Hochschule Aalen

Bachelorarbeit
Studiengang : Data Science

Implementierung und Evaluation einer Datenkorruptionsmethode für die Imputation mehrerer Merkmale

von

Levin Haag

80274

Betreuender Professor: Prof. Dr. Martin Heckmann

Einreichungsdatum : 02. Februar 2024

Eidesstattliche Erklärung

Hiermit erkläre ich, **Levin Haag**, dass ich die vorliegenden Angaben in dieser Arbeit wahrheitsgetreu und selbständig verfasst habe.

Weiterhin versichere ich, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt zu haben, dass alle Ausführungen, die anderen Schriften wörtlich oder sinngemäß entnommen wurden, kenntlich gemacht sind und dass die Arbeit in gleicher oder ähnlicher Fassung noch nicht Bestandteil einer Studien- oder Prüfungsleistung war.

Ort, Datum Unterschrift (Student)

Kurzfassung

Die vorliegende Arbeit widmet sich der eingehenden Analyse und dem Vergleich von Imputationsmethoden im Kontext mehrspaltiger Datenkorruption. Dabei liegt der Fokus auf unterschiedliche Aufgabenstellungen zu binärer Klassifikation, mehrklassiger Klassifikation und Regression. Die experimentelle Durchführung der Imputationen ermöglicht es, differenzierte Einblicke in die Leistungsfähigkeit verschiedener Imputationsansätze zu gewinnen.

Die Ergebnisse zeigen, dass einfache Methoden wie Mittelwert/Modus und K-NN besonders gut für Klassifikationsaufgaben geeignet sind, aufgrund ihrer stabilen Leistung und geringen Rechenkosten. Im Gegensatz dazu zeigen Deep Learning-Modelle wie Variational Autoencoder (VAE), Discriminative Deep Learning (DDL) und Generative Adversarial Imputation Network (GAIN) geringere Leistungen und höhere Rechenkosten, wodurch sie weniger empfehlenswert erscheinen. Bei Regressionsaufgaben erzielt DDL die besten Ergebnisse.

Die Betrachtung von mehrspaltiger Datenkorruption im Vergleich zur einspaltigen Variante verdeutlicht, dass die Wahl der Korruptionsmethode Auswirkungen auf die Imputationsleistung hat. Bei binärer Klassifikation führen Mittelwert/Modus, K-NN und Random Forest zu verbesserten Ergebnissen, während DDL und GAIN schlechtere Resultate zeigen. In der mehrklassigen Klassifikation sind GAIN und K-NN besonders hervorzuheben, während Random Forest die größte Verschlechterung aufweist. In Regressionsdatensätzen steigert sich die Leistung des DDL im Vergleich zur einspaltigen Datenkorruption, während Random Forest deutlich nachlässt.

Insgesamt bestätigen die Ergebnisse die Relevanz der sorgfältigen Auswahl von Imputationsmethoden je nach Aufgabenstellung und Kontext. Die Konstanz und Leistung des K-NN-Imputers unterstreichen seine Eignung für verschiedene Szenarien. Ein Ausblick auf zukünftige Forschungen schließt die Arbeit ab, indem die Erweiterung der Imputationsmethoden und die Berücksichtigung alternativer Ansätze vorgeschlagen werden.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

1 Einleitung

1.1 Problemstellung

Durch die zunehmende Bedeutung und Komplexität von Datenpipelines wird das Thema Datenqualität eine zentrale Herausforderung aller Datenwissenschaftler. Hohe Datenstandards sind essentiell um beispielsweise eine robuste Vorhersageleistung und Nutzung der automatisierten Entscheidungsfindung von Machine Learning Modellen zu gewährleisten. Ein häufiges Problem, vor allem bei großen Datenmengen, ist das Fehlen von Daten. Unvollständige Datensätze können Datenpipelines unterbrechen und verheerende Auswirkungen auf Machine Learning Anwendungen haben, wenn sie nicht erkannt werden. Während Statistiker und auch Datenforscher eine Vielzahl von Ansätzen zur Imputation fehlender Werte eingeführt haben, sind umfassende Vergleichsmaße, die klassische und moderne Imputationsansätze unter fairen und realistischen Bedingungen vergleichen, unterrepräsentiert.(Jäger et al., 2021))

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist die Erweiterung bestehender Methoden, die die Korruption von Daten mehrerer Merkmale automatisiert durchführen. Diese neuen Methoden sollen verwendet werden um Evaluationsergebnisse von Daten mit mehreren korrumpierten Merkmalen zu untersuchen und auszuwerten. Darüber hinaus werden die Ergebnisse mit den Resultaten von Dittrich, P. (2023) verglichen. Die erzielten Ergebnisse sollen dazu beitragen, die optimale Strategie für die Bearbeitung umfassend korrumpierter Datensätze zu identifizieren. Auf diese Weise wird ein wertvoller Beitrag zum Benchmark der Imputationsmethoden geleistet.

1.3 Vorgehen

Um das angestrebte Ziel zu erreichen, erfolgt zunächst eine eingehende Erläuterung der thematischen Grundlagen. Anschließend wird sowohl auf die angewandte Methodik als auch auf den aktuellen Stand der Forschung eingegangen, Schelter et al. (2021),unter besonderer Berücksichtigung der verwandten Arbeiten von Schelter et al. (2021), Jäger et al. (2021) und Dittrich, P. (2023), auf die diese Arbeit aufbaut. Nachfolgend werden die notwendigen Implementierungen und Experimente detailliert beschrieben. Der zentrale Aspekt dieser Arbeit bildet das Kapitel Ergebnisse, in dem die Resultate der durchgeführten Experimente analysiert werden. Auf Grundlage dieser Ergebnisse wird ein Fazit gezogen und eine Diskussion angestoßen. Zum Abschluss der Arbeit erfolgt eine zusammenfassende Darstellung sowie ein Ausblick auf zukünftige Entwicklungen.

2 Grundlagen

2.1 Datenqualität

Die Datenqualität misst, wie gut die Qualitätskriterien eines Datensatzes erfüllt werden. Wenn die Datenqualität dem Standard für den beabsichtigten Verwendungszweck entspricht, können die Datenkonsumenten den Daten vertrauen und sie beispielsweise zur Verbesserung der Entscheidungsfindung nutzen. Qualitativ hochwertige Daten sind essentiell für die Anwendung von künstlicher Intelligenz oder Automatisierungstechnologien. Ein altes Sprichwort besagt: „Garbage in, garbage out“, und das gilt auch für Algorithmen des maschinellen Lernens. Wenn der Algorithmus lernt, auf der Grundlage schlechter Daten Vorhersagen zu treffen oder zu klassifizieren, können wir davon ausgehen, dass er ungenaue Ergebnisse liefert. Die Datenqualität wird anhand folgender Kriterien gemessen(IBM, 2024c):

- Vollständigkeit: die Menge an Daten, die brauchbar oder vollständig ist
- Einzigartigkeit: hier wird die Anzahl der doppelten Daten in einem Datensatz berücksichtigt
- Validität: misst, inwieweit die Daten dem erforderlichen Format entsprechen.
- Pünktlichkeit: bezieht sich auf die Bereitschaft der Daten innerhalb eines erwarteten Zeitrahmens
- Genauigkeit: bezieht sich auf die Korrektheit der Datenwerte
- Konsistenz: Vergleichbarkeit mit anderen Datensätzen
- Zweckmäßigkeit: Datenbestand muss seinen Zweck erfüllen

Diese Kennzahlen helfen bei der Bewertung der Datenqualität, um zu beurteilen, wie informativ und nützlich die Daten für einen bestimmten Zweck sind.(IBM, 2024c)

2.2 Fehlende Daten

Selbst in gut konzipierten und kontrollierten Studien findet man bei fast allen Forschungsarbeiten fehlende Daten. Fehlende Daten können die statistische Aussagekraft einer Studie verringern und zu verzerrten Schätzungen und damit zu ungültigen Schlussfolgerungen führen.(Kang H., 2013)

2.2.1 Datenkorruption

Als Datenkorruption wird jede Änderung bezeichnet, die an einer Datei beispielsweise während der Speicherung, Übertragung oder Verarbeitung vorgenommen wird. Eine veränderte oder beschädigte Datei kann unbrauchbar, ungenau, unlesbar oder in irgendeiner Weise unzugänglich für einen Benutzer oder eine zugehörige Anwendung werden.(Velimirovic A., 2022) In dieser Arbeit sollen fehlende Daten bewusst provoziert werden. Dies geschieht mit angewandter Datenkorruption, indem Daten aus Datensätzen schematisch gelöscht werden.

2.3 Muster von fehlenden Daten

Die Gründe und Ursachen des Fehlens von Daten werden meistens in drei verschiedene Muster unterteilt:

- Missing Completelty At Random (MCAR)
- Missing At Random (MAR)
- Missing Not At Random (MCAR)

2.3.1 Missing Completely at Random

MCAR fässt alle fehlenden Daten zusammen, dessen Wahrscheinlichkeit zu fehlen weder mit dem spezifischen Wert, der erhalten werden soll, noch mit der Menge der Beobachtungen zusammenhängt. (Kang H., 2013) Diese Art des Fehlens wird von Arbeiten oder Studien häufig nachgestellt, da sie einfach zu implementieren ist. Reale Beispiele sind zum Beispiel das Fehlen aufgrund von Pech oder eine unbeobachtete Stichprobe von Teilen der Bevölkerung. Hier hat jede Person die gleiche Chance, für diese Stichprobe ausgewählt zu werden. (Dittrich, P., 2023)
Jäger et al. (2021) hat MCAR in einem einfachen Beispiel dargestellt. Fehlende Daten hängen nicht mit dem Wert der Größe zusammen, sondern sind völlig zufällig:

Height	$\text{Height}_{\text{MCAR}}$
179.0	?
192.0	?
189.0	189.0
156.0	156.0
175.0	?
170.0	170.0
181.0	?
197.0	?
156.0	156.0
160.0	160.0

Abbildung 2.1: Beispiel von MCAR Daten

2.3.2 Missing at Random

Hängt die Wahrscheinlichkeit vom Fehlen der Daten von den anderen Beobachtung der Daten, aber nicht von den fehlenden Daten selbst, ab, so spricht man von MAR. (Kang H., 2013) Ein anschauliches Beispiel liefert wieder Jäger et al. (2021). Hier hängt das Fehlen der Daten mit dem Geschlecht zusammen, aber nicht mit der Größe selbst:

Height	Gender	$\text{Height}_{\text{MAR}}$
200.0	M	?
191.0	M	?
198.0	F	198.0
155.0	M	?
206.0	M	?
152.0	F	152.0
175.0	F	175.0
159.0	M	?
153.0	F	153.0
209.0	M	209.0

Abbildung 2.2: Beispiel von MAR Daten

2.3.3 Missing Not at Random

Fällt das Fehlen der Daten nicht in die Kategorien MCAR und MAR, gehören sie zum übrigen Muster MNAR. Hier hängt das Fehlen der Daten mit dem Wert der fehlenden Daten selbst ab. (Dittrich, P., 2023) Ein gutes Beispiel liefert wieder Jäger et al. (2021), bei dem man sehen kann, dass alle Größen fehlen, die verhältnismäßig klein sind:

Height	$\text{Height}_{\text{MNAR}}$
154.0	?
181.0	181.0
207.0	207.0
194.0	194.0
153.0	?
156.0	?
198.0	198.0
185.0	185.0
155.0	?
164.0	?

Abbildung 2.3: Beispiel von MNAR Daten

2.4 Umgang mit fehlenden Daten

2.4.1 Datenimputation

Es gibt zwei Möglichkeiten mit fehlenden Daten umzugehen. Die erste Möglichkeit ist die Datenimputation. Bei der Datenimputation werden die fehlenden Daten anhand von Schätzungen ersetzt. Dabei werden Informationen der vorhandenen Daten genutzt um die fehlenden Daten aufzufüllen. (Emmanuel et al., 2021) Es gibt verschiedene Methoden um die fehlenden Daten zu schätzen oder ersetzen. In dieser Arbeit wurden sechs verschiedene Ansätze untersucht.

Mittelwert/Modus

Da Datensätze sowohl numerische, als auch kategoriale Werte enthalten, werden Mittelwert und Modus als gemischter Ansatz eingesetzt. Der Mittelwert ergibt sich aus der Summe aller Werte, geteilt durch die Anzahl der Werte und wird für numerische Features verwendet. Der Wert, der am häufigsten vorkommt, ist der Modus, dieser wird bei kategorialen Features eingesetzt. Der Vorteil dieses Ansatzes ist, dass kein trainiertes Modell benötigt wird, wodurch sich die Rechenzeiten gering halten, allerdings auch zu einer Veränderung der Verteilungsform und stark verzerrten Parameterschätzungen führen kann. (Jadhav et al., 2019)

K-Nearest Neighbour (K-NN)

Der K-Nearest Neighbour-Algorithmus(K-NN) ist ein überwachter Lernklassifikator, der sowohl für Klassifikations- als auch Regressionsprobleme geeignet ist. Der K-NN ermittelt die nächst gelegenen Datenpunkte eines Abfragepunktes. Zur Berechnung nutzt er verschiedene Abstandsmetriken wie den Euklidischen Abstand, den Manhatten-Abstand, den Minkowski-Abstand oder den Hamming-Abstand. Mit dem k-Wert wird bestimmt wie viele Nachbarn berücksichtigt werden um die Klassifizierung eines bestimmten Abfragepunkts zu berechnen. Ist beispielsweise k=1 wird dem Abfragepunkt nur die Klasse seines nächsten Nachbarn zugewiesen. Der Vorteil des K-NN ist die Simplizität der Anwendung, zum Beispiel durch die Nutzung weniger Hyperparameter. Allerdings neigt der Algorithmus zur Überanpassung(IBM, 2024a) In dieser Arbeit wurde scikit-learn's KNeighborsClassifier für kategoriale Features und der KNeighborsRegressor für numerische Features eingesetzt.(Jäger et al., 2021)

Random Forest

Leo Breiman und Adele Cutler entwickelten gemeinsam den Random Forest Algorithmus, welcher die Ausgabe mehrerer Entscheidungsbäume kombiniert, um ein einzelnes Ergebnis zu erhalten. Der Algorithmus kann Regressions- und Klassifikationsaufgaben bewältigen. Bei einem Regressionsproblem wird der Mittelwert der Entscheidungsbäume gewählt, bei einem Klassifikationsproblem wird die häufigste Klasse der Entscheidungsbäume bestimmt. Der Random Forest ist aufgrund seiner simplen Benutzung und Flexibilität sehr beliebt. Zudem hat der Algorithmus ein geringes Risiko für Überanpassung. Je nach Datenlage ist allerdings mit einem erhöhten Zeitaufkommen zu rechnen.(IBM, 2024b) Der in der Arbeit verwendete Random Forest ist der RandomForestClassifier von scikit-learn.(Jäger et al., 2021)

Discriminative Deep Learning (DDL)

Das Thema Deep Learning hat in den vergangenen Jahren stark an Popularität zugenommen, weshalb auch die Anwendung von Deep Learning Modellen bei Imputationsaufgaben beliebter wurde.(Jäger et al., 2021) Für diese Arbeit sind zwei Deep Learning Arten relevant, nämlich das diskriminative und das generative Deep Learning. In diesem Abschnitt wird das diskriminative Deep Learning beschrieben. Der diskriminative Ansatz wird häufig für überwachtes maschinelles Lernen verwendet.Das Hauptziel besteht darin, Grenzen zu lernen und die Datenpunkte mithilfe von Wahrscheinlichkeitsschätzungen und Maximum Likelihood in verschiedene Klassen zu unterteilen.(Dittrich, P., 2023) Für diese Arbeit wurden für das diskriminative Deep Learning zwei verschiedene Imputer von AutoKeras verwendet. Bei kategorialen Features wurde der StructuredDataClassifier und für numerische Features der StructuredDataRegressor eingesetzt. Beide Modelle sorgen für eine korrekte Codierung, sowie die Optimierung der Architektur und Hyperparameter. (Jäger et al., 2021)

Generative Deep Learning

Im Gegensatz zum diskriminativen Deep Learning wird das generative Deep Learning als unüberwachtes maschinelles Lernen klassifiziert. Das Ziel dieser Modelle ist es in die Tiefe zu gehen um die tatsächliche Datenverteilung zu modellieren und die verschiedenen Datenpunkte zu lernen, anstatt nur die Entscheidungsgrenze zwischen Klassen zu modellieren. Generative Modelle eignen sich besonders für die Vorhersage von Wahrscheinlichkeiten, da sie sich auf die Wahrscheinlichkeitsverteilungen der gegebenen Daten konzentrieren. Der Nachteil der Modelle ist die Anfälligkeit für Outlier.(Dittrich, P., 2023) In dieser Arbeit wurden mit dem Variational Autoencoder(VAE) und den Generative Adversarial Networks(GAN) zwei generative Deep Learning Modelle verwendet.

Variational Autoencoder (VAE)

Der VAE besteht aus einem Encoder und einem Decoder, die einen Engpass verursachen, welchen die Daten passieren müssen. (Dittrich, P., 2023) VAEs lernen, ihre Eingaben in eine Verteilung über den latenten Raum zu kodieren und durch Stichproben aus dieser Verteilung zu dekodieren.(Jäger et al., 2021) Durch Training mit Gradientenabstiegsiterationen wird erreicht, dass während des Kodierungs- und Dekodierungsprozesses nur eine minimale Menge an Informationen verloren wird. (Dittrich, P., 2023)

Generative Adversarial Networks (GAIN)

GAINs gehören wie die VAE zur Gruppe der generativen Modelle, was bedeutet, dass sie in der Lage sind, aus einer vorgegebenen, meist komplexen Wahrscheinlichkeitsverteilung neue Daten zu generieren . Sie bestehen aus einem Generator und einem Diskriminator. Der Generator generiert eine Stichprobe, die der Datenverteilung möglichst nahe kommt, während der Diskriminator unterscheidet, ob ein Beispiel wahr oder generiert ist.(Jäger et al., 2021) Ein Anwendungsfall, bei dem GANs häufig verwendet werden, ist die KI-Bildererzeugung.(Dittrich, P., 2023)

2.4.2 Löschen von Daten

Die zweite Möglichkeiten mit fehlenden Daten umzugehen, ist das Löschen betroffener Datenpunkte. Vor allem bei kleinen Datensätzen sollte das Löschen der Daten vermieden werden, da dadurch der Datenumfang noch geringer wird und Verzerrungen im Datensatz entstehen können.

3 Methodik

Die Grundlage dieses Projekts basiert auf den Arbeiten von Jäger et al. (2021), Schelter et al. (2021) und Dittrich, P. (2023). Daher orientiert sich das Vorgehen an den Methoden der beiden Quellen. Genau wie bei Dittrich, P. (2023) wurde die Design Science Methodik verwendet. Diese Methodik setzt sich aus sechs Aktivitäten zusammen. Zuerst wird die Problemstellung und Motivation identifiziert. Danach definiert man sich Ziele, welche rational aus der Problemspezifikation abgeleitet werden. Als nächstes erstellt man ein Konzept, wie man das Problem lösen möchte. Dafür wird die gewünschte Funktionalität und Architektur bestimmt. Im vierten Schritt folgt die Anwendung, bei der man beispielsweise Experimente, Simulationen oder Fallstudien durchführt. Danach folgt die Evaluation, bei der man die ursprünglichen Ziele mit den tatsächlich beobachteten Ergebnissen vergleicht und bewertet. Die letzte Aktivität ist die Kommunikation, bei der man allen Beteiligten und Interessierten die Ergebnisse mitteilt.(Jan vom Brocke, 2020)

Neben den Arbeiten von Jäger et al. (2021), Dittrich, P. (2023) und Schelter et al. (2021) wurden für die Recherche insbesondere englisch- und deutschsprachige Literatur in Betracht gezogen. Es wurde keine zeitliche Begrenzung für den Erscheinungszeitraum festgelegt, jedoch wurde neuere Literatur, insbesondere solche seit 2018, bevorzugt.

