Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit

Maximierung der Modell-Diversität in Modell-Ensembles für lokal faire Klassifikationen

Ole Nies

Studiengang:	Data Science
Prüfer/in:	Prof. Dr. rer. nat. Melanie Herschel
Betreuer/in:	Nico Lässig, M.Sc.
Beginn am:	4. November 2022
Beendet am:	4. Mai 2023

Kurzfassung

Algorithmen des maschinellen Lernens haben in den vergangenen Jahren zunehmend an Bedeutung gewonnen, doch ihr Einsatz in sensiblen Bereichen, wie Medizin oder Strafverfolgung, birgt die Gefahr, bestehende Vorurteile zu verstärken. Das faire maschinelle Lernen zielt darauf ab, Methoden zu entwickeln, um Ungerechtigkeiten in algorithmischen Vorhersagen zu erkennen und abzuschwächen. Diese Bachelorarbeit befasst sich mit der Entwicklung eines automatisierten Verfahrens, um die Diversität von Modell-Ensembles zu maximieren und die Auswirkungen der Diversität auf die Fairness der Klassifikationen zu untersuchen. Dabei werden die Quantifizierung der Diversität, die automatisierte Generierung von diversen Modell-Ensembles und die Auswirkungen von Diversität auf Fairness untersucht.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

In den vergangenen Jahrzehnten sind Algorithmen des maschinellen Lernens zu einem festen Bestandteil unseres täglichen Lebens geworden. Die Anwendungen reichen von Systemen für personalisierte Empfehlungen und medizinischen Diagnosen, hin zu Kreditwürdigkeitsprüfungen und Verbrechensprognosen. Da diese Algorithmen immer komplexer und einflussreicher werden, muss sichergestellt werden, dass sie bestehende Vorurteile und Ungleichheiten aus unserer Gesellschaft nicht übernehmen oder gar verschärfen. Der Bereich des fairen maschinellen Lernens hat sich zu einem wichtigen Forschungsbereich entwickelt, der darauf abzielt, Methoden zu entwickeln, die Ungerechtigkeiten in algorithmischen Vorhersagen erkennen, quantifizieren und abschwächen [MMS+21].

Die Wichtigkeit von Fairness in maschinellem Lernen verdeutlicht folgendes bekanntes Beispiel für vorausschauende Polizeiarbeit aus der Praxis. Die PredPol-Software ermöglicht es, der Polizei Gebiete zu ermitteln, welche eine hohe Wahrscheinlichkeit auf kriminelle Aktivitäten haben. Eine Studie von Lum und Isaac [LI16] zeigte jedoch, dass aufgrund der verwendeten Trainingsdaten, die zu der Erstellung solcher Modelle verwendet werden, der Algorithmus bestimmte Bevölkerungsgruppen unverhältnismäßig stark als kriminell kennzeichnete. Dadurch wurden bestehende Ungleichheiten aufrechterhalten und verstärkt.

Ein weiteres Beispiel zur Verdeutlichung kommt aus dem Gesundheitssektor, in dem maschinelles Lernen zunehmend dazu verwendet wird, klinische Entscheidungen über Patienten zu treffen. Jüngste Studien zeigen, dass bestimmte weitverbreitete Algorithmen rassistische Vorurteile aufweisen, was zu einem ungleichen Zugang zu medizinischer Versorgung für Patienten bestimmter Ethnien führt. So ergab eine Studie von Obermeyer et al. [OPVM19], dass diese Algorithmen schwarzen Patienten unverhältnismäßig niedrige Risikowerte zuweisen, was zu einem eingeschränkten Zugang wichtiger Gesundheitsprogramme führt.

Es ist nicht ausreichend während des Entwurfes und Einsatzes solcher Modelle die sensitiven Attribute wie Geschlecht oder Ethnie nicht zu erfassen, da andere nicht-sensitive Attribute oft eine Korrelation mit sensitiven Attributen aufweisen und somit eine indirekte Diskriminierung vorhanden bleibt. Die Bekämpfung solcher Ungleichheiten ist sowohl aus ethischer Sicht als auch für die Akzeptanz der praktischen Notwendigkeit solcher Algorithmen wichtig. Mit dem wachsendem Interesse an diesem Bereich wurden verschiedene Verfahren entwickelt, mit denen Fairness in maschinellem Lernen gemessen und gewährleistet werden kann. Das FALCC-Framework [LH23] ist eines der jüngeren Ansätze und ermöglicht lokal faire Klassifikationen, mithilfe von diversifizierten Modell-Ensembles. Das dazugehörige Paper validierte, dass eine höhere Modell-Diversität in dem Framework zu faireren Klassifikationen führt. Diese Bachelorarbeit befasst sich mit der Entwicklung eines automatisierten Verfahrens, um die Diversität von Modell-Ensembles zu maximieren und die Auswirkungen der Diversität auf die Fairness der Klassifikationen genauer zu untersuchen.

1.1 Ziele dieser Arbeit

Das übergeordnete Ziel dieser Bachelorarbeit ist es, eine Lösung zu entwickeln, die die Diversität in Modell-Ensembles maximiert und die Auswirkungen der Diversität eines Ensembles auf lokal faire Klassifikationen zu untersuchen. Der Fokus liegt hierbei auf das Behandeln der folgenden drei Fragen:

1. Wie kann die Diversität von Modell-Ensembles quantifiziert und gemessen werden?
2. Wie können Modell-Ensembles automatisiert generiert werden, welche auf maximale Diversität getrimmt sind?
3. Welche Auswirkungen hat die Diversität von Modell-Ensembles auf die lokale Fairness von Klassifikationen?

Der weitere Verlauf dieser Arbeit ist wie folgt. Kapitel 2 befasst sich mit den nötigen Grundlagen für diese Arbeit und verschafft dabei einen Überblick über verwandte Arbeiten. Kapitel 3 stellt den Ansatz zur Maximierung der Modell-Diversität vor und Kapitel 4 stellt die Resultate vor und bewertet diese. Letztlich wird in Kapitel 5 der Verlauf der Arbeit zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Grundlagen und verwandte Arbeiten

Dieses Kapitel beinhaltet die Grundlagen für diese Arbeit, sowie einen Überblick über einschlägige Arbeiten. Dadurch wird die Ausgangssituation vor dieser Arbeit beschrieben und die Bedeutung der Fachbegriffe erklärt. Abschnitt 2.1 wird auf die Grundlagen von maschinellem Lernen, Modell-Ensembles und Ensemble-Learning-Prozesse erläutert. In Abschnitt 2.2 wird die Bedeutung von Diversität für Modell-Ensembles erläutert und anschließend einige verbreitete Metriken definiert. Abschnitt 2.3 beinhaltet die Grundlagen zu Hyperparametern in Modell-Ensembles, sowie zu Hyperparameter-Optimierung und zugehörige Verfahren. Anschließend wird in Abschnitt 2.4 die Begriffe der Fairness in maschinellem Lernen und Methoden zu deren Gewährleistung beschrieben. Die Fundamente des FALCC-Frameworks, welches für die Evaluation der lokalen Fairness in dieser Arbeit verwendet wurde, wird in Abschnitt 2.5 behandelt. Das Kapitel schließt mit einer Abgrenzung dieser Arbeit von den verwandten Arbeiten in Abschnitt 2.6 ab.

2.1 Modell-Ensembles

Für die Klassifizierung von Einträgen in Datensätzen kommen in der Regel Algorithmen des *maschinellen Lernens* zum Einsatz. Jeder Eintrag eines Datensatzes besteht dabei aus einer Anzahl von *Attributen* inklusive einem *Label*. Attribute und das Label sind Eigenschaften eines Eintrags, wobei das Label die zu klassifizierende Eigenschaft des Eintrages ist. Liegt etwa ein Datensatz vor, der Personen beschreibt und deren Kreditwürdigkeit klassifiziert, dann können die Attribute Eigenschaften wie *Alter, Geschlecht, Gehalt usw.* sein, mit dem Label *Kreditwürdigkeit*, die den Einträgen von Personen eine Kreditwertigkeitsstufe zuordnet.

Maschinelles Lernen ist ein Fachgebiet, welches sich mit dem Entwickeln, von Methoden beschäftigt, die anhand von Daten künstliches Wissen generieren. In dieser Arbeit wird sich ausschließlich mit der binären Klassifikation von Daten befasst. Algorithmen des maschinellen Lernens trainieren anhand von bereits manuell klassifizierten Daten, auch *Trainingsdaten* genannt, ein *Modell*, das dann schließlich neue, unklassifizierte Einträge automatisch klassifizieren kann. Mit weiteren, bereits klassifizierten Daten lässt sich die Genauigkeit der Klassifikationen eines trainierten Modells dann überprüfen. Diese Daten werden als *Testdaten* bezeichnet.

Einzelne trainierte Modelle können je nach Algorithmus verschiedenen Kategorien zugeordnet werden. Die häufigsten sind Entscheidungsbäume [Qui86] und Regressionsmodelle [FSC+19]. Ein trainiertes Modell kann nur Daten klassifizieren, die die gleiche Struktur wie die Trainingsdaten besitzen. Das heißt neue zu klassifizierende Daten müssen immer dieselben Attribute aufweisen wie die Trainingsdaten.

Ein *Modell-Ensemble* ist eine Zusammensetzung von einzelnen Modellen. Die unterliegenden Modelle werden auch als Basismodelle bezeichnet. Die Anwendung von Modell-Ensembles ist weitverbreitet und unterliegt bis heute aktiver Forschung. In der Regel ist ein Modell-Ensemble

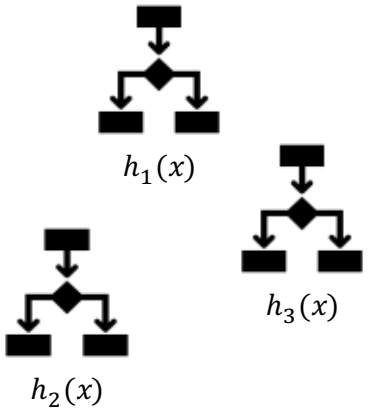


Abbildung 2.1: Ensemble E bestehend aus drei Entscheidungsbäumen h_1, h_2, h_3 als Basismodelle

gegenüber Einzelmodellen vorzuziehen, da eine Gruppe von unabhängigen Modellen, beispielsweise über einen Mehrheitsentscheid, insbesondere akkuratere Klassifikationen treffen kann. Voraussetzung dafür ist, dass die Basismodelle akkurat und unabhängige Klassifikationen untereinander treffen [HS90]. Das heißt, die Genauigkeit der Basismodelle muss in mindestens 50% der Fälle neue Einträge korrekt klassifizieren und die Klassifikationen der Basismodelle sollen nicht untereinander korrelieren. Dadurch werden Fehlentscheidungen von Basismodellen an jeweils unterschiedlichen Einträgen im Datensatz getroffen. Für ein besseres Verständnis, warum Modell-Ensembles genauere Klassifikationen treffen, als deren Basismodelle, stelle man sich ein Ensemble aus drei Entscheidungsbäumen aus Abbildung 2.1 vor. Jeder Entscheidungsbaum habe eine Fehlerrate von 30% in deren Klassifikationen. Wenn davon ausgegangen wird, dass diese Entscheidungsbäume gleich sind, also nicht unabhängig, dann klassifizieren sie jeden neuen Eintrag gleich und das Ensemble hat ebenfalls eine Fehlerrate von 30%. Sind diese Basismodelle aber divers und deren Klassifikationen untereinander unabhängig, dann können beispielsweise h_2 und h_3 für einen neuen Eintrag eine korrekte Klassifikation treffen, auch wenn h_1 eine falsche trifft. Dadurch wird über den Mehrheitsentscheid des Ensembles der Eintrag letztlich korrekt klassifiziert. Allgemein gesagt, trifft das Modell-Ensemble durch Mehrheitsentscheid genau dann falsche Klassifikationen, wenn mindestens zwei Basismodelle falsch liegen. Damit liegt die Fehlerrate dieses Modell-Ensembles mit unabhängigen Fehlerraten von 30% der Basismodelle bei $\binom{3}{2}(0.3^2 \cdot 0.7) + 0.3^3 = 0.216$, welche geringer ist, als die der Basismodelle.

Bei der Generierung von Modell-Ensembles besteht die Herausforderung darin, unabhängige Basismodelle mit nur einem Trainingsdatensatz zu trainieren. Dieser Prozess wird auch als *Ensemble-Learning* [DYC+19; SR18] bezeichnet und es gibt zwei grundlegende Ansätze zu der Generierung von Modell-Ensembles. Bezeichnet werden diese als *homogenes* Ensemble-Learning und *heterogenes* Ensemble-Learning, welche in den folgenden Abschnitten behandelt werden.

2.1.1 Homogenes Ensemble-Learning

Homogenes Ensemble-Learning zeichnet sich dadurch aus, dass die Basismodelle alle von gleicher Art sind. Darunter fällt auch das Ensemble E aus Abbildung 2.1, da es ausschließlich aus Entscheidungsbäumen besteht. Um eine Diversität und Unabhängigkeit trotz derselben Basismodelle untereinander anzuregen, greift man bei dem Training dieser auf folgende Methoden zurück: *Bagging* und *Boosting*.

Bagging

Bagging steht für *Bootstrap aggregating* und verfolgt den Ansatz, die Basismodelle auf jeweils unterschiedlichen Teilmengen des Trainingsdatensatzes zu trainieren [Bre96]. Somit schneiden die Basismodelle auch jeweils auf unterschiedlichen Teilen des Trainingsdatensatzes besser und schlechter ab, wodurch eine Unabhängigkeit der Klassifikationen untereinander gefördert wird. Ein weitverbreitetes Ensemble-Learning-Verfahren, welches auf Bagging basiert, sind Random-Forests [Bre01]. Hierbei werden mehrere unabhängige Entscheidungsbäume mit Bagging und Randomisierung generiert. Die Klassifizierung eines Eintrags erfolgt dann über den Mehrheitsentscheid aller Bäume.

Boosting

Bei Boosting hingegen wird jedes Basismodell auf dem gesamten Datensatz trainiert [Sch90]. Damit die Basismodelle unterschiedliche Klassifikationen treffen, trainiert diese Methode jedes Basismodell sequenziell nacheinander. Nach jeder Iteration werden die Gewichte der Einträge so angepasst, dass sie in der nächsten korrekt klassifiziert werden können. Adaboost ist ein weitverbreiteter Boosting-Algorithmus, welcher diese Methode umsetzt [FS97].

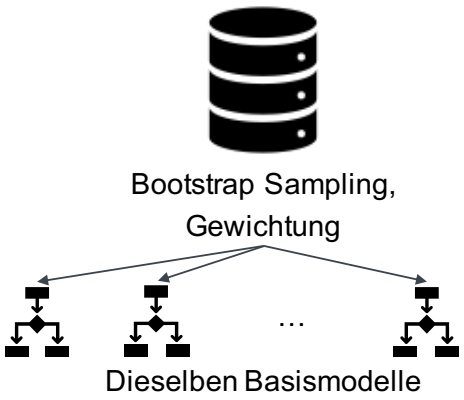


Abbildung 2.2: Homogener Ensemble-Learning-Prozess

Der homogene Ensemble-Learning-Prozess wird in Abbildung 2.2 nochmals visuell dargestellt. Zu sehen ist, wie eine Menge von denselben Basismodellen aus einem Datensatz über Bootstrap aggregating und Gewichtung entstehen.

2.1.2 Heterogenes Ensemble-Learning

Mit heterogenem Ensemble-Learning bezeichnet man die Art von Modell-Ensembles, welche durch Zufallspaarung aus einem Pool von bereits vortrainierten Basismodellen entstehen. Dabei können die Basismodelle des Pools aus unterschiedlichen Verfahren bestehen. In Abbildung 2.3

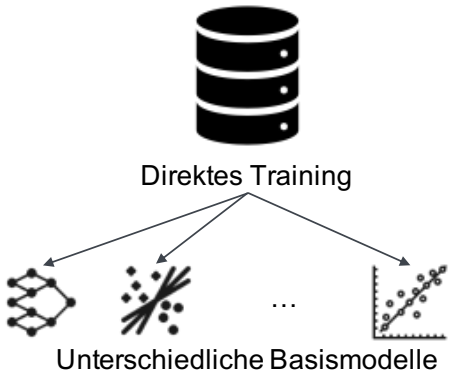


Abbildung 2.3: Heterogener Ensemble-Learning-Prozess

ist der heterogene Ensemble-Learning-Prozess visuell dargestellt. Beispielsweise kann ein Pool einen Entscheidungsbaum, ein neurales Netz, und eine Support-Vektor-Maschine enthalten. Die Herausforderung bei heterogenem Ensemble-Learning besteht darin, die Paarung von Basismodellen zu finden, welche die qualitativ besten Entscheidungen trifft.

