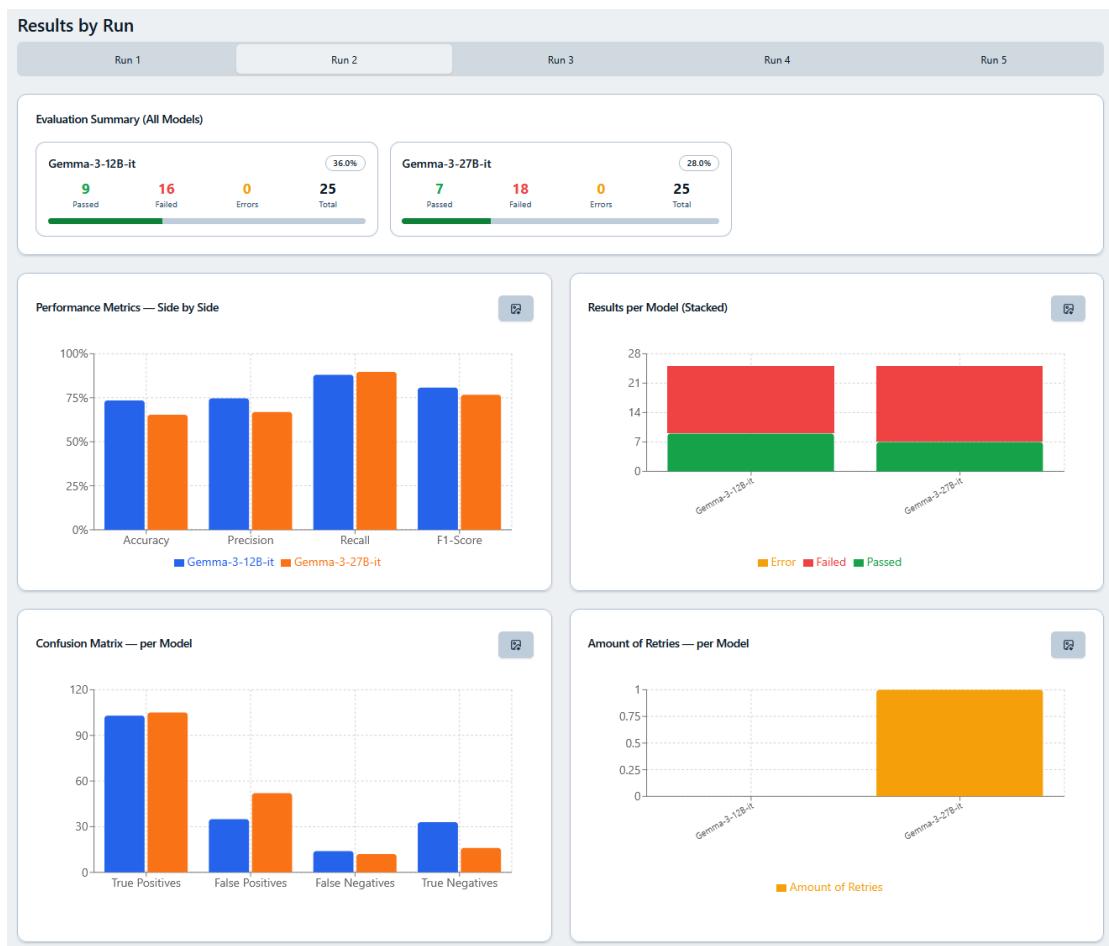


Abbildung 6.4: Ergebnisse pro Wiederholung mit exemplarischen Resultaten.

Ergebnisse pro Modell

Für eine vertiefte Analyse stellt das Frontend für jedes Modell eine Detailansicht bereit, die alle Kennzahlen über sämtliche Testfälle einer Wiederholung aggregiert. Abbildung 6.5 zeigt diese Ansicht. Über Tabs kann zwischen den Modellen und Wiederholungen gewechselt werden, was einen schnellen Vergleich unterschiedlicher Modelle oder Läufe ermöglicht.

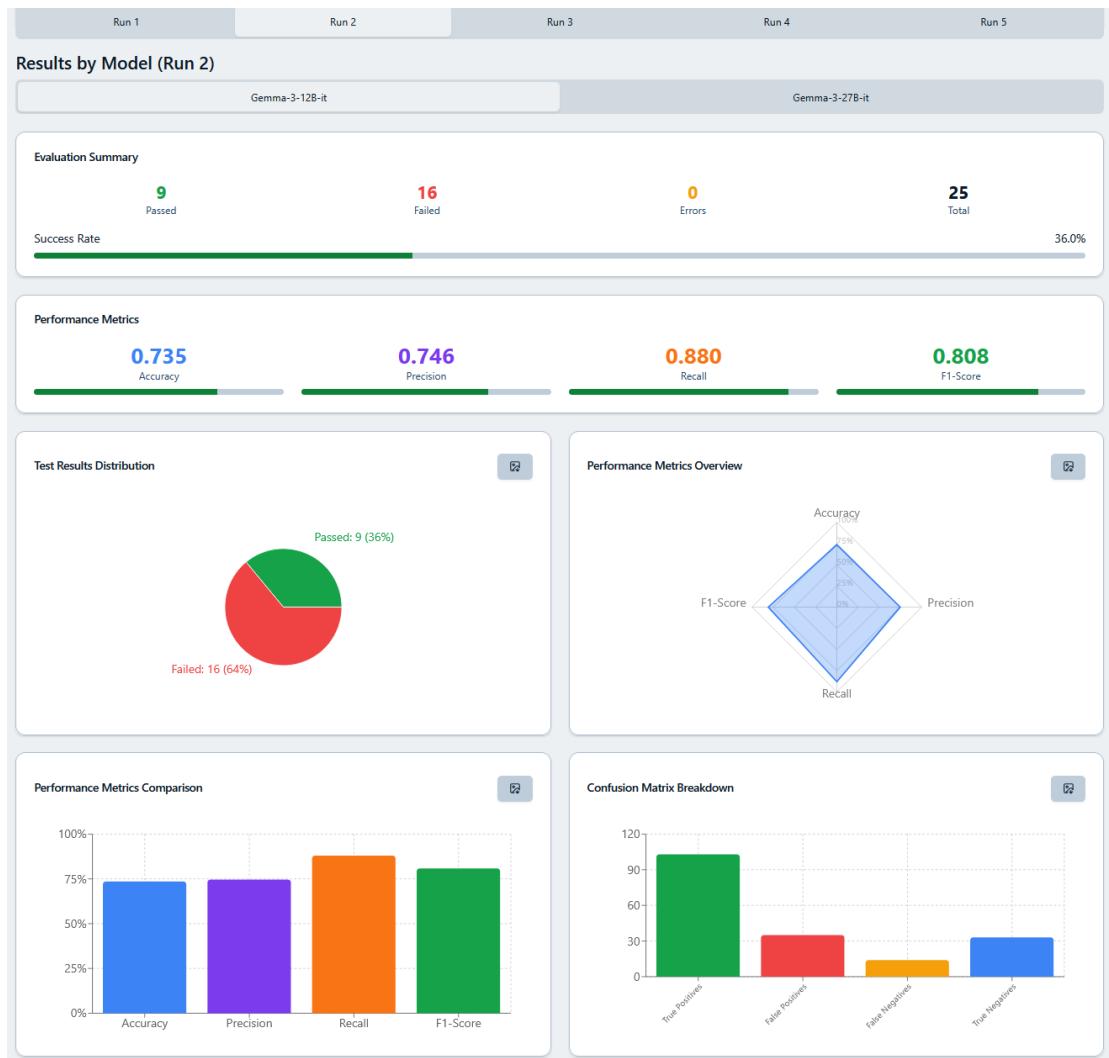


Abbildung 6.5: Modell-Detailansicht mit exemplarischen Ergebnissen.

Ergebnisse pro Testfall

Neben den aggregierten Ergebnissen pro Modell lassen sich auch die Resultate einzelner Testfälle je Modell untersuchen. Abbildung 6.6 zeigt die Detailseite eines Testfalls. Sie enthält unter anderem den Status, die erwarteten gelabelten Aktivitäten und die vom Modell detektierten Aktivitäten. Zusätzlich visualisiert eine BPMN-Darstellung den Prozess, und die Aktivitäten sind je nach korrekter oder inkorrekt Klassifizierung farblich markiert. Falls vorhanden, wird außerdem die vom LLM gelieferte Begründung pro Aktivität angezeigt.

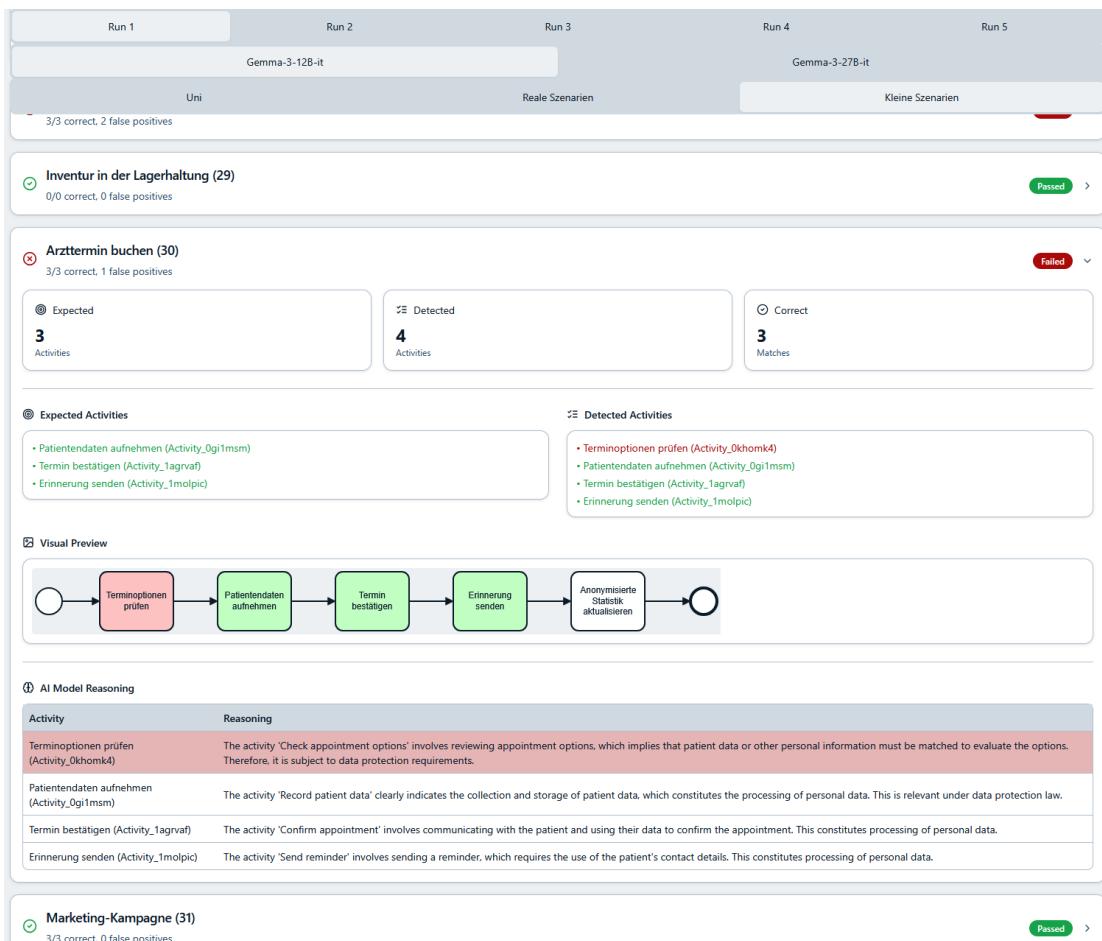


Abbildung 6.6: Detailseite eines Testfalls mit exemplarischen Ergebnissen.

Abweichungen werden dadurch unmittelbar sichtbar, und typische Fehlmuster wie systematische FP bei bestimmten Aktivitätstypen lassen sich schnell erkennen.

6 Evaluationsframework

Testfälle, die aufgrund technischer Fehler nicht klassifiziert werden konnten, werden mit der entsprechenden Fehlermeldung aufgeführt. Über die Tabs am oberen Rand kann zwischen den Modellen, Wiederholungen und Testdatensätzen gewechselt werden, um verschiedene Perspektiven auf die Ergebnisse der Testfälle zu erhalten.

7 Modellauswahl

Aufbauend auf dem im letzten Kapitel vorgestellten Evaluationsframework werden in diesem Kapitel die für die Klassifizierungsaufgabe zu vergleichenden LLMs vorgestellt. Um die Auswahl nachvollziehbar zu machen und künftige Arbeiten mit ähnlicher Zielsetzung zu unterstützen, werden zunächst die Auswahlkriterien der Modelle erläutert. Anschließend folgt eine Vorstellung der ausgewählten Modelle.

7.1 Kriterien

Dieser Abschnitt legt die Auswahlkriterien der LLMs offen, nach denen die Modelle ausgewählt und kategorisiert wurden. Tabelle 7.1 zeigt eine Übersicht der Kriterien. Diese Kriterien helfen, die Modelle systematisch zu vergleichen und ihre Eignung für die Klassifizierungsaufgabe zu bewerten.

Tabelle 7.1: Übersicht der Kriterien zur Modellauswahl.

Kriterium	Beschreibung
Herkunft	Das Land in dem das Modell entwickelt wurde bzw. der Hauptsitz des Anbieters
Lizenz	Art der Lizenz, wie bspw. Open-Source oder proprietär
Größe	Anzahl der Parameter in Milliarden (B)
Kontext	Maximale Anzahl der Token, die das Modell verarbeiten kann
Letztes Update	Datum der letzten Aktualisierung des Modells bei Hugging Face [31]
Downloads	Anzahl der Downloads des Modells bei Hugging Face, sofern verfügbar

Ein wesentliches Auswahlkriterium ist die geografische **Herkunft** der Modelle. Als *EU-Modell* gelten Modelle, deren Anbieter ihren Hauptsitz in der EU haben, deren Veröffentlichung in der EU erfolgt oder die schwerpunktmäßig in der EU entwickelt oder verfeinert wurden. Alle anderen Modelle werden als *international* eingeordnet. Diese Unterscheidung ist relevant, da europäische Modelle sowohl beim Training als auch beim Betrieb stärker den europäischen Datenschutzbestimmungen unterliegen und somit potenziell besser für den Einsatz in datensensiblen Bereichen wie der Klassifizierung von BPMN-Modellen geeignet sind. Zudem ist die DSGVO eine unmittelbar geltende EU-Verordnung und prägt dadurch die regulatorischen Anforderungen an Entwicklung und Betrieb besonders stark.

Ein weiteres zentrales Kriterium ist die **Lizenzierung** der Modelle. Als *Open-Source* veröffentlichte Modelle im Sinne der *Open Source Definition* der Open Source Initiative (OSI) erlauben Nutzung, Studium, Veränderung und Weiterverbreitung unter einer konformen Lizenz [59]. Davon zu unterscheiden sind *Open-Weights*-Modelle: Hier sind die Gewichte zwar öffentlich beziehbar, wodurch Modelle bspw. selbst betrieben werden können, die zugehörige Lizenz kann jedoch restriktive Klauseln enthalten (z. B. Nutzungs- oder Output-Beschränkungen). Daher gelten sie rechtlich nicht als OSI-Open-Source. Ein Beispiel hierfür ist die Meta-Llama 3 Community Lizenz [45]. In dieser Arbeit gilt ein Modell als *offen* bzw. *Open-Source-nah*, wenn

1. die Gewichte frei zugänglich sind und
2. eine *permissive* Lizenz (z. B. Apache-2.0 oder MIT) eine breite kommerzielle Nutzung erlaubt (z. B. Mistral 7B [32], GPT-OSS [62, 64] oder DeepSeek V3.1 [28]).

Modelle mit *Community-* oder *Eigennutzer*-Lizenzen, z. B. Mistral Large Instruct unter Mistral Research License [33, 49], werden rechtlich *nicht* als OSI-Open-Source gewertet, können aber technisch als Vergleich herangezogen werden.

Die **Modellgröße** wird in *Anzahl der Parameter* angegeben. Meist in *Milliarden* (Billionen, engl. *Billion*) Parametern (B, engl. *Billion*). 1 B = 10^9 Parameter. Diese Zahl korreliert mit dem Ressourcenbedarf für Training und Inferenz sowie der Leistungsfähigkeit [56]. Für die Einordnung werden hier folgende Klassen verwendet:

- **Klein** ($\leq \sim 25$ B Parameter): z. B. Mistral 7B Instruct ($\sim 7,3$ B) [32].
- **Groß** ($> \sim 25$ B Parameter): z. B. GPT-OSS 120B (~ 117 B) [62].

Die Klassifikation dient als methodische Abgrenzung für die Experimente. Kleinere Modelle lassen sich häufig lokal ausführen, größere erfordern typischerweise mehrere GPUs. Parameterzahl ist dabei ein nützlicher, wenn auch unvollständiger Indikator für Ressourcenbedarf und erwartete Leistung. Dies ermöglicht konsistente Entscheidungen zu Deployment und Kosten.