4 Aktueller Stand der Forschung

Ein wichtiger Grund, dass die Arbeiten von Jäger et al. (2021) und Dittrich, P. (2023) entstanden, war die mangelnde Forschungslage in Bezug einer Benchmark zur Leistung von Imputationsmethoden. Jäger et al. (2021) nahm zu bisherigen Arbeiten in seiner Arbeit Stellung. Laut Jäger et al. (2021) gab es zuvor zwei Arten von Arbeiten. Zum einen Arbeiten, die neue oder verbesserte Imputationsmethoden vorstellen und sie mit bestehenden und grundlegenden Ansätzen in breiteren Umgebungen vergleichen. Zum anderen Benchmark-Studien und Vergleiche von Imputationsstrategien. Beiden Arten haben jedoch gemeinsam, dass sie sich häufig auf bestimmte Aspekte oder Anwendungsfälle konzentrieren und nicht auf einen umfassenden Vergleich abzielen. Folgende Grafik fasst die bisherigen Studien zusammen (Jäger et al., 2021):

Study	# Datasets/tasks	# B	Missingness		Evaluation		Training on	
			Pattern	Fraction	Imp	Down	Comp	Incomp
Poulos and Valle (2018)	2 binary classification	6	MCAR MAR	0%, 10%, 20%, 30%, 40%	No	Yes	*Unclear*	
Jadhav et al. (2019)	5 datasets	7	*Unclear*	10%, 20%, 30%, 40%, 50%	Yes	No	*Unclear*	
Woznica and Biecek (2020)	13 binary classification	7	*Unclear*[a]	1%– ~ 33%	No	Yes	No	Yes
Zhang et al. (2018)	10 classification 3 regression	11	MNAR	25%, 50%, 75%	Yes	Yes[b]	*Unclear*	
Bertsimas et al. (2017)	84 datasets (classification and regression)	5	MCAR MNAR	10%, 20%, 30%, 40%, 50%	Yes	Yes[b]	*Unclear*	
Ours	21 regression 31 binary classification 17 multiclass classification	6	MCAR MAR MNAR	1%, 10%, 30%, 50%	Yes	Yes	Yes	Yes

[a]*Authors use incomplete datasets and, therefore, do not know the missingness pattern*
[b]*For a subset of the experiments, i. e, not systematical.*

Abbildung 4.1: Vergleich der bisherigen Studien

Jason Poulos (2018) verglichen die Leistung von Downstream Aufgaben anhand zweier binärer Klassifizierungsdatensätze mit imputierten und unvollständigen Daten. Daher variierten sie die Menge der MCAR- und MNAR-Werte von 0 % bis 40 % in kategorialen Merkmalen. Für die Imputation verwendeten sie sechs Modelle: Modus, Zufall, K-NN, logistische Regression, Random Forest und Support Vector Machine (SVM). Jason Poulos (2018) optimierten die Hyperparameter für eine der drei Downstream Aufgaben, nicht jedoch für die Imputationsmodelle. Sie kommen zu dem Schluss, dass die Verwendung eines K-NN-Imputationsmodells in den meisten Situationen die beste Leistung erbringt. (Jäger et al., 2021)

Jadhav et al. (2019) verglichen sieben Imputationsmethoden (Zufall, Median, K-NN, Predictive Mean Matching, Bayesianische lineare Regression, lineare Regression und Nicht-Bayesian) ohne Optimierung ihrer Hyperparameter auf der Grundlage von fünf kleinen und numerischen Datensätzen. Sie haben die Imputationsleistung der Methoden für 10% bis 50 % fehlender Werte gemessen. Auch hier zeigen die Autoren, dass die K-NN-Imputation unabhängig vom Datensatz und der fehlenden Fraktion am besten ist. (Jäger et al., 2021)

Woznica, K. (2020) bewerteten und verglichen sieben Imputationsmethoden (Zufall, Mittelwert, SoftImpute, Miss-Forest, VIM kknn, VIM Hotdeck und MICE) in Kombination mit fünf Klassifizierungsmodellen hinsichtlich ihrer Vorhersageleistung. Der Anteil der fehlenden Werte lag zwischen 1 % und etwa 33 %. Im Gegensatz zur Arbeit von Jason Poulos (2018) und Jadhav et al. (2019) konnten die Autoren damit umgehen, dass für das Training nur unvollständige Daten zur Verfügung standen. In ihrem Szenario konnten sie keine beste Imputationsmethode finden. (Jäger et al., 2021)

Zhang, H. (2018) implementierten einen iterativen Erwartungsmaximierungsalgorithmus, der eine latente Darstellung der Datenverteilung, parametrisiert durch ein tiefes neuronales Netzwerk, lernt und optimiert, um die Imputation durchzuführen. Zum Vergleich verwendeten sie zehn Klassifizierungs- und drei Regressionsaufgabendatensätze sowie 11 Baseline-Imputationen. Zhang, H. (2018) führten sowohl Bewertungen, als auch Imputation und Downstream Aufgaben durch, wobei 25 %, 50 % und 75 % der MNAR-Werte fehlten, und zeigten, dass ihre Methode die Basiswerte übertrifft. (Jäger et al., 2021)

Bertsimas, et al. (2018) lieferten den größten und umfangreichsten Vergleich, konzentrierten sich jedoch auf die Einführung eines Imputationsalgorithmus und stellten dessen Verbesserungen vor. Der vorgeschlagene Algorithmus validiert die Wahl der besten Imputationsmethode aus K-NN, SVM oder baumbasierten Imputationsmethoden, wobei auch die Hyperparameter kreuzvalidiert werden. Der Ansatz wurde anhand 84 Klassifizierungs- und Regressionsaufgaben mit fünf Imputationsmethoden, Mittelwert, prädiktives Mittelwert-Matching, Bayesian PCA, K-NN und iteratives K-NN, verglichen. Bertsimas, et al. (2018) zeigten, dass die vorgeschlagene Methode die Baselines übertrifft, dicht gefolgt von K-NN und iterativem K-NN. (Jäger et al., 2021)

5 Verwandte Arbeiten

Wie bereits erwähnt baut diese Arbeit auf den Werken von Schelter et al. (2021), Jäger et al. (2021) und Dittrich, P. (2023) auf. In diesem Kapitel wird näher auf diese drei Quellen eingegangen.

5.1 JENGA - A Framework to Study the Impact of Data Errors on the Predictions of Machine Learning Models

5.1.1 Motivation

Schelter et al. (2021) erkannten, dass Datenfehler in Machine Learning Umgebungen ein großes Problem sind, welches die Vorhersagequalität der Modelle stark mindert. Häufig ist es sehr schwierig den Fehler und seine Ursache zu entdecken. Die aktuelle Forschung konzentriert sich hauptsächlich auf die Erkennung und Behandlung solcher Datenfehler, zum Beispiel durch das Vorschlagen von Unit-Tests und Integritätsbeschränkungen für Machine-Learning-Daten , die Machine-Learning-basierte Imputation fehlender Werte und die Validierung der Vorhersagen von Black-Box-Modellen. Laut Schelter et al. (2021) ist es schwierig, allgemeine empirische Bewertungen dieser Ansätze bereitzustellen und künstlich verfälschte Daten zu generieren, die die Szenarien widerspiegeln, denen wir in der realen Welt begegnen. Deshalb entwarfen sie die Jenga-Bibliothek, die es Datenwissenschaftlern ermöglicht, die Robustheit ihrer Modelle gegenüber Fehlern zu untersuchen, die häufig in Produktionsszenarien auftreten. (Schelter et al., 2021)

5.1.2 Framework Design

Jenga wurde für drei Absonderungen entwickelt:

- Aufgaben enthalten einen Rohdatensatz, ein Machine-Learning-Modell und stellen eine Vorhersage dar
- Datenkorruption erhalten rohe Eingabedaten und wenden zufällig bestimmte Datenfehler auf sie an, zum Beispiel in Form von fehlenden Daten
- Evaluatoren erhalten eine Aufgabe und Datenkorruptionen, und führen die Evaluation durch, indem sie die Testdaten der Aufgabe wiederholt verfälschen und die Vorhersageleistung des Modells anhand der beschädigten Testdaten aufzeichnen

Jenga ermöglicht verschiedene Arten der Datenkorruptionen. Es ist möglich Werte auszutauschen , wobei ein bestimmtes Verhältnis der Werte einer Spalte durch Werte aus einer anderen Spalte ersetzt werden. Des weiteren ist es möglich mit dem Toolkit numerische Werte falsch zu skalieren oder Rauschen einzufügen. In diesem Fall wird ein Teil der Daten durch Hinzufügen von Gauß-Rauschen gestört, welches am Datenpunkt zentriert ist und dessen Standardabweichung zufällig aus dem Intervall von zwei bis fünf ausgewählt wird. Eine weitere Funktionalität ist die falsche Kodierung von Zeichen. Hier werden beispielsweise Strings verändert(a wird zu á). Auch die Korruption von Bildern ist mit Jenga möglich.(Schelter et al., 2021).
Für diese Arbeit ist die nützlichste Funktionalität Jengas die Datenkorruption in Form von fehlender Daten. Jenga ermöglicht es diese Datenkorruption in verschiedenen Mustern und Anteilen durchzuführen.(Dittrich, P., 2023)
Eine weitere wichtige Funktionalität des Frameworks befasst sich mit der Evaluation und Auswirkung der Datenfehler auf die Vorhersageleistung von Machine-Learning-Modellen. Dafür stellt es zwei verschiedene Evaluatoren zur Verfügung. Zum einen den Corruption-Impact-Evaluator, welcher eine vorgegebene Aufgabe, ein trainiertes Modell und eine manuell ausgewählte Liste von Korruptionen erfordert. Er wendet die Datenkorruption auf den Testsatz der Aufgaben an und berechnet die Vorhersageleistung der Modelle unter Berücksichtigung der Verfälschung. Der zweite Evaluator ermöglicht es Benutzern, zusätzlich ein Datenvalidierungsschema ihrer Wahl in die Auswertung zu integrieren. (Dittrich, P., 2023)

5.2 A Benchmark for Data Imputation Methods

5.2.1 Motivation und Zielsetzung

Da in der jüngeren Vergangenheit immer mehr Imputationsverfahren eingeführt wurden, um dem Fehlen von Daten entgegenzuwirken, nahm sich Jäger et al. (2021) zum Ziel ein Vergleichsmaßstab von verschiedenen Imputationsverfahren zu erstellen. Die Lücke von fehlenden Studien, die Imputationsverfahren in fairen, vergleichbaren Bedingungen untersucht, soll geschlossen werden. (Jäger et al., 2021)

5.2.2 Vorgehen

Um ein Vergleichsmaßstab zu erstellen wurden eine umfassende Reihe von Experimenten mit einer großen Anzahl von Datensätzen durchgeführt. Dabei wurden sowohl simple Imputationsverfahren, als auch neuartige Deep-Learning Methoden eingesetzt. Jede Imputationsmethode wird hinsichtlich der Imputationsqualität und der Auswirkung der Imputation auf eine Machine-Learning-Aufgabe bewertet. Für die Experimente wurden 69 verschiedene Datensätze benutzt. Davon 21 mit Regressions-, 31 mit binären Klassifikations- und 17 mit Mehrklassenklassifikationsaufgaben. Alle Datensätze sind vollständig und haben keine fehlenden Werte. Die Größe der Datensätze variiert von 5 bis 25 Features und von 3.000 bis 100.000 Beobachtungen. Die Experimente wurden mit unterschiedlichen Szenarien durchgeführt. Als Fehlmuster wurden MNAR, MAR und MCAR verwendet(siehe Abschnitt 2.3). Außerdem gab es vier verschiedene Grade als Anteil der Daten, die im Datensatz fehlen: 1%, 10%, 30% und 50%. Als Imputationsmethoden wurden die sechs, in Unterabschnitt 2.4.1 vorgestellten, Imputatoren verwendet. Die Bewertung erfolgt durch den Vergleich der Imputationsleistung als auch die Auswirkung auf das Downstream Modell. Als Bewertungsmetriken wurden der F1-Macro-Score und der Root Mean Square Error (RMSE) benutzt. Der F1-Macro-Score wird verwendet, um die Leistung von Klassifikationsaufgaben zu bewerten, während der RMSE für die Regressionsaufgaben verwendet wird. Während Letzterer als Fehlermaß gilt und ein kleiner Wert auf eine bessere Leistung hinweist, wird bei F1-Macro-Ergebnissen ein höherer Wert bevorzugt. (Jäger et al., 2021)

5.2.3 Experimente

Der gesamte Prozess besteht aus zwei unterschiedlichen Experimenten:

Im ersten Experiment wurde geprüft wie genau die Imputationsmethoden die ursprünglichen Werte ersetzen können. Mithilfe des Toolkits Jenga von Schelter et al. (2021) wurden Datenpunkte im Datensatz in unterschiedlichen Mustern gelöscht. Das Muster war abhängig von den jeweiligen Einstellungen, was den Anteil der gelöschten Daten und das Muster der fehlenden Daten betrifft(siehe Abschnitt 2.3). Im Anschluss wurden auf die gelöschten Daten die jeweilige Imputationsmethode angewendet. Danach wurde der eingesetzte Wert mit dem verworfenen Groundtruth-Wert verglichen. Die Bewertung der Imputationsmethoden erfolgte bei numerischen Werten mit dem RMSE, und bei kategorialen Werten mit dem F1-Score.(Jäger et al., 2021)

Das zweite Experiment untersucht die Auswirkungen der Imputation auf Downstream Aufgaben. Ein wichtiger Hinweis hierfür ist, dass es für diskriminierende Modelle notwendig ist, für jede Spalte mit fehlenden Werten ein Imputationsmodell zu trainieren. Das führt in Kombination mit der Vielzahl an Datensätzen dieses Versuchs zu enormen Rechenzeiten. Um diese zu reduzieren, wurden die Werte nur in der zu imputierenden Spalte der Testsätze verworfen.(Jäger et al., 2021)
Zu Beginn des Experiments wurde das Baseline Modell mit den Trainingsdaten trainiert und das Ergebnis der Testdaten in Form von RMSE oder F1-Score Werten zurückgeliefert. Danach wurden Daten per Datenkorruption verworfen und auf diesen veränderten Datensatz das Baseline Modell angewendet und der Score auf dem nicht vollständigen Testdatensatz(„Unvollständig") berechnet. Als nächstes wurden die fehlenden Werte imputiert und das Baseline Modell erneut angewendet, sodass man einen „Imputed“ Score ermitteln kann. Zum Schluss wurden die Auswirkungen der Imputation festgestellt, indem die prozentuale Veränderung berechnet wurde:(Jäger et al., 2021)

$$\text{Auswirkung auf Downstream Aufgabe} = \frac{Imputed - Unvollständig}{Baseline}$$

5.2.4 Ergebnisse

Um die unterschiedlichen Imputationsmethoden zu vergleichen wurde ein Rangsystem eingesetzt. Die Methode mit dem jeweils besten Ergebnis des Experiments bekam Rang 1, während die Methode mit dem schlechtesten Ergebnis Rang 6 zugewiesen bekam. Schlug der Durchlauf einer Methode fehl, bekam er automatisch den schlechtesten Rang.(Jäger et al., 2021)
Bei der Leistung der Imputation schnitten Random Forest, K-NN und DDL in den meisten Fällen am besten ab. Insgesamt erreichte der Random Forest am häufigsten Rang 1. Der GAIN Imputer schlug in 33% der Experimente fehl, weshalb er am Schlechtesten abschnitt. Eine Ursachenuntersuchung ergab, dass der Diskriminatorverlust von GAIN irgendwann Not a Number (NaN) erreicht, was zu Fehlern bei weiteren Berechnungen und einem Trainingsabbruch führt.
Die Ergebnisse ergaben, dass einfache Imputationsmethoden wie K-NN und Random Forest häufig die beste Leistung erbringen, dicht gefolgt vom diskriminierenden Deep-Learning-Ansatz. Eine interessante Beobachtung war das sehr gute Abschneiden von Mittelwert/Modus bei MNAR Experimenten mit einem hohen Anteil von fehlenden Daten. Die generativen Ansätze erreichen mittlere Ränge (VAE) oder liegen auf den schlechtesten Rängen (GAIN).(Jäger et al., 2021)

Die Ergebnisse bei den Experimenten bezüglich der Auswirkung auf Downstream Aufgaben ergaben, dass in mehr als 75% der Experimente die Vorhersageleistung um mindestens 10% gesteigert werden konnte. Die Autoren hoben den Random Forest Algorithmus hervor, da er angesichts der Leistungsverbesserung und der Gesamtrechenzeit das insgesamt beste Ergebnis lieferte.(Jäger et al., 2021)(Dittrich, P., 2023)

5.3 Assessing and Predicting the Optimal Imputation Method Regarding the Predictive Performance of Machine Learning Models

5.3.1 Motivation und Zielsetzung

Die Abschlussarbeit von Dittrich, P. (2023) erweitert den Ansatz von Jäger et al. (2021). Während sich die Experimente bei Jäger et al. (2021) hauptsächlich auf die Testdaten bezog, wählte Dittrich, P. (2023) einen anderen Ansatz und weitete die Experimente auf die Trainingsdaten aus. Ziel der Arbeit war es, wie auch schon bei Jäger et al. (2021), Unterschiede zwischen den sechs Imputationsmethoden für die unterschiedlichen Szenarien zu identifizieren und, falls möglich, eine beste Imputationsmethode zu bestimmen.(Dittrich, P., 2023)

5.3.2 Vorgehen und Experimente

Die sechs Imputationsmethoden (Mittelwert/Modus, Random Forest, K-NN, diskriminatives Deep Learning, VAE und GAIN) wurden auf die zuvor korrumpierten Trainingsdaten angewendet, die wiederum zum Trainieren des Downstream Modells verwendet wurden. Folgendes Schema veranschaulicht den Prozess:

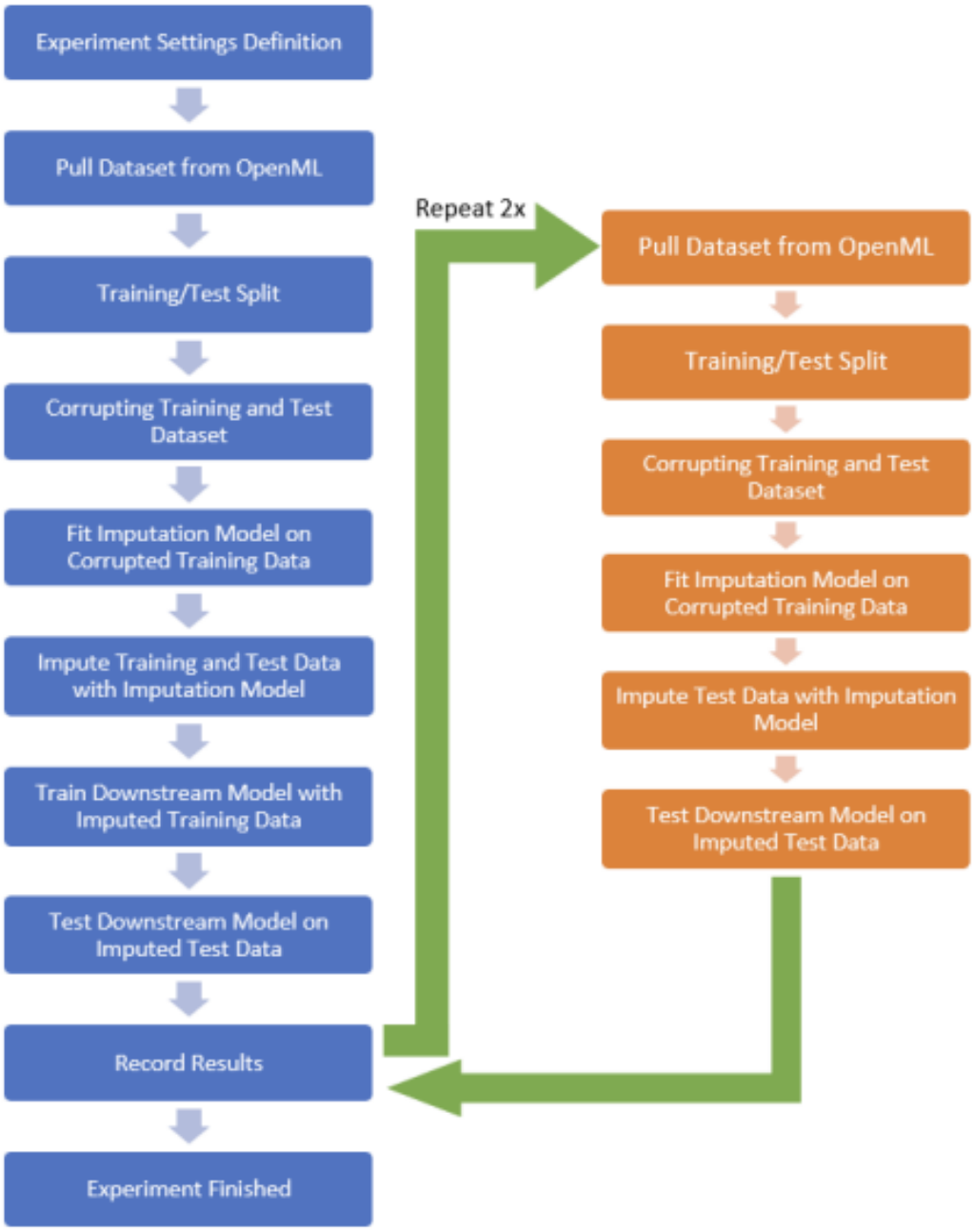

Abbildung 5.1: Prozess der Experimente(Dittrich, P., 2023)

5.3.3 Ergebnisse

Genauso wie bei Jäger et al. (2021) wurde ein Rangsystem angewendet um die Ergebnisse der Imputationsmethoden zu vergleichen. Die Imputationsmethode mit dem besten Ergebnis bekommt jeweils Rang 1 zugewiesen, während die Methode mit dem schlechtesten Ergebnis jeweils Rang 6 zugewiesen bekommt. In manchen Fällen schlugen Experimente fehl und kamen zu keinem Ergebnis. In diesem Fall bekam das Experiment automatisch den schlechtesten Rang zugewiesen. Wie schon bei Jäger et al. (2021) war GAIN auffällig oft von fehlgeschlagenen Experimente betroffen. Bei Dittrich, P. (2023) waren es 7% aller Ergebnisse. Neben GAIN schlug das Training auch für K-NN (5 Mal), Random Forest (12 Mal) und Deep Learning (19 Mal) fehl.