2.2 Diversität von Modell-Ensembles

Die Diversität von Modell-Ensembles bezieht sich darauf, wie unterschiedlich die Klassifikationen der Basismodelle untereinander ausfallen. Dabei bedeutet Diversität in Modell-Ensembles nicht gleich unabhängig, da die Diversität sich lediglich auf die Vielfalt der Klassifikationen bezieht. Um dies zu verdeutlichen, stelle man sich ein Modell-Ensemble bestehend aus zwei Basismodellen h_1 und h_2 vor. Wenn h_1 immer das Gegenteil wie h_2 klassifiziert, dann liegen offenbar ausschließlich unterschiedliche Klassifikationen vor und das Ensemble hat maximale Diversität. Da die Klassifikationen von h_1 und h_2 abhängig sind, liegt keine Unabhängigkeit vor. Die Basismodelle haben erst vollständige Unabhängigkeit untereinander, falls die Klassifikationen keine Korrelation aufweisen, dies wäre der Fall, wenn die Basismodelle alle zufällige Klassifikationen untereinander treffen.

Vorausgehende Forschungen zeigen, dass die Diversität der Basismodelle eine große Rolle darin spielt, wie die Qualität der Klassifikationen des Ensembles ausfällt [Rok09]. Während die in Abschnitt 2.1 beschriebenen Verfahren bereits alle intuitiv durch deren beschriebenen Ansätze die Basismodelle diversifizieren, findet jedoch keine explizite Quantifizierung der Diversität statt. Zudem gibt es keine allgemein angesehene Metrik für die Messung der Diversität in Modell-Ensembles. Der Bericht von Kuncheva et al. [KW03] hat einige Diversitätsmetriken aus der Statistik zusammengetragen und untersucht, die sich für die Quantifizierung der Diversität von Modell-Ensembles

eignen. Im Weiteren wird sich auf die dort aufgelisteten Metriken bezogen. Es wird zwischen zwei Kategorien von Diversitätsmetriken unterschieden. Paarweise Metriken und nicht-paarweise Metriken.

2.2.1 Paarweise Diversitätsmetriken

Die paarweisen Diversitätsmetriken messen die Diversität zwischen zwei trainierten Modellen. Der Ansatz hierfür ist, die Häufigkeit von korrekten und falschen Klassifizierungen eines Testdatensatzes der Modelle mittels Kontingenztafel Tabelle 2.1 zu zählen. Hierbei sind h_i und h_j zwei trainierte

	h_j korrekt (1)	h_j falsch (0)
h_i korrekt (1)	N^{11}	N^{10}
h_i falsch (0)	N^{01}	N^{00}

Tabelle 2.1: Häufigkeit der korrekten und falschen Klassifizierungen zweier Modelle untereinander.

Modelle. N^{11} die Anzahl an Klassifizierungen, bei denen beide Modelle den gleichen Eintrag korrekt klassifizierten, N^{10} beziehungsweise N^{01} die Anzahl, in der nur h_i beziehungsweise h_j korrekt klassifizierten und N^{00} die Anzahl der Einträge, bei denen beide Modelle falsch klassifizierten. Somit ist $N = N^{11} + N^{10} + N^{01} + N^{00}$ die Größe des Testdatensatzes. Mit diesen Häufigkeiten lassen sich nun verschiedene paarweise Diversitätsmetriken für zwei Modelle h_i und h_j berechnen. Im Folgenden werden zwei verbreitete paarweise Diversitätsmetriken definiert.

Definition 2.2.1 (Das Maß der Uneinigkeit [Ska96])

$$Dis_{i,j} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{2.1}$$

Das Maß der Uneinigkeit ist die Relation zwischen der Anzahl aller Einträge, bei denen die Modelle jeweils unterschiedlich klassifiziert haben, zwischen der Anzahl aller Einträge des Datensatzes. Die Größe der Metrik befindet sich immer in einem Intervall von 0 bis 1. Je größer das Ergebnis, desto diverser sind die beiden untersuchten Modelle und desto unterschiedlicher fallen die Klassifikationen aus. Das heißt ist das Maß gleich 1, dann haben die Modelle ausschließlich unterschiedliche Klassifikationen getroffen und ist das Maß 0, dann waren sich die Modelle bei jeder Klassifikation einig.

Definition 2.2.2 (Die Q-Statistik [Yul00])

$$Q_{i,j} = \frac{N^{11} \cdot N^{00} - N^{01} \cdot N^{10}}{N^{11} \cdot N^{00} + N^{01} \cdot N^{10}} \tag{2.2}$$

Die Q-Statistik multipliziert jeweils die Anzahl der Einträge, bei denen beide Modelle korrekt klassifizierten mit der Anzahl der Einträge, bei denen beide Modelle falsch klassifizierten und subtrahiert im Zähler, beziehungsweise addiert im Nenner das Produkt aus den Anzahlen der unstimmig klassifizierten Einträge. Dieses Maß befindet sich immer im Intervall von -1 und 1. Je kleiner das Ergebnis, desto größer ist die Diversität der Modelle und je größer das Ergebnis,

desto stimmiger sind sich die Modelle. Hier bedeutet ein Ergebnis von -1, dass beide Modelle ausschließlich unterschiedliche Klassifikationen getroffen haben und 1 bedeutet, dass beide Modelle bei allen Einträgen jeweils dieselbe Klassifikation trafen. Das Besondere an dieser Metrik ist zudem, dass ein Erwartungswert von 0 vorliegt, wenn beide Modelle untereinander unabhängig sind. Daher ähnelt diese Metrik einem Korrelationskoeffizienten zwischen beiden Modellen, ist jedoch einfacher zu berechnen und wird daher auch bevorzugt.

Ein Problem von paarweisen Metriken ist, dass diese nicht direkt auf Modell-Ensembles mit mehr als zwei Basismodellen angewandt werden kann. Eine Möglichkeit dieses Problem zu umgehen, ist eine Diversitätsmatrix mit paarweisen Metriken aller Paarungskombinationen der Basismodelle zu füllen und anschließend den Durchschnitt dieser Werte zu bestimmen. Beispielsweise berechnet sich die durchschnittliche Q-Statistik für ein Modell-Ensemble mit L Basismodellen wie folgt.

Definition 2.2.3 (Die durchschnittliche Q-Statistik für $L > 2$ Basismodelle [KW03])

$$Q_{ges} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{j=i+1}^{L} Q_{i,j} \tag{2.3}$$

Analog lässt sich wie in Definition 2.2.3 der Durchschnitt auch für das Maß der Uneinigkeit aus Definition 2.2.1 berechnen.

2.2.2 Nicht-Paarweise Diversitätsmetriken

Die nicht-paarweisen Diversitätsmetriken sind explizit für Modell-Ensembles der Größe $L > 2$ bestimmt. Im Folgenden wird sich auf binäre Klassifikationen (0 und 1) beschränkt. Anstatt zu zählen, wie Basismodelle die jeweiligen Einträge eines Datensatzes klassifizieren, ist der Ansatz der nicht-paarweisen Diversitätsmetriken, die Anzahl der Basismodelle zu zählen, die einen Eintrag e_k gleich klassifizieren. Somit gilt, falls $\lfloor L/2 \rfloor$ Basismodelle e_k mit demselben Wert (0 oder 1) klassifizieren und der Rest $L - \lfloor L/2 \rfloor$ der Basismodelle mit dem anderen Wert, dann liegt eine maximale Diversität der Basismodelle für Eintrag e_k vor. Andererseits, falls alle, also L Basismodelle Eintrag e_k gleich klassifizieren, liegt keine Diversität vor.

Eine Weise die Diversität mit diesem Ansatz zu berechnen zeigt folgende Gleichung.

Definition 2.2.4 (Das Entropiemaß E [CC00])

$$E = \frac{1}{N} \sum_{k=1}^{N} \frac{1}{(L - \lceil L/2 \rceil)} \min\{l(e_k), L - l(e_k)\} \tag{2.4}$$

Dabei sind e_k die Einträge eines (Test-)Datensatzes, N die Größe des Datensatzes und $l(e_k)$ die Anzahl an Basismodellen des untersuchten Modell-Ensembles, welche e_k korrekt klassifizieren. Das Entropiemaß berechnet einen normierten Durchschnitt des in Abschnitt 2.2.2 beschriebenen Ansatzes über alle Einträge eines Datensatzes. So bewegt sich das Entropiemaß in einem Intervall von 0 bis 1, mit Größe 0 als Beleg dafür, dass alle Basismodelle gleich sind und Größe 1 als größtmögliche Diversität der Basismodelle gilt.

2.3 Hyperparameter-Optimierung

Hyperparameter-Optimierung ist die automatisierte Suche nach der optimalen Modellarchitektur über Tuning der Hyperparameter des Modells [EMS19]. Dieser Abschnitt erläutert, die Grundlagen von Hyperparametern in Modellen des maschinellen Lernens und stellt die wichtigsten Methoden zur Optimierung dieser vor.

2.3.1 Hyperparameter eines Modells

Ein Modell (z. B. Entscheidungsbaum) besteht aus zwei Klassen von Parametern. Die Parameter, die während der Trainingsphase von dem Algorithmus initialisiert und aktualisiert werden, auch *Modellparameter* genannt und die Parameter, welche vor dem Training dem Algorithmus zur Modellgenerierung manuell übergeben werden müssen, die *Hyperparameter* [YS20]. Die Modellparameter eines Entscheidungsbaumes sind beispielsweise nach dem Algorithmus aus [Qui86] die besten Attribute, nach denen pro Knoten im Baum gesplittet wird, da diese automatisiert während des Trainings ermittelt werden. Hyperparameter sind hier etwa die maximale Baumtiefe als eine Abbruchbedingung, die dem Algorithmus im Vorhinein übergeben werden.

Die Hyperparameter eines Modell-Ensembles umschließt die Hyperparameter des Modell-Ensemble-Verfahrens sowie die jeweiligen Hyperparameter der unterliegenden Basismodelle. Beispielsweise bei Adaboost sind das insbesondere die Anzahl der Iterationen und dazu alle Hyperparameter des Basismodells, wie die eines Entscheidungsbaums. Der Suchraum der Optimierung setzt sich dann aus allen Kombinationen von Hyperparameterwerten des Modell-Ensembles-Verfahrens zuzüglich aller Kombinationen von Werten des Basismodells zusammen. Dabei wird sich aufgrund der Komplexität darauf beschränkt, dass die Basismodelle bei homogenem Ensemble-Learning untereinander immer dieselben Hyperparameterwerte pro Kombination erhalten.

Hyperparameter haben einen großen Einfluss auf die Struktur der unterliegenden Modelle und daher auch auf die Qualität der Klassifikationen, die das Modell trifft. Die Qualität wird durch eine sogenannte *Score-Funktion* quantifiziert, insbesondere kann diese Funktion die Genauigkeit sein. Die Score-Funktion ist die zu optimierende Funktion in diesem Prozess, e.g. es wird nach den Hyperparametern gesucht, die zu maximaler Genauigkeit des Modells führen. Daher ist die Hyperparameter-Optimierung ein essenzieller Bestandteil in der Konstruktion von Modellen, insbesondere für diese, die Entscheidungsbäume mit vielen Hyperparametern beinhalten [YS20]. Da die Hyperparameter sich nicht aus dem Datensatz ableiten lassen, ist es notwendig, mehrere Trainings des Modells mit unterschiedlichen Hyperparametern durchzuführen, um die Score-Funktion zu evaluieren. Es stellt sich die Frage, wie man ohne großen Aufwand zu einer möglichst optimalen Konfiguration der Hyperparameter gelangt, um die Score-Funktion zu optimieren. Dieses Problem bezeichnet man als Hyperparameter-Optimierung und wird im folgenden Kapitel weiter erläutert.

2.3.2 Verfahren zur Optimierung der Hyperparameter

Die geeigneten Verfahren zur Hyperparameter-Optimierung unterscheiden sich abhängig vom Algorithmus des maschinellen Lernens. Grund dafür sind die unterschiedlichen Typen von Hyperparametern, die jeweils auftreten können, wie kategorische, diskrete und kontinuierliche Hyper-

parameter [DCGC19]. Manuelles Tuning dieser Parameter ist in den meisten Fällen ineffektiv, da meist eine große Menge an unterschiedlichen Hyperparametern vorliegen kann und die einzelnen Evaluationen zeitaufwendig sind.

Der grundlegende Ablauf der Hyperparameter-Optimierung für einen Algorithmus des maschinellen Lernens besteht aus folgenden Schritten:

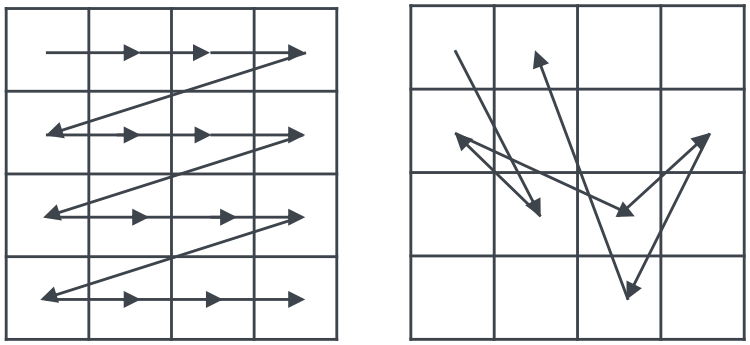
1. Wähle eine passende Score-Funktion, die maximiert werden soll.
2. Wähle die Hyperparameter, die optimiert werden sollen und bestimme deren Typ (i.e. kategorisch, diskret oder kontinuierlich) und wähle die Optimierungstechnik.
3. Trainiere ein Modell mit Standard-Hyperparametern als Ausgangs-Modell.
4. Beginne den Optimierungsprozess mit einem großen Suchraum aus Hyperparametern, der auf den Ergebnissen kleinerer Tests oder Vorwissen basiert.
5. Grenze den Suchraum schrittweise ein, auf Grundlage von getesteten Suchregionen oder Hyperparameter, welche besonders großen oder kleinen Einfluss auf die Score-Funktion haben.
6. Gib die besten Hyperparameter, die zu optimalen Werten der Score-Funktion geführt haben, zurück.

Im Folgenden werden drei verbreitete Optimierungsverfahren für Hyperparameter vorgestellt.

Grid-Search und Random-Search

Bei *Grid-Search* wird ein gegebener Suchraum vollständig abgesucht und zu jeder möglichen Hyperparameterkombination die Score-Funktion evaluiert und schließlich werden die Hyperparameter zurückgegeben, die zum besten Wert führen (z. B. höchste Genauigkeit) [BBBK11]. Dabei wird im Voraus ein Suchraum definiert, der für alle Hyperparameter die Werte definiert, die in die Suche eingeschlossen werden sollen. Hierfür benötigt es ausschließlich kategorische oder diskrete Hyperparameterwerte, sodass der Suchraum endlich bleibt. Beispielweise kann ein Suchraum für die Hyperparameter *Anzahl der Bäume*, *maximale Baumtiefe* und *Split-Kriterium* von Random-Forest folgendermaßen aussehen. *Anzahl der Bäume* wird auf das Intervall $[3, 50]$ beschränkt, die *maximale Baumtiefe* auf [1,10] und das *Split-Kriterium* auf die Werte innerhalb $\{Gini, Entropie\}$. Die Größe des daraus resultierenden Suchraums ergibt sich aus allen möglichen Kombinationen, die sich aus diesen vorgegebenen Werten bilden lassen. In diesem Beispiel besteht der Suchraum aus $48 \cdot 10 \cdot 2 = 960$ möglichen Hyperparameterkombinationen. Grid-Search trainiert mit jeder dieser Kombinationen nun ein Modell und evaluiert die Score-Funktion.