Der **Kontext** gibt an, wie viele Token ein Modell gleichzeitig verarbeiten kann. Ein Token ist dabei eine Grundeinheit von Text, die ein Wort, einen Teil eines Wortes oder sogar ein einzelnes Zeichen darstellen kann. Die Größe des Kontextfensters beeinflusst maßgeblich, wie gut ein Modell längere Texte verstehen und darauf reagieren kann [7].

Neben den genannten Hauptkriterien werden weitere Merkmale erfasst, um die Modelle umfassend zu charakterisieren. Dazu gehören das **letzte Update** des Modells, um einordnen zu können, wie aktuell das Modell ist, und wie viele **Downloads** das Modell hat, sofern verfügbar. Diese Informationen helfen, die Popularität und Akzeptanz der Modelle in der Community einzuschätzen.

Im nächsten Abschnitt werden auf Basis dieser Kriterien die ausgewählten Modelle vorgestellt.

7.2 Modellvorstellung

Tabelle 7.2 stellt die für diese Arbeit ausgewählten Modelle kurz vor. Der Stichtag der Modellauswahl war der 30. September 2025. Im Folgenden wird die Modellauswahl im Detail erläutert.

Zusätzlich zu Herkunft, Lizenz, Größe und Kontext wurden die Modelle anhand *etablierter Benchmarks für generelle Fähigkeiten* validiert: (1) *MMLU-Pro* [86] - eine robustere Weiterentwicklung von MMLU [24] - das fachübergreifendes Wissen und anspruchsvolles Schlussfolgern mit zehn Antwortoptionen pro Aufgabe prüft. (2) *BIG-bench Hard (BBH)* [82], das mehrschrittiges Denken auf 23 besonders schwierigen Aufgaben evaluiert. (3) Commonsense-Benchmarks wie *HellaSwag* [87] und *WinoGrande* [76], die alltagstaugliches Verständnis prüfen. Bei HellaSwag wählt das Modell die plausibelste Fortsetzung einer kurzen Szene und bei WinoGrande klärt es, worauf sich ein Pronomen im Satz bezieht. (4) *AIME* [44], ein anspruchs-

Tabelle 7.2: Übersicht aller Modelle mit technischen Eckdaten (Stand 30.09.2025).

(a) Technische Eckdaten und Herkunft der Modelle.

Modell	Parameter (B)	Kontext (Tokens)	Herkunftsland
Mistral-7B-Instruct-v0.3	7,24	32.000	Frankreich [32]
Mixtral-8x7B-Instruct-v0.1	46,7 total / 12,9 aktiv ¹	32.000	Frankreich [34, 50]
Mistral-Large-Instruct-2411	123	128.000	Frankreich [33]
Mistral Medium 3.1	n.v.	128.000	Frankreich [52]
Gemma-3-12B-it	12,2	128.000	Großbritannien ² [29]
Gemma-3-27B-it	27,4	128.000	Großbritannien [30]
Qwen2.5-7B-Instruct	7,62	131.072	China [36]
Qwen3-235B-A22B-Thinking-2507	235	256.000	China [35]
DeepSeek-R1-Distill-Qwen-14B	14,8	131.072	China [26]
DeepSeek-V3.1	671 total / 37 aktiv	128.000	China [28]
GPT-OSS-20B	20,91 total / 3,61 aktiv	131.072	USA [62]
GPT-OSS-120B	116,83 total / 5,13 aktiv	131.072	USA [62]
GPT-4o (2024-11-20)	n.v.	128.000	USA [63]

(b) Lizenz, letzte Updates und Downloads der Modelle.

Modell	Lizenz	Letztes Update	Downloads
Mistral-7B-Instruct-v0.3	Apache-2.0	24.07.2025	687.000 [32]
Mixtral-8x7B-Instruct-v0.1	Apache-2.0	24.07.2025	43.500 [34]
Mistral-Large-Instruct-2411	Mistral Research License	28.07.2025	4.200 [33, 49]
Mistral Medium 3.1	Proprietär	25.08.2025	n.v. [52]
Gemma-3-12B-it	Gemma	21.03.2025	523.000 [23, 29]
Gemma-3-27B-it	Gemma	21.03.2025	1.180.000 [23, 30]
Qwen2.5-7B-Instruct	Apache-2.0	12.01.2025	5.100.000 [36]
Qwen3-235B-A22B-Thinking-2507	Apache-2.0	17.08.2025	52.900 [35]
DeepSeek-R1-Distill-Qwen-14B	MIT	24.02.2025	341.000 [26]
DeepSeek-V3.1	MIT	05.09.2025	447.000 [28]
GPT-OSS-20B	Apache-2.0	26.08.2025	6.450.000 [62]
GPT-OSS-120B	Apache-2.0	26.08.2025	3.600.000 [62]
GPT-4o (2024-11-20)	Proprietär	20.11.2024	n.v. [63]

¹ Mixtral nutzt Mixture-of-Experts (MoE) mit 8 Experten als Architektur. Die Gesamtparameterzahl bezieht sich auf alle Experten, die aktive Parameterzahl auf den jeweils genutzten Expertenanteil pro Inferenzdurchlauf [50].

² Google DeepMind hat seinen Hauptsitz in London, gehört jedoch zu Alphabet (USA). Wo genau trainiert wurde, ist unklar.

voller Mathematik-Benchmark auf Grundlage der *American Invitational Mathematics Examination*, bewertet präzises mehrschrittiges Problemlösen an wettbewerbsnahen Aufgaben. In diesen Benchmarks schneiden die in Tabelle 7.2 aufgeführten Modelle in ihren jeweiligen Größenklassen durchweg gut ab. Dies wird u. a. durch die veröffentlichten Auswertungen von Mistral [47, 51], OpenAI [62], Google DeepMind [29, 30], Alibaba [35, 71] sowie DeepSeek [26, 28] gestützt.

Die französische Firma Mistral AI bietet mehrere leistungsstarke Modelle an, die zum Großteil offene Gewichte haben. Durch ihre Herkunft repräsentieren die Mistral Modelle in dieser Arbeit die EU-Modelle. **Mistral-7B-Instruct-v0.3** [32] ist ein 7,24 B Parameter großes Modell mit einem Kontextfenster von 32.000 Tokens. Es wurde speziell für Anweisungsfolgen (engl. Instruct) optimiert und ist unter der Apache-2.0-Lizenz frei verfügbar. Das **Mixtral-8x7B-Instruct-v0.1** Modell [34] nutzt eine Mixture-of-Experts-Architektur mit insgesamt 46,7 B Parametern, von denen jedoch nur 12,9 B aktiv genutzt werden. Es hat ebenfalls ein Kontextfenster von 32.000 Tokens und ist unter der Apache-2.0-Lizenz verfügbar. Das **Mistral-Large-Instruct-2411** Modell [33] ist mit 123 B Parametern deutlich größer und bietet ein Kontextfenster von 128.000 Tokens. Es wird unter der Mistral Research License veröffentlicht, die die Nutzung auf nicht-kommerzielle Forschung beschränkt [49]. Das Modell **Mistral Medium 3.1** [52] bietet ein Kontextfenster von 128.000 Tokens und gilt als aktuelles Spitzenmodell der Mistral-Modellreihe. Anders als die übrigen Mistral-Modelle ist es proprietär und wird von Mistral AI auf EU-Servern unter Beachtung der DSGVO betrieben [48, 53]. Damit ist die Verarbeitung sensibler Daten möglich. Das Modell eignet sich daher - trotz nicht veröffentlichter Gewichte - für den Einsatz in datenschutzkritischen Szenarien.

Die Gemma-3-Modelle von Google DeepMind repräsentieren eine neue Generation multimodaler LLMs mit offenen Gewichten. Die hier betrachteten Varianten sind **Gemma-3-12B-it** [29] mit 12,2 B Parametern und **Gemma-3-27B-it** [30] mit 27,4 B Parametern. Beide Modelle verfügen über ein großes Kontextfenster von 128.000 Tokens und sind unter der proprietären Gemma-Lizenz veröffentlicht, die eine breite kommerzielle Nutzung erlaubt [23]. Die genaue Herkunft der Modelle ist unklar, da Google DeepMind seinen Hauptsitz in Großbritannien hat, jedoch zu Alphabet in den USA gehört. Wo genau die Modelle trainiert wurden, ist nicht bekannt.

Die Qwen-Modelle wurden von Alibaba Cloud in China entwickelt. Das kleinere Modell **Qwen2.5-7B-Instruct** [36] hat 7,62 B Parameter, ein Kontextfenster von

131.072 Tokens und ist unter der Apache-2.0-Lizenz frei verfügbar. Das größere Modell **Qwen3-235B-A22B-Thinking-2507** [35] verfügt über 235 B Parameter, ein Kontextfenster von 256.000 Tokens und ist ebenfalls unter der Apache-2.0-Lizenz veröffentlicht. Zugleich ist bei außerhalb der EU entwickelten Modellen - daher auch bei den chinesischen Qwen-Modellen - besondere Vorsicht geboten, da die Trainingsdaten und -methoden nicht immer transparent sind und möglicherweise nicht den europäischen Datenschutzstandards und -ethiken entsprechen.

Das chinesische Unternehmen DeepSeek AI hat mit **DeepSeek-R1-Distill-Qwen-14B** [26] ein Modell mit 14,8 B Parametern veröffentlicht, das auf Qwen-2.5 basiert, ein Kontextfenster von 131.072 Tokens bietet und unter der MIT-Lizenz frei verfügbar ist. Das größere Modell **DeepSeek-V3.1** [28] setzt auf eine Mixture-of-Experts-Architektur mit insgesamt 671 B Parametern, von denen pro Token 37 B aktiv sind. Es bietet ein Kontextfenster von 128.000 Tokens und ist ebenfalls MIT-lizenziert. Besonders bemerkenswert sind die DeepSeek-Modelle, weil DeepSeek im Januar 2025 mit *DeepSeek-R1-Zero* [27] eines der ersten permissiv lizenzierten Reasoning-Modelle in OpenAI-Größenordnung vorlegte und zugleich einen Trainings-Ansatz etablierte, bei dem LLM-Reasoning nahezu ausschließlich über Reinforcement Learning erlernt wird [17].

GPT-4o [63] ist ein proprietäres Modell von OpenAI mit einem Kontextfenster von 128.000 Tokens. Es wurde am 20. November 2024 veröffentlicht und ist das einzige internationale proprietäre Modell in dieser Arbeit. Die genauen Parameterzahlen sind nicht bekannt. GPT-4o wird über die OpenAI-API bereitgestellt und ist das erste Omni-Modell von OpenAI, das neben Text auch Bilder als Eingabe akzeptieren kann. Durch ChatGPT [60] sind die Funktionen und Fähigkeiten des Modells bekannt und es gilt als bestes Modell für „generelle Fähigkeiten“. Dadurch gilt es als der De-facto-Standard in der Industrie. Das Modell dient in diesem Vergleich als Referenzpunkt für den aktuellen Stand der Technik. Mit den GPT-OSS-Modellen [62] hat OpenAI zudem zwei Modelle mit offenen Gewichten unter Apache-2.0-Lizenz veröffentlicht, die explizit für Forschung und kommerzielle Nutzung freigegeben sind. Das **GPT-OSS-20B** Modell hat 20,91 B Parameter (3,61 B aktiv) und ein Kontextfenster von 131.072 Tokens. Das größere **GPT-OSS-120B** Modell verfügt über 116,83 B Parameter (5,13 B aktiv) und das gleiche Kontextfenster. Sie sind spannende Vergleichsmodelle, da sie von einem der führenden LLM-Anbieter stammen und dennoch offen verfügbar sind.

7 Modellauswahl

Insgesamt deckt die Modellauswahl in dieser Arbeit eine breite Palette von Modellgrößen, Architekturen und Lizenztypen ab. Die Mistral-Modelle repräsentieren die EU-Modelle mit offenen Gewichten, während die Gemma-, Qwen- und GPT-OSS-Modelle internationale Alternativen aus verschiedenen Herkunftsländern darstellen. Die DeepSeek-Modelle bieten innovative Ansätze im Reasoning-Bereich, und GPT-4o dient als aktueller Industriestandard. Diese Vielfalt ermöglicht eine umfassende Evaluation der Modelle hinsichtlich ihrer Eignung für die Klassifizierungsaufgabe von BPMN-Modellen. Im nächsten Kapitel wird aufbauend auf den vorgestellten Modellen der Versuchsaufbau und die Durchführung der Experimente beschrieben.

8 Versuchsaufbau und Durchführung

Wie in Abschnitt 3.4 beschrieben, soll ein fairer Vergleich verschiedener LLMs erreicht werden. Dazu werden alle der im vorherigen Kapitel beschriebenen Modelle durch dieselbe Klassifikationspipeline geschickt und anhand der im Kapitel 3.2 definierten Metriken (Accuracy, Precision, Recall, F1) bewertet. Außerdem wird betrachtet, wie viele Testfälle erfolgreich klassifiziert wurden und wie robust die Modelle sind. Dieses Kapitel beschreibt den konkreten Versuchsaufbau und die Durchführung der Experimente. Die hier dokumentierten Parameter und Konfigurationen sind wesentlich, um die Ergebnisse nachvollziehbar und reproduzierbar zu machen.

Um sowohl kleine als auch große Modelle testen zu können, wurde *OpenRouter* [67] als API-Anbieter genutzt. Über diese Cloud-basierte Schnittstelle lassen sich auch Modelle ausführen, die lokal aufgrund begrenzter Hardware nicht betrieben werden können. Der API-Schlüssel wird über eine Umgebungsvariable in die Konfigurationsdatei eingebunden, um sensible Daten aus den Konfigurationen fernzuhalten.

In den Experimenten wurden mehrere Modelle aus unterschiedlichen Anbieterfamilien getestet. Für jeden Anbieter gibt es ein eigenes Experiment, in dem mehrere Modellgrößen (z. B. 7B, 8x7B, Large) gegeneinander verglichen werden. Da alle Experimente die gleiche Pipeline und die gleichen Datensätze verwenden, können auch die Ergebnisse verschiedener Anbieter untereinander verglichen werden. Diese Aufteilung in verschiedene Experimente dient lediglich der Übersichtlichkeit in der Benutzeroberfläche des Evaluationsframeworks.

8.1 Vergleichbarkeit

Um die Vergleichbarkeit der Experimente zu gewährleisten, werden alle Modelle durch dieselbe Klassifikationspipeline geschickt. Die technische Implementierung dieser Pipeline wurde in Kapitel 4 beschrieben und kann im Evaluationsframework genutzt werden. Jeder Testfall besteht aus einem BPMN-Prozess mit Labeln für DSGVO-kritische Aktivitäten. Ein Testfall gilt als korrekt klassifiziert, wenn genau die als kritisch gelabelten Aktivitäten auch als kritisch erkannt werden - bereits ein FP oder FN führt zu einem nicht bestandenen Testfall.

Als Datenbasis kommen drei im Labeling-Tool erzeugte Testdatensätze zum Einsatz. Diese decken unterschiedliche Prozesskontakte ab und werden in den Experimenten mit den ids 1 (*kleine Prozesse*), 2 (*Universität*) und 7 (*mittelgroße Praxisbeispiele*) referenziert. Für jedes Experiment werden alle verfügbaren Datensätze verwendet. Auf diese Weise können Unterschiede zwischen den Modellen nicht auf unterschiedliche Datenquellen zurückgeführt werden.

8.2 Konfigurationen

Die Konfigurationen der Experimente sind im YAML-Format in Listings 8.1, 6, 7, 8 und 9 dargestellt. Sie enthalten die zu evaluierenden Modelle, die zu verwendenden Datensätze, den Basis-Seed sowie die Anzahl der Wiederholungen und weitere Rahmenparameter. In Listing 8.1 wird ein Experiment dargestellt, in dem vier verschiedene Mistral-Modelle über OpenRouter evaluiert werden. Die Datensätze werden jeweils fünf Mal durchlaufen. Der Basis-Seed ist auf 24523833 gesetzt.

Auf Basis des Seeds aus der Konfiguration und der aktuellen Wiederholungsnummer wird in dem Evaluationsframework für jede Wiederholung deterministisch ein neuer Seed generiert. Dadurch sind die Ergebnisse reproduzierbar und dennoch wird die Stabilität der Modelle über mehrere Wiederholungen mit unterschiedlichen Seeds abgebildet. Alle Datensätze, Konfigurationen und die daraus resultierenden Ergebnisse sind außerdem im GitLab-Repository verfügbar¹.