Die Ergebnisse der Imputationsexperimente von Dittrich, P. (2023) zeigen, dass Random Forest im Durchschnitt die besten Resultate erzielt, sowohl bei Regressionsaufgaben, als auch bei Binärer- und Mehrklassenklassifikation. Außerdem zeigt sich, dass Random Forest zwar nicht immer das beste Ergebnis hat, allerdings wird die Methode nur selten von einer anderen deutlich übertroffen. Auch was die Qualität der Imputation angeht, zeigen die vom Random Forest imputierten Werte die höchste Genauigkeit im Vergleich zu den Groundtruth-Werten.
Als Beobachtung konnte auch festgestellt werden, dass die einfacheren Imputationsmethoden wie Mittelwert/Modus, K-NN und Random Forest im Vergleich zu den diskriminierenden und generativen Deep-Learning Ansätzen sehr gute Ergebnisse bei im Vergleich sehr geringen Rechenzeiten erzielen konnte. Dennoch boten die generativen Deep-Learning-Ansätze, häufig in Kombination mit höheren Anteilen imputierter Daten, ein höheres Potenzial für signifikante Verbesserungen oder Verschlechterungen für die Vorhersageleistung des Downstream Lernmodells, welches mit von diesen Methoden imputierten Daten trainiert wurde.
Das Fazit der Arbeit war, dass die Ergebnisse der unterschiedlichen Imputationsmethoden bei kleinen Datensätzen vergleichbar sind. Generell wird empfohlen den Random Forest Imputer zu nutzen, wenn man Daten imputieren möchte, die für Trainings und Tests von Downstream Machine-Learning-Modellen genutzt werden sollen. Bei gewissen Szenarien erzielen andere Methoden zwar bessere Ergebnisse, allerdings erzielt der Random Forest Imputer in der Regel konkurrenzfähige Ergebnisse und hat gleichzeitig einen sehr effiziente Ressourcen- und Zeitverbrauch. (Dittrich, P., 2023)

6 Implementierung und Experimente

6.1 Datenkorruption mehrerer Merkmale

Während Jäger et al. (2021) und Dittrich, P. (2023) bei der Datenkorruption in ihren Experimenten nur das Targetfeature korrumpierten, wurde bei dieser Arbeit ein anderer Ansatz gewählt. Durch eine Anpassung des Codes wurden die Methoden erweitert und ergänzt, sodass alle Features der Datensätze mit fehlenden Daten manipuliert werden können. Die Tabellen 6.1 und 6.2 veranschaulichen als Beispiele den Unterschied der Datenkorruption bei einem Anteil von 50% an fehlenden Daten mit dem Target Feature „Gewicht“.

Name	Größe in Zentimeter	Gewicht in Kilogramm
Anton Adler	182	NaN
Beate Blau	165	61
Christian Chaos	175	78
Dagmar Dorsch	170	69
Emil Ehrenreich	192	NaN
Frauke Fuchs	158	NaN

Tabelle 6.1: Datenkorruption nur in Target Feature

Name	Größe in Zentimeter	Gewicht in Kilogramm
Anton Adler	NaN	81
Beate Blau	NaN	61
NaN	NaN	78
Dagmar Dorsch	170	NaN
NaN	192	NaN
Frauke Fuchs	NaN	NaN

Tabelle 6.2: Datenkorruption in allen Spalten

Ein Grund dieses Ansatzes ist der Versuch eine komplexere Datenlage für die unterschiedlichen Imputationsmethoden zu schaffen. Dadurch, dass in allen Spalten Daten fehlen, sollten es die Imputer schwerer haben möglichst genaue Werte zu

imputieren. Da bei Jäger et al. (2021) und Dittrich, P. (2023) die simpleren Imputationsmethoden wie Random Forest, Mittelwert/Modus und K-NN am besten abschnitten, sollte untersucht werden, ob die komplexere Datenlage zu einem anderen Ergebnis führt.

6.2 Imputationsexperimente

Um eine möglichst hohe Vergleichbarkeit der Ergebnisse zu schaffen wurde bei den Experimenten das selbe Vorgehen wie bei Dittrich, P. (2023) angewendet. Die sechs Imputationsmethoden Mittelwert/Modus, Random Forest, K-NN, VAE, GAIN und DDL wurden gegenüber Jäger et al. (2021) auch auf die zuvor korrumpierten Trainingsdaten angewendet, die wiederum zum Trainieren des Downstream Lernmodells verwendet worden sind. Eine Veranschaulichung davon sah man bereits in Unterkapitel 5.3.2 bei Abbildung 5.1.
Um ein Experiment auszuführen, gab es verschiedene Parameter, die man definieren musste:

- Datensatz ID
- Imputationsmethode
- Name des Experiments
- Anteil der fehlenden Daten
- Muster der fehlenden Daten
- Imputationsstrategie
- Anzahl der Wiederholungen des Experiments
- Pfad um Ergebnisse zu speichern

Die Datensatz Identität (ID) referenziert zu dem ausgewählten Datensatz anhand seiner OpenML ID. Hierfür wurden die selben Datensätze wie Jäger et al. (2021) verwendet(siehe Unterkapitel 5.2.2) Mit der Imputationsmethode wählt man einer der sechs Imputern aus(siehe Unterkapitel 2.4.1). Außerdem gibt man dem Experiment einen Namen um es gegebenenfalls von anderen Versuchen zu unterscheiden. Um bei den Anteilen der fehlenden Daten verschiedene Szenarien zu unterscheiden, wurden die Experimente mit 1%, 10%, 30% und 50% fehlenden Daten durchgeführt. Mögliche Eingaben für die Muster der fehlenden Daten sind „MNAR“, „MAR“ und „MCAR“(siehe Unterkapitel 2.3). Als Imputationsstrategie wurde „single_all“ gewählt, da diese Strategie alle Spalten zur Berechnung des zu imputierenden Werts berücksichtigt und nur Werte der Zielspalte imputiert. Um die Ergebnisse zu festi-

gen, ist die Anzahl der Wiederholungen jeweils drei. Damit man die Ergebnisse nach dem Experiment abrufen kann, muss ein Pfad gesetzt werden, der die Ergebnisse speichert. Ein einzelnes Experiment definiert sich mit jeweils einem Datensatz, einer Imputationsmethode, einem Anteil an fehlenden Daten und einem Muster der fehlenden Daten. Um ein einzelnes Experiment durchzuführen gibt es das Pythonskript „run-experiment.py“. Der Programmaufruf eines einzelnen Experiments sieht wie folgt aus:

```
python run-experiment.py 42545 knn experiment_corrupted
    --missing-fractions 0.3 - missing-types MCAR --strategies
    single_all --num-repetitions 3 --base-path ../result_path
```

Quelltext 6.1: Beispielaufruf eines einzelnen Experiments

Nach dem Programmaufruf versucht die angebundene OpenML-API mit der Datensatz ID den gesuchten Datensatz zu importieren. Anschließend werden die Daten in Trainings-(80%) und Testsätze(20%) aufgeteilt. Um die größtmögliche Vergleichbarkeit zu gewährleisten, wurde der selbe Randomseed wie bei Dittrich, P. (2023) verwendet. Als nächstes werden die Daten nach den angegebenen Parametern mit Jenga((Schelter et al., 2021)) korrumpiert. Danach sind die Daten bereit imputiert zu werden und das Imputationsmodell wird eingerichtet. Dafür wird das Modell mit den korrumpierten Trainingsdaten trainiert. Zu Beginn des Imputationsprozesses wird außerdem begonnen die Zeit zu messen, um nachzuvollziehen wie lange jeweils die Imputation dauert. Wenn das Imputationsmodell trainiert wurde, werden die fehlenden Trainings- und Testdaten imputiert. Anschließend werden die imputierten Trainingsdaten genutzt um das Downstream Modell zu trainieren. Bei diesem Vorgang wird wie bei Jäger et al. (2021) und Dittrich, P. (2023) der scikit-learn OneHot-Encoder((scikit-learn, 2024b)) für kategoriale Daten und der scikit-learn StandardScaler((scikit-learn, 2024e)) für numerische Daten genutzt. Ist das Downstream Modell trainiert, verwendet das Programm den scikit-learn Stochastic Gradient Descent (SGD) Classifier((scikit-learn, 2024c)) für Klassifikationsaufgaben und den scikit-learn SGDRegressor((scikit-learn, 2024d)) für Regressionsaufgaben. Zur Vergleichbarkeit sind die Hyperparameter dieselben wie bei Jäger et al. (2021) verwendet worden sind. Um weitere Reproduzierbarkeit zu gewährleisten, bleibt der Randomseed für die Downstream Modelle während der Experimente ebenfalls statisch. Mit dem Gridsearch-Modul ((scikit-learn, 2024a)) wählt das Programm die besten Schätzer für die endgültige Modellanpassung. Nach dem Training wird das Modell zuerst auf den Testdaten des ersten Durchlaufs des Experiments verwendet und der Score berechnet. Es ist wichtig zu betonen, dass das Downstream Modell nur während des ersten Durchlaufs des Experiments trainiert wird. Bei dem zweiten und dritten Durchlauf wird dieses Modell erneut angewandt. Nach jedem Durchlauf werden die Ergebnisse der Experimente in einer CSV-Datei, im vom Benutzer festgelegten Pfad, gespeichert. Es werden sowohl die Ergebnisse jedes Durchlaufs, als auch eine Zusammenfassung aller Durchläufe gespeichert. Außerdem wird auch die Zeit gespeichert, die für einen Durchlauf benötigt wurde. (Dittrich, P., 2023)

7 Ergebnisse

Dieses Kapitel fasst die Ergebnisse der Experimente dieser Arbeit zusammen. Im ersten Teil werden die Ergebnisse der Experimente mit der Datenkorruption mehrerer Spalten analog zum Vorgehen von Dittrich, P. (2023) analysiert. Im zweiten Teil der Analyse werden die Unterschiede zwischen Dittrich, P. (2023) und dieser Arbeit untersucht.

7.1 Ergebnisse mit Datenkorruption mehrerer Merkmale

In diesem Unterkapitel werden die Imputationsmethoden hinsichtlich ihrer Leistung untersucht, wenn die Daten zuvor in mehreren Spalten korrumpiert wurden. Um die Methoden zu vergleichen wurde wie bei Jäger et al. (2021) und Dittrich, P. (2023) ein Rangsystem angewendet. Die Methode mit dem besten Ergebnis im jeweiligen Experiment bekommt den Rang 1. Rang 6 und damit den schlechtesten Rang bekommt der Imputer mit dem schlechtesten Ergebnis. Schlägt ein Experiment fehl, wird der Imputationsmethode automatisch der schlechteste Rang zugewiesen. Haben zwei oder mehr Methoden die selben Ergebnisse, werden die Imputer in der Reihenfolge ihrer benötigten Zeit für das Experiment bewertet. Die Methode, die am schnellsten ihre Durchläufe abgeschlossen hat, bekommt den besten Rang. Die Ergebnisse sind abhängig von ihrem erzielten durchschnittlichen F1-Score oder RMSE Wert ihrer drei Durchläufe. Klassifikationsaufgaben werden mit dem F1-Score bewertet. In diesem Fall ist das Ergebnis besser, je näher der Score an den Wert 1 herankommt. Bei Regressionsaufgaben wird der RMSE Wert als Bewertungsmaß genommen. Hier ist das Ergebnis besser, je geringer dieser Wert ist.
Gegenüber Jäger et al. (2021) sind alle Experimente der Datensätze „42675“ und “42669“ gescheitert, da diese beiden Datensätze nicht mehr bei OpenML verfügbar sind. Da die Daten mit einer Application Programming Interface (API) geladen und nicht lokal gespeichert werden, konnte das Projekt nicht mehr auf diese Datensätze zurückgreifen und brach das Experiment jeweils ab.
Wie schon bei Jäger et al. (2021)(33%) und Dittrich, P. (2023)(7%) war es auffällig, dass verhältnismäßig viele Experimente des GAIN-Imputers fehlschlugen(Gründe siehe 5.2.4). In dieser Arbeit waren es 156 fehlgeschlagene Experimente und somit rund 19,7% aller GAIN-Experimente. Die Methode DDL hatte mit 42 Experimente(5,3%) die zweitmeisten Fehlschläge. Außerdem schlugen bei Random Forest 12 Experimente(2,4%) fehl. Die Methoden Mittelwert/Modus, K-NN und VAE konnten

alle Experimente erfolgreich durchlaufen.

7.1.1 Binäre Klassifikation

Bei der binären Klassifikation wurden Experimente zu 31 verschiedenen Datensätzen durchgeführt. Die Größe der Datensätze unterscheidet sich von Kleinen mit knapp 3.000 bis zu großen Datensätzen mit über 96.000 Beobachtungen. Berücksichtigt man die drei Muster an fehlenden Daten und die vier Anteile an fehlenden Daten hat man insgesamt 372 verschiedene Szenarien. Diese Szenarien wurden auf die sechs Imputationsmethoden angewandt, was eine Gesamtzahl von 2.232 durchlaufene Experimente macht. Davon ergaben sich 2.105 gültige Ergebnisse. Rund 5,7% der Ergebnisse der binären Klassifikation sind also fehlgeschlagen.
Betrachtet man die Ergebnisse des Rangsystems(siehe 7.1) fällt direkt das starke Abschneiden des Mittelwert/Modus Imputers auf. Er erzielte in 201 Experimenten das beste Ergebnis, was einer Quote von 54% entspricht. Neben Mittelwert/Modus erzielten auch die beiden simpleren Methoden K-NN und Random Forest sehr gute Ergebnisse. Die drei diskriminativen und generativen Deep-Learning Methoden fielen dagegen deutlich ab. Die Ergebnisse von GAIN leiden darunter, dass 21,5% der Versuche fehlschlugen und diese 80 Experimente automatisch den schlechtesten Rang bedeuten.

Imputationsmethode	Durchschnittlicher Rang	Anzahl Rang 1
Mittelwert/Modus	2,34	201
K-NN	2,64	48
Random Forest	3,01	29
VAE	3,70	35
DDL	4,36	23
GAIN	4,47	36

Tabelle 7.1: Überblick der Ergebnisse für Binäre Klassifikation

Die Analyse ergibt, dass der durchschnittliche Unterschied von der jeweils besten Methode zur durchschnittlichen besten Methode(Mittelwert/Modus) ungefähr 0.002 F1 Score Punkte beträgt. Dies entspricht einer prozentualen Verbesserung von 0.004%.
Schaut man sich die Abbildungen 7.1, 7.2 und 7.3 an sieht man, dass die Ergebnisse alle sehr nah beieinander liegen. Nahezu alle Werte liegen im Intervall von -0,01 bis 0,01 F1 Punkten innerhalb der Mittelwert/Modus Ergebnisse. Es gibt nur wenige Ausreiser.
Bei den Schaubildern 7.1 und 7.2 lassen sich kein Trend ableiten. Bei Grafik 7.3 erkannt man, dass die Experimente, die nicht zum Intervall -0,01 bis 0,01 gehören, hauptsächlich MCAR Experimente sind.

Untersucht man die Grafiken 7.4, 7.5 und 7.6, welche die Differenz des besten Imputerergebnis mit dem jeweiligen Ergebnis von Mittelwert/Modus vergleicht und visualisiert, erkennt man wieder, dass die Ausreiser ausschließlich MCAR Experimente sind(7.6).
In den Grafiken 7.4 und 7.5 lässt sich zu Methoden und Anteil fehlender Daten keine Erkenntnis gewinnen, da alle Intervalle relativ gleichmäßig aufgeteilt sind.

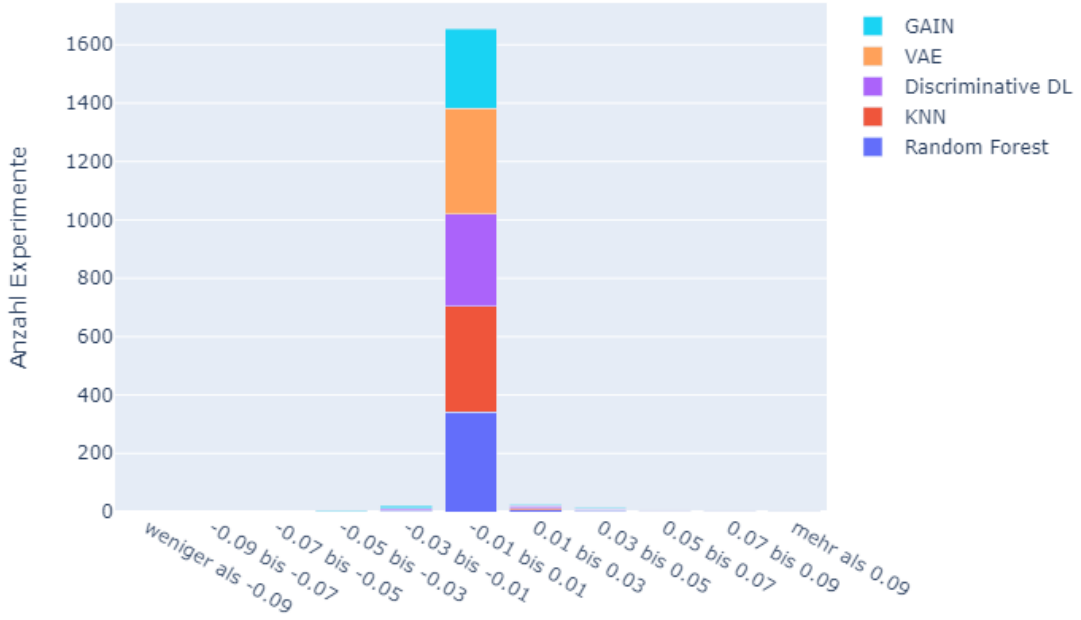

Abbildung 7.1: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Methode

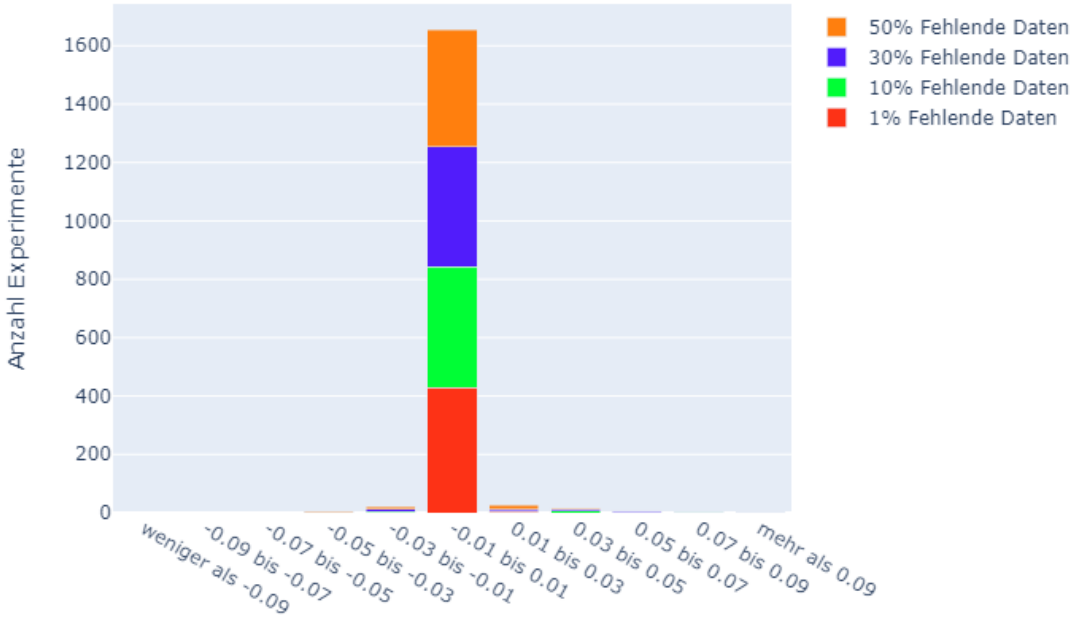

Abbildung 7.2: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Anteil fehlender Daten

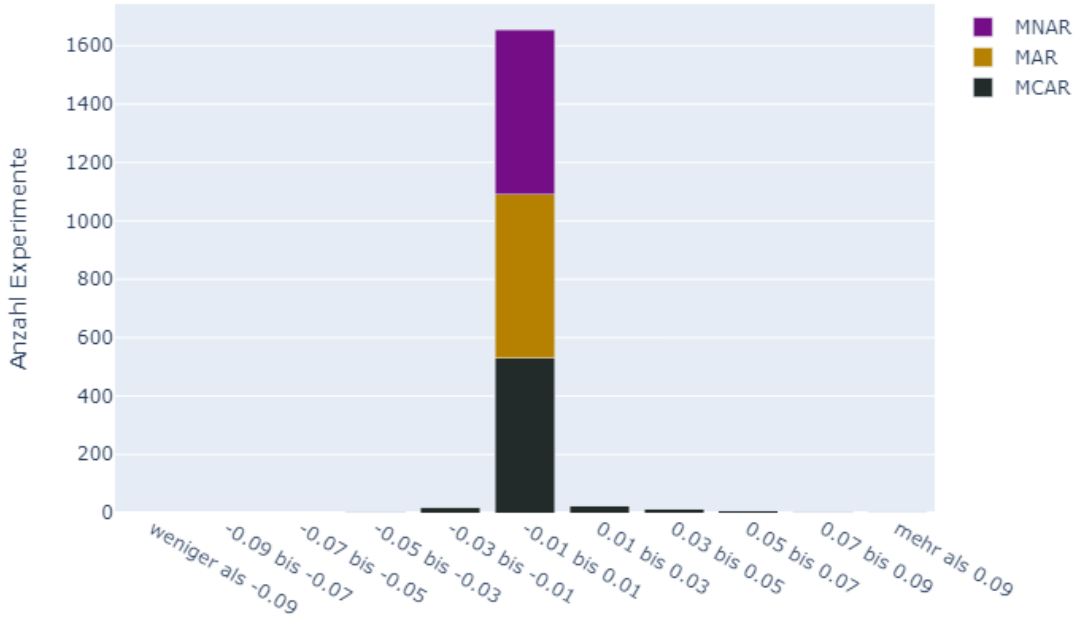

Abbildung 7.3: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Muster der fehlenden Daten