Vorteil dieses Optimierungsverfahren ist, dass es das bestmögliche Optimum dieses Suchraums garantiert findet. Zudem kann der Suchprozess parallelisiert werden. Da der Suchraum bei vielen möglichen Hyperparametern mit entsprechendem Wertebereich schnell sehr groß wird, ist Grid-Search in einigen Anwendungsfällen ineffizient. *Random-Search* kommt dem entgegen und sucht sich eine vorgegebene Anzahl an zufällig ausgesuchten Hyperparameterkombinationen aus dem Suchraum aus, um die Suche effizienter, aber dafür weniger effektiver zu gestalten [BB12].



(a) Grid-Search **(b)** Random-Search

Abbildung 2.4: Suchverlauf von Grid-Search und Random-Search im Vergleich

In Abbildung 2.4 ist der Suchverlauf von Grid- und Random-Search schematisiert. Gegeben ist ein zweidimensionaler Suchraum mit jeder Zelle als eine Hyperparameterkombination. Grid-Search testet den gesamten Suchraum, während Random-Search nur eine begrenzte, zufällige Anzahl von Kombinationen aus den Hyperparametern testet.

Bayesian-Optimization

Bayesian-Optimization für Hyperparameter [EFH+13] bestimmt, im Gegensatz zu Grid-Search und Random-Search, die nächste Hyperparameterkombination anhand von vorherigen Ergebnissen der Score-Funktion, mit deren Hyperparametern. Dabei wird bei jeder Iteration eine Abwägung zwischen Erkunden von neuen Kombinationen und der Ausnutzung von bereits gefundenen Kombinationen mit guten Score-Funktionswerten vorgenommen. Die Tendenz der Abwägung, sowie die Anzahl an Iterationen, wird im voraus manuell festgelegt. Damit ist dieses Verfahren eine intelligentere Suche und es können viele überflüssige Kombinationen ausgelassen werden, um das Optimum zu erreichen. Hier sollte der Suchraum nicht aus nominal kategorischen Werten bestehen, da Bayesian-Optimization diese nicht interpretieren kann.

2.4 Fairness in maschinellem Lernen

Der Begriff der Fairness in dem Bereich von maschinellem Lernen bezieht sich darauf, dass trainierte Modelle auf personenbezogene Daten keine Vorurteile oder Diskriminierungen aufgrund von persönlichen Merkmalen wie Geschlecht oder ethnischer Zugehörigkeit in ihren Klassifikationen enthalten sollen. Solche sensitiven Merkmale werden auch als geschützte Attribute bezeichnet. Dieser Abschnitt stellt verbreitete Methoden zu der Quantifizierung der Fairness von Klassifikationen vor und gibt anschließend einen Überblick über die wichtigsten Ansätze, welche faire Klassifikationen im Bereich maschinelles Lernen gewährleisten sollen.

2.4.1 Quantifizierung von Fairness

Es gibt verschiedene Ansätze, um die Fairness von trainierten Modelle und deren Klassifikationen zu bewerten. Diese lassen sich in Gruppenfairness und individuelle Fairness kategorisieren. Bei dem Begriff der Gruppenfairness wird diese über den gesamten Datensatz gemessen. Diese wird hier daher auch als globale (Gruppen-)Fairness bezeichnet. In jüngeren Arbeiten wird die Messung auch auf Untermengen des Datensatzes durchgeführt [LOH21]. Diese wird hier dann als lokale Fairness bezeichnet.

In den folgenden Abschnitten wird genauer auf die einzelnen Kategorien eingegangen.

Globale Gruppenfairness

Die globale Gruppenfairness beschreibt, wie ähnlich die Klassifikationen in jeweiligen Gruppen im Datensatz untereinander sind. Das heißt, es soll sichergestellt werden, dass der Durchschnitt von positiven Klassifikationen in allen Gruppen im Datensatz möglichst gleich ist. Die Gruppen werden im Vorhinein anhand der gegebenen geschützten Attribute gebildet. Zum Beispiel lässt sich ein Datensatz mit Geschlechterattribut in die Gruppen Männer, Frauen und Divers unterteilen. Um die globale Fairness zu messen, gibt es einige verbreitete Metriken, die im Folgenden kurz erläutert werden.

Die wörtlichen Formulierungen wurden aus [MMS+21] abgeleitet.

Definition 2.4.1 (Demographic Parity [DHP+12])
Ein Modell $\hat{Y}$, das binäre Klassifikationen (e.g. 0 und 1) trifft, erfüllt demographic parity in Bezug auf das geschützte Attribut $A \in \{0, 1\}$, wenn

$$P(\hat{Y}|A = 0) = P(\hat{Y}|A = 1) \tag{2.5}$$

gilt.

Die Wahrscheinlichkeiten für positive und negative Klassifikationen sollen unterhalb der Gruppen gleich sein.

Definition 2.4.2 (Equalized Odds [HPPS16])
Ein Modell $\hat{Y}$, das binäre Klassifikationen trifft, erfüllt equalized odds in Bezug auf das geschützte Attribut $A \in \{0, 1\}$ und die echte Klasse $Y \in \{0, 1\}$, wenn

$$P(\hat{Y} = 1|A = 0, Y = y) = P(\hat{Y} = 1|A = 1, Y = y) \tag{2.6}$$

mit $y \in \{0, 1\}$ gilt.

Das Verhältnis der true positive und false positives Klassifikationen soll unterhalb der Gruppen gleich sein.

Definition 2.4.3 (Treatment Equality [BHJ+18])
Ein Modell $\hat{Y}$, das binäre Klassifikationen trifft, erfüllt equalized odds in Bezug auf das geschützte Attribut $A \in \{0, 1\}$ und die echte Klasse $Y \in \{0, 1\}$, wenn

$$P(\hat{Y} = 1|A = a, Y = 0) = P(\hat{Y} = 0|A = a, Y = 1) \tag{2.7}$$

mit $a \in \{0, 1\}$ gilt.

Das Verhältnis der false positive und false negative Klassifikationen soll unterhalb der Gruppen gleich sein.

Diese Fairness-Metriken beschreiben die Klassifikationen in Datensätzen und deren Fairness auf hoher Ebene ohne einen großen Rechenaufwand, da die Klassifikationen nur zusammen aggregiert werden, um die Wahrscheinlichkeiten zu berechnen. Allerdings werden in diesem Begriff der Fairness mögliche Ungleichheiten auf individueller Ebene, wie innerhalb der Gruppen nicht berücksichtigt. Der Begriff der individuellen Fairness hingegen soll genau diese quantifizieren.

Individuelle Fairness

Die individuelle Fairness bewertet, ob Einträge im Datensatz, mit ähnlichen Attributen gleich klassifiziert werden. Dabei muss im Vorhinein bestimmt werden, welche Einträge sich untereinander ähneln und diese danach in Regionen einteilen. Bekannte Metriken zur Quantifizierung der individuellen Fairness werden im folgenden kurz erläutert.

Definition 2.4.4 (Fairness Through Awareness [DHP+12])
Alle Einträge, die sich nach einer Ähnlichkeitsmetrik ähneln, sollen von dem Modell gleich klassifiziert werden.

Definition 2.4.5 (Fairness Through Unawareness [KLRS17])
Klassifikationen eines Modells sind fair, solange keine geschützten Attribute während des Klassifikationsprozesses berücksichtigt werden.

Definition 2.4.6 (Konsistenz [ZWS+13])
Die Konsistenz vergleicht die Klassifikationen $\hat{y} \in \{0, 1\}$ für Einträge x_n mit $n \in [0, N]$ eines Modells, zu deren k nächsten Nachbarn $kNN(x)$:

$$yNN = 1 - \frac{1}{Nk} \sum_{n=0}^{N-1} |\hat{y}_n - \sum_{j \in kNN(x_n)} \hat{y}_j| \tag{2.8}$$

Es gilt $yNN \in [0, 1]$ und je größer der Wert, desto fairer die individuellen Klassifikationen.

Für die spätere Implementierung wird für Konsistenz die Formel $1 - yNN$ verwendet, sodass wie bei den anderen Fairness-Metriken auch kleinere Werte höhere Fairness bedeuten.

Während dieser Begriff die Fairness auf tiefster Ebene betrachtet, kann es schwierig sein, die Ähnlichkeiten von Einträgen im Datensatz zu definieren. Außerdem kann die individuelle Fairness im Widerspruch zu der globalen Gruppenfairness stehen. Nach der individuellen Fairness ist es etwa fair, dass mit dem geschützten Attribut Geschlecht nur Männer positiv klassifiziert werden, da sich ihre übrigen Attribute ähneln und die übrigen Gruppen negativ klassifiziert werden. Dies steht im Widerspruch zu der globalen Gruppenfairness und damit können nicht immer beide Begriffe von Fairness gewährleistet werden. Um diese Fälle zu vermeiden, bildet der Begriff der lokalen Fairness einen Mittelweg.

Lokale Fairness

Der Begriff der lokalen Fairness ist eine Kreuzung der Begriffe von globaler und individueller Fairness und soll jeweils die grundlegenden Eigenschaften übernehmen. So wird bei lokaler Fairness der Begriff der globalen Gruppenfairness aus Abschnitt 2.4.1 innerhalb von Regionen im Datensatz, die aus jeweils ähnlichen Einträgen, angewendet [LOH21]. Somit können alle Metriken zur Quantifizierung der globalen Gruppenfairness, zur Bestimmung der lokalen Fairness verwendet werden, wenn der Datensatz in Regionen nach einer Ähnlichkeitsmetrik eingeteilt ist, wie bei individueller Fairness. Somit beschreibt die lokale Fairness auf einer Ebene zwischen globaler und individueller Fairness die Ungleichheiten im Datensatz.

2.4.2 Fairness in maschinellem Lernen gewährleisten

Zur Gewährleistung von fairen Klassifikationen wurden in den vergangenen Jahren einige Strategien für faire Klassifikationen entwickelt [MMS+21]. Diese setzen an unterschiedlichen Phasen des maschinellen Lernens an und werden dementsprechend unterteilt in *Pre-Processing-, In-Processing-* und *Post-Processing-Methoden* [dOL17].

Pre-Processing-Methoden

Die Pre-Processing-Methoden basieren auf der Modifizierung des Trainingsdatensatzes vor dem Trainieren des Modells [KC11]. Dabei sollen im Datensatz mögliche Vorurteile und Ungleichheiten entfernt werden, sodass die Modelle diese während des Trainings nicht übernehmen. Die Modifikationen beinhalten insbesondere Sampling-Strategien [HD13] und das Einführen einer Gewichtung der Einträge [ZVRG17], um unfaire Unterschiede zwischen Gruppen oder individuellen Personen zu minimieren.

In-Processing-Methoden

Bei In-Processing-Methoden werden Fairnessbedingungen während der Trainingsphase des Modells eingebunden. So wird während des Trainings das Modell für faire Klassifikationen belohnt, beispielsweise über sogenannte *Regularisierungsterme* [ZVRG17] oder mit *adversarial training* [ZLM18], sodass Diskriminierung minimiert wird.

Post-Processing-Methoden

Die Post-Processing-Methoden passen die Klassifikationen eines Modells nach dessen Training an, um Fairness zu versichern. Diese können unter anderem eine Änderung der Entscheidungsschwellen [FFM+15] oder eine Neukalibrierung der vorhergesagten Wahrscheinlichkeiten [HPPS16] enthalten, um die Klassifikationen zwischen verschiedenen Gruppen oder Personen auszugleichen.

2.5 Das FALCC-Framework

Das FALCC-Framework (*Fair and Accurate Local Classifications by leveraging Clusters*) und die unterliegenden Algorithmen ermöglichen lokal faire binäre Klassifikationen mit hoher Genauigkeit und Effizienz [LH23]. Dieses Verfahren lässt sich den In-Processing-Methoden zuordnen. Im Folgenden wird auf die Verfahrensweise eingegangen.

2.5.1 FALCC und die zugehörigen Algorithmen

Das Framework lässt sich in eine offline und in eine Online-Phase unterteilen. Die Offline-Phase umfasst unter anderem eine Aufbereitung des Datensatzes, sowie das Trainieren eines Modell-Ensembles. Die Offline-Phase muss somit nur einmal ausgeführt werden, während die Schritte der Online-Phase für jede neue Klassifikation erfolgen. Im Folgenden wird auf die wichtigsten Details dieser Phasen eingegangen.

Zu Beginn der Offline-Phase liegt ein Datensatz vor, welcher in Trainings- und Validierungs- und Testdaten unterteilt wird. Um die lokale Fairness zu Gewährleistung teilt ein Algorithmus die Validierungsdaten durch ein Clustering-Verfahren [XW05] in Regionen ein. Dabei werden die Gruppen anhand von Ähnlichkeit der nicht-geschützten Attribute gebildet, sodass sich die Einträge pro Gruppe untereinander ähnlich sind. Diese Regionen bilden die Untergruppen, anhand denen die lokale Fairness bestimmt wird. Außerdem wird anhand der Trainingsdaten ein Modell-Ensemble trainiert, dessen Basismodelle jeweils auf jeder gebildeten Region nach Fairness und Genauigkeit bewertet wird. Anschließend wird jeder Region eine Kombination von Basismodellen zugeordnet, welche den besten Mittelwert zwischen Fairness und Genauigkeit in den Klassifikationen für diese Region aufwiesen. Die Fairness-Metrik, nach der bewertet und nach der auch die optimalen Basismodelle pro Region gewählt werden soll, kann frei aus Abschnitt 2.4.1 gewählt werden.

Die Online-Phase besteht aus dem Klassifikationsvorgang neuer Einträge. Neue Einträge, die klassifiziert werden sollen, werden dann einer der Regionen mit der höchsten Ähnlichkeit zugeordnet und mit der dazugehörigen Basismodell-Kombination klassifiziert. Mit den Testdaten wird die Genauigkeit und die Fairness der Klassifikationen bestimmt.

2.5.2 AdaptedAdaboost

Die Arbeit [LH23] zeigt neben Gewährleistung von lokaler Fairness und hoher Genauigkeit dieses Verfahrens auch, dass je höher die Diversität des trainierten Modell-Ensembles ist, desto besser die Fairness der Klassifikationen. Um ein diverses Modell-Ensemble zu trainieren, nutzt dieses Verfahren eine abgewandelte Version des Adaboost-Algorithmus, auch *AdaptedAdaboost* genannt. AdaptedAdaboost funktioniert größtenteils wie das normale Adaboost-Verfahren, nur dass bei jeder Iteration während der Trainingsphase des Algorithmus, das Basismodell gewechselt wird. Das heißt, es liegt eine Menge an verschiedenen Modell-Arten, wie Entscheidungsbaum, Regressionsmodell oder Support-Vektor-Maschine, die bei jeder Iteration abwechselnd als Basismodell trainiert werden. Die Einbindung unterschiedlicher Arten von Basismodellen, soll auch die Klassifikationen dieser, diverser ausfallen lassen, ähnlich wie bei heterogenem Ensemble-Learning. Damit ist AdaptedAdaboost eine Kreuzung zwischen homogenem und heterogenem Ensemble-Learning und ermöglicht so eine hohe Modell-Diversität.

2.6 Abgrenzung dieser Arbeit

Dieser Abschnitt beschreibt, wie sich diese Arbeit von den existierenden abgrenzt.

Die Diversität von Modell-Ensembles wurde bisher nur zu der Verbesserung der Genauigkeit der Klassifikationen des Ensembles genutzt. In [LH23] wurde erstmals der Zusammenhang zwischen Diversität eines Ensembles und Fairness der Klassifikationen berücksichtigt. Während das Paper bereits einen Ansatz mit AdaptedAdaboost besitzt, um diverse Modell-Ensembles zu generieren, stellt diese Arbeit einen neuen Ansatz für Diversifizierung der Modell-Ensembles vor. Der Ansatz dieser Arbeit soll noch diversere Modelle als AdaptedAdaboost generieren und somit die Fairness der Klassifikationen durch das FALCC-Framework steigern.

Über Hyperparameter-Optimierung soll das diverseste Modell-Ensemble eines gegebenen Suchraums gefunden werden. Auch Hyperparameter-Optimierung wird in der Regel für die Verbesserung der Genauigkeit der Klassifikationen genutzt. In dieser Arbeit wird jedoch der Prozess verwendet, um die Diversität der Klassifikationen beziehungsweise des Ensembles zu maximieren. Dabei werden die Fairness-Metriken direkt in den Optimierungsprozess eingebunden, wodurch der bisher eher intuitive Begriff der Diversität von Modell-Ensembles konkreter wird.