¹Siehe GitLab Repository: <https://gitlab.uni-ulm.de/merten/gripl-master-thesis>

Listing 8.1: Konfigurationsdatei des Experiments mit Mistral Modellen

```
1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering
2 seed: 24523833
3 maxConcurrent: 10
4 repetitions: 5
5 models:
6   - label: Mistral-7B-Instruct-v0.3
7     llmProps:
8       baseUrl: https://openrouter.ai/api/v1
9       modelName: mistralai/mistral-7b-instruct-v0.3
10      apiKey: ${OPEN_ROUTER_API_KEY}
11      temperature: 0.1
12      topP: 1
13   - label: Mistral-8x7B-Instruct-v0.1
14     llmProps:
15       baseUrl: https://openrouter.ai/api/v1
16       modelName: mistralai/mixtral-8x7b-instruct
17       apiKey: ${OPEN_ROUTER_API_KEY}
18       temperature: 0.1
19       topP: 1
20   - label: Mistral-Large-Instruct-2411
21     llmProps:
22       baseUrl: https://openrouter.ai/api/v1
23       modelName: mistralai/mistral-large-2411
24       apiKey: ${OPEN_ROUTER_API_KEY}
25       temperature: 0.1
26       topP: 1
27   - label: Mistral Medium 3.1
28     llmProps:
29       baseUrl: https://openrouter.ai/api/v1
30       modelName: mistralai/mistral-medium-3.1
31       apiKey: ${OPEN_ROUTER_API_KEY}
32       temperature: 0.1
33       topP: 1
34 datasets:
35   - 2
36   - 7
37   - 1
```

Um bei der Zero-Shot-Klassifikation deterministische und formatkonsistente Ergebnisse zu erzielen, wurden die Inferenz-Hyperparameter temperature und topP bewusst konservativ gewählt. Der Parameter temperature steuert die Zufälligkeit der Modellausgabe: niedrige Werte priorisieren die wahrscheinlichsten Tokens und machen die Ausgabe deterministischer. Für Aufgaben, die faktische Genauigkeit und Precision erfordern, empfehlen aktuelle Arbeiten daher sehr niedrige Temperaturen; höhere Werte (z. B. $\text{temperature}=0,8$ oder 2) verschlechtern hingegen die Klassifikationsleistung und führen zu nicht-reproduzierbaren Ausgaben [54, 73]. In dieser Arbeit wird durchgängig $\text{temperature}=0,1$ verwendet. Dieser Wert reduziert Zufallseffekte deutlich, ohne den Output übermäßig einzuschränken, und folgt den Empfehlungen vergleichbarer Studien zur Zero-Shot-Klassifikation [54]. Auf $\text{temperature}=0$ wurde bewusst verzichtet: Zwar wäre die Ausgabe damit theoretisch noch deterministischer, in eigenen Tests traten jedoch vermehrt Formatfehler auf (z. B. fehlende oder zusätzliche Zeichen im JSON-Output). $\text{temperature}=0,1$ stellt daher einen guten Kompromiss zwischen Determinismus und Formatstabilität dar.

Der Parameter topP steuert, wie „eng“ die Auswahl für das nächste Token gefasst wird. Dafür werden die möglichen Fortsetzungen nach ihrer Wahrscheinlichkeit sortiert und nur so viele der wahrscheinlichsten genommen, bis zusammen etwa topP erreicht ist (z. B. $0,9 \hat{=} 90\%$). Aus diesen wird dann gewählt. Beispiel: Bei „Der Himmel ist ...“ stehen „blau“, „bewölkt“ oder „klar“ meist weit oben. Mit $\text{topP}=0,9$ bleiben diese Kandidaten im Rennen, während seltene oder unpassende Fortsetzungen („gestern“, „eine“) ignoriert werden. Bei $\text{topP}=0,2$ bleibt oft fast nur „blau“ übrig, weil es das Wahrscheinlichste ist. Mit $\text{topP}=1$ wird nichts vorab ausgeschlossen - dann bestimmt allein die temperature den Zufallsanteil [73]. In Kombination mit sehr niedriger temperature liefert $\text{topP}=1$ fokussierte, weitgehend deterministische Ausgaben bei gleichzeitig maximalem Stichprobenraum [54].

8.3 Durchführung

Die Durchführung der Experimente erfolgt automatisiert über das Evaluationsframework. Für jede in der Konfigurationsdatei angegebene Modellvariante werden alle Testfälle aus den ausgewählten Datensätzen an die Klassifikationspipeline überge-

ben. Während der Ausführung werden für jeden Testfall die Einzelergebnisse der Konfusionsmatrix sowie der Status „bestanden“ oder „nicht bestanden“ bestimmt. Diese Kennzahlen werden pro Modell aggregiert und anschließend genutzt, um die aus Kapitel 3.2 bekannten Metriken zu berechnen.

Für jedes Modell werden außerdem über alle Wiederholungen hinweg sowohl die Durchschnittswerte als auch dessen Standardabweichung für die Metriken berechnet. Die Standardabweichung gibt an, wie stark die Ergebnisse der einzelnen Läufe um den Mittelwert streuen. Ein niedriger Wert deutet auf eine hohe Stabilität des Modells hin, während ein hoher Wert auf eine größere Variabilität in den Ergebnissen hinweist. Diese Information ist besonders wichtig, um die Zuverlässigkeit der Modelle zu bewerten, da einige LLMs aufgrund ihrer nicht-deterministischen Natur unterschiedliche Ergebnisse bei wiederholten Ausführungen desselben Testfalls liefern können.

Im nächsten Kapitel werden die erzielten Ergebnisse dieser Experimente detailliert vorgestellt und analysiert.

9 Ergebnisse

Dieses Kapitel präsentiert die Resultate der Klassifizierungsexperimente und stellt sie in den Kontext der in Abschnitt 1.2 formulierten Forschungsfragen und Qualitätsziele. Ziel der Untersuchung ist es, LLMs auf ihre Fähigkeit zur Identifikation DSGVO-kritischer Aktivitäten in BPMN-Prozessmodellen zu prüfen. Die folgenden Abschnitte fassen die zentralen Erkenntnisse zusammen, analysieren die Ergebnisse entlang verschiedener Modellkategorien, bewerten die Robustheit und veranschaulichen typische Fehlerbilder anhand von Fallstudien. Abschließend werden die Forschungsfragen beantwortet.

Alle Abbildungen und Tabellen in diesem Kapitel wurden auf Basis der im Evaluationsframework generierten Experimentergebnisse erstellt. Dabei wurden die Ergebnisse aus den einzelnen Experimenten zusammengeführt, sodass sämtliche Modelle gemeinsam verglichen werden können. Die im Evaluationsframework erzeugten Diagramme wurden bewusst nicht direkt übernommen, da sie bei vielen Modellen nur schwer skalieren. Die vollständigen Reports und Rohdaten der einzelnen Experimente können weiterhin dem Repository¹ entnommen und über das Evaluationsframework erkundet werden.

9.1 Zusammenfassung der Ergebnisse

Für jedes der dreizehn untersuchten Modelle wurden fünf unabhängige Läufe mit unterschiedlichen Seeds durchgeführt. Abbildung 9.1 visualisiert die mittleren Werte für Precision, Recall, F1-Score und Accuracy jeweils inklusive Standardabweichung über alle Wiederholungen hinweg. Für einen besseren Vergleich sind proprietäre Modelle rot, kleinere Modelle orange und größere Modelle blau eingefärbt. Die Einteilung entspricht der in Kapitel 7 beschriebenen Kategorisierung.

¹Siehe GitLab Repository: <https://gitlab.uni-ulm.de/merten/gripl-master-thesis>

9 Ergebnisse

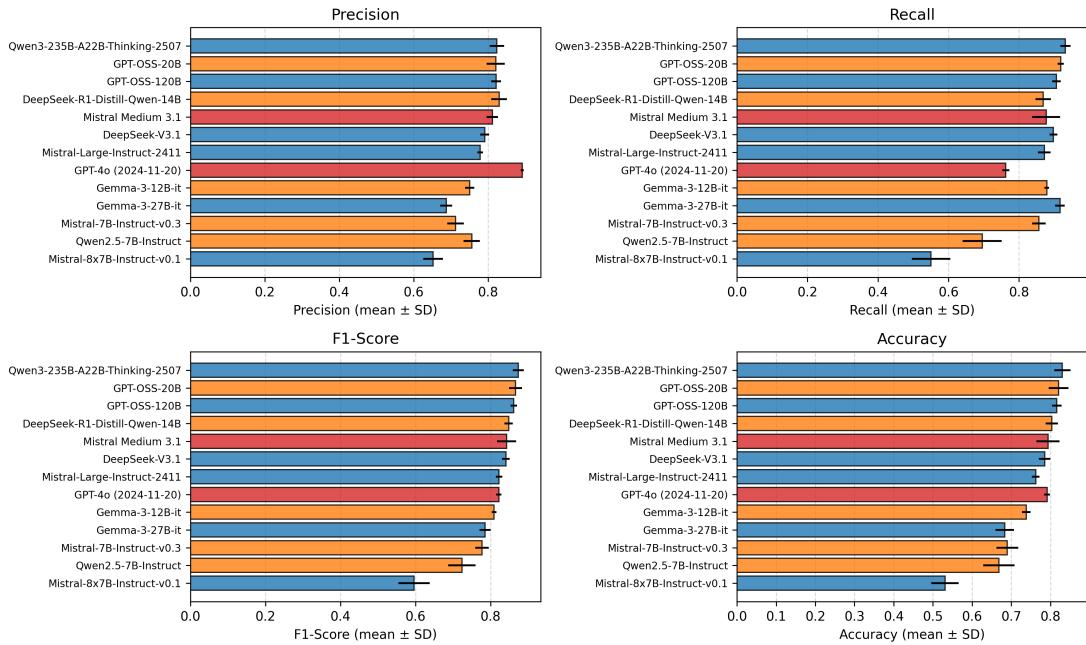


Abbildung 9.1: Durchschnittliche Werte für Precision, Recall, F1-Score und Accuracy der untersuchten Modelle über alle Wiederholungen hinweg inklusive Standardabweichung.

Für die Einordnung der Ergebnisse ist entscheidend, wie Precision und Recall zu interpretieren sind: Ein höherer *Recall* bedeutet, dass ein Modell nur wenige kritische Aktivitäten übersieht (wenige FN) und eignet sich damit besonders gut als vorsorgliches *Screening*. Eine höhere *Precision* zeigt dagegen, dass selten unkritische Aktivitäten fälschlich als kritisch markiert werden (wenige FP), sodass der manuelle Prüfaufwand sinkt. Der *F1-Score* als harmonisches Mittel aus Precision und Recall fasst beide Perspektiven zu einer Kennzahl zusammen: Er steigt nur dann, wenn *beide* Werte hoch sind, und bestraft Ungleichgewichte (z. B. sehr hohe Precision bei niedrigem Recall). Entsprechend setzen die Qualitätsziele in Abschnitt 3.2 einen F1-Score $\geq 0,80$ als Maß für die kombinierte Screening-Leistung, priorisieren aber den Recall, um das Übersehen kritischer Aktivitäten zu vermeiden. Vor diesem Hintergrund lassen sich die in Abbildung 9.1 sichtbaren Unterschiede zwischen den Modellen einordnen und erklären.

Insgesamt erfüllen neun der dreizehn Modelle den Zielwert eines F1-Scores $\geq 0,80$ aus Abschnitt 3.2. Besonders gut schneiden Qwen3-235B-A22B-Thinking-2507,

9 Ergebnisse

GPT-0SS-120B und GPT-0SS-20B ab, die F1-Scores zwischen 0,862 und 0,874 erreichen. Hervorzuheben ist, dass auch mehrere kleinere Modelle diesen Zielwert überschreiten. Am unteren Ende des Spektrums liegt das europäische Mixtral-8x7B-Instruct-v0.1, das sowohl in Precision als auch in Recall schwach abschneidet und als einziges Modell einen F1-Score deutlich unter 0,60 erzielt.

Die Abbildung zeigt zudem, dass sich die Modelle hinsichtlich Precision und Recall unterschiedlich verhalten. GPT-4o erreicht beispielsweise mit 0,892 die höchste Precision, liegt jedoch beim Recall mit 0,762 unter dem Mindestziel von 0,80 - es würde demnach in der Praxis vergleichsweise viele kritische Aktivitäten übersehen. Umgekehrt erzielt Gemma-3-27B-it einen sehr hohen Recall von 0,916, wird aber durch eine niedrige Precision von 0,687 ausgebremst. Dadurch klassifiziert es viele unkritische Aktivitäten fälschlich als kritisch, was den manuellen Prüfaufwand erhöhen würde. Modelle wie Qwen3-235B-A22B-Thinking-2507, GPT-0SS-20B und DeepSeek-R1-Distill-Qwen-14B bieten eine ausgewogene Balance und zählen deshalb zu den Spitzenreitern.

Tabelle 9.1 fasst alle Metriken mit ihren Mittelwerten und Standardabweichungen tabellarisch zusammen. Für die weitere Analyse werden diese Werte nur noch punktuell zitiert, um Wiederholungen zu vermeiden.

Tabelle 9.1: Aggregierte Mittelwerte und Standardabweichungen der Evaluationsmetriken über alle fünf Wiederholungen hinweg.

Modell	Precision	Recall	F1-Score	Accuracy
DeepSeek-V3.1	$0,791 \pm 0,012$	$0,897 \pm 0,011$	$0,841 \pm 0,011$	$0,785 \pm 0,015$
DeepSeek-R1-Distill-Qwen-14B	$0,829 \pm 0,021$	$0,868 \pm 0,022$	$0,848 \pm 0,011$	$0,803 \pm 0,016$
Gemma-3-12B-it	$0,751 \pm 0,013$	$0,879 \pm 0,006$	$0,810 \pm 0,006$	$0,738 \pm 0,011$
Gemma-3-27B-it	$0,687 \pm 0,016$	$0,916 \pm 0,014$	$0,785 \pm 0,015$	$0,683 \pm 0,023$
Mistral-7B-Instruct-v0.3	$0,712 \pm 0,022$	$0,856 \pm 0,019$	$0,777 \pm 0,018$	$0,690 \pm 0,028$
Mixtral-8x7B-Instruct-v0.1	$0,652 \pm 0,027$	$0,550 \pm 0,054$	$0,596 \pm 0,042$	$0,531 \pm 0,035$
Mistral-Large-Instruct-2411	$0,779 \pm 0,008$	$0,872 \pm 0,018$	$0,823 \pm 0,008$	$0,762 \pm 0,010$
Mistral Medium 3.1	$0,811 \pm 0,015$	$0,877 \pm 0,040$	$0,843 \pm 0,025$	$0,794 \pm 0,029$
GPT-0SS-20B	$0,820 \pm 0,024$	$0,918 \pm 0,009$	$0,866 \pm 0,017$	$0,821 \pm 0,025$
GPT-0SS-120B	$0,822 \pm 0,013$	$0,906 \pm 0,012$	$0,862 \pm 0,009$	$0,816 \pm 0,012$
GPT-4o	$0,892 \pm 0,004$	$0,762 \pm 0,010$	$0,822 \pm 0,007$	$0,791 \pm 0,007$
Qwen2.5-7B-Instruct	$0,756 \pm 0,022$	$0,696 \pm 0,055$	$0,724 \pm 0,037$	$0,668 \pm 0,040$
Qwen3-235B-A22B-Thinking-2507	$0,824 \pm 0,019$	$0,932 \pm 0,014$	$0,874 \pm 0,015$	$0,830 \pm 0,021$

Neben den zusammengeführten Metriken wurden die 25 Testfälle je Modell auch hinsichtlich der Frage ausgewertet, wie viele Testfälle das Modell bestanden hat.

9 Ergebnisse

Ein Testfall gilt als bestanden, wenn alle DSGVO-kritischen Aktivitäten korrekt klassifiziert und keine unkritischen Aktivitäten fälschlicherweise als kritisch markiert wurden. Schlägt ein Testfall fehl, liegt mindestens eine Fehlklassifikation (FP oder FN) vor. Daneben gab es wenige technische Fehler, etwa Parsing-Fehler oder Timouts, die selbst nach dem in Abschnitt 4.3 beschriebenen Retry-Mechanismus nicht behoben werden konnten. Diese technischen Fehler wurden, wie in Abschnitt 3.3 festgehalten, nicht in die Metriken einbezogen und sind hier separat ausgewiesen.

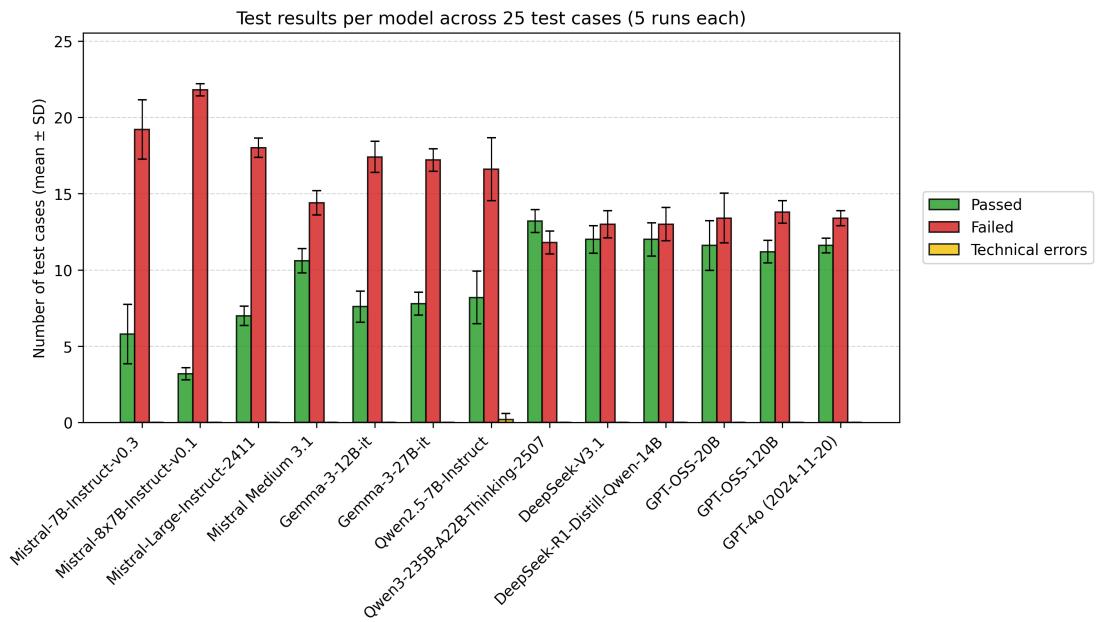


Abbildung 9.2: Durchschnittliche Testergebnisse pro Modell in Bezug auf die 25 Testfälle über fünf Wiederholungen hinweg.