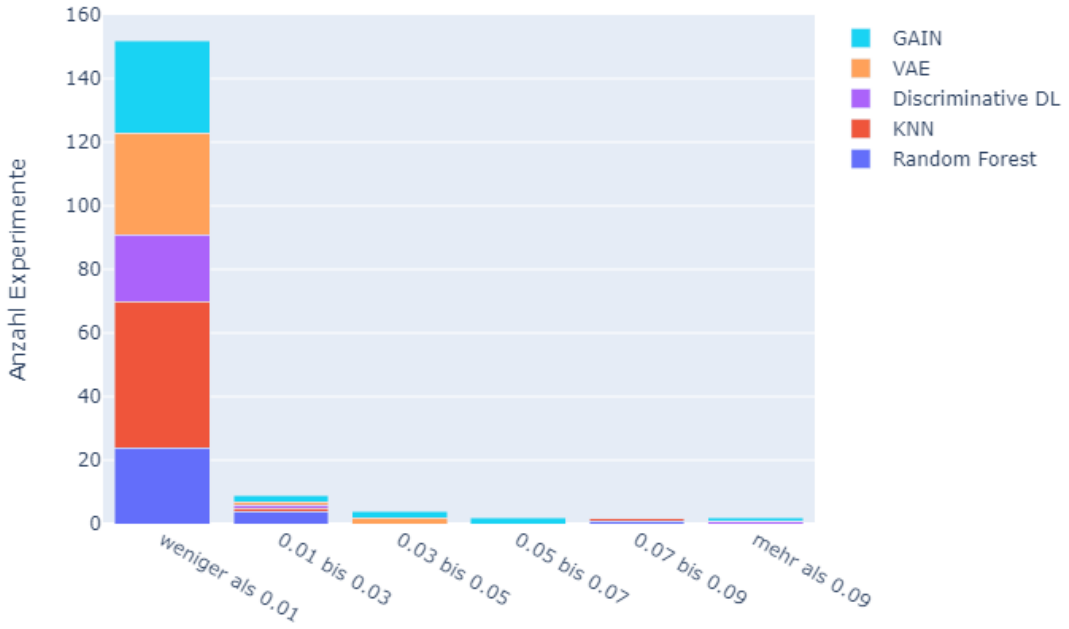

Abbildung 7.4: Differenz vom Ergebnis in Prozent des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Methode

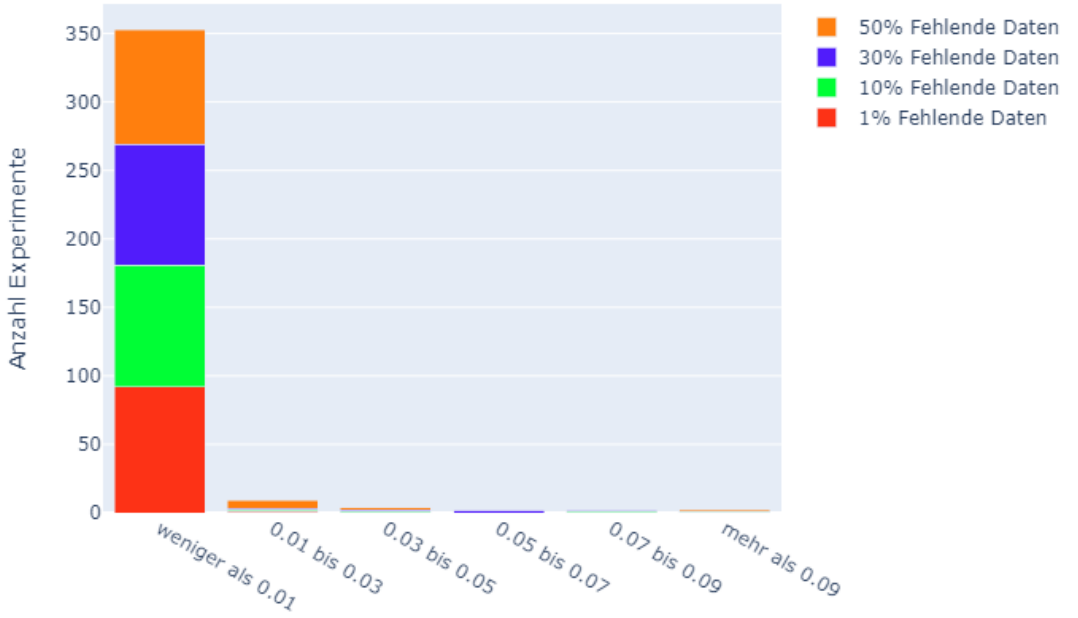

Abbildung 7.5: Differenz vom Ergebnis des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Anteil fehlender Daten

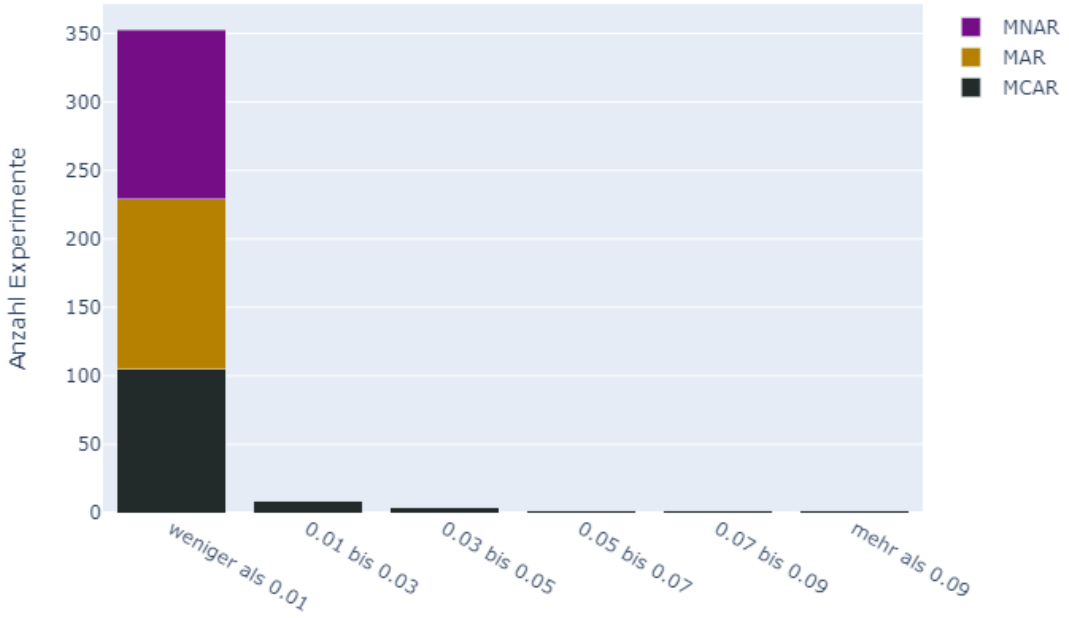

Abbildung 7.6: Differenz vom Ergebnis des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Muster der fehlenden Daten

7.1.2 Mehrklassen Klassifikation

Die Analyse der Mehrklassen Klassifikation umfasst Experimente zu 17 verschiedenen Datensätze, welche zwischen einer Größe von 3.700 bis 58.000 Observationen und 5 bis 25 Merkmalen variieren. Das bedeutet es gab eine Anzahl von 204 verschiedener Szenarien und somit 1.224 verschiedene Experimente. Am Ende konnten 1.129 Experimente gültig durchgeführt werden, was einer Quote von 92,2% gültigen Experimenten entspricht. Bei GAIN schlugen 23 Experimente fehl. Die restlichen Imputationsmethoden hatten keine Fehler. Mit einem Blick auf Tabelle 7.3 zeigt sich direkt eine Änderung zum Ergebnis der Binären Klassifikation(7.1). K-NN schneidet mit dem klar besten durchschnittlichen Rang ab, während Mittelwert/Modus gegenüber der Binären Klassifikation auf Platz 2 abrutscht. Allerdings erreicht Mittelwert/Modus weiterhin am häufigsten den besten Rang, was für inkonstante Ergebnisse spricht. Die Plätze 3-6 sind deutlich enger, was den durchschnittlichen Rang betrifft. Sie liegen alle innerhalb eines Differenz von weniger als 0,1. GAIN schlägt trotz einer Fehlschlagquote von 12% die Methoden DDL und VAE im durchschnittlichen Rang. Auch bei der Anzahl an Rang 1 Ergebnissen, erreicht GAIN das beste Ergebnis nur einmal weniger als das durchschnittlich beste Ergebnis K-NN.

Imputationsmethode	Durchschnittlicher Rang	Anzahl Rang 1
K-NN	2,96	34
Mittelwert/Modus	3,31	50
Random Forest	3,58	19
GAIN	3,59	33
DDL	3,59	26
VAE	3,67	30

Tabelle 7.2: Überblick der Ergebnisse für Multiklassen Klassifikation

Berechnungen der Ergebnisse ergaben, dass der durchschnittliche Unterschied von der jeweils besten Methode zur durchschnittlichen besten Methode(Mittelwert/Modus) ungefähr 0.01 F1 Score Punkte beträgt. Dies entspricht einer prozentualen Verbesserung von 0.02%. Betrachtet man die Grafiken 7.7, 7.8 und 7.9, welche die Differenz der Bewertung zwischen den Experimenten der durchschnittlich besten Methode K-NN und den anderen Imputationsmethoden vergleicht, sieht man eine leicht breitere Verteilung als bei der binären Klassifikation. Die deutliche Mehrheit liegt immer noch im Bereich -0,01 bis 0,01, was für häufig sehr ähnlich gute Ergebnisse spricht.
Zoomt man in die Grafik 7.7, sieht man, dass die sowohl positiven als auch negativen Ausreiser hauptsächlich den Deep Learning Modellen DDL, GAIN und VAE zuzuordnen sind.
In Schaubild 7.8 erkennt man, dass die Ausreiser hauptsächlich Experimente mit einem hohen Anteil(30% und 50%) von fehlenden Daten sind, während der Anteil

der Experimente mit niedrigem Anteil(1% und 10%) umso weniger wird, je größer die Differenz wird.
Untersucht man die verschiedenen Muster der fehlenden Daten in Grafik 7.9 sieht man wie schon bei der Binären Klassifikation, dass die Ausreißer fast ausschließlich aus MCAR Experimenten bestehen.
Auch bei den Grafiken 7.10, 7.11 und 7.12, welche die Differenz des besten Imputerergebnis mit dem jeweiligen Ergebnis von K-NN vergleicht, erkennt man gegenüber der Binären Klassifikation eine breitere Verteilung. Allerdings liegt die große Mehrheit der Experimente immer noch im Bereich weniger als 0,01, was bedeutet, dass es wenige Experimente gibt, die ein klar besseres Ergebnis als K-NN erzielen.
Ausreißer mit einer Verbesserung von über 0,05 sind hauptsächlich die Deep Learning Methoden DDL, GAIN und VAE(7.10).
In Grafik 7.11 erkennt man, dass bei steigender Verbesserung der Anteil an Experimenten mit 1% fehlender Daten nachlässt.
Aus Schaubild 7.12 lässt sich schließen, dass Experimente mit einer Verbesserung von mehr als 0,07 F1-Punkten nur aus MNAR und MCAR Experimenten besteht.

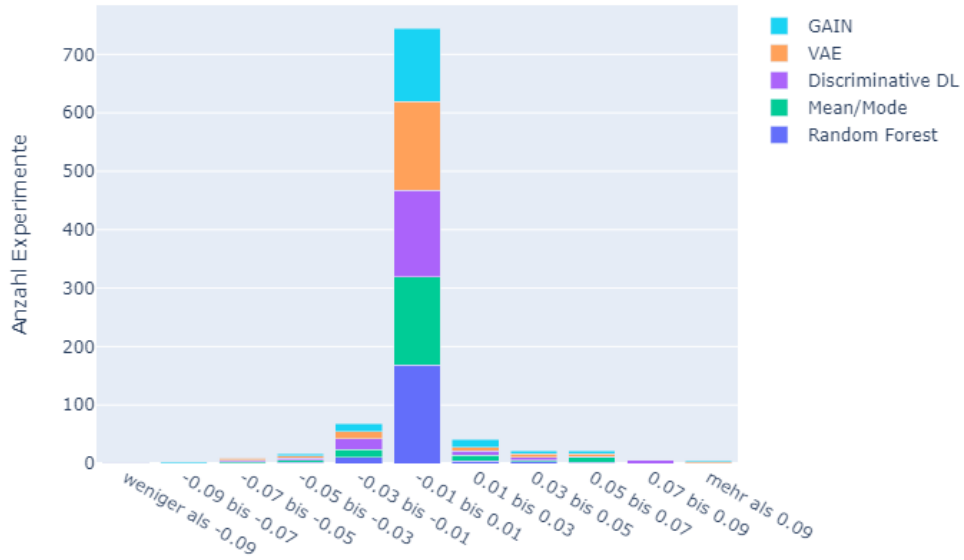

Abbildung 7.7: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Methode

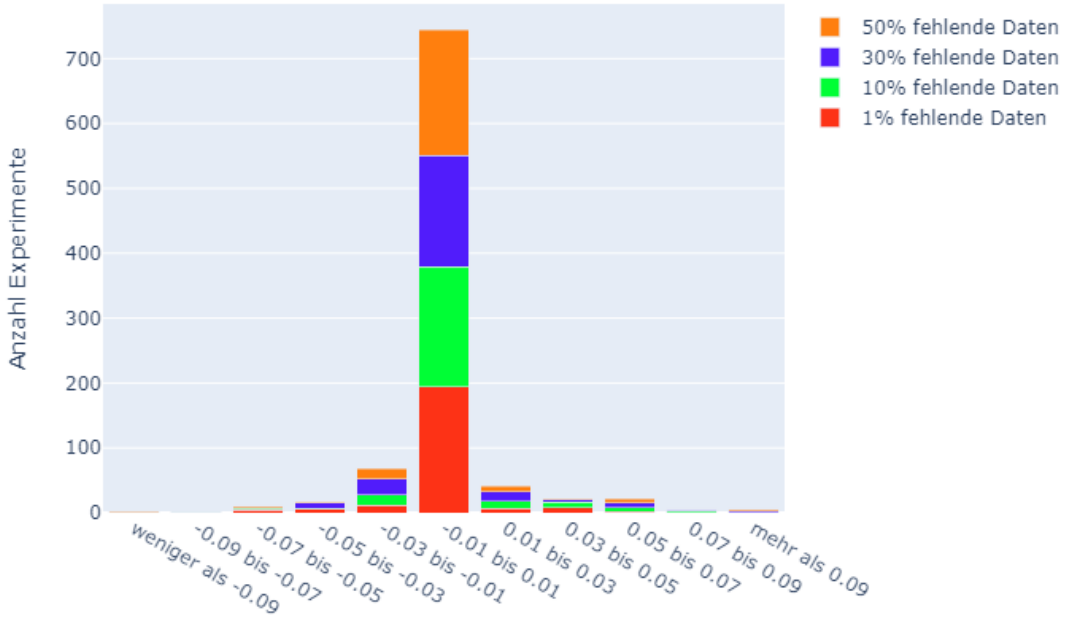

Abbildung 7.8: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Anteil fehlender Daten

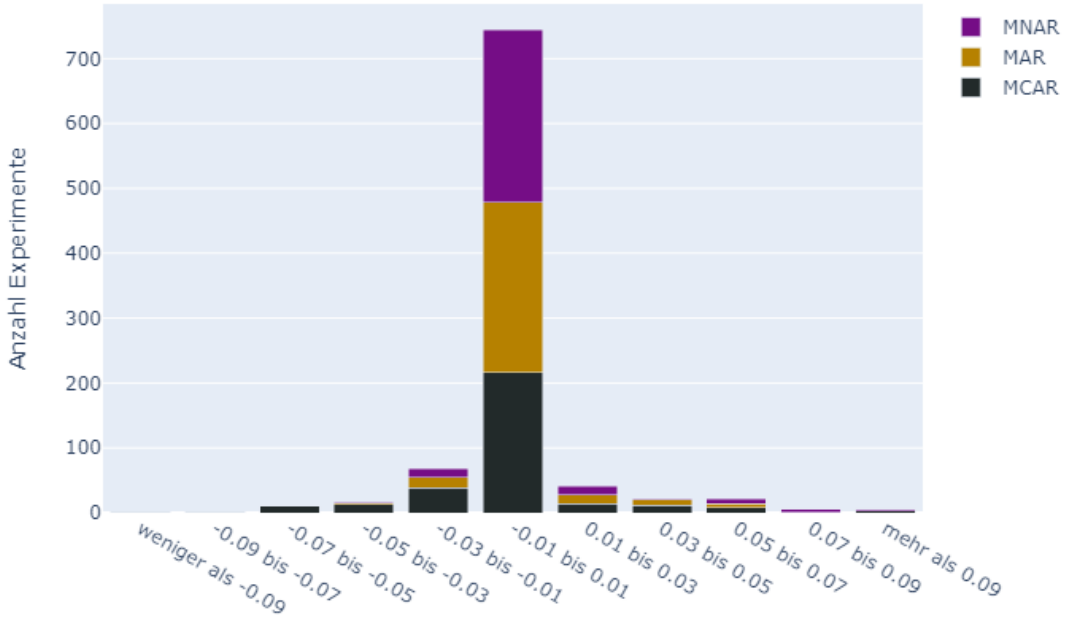

Abbildung 7.9: Differenz zum Ergebnis des durchschnittlichen Ergebnisses, gruppiert nach Muster der fehlenden Daten

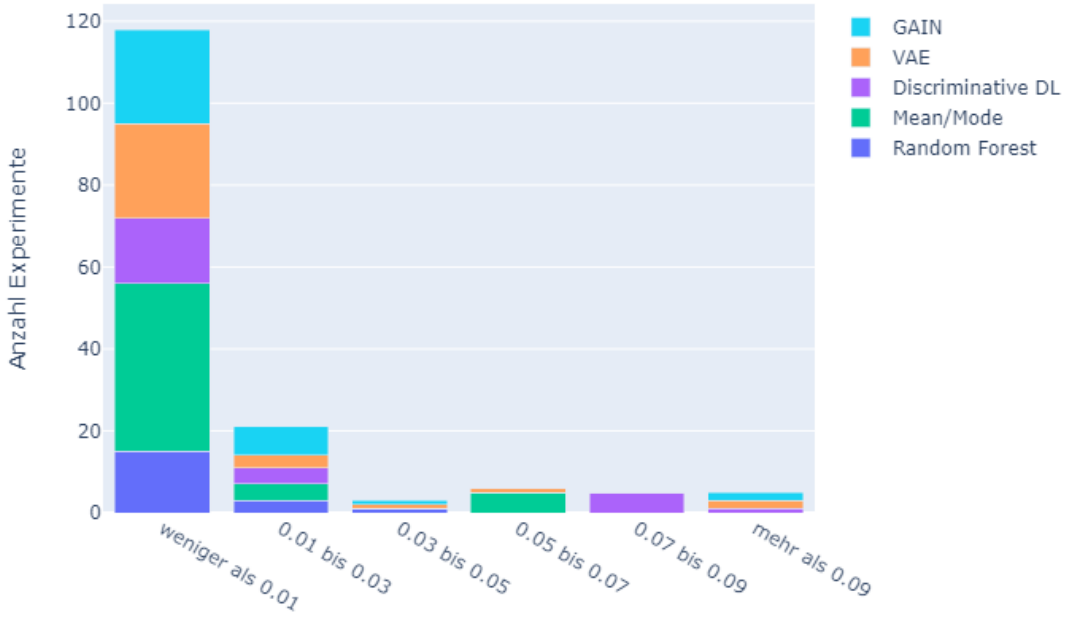

Abbildung 7.10: Differenz vom Ergebnis in Prozent des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Methode