3 Maximierung der Modell-Diversität

Dieses Kapitel beschreibt meinen Ansatz, wie die Modell-Diversität maximiert wird. Dabei wird in Abschnitt 3.1 dieser Ansatz beschrieben und dessen Wahl begründet. Abschnitt 3.2 geht in die Details des Ansatzes und beschreibt die genaue Umsetzung sowie die Implementierung. Abschnitt 3.3 geht kurz darauf ein, wie dieser Ansatz in dem FALCC-Framework für lokal faire Klassifikationen genutzt wird.

3.1 Lösungsansatz

Der gewählte Lösungsansatz besteht darin, mittels Hyperparameter-Optimierung die Diversität von Modell-Ensembles zu maximieren. Da sich mit Hyperparametertuning, wie in Abschnitt 2.3 beschrieben, die Modellarchitektur stark beeinflussen lässt, hat dies auch Auswirkungen auf die Diversität der Klassifikationen des Modell-Ensembles. Durch einen Optimierungsprozess wird dann die Kombination von Hyperparametern gefunden, durch die ein Modell-Ensemble eine optimale Struktur für diverse Klassifikationen erlangt. Ich habe als zu maximierende Score-Funktion eine Diversitätsmetrik gewählt (siehe Abschnitt 2.2), sodass der Optimierungsprozess die Hyperparameterkombination sucht, die das Modell-Ensemble nach dieser Metrik maximal diversifiziert.

Mit diesem Ansatz wird durch effiziente Suche ein diversifiziertes Modell-Ensemble gefunden.

Effizient, weil dieser Prozess parallelisiert werden kann, mit Grid-Search oder Random-Search. Dies beinhaltet das Trainieren von Modell-Ensembles pro Hyperparameterkombination und die Evaluierung der Score-Funktion. Mit Bayesian-Optimization wird zusätzlich eine effiziente und informierte Suche mit weniger Iterationen als obige Verfahren bereitgestellt. Damit findet sich schnell die eine optimale Kombination von Hyperparametern, trotz eines großen Suchraums, bestehend aus vielen verschiedenen Kombinationen.

Zudem ergibt sich durch die vielen möglichen Konfigurationen der Hyperparameter eines Modell-Ensembles die Möglichkeit, sich besser auf den unterliegenden Trainingsdatensatz anzupassen. Durch die Flexibilität des Ensembles über dessen Hyperparameter, ist die Wahrscheinlichkeit erhöht, dass sich dieses auf einen individuellen Datensatz anpassen kann. Damit wird auch eine hohe Diversität des Modell-Ensembles für jeden Datensatz gewährleistet.

Zuletzt lässt sich durch diese Optimierung der Hyperparameter auf Diversität, die Parameter und Modellstrukturen identifizieren, die eine besonders hohe Auswirkung auf die Diversität der Klassifikationen verursachen. Dies ermöglicht einen tieferen Einblick auf die Ursachen von Diversität in Modell-Ensembles, sowie eine Einschränkung des Suchraums durch vorausgegangene Experimente.

3.2 Beschreibung der Lösung

Dieser Abschnitt beschreibt genauer die Details meines Lösungsansatzes, sowie Einzelheiten der Implementierung. Es stellen sich vier Fragen, die vor dieser Hyperparameter-Optimierung beantwortet werden müssen.

1. Welche Ensemble-Methode eignet sich für die Maximierung der Diversität am besten?
2. Welche Diversitätsmetrik wird zur Quantifizierung verwendet?
3. Wie wird der Hyperparameter-Suchraum definiert?
4. Welche Hyperparameter-Suche eignet sich am besten?

Im Folgenden werden alle diese Fragen beantwortet.

3.2.1 Ensemble-Methode für die Maximierung der Diversität

Wie in Abschnitt 2.1 beschrieben, gibt es verschiedene Ansätze und Verfahren von Ensemble-Learning. Zuerst muss zwischen homogenen und heterogenen Ensemble-Learning abgewogen werden. Heterogenes Ensemble-Learning hat bereits eine fundamentale Modell-Diversität, dadurch, dass die Basismodelle aus unterschiedlichen Typen bestehen. Dadurch können die Basismodelle jeweils unterschiedliche Aspekte und Muster im Datensatz besser erkennen, als Basismodelle vom gleichen Typ, wodurch sich diese Diversität auch in den Klassifikationen widerspiegeln kann.

Dennoch habe ich mich in dieser Lösung gegen heterogene Ensemble-Learning Verfahren entschieden und homogene vorgezogen, aufgrund folgender Gesichtspunkte. Bei homogenem Ensemble-Learning müssen, im Gegensatz zu heterogenem, nur die Hyperparameter eines einzelnen Basismodells optimiert werden. Denn wie bereits in Abschnitt 2.3.1 erklärt, werden für jedes Basismodell die gleichen Hyperparameter pro Instanz verwendet. Dies vereinfacht zum einen die Suche deutlich, da sich der Suchraum dadurch erheblich reduziert, was der Rechenzeit entgegenkommt, insbesondere bei großen Datensätzen. Zusätzlich vereinfacht sich auch die Identifizierung wichtiger Eigenschaften der Basismodelle, die große Auswirkungen auf die Diversität haben, wie zum Beispiel eine bestimmte Hyperparamterkombination der Basismodelle. Aus diesen Gründen eignen sich homogene Ensemble-Learning-Verfahren mehr als heterogene.

Als Nächstes wird sich für zwei solcher homogenen Verfahren entschieden. Es gibt zwei grundlegende Kategorien von homogenen Ensemble-Learning-Verfahren, wie in Abschnitt 2.1.1 beschrieben wurde, Bagging und Boosting. Es wird jeweils ein weitverbreitetes Verfahren von Bagging und Boosting gewählt: Random-Forest und Adaboost. Dies ermöglicht ein besseres Verständnis und Identifizierung eventueller Kriterien für hohe Modell-Diversität. Random-Forest basiert auf dem Bagging-Ansatz mit zusätzlicher Randomisierung in der Attributauswahl für die Teilung in den einzelnen Entscheidungsbäumen. Adaboost hingegen ist ein bekannter Vertreter der Boosting Verfahren.

Um die Unterschiede von Hyperparameter-Optimierung, Diversität und Fairness der Basismodelle von diesen Verfahren einfacher zu vergleichen, habe ich bei Adaboost, wie bei Random-Forest ein Entscheidungsbaum als Basismodell verwendet. Damit lassen sich insbesondere Abweichungen der Diversität und Fairness zwischen diesen Verfahren vergleichen, wenn die Basismodelle dieser die gleichen Hyperparameter besitzen.

Für die Implementierung von Random-Forest [Bre01] und Adaboost [FS97] wird auf die *scikit-learn*-Bibliothek [PVG+11] für *Python* 3.8 zurückgegriffen. Diese Bibliothek bietet außerdem Implementierungen für Hyperparameter-Optimierung mit Grid- und Random-Search an, die in dieser Arbeit verwendet wurden. Für die Implementierung von Bayesian-Optimization wird die Bibliothek von [Nog–] verwendet.

Vorläufige Tests zeigen, welches Ausmaß unterschiedliche Hyperparameter auf die Diversität des Modell-Ensembles haben. In Abbildung 3.1 ist dies visuell für einen synthetischen Testdatensatz dargestellt.

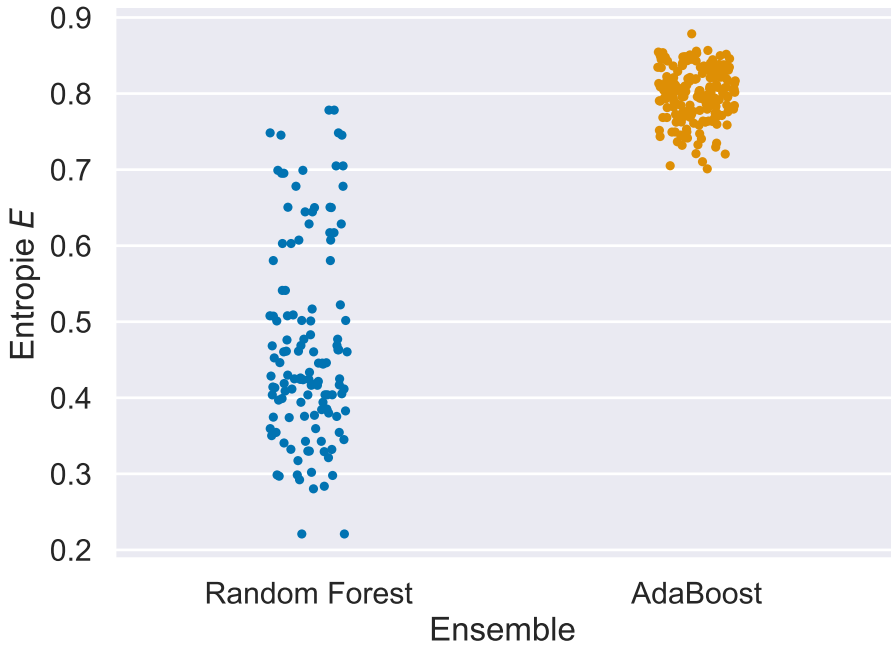


Abbildung 3.1: Modell-Ensemble-Diversität von Random-Forest und Adaboost mit unterschiedlichen Hyperparametern

Dabei ist jeder Punkt ein trainierter Random-Forest oder Adaboost mit einer Hyperparameterkombination aus dem Suchraum Tabelle 3.1. Alle nicht aufgelisteten sonstige Hyperparameter

Hyperparameter	Random-Forest	Adaboost
Anzahl Bäume	[3,10]	[15,19]
Split-Kriterium	{Gini, Entropie}	{Entropie}
max. Tiefe	[1,4]	[1,7]
max. Split-Attribute	{sqrt, log2}	{sqrt, log2, None}
Splitter	-	{Bester, Zufall}

Tabelle 3.1: Suchraum für Random-Forest- und Adaboost-Hyperparameter

der Verfahren, haben ihre Standardwerte, die Scikit-learn vorgibt angenommen. Damit ergibt sich eine Suchraumgröße von 128 für die Random-Forest-Hyperparameter und 210 für die Adaboost-Hyperparameter. In diesem Suchraum wurde jede Hyperparameterkombination mittels Grid-Search

für Random-Forest und Adaboost ausgewertet und die dazugehörige Entropie des Ensembles berechnet. Die Entropie-Werte sind immer in einem Intervall von $[0, 1]$ und je höher der Wert, desto größer ist die Diversität des Ensembles (siehe Definition 2.2.4). In Abbildung 3.1 ist zu sehen, wie sich die Diversitäten, insbesondere bei Random-Forest über einen großen Wertebereich verteilen. Während die Diversitäten von Adaboost in einem kleineren Intervall verteilt liegen, erreichen die Diversitäten eine deutlich höhere Entropie, als die der Random-Forests. Dies verdeutlicht die unterschiedlich starken Auswirkungen von Bagging und Boosting auf die Diversität. Für höchste Diversität ist Boosting gegenüber Bagging vorzuziehen. Sind jedoch Ensembles mit einem breiten Spektrum an Diversität gewünscht, ist Bagging vorzuziehen.

3.2.2 Diversitätsmetrik für die Maximierung

Wie in Abbildung 3.1, habe ich für alle späteren Evaluationen die Entropie aus Definition 2.2.4 als Metrik für Diversität eines Modell-Ensembles verwendet. Grundsätzlich gilt, dass alle hier vorgestellten Diversitätsmetriken stark untereinander korrelieren und sie sich lediglich in bestimmten Eigenschaften, wie Erwartungswert oder Rechenaufwand unterscheiden. Das heißt, für die Maximierung der Diversität ist es irrelevant, welche Diversitätsmetrik dafür vorgezogen wird. In Abbildung 3.2 ist die Korrelation zwischen den vorgestellten Diversitätsmetriken visuell dar-

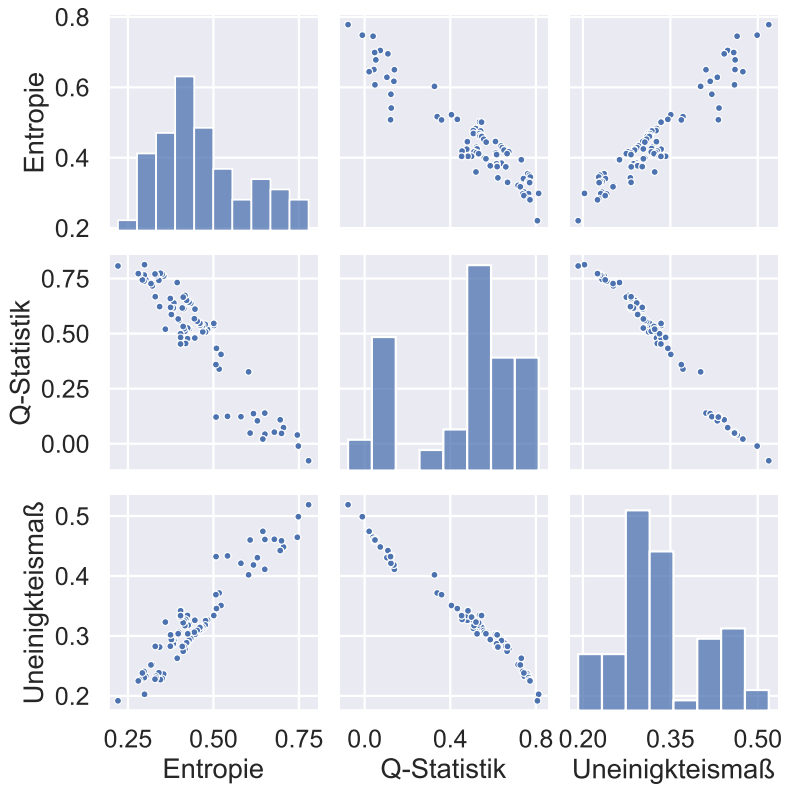


Abbildung 3.2: Korrelationen zwischen Diversitätsmetriken, sowie Histogramme, welches besagen, wie viele Random-Forests in gegebene Intervalle fallen

gestellt. Diese Ergebnisse kommen mittels eines Testdatensatzes, auf dem Random-Forests mit

verschiedenen Hyperparameterkombinationen trainiert wurden, zustande. So ist jeder Punkt ein Random-Forest und nach seiner Diversität angeordnet. Bemerkenswert ist die jeweils negative Korrelation zu der Q-Statistik. Wie in Abschnitt 2.2 beschrieben, ist die Q-Statistik hier die einzige Metrik, bei der geringere Werte für hohe Diversität stehen und daher negativ mit den anderen korreliert. Die Testergebnisse stimmen mit den Ergebnissen aus [KW03] überein. Dort wurden zu den hier vorgestellten Diversitätsmetriken noch zusätzlich sieben weitere verglichen und alle wiesen eine hohe Korrelation untereinander auf.