Wie Abbildung 9.2 zeigt, korrespondieren die Ergebnisse der Testfälle weitgehend mit den Metriken aus Tabelle 9.1. Modelle wie Qwen3-235B-A22B-Thinking-2507, DeepSeek-V3.1 und GPT-OSS-20B bestehen im Durchschnitt fast die Hälfte der Testfälle, während Mistral-7B-Instruct-v0.3 und Mistral-8x7B-Instruct-v0.1 die meisten Testfälle verfehlten. Auffällig ist, dass Qwen2.5-7B-Instruct als einziges Modell vereinzelte technische Fehler zeigte. Dabei ging es um Parsing-Fehler. Diese treten jedoch selten auf und haben keinen Einfluss auf die berechneten Metriken. Insgesamt bestätigt die Testfallanalyse, dass die leistungsfähigsten Modelle nicht nur hohe Metrikwerte erzielen, sondern auch die meisten Testfälle bestehen.

9.2 Analyse nach Modellkategorien

Für ein besseres Verständnis der Leistungsunterschiede werden die Modelle im Folgenden nach verschiedenen Kriterien gruppiert und verglichen. Dabei wird jeweils diskutiert, wie sich die Gruppen in Bezug auf die Qualitätsziele unterscheiden und welche Trends sich beobachten lassen. Die Implikationen für Praxis und Forschungsfragen werden am Ende dieses Kapitels in Abschnitt 9.5 gebündelt beantwortet.

Proprietäre versus Open-Weight Modelle

Die beiden proprietären Modelle GPT-4o und Mistral Medium 3.1 erreichen F1-Scores von 0,822 bzw. 0,843. Trotz seiner exzellenten Precision von 0,892 verfehlt GPT-4o aufgrund des niedrigen Recalls von 0,762 das Mindestziel und übersieht damit relativ viele kritische Aktivitäten. Mistral Medium 3.1 bietet mit einem Recall von 0,877 eine bessere Balance und erfüllt alle Qualitätsziele.

Die offene Kategorie zeigt ein heterogenes Bild. Mehrere Modelle wie Qwen3-235B-A22B-Thinking-2507 mit einem F1-Score = 0,874, GPT-OSS-20B mit F1-Score = 0,866 und DeepSeek-R1-Distill-Qwen-14B mit F1-Score = 0,848 übertreffen die proprietären Modelle. Sie erkennen kritische Aktivitäten sehr zuverlässig und klassifizieren nur wenige unkritische Aktivitäten fälschlich als kritisch. Gleichzeitig gibt es mit Mixtral-8x7B-Instruct-v0.1, das F1-Score = 0,596 erzielte, auch klare Ausreißer nach unten, die weder genug kritische Aktivitäten erkennen noch eine akzeptable Precision bieten.

Insgesamt zeigt sich, dass hochwertige offene Modelle ein besseres Verhältnis von Recall und Precision aufweisen und die Qualitätsziele häufig klar erfüllen. Für die Praxis bedeutet dies, dass offene Modelle eine attraktive Alternative zu proprietären Lösungen darstellen, jedoch ist die Auswahl des Modells entscheidend, da die Leistungsunterschiede innerhalb der offenen Kategorie erheblich sind.

Kleine versus Große Modelle

Tabelle 9.2 vergleicht die Mittelwerte der Metriken für kleine Modelle ($\leq 25B$ Parameter) und große Modelle ($> 25B$ Parameter). Im Durchschnitt unterscheiden sich die Gruppen nur geringfügig. Die kleinen Modelle erreichen einen mittleren F1-Score von 0,805 und die großen Modelle 0,806, wobei die großen Modelle ohne den Ausreißer Mixtral-8x7B-Instruct-v0.1 einen leicht höheren Durchschnitt von 0,836 erreichen. Der beste F1-Score unter den kleinen Modellen stammt von GPT-OSS-20B mit 0,866; bei den großen Modellen führt Qwen3-235B-A22B-Thinking-2507 mit 0,874. Bemerkenswert ist der leicht höhere durchschnittliche Recall der kleinen Modelle von 0,843 gegenüber den großen mit 0,839, wohingegen Precision und Accuracy annähernd identisch sind.

Tabelle 9.2: Kleine vs. große Modelle: Durchschnittswerte pro Gruppe und jeweils bestes Modell.

Metrik	Klein ($\leq 25B$)	Groß ($> 25B$)
Anzahl Modelle ¹	5	8
$\bar{\varnothing}$ F1-Score \pm SD ²	$0,805 \pm 0,057$	$0,806 \pm 0,089$
$\bar{\varnothing}$ Precision \pm SD	$0,774 \pm 0,050$	$0,779 \pm 0,085$
$\bar{\varnothing}$ Recall \pm SD	$0,843 \pm 0,086$	$0,839 \pm 0,128$
$\bar{\varnothing}$ Accuracy \pm SD	$0,744 \pm 0,067$	$0,749 \pm 0,099$
Bester F1-Score	0,866	0,874
Bestes Modell (F1-Score)	GPT-OSS-20B	Qwen3-235B-A22B-Thinking-2507
Bester Precision	0,829	0,892
Bestes Modell (Precision)	DeepSeek-R1-Distill-Qwen-14B	GPT-4o
Bester Recall	0,918	0,932
Bestes Modell (Recall)	GPT-OSS-20B	Qwen3-235B-A22B-Thinking-2507
Beste Accuracy	0,821	0,830
Bestes Modell (Accuracy)	GPT-OSS-20B	Qwen3-235B-A22B-Thinking-2507

¹ Einteilung nach gesamten Milliarden Parametern bei MoE. Die Proprietären Modelle GPT-4o und Mistral Medium 3.1 wurden trotz fehlender Parameterangabe als große Modelle eingeordnet.

² Ohne Mixtral-8x7B-Instruct-v0.1 beträgt der Durchschnitt der großen Modelle \pm SD 0,836 \pm 0,029.

Diese Ergebnisse bestätigen, dass die Modellgröße allein kein Garant für eine bessere Klassifikationsleistung ist. Kleinere Modelle wie GPT-OSS-20B liefern sehr starke Screening-Leistung bei geringeren Kosten und lassen sich leichter On-Premises² betreiben. In der Praxis sollte daher das Auswahlkriterium für ein Modell

²On-Premises bezeichnet den Betrieb von IT-Systemen im eigenen Rechenzentrum statt in der Cloud.

die *Balance aus Recall, Precision und Betriebskosten* sein und nicht die Parameterzahl.

Europäische vs. internationale Modelle

Tabelle 9.3 stellt die Mittelwerte der europäischen Mistral-Modelle den übrigen internationalen Modellen gegenüber. Die europäischen Modelle zeigen eine größere Streuung: das kommerzielle Mistral Medium 3.1 erfüllt mit einem F1-Score von 0,843 und Recall von 0,877 alle Zielkriterien und liegt knapp vor dem Referenzmodell GPT-4o. Ähnlich sieht es bei dem Open-Weight-Modell Mistral-Large-Instruct-2411 aus. Dagegen verfehlten Mistral-7B-Instruct-v0.3 und insbesondere Mixtral-8x7B-Instruct-v0.1 die Qualitätsziele deutlich. Im Durchschnitt bleiben die europäischen Modelle hinter den internationalen Spitzenreitern zurück. Letztere - allen voran Qwen3-235B-A22B-Thinking-2507 mit einem F1-Score von 0,874 und GPT-OSS-20B mit 0,866 - erreichen im Mittel einen höheren F1-Score sowie Recall und weisen eine geringere Varianz auf.

Tabelle 9.3: Europäische vs. internationale Modelle: Durchschnittswerte pro Gruppe und jeweils bestes Modell.

Metrik	EU-Modelle	Internationale Modelle
Anzahl Modelle	4	9
Ø F1-Score ± SD	0,760 ± 0,098	0,826 ± 0,045
Ø Precision ± SD	0,738 ± 0,061	0,797 ± 0,056
Ø Recall ± SD	0,789 ± 0,138	0,864 ± 0,076
Ø Accuracy ± SD	0,694 ± 0,101	0,771 ± 0,057
Bester F1-Score	0,843	0,874
Bestes Modell (F1-Score)	Mistral Medium 3.1	Qwen3-235B-A22B-Thinking-2507
Bester Precision	0,811	0,892
Bestes Modell (Precision)	Mistral Medium 3.1	GPT-4o
Bester Recall	0,877	0,932
Bestes Modell (Recall)	Mistral Medium 3.1	Qwen3-235B-A22B-Thinking-2507
Beste Accuracy	0,794	0,830
Bestes Modell (Accuracy)	Mistral Medium 3.1	Qwen3-235B-A22B-Thinking-2507

Die EU-Modelle umfassen die Mistral Modelle.

Diese Vergleiche belegen, dass ein europäischer Ursprung nicht zwangsläufig mit einer geringeren Leistung einhergeht - Mistral Medium 3.1 erreicht gute Werte,

dicht gefolgt von **Mistral-Large-Instruct-2411**. Allerdings zeigen die Ergebnisse auch, dass einige europäische Modelle hinter den internationalen Konkurrenten zurückbleiben. Insgesamt sind die internationalen Modelle im Durchschnitt leistungsfähiger und stabiler, da sie die europäischen Modelle in jeder Metrik im Durchschnitt übertreffen und eine geringere Varianz aufweisen. Zudem ist in jeder Metrik ein internationales Modell führend.

9.3 Robustheit

Die Robustheit der Modelle wird anhand zweier Kriterien bewertet: der Varianz der F1-Scores über die verschiedenen Seeds und der Anzahl der Retries, die erforderlich waren, um eine formatkorrekte JSON-Antwort von den LLMs in der Klassifizierungspipeline zu erhalten. Beide Größen geben Aufschluss darüber, wie stabil ein Modell im produktiven Einsatz ist.

Abbildung 9.3 zeigt die Standardabweichungen der F1-Scores über fünf unabhängige Läufe mit unterschiedlichen Seeds. Die Mehrzahl der Modelle weist Werte von deutlich unter 0,02 auf. Sie liefern damit weitgehend gleiche Ergebnisse, unabhängig vom gewählten Seed, und gelten als stabil. Dazu zählen **Gemma-3-12B-it**, **Mistral-Large-Instruct-2411**, **GPT-OSS-120B** und **DeepSeek-R1-Distill-Qwen-14B**.

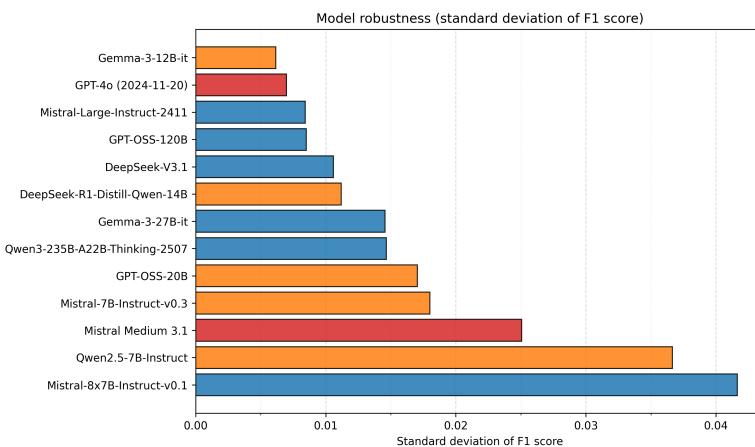


Abbildung 9.3: Robustheit der Modelle gemessen an der Standardabweichung des F1-Scores über alle Wiederholungen hinweg.

9 Ergebnisse

Demgegenüber weisen **Mistral Medium 3.1**, **Qwen2.5-7B-Instruct** und vor allem **Mixtral-8x7B-Instruct-v0.1** mit Standardabweichungen zwischen 0,025 und über 0,04 eine deutlich höhere Varianz auf. Dies bedeutet, dass ihre Leistung stärker vom gewählten Seed abhängt, was die Vergleichbarkeit und Zuverlässigkeit verringert.

Neben der Varianz des F1-Scores ist auch entscheidend, wie oft das Modell nachgefragt werden muss, bis eine gültige JSON-Struktur zurückgegeben wird. Abbildung 9.4 zeigt die durchschnittliche Anzahl notwendiger Retries pro Modell über 25 Testfälle. Die meisten Modelle lieferten bereits im ersten Versuch oder nach maximal einem zusätzlichen Aufruf eine korrekte Antwort. Hervorzuheben sind **DeepSeek-R1-Distill-Qwen-14B**, **GPT-4o**, **Gemma-3-12B-it** und **Mistral-Large-Instruct-2411**, die keinerlei Retries benötigten.

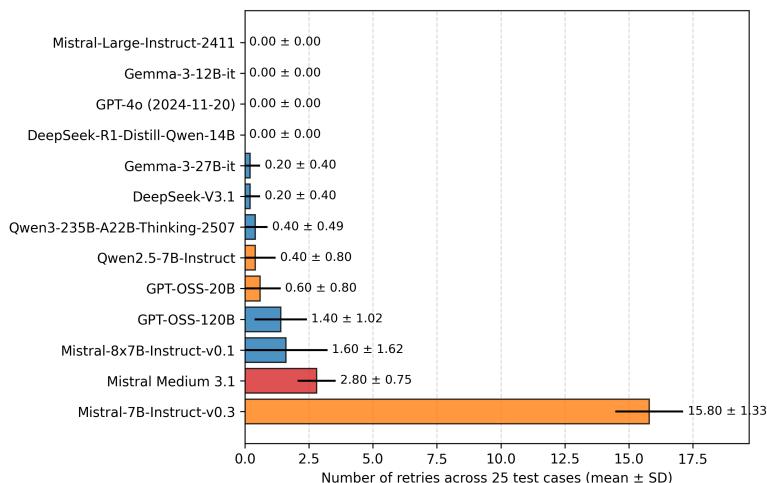


Abbildung 9.4: Durchschnittliche Anzahl der Retries, die notwendig waren, um für alle 25 Testfälle eine formatkorrekte JSON-Antwort zu erhalten.

Am anderen Ende des Spektrums steht **Mistral-7B-Instruct-v0.3**, das im Mittel 15,8 zusätzliche Aufrufe benötigte, was im Durchschnitt etwa 0,63 Retries pro Testfall entspricht. Diese hohe Zahl verdeutlicht, dass das Modell Schwierigkeiten hat, die vorgegebenen Formatierungsregeln zuverlässig einzuhalten.

In Kombination mit der geringen Varianz sind **Gemma-3-12B-it**, **Mistral-Large-Instruct-2411** und **DeepSeek-R1-Distill-Qwen-14B** die robustesten Modelle: Sie liefern konsistente Ergebnisse und halten das Ausgabeschema zuverlässig ein. Modelle wie **Mistral Medium 3.1**, **Qwen2.5-7B-Instruct** und **Mixtral-8x7B-**

Instruct-v0.1 sind hingegen anfälliger für Schwankungen und erfordern häufiger Wiederholungen.

9.4 Fallstudien

Die aggregierten Kennzahlen liefern einen guten Überblick über die Modellqualität, verbergen jedoch individuelle Fehlklassifikationen.

Zusätzlich vermittelt Abbildung 9.2 auf den ersten Blick den Eindruck, dass viele Modelle nur wenige der Testfälle bestehen. Eine genaue Analyse zeigt jedoch, dass hinter vielen dieser fehlgeschlagenen Testfälle lediglich ein oder wenige plausible FP stehen. Das bedeutet: Oft markiert das Modell eine zusätzliche Aktivität als kritisch, während alle tatsächlich kritischen Aktivitäten korrekt identifiziert werden. Solche Fälle lassen den Testfall formal als „nicht bestanden“ erscheinen, obwohl die Gesamtleistung des Modells durchaus hoch ist.

Im Folgenden verdeutlichen daher drei ausgewählte Fallstudien typische Muster von FP und FN.

Sales Warehouse

Der englische Prozess „Sales Warehouse“ aus dem Datensatz „Universität“ enthält vier als kritisch gelabelte Aktivitäten und ist in Abbildung 9.5 zu sehen. Das Modell Qwen3-235B-A22B-Thinking-2507 klassifiziert die gelabelten korrekt, markiert jedoch zusätzlich die Aktivität „Ship product“ als kritisch, was in Abbildung 9.5 als rot markierte Aktivität dargestellt ist. Die Begründung verweist auf die Nutzung der Kundenadresse für Versand und Zustellung. Obwohl während der Modellierung nur ein rein logistischer Schritt vorgesehen war - der laut Tabelle 5.2 in dem Labeling-Guide nicht kritisch ist - interpretiert das Modell den möglichen Datenfluss und wählt eine konservative Einstufung. Angesichts des Zielkriteriums eines hohen Recalls ist dieses FP vertretbar.