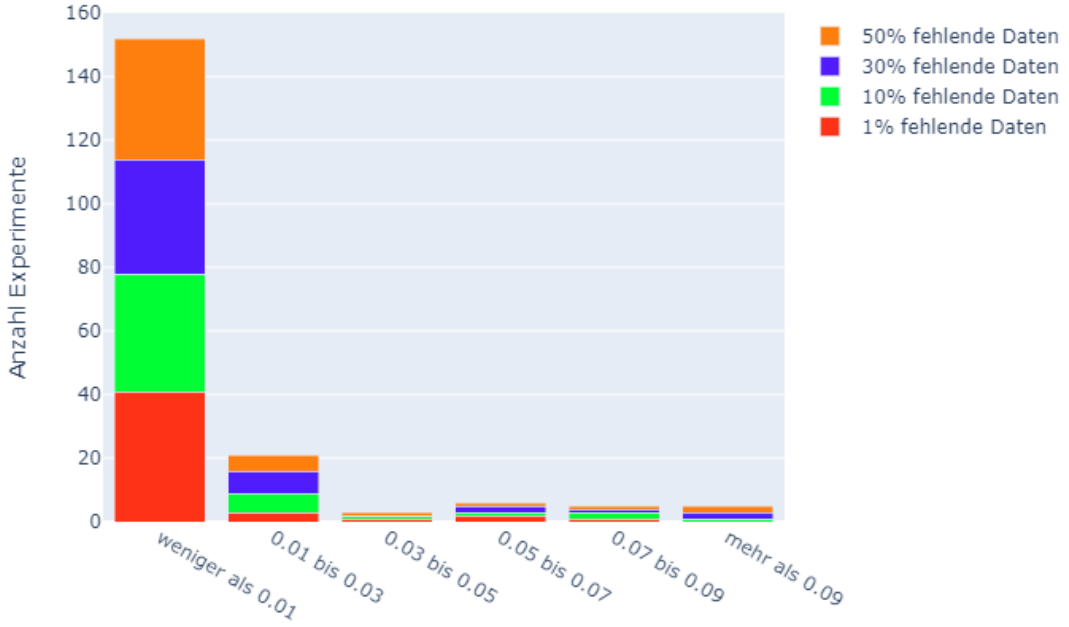

Abbildung 7.11: Differenz vom Ergebnis des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Anteil fehlender Daten

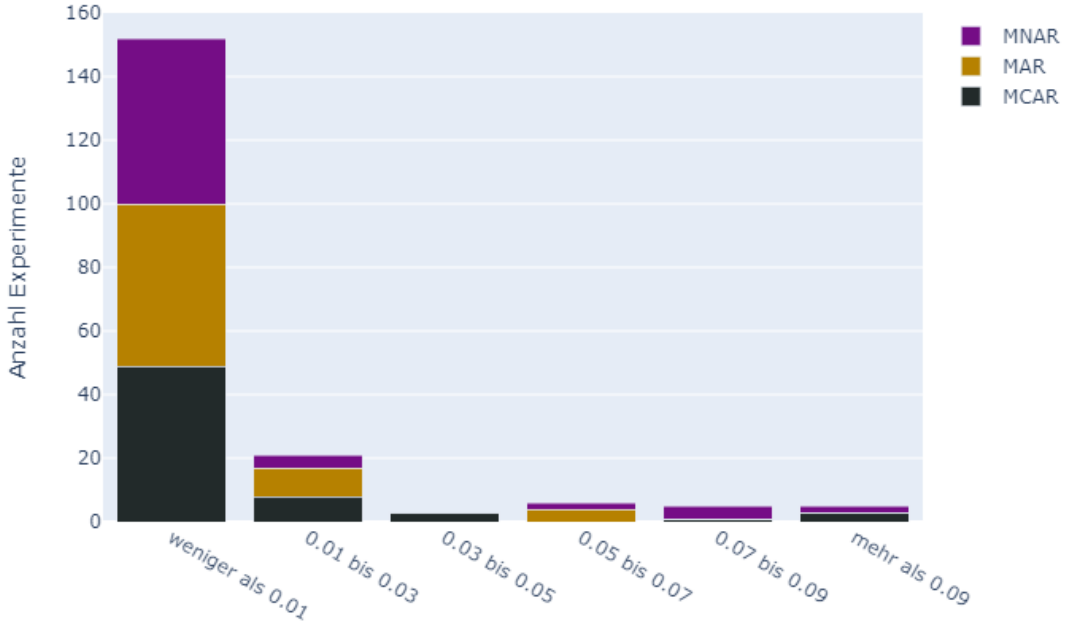

Abbildung 7.12: Differenz vom Ergebnis des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Muster der fehlenden Daten

7.1.3 Regression

Die Experimente mit Regression umfassen 19 Datensätze. Ursprünglich nutzte Jäger et al. (2021) 21 Datensätze, allerdings sind zwei davon nicht mehr verfügbar via OpenML(siehe 7.1). Die 19 Datensätze unterscheiden sich in der Größe von 4.500 bis zu fast 96.000 Beobachtungen und zwischen 6 und 22 Features. Das führt zu 228 verschiedenen Szenarien und somit 1368 Experimente. Davon waren 1308 erfolgreich, was einer Quote von 95,6% entspricht. Erneut verbuchte der GAIN Imputer mit 53 die meisten Abbrüche, was eine Fehlerquote von 23,3% bedeutet. Auch DDL verursachte 7 fehlerhafte Experimente(3% Fehlerquote).
Analysiert man die Ergebnisse des Rangsystems fällt sofort das sehr gute Ergebnis der DDL Methode auf. Mit einem Durchschnitt von 3,03 erreicht DDL klar den besten Wert aller Imputer. K-NN reiht sich dahinter auf Platz 2 ein mit einem Schnitt von 3,17. Die restlichen Imputer liegen im Durchschnittswert innerhalb eines Intervalls von 0,06 eng beieinander. Im Vergleich zu DDL und K-NN schneiden sie allerdings im durchschnittlichen Rang deutlich schlechter ab. Der DDL erreicht mit 56-mal Rang 1 auch am häufigsten das beste Ergebnis. Erwähnenswert ist noch VAE, die Methode ist im Durchschnitt der zweitschlechteste Imputer, erzielt allerdings 44 mal Rang 1 und damit am zweithäufigsten.

Imputationsmethode	Durchschnittlicher Rang	Anzahl Rang 1
DDL	3,03	56
K-NN	3,17	35
GAIN	3,50	35
Random Forest	3,54	28
VAE	3,55	44
Mittelwert/Modus	3,56	30

Tabelle 7.3: Überblick der Ergebnisse für Regression

Die durchschnittliche Differenz der Verbesserung von der besten Methode zu DDL beträgt einen RMSE von -0,013, was eine prozentuale Verschlechterung von 0.0004% zur Folge hat. Das bedeutet die DDL Methode ist im Durchschnitt besser als die jeweils beste Methode der übrigen Imputer.
Bei den Schaubildern 7.13, 7.15 und 7.17 fällt auf, dass fast alle Experimente im Intervall -0,01 bis 0,01 Prozent Verbesserung liegen. Um Details zu den Ausreisern zu erfahren muss man in die Grafik zoomen(siehe: 7.14, 7.16, 7.18) Dabei erkennt man, dass die Aufteilung der Methoden in das Ausreisern sehr ausgeglichen ist. Nur die Methode Random Forest hat keinen Ausreißer(7.14). Bei den Anteilen der fehlenden Daten erkennt man, dass nur Experimente mit den hohen Anteilen 30% und 50% Ausreiser enthalten(7.16). Außerdem bestehen die Outlier nur aus MCAR Experimenten(7.18). Bei den Grafiken 7.19, 7.20 und 7.21 gibt es nur ein Experiment, dass eine Verbesserung von über 0,01 Prozent RMSE bringt. Dies ist ein Mittelwert

Experiment mit dem Muster MCAR und dem höchsten Anteil fehlender Daten 50%.

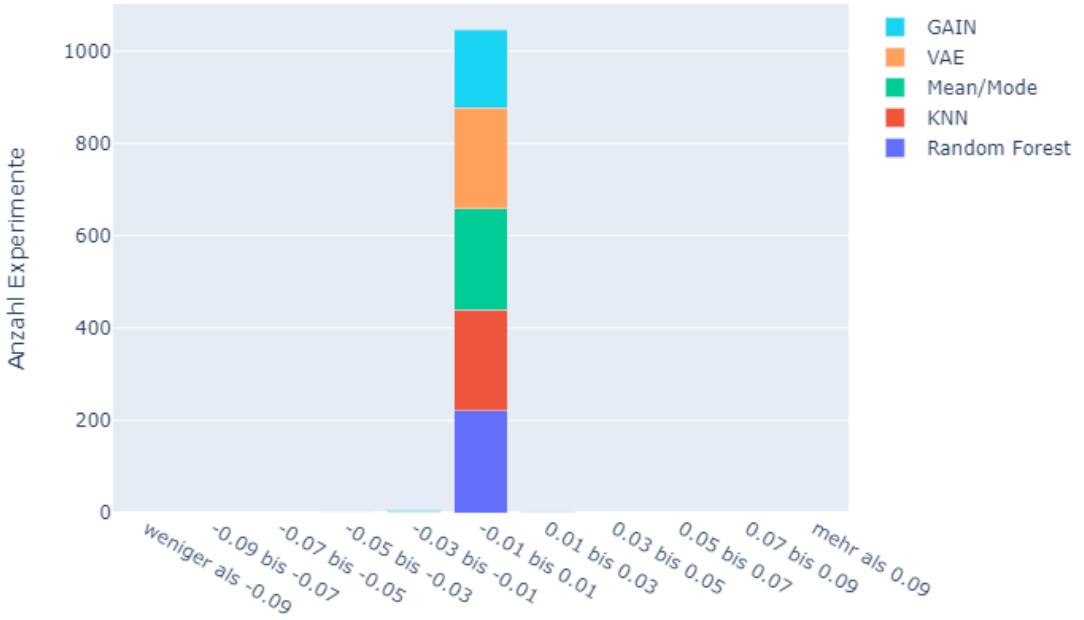

Abbildung 7.13: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Methode

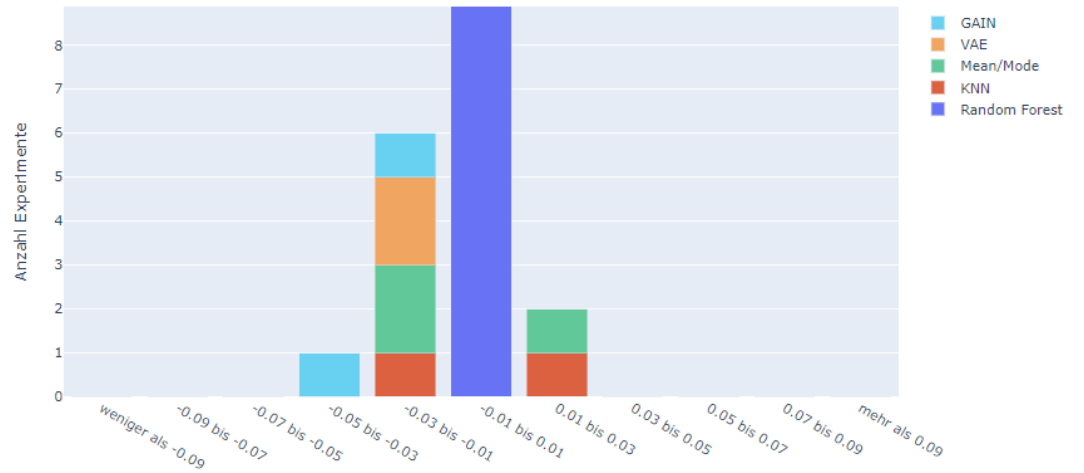

Abbildung 7.14: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Methode (stark vergrößert)

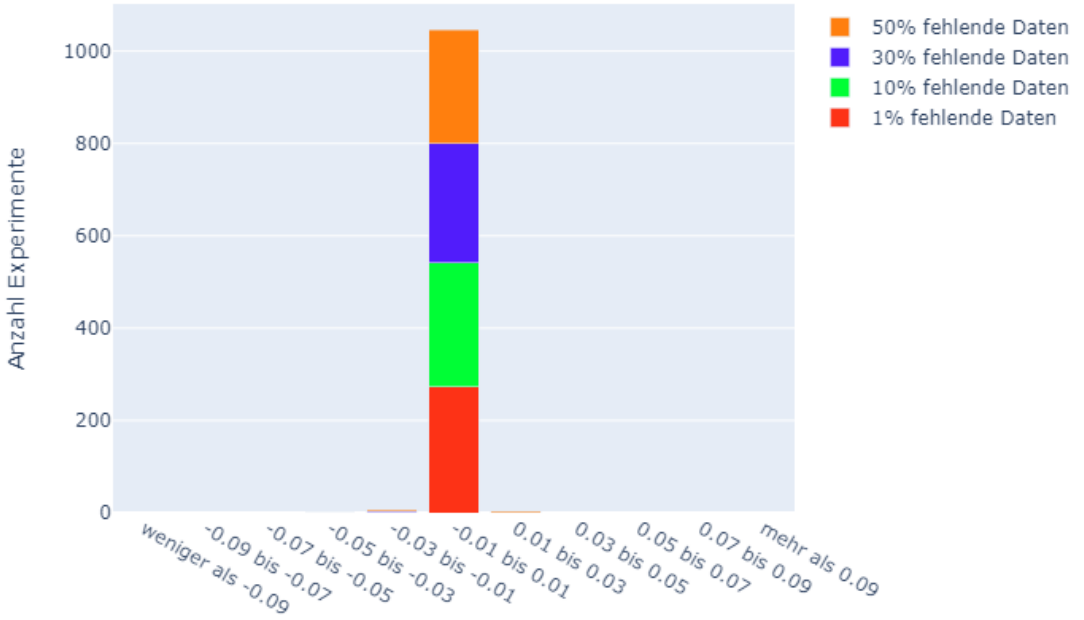

Abbildung 7.15: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Anteil fehlender Daten

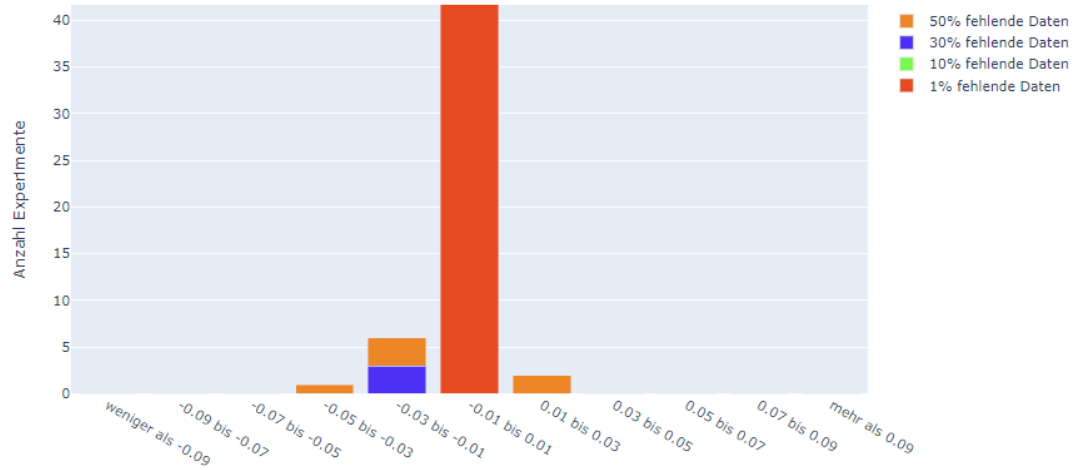

Abbildung 7.16: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Anteil fehlender Daten (stark vergrößert)

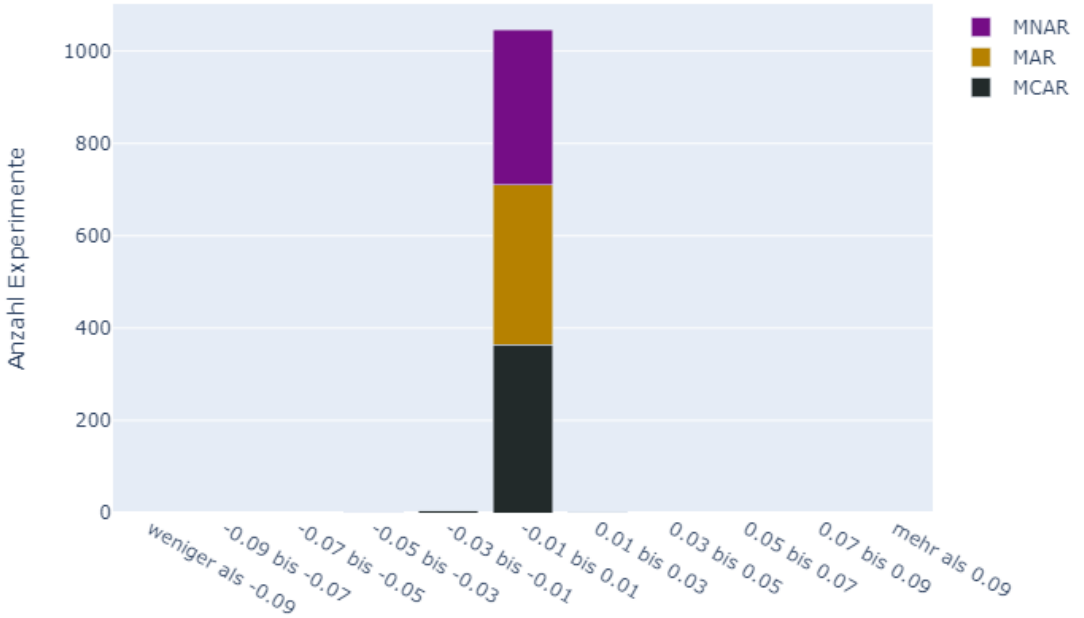

Abbildung 7.17: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Muster der fehlenden Daten

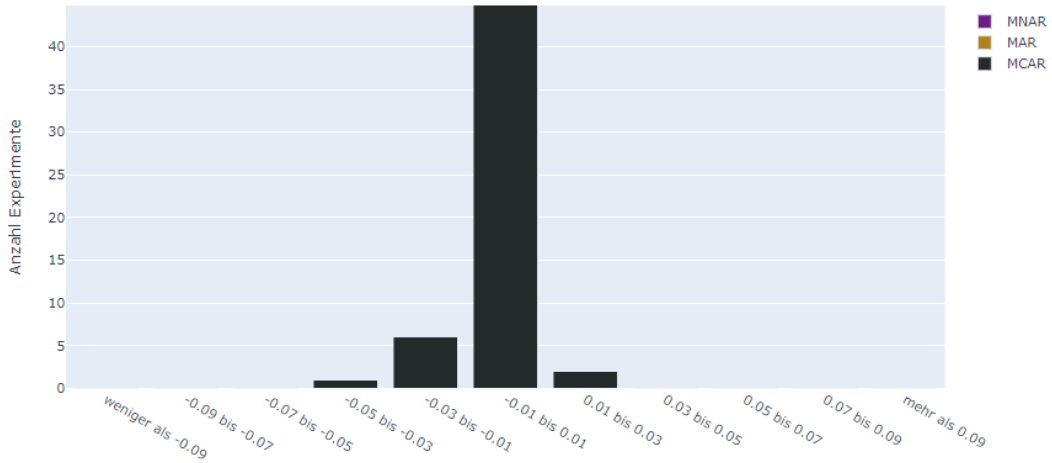

Abbildung 7.18: Differenz zum Ergebnis in Prozent des durchschnittlichen Ergebnisses, gruppiert nach Muster der fehlenden Daten (stark vergrößert)

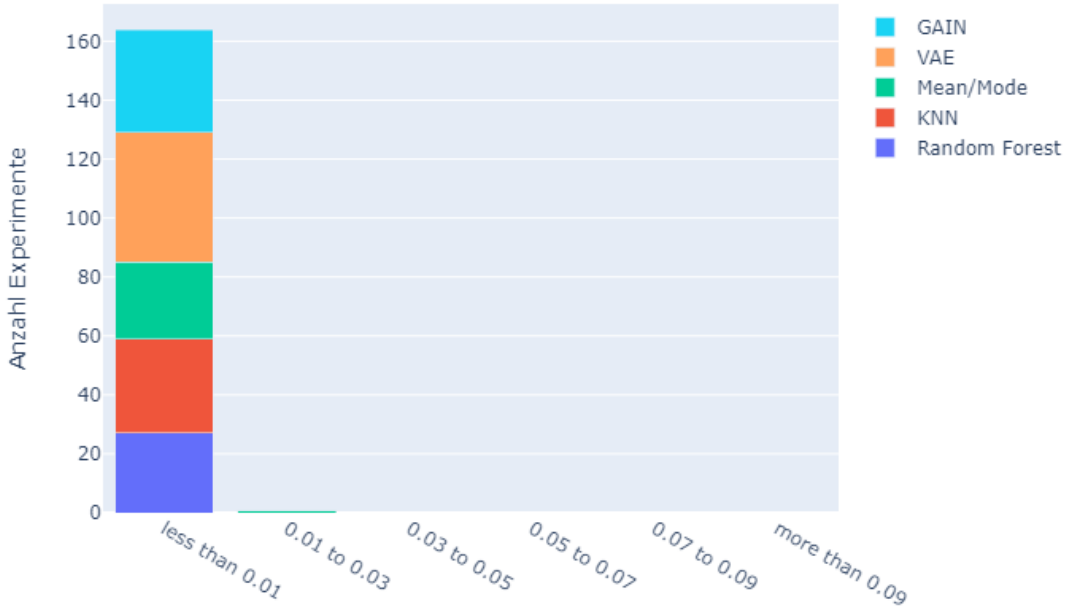

Abbildung 7.19: Differenz vom Ergebnis in Prozent des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Methode