Außerdem habe ich mich für die Entropie als Diversitätsmetrik entschieden, da sie nicht paarweise berechnet wird und daher eine deutlich geringere Rechenzeit, als die paarweisen Diversitätsmetriken erfordert. Wie in Abbildung 3.3 zu sehen ist, steigt die durchschnittliche Rechenzeit der paarweisen

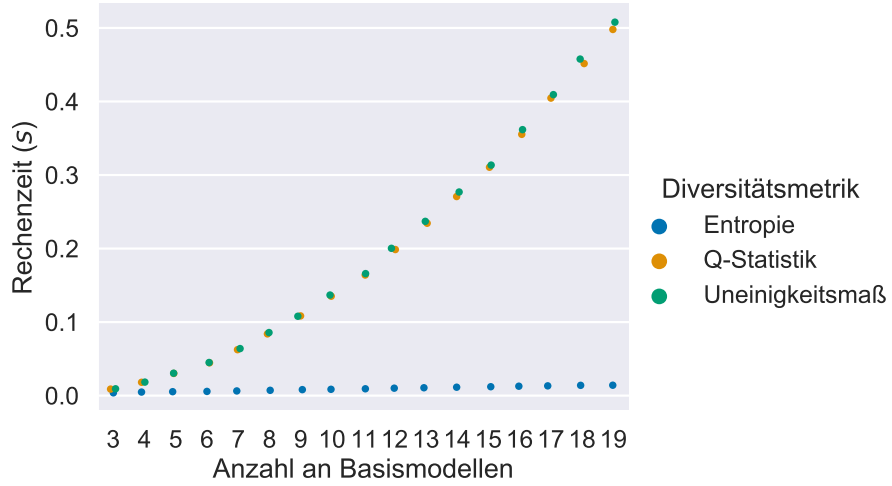


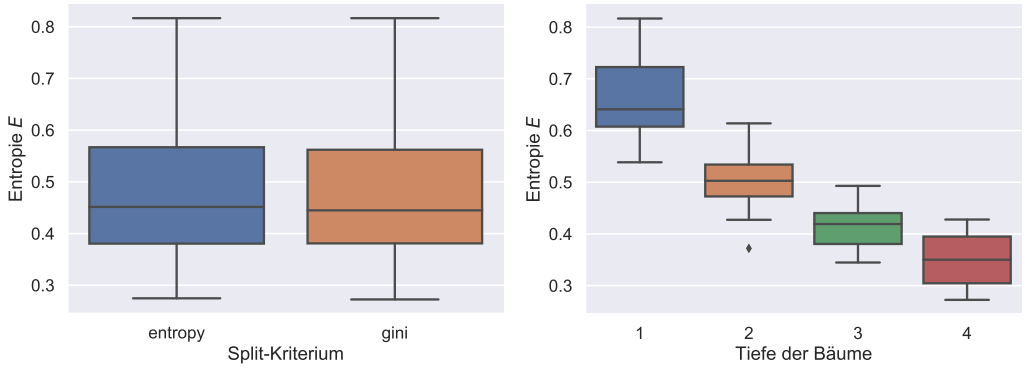
Abbildung 3.3: Vergleich der durchschnittlichen Rechenzeiten von Diversitätsmetriken in Relation zur Ensemblegröße für einen Testdatensatz

Diversitätsmetriken Entropie und Uneinigkeitsmaß quadratisch mit Anzahl an Basismodellen im Ensemble an. Auch diese Ergebnisse kommen mittels eines Testdatensatzes, auf dem Random-Forests mit verschiedenen Hyperparameterkombinationen, insbesondere Anzahl der Bäume, trainiert wurden, zustande. Die quadratische Rechenzeit kommt durch das Berechnen der einzelnen Diversitäten der Modelle untereinander zustande (siehe Definition 2.2.3). Es wird also jede mögliche Kombination von Basismodellen betrachtet und die paarweise Diversität berechnet, bis schlussendlich der Durchschnitt aller dieser Kombinationen als allgemeine Diversität zurückgegeben wird. Einen solchen Prozess gibt es bei der Berechnung der Entropie nicht. Die Rechenzeit dieser skaliert lediglich linear mit Anzahl an Basismodellen und ist daher auch für besonders große Ensembles geeignet. Linear, weil für einen festen Datensatz, für jeden Eintrag die Klassifikationen der Basismodelle dieses Eintrages gezählt wird (siehe Definition 2.2.4).

Aus diesen Gründen ist die Entropie eine effiziente und geeignete Metrik, nach der die Diversität von Modell-Ensembles maximiert werden kann.

3.2.3 Definition des Hyperparameter-Suchraums

Um zu bestimmen, welche Hyperparameter einen Einfluss auf die Diversität haben, habe ich vorläufige Tests anhand mehrerer Datensätze durchgeführt. Es wurden jeweils pro Datensatz Modell-Ensembles mit unterschiedlichen Hyperparameterkombinationen trainiert und deren Diversität gemessen. Dieser Suchraum enthält dabei alle möglichen Hyperparameter mit einem möglichst großen Wertebereich, welche für eine Optimierung infrage kommen. Anschließend wird nach den Werten eines zu untersuchenden Hyperparameter gruppiert und die Mittelwerte der Diversitäten verglichen. Aus diesen Vergleichen werden dann die Hyperparameter mit stärkstem Einfluss auf die Diversität in den finalen Suchraum einbezogen. Anhand dieser Ergebnisse werden Hyperparameter aus der Optimierung ausgeschlossen, welche keinen Einfluss auf die Diversität des Modell-Ensembles haben. In Abbildung 3.4 ist zu sehen, wie die Hyperparameter *Split-Kriterium* und



(a) Hyperparameter: Split-Kriterium der Bäume **(b)** Hyperparameter: Maximale Tiefe der Bäume

Abbildung 3.4: Einfluss von zwei Hyperparametern eines Random-Forest auf die Modell-Diversität als Boxplots

maximale Tiefe unterschiedlichen Einfluss auf die Diversität eines Random-Forest hat. Es ist zu erkennen, wie die maximale Tiefe der Bäume einen deutlichen Einfluss auf die Diversität hat. Eine geringe Tiefe der Bäume in Random-Forest führt zu mehr Diversität. Dies lässt sich durch die Vorgehensweise des unterliegenden Algorithmus für Random-Forests [Bre01] auch erklären. Da für jeden Split eines Baumes nur eine zufällige Teilmenge an Attributen betrachtet wird, besteht eine hohe Wahrscheinlichkeit, dass die einzelnen Bäume nach unterschiedlichen Attributen gesplittet werden. Hinzukommt, dass bei einer geringen Tiefe auch entsprechend weniger gesplittet wird und die Bäume daher sehr grob die Daten aufteilen und nur mit geringer Genauigkeit klassifiziert werden kann. Dies resultiert dann zu vielen unterschiedlichen Klassifikationen der einzelnen Bäume, was die intuitive Definition für Diversität ist. Im Gegensatz dazu hat das Split-Kriterium kaum einen Einfluss auf die Diversität von Random-Forests und kann daher bei dem finalen Suchraum ausgeschlossen werden. Analog lassen sich so alle Hyperparameter einer Ensemble-Strategie untersuchen und nur die effektivsten in die Suche einbinden.

Bezüglich des Wertebereichs wird ähnlich vorgegangen. Es werden lediglich Werte der jeweiligen Hyperparameter in den Suchraum eingebunden, welche zu hoher Diversität führen können. Abbildung 3.5 zeigt beispielsweise, dass der mittlere Diversitätswert von Adaboost gegen eine Tiefe von

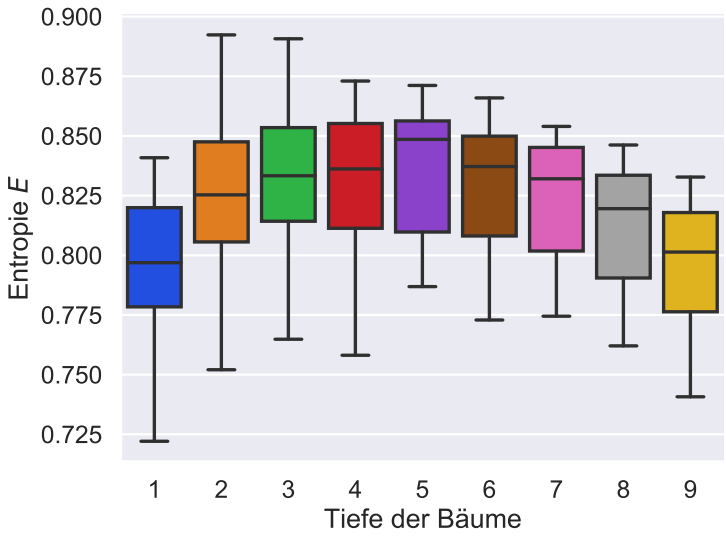


Abbildung 3.5: Einfluss des Hyperparameters *maximale Tiefe der Bäume* von Adaboost auf die Diversität

5 ein Maximum bildet. Dadurch kann der Wertebereich auf ein Intervall um 5 beschränkt werden. Trotzdem sollte der Wertebereich der Hyperparameter nicht zu gering gewählt werden, da sich die Modelle sonst schlecht auf andere Datensätze optimieren lassen.

Letztlich habe ich mit dem obigen Verfahren die Suchräume aus Tabelle 3.1 für Random-Forest und Adaboost festgelegt.

3.2.4 Das Hyperparameter-Optimierungsverfahren

In Rahmen dieser Arbeit habe ich mich für Grid-Search als Hyperparameter-Optimierungsverfahren entschieden. Grund dafür ist, dass Grid-Search trotz dessen Komplexität für die hier verwendeten Suchräume noch effektiv einsetzbar ist und die Auswertungen aller möglichen Hyperparameterkombinationen des Suchraums für die spätere Evaluation ausleuchtend ist. Insbesondere ist es dadurch möglich, auf alle generierten Modell-Ensembles des Suchraums über deren Diversität zuzugreifen. Für die spätere Evaluation wurde ein Algorithmus implementiert, der es ermöglicht, ein Modell-Ensemble mit einer gewünschten Diversität aus dem Suchraum zurückzugeben. Der Algorithmus gibt dann das Modell-Ensemble des Suchraums zurück, dessen Diversität am nächsten bei der Eingabe liegt. Da Grid-Search immer garantiert die Kombination an Hyperparametern findet, die das Ensemble maximal diversifiziert, kann insbesondere dieses Modell-Ensemble immer zurückgegeben werden. Außerdem liegen für den gewählten Suchraum (siehe Tabelle 3.1) ausschließlich kategorische und diskrete Hyperparameter vor, wofür Grid-Search geeignet ist.

Dennoch wurden in das Framework zusätzlich Random-Search und Bayesian-Optimization implementiert. Grund dafür sind die möglich geringeren Rechenzeiten dieser Optimierungsverfahren für dennoch hohe resultierende Diversität.

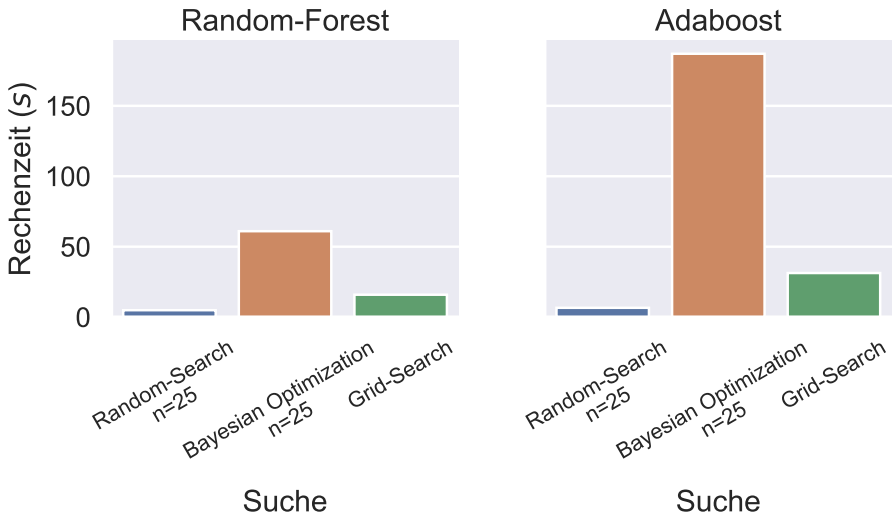


Abbildung 3.6: Rechenzeit der Optimierungsverfahren im Vergleich

Abbildung 3.6 zeigt die Rechenzeit der Suchverfahren im Vergleich für Random-Forest und Adaboost für die Suchräume aus Tabelle 3.1 auf einem synthetischen Testdatensatz. Random-Search und Bayesian-Optimization wurden hier auf 25 Iterationen beschränkt, das heißt Random-Search wertet 25 zufällige Hyperparameterkombinationen aus und Bayesian-Optimization führt 25 Suchschritte aus. Grid-Search sucht den gesamten Suchraum, von einer Größe von 128 für Random-Forest und 180 für Adaboost ab. Die Rechenzeit hier ist die Summe der Optimierungszeiten von zwölf Datensätzen, auf denen die Ensembles trainiert wurden. Zu sehen ist, dass Bayesian-Optimization mit Abstand die längste Rechenzeit benötigt. Dies kann daran liegen, dass Bayesian-Optimization einen größeren Overhead hat, als Grid- oder Random-Search und daher erst bei besonders großen Suchräumen effektiver wird. Dieses Verhältnis der Rechenzeiten zeigte sich auch über weitere Testdatensätze hinweg.

Abbildung 3.7: Maximale erreichte Diversität der Suchverfahren

Abbildung 3.7 zeigt die Effektivität dieser Optimierungsverfahren und bildet jeweils die maximal erreichte Entropie dieser ab. Auch diese Werte ergaben sich für einen synthetischen Testdatensatz und das Verhältnis ist Datensatz übergreifend. Wie zu erwarten, findet Grid-Search immer die Kombination, die zu höchster Entropie führt. Bayesian-Optimization schließt etwas besser ab als Random-Search, aufgrund der informierten Suche, welche die Wahrscheinlichkeit auf bessere Entropie-Werte erhöht, während es bei Random-Search dem Zufall überlassen wird.

3.3 Einbindung in das FALCC-Framework

Wie in [LH23] bereits validiert wurde, führt eine hohe Modell-Diversität in einem Ensemble in der Regel zu faireren lokalen Klassifikationen mit dem FALCC-Framework. Daher wird die obige Methode zur Maximierung der Modell-Diversität in das FALCC-Framework eingebunden, um die Fairness zu steigern. Letztlich wird der Prozess der Diversifizierung von Modell-Ensembles mit AdaptedAdaboost durch den obigen Ansatz ersetzt. Die Auswirkungen von Modell-Diversität auf Fairness und Genauigkeit werden dennoch im folgenden Kapitel untersucht.

4 Evaluation

In diesem Kapitel werden Ergebnisse von unterschiedlichen Experimenten evaluiert. Abschnitt 4.1 erläutert den Versuchsaufbau sowie die Ziele der Evaluation. Anschließend werden die jeweiligen Ziele in Abschnitt 4.2 bis 4.6 dieser Evaluation behandelt.

4.1 Versuchsaufbau

Im Folgenden wird schrittweise der grundlegende Versuchsaufbau der Experimente erläutert.

4.1.1 Modell-Ensemble-Verfahren

In den Versuchen wird die Rechenzeit, Diversität und die anschließende Fairness der Klassifikationen in dem FALCC-Framework von verschiedenen Modell-Ensemble-Verfahren verglichen. Einbezogen werden die Ensembles AdaptedAdaboost, sowie Random-Forest und Adaboost mit optimierten Hyperparametern für maximale Diversität. AdaptedAdaboost generiert dabei als Voreinstellung ein Ensemble bestehend aus zwei logistischen Regressionen, zwei linearen Support-Vektor-Maschinen und zwei Entscheidungsbäumen als Basismodelle. Die Random-Forest- und Adaboost-Ensembles bewegen sich innerhalb des oben definierten Suchraums inTabelle 3.1. Falls nicht anders angegeben, werden diese Ensembles immer mit höchster erreichter Diversität aus dem Suchraum verglichen.

4.1.2 Metriken

Die Diversität der Modell-Ensembles wird wie in Abschnitt 3.2.2 begründet mit Entropie (siehe Definition 2.2.4) gemessen. Die daraus resultierende lokale Fairness des FALCC-Frameworks wird mit den Metriken aus Abschnitt 2.4.1 gemessen. Diese werden so normiert, dass sie alle zwischen 0 (= *maximal Fair*) und 1 (= *maximal Unfair*) liegen. Das Ergebnis der lokalen Fairness ist der jeweilige Durchschnitt dieser Metriken über alle lokalen Regionen, welche das FALCC-Framework bildet. Zusätzlich dazu wird auch die globale Fairness mit diesen Metriken gemessen und die individuelle mit der Konsistenz-Metrik (siehe Definition 2.4.6). Die Genauigkeit der Klassifikationen ist der Anteil der korrekt klassifizierten Einträge von allen Einträgen.

Zudem werden die Rechenzeiten der Verfahren verglichen. Diese werden aufgeteilt in Rechenzeit für die Generierung eines Ensembles, dies beinhaltet insbesondere auch die Laufzeit der Hyperparameter-Optimierung von Random-Forest und Adaboost und in die Laufzeit der Bewertungsphase des FALCC-Frameworks. Diese Aufteilung wird gemacht, da davon ausgegangen werden kann, dass je größer das Ensemble ist, desto mehr Basismodelle müssen für jede Region des Datensatzes durch die Offline-Phase des FALCC-Frameworks bewertet werden.