Das Beispiel verdeutlicht eine grundsätzliche Limitierung der Klassifizierung. Wenn in einem BPMN-Modell explizite Informationen über Datenverarbeitungen fehlen, ist es für das LLM schwierig, eine eindeutige Klassifikation vorzunehmen.

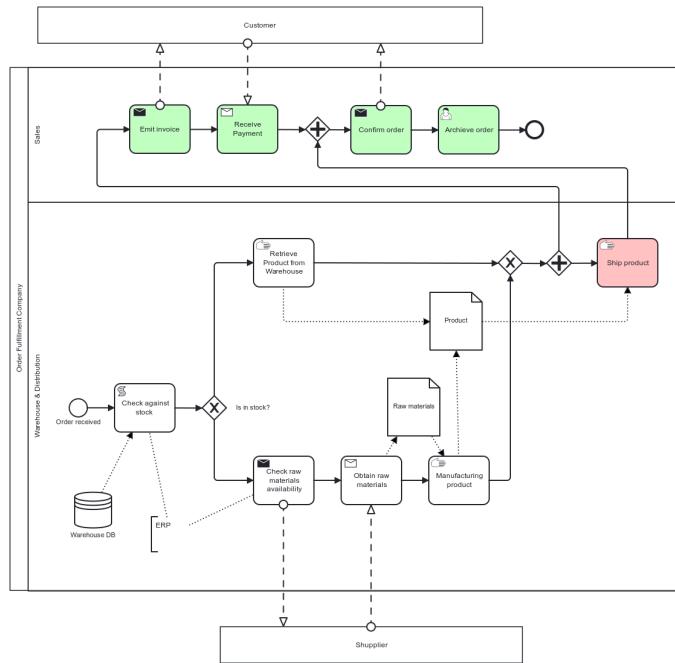


Abbildung 9.5: Ergebnis des Testfalls „Sales Warehouse“ mit farblich hervorgehobenen Aktivitäten. Grün markierte Aktivitäten sind korrekt als kritisch erkannt, rot markierte stellen FP dar.

Marketing-Kampagne

Im deutschen Testfall „Marketing-Kampagne“ aus dem Datensatz „Kleine Szenarien“ sind drei Aktivitäten als kritisch gelabelt: „Leads sammeln“, „Newsletter versenden“ und „CRM aktualisieren“. GPT-0SS-20B identifiziert diese korrekt, markiert aber zusätzlich die Aktivität „Klickraten auswerten“ als kritisch. Im Prozessmodell wird davon ausgegangen, dass die Klickdaten vollständig anonymisiert werden, jedoch ist diese Information nicht explizit hinterlegt. Mehrere Modelle, darunter Qwen3-235B-A22B-Thinking-2507, stufen diesen Schritt daher als potenziell personenbezogen ein, wie in Abbildung 9.6 zu sehen ist. Die Anonymisierung der Daten wurde nur von Mistral-7B-Instruct-v0.3 in zwei von fünf und von den Gemma-Modellen in allen Wiederholungen korrekt antizipiert und die Aktivität als unkritisch klassifiziert. Dieses Beispiel zeigt, wie fehlende Kontextinformationen zu vorsichtiger Klassifikation und damit zu FP führen können.

Dieses Beispiel zeigt, dass ohne genaue Kontextangaben zur Anonymisierung selbst scheinbar unbedenkliche Auswertungen als datenschutzrelevant erscheinen

9 Ergebnisse



Abbildung 9.6: Ergebnis des Testfalls „Marketing-Kampagne“ mit farblich hervorgehobenen Aktivitäten. Die Aktivität „Klickraten auswerten“ wurde als zusätzliches kritisches Element markiert.

können. Es unterstreicht, dass die LLMs im Zweifel eher ein kritisches Label vergeben, um FN zu vermeiden, wie es das Hauptziel der Klassifikation aus Abschnitt 3.2 vorsieht.

Karten-App - Standort erfassen

Der Testfall „Karten-App - Standort erfassen“ besteht aus zwei Aktivitäten: „Standort erfassen“ und „Route berechnen“. Beide sollten als kritisch klassifiziert werden, da im zweiten Schritt der erfasste Standort zur Berechnung der Route genutzt wird. Mistral-Large-Instruct-2411 kennzeichnet jedoch in drei von fünf Wiederholungen nur die erste Aktivität als kritisch, wie in Abbildung 9.7 zu sehen.

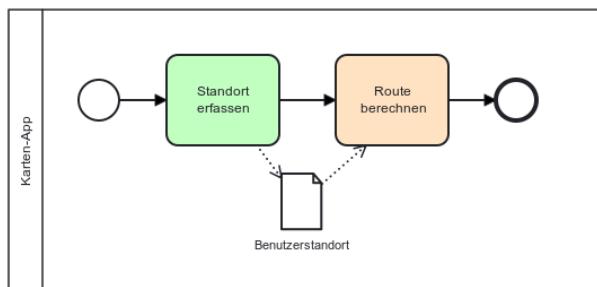


Abbildung 9.7: Ergebnis des Testfalls „Karten-App - Standort Erfassen“ mit farblich hervorgehobenen Aktivitäten. Die Aktivität „Route berechnen“ wurde fälschlicherweise nicht als kritisch markiert.

In der Begründung wird zwar die Verarbeitung personenbezogener Daten beim Erfassen des Standorts erkannt, dieser Zusammenhang aber nicht auf die nachfolgende Aktivität übertragen. Dieses FN steht dem angestrebten hohen Recall entgegen und zeigt, dass einige Modelle Schwierigkeiten haben, Datenflüsse über mehrere Schritte hinweg zu verfolgen.

Auf Basis der Erkenntnisse dieses gesamten Kapitels werden im folgenden Abschnitt die formulierten Forschungsfragen beantwortet. Dabei wird untersucht welches Modell sich insgesamt am besten für die Identifikation DSGVO-kritischer Aktivitäten eignet.

9.5 Beantwortung der Forschungsfragen

Auf Basis der vorhergehenden Auswertungen lassen sich die in Abschnitt 1.2 formulierten Forschungsfragen beantworten. Die folgenden Antworten berücksichtigen sowohl die quantitativen Ergebnisse als auch die qualitative Beobachtungen aus den Fallstudien und ordnen sie unter Berücksichtigung der Qualitätsziele ein.

UF1: Wie gut schneiden europäische Modelle im Vergleich zu internationalen Modellen ab? Die europäischen Modelle zeigen eine große Bandbreite in ihrer Leistungsfähigkeit. Mistral Medium 3.1 erfüllt mit einem F1-Score = 0,843, einem Recall = 0,877 und einer Precision = 0,811 sämtliche Qualitätsziele und übertrifft das Referenzmodell GPT-4o. Mistral-Large-Instruct-2411 erreicht mit einem F1-Score = 0,823 ebenfalls alle Zielwerte. Dagegen schneiden Mistral-7B-Instruct-v0.3 mit F1-Score = 0,777 und insbesondere Mixtral-8x7B-Instruct-v0.1 mit F1-Score = 0,596 deutlich schlechter ab. Im Durchschnitt liegen die internationalen Modelle - insbesondere die Qwen- und GPT-OSS-Varianten - vor den europäischen und bieten eine robustere Balance aus Recall und Precision. Dennoch zeigen Mistral Medium 3.1 und Mistral-Large-Instruct-2411, dass leistungsfähige europäische Alternativen existieren.

UF2: Wie unterscheiden sich große und kleine Modelle in ihrer Leistungsfähigkeit? Der direkte Vergleich zeigt, dass sich kleine ($\leq 25B$ Parameter) und große Modelle kaum im Durchschnitt ihrer Metriken unterscheiden. Beide Größenklassen erreichen praktisch identische mittlere F1-Scores von etwa 0,80. Ohne das Ausreißermodell Mixtral-8x7B-Instruct-v0.1 liegt der Durchschnitt der großen Modelle mit 0,836 zwar etwas höher, doch belegen Modelle wie GPT-OSS-20B, dass kleinere Modelle mit den großen mithalten können. Entscheidend sind Trainingsdaten, Feinabstimmung und Architektur, nicht allein die Parameteranzahl.

UF3: Welche Open-Source-Modelle (insbesondere aus der EU) erzielen die besten Ergebnisse? Unter den offenen Modellen dominieren die chinesischen Qwen-Varianten und die GPT-OSS-Modelle. Qwen3-235B-A22B-Thinking-2507 erreicht mit einem F1-Score = 0,874 und einem Recall = 0,932 die Spitzenposition, gefolgt von GPT-OSS-20B mit F1-Score = 0,866 und Recall = 0,918 und DeepSeek-R1-Distill-Qwen-14B mit F1-Score = 0,848 und Precision = 0,829. Diese Modelle übertreffen die proprietären Benchmarks deutlich. Das leistungsstärkste offene EU-Modell ist Mistral-Large-Instruct-2411 mit F1-Score = 0,823, während Mistral-7B-Instruct-v0.3 und Mixtral-8x7B-Instruct-v0.1 die Zielwerte verfehlten.

UF4: Wie gut schneiden Open-Source-Modelle gegenüber kommerziellen Modellen wie GPT-4o ab? Mehrere Open-Source-Modelle übertreffen die kommerziellen Vertreter. Qwen3-235B-A22B-Thinking-2507, GPT-OSS-20B und DeepSeek-R1-Distill-Qwen-14B erreichen höhere F1- und Recall-Werte als sowohl GPT-4o als auch Mistral Medium 3.1. GPT-4o überzeugt mit einer außergewöhnlich hohen Precision von 0,892, verfehlt aber das Recall-Mindestziel. Mistral Medium 3.1 bietet einen ausgewogenen Kompromiss und erfüllt alle Zielwerte, liegt aber hinter den besten Open-Source-Modellen. Insgesamt zeigen hochwertige Open-Source-Modelle die beste Balance zwischen hohem Recall und akzeptabler Precision.

Auf Basis der durchgeführten Experimente, Analysen und Antworten auf die Unterfragen lässt sich die Hauptforschungsfrage im Folgenden beantworten.

FF1: Wie zuverlässig identifizieren LLMs DSGVO-kritische Aktivitäten in BPMN-Prozessmodellen? Die überwiegende Mehrheit der Modelle kann kritische Aktivitäten mit hoher Zuverlässigkeit erkennen. Neun von dreizehn Modellen erreichen einen F1-Score von mindestens 0,80 und erfüllen damit den Zielwert. Die Spitzenmodelle Qwen3-235B-A22B-Thinking-2507, GPT-OSS-20B, DeepSeek-R1-Distill-Qwen-14B und Mistral Medium 3.1 erzielen F1-Scores zwischen 0,843 und 0,874 bei Recall-Werten von 0,868 bis 0,932. Gleichzeitig gibt es Modelle wie Mixtral-8x7B-Instruct-v0.1 und Qwen2.5-7B-Instruct, die deutlich abfallen.

9 Ergebnisse

Die Robustheitsanalyse zeigt, dass die meisten Modelle, mit einer Standardabweichung der F1-Scores von $\leq 0,02$, eine geringe Varianz über verschiedene Seeds aufweisen und häufig keine Retries benötigen, um eine korrekte JSON-Ausgabe zu produzieren. Ausreißer wie Mistral Medium 3.1 (höhere Varianz) oder Mistral-7B-Instruct-v0.3 (viele Retries) sollten im praktischen Einsatz sorgfältig überprüft werden.

Die Fallstudien unterstreichen, dass FP vor allem dann entstehen, wenn im Prozessmodell wichtige Kontextinformationen fehlen, wie z.B. Anonymisierung von Klickraten, und Modelle daher konservativ entscheiden. FN treten auf, wenn Datenflüsse über mehrere Aktivitäten nicht korrekt erkannt werden. Trotz dieser Fehlerbilder zeigen die Experimente, dass LLMs für ein automatisiertes Screening von BPMN-Prozessen sehr gut geeignet sind. Eine nachgelagerte menschliche Prüfung bleibt jedoch sinnvoll, um verbleibende FP und FN zu adressieren und die Ergebnisse kontextsensitiv zu bewerten.

10 Diskussion

Dieses Kapitel ordnet die empirischen Befunde kritisch ein und stellt ihre Tragweite für Forschung und Praxis heraus. Ausgangspunkt bilden die in Abschnitt 9.5 zusammengeführten Antworten auf *UF1-UF4* und *FF1*. Die Resultate werden in Bezug auf die Qualitätsziele aus Abschnitt 3.2 interpretiert, zentrale Trade-offs (Recall vs. Precision) herausgearbeitet und Besonderheiten der Modellklassen gegenübergestellt. Darauf aufbauend erfolgen eine vergleichende Einordnung der Modelle und ihrer Herkunft, eine Analyse von Robustheit und Pipeline-Einflüssen sowie eine Betrachtung typischer Fehlerbilder und Grenzen. Abschließend werden Implikationen für Anwendungsszenarien und zukünftige Forschung abgeleitet.

10.1 Einordnung und Interpretation

Die Experimente zeigen, dass moderne LLMs DSGVO-kritische Aktivitäten in BPMN-Prozessen zuverlässig identifizieren können. Neun von dreizehn Modellen erreichen den angestrebten F1-Score von $\geq 0,80$ und erfüllen damit die in Abschnitt 3.2 definierten Qualitätsziele für ein wirksames Screening, das *FN* minimiert, ohne den nachfolgenden Prüfaufwand durch *FP* unverhältnismäßig zu erhöhen. Diese Priorisierung auf den *Recall* ist sachlich begründet, da übersehene kritische Aktivitäten schwerwiegende rechtliche und finanzielle Konsequenzen nach sich ziehen können. Im Gegensatz dazu erhöhen *FP* zwar den Prüfaufwand, führen aber nicht zu direkten Verstößen gegen die DSGVO. Die Resultate zeigen ein heterogenes Bild: GPT-4o erzielt die höchste Precision mit 0,892, verfehlt jedoch das Recall-Ziel. Umgekehrt erreicht Gemma-3-27B-it einen sehr hohen Recall von 0,916, wird aber durch eine niedrige Precision von 0,687 ausgebremst. Modelle wie Qwen3-235B-A22B-Thinking-2507, GPT-OSS-20B und DeepSeek-R1-Distill-Qwen-14B bieten eine ausgewogene Balance und zählen deshalb zu den

Spitzenreitern. Die aggregierten Metriken in Abbildung 9.1 und Tabelle 9.1 bestätigen diese Einschätzung.

Ein Teil der Testfälle wurde formal als fehlerhaft gewertet, obwohl alle kritischen Aktivitäten korrekt klassifiziert worden sind. Es wurden jedoch zusätzliche Aktivitäten als kritisch markiert (FP) und mit plausiblen Begründungen versehen. Für ein Risiko-Vorscreening ist dieses Verhalten akzeptabel und sogar wünschenswert, solange die FP-Last den manuellen Prüfaufwand nicht unverhältnismäßig erhöht. Gleichzeitig zeigt dieses Muster die Notwendigkeit robuster Testdaten auf, die belastbare Labels enthalten: Statt Datensätze nachträglich an Modellausgaben anzupassen, empfiehlt sich für künftige Benchmarks ein Labeling mit mehreren unabhängigen Gutachtern, um subjektive Interpretationen zu minimieren und Grenzfälle klar zu definieren.

10.2 Modelle im Vergleich

Hinsichtlich der Herkunft sind einzelne europäische Modelle wettbewerbsfähig. Dazu gehören das proprietäre **Mistral Medium 3.1** mit F1-Score = 0,843, Recall 0,877 und Precision 0,811 sowie das Open-Source-Modell **Mistral-Large-Instruct-2411** mit F1-Score = 0,823 und Recall = 0,872. Im Durchschnitt liegen jedoch internationale Modelle vorne und zeigen eine geringere Varianz und höhere Robustheit. Daraus folgt, dass die Auswahl des Modells nicht allein auf die Herkunft gestützt werden sollte, sondern dass eine ganzheitliche Bewertung der Leistungsfähigkeit, Stabilität und Transparenz beim Training und Betrieb erfolgen muss.

Beim Vergleich *Open-Source vs. proprietär* übertreffen mehrere offene Modelle die proprietären Vertreter in F1-Score und Recall. GPT-4o überzeugt zwar durch eine sehr hohe Precision von 0,892, verfehlt aber das Recall-Mindestziel. Mistral Medium 3.1 bietet einen ausgewogenen Kompromiss und erfüllt alle Zielwerte, liegt aber hinter den besten Open-Source-Modellen. Für Recall-priorisierte Vorscreenings sind Qwen3-235B-A22B-Thinking-2507, GPT-OSS-20B und DeepSeek-R1-Distill-Qwen-14B die stärksten Kandidaten. Diese Ergebnisse unterstreichen, dass Open-Source-Modelle in spezialisierten Aufgaben konkurrenzfähig und proprietäre Lösungen nicht zwangsläufig überlegen sind.