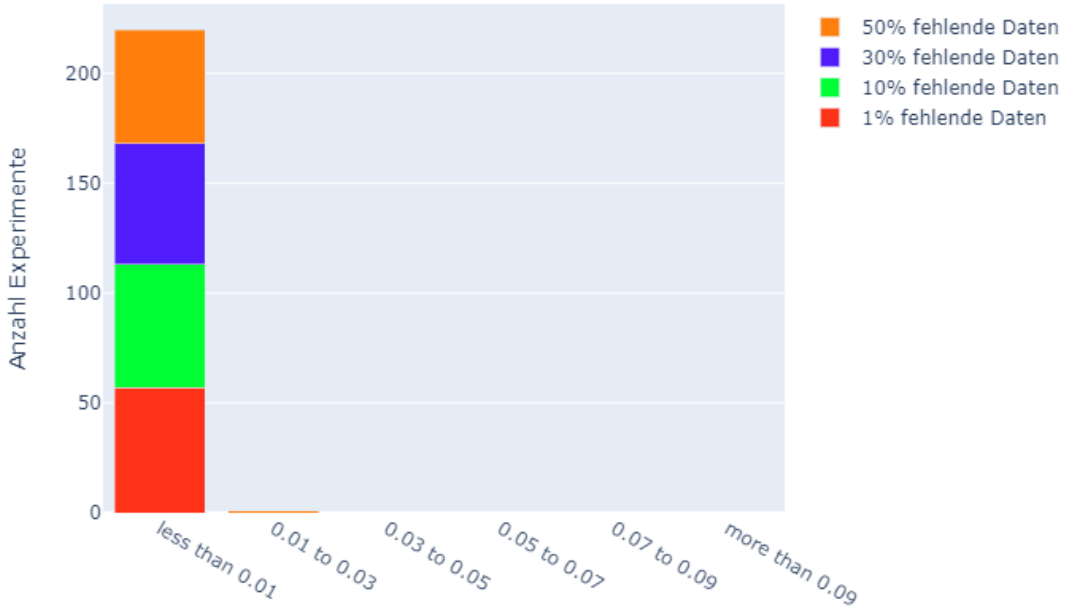

Abbildung 7.20: Differenz vom Ergebnis in Prozent des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Anteil fehlender Daten

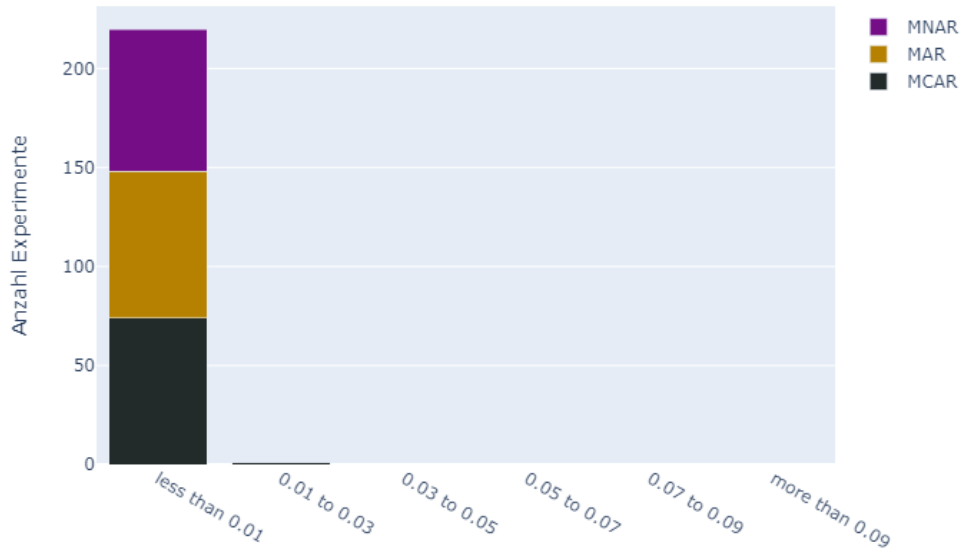

Abbildung 7.21: Differenz vom Ergebnis in Prozent des durchschnittlichen Ergebnisses zum Ergebnis des besten Imputers, gruppiert nach Muster der fehlenden Daten

7.2 Vergleich mit Dittrich, P. (2023)

Dieses Unterkapitel analysiert die Unterschiede zwischen den Ergebnissen von Dittrich, P. (2023) und dieser Arbeit. Zentraler Unterschied der beiden Arbeiten sind die unterschiedlichen Herangehensweisen bei der Datenkorruption der Experimente. Dittrich, P. (2023) manipulierte nur die Daten der Zielspalte, während bei dieser Arbeit Daten in allen Spalten korrumpiert wurden. Es werden sowohl die Ergebnisse des Rangsystems als auch die absoluten Ergebnisse miteinander verglichen. Die Ergebnisse von Binärer Klassifikationen, Mehrklassen Klassifikation und Regression werden wieder einzeln betrachtet.

7.2.1 Binäre Klassifikation

Vergleicht man die Durchschnittsränge der beiden Arbeiten in Tabelle 7.4 fallen die beiden extremen Änderungen von Mittelwert/Modus und DDL auf. Die Methode Mittelwert/Modus verbesserte sich im Durchschnitt um über einen Rang, während DDL sich um etwas weniger als einen Rang im Schnitt verschlechterte. Auch K-NN zeigte eine deutliche Verbesserung. Random Forest und VAE hatten kaum bis gar keine Änderungen im Durchschnittsrang. GAIN erzielte leicht schlechtere Ergebnisse im Vergleich.

Imputationsmethode	Einspaltig	Mehrspaltig	Differenz
Mittelwert/Modus	3,41	2,34	1,07
K-NN	3,15	2,64	0,51
Random Forest	3,00	3,01	-0,01
DDL	3,40	4,36	-0,96
VAE	3,70	3,70	0
GAIN	4,36	4,47	-0,11

Tabelle 7.4: Vergleich der Durchschnittsränge der unterschiedlichen Korruptionsarten

Die Visualisierungen von 7.22 zeigen die Änderungen der Ränge pro Imputationsmethode im Vergleich von Dittrich, P. (2023) zu dieser Arbeit. Ein Experiment erzielt eine Verbesserung, wenn sich der Rang gegenüber Dittrich, P. (2023) im jeweiligen äquivalenten Experiment in dieser Arbeit verbessert hat. Auffällig ist, dass jeweils circa 50% der Experimente der Methoden K-NN, Mittelwert/Modus und Random Forest eine Rangverbesserung verbuchen konnten. Diese Erkenntnis passt auch zu den positiven Differenzen in Tabelle 7.4. Analog erreichten die Methoden DDL, GAIN und VAE jeweils in knapp unter 50% aller Ergebnisse einen schlechteren Rang.

Abbildung 7.22: Vergleich der Rangergebnisse von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Vergleicht man die Imputationsleistung in Form des F1-Scores zwischen der einspaltigen und der mehrspaltigen Datenkorruption(siehe 7.23 fällt auf, dass jeweils in circa 60% der Fälle die Ergebnisse bei der mehrspaltigen Datenkorruption schlechter wurden. Dies lässt sich damit erklären, dass es die Modelle aufgrund der mehrspaltig gelöschten Daten schwerer haben ein genaues Modell zu trainieren. Nur in jeweils rund 30% der Experimente gab es eine Verbesserung im F1-Score.

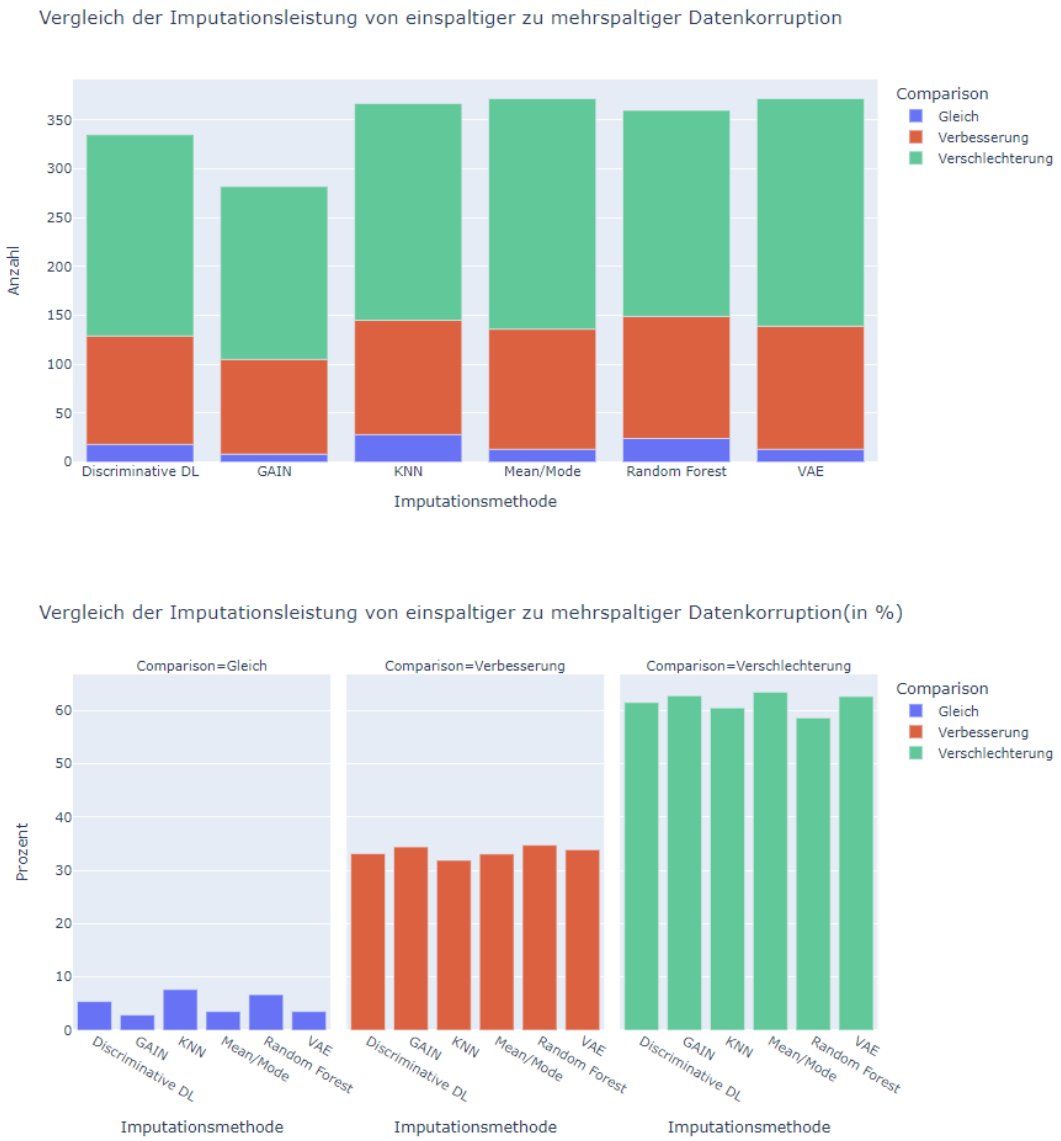

Abbildung 7.23: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Die Grafiken 7.24 und 7.25 vergleichen die Ergebnisse gruppiert nach Anteil fehlender Daten. Man erkennt, dass je größer der Anteil der fehlenden Daten wird, desto häufiger gibt es größere Unterschiede zwischen den Experimenten. Bei einem Anteil von 1% fehlender Daten gibt es noch sehr wenig Unterschiede zwischen den Experimenten und fast alle liegen im Intervall -0,01 bis 0,01. Im Vergleich dazu sieht man an der Grafik mit 50% fehlenden Daten, dass die Verteilung deutlich breiter wird. Interessanterweise tendiert die Verteilung sich nach rechts auszubreiten, was bedeutet, dass die Ergebnisse besser werden.

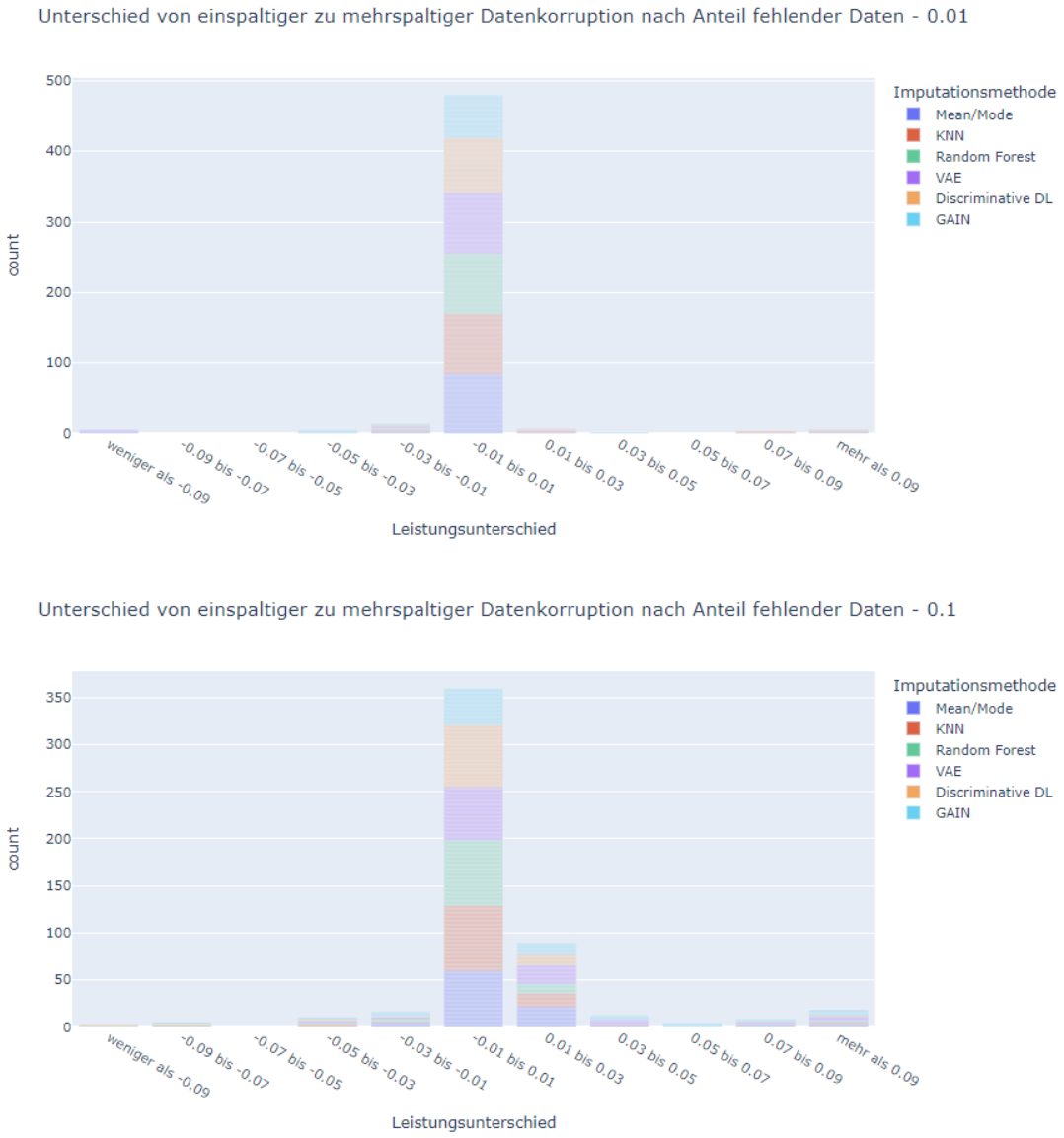

Abbildung 7.24: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, gruppiert nach kleinen Anteilen fehlender Daten(1% und 10%)

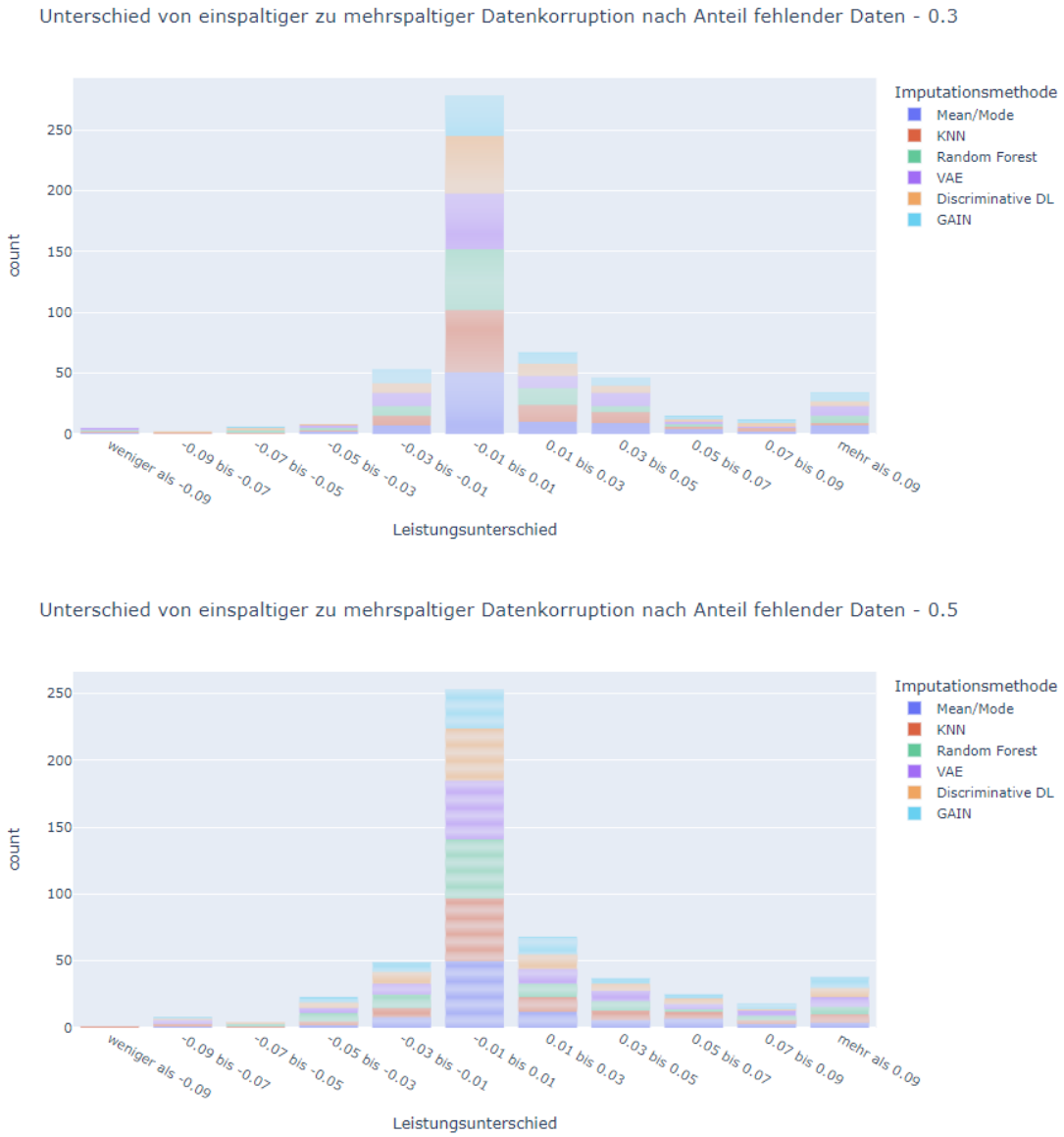

Abbildung 7.25: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, , gruppiert nach großen Anteilen fehlender Daten(30% und 50%)

7.2.2 Mehrklassen Klassifikation

Die Differenz der durchschnittlichen Ränge(siehe Grafik 7.5) zeigt, dass K-NN bei der Multiklassen Klassifikation am meisten verbessern konnte. Auch GAIN und DDL erzielten bessere Ergebnisse, während Mittelwert/Modus nur einen leicht besseren durchschnittlichen Rang erreichen konnte. Der Random Forest erhielt die größte Verschlechterung. Bei Dittrich, P. (2023) war die Methode auch die durchschnittlich Beste, während in dieser Arbeit der K-NN am Besten bei Multiklassen Klassifikationsaufgaben abschnitt. Auch VAE verschlechterte sich gegenüber der einspaltigen Datenkorruption.

Imputationsmethode	Einspaltig	Mehrspaltig	Differenz
Mittelwert/Modus	3,37	3,31	0,06
K-NN	3,32	2,96	0,36
Random Forest	3,27	3,58	-0,31
DDL	3,75	3,59	0,16
VAE	3,48	3,67	-0,19
GAIN	3,80	3,59	0,21

Tabelle 7.5: Vergleich der Durchschnittsränge der unterschiedlichen Korruptionsarten

Schaut man sich die Rangveränderungen in Grafik 7.26 an, fällt auf, dass der DDL in über 50% der Experimente einen schlechteren Rang erzielen konnte. Auch GAIN verschlechterte sich in knapp 45% der Fällen. Am besten fiel der Vergleich bei Mittelwert/Modus aus. Die Methode konnte sich in über 50% der Experimente verbessern. Das passt auch zum Ergebnis in 7.1.2.