4.1.3 Datensätze

Die Evaluation wird anhand verschiedener realer und synthetischer Datensätze durchgeführt. Die verwendeten transformierten Versionen der realen Datensätze stammen alle aus dem Repository des FALCC-Frameworks[1].

Die realen Datensätze umfassen den *US Census Demographic* (kurz *ACS2017*) Datensatz [Bur19] und den *Adult* Datensatz [DG17], welche beide das Gehalt und weitere Attribute von Bürgern verschiedener Regionen in den USA enthält. Das geschützte Attribut ist hier die Ethnie beziehungsweise das Geschlecht der Bürger und das Label Gehalt. Zudem wurde der *Communities and Crime* (kurz *Communities*) [RB02] Datensatz verwendet. Dieser Datensatz enthält Daten über gewaltvolle Verbrechen der Bevölkerung aus den USA innerhalb verschiedener Gemeinschaften. Das geschützte Attribut ist hier die Ethnie. Letztlich wurde noch der *COMPAS Recidivism Racial Bias* (kurz *COMPAS*) [Pro17] Datensatz einbezogen. Dieser Datensatz enthält Daten zu Straftätern und deren Bewertung der Rückfallwahrscheinlichkeit. Das geschützte Attribut ist auch hier die Ethnie.

Die synthetischen Datensätze lassen sich in zwei Varianten einteilen. Der *social* Datensatz besitzt einen 30prozentigen direkten Bias, das heißt, das geschützte Attribut korreliert direkt mit dem Label. Der *implicit* Datensatz besitzt einen 30-prozentigen indirekten Bias. Das heißt, das geschützte Attribut hat keinen direkten Einfluss auf das Label, jedoch korreliert es mit mehreren weiteren Attributen, welche einen Einfluss haben.

Datensatz	geschütztes Attribut	# an Einträgen	# an Attributen
ACS2017	Ethnie	$72k$	23
Adult	Geschlecht	$46k$	21
Communities	Ethnie	$2k$	91
COMPAS	Ethnie	$6.1k$	7
social30	Label	$14.6k$	8
implicit30	Label	$14.5k$	8

Tabelle 4.1: Eckdaten der verwendeten Datensätze

In Tabelle 4.1 sind die wichtigsten Eckdaten, wie die Größen der Datensätze aufgelistet. Alle Datensätze werden zufällig von dem FALCC-Framework in 50% für Training, 35% für Validierung und 15% für Testen geteilt.

4.1.4 Ziele der Evaluation

Für die Evaluation habe ich folgende Vergleiche zum Ziel gesetzt. Vergleiche die Diversität (1), Fairness (2), Genauigkeit (3) und Laufzeit (4) von AdaptedAdaboost, sowie Random-Forest und Adaboost mit jeweils optimierten Hyperparametern.

[1] https://github.com/tLslhyCL/FALCC

Die Messung und der Vergleich der Modell-Diversitäten von den Modell-Ensembles, welche im FALCC-Framework eingebracht werden, soll in erster Linie die Effektivität der Hyperparameter-Optimierung bestätigen. Von dort aus können dann die Auswirkungen der unterschiedlich diversen Modell-Ensembles auf die Fairness und die Genauigkeit untersucht werden. Zum einen kann die Hypothese aus [LH23], dass FALCC mit diverseren Modell-Ensembles zu lokal faireren Klassifikationen führt, nochmals validiert werden. Zum anderen kann mit der Messung der Genauigkeit der Klassifikationen auch ein Trade-off zwischen Fairness und Genauigkeit hervorgehoben werden. Die Laufzeit wird gemessen, um mögliche Bottlenecks des Algorithmus und Stellen für weitere Optimierungen zu identifizieren.

4.2 Evaluationsziel 1: Vergleich der Diversitäten

Das AdaptedAdaboost-Verfahren stellt eine Erweiterung des grundlegenden Adaboost-Verfahrens dar, indem es eine zusätzliche Rotation der Basismodelle für mehr Modell-Diversität einbringt. Um die Diversität des AdaptedAdaboost-Verfahrens besser einschätzen und mit anderen Ensemble-Methoden vergleichen zu können, wird in diesem Abschnitt eine Quantifizierung der Diversität mittels Entropie vorgenommen. Somit die Entropie-Werte der verschiedenen Ensemble-Verfahren, nämlich AdaptedAdaboost, Random-Forest und Adaboost, für unterschiedliche Datensätze miteinander verglichen. Hierbei ist es wichtig zu beachten, dass die Hyperparameter von Random-Forest sowie Adaboost speziell auf eine hohe Diversität ausgelegt sind. Der Vergleich dieser Verfahren ist notwendig, um in den folgenden Abschnitten weitere Folgerungen über die Auswirkungen der Diversität auf die Qualität der Klassifikationen ziehen zu können.

Ensemble	ACS2017	Adult	Communities	COMPAS	social30	implicit30
AdaptedAdaboost	0.0732	0.3549	0.1412	0.6418	0.3067	0.3469
Random-Forest	0.5305	0.1868	0.4433	0.6590	0.7568	0.7956
AdaBoost	0.8273	0.8922	0.8534	0.9110	0.8716	0.8859

Tabelle 4.2: Diversitäten (Entropie) der Ensemble-Verfahren im Vergleich

Die in Tabelle Tabelle 4.2 aufgeführten Entropie-Werte (höhere Entropiewerte bedeuten mehr Modell-Diversität) für die verschiedenen Ensemble-Verfahren zeigen einige bemerkenswerte Erkenntnisse. Mit Ausnahme eines Datensatzes weisen die Random-Forest- und Adaboost-Verfahren, die jeweils mit optimierten Hyperparametern trainiert wurden, eine teilweise deutlich höhere Diversität auf als das AdaptedAdaboost-Verfahren (bis zu zehnmal höher). Zudem ist Adaboost in allen Fällen diverser als Random-Forest. Diese Ergebnisse verdeutlichen zum einen die Effektivität der Hyperparameter-Optimierung bei der Maximierung der Modell-Ensemble-Diversität und zum anderen die Überlegenheit von Adaboost gegenüber Random-Forest, insbesondere bei Datensätzen, bei denen Random-Forest Schwierigkeiten hat, eine hohe Diversität zu erreichen. Es ist jedoch wichtig zu betonen, dass die höhere Diversität allein nicht zwangsläufig zu einer besseren Klassifikationsleistung führt. In den nachfolgenden Abschnitten wird daher untersucht, wie sich die beobachtete Diversität auf die Fairness der Klassifikationen und letztlich auf die Genauigkeit der Modelle auswirkt.

4.3 Evaluationsziel 2: Vergleich der Fairness

Wie in [LH23] bereits gezeigt, ermöglicht das FALCC-Framework mit AdaptedAdaboost über lokal faire Klassifikationen. In diesem Abschnitt wird diese Fairness mit der Fairness von FALCC verglichen, wenn optimierte Random-Forest- und Adaboost-Ensembles als zugrunde liegende Verfahren verwendet werden. Da das FALCC-Framework eine spezifische lokale Fairness-Metrik zur Optimierung benötigt, zeigen die Ergebnisse der lokalen Fairness auch immer die Werte für die optimierte Metrik.

Abbildung Abbildung 4.1 zeigt die gemessenen Fairness-Werte von FALCC für verschiedene Datensätze. Je niedriger die Werte, desto fairer sind die Klassifikationen von FALCC in Bezug auf das jeweilige Ensemble-Verfahren und die betrachtete Fairness-Metrik. Zu sehen ist, dass mit den Ensemble-Verfahren Random-Forest oder Adaboost auf den realen Datensätzen überwiegend (nach 16 von 18 Fairness-Metriken) lokal fairere Klassifikationen getroffen werden, als mit AdaptedAdaboost. Überwiegend (11 von 18 Fairness-Metriken) werden mit Adaboost zudem fairere Klassifikationen getroffen, als mit Random-Forest als Ensemble-Verfahren. Auffällig ist, dass bei den synthetischen Datensätzen Random-Forest für Fairness gegenüber Adaboost überlegen ist. Diese Ergebnisse legen nahe, dass die höhere Modell-Diversität von Random-Forest und Adaboost mit optimierten Hyperparametern zu lokal faireren Klassifikationen führt als bei Verwendung von AdaptedAdaboost. Dies unterstreicht die Bedeutung von Diversität in Modell-Ensembles und deren Einfluss auf die Fairness von Klassifikationen über das FALCC-Framework.

Neben den lokalen Fairness-Metriken wurden auch die globale und individuelle Fairness in den betrachteten Ensemble-Methoden untersucht. Das FALCC-Paper geht davon aus, dass eine hohe lokale Fairness auch eine hohe globale und individuelle Fairness induziert. Abbildung Abbildung 4.2 zeigt die Werte der globalen Fairness-Metriken für zwei Datensätze. Es ist erkennbar, dass die Fairness-Werte der optimierten Ensemble-Verfahren Random-Forest und Adaboost denen von AdaptedAdaboost nicht stark unterlegen sind. Ähnliche Verhältnisse zeigen sich in den übrigen Datensätzen.

Abbildung 4.3 zeigt die durchschnittlichen Werte der individuellen Fairness-Metriken für alle Datensätze. Die Durchschnitte entstehen durch die drei Durchläufe von FALCC in denen jeweils auf demographic parity, equalized odds und treatment equality optimiert werden. Zudem wird durch die schwarzen Linien das Maximum und das Minimum der erreichten Konsistenzen dargestellt. Die Ergebnisse zeigen, dass die Konsistenzen, die durch Random-Forest und Adaboost entstehen, sich nicht stark von den Werten mit AdaptedAdaboost unterscheiden. Dies legt nahe, dass die Verwendung von optimierten Random-Forest- und Adaboost-Ensembles in Bezug auf globale und individuelle Fairness-Metriken ähnliche und überwiegend bessere Ergebnisse wie AdaptedAdaboost erzielen kann.

Insgesamt verdeutlichen diese Resultate die Bedeutung der Auswahl geeigneter Ensemble-Methoden für das FALCC-Framework, um sowohl lokale als auch globale und individuelle Fairness zu steigern. Während AdaptedAdaboost in einigen Fällen eine gute Leistung erzielt, zeigen die Ergebnisse, dass Random-Forest und Adaboost mit optimierten Hyperparametern in vielen Fällen vergleichbare und bessere Fairness aufweisen. Wichtig ist jedoch, dass Fairness nicht der einzige Aspekt ist, der bei der Bewertung dieser Ensemble-Verfahren in FALCC berücksichtigt werden sollte. Die Genauigkeit ist ebenfalls entscheidend für die Auswahl eines geeigneten Modell-Ensembles.

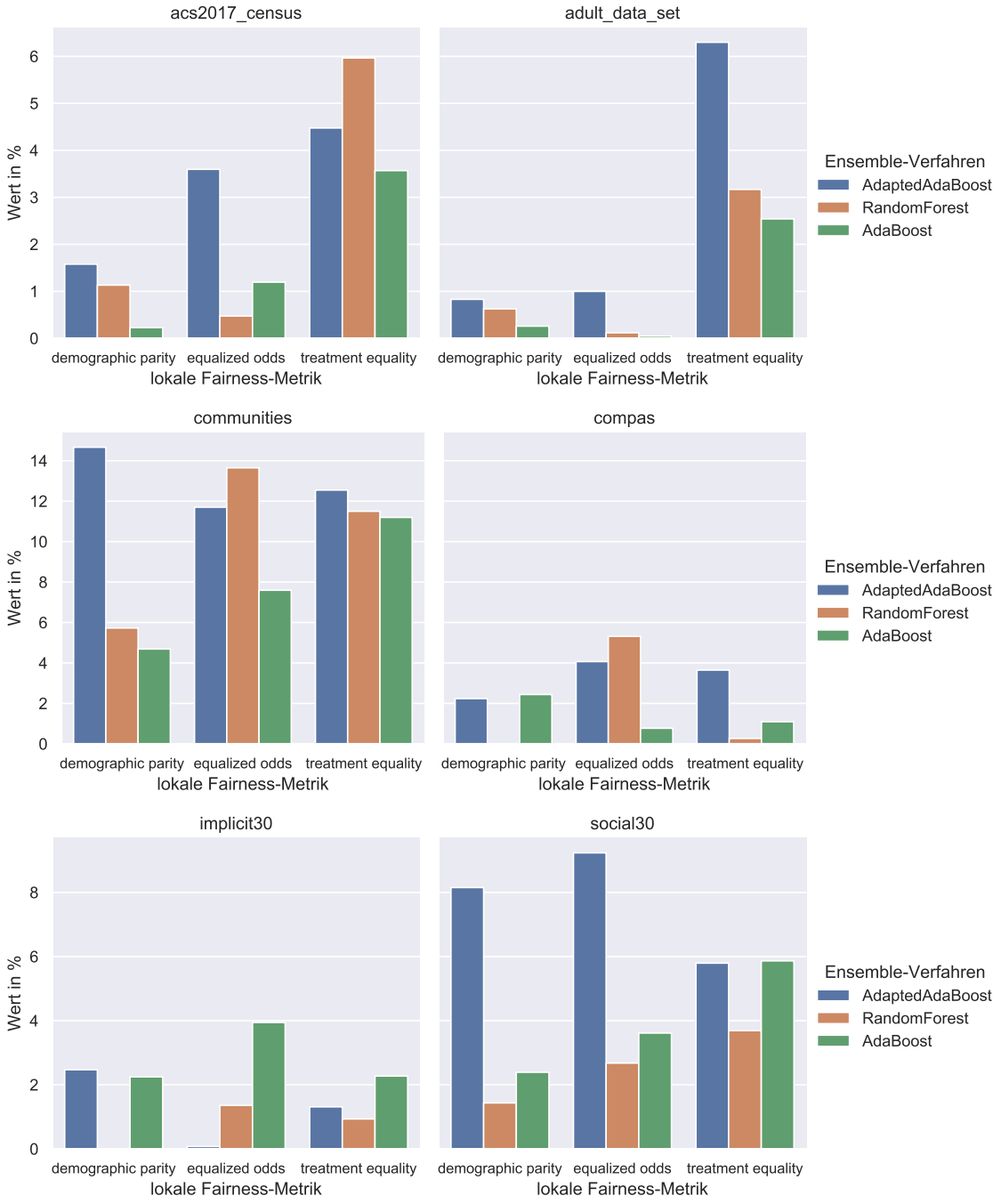


Abbildung 4.1: Vergleich der lokalen Fairness der verschiedenen Ensemble-Verfahren mit FALCC auf unterschiedlichen Datensätzen.

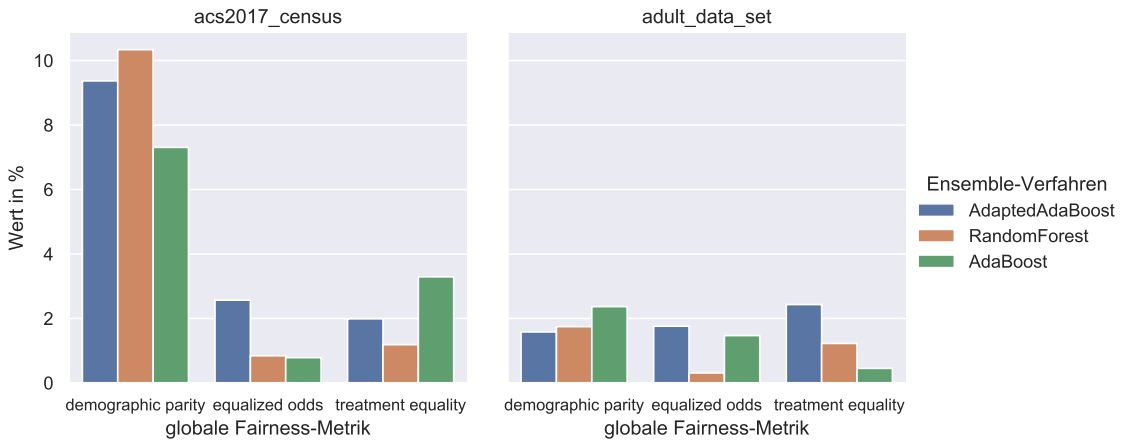


Abbildung 4.2: Vergleich der globalen Fairness der verschiedenen Ensemble-Verfahren mit FALCC auf unterschiedlichen Datensätzen.