Mit Blick auf die *Modellgröße* liegen kleine ($\leq 25B$) und große Modelle ($> 25B$) im Mittel nah beieinander. Kleine Modelle wie GPT-0SS-20B halten bei deutlich größeren Modellen mit oder übertreffen sie sogar. Zudem sind sie durch geringe Hardware-Anforderungen gut für On-Premises-Betrieb geeignet, was in datenschutzsensiblen Kontexten und im Hinblick auf Kosten von Vorteil ist. Die reine Parameteranzahl erweist sich damit nicht als hinreichendes Kriterium für die Modellauswahl. Vielmehr sind die Trainingsdaten, Feinabstimmung und Architektur der Modelle für ihre Leistung entscheidend. Für die Praxis bedeutet das: Die Modellauswahl sollte entlang der Zielmetrik (Recall-Priorität), dem erwarteten nachfolgenden Prüfungsaufwand (Precision) und den betrieblichen Rahmenbedingungen (Kosten, Hosting) erfolgen. Besonders wenn personenbezogene Daten verarbeitet werden, sind EU-Hosting und On-Premises-Betrieb wichtige Kriterien, die durch kleinere, leistungsfähige Modelle erleichtert werden.

10.3 Robustheit

Die Robustheitsanalyse über mehrere Seeds unterstreicht die Praxistauglichkeit der meisten Modelle in Kombination mit der entwickelten Klassifizierungspipeline. Für die Mehrzahl der LLMs liegt die Standardabweichung des F1-Scores über fünf Wiederholungen bei $\leq 0,02$, was auf eine geringe Varianz und konsistente Leistung hinweist. Vereinzelt zeigen Modelle eine höhere Varianz oder benötigen mehr Wiederholungen zur Korrektur von Parsing-Fehlern. Solche Unterschiede sind für den operativen Einsatz relevant, da sie sich direkt in Durchsatz, Latenz und Stabilität der Gesamtpipeline niederschlagen. Modelle mit erhöhter Varianz sollten daher im produktiven Betrieb sorgfältig überwacht und validiert werden, um unerwartete Leistungseinbußen zu vermeiden.

Wesentlich zur Zuverlässigkeit trägt die entwickelte Klassifizierungspipeline bei. *Structured Output* via Langchain4j mit explizitem JSON-Schema (und, wo verfügbar, API-seitig erzwungenem `response_format`) erhöht die Format-Treue, ein explizites `isRelevant`-Flag mit nachgelagertem Relevanz-Filter entschärft Widersprüche zwischen Begründung und Klassifikation, die *id-Validierung/-Vervollständigung* reduziert typische Ausgabefehler und der *Retry-Mechanismus* behebt Parsing-Fehler automatisiert. Diese Maßnahmen tragen wesentlich zur Ergebnis-

stabilität bei und sollten in produktiven Systemen implementiert werden, um die Zuverlässigkeit der Modellausgaben zu gewährleisten.

Ob das Preprocessing der Klassifizierungspipeline zur Leistung beiträgt, lässt sich nicht abschließend beurteilen. Die im nächsten Abschnitt beschriebenen Fallstudien legen nahe, dass trotz des im Preprocessing bereitgestellten Kontexts Datenflüsse und Prozesszusammenhänge durch LLM weiterhin unberücksichtigt bleiben oder falsch interpretiert werden. Hier könnten künftige Anpassungen der Pipeline ansetzen, um den Kontext für die Modelle weiter zu verbessern.

10.4 Fehlerbilder und Grenzen

Die Fallstudien verdeutlichen zwei zentrale Schwachstellen. Erstens führen *implizite Annahmen* zu konservativen, FP-lastigen Entscheidungen, wenn entsprechende Hinweise im BPMN-Modell nicht explizit modelliert sind. Das war beispielsweise bei der Anonymisierung von Klickraten im Testfall „Marketing-Kampagne“ der Fall. Zweitens ist die *Kontextverfolgung über Aktivitätsketten* nicht immer zuverlässig: Weiterverarbeitungen, die sich aus zuvor erhobenen Daten ergeben, werden vereinzelt nicht erkannt, was zu FN führt. Dies zeigte sich etwa im Testfall „Route berechnen“, in dem personenbezogene Standortdaten aus einer vorherigen Aktivität an die Routenberechnung explizit weitergegeben wurden, ohne dass das Modell dies als kritisch erkannte. In der Praxis lassen sich daraus konkrete Gegenmaßnahmen ableiten: Zum einen sollten BPMN-Modelle Datenflüsse sichtbarer machen, indem wichtige Elemente für den Kontext wie Datenobjekte/-speicher, Datenassoziationen, Nachrichtenflüsse, Annotationen sowie Pools/Lanes klar modelliert werden. Zum anderen sollten die Prompting-Strategien weiterentwickelt werden, um Modelle explizit auf die Bedeutung von Datenflüssen und Kontextinformationen hinzuweisen. Dazu zählt auch die Überprüfung, ob das Preprocessing der Klassifizierungspipeline ausreichend Kontext bereitstellt oder ob hier Anpassungen notwendig sind.

Die Generalisierbarkeit der Ergebnisse ist durch die Anzahl der Testfälle und deren Annotationen sowie durch die Auswahl der Modelle begrenzt. Zwar decken die 25 Testfälle eine breite Palette typischer DSGVO-kritischer Szenarien ab, doch können sie nicht alle denkbaren Prozessvarianten und -kontakte repräsentieren. Künf-

tige Studien sollten den Datensatz erweitern, um eine größere Vielfalt an Prozessen, Branchen und Komplexitätsgraden abzubilden. Dafür wurde in dieser Arbeit ein Labeling-Tool entwickelt. Zudem könnten weitere LLMs evaluiert werden, insbesondere solche mit neuen Architekturen oder Trainingsansätzen, um die Vergleichbarkeit zu erhöhen. Hinzu kommen technische Grenzen: Sehr große BPMN-Prozesse können die Token-Limits der Modelle überschreiten, was eine Anpassung der Pipeline erfordert würde, etwa durch Prozesssegmentierung. Das Preprocessing verringert bereits die Token-Anzahl, doch sind weitere Optimierungen denkbar.

Schließlich werden die Ergebnisse durch die spezifischen Prompting-Strategien in der Klassifizierungspipeline beeinflusst. So ist etwa im genutzten System-Prompt verankert, dass die Klassifikation bei unklaren Aktivitäten eher konservativ ausfallen soll. Alternative Ansätze könnten zu unterschiedlichen Ergebnissen führen, vor allem mit Blick auf die Balance zwischen FP und FN. Künftige Arbeiten sollten verschiedene Prompting-Techniken und Pipeline-Designs vergleichen, um die bestmögliche Leistung zu erzielen. Die Leistungsbewertung könnte zudem domänenabhängig sein: In anderen Bereichen bestehen möglicherweise nicht so strenge Anforderungen an Recall und Precision wie im DSGVO-Kontext. Abschließend ist zu betonen, dass die Klassifikation keine qualifizierte rechtliche Prüfung ersetzt, sondern als unterstützendes Werkzeug zur Risikominimierung dient.

11 Fazit

Diese Arbeit untersuchte, inwieweit moderne LLMs DSGVO-kritische Aktivitäten in BPMN-Prozessmodellen zuverlässig identifizieren können. Als Qualitätsziele galten Recall-Priorität und ein angestrebter F1-Score von $\geq 0,80$. In einer systematischen Evaluation mit 13 Modellen, 25 Testfällen und jeweils fünf Wiederholungen erreichten *neun* Modelle die Zielwerte. Offene Modelle wie Qwen3-235B-A22B-Thinking-2507 und GPT-0SS-20B kombinierten hohen Recall mit solider Precision. Das proprietäre Benchmark-Modell GPT-4o erzielte zwar die höchste Precision, verfehlte jedoch das Recall-Mindestziel. Kleinere Modelle ($\leq 25B$) hielten mit größeren mit und sind für On-Premises-Szenarien attraktiv. Europäische Modelle wie das proprietäre Mistral Medium 3.1 und das offene Mistral-Large-Instruct-2411 erwiesen sich als wettbewerbsfähig, lagen jedoch im Mittel im Vergleich zu internationalen Spitzenmodellen zurück. Die Robustheitsanalyse zeigte über die meisten Modelle hinweg geringe Seed-Varianzen und damit praxisrelevante Stabilität.

Ermöglicht wurden diese Ergebnisse durch die in dieser Arbeit entwickelte Infrastruktur: (1) eine Klassifizierungspipeline mit strukturiertem JSON-Output, id-Validierung-/Vervollständigung und automatischem Retry zur Erhöhung der Format-Treue und Reduzierung von Fehlern, (2) ein Labeling-Tool zur Erstellung und Pflege gelabelter BPMN-Testfälle über mehrere Domänen und Sprachen, sowie (3) ein Evaluationsframework mit deklarativer YAML-Konfiguration, standardisierter HTTP-Schnittstelle für Klassifizierungsalgorithmen und Frontend zur Visualisierung der Ergebnisse und Metriken. Diese Bausteine unterstützen belastbare, wiederholbare Experimente und einen fairen Modellvergleich und erlauben künftigen Arbeiten, neue Modelle, Prompting-Strategien, Klassifizierungsalgorithmen oder Testdaten nahtlos zu integrieren.

Die Fallstudien machten zwei wiederkehrende Fehlerbilder sichtbar: (1) konservative *FP*, wenn wesentlicher Kontext im Modell nicht explizit ist und (2) *FN*, wenn Datenflüsse über Aktivitätsketten nicht vollständig verfolgt werden. Für die Praxis folgt daraus, dass der Kontext und die Datenflüsse in BPMN-Modellen explizit modelliert werden sollten und ein Recall-orientiertes Vorscreening stets durch eine nachgelagerte fachliche Prüfung zu ergänzen ist.

Offen bleiben vor allem fünf Punkte:

1. Ausbau und Qualitätssteigerung der Testdaten durch weitere Domänen, Sprachen und Prozessvielfalt sowie robustere Labels (z. B. durch mehrere Personen),
2. systematische A/B-Vergleiche alternativer Pipeline-Varianten über die bestehende Schnittstelle, in denen z. B. andere Prompting-Strategien, Preprocessing-Methoden oder andere graphbasierte Kontextrepräsentationen getestet werden,
3. Integration von wissens- und regelgestützten Komponenten, wie z. B. nachgelagerte Konsistenz-/Entailment-Prüfungen oder RAG, zur Reduktion von Fehlklassifikationen,
4. Strategien für sehr große Prozesse, um Token-Limits zu umgehen - etwa durch Prozesssegmentierung - und
5. betriebliche Aspekte wie die Modellwahl unter Datenschutz- und Hostingvorgaben.

In Summe zeigt die Arbeit: LLMs bilden in Kombination mit einer robusten Pipeline ein wirksames, reproduzierbares Screening-Werkzeug und eine tragfähige Grundlage für weiterführende Forschung und Anwendungen. Internationale Spitzenmodelle lieferten die besten Ergebnisse; dennoch sind europäische sowie kleinere Modelle konkurrenzfähig - insbesondere unter On-Premises- und Datenschutzaforderungen. Die bereitgestellte Infrastruktur aus Pipeline, Labeling- und Evaluationsframework schafft eine belastbare Basis, auf der zukünftige Arbeiten aufbauen können und die den Einsatz von LLMs im Datenschutzkontext weiter voranbringen kann.

Literatur

- [1] European Data Protection Board (EDPB). *1.2 billion euro fine for Facebook as a result of EDPB binding decision*. Mai 2023. URL: https://www.edpb.europa.eu/news/news/2023/12-billion-euro-fine-facebook-result-edpb-binding-decision_en (besucht am 02.10.2025).
- [2] Simone Agostinelli u. a. „Achieving GDPR Compliance of BPMN Process Models“. In: Mai 2019, S. 10–22. ISBN: 978-3-030-21296-4. DOI: 10.1007/978-3-030-21297-1_2. URL: https://www.researchgate.net/publication/333312868_Achieving_GDPR_Compliance_of_BPMN_Process_Models.
- [3] DeepSeek AI. *DeepSeek AI Open Source Hugging Face Models*. 2025. URL: <https://huggingface.co/deepseek-ai> (besucht am 17.07.2025).
- [4] Mistral AI. *Mistral AI*. 2025. URL: <https://mistral.ai/> (besucht am 21.09.2025).
- [5] Mistral AI. *Mistral AI - Structured Output*. 2025. URL: https://docs.mistral.ai/capabilities/structured-output/structured_output_overview/ (besucht am 11.07.2025).
- [6] Ivan Belcic und Cole Stryker. *Was ist ein GPT (Generative Pre-Trained Transformer)*? Sep. 2024. URL: <https://www.ibm.com/de-de/think/topics/gpt> (besucht am 18.09.2025).
- [7] Dave Bergmann. *What is a context window?* 2025. URL: <https://www.ibm.com/think/topics/context-window> (besucht am 03.10.2025).
- [8] Mario Luca Bernardi u. a. „Conversing with Business Process-Aware Large Language Models: The BPLLM Framework“. In: *Journal of Intelligent Information Systems* 62.6 (2024), S. 1607–1629. URL: <https://link.springer.com/article/10.1007/s10844-024-00898-1>.

Literatur

- [9] Harrison Blake und Dorcas Esther. „Impact of Dataset Diversity on Model Evaluation Metrics“. In: (Jan. 2025). URL: https://www.researchgate.net/publication/387898702_Impact_of_Dataset_Diversity_on_Model_Evaluation_Metrics.
- [10] Tom Brown u. a. „Language Models are Few-Shot Learners“. In: *Advances in Neural Information Processing Systems*. Hrsg. von H. Larochelle u. a. Bd. 33. Curran Associates, Inc., 2020, S. 1877–1901. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418fb8ac142f64a-Paper.pdf.
- [11] Bundesministerium der Justiz. *Gesetz über außergerichtliche Rechtsdienstleistungen (Rechtsdienstleistungsgesetz - RDG)*. <https://www.gesetze-im-internet.de/rdg/>. Dez. 2007. (Besucht am 15.08.2025).
- [12] Camunda Services GmbH. *BPMN Model API*. <https://docs.camunda.org/manual/latest/user-guide/model-api/bpmn-model-api/>. 2025. (Besucht am 16.06.2025).
- [13] Camunda Services GmbH. *BPMN Model API — Read a Model*. <https://docs.camunda.org/manual/latest/user-guide/model-api/bpmn-model-api/read-a-model/>. 2025. (Besucht am 16.06.2025).
- [14] Antonio Capodieci u.a. „BPMN-Enabled Data Protection and GDPR Compliance“. In: *IS-EUD Workshops*. 2023. URL: <https://api.semanticscholar.org/CorpusID:259099646>.
- [15] Giovanni Ciaramella u.a. „Leveraging Pre-trained LLMs for GDPR Compliance in Online Privacy Policies“. In: (2022). URL: <https://ceur-ws.org/Vol-3962/paper44.pdf>.
- [16] Datenschutzticker. *Gericht bestätigt Rekordbußgeld gegen Amazon*. Apr. 2025. URL: <https://datenschutzticker.de/2025/04/gericht-bestatigt-rekordbussgeld-gegen-amazon/> (besucht am 02.10.2025).
- [17] DeepSeek-AI u.a. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. arXiv: 2501.12948 [cs.CL]. URL: <https://arxiv.org/abs/2501.12948>.

Literatur

- [18] Europäische Union. *Verordnung (EU) 2016/679 des Europäischen Parlaments und des Rates vom 27. April 2016 zum Schutz natürlicher Personen bei der Verarbeitung personenbezogener Daten, zum freien Datenverkehr und zur Aufhebung der Richtlinie 95/46/EG (Datenschutz-Grundverordnung)*. <https://eur-lex.europa.eu/legal-content/DE/TXT/?uri=CELEX:32016R0679>. 2016.
- [19] European Data Protection Board. *Guidelines 4/2019 on Article 25 Data Protection by Design and by Default*. https://www.edpb.europa.eu/sites/default/files/files/file1/edpb_guidelines_201904_dataprotection_by_design_and_by_default_v2.0_en.pdf. Version 2.0. Okt. 2020.
- [20] Camunda Services GmbH. *bpmn-js - BPMN 2.0 viewer and editor*. 2025. URL: <https://bpmn.io/toolkit/bpmn-js/> (besucht am 20.06.2025).
- [21] Camunda Services GmbH. *BPMN.io - Web-based tooling for BPMN, DMN and Forms*. 2025. URL: <https://bpmn.io/> (besucht am 22.09.2025).
- [22] Camunda Services GmbH. *Camunda Platform*. 2025. URL: <https://camunda.com/de/> (besucht am 22.09.2025).
- [23] Google. *Gemma 3 License Terms*. März 2025. URL: <https://ai.google.dev/gemma/terms> (besucht am 30.09.2025).
- [24] Dan Hendrycks u. a. *Measuring Massive Multitask Language Understanding*. 2021. arXiv: 2009.03300 [cs.CY]. URL: <https://arxiv.org/abs/2009.03300>.
- [25] Ashish Hooda u. a. *PolicyLR: A Logic Representation For Privacy Policies*. 2024. arXiv: 2408.14830 [cs.CR]. URL: <https://arxiv.org/abs/2408.14830>.
- [26] Hugging Face. *deepseek-ai/DeepSeek-R1-Distill-Qwen-14B — Model Card*. 2025. URL: <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-14B> (besucht am 30.09.2025).
- [27] Hugging Face. *deepseek-ai/DeepSeek-R1-Zero — Model Card*. 2025. URL: <https://huggingface.co/deepseek-ai/DeepSeek-R1-Zero> (besucht am 29.09.2025).

- [28] Hugging Face. *deepseek-ai/DeepSeek-V3.1 — Model Card*. 2025. URL: <https://huggingface.co/deepseek-ai/DeepSeek-V3.1> (besucht am 30.09.2025).
- [29] Hugging Face. *google/gemma-3-12b-it — Model Card*. 2025. URL: <https://huggingface.co/google/gemma-3-12b-it> (besucht am 30.09.2025).
- [30] Hugging Face. *google/gemma-3-27b-it — Model Card*. 2025. URL: <https://huggingface.co/google/gemma-3-27b-it> (besucht am 30.09.2025).
- [31] Hugging Face. *Hugging Face - The AI community building the future*. 2025. URL: <https://huggingface.co/> (besucht am 09.10.2025).
- [32] Hugging Face. *mistralai/Mistral-7B-Instruct-v0.2 — Model Card*. 2025. URL: <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2> (besucht am 30.09.2025).
- [33] Hugging Face. *mistralai/Mistral-Large-Instruct-2411 — Model Card*. 2025. URL: <https://huggingface.co/mistralai/Mistral-Large-Instruct-2411> (besucht am 30.09.2025).
- [34] Hugging Face. *mistralai/Mixtral-8x7B-Instruct-v0.1 — Model Card*. 2025. URL: <https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1> (besucht am 30.09.2025).
- [35] Hugging Face. *Qwen3-235B-A22B-Thinking-2507 — Model Card*. 2025. URL: <https://huggingface.co/Qwen/Qwen3-235B-A22B-Thinking-2507> (besucht am 30.09.2025).
- [36] Hugging Face. *unsloth/Qwen2.5-7B-Instruct*. 2025. URL: <https://huggingface.co/Qwen/Qwen2.5-7B-Instruct> (besucht am 30.09.2025).
- [37] Ziwei Ji u. a. „Survey of Hallucination in Natural Language Generation“. In: *ACM Comput. Surv.* 55.12 (März 2023). ISSN: 0360-0300. DOI: 10.1145/3571730. URL: <https://doi.org/10.1145/3571730>.
- [38] Adam Tauman Kalai u. a. *Why Language Models Hallucinate*. 2025. arXiv: 2509.04664 [cs.CL]. URL: <https://arxiv.org/abs/2509.04664>.
- [39] Humam Kourani u. a. „Evaluating Large Language Models on Business Process Modeling: Framework, Benchmark, and Self-Improvement Analysis“. In: *Software and Systems Modeling* (2025), S. 1–36. URL: <https://link.springer.com/article/10.1007/s10270-025-01318-w>.

- [40] Langchain4j. *Class OpenAiChatModel.OpenAiChatModelBuilder*. 2025. URL: <https://javadoc.io/doc/dev.langchain4j/langchain4j-openai/latest/dev/langchain4j/model/openai/OpenAiChatModel.OpenAiChatModelBuilder.html> (besucht am 14.06.2025).
- [41] Langchain4j. *LangChain4j Documentation 2025*. 2025. URL: <https://docs.langchain4j.dev/> (besucht am 14.06.2025).
- [42] Pengfei Liu u. a. „Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing“. In: *ACM Comput. Surv.* 55.9 (Jan. 2023). ISSN: 0360-0300. DOI: 10.1145/3560815. URL: <https://doi.org/10.1145/3560815>.
- [43] Christopher D. Manning, Prabhakar Raghavan und Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008. ISBN: 9780521865715. URL: <https://nlp.stanford.edu/IR-book/>.
- [44] Mathematical Association of America. *MAA Invitational Competitions: American Invitational Mathematics Examination (AIME)*. <https://maa.org/maa-invitational-competitions/>. Zugriff: 2025-10-19. MAA American Mathematics Competitions, 2025. (Besucht am 07.10.2025).
- [45] Meta. *Meta Llama 3 Community License Agreement*. 2024. URL: <https://www.llama.com/llama3/license/> (besucht am 30.09.2025).
- [46] Shervin Minaee u. a. *Large Language Models: A Survey*. 2025. arXiv: 2402.06196 [cs.CL]. URL: <https://arxiv.org/abs/2402.06196>.
- [47] Mistral AI. *Au Large: Announcing Mistral Large*. <https://mistral.ai/news/mistral-large>. Feb. 2024. (Besucht am 09.10.2025).
- [48] Mistral AI. *How can I exercise my GDPR rights?* 2025. URL: <https://help.mistral.ai/en/articles/347639-how-can-i-exercise-my-gdpr-rights> (besucht am 09.10.2025).
- [49] Mistral AI. *Mistral AI Research License (MRL-0.1)*. 2024. URL: <https://mistral.ai/static/licenses/MRL-0.1.md> (besucht am 05.10.2025).
- [50] Mistral AI. *Mixtral of Experts: Mixtral 8x7B*. 2023. URL: <https://mistral.ai/news/mixtral-of-experts> (besucht am 01.10.2025).
- [51] Mistral AI. *Models Benchmarks*. 2025. URL: <https://docs.mistral.ai/getting-started/models/benchmark/> (besucht am 09.10.2025).

Literatur

- [52] Mistral AI. *Models Overview*. 2025. URL: https://docs.mistral.ai/getting-started/models/models_overview/ (besucht am 09.10.2025).
- [53] Mistral AI. *Where do you store my data or my Organization's data?* 2025. URL: <https://help.mistral.ai/en/articles/347629-where-do-you-store-my-data-or-my-organization-s-data> (besucht am 09.10.2025).
- [54] Yida Mu u.a. „Navigating prompt complexity for zero-shot classification: a study of large language models in computational social science“. In: *Proceedings of LREC-COLING 2024*. European Language Resources Association, 2024. DOI: [10.48550/arXiv.2305.14310](https://doi.org/10.48550/arXiv.2305.14310). arXiv: 2305.14310.
- [55] Leonard Nake u.a. „Towards identifying gdpr-critical tasks in textual business process descriptions“. In: (2023). URL: <https://dl.gi.de/server/api/core/bitstreams/84ac5110-1a0f-4e3c-bdf8-6393555a7212/content>.
- [56] Maud Nalpas. *Understand LLM sizes*. Mai 2024. URL: <https://web.dev/articles/llm-sizes> (besucht am 03.10.2025).
- [57] Joshua Noble. *What is LLM Temperature?* URL: <https://www.ibm.com/think/topics/llm-temperature>.
- [58] OMG. *Business Process Model and Notation (BPMN)*. Version 2.0.2. Dez. 2013. URL: <https://www.omg.org/spec/BPMN/2.0.2/PDF> (besucht am 03.06.2025).
- [59] Open Source Initiative. *The Open Source Definition*. 2006. URL: <https://opensource.org/osd> (besucht am 05.10.2025).
- [60] OpenAI. *ChatGPT ist da*. Nov. 2022. URL: <https://openai.com/de-DE/index/chatgpt/> (besucht am 19.10.2025).
- [61] OpenAI. *Function calling and other API updates*. <https://openai.com/index/function-calling-and-other-api-updates/>. 2023. (Besucht am 10.07.2025).
- [62] OpenAI. *gpt-oss-120b & gpt-oss-20b Model Card*. 2025. URL: https://cdn.openai.com/pdf/419b6906-9da6-406c-a19d-1bb078ac7637/oai_gpt-oss_model_card.pdf (besucht am 02.10.2025).

Literatur

- [63] OpenAI. *Hello GPT-4o*. Mai 2024. URL: <https://openai.com/index/hello-gpt-4o/> (besucht am 21.07.2025).
- [64] OpenAI. *Introducing gpt-oss*. 2025. URL: <https://openai.com/index/introducing-gpt-oss/> (besucht am 02.10.2025).
- [65] OpenAI. *Model Overview*. 2025. URL: <https://platform.openai.com/docs/models> (besucht am 18.09.2025).
- [66] OpenAI. *OpenAI - Structured model outputs*. URL: https://docs.mistral.ai/capabilities/structured-output/structured_output_overview/ (besucht am 11.07.2025).
- [67] OpenRouter. *The Unified Interface For LLMs*. 2025. URL: <https://openrouter.ai> (besucht am 09.10.2025).
- [68] KC Pragyan u. a. „Toward Regulatory Compliance: A few-shot Learning Approach to Extract Processing Activities“. In: *2024 IEEE 32nd International Requirements Engineering Conference Workshops (REW)*. IEEE. 2024, S. 241–250. URL: <https://ieeexplore.ieee.org/abstract/document/10628578>.
- [69] Quarkiverse Contributors. *AI Services Reference (Quarkus LangChain4j)*. URL: <https://docs.quarkiverse.io/quarkus-langchain4j/dev/ai-services.html> (besucht am 14.06.2025).
- [70] Alibaba Qwen. *Qwen Open Source Hugging Face Models*. 2025. URL: [http://huggingface.co/Qwen](https://huggingface.co/Qwen) (besucht am 17.07.2025).
- [71] Qwen u. a. *Qwen2.5 Technical Report*. 2025. arXiv: 2412.15115 [cs.CL]. URL: <https://arxiv.org/abs/2412.15115>.
- [72] Nils Reimers und Iryna Gurevych. „Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging“. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Hrsg. von Martha Palmer, Rebecca Hwa und Sebastian Riedel. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, S. 338–348. DOI: 10.18653/v1/D17-1035. URL: <https://aclanthology.org/D17-1035/>.

- [73] Matthew Renze und Erhan Guven. „The effect of sampling temperature on problem solving in large language models“. In: *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024, S. 7346–7356. DOI: 10.48550/arXiv.2402.05201. arXiv: 2402.05201.
- [74] Reuters. *Amazon hit with record EU data privacy fine*. Juli 2021. URL: https://www.reuters.com/business/retail-consumer/amazon-hit-with-886-million-eu-data-privacy-fine-2021-07-30/?utm_source=chatgpt.com (besucht am 02.10.2025).
- [75] David Rodriguez u. a. „Large Language Models: A New Approach for Privacy Policy Analysis at Scale“. In: *Computing* 106.12 (2024), S. 3879–3903. DOI: 10.1007/s00607-024-01331-9. URL: <https://doi.org/10.1007/s00607-024-01331-9>.
- [76] Keisuke Sakaguchi u. a. „WinoGrande: an adversarial winograd schema challenge at scale“. In: *Commun. ACM* 64.9 (Aug. 2021), S. 99–106. ISSN: 0001-0782. DOI: 10.1145/3474381. URL: <https://doi.org/10.1145/3474381>.
- [77] Konrad Schneid u. a. „Uncovering data-flow anomalies in BPMN-based process-driven applications“. In: *Proceedings of the 36th Annual ACM Symposium on Applied Computing*. 2021, S. 1504–1512. URL: <https://dl.acm.org/doi/abs/10.1145/3412841.3442025>.
- [78] Torsten Scholak, Nathan Schucher und Dzmitry Bahdanau. „PICARD: Parsing Incrementally for Constrained Auto-Regressive Decoding from Language Models“. In: *CoRR* abs/2109.05093 (2021). arXiv: 2109.05093. URL: <https://arxiv.org/abs/2109.05093>.
- [79] Magdalena von Schwerin und Manfred Reichert. „A systematic comparison between open-and closed-source large language models in the context of generating gdpr-compliant data categories for processing activity records“. In: *Future Internet* 16.12 (2024), S. 459. URL: <https://www.mdpi.com/1999-5903/16/12/459>.
- [80] Bhanuka Silva u. a. „Entailment-Driven Privacy Policy Classification with LLMs“. In: *2024 Conference on Building a Secure & Empowered Cyberspace (Build-*

- SEC). 2024, S. 8–15. DOI: 10.1109/BuildSEC64048.2024.00010. URL: <https://ieeexplore.ieee.org/document/10874334>.
- [81] Marina Sokolova und Guy Lapalme. „A systematic analysis of performance measures for classification tasks“. In: *Information processing & management* 45.4 (2009), S. 427–437. URL: <https://www.sciencedirect.com/science/article/pii/S0306457309000259>.
- [82] Mirac Suzgun u.a. *Challenging BIG-Bench Tasks and Whether Chain-of-Thought Can Solve Them*. 2022. arXiv: 2210.09261 [cs.CL]. URL: <https://arxiv.org/abs/2210.09261>.
- [83] Ángel Jesús Varela-Vaca u.a. „Business process models and simulation to enable GDPR compliance“. In: *International Journal of Information Security* 24.1 (2025), S. 41. URL: <https://link.springer.com/article/10.1007/s10207-024-00952-7>.
- [84] Ashish Vaswani u.a. „Attention Is All You Need“. In: *CoRR* abs/1706.03762 (2017). arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762>.
- [85] Maxim Vidgof, Stefan Bachhofner und Jan Mendling. *Large Language Models for Business Process Management: Opportunities and Challenges*. 2023. arXiv: 2304.04309 [cs.SE]. URL: <https://arxiv.org/abs/2304.04309>.
- [86] Yubo Wang u.a. *MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark*. 2024. arXiv: 2406.01574 [cs.CL]. URL: <https://arxiv.org/abs/2406.01574>.
- [87] Rowan Zellers u.a. „HellaSwag: Can a Machine Really Finish Your Sentence?“ In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Hrsg. von Anna Korhonen, David Traum und Lluís Màrquez. Florence, Italy: Association for Computational Linguistics, Juli 2019, S. 4791–4800. DOI: 10.18653/v1/P19-1472. URL: <https://aclanthology.org/P19-1472/>.

Quelltexte

In diesem Anhang sind mehrere Quellcode-Ausschnitte aufgeführt.

Listing 1: System-Prompt der DSGVO-Klassifikation von BPMN-Aktivitäten

- 1 You are an expert in analysing Business Process Model and Notation (→ BPMN) diagrams for GDPR compliance. Your task is to identify and → return a list of the IDs of all Activity (Task) elements that → process personal **data**. Ignore all other element types. Always → consider every activity in the process; do not omit any activity → from your assessment.
- 2
- 3 Use all available context for each activity - including the activity's → name, description, annotations, associated **data** objects, and → message or **data** associations - to determine whether the activity → processes personal **data**. Under Article 4 of the GDPR, personal → **data** is any information relating to an identified or identifiable → natural person, including names, addresses, email addresses, → phone numbers, identification numbers, payment or bank details, → employment records, academic records, location **data**, IP addresses → , online identifiers, images, audio/video recordings, biometric → identifiers, health **data** or other information that can be linked → to a specific person. "Processing" includes any operation → performed on personal **data**, such as collecting, recording, → organising, structuring, storing, retrieving, consulting, using, → analysing, transmitting, printing, disseminating, aligning, → combining, altering, restricting, erasing or destroying the **data**.
- 4
- 5 Classify an activity as GDPR-relevant whenever it performs or enables → processing of personal **data**. Indicators include (but are not → limited to):

- 6
- 7 - **Collection and entry of personal data**: Activities that collect
 ↳ or capture personal information, for example entering contact
 ↳ details, addresses, payment information, job applications, health
 ↳ information, student enrolments, membership **data**, tax
 ↳ declarations, registration forms or other forms with personally
 ↳ identifiable information.
- 8 - **Creation, storage and updating of records**: Activities that
 ↳ create, save or update records containing personal **data**, such as
 ↳ opening customer accounts, storing order or appointment details,
 ↳ creating personnel files, enrolling students, setting up
 ↳ insurance cases or filing a medical record.
- 9 - **Transmission or disclosure of personal data**: Activities that
 ↳ send, print or otherwise disclose personal **data** to another
 ↳ participant, system or third party. Examples include printing
 ↳ shipping labels or prescriptions, sending orders or personal **data**
 ↳ to logistics partners, pharmacies, insurers or authorities,
 ↳ generating payroll reports for external providers, notifying
 ↳ universities about student records, transmitting tax or social
 ↳ security **data**, sending confirmations or queries that rely on a
 ↳ person's contact details, or transferring **data** to non-EU
 ↳ locations.
- 10 - **Payments and financial transactions**: Activities that process
 ↳ personal financial **data**, such as initiating or verifying payments
 ↳ , processing bank account or credit-card information, executing
 ↳ payroll, handling reimbursements or insurance payouts, managing
 ↳ expense claims or collecting membership fees.
- 11 - **Use of health, biometric or other special categories of data**:
 ↳ Activities that handle medical diagnoses, prescriptions,
 ↳ insurance claims, disability information, photos of damages or
 ↳ patients, biometric identifiers (fingerprints, facial images,
 ↳ voice), racial or ethnic **data**, political opinions, religious
 ↳ beliefs or union membership. Processing these "special categories
 ↳ " always triggers GDPR relevance.
- 12 - **Audio/Video and communications**: Activities that initiate or join
 ↳ audio or video calls, record calls or meetings, capture
 ↳ surveillance footage, or communicate directly with a **data** subject

- via email, chat, SMS or other channels. Simply using a person's
 - contact **data** to send reminders, marketing messages or
 - notifications is processing.
- 13 - ****Profiling, scoring and decision-making**:** Activities that analyse
 → or evaluate a person's performance, behaviour or characteristics
 → for purposes such as credit scoring, hiring, admissions,
 → insurance underwriting, marketing segmentation, customer value
 → analysis or automated decision-making.
- 14 - ****Logging, tracking and location **data****:** Activities that log user
 → activity, record access or usage **data**, track geolocation (e.g.
 → telematics, fleet or mobile tracking), monitor attendance or
 → timekeeping, or collect IP addresses or device identifiers.
- 15 - ****Consent and **data**-subject rights**:** Activities that obtain, record
 → or manage consent; respond to requests for access, rectification,
 → restriction, erasure, **data** portability or objections; or
 → document lawful bases for processing.
- 16 - ****Deletion, anonymisation or pseudonymisation**:** Activities that
 → erase, anonymise or pseudonymise personal **data**, even if the goal
 → is to remove identifiers, because these operations manipulate
 → personal **data**.
- 17
- 18 When assessing an activity, consider synonyms or domain-specific terms
 → : activities referring to customers, patients, applicants,
 → employees, students, voters, taxpayers, residents or members
 → often imply personal **data** processing, even if names like "address
 → " or "contact" are absent. Use context - **data** objects,
 → annotations or typical process semantics - to infer personal **data**
 → involvement. Do not rely solely on explicit **data-object** links;
 → many process names ("Anmeldung pruefen", "Aufnahmeantrag
 → bearbeiten", "Kundeninfo aktualisieren", "Registrierung
 → bestaetigen", "Kreditwuerdigkeit berechnen") themselves indicate
 → personal **data** processing.
- 19
- 20 Do **not** classify an activity as GDPR-relevant when it only performs
 → administrative or logistic tasks that do not involve personal
 → **data**. Examples include picking or packing goods, routing vehicles
 → without using specific addresses, printing generic pick lists,

```
→ moving items in inventory, or checking if a document exists  
→ without viewing its contents. Likewise, activities using truly  
→ aggregated or irreversibly anonymised data can be ignored if no  
→ individual can be reidentified.  
21  
22 In your output, return only the IDs of activities you classify as GDPR  
→ -relevant. For each, provide a clear explanation in englisch  
→ using the activity's name and description to justify why it  
→ processes personal data. Do not reference element IDs in your  
→ explanation; use the activity names instead. Exclude from your  
→ result any activities that do not process personal data and any  
→ elements that are not activity/task elements.
```

Listing 2: Antworttyp der Klassifizierung

```
1  data class BpmnAnalysisResult(  
2      @Description("List of Activity Elements that are classified as  
3      → relevant for GDPR compliance")  
4      var elements: List<Element>  
5  ) {  
6  
7      init {  
8          elements = elements.filter { it.isRelevant }  
9      }  
10  
11      @Description("Represents an Activity/Task Element that is  
12      → classified as relevant for GDPR compliance")  
13      data class Element(  
14          @Description("The ID of the Activity Element")  
15          val id: String,  
16          @Description("The detailed reason why the Activity Element is  
17          → relevant for GDPR compliance and why you think personal data is  
18          → processed.")  
19          val reason: String,  
20          @Description("Indicates whether the Activity Element is  
21          → relevant for GDPR compliance")  
22          val isRelevant: Boolean = true  
23      )
```

```

19
20     /* Andere Methoden dieser Klasse sind weggelassen */
21 }
```

Listing 3: Kern der id-Validierung und -Vervollständigung

```

1 fun resolveActivityIds(actualBpmnElements: Set<BpmnElement>):
2      $\hookrightarrow$  BpmnAnalysisResult {
3         val existingActivityIds = actualBpmnElements
4             .filter { it.type.lowercase().contains("task") }
5             .map { it.id }.toSet()
6
7         val resolvedDistinct = elements.mapNotNull { element ->
8             val resolvedId = resolveActivityIdUniquely(element.id,
9                  $\hookrightarrow$  existingActivityIds)
10            resolvedId?.let { if (it == element.id) element else element.
11                 $\hookrightarrow$  copy(id = it) }
12            }.distinctBy { it.id }
13
14        return BpmnAnalysisResult(elements = resolvedDistinct)
15    }
16
17    private fun resolveActivityIdUniquely(partialId: String,
18         $\hookrightarrow$  existingActivityIds: Set<String>): String? {
19        if (partialId in existingActivityIds) return partialId
20        existingActivityIds.filter { it.startsWith(partialId) }.
21             $\hookrightarrow$  singleOrNull()?.let { return it }
22        return existingActivityIds.filter { it.contains(partialId) }.
23             $\hookrightarrow$  singleOrNull()
24    }
25 }
```

Listing 4: Schema der YAML-Evaluationskonfiguration

```

1 {
2     "$schema": "https://json-schema.org/draft/2020-12/schema",
3     "$ref": "#/definitions/Configuration",
4     "definitions": {
5         "Configuration": {
6             "type": "object",
```

```
7      "additionalProperties": false,
8      "properties": {
9          "defaultEvaluationEndpoint": {
10              "type": "string"
11          },
12          "maxConcurrent": { "type": "integer" },
13          "repetitions": { "type": "integer" },
14          "models": {
15              "type": "array",
16              "items": { "$ref": "#/definitions/Model" }
17          },
18          "datasets": {
19              "type": "array",
20              "items": { "type": "integer" }
21          },
22          "seed": { "type": "integer" }
23      },
24      "required": [
25          "defaultEvaluationEndpoint",
26          "models",
27          "datasets",
28      ],
29      "title": "Configuration"
30  },
31  "Model": {
32      "type": "object",
33      "additionalProperties": false,
34      "properties": {
35          "label": { "type": "string" },
36          "evaluationEndpoint": { "type": "string" },
37          "llmProps": { "$ref": "#/definitions/LlmProps" }
38      },
39      "required": [ "label" ],
40      "title": "Model"
41  },
42  "LlmProps": {
43      "type": "object",
```

```
44         "additionalProperties": false,
45         "properties": {
46             "baseUrl": {
47                 "type": "string",
48                 "format": "uri",
49                 "qt-uri-protocols": [ "https" ]
50             },
51             "modelName": { "type": "string" },
52             "apiKey": { "type": "string" },
53             "timeoutSeconds": { "type": "number" },
54             "temperature": { "type": "number" },
55             "topP": { "type": "number" },
56         },
57         "required": [],
58         "title": "LlmProps"
59     }
60 }
61 }
```

Listing 5: Zusammengefasster Logauszug zum Retry-Mechanismus

```
1 2025-10-03T19:11:51.152+02:00  INFO  BpmnExtractor  : Extracting BPMN
   ↳ elements from XML
2
3 # 1) Erste Anfrage an das LLM (gekürzt: Prompt/Headers/Body)
4 2025-10-03T19:11:51.156+02:00  INFO  LoggingHttpClient  : HTTP POST
   ↳ https://openrouter.ai/api/v1/chat/completions
   model: openai/gpt-oss-20b
   messages: [system: (System-Prompt), user: (User-Prompt mit
   ↳ BpmnElement-Liste und Format-Anweisung)]
7
8 # 2) Antwort des LLM mit fehlerhaftem JSON (verkürzt)
9 2025-10-03T19:11:56.671+02:00  INFO  LoggingHttpClient  : HTTP 200
10 assistant:
11 {
12     "elements": [
13         { "id": "Activity_09ehuui", "reason": "...", "isRelevant": true },
14         { "id": "Activity_1la5hsp", "reason": "...", "isRelevant":    }
```

```
15    ↳ <-- fehlender Bool-Wert
16      { "id": "Activity_0rfgrlm", "reason": "...", "isRelevant": true }
17    ]
18
19 # 3) Parser-Fehler + Retry-Ankündigung (gekürzt)
20 2025-10-03T19:11:56.691+02:00  WARN  SafetyNet  : Parsing failed.
21     ↳ Attempting to fix JSON and retry... (Attempt 1 of 2)
22 dev.langchain4j.service.output.OutputParsingException:
23     Caused by: com.fasterxml.jackson.core.JsonParseException:
24     Unexpected character ('}') ... at elements[1].isRelevant
25
26 # 4) Zweite Anfrage zum beheben des JSON mit Chat-Verlauf und
27     ↳ Fehlermeldung (n-mal wiederholt, bis erfolgreich)
28 2025-10-03T19:11:56.721+02:00  INFO  LoggingHttpClient  : HTTP POST
29     ↳ https://openrouter.ai/api/v1/chat/completions
30 messages: [
31   system: (System-Prompt),
32   user: (User-Prompt mit BpmnElement-Liste und Format-Anweisung),
33   assistant: (Fehlerhafte JSON-Antwort),
34   system: (Fix-JSON System-Prompt),
35   user: (Fehlermeldung)
36 ]
37
38 # 5) Korrigierte JSON-Antwort des LLM
39 2025-10-03T19:12:01.519+02:00  INFO  LoggingHttpClient  : HTTP 200
40 assistant:
41 {
42   "elements": [
43     { "id": "Activity_09ehuui", "reason": "...", "isRelevant": true },
44     { "id": "Activity_1la5hsp", "reason": "...", "isRelevant": true },
45     ↳ <-- jetzt mit Bool-Wert
46     { "id": "Activity_0rfgrlm", "reason": "...", "isRelevant": true }
47   ]
48 }
49
50 # 6) Erfolgreiches Parsing und Weiterverarbeitung
```

```
47 2025-10-03T19:12:01.519+02:00 INFO PromptBpmnAnalyzer : BPMN  
→ Analysis Result: elements=[... isRelevant=true ...]
```

Listing 6: Konfigurationsdatei vom Experiment mit Gemma Modellen

```
1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering  
2 seed: 24523833  
3 maxConcurrent: 10  
4 repetitions: 5  
5 models:  
6   - label: Gemma-3-12B-bit  
7     llmProps:  
8       baseUrl: https://openrouter.ai/api/v1  
9       modelName: google/gemma-3-12b-bit  
10      apiKey: ${OPEN_ROUTER_API_KEY}  
11      temperature: 0.1  
12      topP: 1  
13   - label: Gemma-3-27B-bit  
14     llmProps:  
15       baseUrl: https://openrouter.ai/api/v1  
16       modelName: google/gemma-3-27b-bit  
17       apiKey: ${OPEN_ROUTER_API_KEY}  
18       temperature: 0.1  
19       topP: 1  
20 datasets:  
21   - 2  
22   - 7  
23   - 1
```

Listing 7: Konfigurationsdatei vom Experiment mit DeepSeek Modellen

```
1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering  
2 seed: 24523833  
3 maxConcurrent: 10  
4 repetitions: 5  
5 models:  
6   - label: DeepSeek-V3.1  
7     llmProps:  
8       baseUrl: https://openrouter.ai/api/v1
```

```
9     modelName: deepseek/deepseek-chat-v3.1
10    apiKey: ${OPEN_ROUTER_API_KEY}
11    temperature: 0.1
12    topP: 1
13  - label: DeepSeek-R1-Distill-Qwen-14B
14    llmProps:
15      baseUrl: https://openrouter.ai/api/v1
16      modelName: deepseek/deepseek-r1-distill-qwen-14b
17      apiKey: ${OPEN_ROUTER_API_KEY}
18      temperature: 0.1
19      topP: 1
20  datasets:
21    - 2
22    - 7
23    - 1
```

Listing 8: Konfigurationsdatei vom Experiment mit Qwen Modellen

```
1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering
2 seed: 24523833
3 maxConcurrent: 10
4 repetitions: 5
5 models:
6   - label: Qwen2.5-7B-Instruct
7     llmProps:
8       baseUrl: https://openrouter.ai/api/v1
9       modelName: qwen/qwen-2.5-7b-instruct
10      apiKey: ${OPEN_ROUTER_API_KEY}
11      temperature: 0.1
12      topP: 1
13   - label: Qwen3-235B-A22B-Thinking-2507
14     llmProps:
15       baseUrl: https://openrouter.ai/api/v1
16       modelName: qwen/qwen3-vl-235b-a22b-thinking
17       apiKey: ${OPEN_ROUTER_API_KEY}
18       temperature: 0.1
19       topP: 1
20 datasets:
```

```
21 - 2  
22 - 7  
23 - 1
```

Listing 9: Konfigurationsdatei vom Experiment mit GPT Modellen

```
1 defaultEvaluationEndpoint: /gdpr/analysis/prompt-engineering  
2 seed: 24523833  
3 maxConcurrent: 10  
4 repetitions: 5  
5 models:  
6   - label: GPT-OSS-20B  
7     llmProps:  
8       baseUrl: https://openrouter.ai/api/v1  
9       modelName: openai/gpt-oss-20b  
10      apiKey: ${OPEN_ROUTER_API_KEY}  
11      temperature: 0.1  
12      topP: 1  
13   - label: GPT-OSS-120B  
14     llmProps:  
15       baseUrl: https://openrouter.ai/api/v1  
16       modelName: openai/gpt-oss-120b  
17       apiKey: ${OPEN_ROUTER_API_KEY}  
18       temperature: 0.1  
19       topP: 1  
20   - label: GPT-4o (2024-11-20)  
21     llmProps:  
22       baseUrl: https://openrouter.ai/api/v1  
23       modelName: openai/gpt-4o-2024-11-20  
24       apiKey: ${OPEN_ROUTER_API_KEY}  
25       temperature: 0.1  
26       topP: 1  
27 datasets:  
28   - 2  
29   - 7  
30   - 1
```

Abbildungsverzeichnis

2.1	Die relevanten BPMN-Elemente in Beziehungen zueinander.	7
2.2	Beispiel einer Datenassoziation als Datenschutzsignal.	9
3.1	Beispielprozess zur Veranschaulichung der Aufgabenstellung.	16
4.1	BPMN-Diagramm der Klassifizierungspipeline.	24
4.2	Sandbox im Frontend mit hervorgehobenen kritischen Aktivitäten nach Analyse.	35
4.3	Exemplarische Begründungen der Klassifikation durch das LLM. . . .	36
5.1	Labeling-Editor im Labeling-Modus mit exemplarischem Modell. . . .	39
5.2	Übersicht der Datensätze im Labeling-Tool.	40
6.1	Architektur des Evaluationsframeworks.	50
6.2	Formular zur Konfiguration einer Evaluation.	52
6.3	Gesamtübersicht einer Evaluierung mit aggregierten Metriken über alle Wiederholungen.	53
6.4	Ergebnisse pro Wiederholung mit exemplarischen Resultaten. . . .	54
6.5	Modell-Detailansicht mit exemplarischen Ergebnissen.	55
6.6	Detailseite eines Testfalls mit exemplarischen Ergebnissen. . . .	56
9.1	Durchschnittliche Werte für Precision, Recall, F1-Score und Accuracy der untersuchten Modelle über alle Wiederholungen hinweg inklusive Standardabweichung.	71
9.2	Durchschnittliche Testergebnisse pro Modell in Bezug auf die 25 Testfälle über fünf Wiederholungen hinweg.	73
9.3	Robustheit der Modelle gemessen an der Standardabweichung des F1-Scores über alle Wiederholungen hinweg.	77

Abbildungsverzeichnis

9.4 Durchschnittliche Anzahl der Retries, die notwendig waren, um für alle 25 Testfälle eine formatkorrekte JSON-Antwort zu erhalten.	78
9.5 Ergebnis des Testfalls „Sales Warehouse“ mit farblich hervorgehobenen Aktivitäten. Grün markierte Aktivitäten sind korrekt als kritisch erkannt, rot markierte stellen FP dar.	80
9.6 Ergebnis des Testfalls „Marketing-Kampagne“ mit farblich hervorgehobenen Aktivitäten. Die Aktivität „Klickraten auswerten“ wurde als zusätzliches kritisches Element markiert.	81
9.7 Ergebnis des Testfalls „Karten-App - Standort Erfassen“ mit farblich hervorgehobenen Aktivitäten. Die Aktivität „Route berechnen“ wurde fälschlicherweise nicht als kritisch markiert.	81

Listings

3.1	BPMN-XML-Auszug des laufenden Beispiels	17
4.1	Interne BPMN-Repräsentation je Flow-Element.	25
4.2	JSON-Schema der <code>llmProps</code>	33
4.3	JSON-Schema der API-Antwort.	34
6.1	Beispiel einer Evaluierungskonfiguration in YAML.	48
8.1	Konfigurationsdatei des Experiments mit Mistral Modellen	67
1	System-Prompt der DSGVO-Klassifikation von BPMN-Aktivitäten . .	101
2	Antworttyp der Klassifizierung	104
3	Kern der <code>id</code> -Validierung und -Vervollständigung	105
4	Schema der YAML-Evaluationskonfiguration	105
5	Zusammengefasster Logauszug zum Retry-Mechanismus	107
6	Konfigurationsdatei vom Experiment mit Gemma Modellen	109
7	Konfigurationsdatei vom Experiment mit DeepSeek Modellen	109
8	Konfigurationsdatei vom Experiment mit Qwen Modellen	110
9	Konfigurationsdatei vom Experiment mit GPT Modellen	111

Tabellenverzeichnis

5.1	Eckdaten der verwendeten Datensätze	41
5.2	Beispielhafte Aktivitäten und Label.	43
7.1	Übersicht der Kriterien zur Modellauswahl.	58
7.2	Übersicht aller Modelle mit technischen Eckdaten (Stand 30.09.2025).	61
9.1	Aggregierte Mittelwerte und Standardabweichungen der Evaluationsmetriken über alle fünf Wiederholungen hinweg.	72
9.2	Kleine vs. große Modelle: Durchschnittswerte pro Gruppe und jeweils bestes Modell.	75
9.3	Europäische vs. internationale Modelle: Durchschnittswerte pro Gruppe und jeweils bestes Modell.	76

Name: Merten Dieckmann

Matrikelnummer: 1058340

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den

Merten Dieckmann