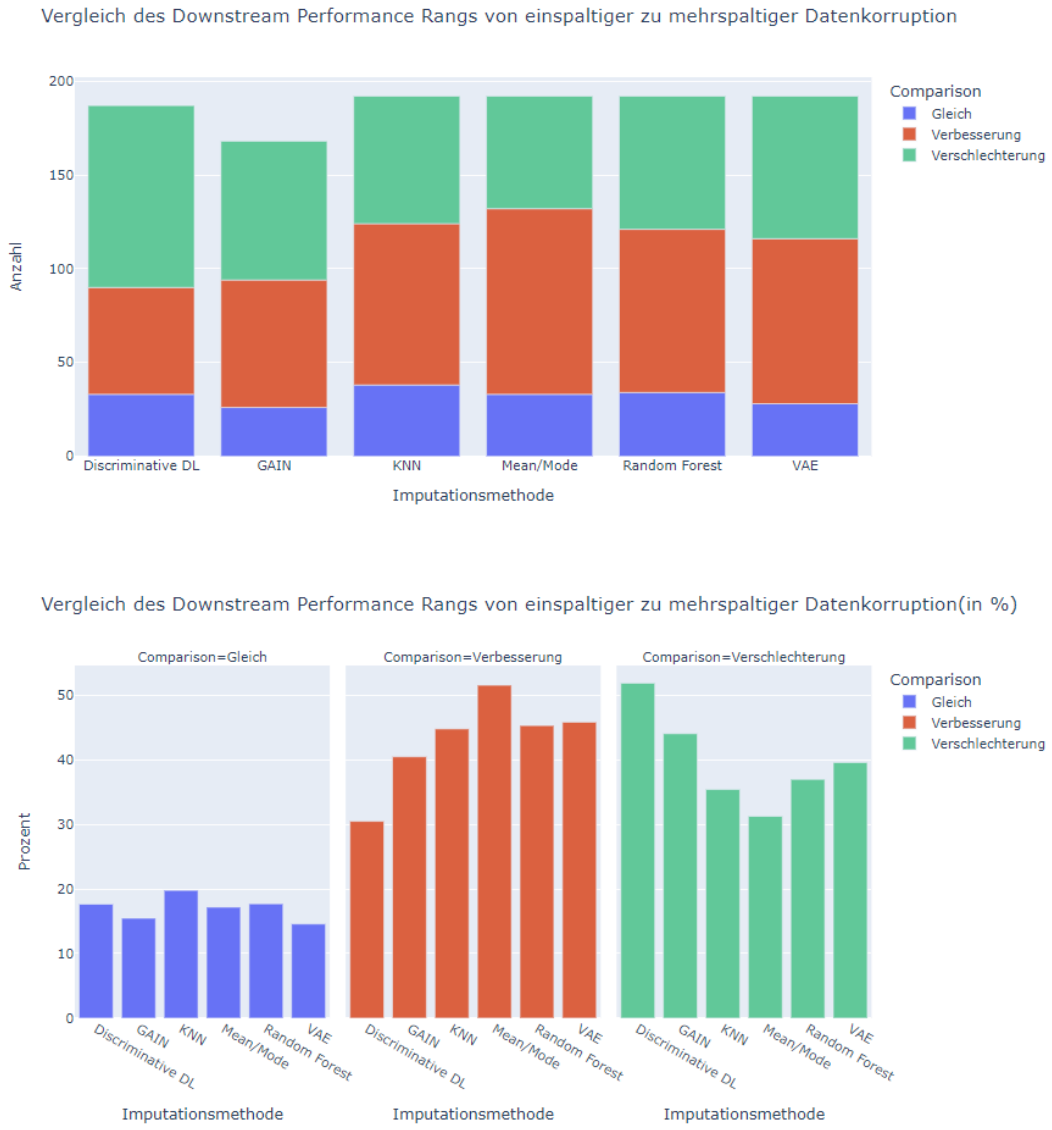

Abbildung 7.26: Vergleich der Rangergebnisse von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Vergleicht man jeweils die reine Verbesserung oder Verschlechterung des Imputationsergebnisses(siehe 7.27) fällt auf, dass der VAE prozentual die meisten Verbesserungen erzielen konnte. Die DDL Methode erzielte im Verhältnis am häufigsten eine Verschlechterung.

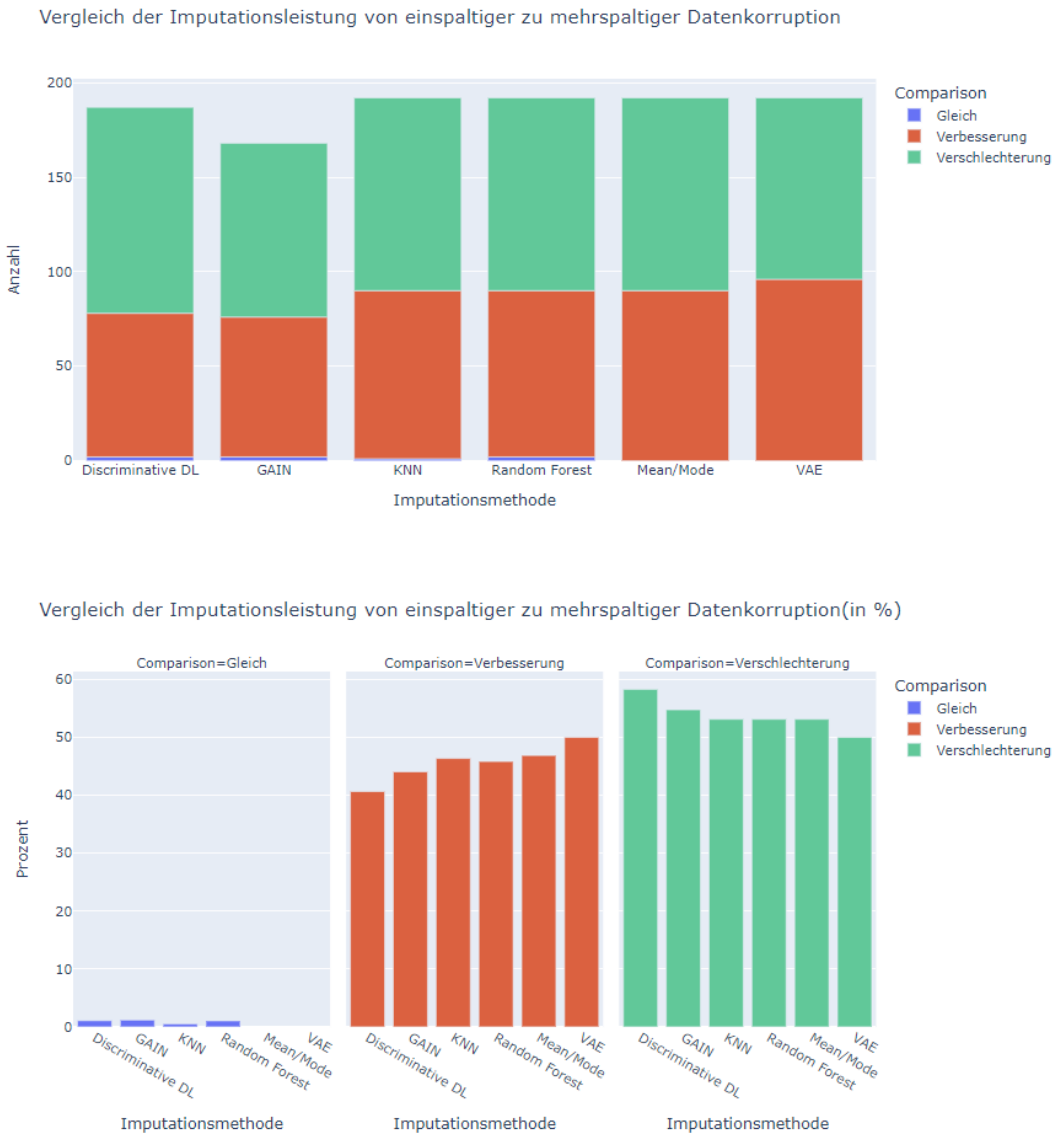

Abbildung 7.27: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Wie schon bei der Analyse der Binären Klassifikation(7.2.1) wird die Abweichung beim Vergleich der Ergebnisse größer, wenn sich der Anteil der fehlenden Daten erhöht(Grafiken 7.28, 7.29). Während sich die Verteilung bei der Binären Klassifikation neigte nach rechts auszubreiten, sind die Verteilungen bei der Mehrklassen Klassifikation ausgeglichener. Allerdings gibt es bei den Diagrammen mit 30% und 50% Anteil an fehlenden Daten(7.29) eine größere Anzahl an Ausreißern mit einer Differenz von mehr als -0,09.

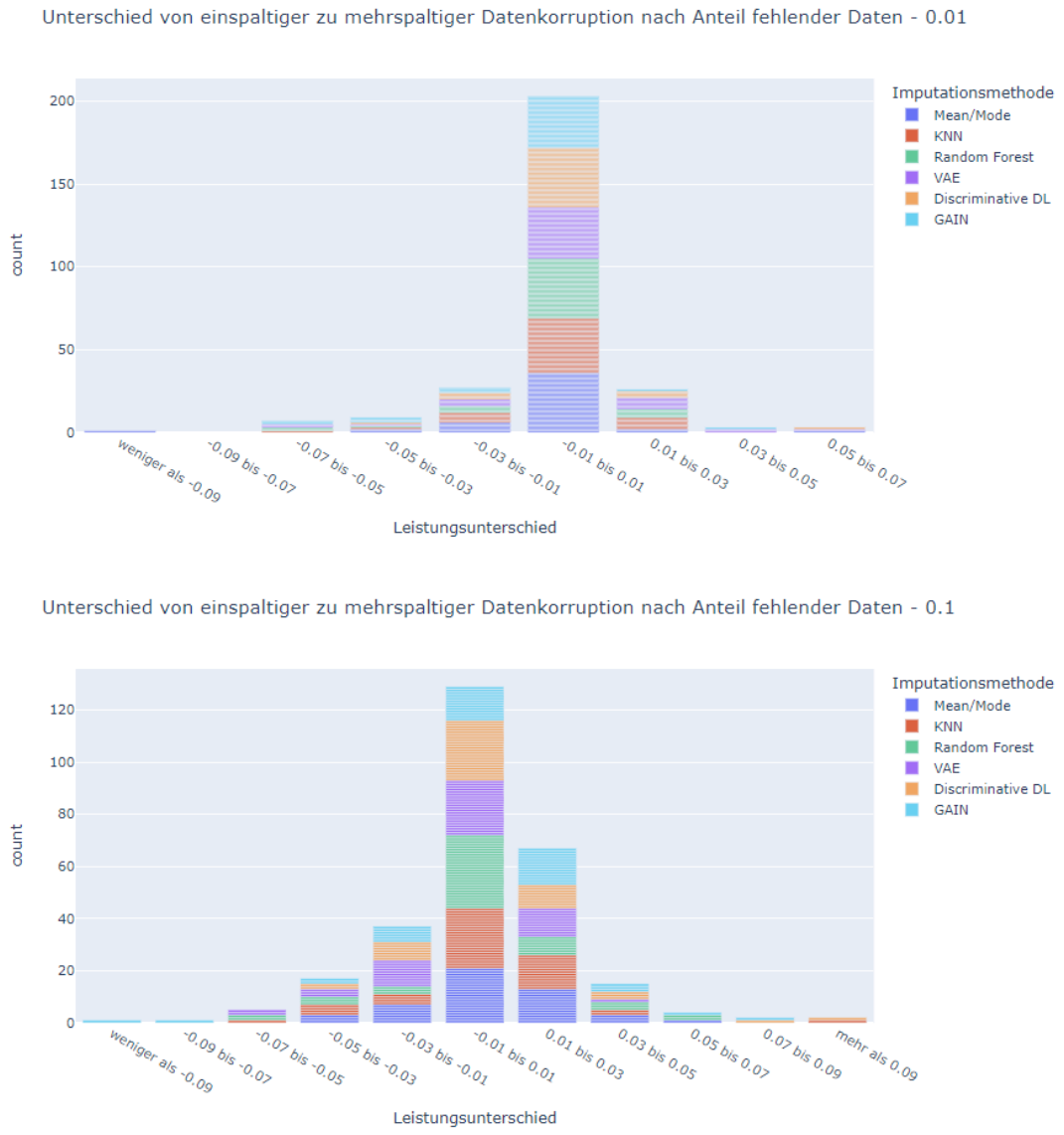

Abbildung 7.28: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, gruppiert nach kleinen Anteilen fehlender Daten(1% und 10%)

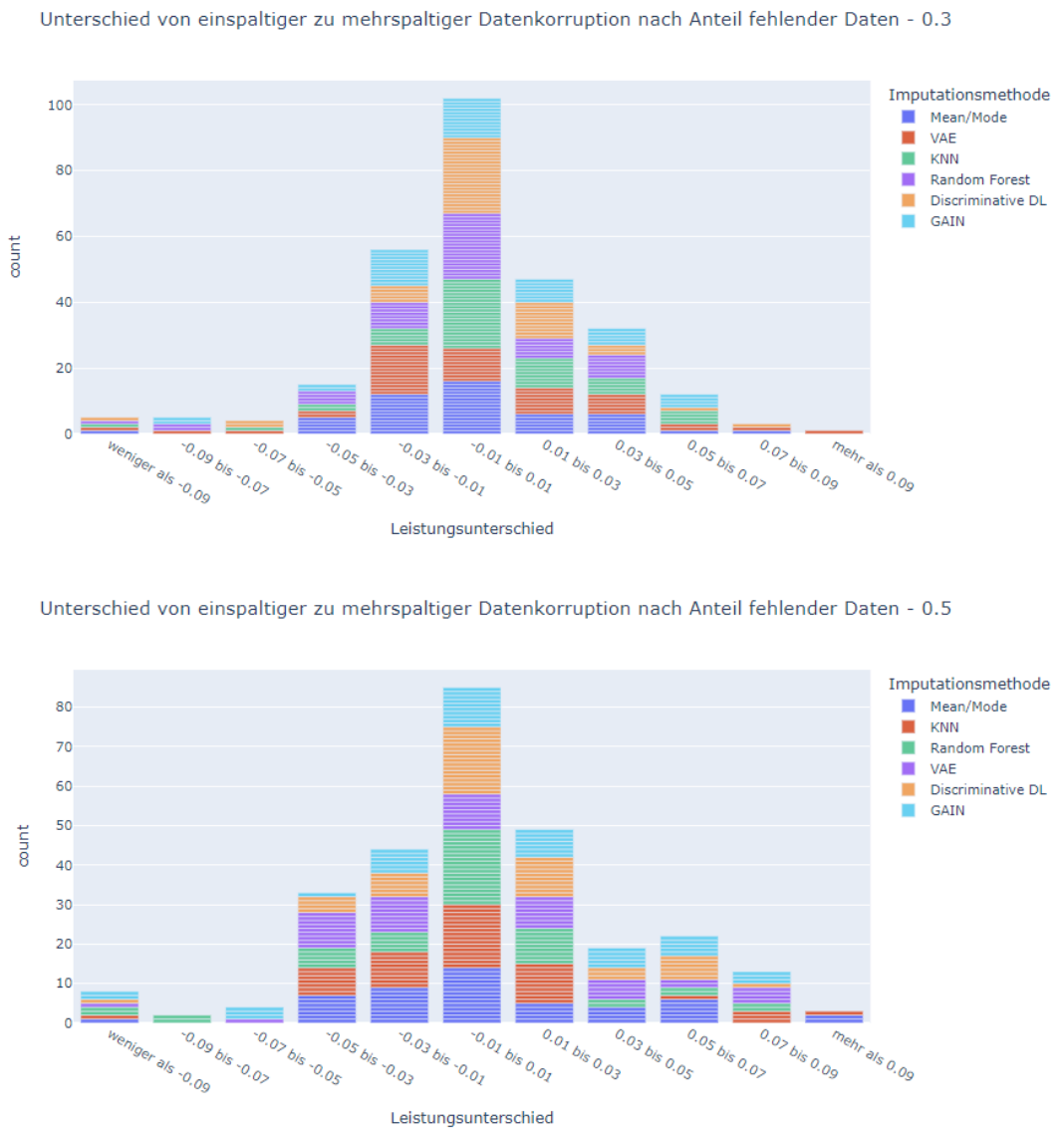

Abbildung 7.29: Vergleich der Imputationsergebnisse(F1-Score) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, gruppiert nach großen Anteilen fehlender Daten(30% und 50%)

7.2.3 Regression

Bei den Änderungen der Durchschnittsränge der Methoden bei Regressionsexperimenten(siehe 7.6 sticht die klare Verbesserung von GAIN heraus. Auch VAE, K-NN und DDL lieferten einen besseren Durchschnittsrang. Mittelwert/Modus verschlechterte sich, genauo wie die Methode Random Forest, welche mit -0,44 die negativste Differenz erzielte.

Imputationsmethode	Einspaltig	Mehrspaltig	Differenz
Mittelwert/Modus	3,30	3,56	-0,26
K-NN	3,30	3,17	0,13
Random Forest	3,10	3,54	-0,44
DDL	3,12	3,03	0,09
VAE	3,80	3,55	0,25
GAIN	4,37	3,50	0,87

Tabelle 7.6: Vergleich der Durchschnittsränge der unterschiedlichen Korruptionsarten

Grafik 7.30 zeigt die einzelnen Rangunterschiede der einzelnen Imputationsmethoden je Experiment. Hier fällt wie schon in Tabelle 7.6 auf, dass GAIN die meisten Verbesserungen im Rang erzielen konnte und Random Forest sowohl die wenigsten Verbesserungen als auch die meisten Verschlechterungen verbuchte. Neben GAIN zeigten auch K-NN, VAE und DDL in über 40% der Experimente eine Verbesserung.

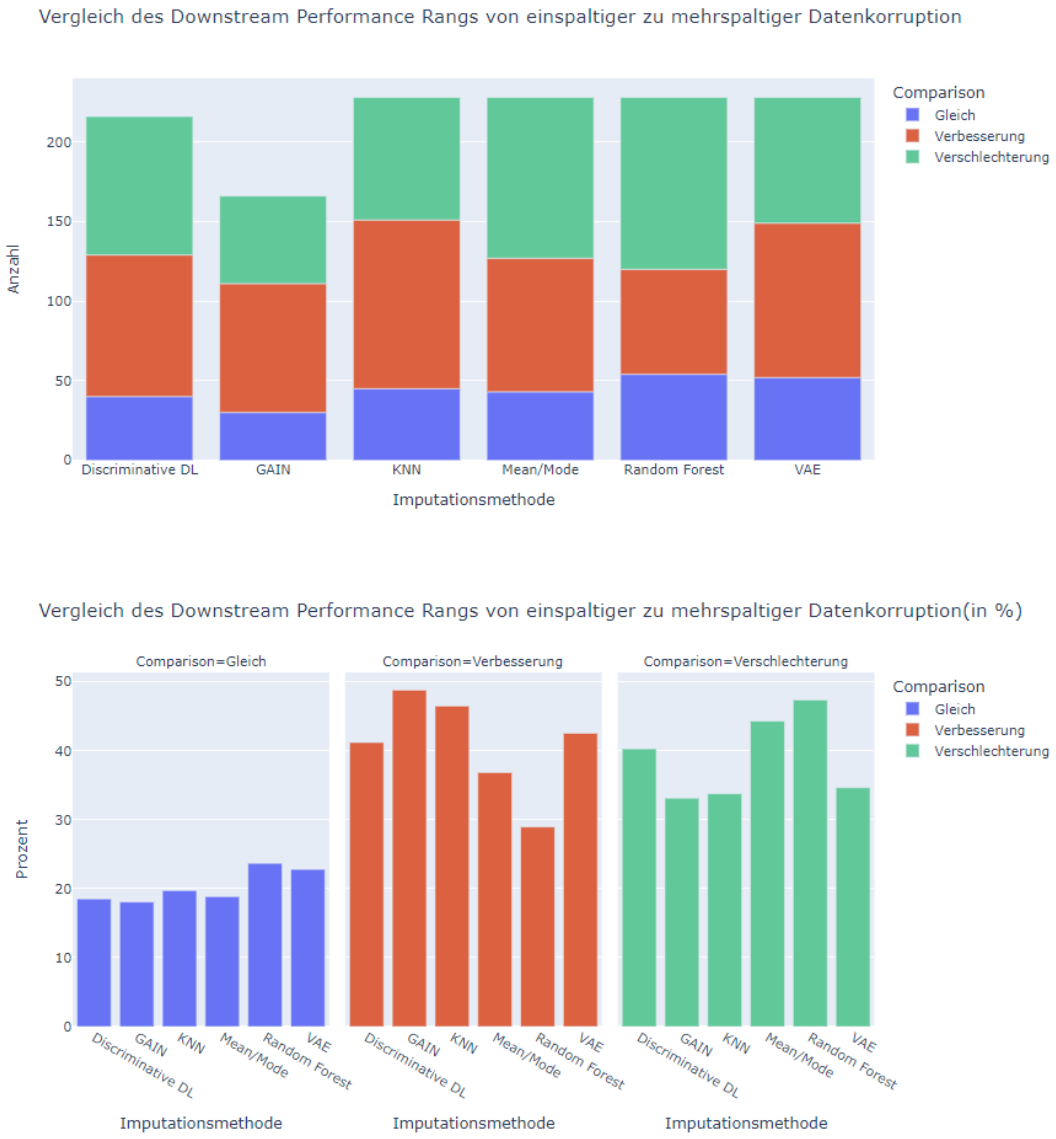

Abbildung 7.30: Vergleich der Rangergebnisse von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Untersucht man die absoluten Verbesserung der Imputationsergebnisse nach dem RMSE Wert, sieht man in Schaubild 7.31, dass die Verbesserungen im Vergleich zu den beiden Klassifikationskapiteln sehr ausgeglichen sind. Jede Methode bewegt sich im Bereich der 50% Verbesserungen. Am auffälligsten ist der VAE Imputer, welcher die meisten Verbesserungen erzielen konnte.

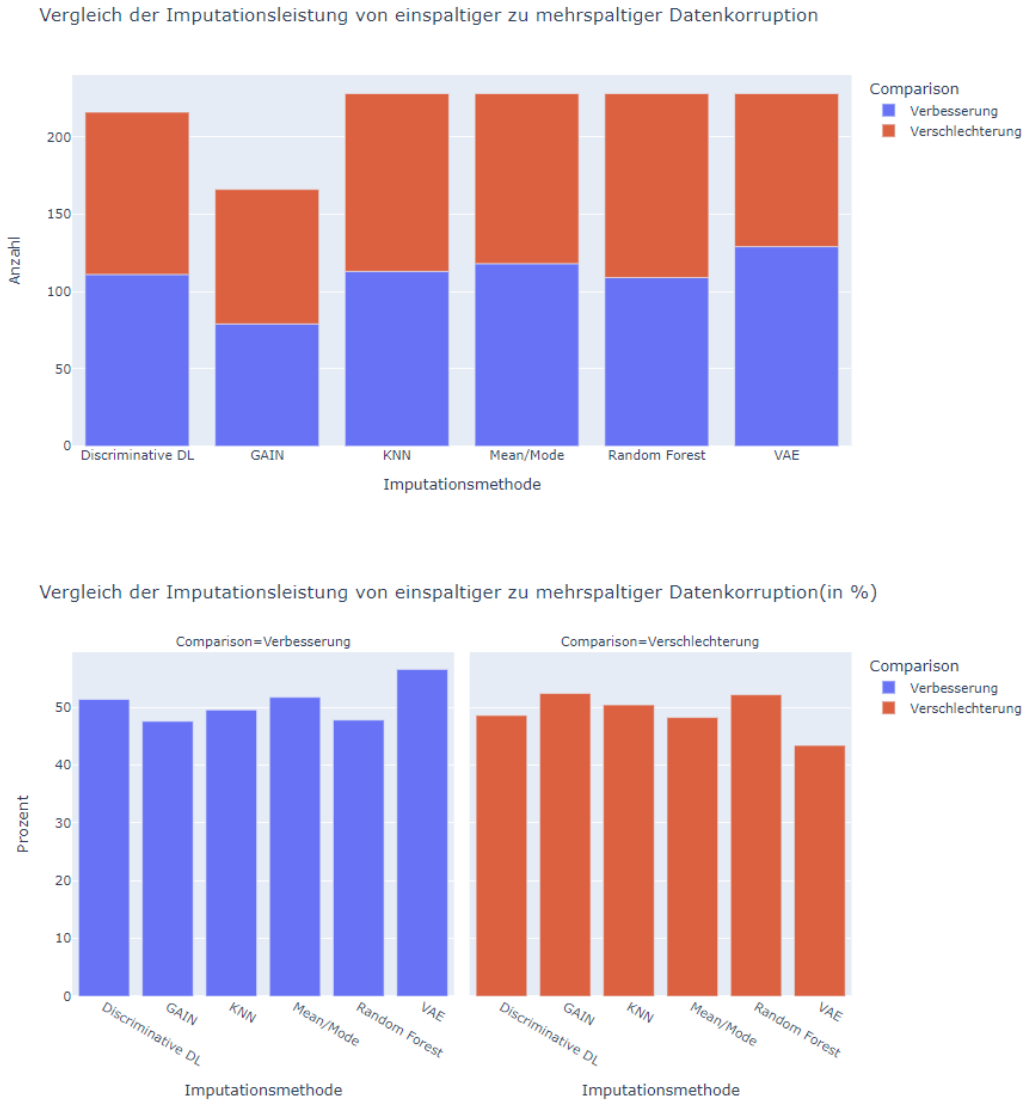

Abbildung 7.31: Vergleich der Imputationsergebnisse(RMSE) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente

Im Vergleich zur Binären und Multiklassen Klassifikation fällt bei 7.32 und 7.33 auf, dass es schon bei dem kleinsten Anteil an fehlenden Daten(1%) Ausreiser gibt. Diese nehmen auch mit größer werdendem Anteil fehlender Daten zu. In Bild 7.33 sieht man wie der Großteil der Daten im Intervall -0,01 bis 0,01 liegt, also bei nahezu keiner Änderung. Gleichzeitig liegen zwischen diesem Intervall und den beiden Extremen weniger als -0,09 und mehr als 0,09 vergleichsweise wenig Experimente. Da die Verteilung der Methoden innerhalb der Extreme sehr ausgeglichen ist, wird der Grund dafür datensatzbedingt sein.

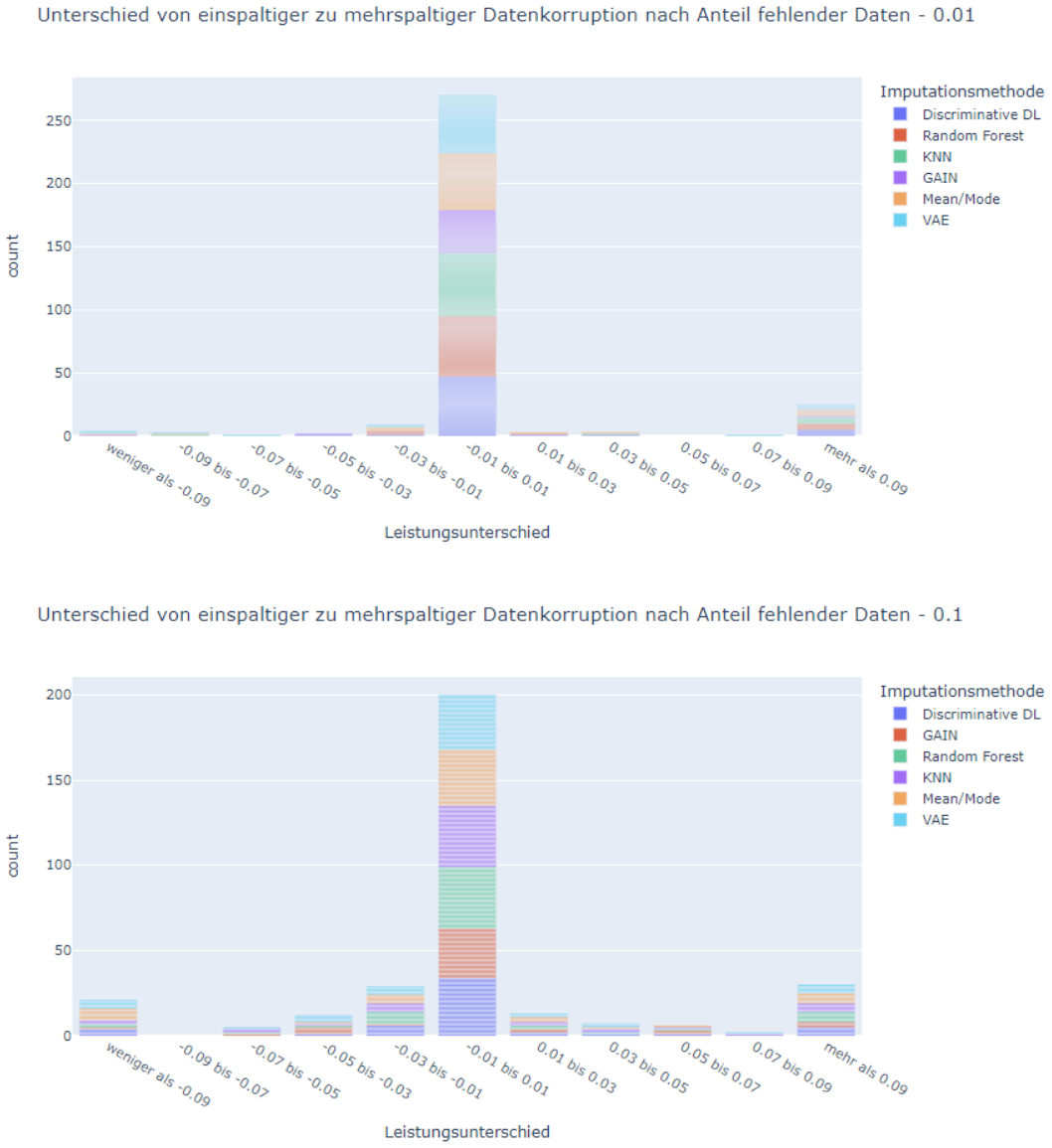

Abbildung 7.32: Vergleich der Imputationsergebnisse(RMSE) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, gruppiert nach kleinen Anteilen fehlender Daten(1% und 10%)

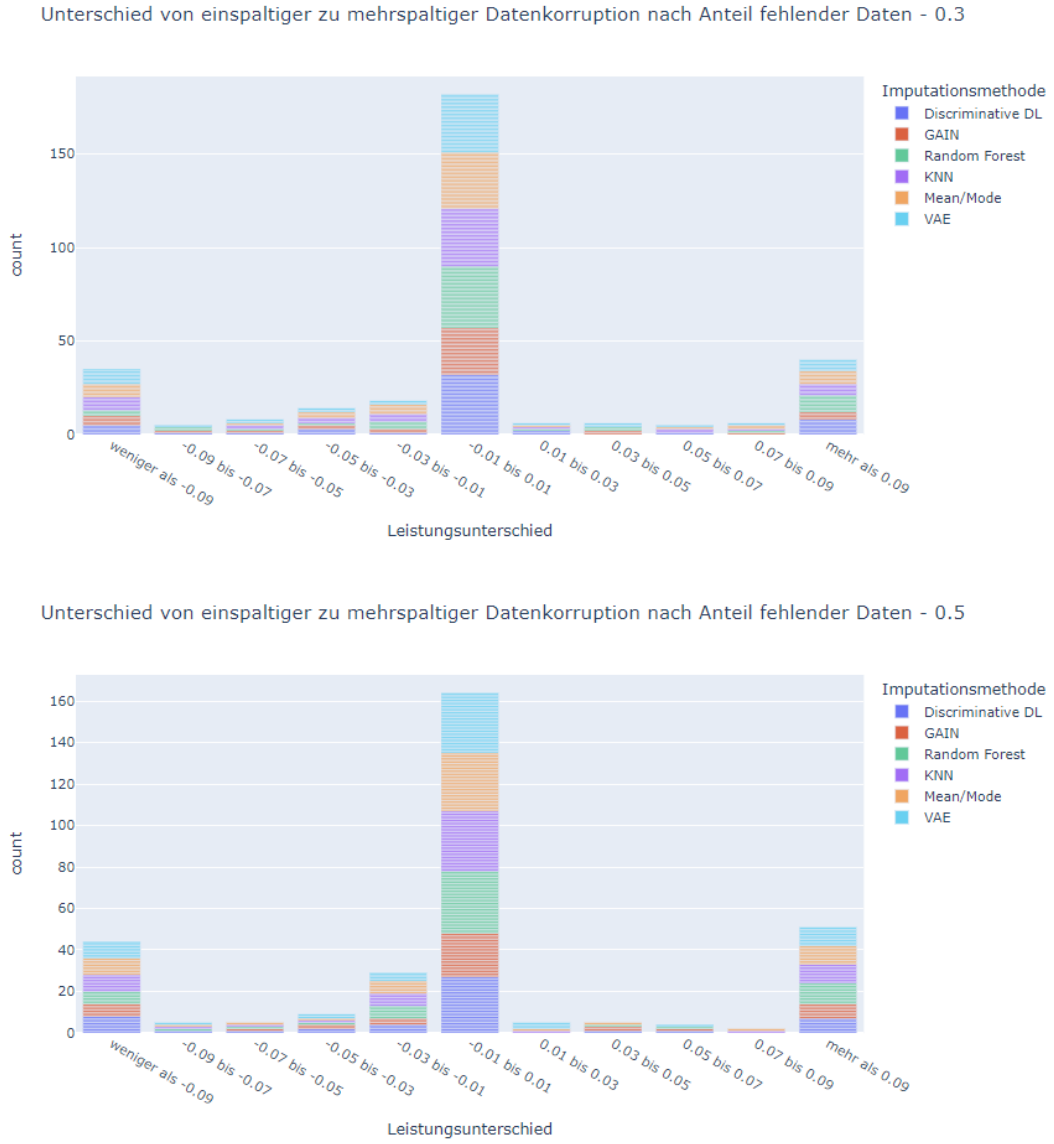

Abbildung 7.33: Vergleich der Imputationsergebnisse(RMSE) von einspaltiger zu mehrspaltiger Datenkorruptionsexperimente, gruppiert nach großen Anteilen fehlender Daten(30% und 50%)

8 Fazit und Diskussion

Die vorliegende Arbeit fokussierte sich auf die Analyse von Imputationsmethoden sowohl im Kontext von mehrspaltiger Datenkorruption, als auch im Vergleich mit vorliegenden Ergebnissen von einspaltiger Datenkorruption. Durch die Anwendung dieser Methoden auf Datensätzen mit binärer Klassifikation, mehrklassiger Klassifikation und Regression wurden vielfältige Erkenntnisse gewonnen. Eine umfassende Zusammenführung der Ergebnisse, ermöglicht eine abschließende Bewertung der Leistungsfähigkeit der verschiedenen Imputationsansätze.

8.1 Datenkorruption mehrerer Merkmale

Die Experimente der Imputationsmethoden nach der Datenkorruption mehrerer Merkmale zeigte unterschiedliche Ergebnisse.
Zum einen lassen sich die Ergebnisse der binären und mehrklassigen Klassifikation zusammenfassen. Hier schnitten die einfacheren Imputationsmethoden in Form von Mittelwert/Modus und K-NN am besten ab. In Kombination mit ihren geringen Rechenkosten können die beiden Imputationsmethoden als beste Strategien bei Klassifikationsaufgaben bezeichnet werden. Nicht zu empfehlen sind hingegen klar die Deep Learning Modelle VAE, DDL und GAIN. Neben fehlgeschlagenen Experimenten waren auch die durchschnittlich erreichten Ergebnisse deutlich schlechter als von K-NN und Mittelwert/Modus. In der absoluten Leistung konnte zwar nachgewiesen werden, dass die Imputationsleistungen nah beieinander liegen und es fast nie signifikant schlechtere Ergebnisse gibt, allerdings haben K-NN und Mittelwert/Modus neben dem konstant leicht besseren Ergebnis auch deutlich geringere Rechenkosten.
Bei den Regressionsdatensätzen zeichnete sich ein leicht anderes Ergebnis ab. Hier konnte sich der DDL deutlich steigern und erzielte das beste durchschnittliche Ergebnis. Wenn man Kapazitäten für hohe Rechenkosten hat, kann der DDL Imputer als beste Strategie gewählt werden. Da die Ergebnisse allerdings wieder sehr nah beieinander lagen, kann im Fall von begrenzten Rechenkapazitäten auch die ressourcenschonende Methode K-NN wieder ausgewählt werden. Abschließend kann man den K-NN Imputer hervorheben, welcher in allen drei Szenarien, Binäre und Mehrklassen Klassifikation sowie Regression, als einzige Methode immer zu den zwei besten Methoden zählte und somit die am meisten konstante Imputationsmethode war.

8.2 Vergleich der unterschiedlichen Datenkorruptionsmethoden

Der Vergleich zwischen einspaltiger und mehrspaltiger Datenkorruption zeigte, dass der Unterschied der Korruptionsmethode durchaus Auswirkungen auf die Leistungen der Imputer hat.
Bei der Binären Klassifikation fiel auf, dass Mittelwert/Modus, K-NN und Random Forest im Vergleich zu den anderen Methoden eine verbesserte Performance aufwies, während DDL, GAIN und VAE schlechtere Ergebnisse erzielten. Insgesamt ließ sich feststellen, dass die Korruptionsstrategie negative Auswirkung auf die absolute Imputationsleistung hatt und die Ergebnisse bei der mehrspaltigen Datenkorruption schlechter wurden.
Die Unterschiede bei der mehrklassigen Klassifikation zeigten moderatere Änderungen. GAIN und K-NN zeigte die vergleichsweise größten Leistungssprünge im Rangsystem. Während Mittelwert/Modus die prozentual meisten Verbesserungen erreichte. Random Forest büßte am meisten im Ergebnis des durchschnittlichen Rangs ein und DDL hatte die prozentual meisten Rangverschlechterungen. In der absoluten Imputationsleistung nach F1-Score hatten alle Methoden mehr Verschlechterungen als Verbesserungen. Also wirkte sich die geänderte Korruptionsstrategie negativ auf die Leistung der Imputationsmethoden aus.
Beim Blick auf die Änderungen bei Regressionsdatensätze fällt auf, dass GAIN das sehr schlechte Abschneiden bei einspaltiger Datenkorruption kaschieren konnte und im Ergebnis des durchschnittlich erzielten Rangs gegenüber den anderen Imputationsmethoden deutlich aufholen konnte. Analog rutschte Random Forest ab. Bei Dittrich, P. (2023) noch die durchschnittlich beste Methode, verschlechterte sich Random Forest um 0,44 Ränge. Die Imputationsleistung aller Methoden blieb bei der Regression gegenüber der einspaltigen Datenkorruption am meisten konstant. Hier hat die Korruptionsstrategie keine signifikant negativen Auswirkungen auf die Leistung.

9 Zusammenfassung und Ausblick

Die vorliegende Arbeit hat sich mit der Evaluation von Imputationsmethoden in Bezug auf mehrspaltige Datenkorruption beschäftigt und dabei wertvolle Erkenntnisse gewonnen. Im Folgenden werden die wichtigsten Ergebnisse zusammengefasst und ein Ausblick auf mögliche zukünftige Forschungsrichtungen gegeben.
Die durchgeführten Experimente zeigten, dass die Imputationsmethoden unterschiedlich auf mehrspaltige Datenkorruption reagieren, abhängig von der Art der Aufgabe (binäre Klassifikation, mehrklassige Klassifikation, Regression) und der spezifischen Methode. In Klassifikationsaufgaben schnitten einfachere Imputationsansätze wie Mittelwert/Modus und K-NN tendenziell besser ab, während Deep Learning-Modelle wie VAE, DDL und GAIN schlechtere Leistungen zeigten. Bei Regressionsdatensätzen konnten DDL und K-NN überzeugen. Insgesamt erzielte K-NN die konstantesten Ergebnisse. Das bestätigt auch die Arbeit der vorhergehenden und verwandten Arbeiten(siehe 4 und 5), welche herausarbeiteten, dass einfachere Imputationsmethoden im Vergleich mit komplexeren Ansätzen konkurrenzfähige Ergebnisse erzielen können.
Die unterschiedlichen Szenarien und die Vielzahl an Datensätzen sorgten für eine realistische und faire Umgebung. Für zukünftige Forschungen wäre eine Erweiterung der Imputationsmethoden interessant. Die vorhergehenden Arbeiten(4) hatten mit beispielsweise mit SVM (Bertsimas, et al., 2018) oder Bayesianischer linearer Regression (Jadhav et al., 2019) alternative Imputationsmethoden, die man in dieser Umgebung untersuchen könnte.

Literatur

Bertsimas, et al. (2018). *From Predictive Methods to Missing Data Imputation: An Optimization Approach*. https://www.jmlr.org/papers/volume18/17-073/17-073.pdf

Dittrich, P. (2023). *Assessing and Predicting the Optimal Imputation Method Regarding the Predictive Performance of Machine Learning Models*.

Emmanuel et al. (2021). *A survey on missing data in machine learning*. https://doi.org/10.1186/s40537-021-00516-9

IBM. (2024a). *Was ist der „k-nearest neighbors algorithm"?* https://www.ibm.com/de-de/topics/knn

IBM. (2024b). *Was ist Random Forest?* https://www.ibm.com/de-de/topics/random-forest

IBM. (2024c). *What is data quality?* https://www.ibm.com/topics/data-quality

Jadhav et al. (2019). *Comparison of Performance of Data Imputation Methods for Numeric Dataset*. https://doi.org/10.1080/08839514.2019.1637138

Jäger et al. (2021). *A Benchmark for Data Imputation Methods*. https://doi.org/10.3389/fdata.2021.693674

Jan vom Brocke. (2020). *Introduction to Design Science Research*. https://link.springer.com/chapter/10.1007/978-3-030-46781-4_1

Jason Poulos, R. (2018). *Missing Data Imputation for Supervised Learning*. https://www.tandfonline.com/doi/full/10.1080/08839514.2018.1448143

Kang H. (2013). *The prevention and handling of the missing data*. https://doi.org/10.4097/kjae.2013.64.5.402

Schelter et al. (2021). *JENGA - A Framework to Study the Impact of Data Errors on the Predictions of Machine Learning Models*. https://www.amazon.science/publications/jenga-a-framework-to-study-the-impact-of-data-errors-on-the-predictions-of-machine-learning-models

scikit-learn. (2024a). *GridSearchCV*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

scikit-learn. (2024b). *OneHotEncoder*. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html

scikit-learn. (2024c). *SGDClassifier*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDClassifier

scikit-learn. (2024d). *SGDRegressor*. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor

scikit-learn. (2024e). *StandardScaler*. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Velimirovic A. (2022). *What is Data Corruption and Can You Prevent It?* https://phoenixnap.com/blog/data-corruption

Woznica, K., B. (2020). *Does Imputation Matter? Benchmark for Predictive Models*.

Zhang, H., X., Xie, P. (2018). *Missing Value Imputation Based on Deep Generative Models*. https://www.tandfonline.com/doi/full/10.1080/08839514.2018.1448143