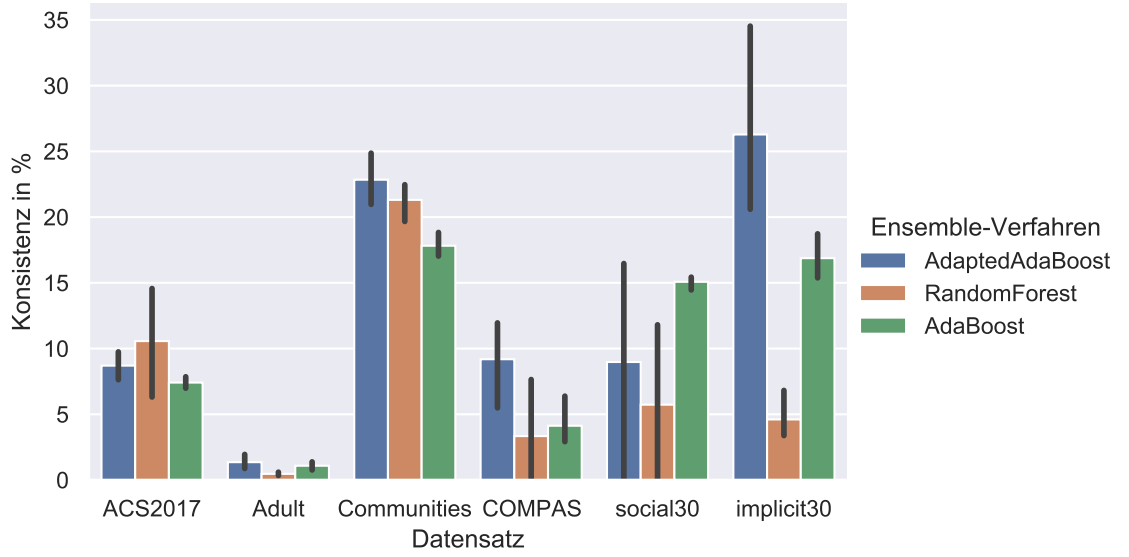


Abbildung 4.3: Vergleich der individuellen Fairness der verschiedenen Ensemble-Verfahren mit FALCC auf unterschiedlichen Datensätzen.

4.4 Evaluationsziel 3: Vergleich der Genauigkeiten

Ziel von FALCC ist es, lokal faire Klassifikationen zu treffen, ohne die Genauigkeit der Klassifikationen beeinträchtigen zu müssen. In [LH23] wurde bereits ein Trade-off zwischen Fairness und Genauigkeit aufgedeckt. Dieser Abschnitt untersucht, welche Auswirkungen die faireren Klassifikationen von FALCC mit diverseren Modell-Ensembles auf die Genauigkeit dieser haben.

In Abbildung 4.4 sind die durchschnittlichen Genauigkeiten von FALCC aus allen Durchläufen, in denen jeweils auf die unterschiedlichen lokalen Fairness-Metriken optimiert wurde. Auch hier stellen die schwarzen Linien das Maximum und das Minimum der erreichten Genauigkeiten dar. Zu sehen

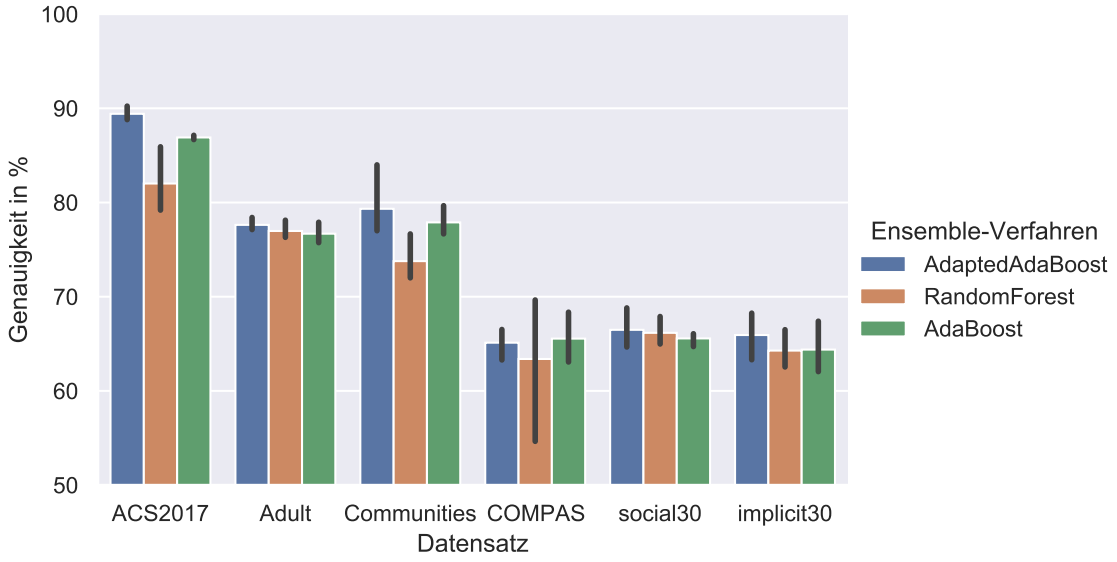


Abbildung 4.4: Vergleich der durchschnittlichen Genauigkeiten von FALCC mit verschiedenen Ensemble-Verfahren auf unterschiedlichen Datensätzen

ist, dass sich die Genauigkeit in der Regel mit Random-Forest und Adaboost leicht, verschlechtert haben. Die Ausnahme bilden die Werte der Genauigkeit mit Random-Forest auf den Datensätzen ACS2017, Communities. und COMPAS, bei denen eine deutlichere Verschlechterung zu vermerken sind. Mit AdaptedAdaboost wird weiterhin in jedem der Datensätze die höchste Genauigkeit erreicht. Bemerkenswert ist dennoch die hohe Genauigkeit von Adaboost mit optimierten Hyperparametern, für die deutlich besseren Fairness-Werte, die damit erreicht wurden.

Zusammenfassend lässt sich daraus schließen, dass sich auch mit den optimierten Modell-Ensembles Random-Forest und Adaboost der Trade-off zwischen Fairness und Genauigkeit nicht unterbinden lässt, dafür sich aber insbesondere mit Adaboost reduziert ließ.

4.5 Evaluationsziel 4: Vergleich der Rechenzeiten

Das FALCC-Framework lässt sich in eine offline und eine Online-Phase einteilen. Die Rechenzeit der Online-Phase verändert sich durch die Implementierung der neuen Ensemble-Verfahren kaum, da in dieser Phase neue Einträge nur einer Region zugeordnet und von dem dazugehörigen Basismodell klassifiziert werden. Damit ändert sich an der Online-Phase nichts, außer dass die Basismodelle in diesem Fall ausschließlich aus Entscheidungsbäumen bestehen, welche die Rechenzeit der Klassifikation nicht beeinträchtigen.

Da sich aber die Trainingsphase des Modell-Ensembles, sowie das Ensemble-Verfahren geändert hat, welche jeweils Teil der Offline-Phase sind, hat dies einen großen Einfluss auf dessen Rechenzeit. Die Messung der Rechenzeiten wird eingeteilt in Trainingsphase und Modell-Bewertung. Die Trainingsphase umschließt den gesamten Hyperparameter-Optimierungsprozess, in dem alle möglichen Hyperparameterkombinationen eines Suchraums auf Modell-Diversität evaluiert werden,

sowie das letztliche Zurückgeben des Modell-Ensembles mit höchster Diversität. Die Modell-Bewertung ist der Teil der Offline-Phase, in dem jedes Basismodell auf jeder Region des Datensatzes auf optimale Fairness und Genauigkeit geprüft wird.

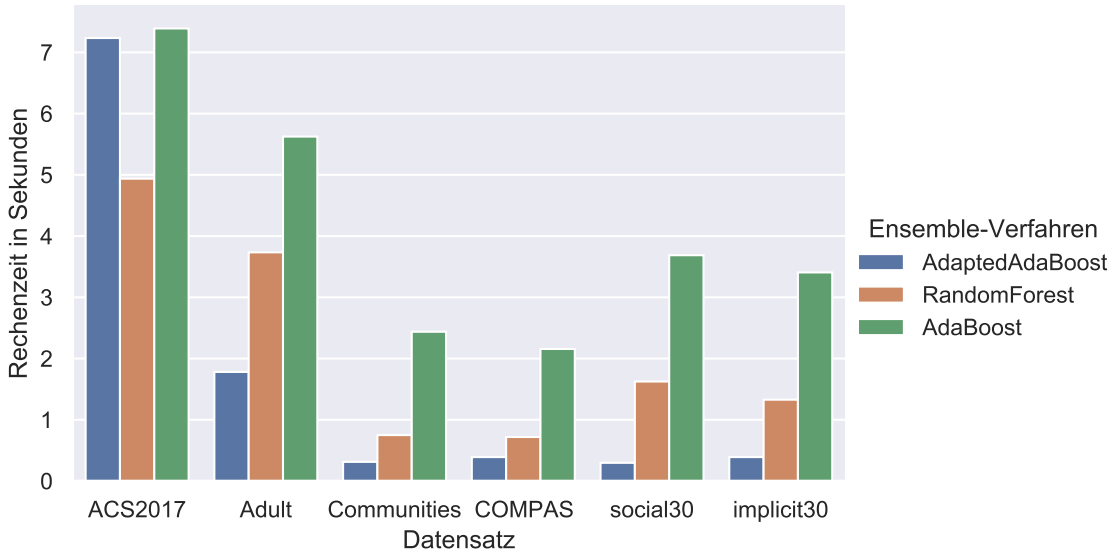


Abbildung 4.5: Vergleich der Rechenzeit der Trainingsphasen der verwendeten Ensemble-Learning-Prozesse auf unterschiedlichen Datensätzen.

Abbildung 4.5 zeigt die Rechenzeiten der Trainingsphasen der jeweiligen Ensemble-Verfahren auf unterschiedlichen Datensätzen. Bis auf den Datensatz ACS2017 sind die Zeiten der Trainingsphasen von Random-Forest und Adaboost länger als die von AdaptedAdaboost. Da ACS2017 der mit Abstand größte Datensatz ist (siehe Tabelle 4.1) ist es möglich, dass die komplexeren Basismodelle, wie die logistische Regression, und die lineare Support-Vektor-Maschine, deutlich mehr Zeit für ihr Training benötigen, als auf kleineren Datensätzen. Ansonsten sind die längeren Rechenzeiten von Random-Forest und Adaboost der Hyperparameter-Optimierung zuzuschreiben. Da der Suchraum und die Ensemblegrößen von Adaboost deutlich größer sind (siehe Tabelle 3.1), benötigt diese Trainingsphase länger als die von Random-Forest.

In Abbildung 4.6 sind die Rechenzeiten der Modell-Bewertungen zu sehen. Hier wird deutlich, dass Adaboost jeweils deutlich (bis zu 5 Minuten) mehr Zeit beansprucht, als die übrigen Ensemble-Verfahren. Grund dafür ist die deutlich größere Ensemblegröße von Adaboost. Der Suchraum Tabelle 3.1 zeigt, dass die Ensemble-Größe von Adaboost aus bis zu 19 Basismodellen bestehen kann, während AdaptedAdaboost aus 6 und Random-Forest aus bis zu 10 bestehen. Da während der Modell-Bewertungsphase alle möglichen Kombinationen von Basismodell und Regionen evaluiert werden müssen, steigt die Anzahl an Evaluationen mit größerer Ensemblegröße exponentiell an, was zu einer deutlich längeren Rechenzeit führt.

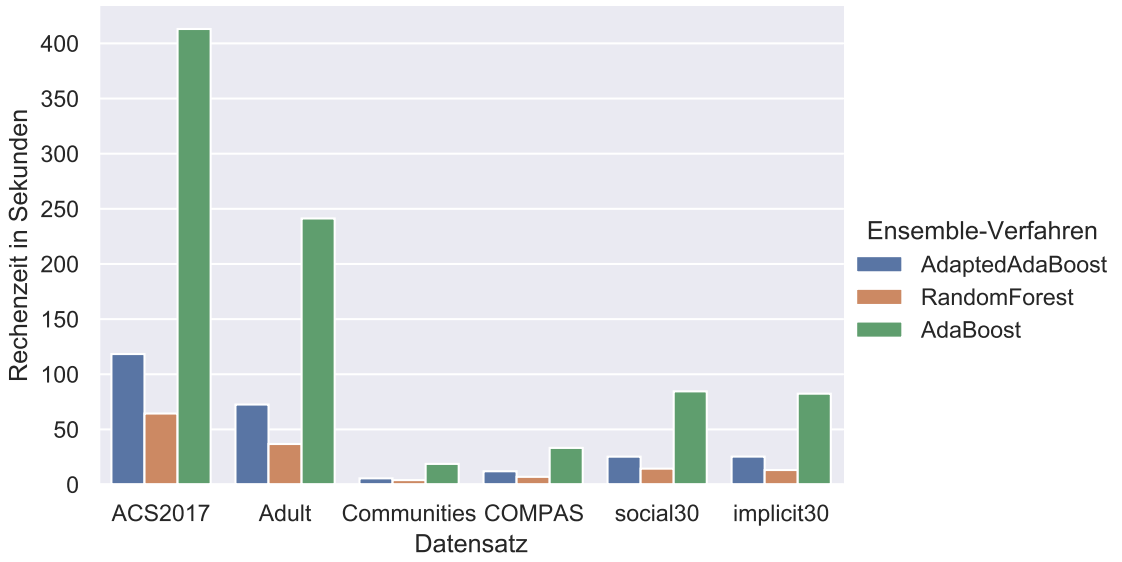


Abbildung 4.6: Vergleich der Rechenzeit der Modell-Bewertungsphase von FALCC mit den verwendeten Ensemble-Learning-Prozessen auf unterschiedlichen Datensätzen.

4.6 Evaluationsziel 5: Einfluss der Diversität auf Qualität der Klassifikationen

In diesem Abschnitt werden abschließend die Zusammenhänge zwischen der Modell-Diversität und Fairness und Genauigkeit der Klassifikationen durch das FALCC-Framework untersucht. Für die Untersuchungen wird sich speziell auf Random-Forest-Ensembles beschränkt, da diese mit unterschiedlichen Hyperparametern ein breites Spektrum an verschiedenen Diversitätswerten bieten. Für die Messungen wurden für unterschiedliche Datensätze Random-Forest-Ensembles mit jeweils niedrigster, viertel, halber, dreiviertel und maximaler Entropie des Suchraums für das FALCC-Framework verwendet und auf Fairness und Genauigkeit untersucht. Abbildung 4.7 bildet

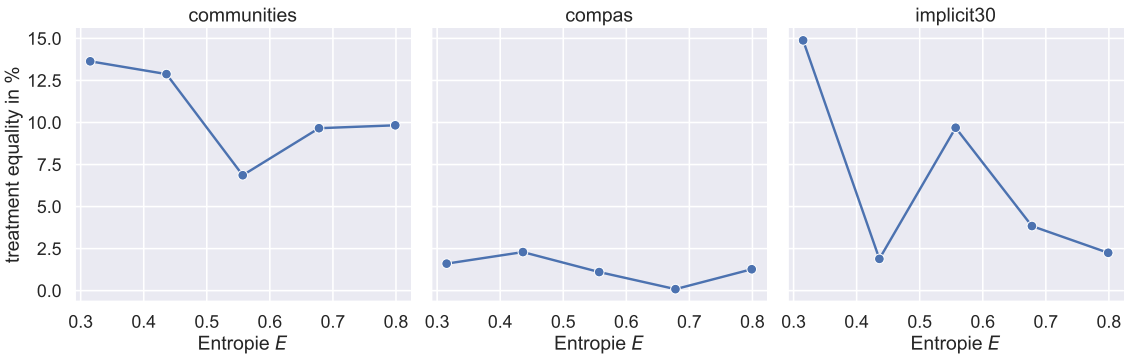


Abbildung 4.7: Zusammenhang zwischen Diversität und lokaler Fairness von Random-Forest in FALCC auf verschiedenen Datensätzen.

die Entropiewerte in Relation zu lokaler Fairness, gemessen in treatment equality ab. Nach wie vor

bedeutet, je höher die Entropiewerte sind, desto diverser ist das Modell-Ensemble. Trotz weniger Schwankungen ist ein allgemeiner Abwärtstrend der treatment equality werte über die Datensätze hinweg zu erkennen. Das heißt, mit steigender Modell-Diversität werden die Klassifikationen in der Regel fairer.

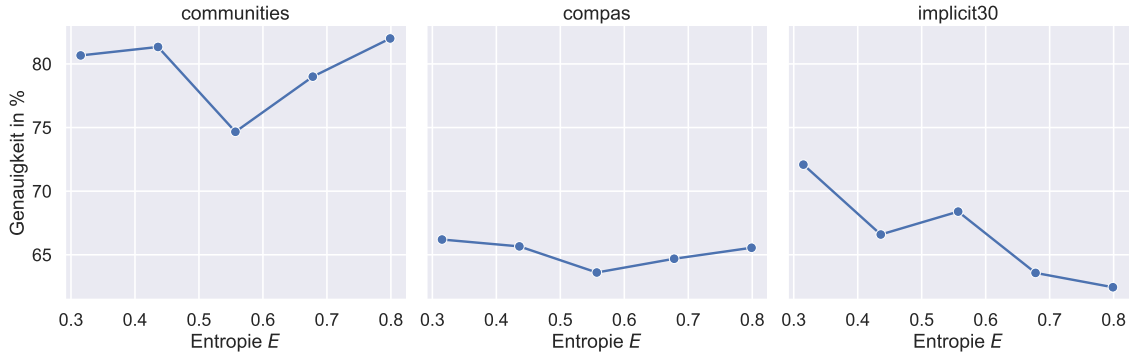


Abbildung 4.8: Zusammenhang zwischen Diversität und Genauigkeit von Random-Forest in FALCC auf verschiedenen Datensätzen.

Abbildung 4.8 bildet die Entropiewerte in Relation zu Genauigkeit ab. In diesen Ergebnissen ist kein allgemeiner Trend oder Zusammenhang zwischen Modell-Diversität und Genauigkeit über die Datensätze hinweg zu erkennen. Stattdessen sprechen diese Messungen für einen vom Datensatz abhängigen Verlauf der Genauigkeit.

5 Zusammenfassung und Ausblick

In dieser Bachelorarbeit wurden automatisierte Methoden zur Maximierung der Modell-Diversität mit Hyperparameter-Optimierung vorgestellt, und in das FALCC-Framework implementiert. Die Hyperparameter-Optimierung auf Diversität eines Modell-Ensembles hat sich für sehr effektiv herausgestellt. Besonders war das große Spektrum an Diversität von Random-Forest-Ensembles, welches bereits mit einem einfachen Suchraum entstand. Adaboost hingegen erreichte ausschließlich ein kleineres Spektrum an Diversität, dafür aber mit höheren Werten. Grund dafür ist das iterative Vorgehen, bei dem Fehlklassifikationen aus vorherigen Schritten korrigiert werden, was die Entropie-Metrik als Maß für die Diversität innerhalb des Ensembles besonders positiv beeinflusst.

Die Rolle von hoher Diversität in Modell-Ensembles für faire Klassifikationen hat sich als bedeutend herausgestellt. Im Vergleich zu AdaptedAdaboost mit allgemein niedrigerer Diversität, haben Random-Forest und Adaboost lokal, sowie auch global und individuell fairere Klassifikationen getroffen. Allerdings zeigten die Messungen für Genauigkeit, dass mit höherer Fairness sich die Genauigkeit der Klassifikationen verschlechterte, insbesondere bei Random-Forests. Daher sollte die Maximierung der Diversität nicht das einzige Kriterium für lokal faire Klassifikationen sein. Dennoch wurde das anfangs gesetzte Ziel dieser Bachelorarbeit, mit dieser Lösung erreicht und die zugehörigen fundamentalen Fragen beantwortet.

Ausblick

Ein Ziel, das weiter bestehen bleibt, ist es, den Trade-off zwischen Fairness und Genauigkeit weiter zu reduzieren. Für zukünftige Arbeit wäre der Ansatz, komplexere Basismodelle wie Entscheidungsbäume in der Generierung von optimierten Modell-Ensembles zu verwenden, besonders zielführend. Da der FALCC-Algorithmus pro Region lediglich ein Basismodell für die Klassifikationen verwendet, wäre ein komplexeres Basismodell für die Genauigkeit höchstwahrscheinlich fördernd. Außerdem wäre es interessant anstelle von Random-Forest und Adaboost ein Ensemble-Verfahren zu optimieren, welches bereits während des Trainings Fairness der Klassifikationen einbindet (z. B. [IN19]). Dadurch könnte die Fairness von Klassifikationen noch eher gewährleistet werden als mit traditionellen Ensemble-Verfahren.

Literaturverzeichnis

[BB12] J. Bergstra, Y. Bengio. „Random Search for Hyper-Parameter Optimization“. In: *J. Mach. Learn. Res.* 13.null (Feb. 2012), S. 281–305. ISSN: 1532-4435. URL: https://dl.acm.org/doi/abs/10.5555/2188385.2188395 (zitiert auf S. 20).

[BBBK11] J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl. „Algorithms for Hyper-Parameter Optimization“. In: *Advances in Neural Information Processing Systems*. Hrsg. von J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger. Bd. 24. Curran Associates, Inc., 2011. URL: https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf (zitiert auf S. 20).

[BHJ+18] R. Berk, H. Heidari, S. Jabbari, M. Kearns, A. Roth. „Fairness in Criminal Justice Risk Assessments: The State of the Art“. In: *Sociological Methods Research* 50.1 (Juli 2018), S. 3–44. DOI: 10.1177/0049124118782533 (zitiert auf S. 22).

[Bre01] L. Breiman. „Random Forests“. In: *Machine Learning* 45.1 (2001), S. 5–32. ISSN: 0885-6125. DOI: 10.1023/a:1010933404324 (zitiert auf S. 15, 29, 32).

[Bre96] L. Breiman. „Bagging predictors“. In: *Machine Learning* 24.2 (Aug. 1996), S. 123–140. DOI: 10.1007/bf00058655 (zitiert auf S. 15).

[Bur19] U. C. Bureau. *US Census Demographic Data*. 2019. URL: https://www.kaggle.com/datasets/muonneutrino/us-census-demographic-data (zitiert auf S. 38).

[CC00] P. Cunningham, J. Carney. „Diversity versus quality in classification ensembles based on feature selection“. In: *Machine Learning: ECML 2000: 11th European Conference on Machine Learning Barcelona, Catalonia, Spain, May 31–June 2, 2000 Proceedings 11*. Springer. Springer Berlin Heidelberg, 2000, S. 109–116. DOI: 10.1007/3-540-45164-1_12 (zitiert auf S. 18).

[DCGC19] N. DeCastro-Garcıa, Á. L. M. Castañeda, D. E. Garcıa, M. V. Carriegos. „Effect of the Sampling of a Dataset in the Hyperparameter Optimization Phase over the Efficiency of a Machine Learning Algorithm“. In: *Complexity* 2019 (Feb. 2019), S. 1–16. DOI: 10.1155/2019/6278908 (zitiert auf S. 20).

[DG17] D. Dua, C. Graff. *Adult Data Set. UCI Machine Learning Repository*. 2017. URL: https://archive.ics.uci.edu/ml/datasets/adult (zitiert auf S. 38).

[DHP+12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel. „Fairness through awareness“. In: *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, Jan. 2012. DOI: 10.1145/2090236.2090255 (zitiert auf S. 22, 23).

[dOL17] B. d'Alessandro, C. O'Neil, T. LaGatta. „Conscientious Classification: A Data Scientist's Guide to Discrimination-Aware Classification“. In: *Big Data* 5.2 (Juni 2017), S. 120–134. DOI: 10.1089/big.2016.0048 (zitiert auf S. 24).

[DYC+19] X. Dong, Z. Yu, W. Cao, Y. Shi, Q. Ma. „A survey on ensemble learning“. In: *Frontiers of Computer Science* 14.2 (Aug. 2019), S. 241–258. DOI: 10.1007/s11704-019-8208-z (zitiert auf S. 14).

[EFH+13] K. Eggensperger, M. Feurer, F. Hutter, J. Bergstra, J. Snoek, H. Hoos, K. Leyton-Brown et al. „Towards an empirical foundation for assessing bayesian optimization of hyperparameters“. In: *NIPS workshop on Bayesian Optimization in Theory and Practice*. Bd. 10. 3. 2013. URL: https://www.cs.ubc.ca/~hoos/Publ/EggEtAl13.pdf (zitiert auf S. 21).

[EMS19] R. Elshawi, M. Maher, S. Sakr. „Automated Machine Learning: State-of-The-Art and Open Challenges“. In: *CoRR* (2019). DOI: 10.48550/ARXIV.1906.02287 (zitiert auf S. 19).

[FFM+15] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, S. Venkatasubramanian. „Certifying and Removing Disparate Impact“. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, Aug. 2015. DOI: 10.1145/2783258.2783311 (zitiert auf S. 24).

[FS97] Y. Freund, R. E. Schapire. „A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting“. In: *Journal of Computer and System Sciences* 55.1 (Aug. 1997), S. 119–139. DOI: 10.1006/jcss.1997.1504 (zitiert auf S. 15, 29).

[FSC+19] M. Fernández-Delgado, M. Sirsat, E. Cernadas, S. Alawadi, S. Barro, M. Febrero-Bande. „An extensive experimental survey of regression methods“. In: *Neural Networks* 111 (März 2019), S. 11–34. DOI: 10.1016/j.neunet.2018.12.010 (zitiert auf S. 13).

[HD13] S. Hajian, J. Domingo-Ferrer. „A Methodology for Direct and Indirect Discrimination Prevention in Data Mining“. In: *IEEE Transactions on Knowledge and Data Engineering* 25.7 (Juli 2013), S. 1445–1459. DOI: 10.1109/tkde.2012.72 (zitiert auf S. 24).

[HPPS16] M. Hardt, E. Price, E. Price, N. Srebro. „Equality of Opportunity in Supervised Learning“. In: *Advances in Neural Information Processing Systems*. Hrsg. von D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, R. Garnett. Bd. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/file/9d2682367c3935defcb1f9e247a97c0d-Paper.pdf (zitiert auf S. 22, 24).

[HS90] L. K. Hansen, P. Salamon. „Neural network ensembles“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.10 (1990), S. 993–1001. DOI: 10.1109/34.58871 (zitiert auf S. 14).

[IN19] V. Iosifidis, E. Ntoutsi. „AdaFair“. In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. ACM, Nov. 2019. DOI: 10.1145/3357384.3357974 (zitiert auf S. 47).

[KC11] F. Kamiran, T. Calders. „Data preprocessing techniques for classification without discrimination“. In: *Knowledge and Information Systems* 33.1 (Dez. 2011), S. 1–33. DOI: 10.1007/s10115-011-0463-8 (zitiert auf S. 24).

[KLRS17] M. J. Kusner, J. Loftus, C. Russell, R. Silva. „Counterfactual Fairness“. In: *Advances in Neural Information Processing Systems*. Hrsg. von I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett. Bd. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf (zitiert auf S. 23).

[KW03] L. I. Kuncheva, C. J. Whitaker. „Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy“. In: *Machine learning* 51.2 (2003), S. 181. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2004.08.019 (zitiert auf S. 16, 18, 31).

[LH23] *FALCC: Efficiently performing locally fair and accurate classifications*. In progress, 2023 (zitiert auf S. 11, 25, 26, 35, 39, 40, 42).

[LI16] K. Lum, W. Isaac. „To Predict and Serve?“ In: *Significance* 13.5 (Okt. 2016), S. 14–19. DOI: 10.1111/j.1740-9713.2016.00960.x (zitiert auf S. 11).

[LOH21] N. Lässig, S. Oppold, M. Herschel. „Using FALCES against bias in automated decisions by integrating fairness in dynamic model ensembles“. en. In: (2021). DOI: 10.18420/BTW2021-08 (zitiert auf S. 22, 24).

[MMS+21] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan. „A Survey on Bias and Fairness in Machine Learning“. In: *ACM Computing Surveys* 54.6 (Juli 2021), S. 1–35. DOI: 10.1145/3457607 (zitiert auf S. 11, 22, 24).

[Nog–] F. Nogueira. *Bayesian Optimization: Open source constrained global optimization tool for Python*. 2014–. URL: https://github.com/fmfn/BayesianOptimization (zitiert auf S. 29).

[OPVM19] Z. Obermeyer, B. Powers, C. Vogeli, S. Mullainathan. „Dissecting racial bias in an algorithm used to manage the health of populations“. In: *Science* 366.6464 (Okt. 2019), S. 447–453. DOI: 10.1126/science.aax2342 (zitiert auf S. 11).

[Pro17] ProPublica. *COMPASS Recidivism Racial Bias*. 2017. URL: https://www.kaggle.com/datasets/danofer/compass (zitiert auf S. 38).

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay. „Scikit-Learn: Machine Learning in Python“. In: *J. Mach. Learn. Res.* 12.null (Nov. 2011), S. 2825–2830. ISSN: 1532-4435. URL: https://dl.acm.org/doi/abs/10.5555/1953048.2078195 (zitiert auf S. 29).

[Qui86] J. R. Quinlan. „Induction of decision trees“. In: *Machine Learning* 1.1 (März 1986), S. 81–106. DOI: 10.1007/bf00116251 (zitiert auf S. 13, 19).

[RB02] M. Redmond, A. Baveja. „A data-driven software tool for enabling cooperative information sharing among police departments“. In: *European Journal of Operational Research* 141.3 (Sep. 2002), S. 660–678. DOI: 10.1016/s0377-2217(01)00264-8 (zitiert auf S. 38).

[Rok09] L. Rokach. „Ensemble-based classifiers“. In: *Artificial Intelligence Review* 33.1-2 (Nov. 2009), S. 1–39. DOI: 10.1007/s10462-009-9124-7 (zitiert auf S. 16).

[Sch90] R. E. Schapire. „The strength of weak learnability“. In: *Machine Learning* 5.2 (Juni 1990), S. 197–227. DOI: 10.1007/bf00116037 (zitiert auf S. 15).

[Ska96] D. B. Skalak. „The sources of increased accuracy for two proposed boosting algorithms“. In: *Proc. American Association for Artificial Intelligence, AAAI-96, Integrating Multiple Learned Models Workshop*. Bd. 1129. Citeseer. 1996, S. 1133. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=fa8115f79d8b951b71c964fc0401d5a716d516ec (zitiert auf S. 17).

[SR18] O. Sagi, L. Rokach. „Ensemble learning: A survey“. In: *WIREs Data Mining and Knowledge Discovery* 8.4 (Feb. 2018). DOI: 10.1002/widm.1249 (zitiert auf S. 14).

[XW05] R. Xu, D. WunschII. „Survey of Clustering Algorithms“. In: *IEEE Transactions on Neural Networks* 16.3 (Mai 2005), S. 645–678. DOI: 10.1109/tnn.2005.845141 (zitiert auf S. 25).

[YS20] L. Yang, A. Shami. „On hyperparameter optimization of machine learning algorithms: Theory and practice“. In: *Neurocomputing* 415 (2020), S. 295–316. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.07.061 (zitiert auf S. 19).

[Yul00] G. U. Yule. „VII. On the association of attributes in statistics: with illustrations from the material of the childhood society, &c“. In: *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character* 194.252-261 (Jan. 1900), S. 257–319. ISSN: 0264-3952. DOI: 10.1098/rsta.1900.0019 (zitiert auf S. 17).

[ZLM18] B. H. Zhang, B. Lemoine, M. Mitchell. „Mitigating Unwanted Biases with Adversarial Learning“. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*. ACM, Dez. 2018. DOI: 10.1145/3278721.3278779 (zitiert auf S. 24).

[ZVRG17] M. B. Zafar, I. Valera, M. G. Rodriguez, K. P. Gummadi. „Fairness constraints: Mechanisms for fair classification“. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. PMLR. 2017, S. 962–970. URL: http://proceedings.mlr.press/v54/zafar17a/zafar17a.pdf (zitiert auf S. 24).

[ZWS+13] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork. „Learning Fair Representations“. In: *Proceedings of the 30th International Conference on Machine Learning*. Hrsg. von S. Dasgupta, D. McAllester. Bd. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, S. 325–333. URL: https://proceedings.mlr.press/v28/zemel13.html (zitiert auf S. 23).

Alle URLs wurden zuletzt am 24. 04. 2023 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift