Institut für Parallele und Verteilte Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit Nr. 65

Kooperative Vorhersage der minimalen Anwendungsausführungszeit

Julian Kuhn

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. rer. nat. Dr. h. c. Kurt Rothermel
Betreuer/in:	Dipl.-Inf. Florian Berg
Beginn am:	5. November 2015
Beendet am:	6. Mai 2016
CR-Nummer:	C.2.4,C.4,D.2.8,D.4.8,I.2.6

Kurzfassung

Code Offloading Frameworks verbessern durch Auslagern von Programmteilen - auch *Offloadingkandidaten* genannt - auf Server die Leistung oder den Energieverbrauch von Geräten mit limitierten Ressourcen. Offloadingkandidaten werden dann ausgelagert, wenn mit Inbetrachtnahme der Übertragung des Kandidaten eine Einsparung im Vergleich zur rein lokalen Ausführung vorliegt.
Die Entscheidung, ob Offloading stattfindet, hängt stark von der Ausführungszeit des Kandidaten ab. Im Fall von Methoden kann die Ausführungszeit je nach aktueller Parameterkonfiguration stark variieren. Da es in vielen Fällen unpraktikabel ist, für jede Parameterkombination Aufzeichnungen durchzuführen, ist die Verwendung von einfachen, historienbasierten Modellen zur Bestimmung der Ausführungszeit ungeeignet. Eine möglichst genaue Angabe der Ausführungszeit wird aber benötigt, um die Offloadingentscheidung korrekt zu treffen.

Ziel der Arbeit war, die Vorhersage von Ausführungszeiten mit Hilfe von Machine Learning Modellen anhand verschiedener Testanwendungen- und Szenarien im Kontext des Code Offloadings zu untersuchen.
Außerdem wurde ein kooperativer Systementwurf vorgestellt und implementiert, der zur Verwaltung von Datensätzen, Vorhersagemodellen und deren Erstellung, sowie zur Vorhersage von Ausführungszeiten verwendet werden kann. Der Entwurf erweitert dabei bestehende Offloadingframeworks.
Es konnte festgestellt werden, dass sich Machine Learning Algorithmen zur Vorhersage und insbesondere zum Verbessern der Offloadingentscheidung eignen.

Inhaltsverzeichnis

1. Einleitung **11**
- 1.1. Motivation . . . 11
- 1.2. Problemstellung und Ziele der Arbeit . . . 12
- 1.3. Verwandte Arbeiten . . . 14

2. Code Offloading **17**
- 2.1. Einleitung . . . 17
- 2.2. Grundlagen . . . 17
- 2.3. Zusammenfassung . . . 20

3. Machine Learning **21**
- 3.1. Einleitung . . . 21
- 3.2. Definitionen . . . 21
- 3.3. Methoden zum Einlernen,Validieren,Optimieren und Bewerten von Modellen . 24
- 3.4. Regressionsalgorithmen . . . 26
- 3.5. Zusammenfassung . . . 39

4. Kooperativer Systementwurf **41**
- 4.1. Einleitung . . . 41
- 4.2. Anforderungen an die Komponenten . . . 43
- 4.3. Metriken für die Ausführungszeit . . . 52
- 4.4. Entwurf . . . 53
- 4.5. Zusammenfassung . . . 61

5. Implementierung **63**
- 5.1. Einleitung . . . 63
- 5.2. Verwendete Frameworks und Kommunikationsmechanismen . . . 65
- 5.3. Implementierung: Nachrichten . . . 67
- 5.4. Implementierung: Machine Learning . . . 67
- 5.5. Implementierung: Persistente Datenhaltung . . . 69
- 5.6. Implementierung: Komponenten . . . 71
- 5.7. Zusammenfassung . . . 74

6. Evaluation **75**
- 6.1. Einleitung . . . 75

6.2. Bewertungskriterien und Vorgehen . . . 76
6.3. Testszenarien . . . 80
6.4. Testanwendungen . . . 82
6.5. Testhardware . . . 88
6.6. Testergebnisse: Machine Learning . . . 89
6.7. Testergebnisse: Leistungsfaktor, Speicherbedarf und Zeit zum Erstellen . . . 95
6.8. Testergebnisse: Offloadingentscheidung . . . 99
6.9. Schlussfolgerung . . . 104

7. Zusammenfassung und Ausblick 109

Literaturverzeichnis 113

A. Testfälle der Text-To-Speech Anwendung 119

B. Machine Learning Ergebnisse 121

C. Offloadingentscheidung: Rauschreduzierung 155
C.1. Energieszenario . . . 155
C.2. Leistungsszenario . . . 156

D. Offloadingentscheidung: Gesichtserkennung 159
D.1. Energieszenario . . . 159
D.2. Leistungsszenario . . . 160

E. Offloadingentscheidung: Text-To-Speech 163
E.1. Energieszenario . . . 163
E.2. Leistungsszenario . . . 164

F. Offloadingentscheidung: Schach(alle Parameter) 167
F.1. Energieszenario . . . 167
F.2. Leistungsszenario . . . 168

G. Offloadingentscheidung: Schach (difficulty = 1) 171
G.1. Energieszenario . . . 171
G.2. Leistungsszenario . . . 172

H. Offloadingentscheidung: Schach (difficulty = 2) 175
H.1. Energieszenario . . . 175
H.2. Leistungsszenario . . . 176

I. Offloadingentscheidung: Schach (difficulty = 3) 179
I.1. Energieszenario . . . 179
I.2. Leistungsszenario . . . 180

J. Leistungsfaktor: Rohdaten **183**
J.1. Schach (difficulty 1) 183
J.2. Schach (difficulty 3) 197

Abbildungsverzeichnis

2.1. Identifikation der Offloadingkandidaten. 18
2.2. Veranschaulichung der Offloadingentscheidung 19
2.3. Sequenzdiagramm: Stellvertretende Ausführung eines Offloadingkandidaten auf dem Server. 20

3.1. Allgemeiner Ablauf der Erstellung eines Vorhersagemodelles. 22
3.2. Vorhersage der Zielvariablen anhand eines unbekannten Samples. 23
3.3. Beispiel einer Mittelgerade eines Linear Regression Modelles. 27
3.4. Beispiel einer Vorhersage eines Polynomial Regression Modelles. 29
3.5. SVM: Linear Separierbarer Raum 30
3.6. Darstellung des Randes bei der Support Vector Regression mit Radial Basis Function(RBF) Kernel. 35
3.7. Aufbau eines zwei-Schichtigen feedforward Netzes. 36
3.8. Aufbau des RBF Network. 37
3.9. Beispiel: Decision Tree 38

4.1. Offloading: Die Entscheidung soll mit Hilfe einer Vorhersage (rot) verbessert werden. 41
4.2. Entwurf: Hauptkomponenten 43
4.3. Zusammenführung von Aufzeichnungen unterschiedlicher Architekturen anhand eines Leistungsfaktors. 53
4.4. Komponente “Server”. 54
4.5. Komponente “Client”. 55
4.6. Struktur der Komponente “Controller” 56
4.7. Alternatives Design für die Controller-Komponente. 58
4.8. Sequenzdiagramm: Erstellen der Vorhersage auf Serverseite und updaten der Clients. 58
4.9. Sequenzdiagramm: Registrierung eines Clients ohne lokal gespeichertes Modell. 59
4.10. Sequenzdiagramm: Registrierung eines Clients mit lokal gespeichertem Modell. 60
4.11. Sequenzdiagramm: Erstellen der Vorhersage auf Serverseite und updaten der Clients. 60

5.1. Integration der Client-Komponente in das Offloadingframework von Berg. . . 64
5.2. Implementierung: Nachrichten 67

5.3. Implementierung der Klasse “Regression”, welche die Erstellung der Vorhersagemodelle ermöglicht. . . . 68
5.4. Implementierung der persistenten Datenhaltung. . . . 70
5.5. Entity-Relationship Modell der SQLite Datenbank. . . . 70
5.6. Implementierung der Komponente “Client”. . . . 72
5.7. Implementierung der Komponente “Controller”. . . . 73
5.8. Implementierung der Komponente “Server”. . . . 74

6.1. **Schach mit difficulty 1**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 89
6.2. **Schach mit difficulty 2**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 90
6.3. **Schach mit difficulty 3**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 90
6.4. **Schach (alle difficulty parameter)**:Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 91
6.5. **Gesichtserkennung**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 93
6.6. **Rauschreduzierung**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 94
6.7. **Text-To-Speech**: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples. . . . 95
6.8. Vergleich der Größen der Vorhersagemodelle in Megabyte. . . . 96
6.9. Dauer zum Erstellen der Modelle der Schachanwendung (alle *difficulty*-Werte) und der Rauschreduzierung. . . . 99

Tabellenverzeichnis

5.1. Zusammenfassung der verwendeten Algorithmen und der verwendeten Parameter. . . . 69

6.1. Annahmen für den Energieverbrauch 78
6.2. Testkonfiguration 1: LTE mit mittleren Datenraten. . . . 80
6.3. Testkonfiguration 2: 3G mit mittleren Datenraten. . . . 80
6.4. Testkonfiguration 3: 3G mit niedrigen Datenraten. . . . 81
6.5. Testkonfiguration 4: Wifi (802.11 g) mit niedrigen Datenraten. . . . 81
6.6. Testkonfiguration 5: Wifi (802.11 g) mit hohen Datenraten. . . . 81

6.7. Zusammenfassung: Anzahl der Parameter und Parameterbeschreibungen sowie die Datensatzgrößen. . . . 87
6.8. Hardware für die Testfallerstellung. . . . 88
6.9. Hardware für die Testfallerstellung der Schachanwendung. . . . 88
6.10. Hardware für die Testfallerstellung der Schachanwendung zur Berechnung eines Leistungsfaktors. . . . 88
6.11. Schachanwendung: Beste Modelle für den jeweils kompletten Datensatz mit verschiedene Datensatzaufteilungen nach Parameter “difficulty”. . . . 92
6.12. Vergleich des Speicherbedarfs zwischen komprimierten (LZMA) und unkomprimierten SVR Vorhersagemodellen der Schachanwendung. . . . 97
6.13. Leistungsfaktorberechnung 98
6.14. Energieszenario für die Schachanwendung. . . . 100
6.15. Leistungsszenario für die Schachanwendung. . . . 100
6.16. Leistungsszenario für die Schachanwendung mit aufgeteiltem “difficulty”-Parameter. . . . 100
6.17. Energieszenario für die Schachanwendung mit aufgeteiltem “difficulty”-Parameter. . . . 101
6.18. Leistungsszenario der Rauschunterdrückung. . . . 101
6.19. Text-To-Speech: Energieszenario. . . . 102
6.20. Text-To-Speech: Leistungsszenario. . . . 102
6.21. Energieverbrauch der Übertragung von Vorhersagemodellen 103
6.22. Tatsächliche Einsparung: 6300 Ausführungen 103
6.23. Tatsächliche Einsparung: 63Ausführungen 104
6.24. Zusammenfassung: Beste Algorithmen pro Testanwendung. . . . 105

Verzeichnis der Listings

5.1. Erstellung eines Linear Regression Vorhersagemodelles mit Hilfe des Weka Frameworks. . . . 68

6.1. Search Methode, welche die Alpha-Beta Heuristik verwendet 83
6.2. Offloadingkandidat der Gesichtserkennung. . . . 84
6.3. Offloadingkandidat der Rauschreduzierung. . . . 86

1. Einleitung

1.1. Motivation

Smartphones und andere Mobilgeräte sind nicht mehr aus dem täglichen Leben wegzudenken. Dabei werden die Geräte immer leistungsfähiger. Der limitierende Faktor bei dieser Geräteklasse ist meist die Energiekapazität. Die Verwendung von rechenintensiven Applikationen, wie beispielsweise Bildbearbeitungsprogramme, kann schnell dazu führen, dass der Akku des Mobilgerätes wieder geladen werden muss. Nutzer sind deshalb darauf bedacht, das Gerät so energieschonend wie möglich zu verwenden. Andererseits kann der Fall auftreten, dass Berechnungsergebnisse so zeitnah wie möglich erhalten werden wollen, ohne auf den Energieverbrauch zu achten.
Die Leistungsfähigkeit kann verbessert werden, indem *Code Offloading* [KLLB13; FHT+15] verwendet wird. Beim Code Offloading werden Berechnungen auf Server ausgelagert, die leistungsstark sind und keine Energieeinschränkungen besitzen.
Code Offloading Systeme unterscheiden sich dabei von klassischen Client-Server Systemen. Bei Client-Server Anwendungen werden Programmteile während der Entwicklung der Anwendung bereits als feste auf Server- oder Clientseite auszuführende Teile definiert.
Offloading Systeme haben zum Ziel, Anwendungen, die normalerweise lokal ausgeführt werden, durch das Auslagern von Programmteilen zu verbessern.
In der Regel wird eine Anwendung durch das Offloading entweder in Hinsicht auf die Leistung oder den Energieverbrauch optimiert. Dazu müssen die Programmteile manuell oder automatisch identifiziert werden, die für das Auslagern in Frage kommen. Diese Programmteile werden im Folgenden als *Offloadingkandidaten* bezeichnet. Ein Beispiel für einen Offloadingkandidat wäre eine Methode oder ein Thread.
Das Auslagern eines Offloadingkandidaten lohnt sich dann, wenn Ausführungszeit oder Energie durch Auslagern auf Server mit Inbetrachtnahme des Übertragungsweges und der Wartezeit eingespart werden kann.

Je nach Charakteristik des Offloadingkandidaten kann der Fall auftreten, dass sich eine Auslagerung nicht lohnt und sich diese sonst unter Umständen negativ auf die Effizienz des Systems auswirken kann. Sind die Offloadingkandidaten parametrisiert und wirken sich die Parameter auf die Ausführungszeit aus, muss die Ausführungszeit des Kandidaten mit der aktuellen Parametrisierung auch in die Entscheidung mit einfließen, ob geoffloaded wird. Diese Entscheidung wird folgend als *Offloadingentscheidung* bezeichnet. Das Wissen über

die Ausführungszeit eines Offloadingkandidaten ist also ein entscheidender Faktor für eine korrekte Offloadingentscheidung.

Wort-Case Execution Time(WCET)[LGZ+09] Analysen können verwendet werden, um die Laufzeit von Programmteilen zu bestimmen. Ausgehend vom längsten Ausführungspfad einer Methode mit einer bestimmten Parameterkonfiguration werden abhängig von der Rechnerarchitektur die CPU-Zyklen pro Maschinencode gezählt und aufaddiert. Die Summe ergibt die WCET. Ändert sich die Architektur, muss der WCET neu berechnet werden. Beeinflussen Parameter die Ausführungszeit einer Methode müsste für jeden möglichen Programmpfad eine WCET-Analyse gemacht werden.

Praktisch ist es jedoch oft unmöglich, für jedes mögliche Ausführungsszenario die Ausführungszeit zu bestimmen. Beispielsweise ist ein Algorithmus zum Berechnung eines Schachzuges abhängig von der aktuellen Aufstellung auf dem Schachbrett. Allein die Anzahl an möglichen Stellungen wird auf 10^{42} geschätzt [Sha50].

Diese zwei Nachteile machen die WCET-Analyse unpraktikabel. Viele Offloading Frameworks wie beispielsweise MAUI[CBC+10] setzen auf historienbasierte Ansätze, bei der Datenbanken für Ausführungsszenarien erstellt werden [dCP16]. Bei Offloadingkandidaten, deren Ausführungszeiten nicht konstant sind, stoßen historienbasierte Ansätze schnell an ihre Grenzen.

Eine andere Möglichkeit ist der Einsatz Schätzverfahren, wie Machine Learning Algorithmen [WF05]. Anhand einer Teilmenge der möglichen Ausführungsszenarien, bei der die Ausführungszeiten bekannt sind, können über Machine Learning Algorithmen Modelle erstellt werden, über die näherungsweise Aussagen über die Ausführungszeit für unbekannte Ausführungsszenarien getroffen werden können.
Die Ausführungszeiten, die für die Erstellung solcher Modelle verwendet werden, sollten dazu möglichst minimal sein. In realen Systemen wird die Ausführungszeit von Faktoren wie der aktuellen Prozessorauslastung oder dem Schedulingverhalten des Betriebssystems beeinflusst. Eine minimale Ausführungszeit soll also in diesem Kontext als eine Ausführungszeit einer Methode mit minimalem Einfluss Betriebssystems zu verstehen sein.

1.2. Problemstellung und Ziele der Arbeit

Die Energie - oder Zeiteinsparung, die die Frameworks bewirken, ist maßgeblich von der korrekten Offloadingentscheidung abhängig. Werden Methoden ausgelagert, deren lokale Ausführung effizienter wäre, wirkt das Offloading kontraproduktiv und verschlechtert die Effizienz. Die Offloadingentscheidung ist also sehr stark an die angenommene Zeit für eine bestimmte Parameterkonfiguration eines Offloadingkandidaten gebunden.
Bestehende Offloading Ansätze (vgl. Kapitel 1.3: *Verwandte Arbeiten*) bieten lediglich simple Modelle zum Vorhersagen der Ausführungszeiten oder betrachten nicht, wie die Datensätze für die Modelle erstellt werden können.

In dieser Arbeit wird untersucht, in wie weit Machine Learning Methoden [Bis06; WF05] verwendet werden können, um Vorhersagemodelle zu erstellen, die die Ausführungszeit abschätzen, um mit Hilfe dieser die Offloadingentscheidung zu treffen.

Die Machine Learning Methoden (vgl. Kapitel 3: *Machine Learning*) benötigen dabei zum Erstellen der Vorhersagemodelle Datensätze mit Testfällen, bei denen die Ergebnisse vorgegeben sind. Im Falle des Offloadings bestehen die Testfälle aus den Parametern des Offloadingkandidaten und der Ausführungszeit, die für diese Parameterkonfiguration aufgezeichnet wurde.
Praktisch ist anzunehmen, dass eine Anwendung, die für das Offloading verwendet werden soll, von vielen Anwendern benutzt wird. Beispiele dafür wären rechenintensive Anwendungen wie ein Schachspiel oder Bildbearbeitungsprogramme.
Eine Idee, ohne von Hand Ausführungsszenarios zu erstellen, ist diese während der normalen Verwendung der Anwendung aufzuzeichnen. Damit die Vorhersagequalität ausreichend gut ist, sind oft große Datensätze nötig[KZP07, S. 14]. Deswegen ist es unpraktikabel, ausschließlich die lokal aufgezeichneten Ausführungen zur Erstellung der Modelle zu verwenden. Außerdem ist das Erstellen der Modelle selbst sehr zeitaufwändig, wie Kapitel 6 zeigt.

Hierfür soll ein kooperativer Systementwurf vorgestellt werden, der es ermöglicht, Datensätze an einer zentralen Stelle zusammenzuführen. Da praktisch unterschiedlichste Architekturen zum Einsatz kommen und die Ausführungszeiten für ein Ausführungsszenario dadurch stark variieren können, soll außerdem untersucht werden, inwiefern eine Ausführungszeit von der einen auf eine andere Architektur übertragen werden kann.

Zusammenfassend stellt diese Arbeit folgende Punkte vor:

- *Verbesserung der Offloadingentscheidung durch Vorhersage der Ausführungszeiten*: Die Entscheidung, ob geoffloaded wird, hängt stark vom Wissen um die Ausführungszeit einer Methode in Abhängigkeit ihrer aktuellen Parametrisierung ab. In dieser Arbeit werden verschiedene Machine Learning Ansätze verwendet und bewertet, in wie weit sie zur Verbesserung der Entscheidung beitragen können.
- *Kooperative Systementwurf*: Jeder Anwender einer Anwendung, die Code-Offloading nutzt, soll dazu beitragen können, das Vorhersagemodell für die Ausführungszeit zu verbessern. Dabei sollen die aufgezeichneten Datensätze zentral gesammelt und für die Erstellung eines Vorhersagemodells verwendet werden. Hierzu wird ein Systementwurf vorgestellt, der in bestehende Offloadingframeworks integriert werden kann.
- *Prototypische Implementierung*: Ausgehend vom Systementwurf wird eine prototypische Implementierung durchgeführt.
- *Evaluation*: Zusätzlich soll die Größe der Vorhersagemodelle, die Zeit zum Erstellen der Modelle, sowie die mögliche Verwendung eines Leistungsfaktors untersucht werden. Ebenso soll betrachtet werden, wie sich die Übertragung von Vorhersagemodellen auf die Gesamteffizienz auswirkt.

1.3. Verwandte Arbeiten

Im Folgenden werden einige Offloadingframeworks beschrieben. Der Fokus wird dabei auf die Offloadingentscheidung gelegt.

Narayanan et al. [NFS00] haben bereits 2000 untersucht inwiefern Machine Learning dafür verwendet werden kann, Ausführungszeiten vorherzusagen. Verwendet wurden dabei lineare Vorhersagemodelle. Dabei kamen sie zu einem positiven Ergebnis, was die Vorhersagequalität und den Overhead durch die Vorhersage auf dem System angeht. Aussagen darüber, wie die Modelle anderen Mobilgeräten verfügbar gemacht werden können, wurden allerdings nicht gemacht.

Spectra [FPS02] ist ein 2002 entwickeltes Offloading Framework, welches über Annotationen Offloadingkandidaten identifiziert. Es werden verschiedene Profiler verwendet, welche die Netz-,CPU- und Batterienutzung überwachen. Spectra erstellt anhand von vergangenen Ausführungen lineare Regressionsmodelle, um die Ausführungszeit für die aktuelle Ausführungszeit zu bestimmen. Es wird keine Aussage über die Speicherung der Aufzeichnungen getroffen oder wie sie anderen Mobilgeräten verfügbar gemacht werden können.

MAUI [CBC+10] ist ein 2010 entwickeltes System, welches zum Offloaden von *managed code* der .NET Common Language Runtime verwendet werden kann. Das Ziel von MAUI ist hierbei, eine größtmögliche Energieersparnis durch das Offloading zu erwirken, während gleichzeitig von Anwendungsentwicklern kaum Modifikationen in den Sources verlangt werden. Es müssen lediglich Annotationen über den gewünschten Methoden eingefügt werden. Der .NET Ansatz gewährleistet eine einfache Umsetzung dank der Portabilität des Codes.
MAUI verwendet auf dem Mobilgerät einen Profiler, welcher pro Ausführung eines Offloadingkandidaten die zu übertragende Zustandsgröße, sowie die benötigte Energie und CPU-Zyklen aufzeichnet. Für den Energieverbrauch wird ein lineares Regressionsmodell abgeleitet. Im Falle der Ausführungszeit wird historienbasierter Ansatz verwendet. Die Offloadingentscheidung wird auf Serverseite getroffen.

CloneCloud [CIM+11] hat zum Ziel, Modifikationen der auszulagernden Applikationen komplett zu vermeiden. Dies wird durch eine Kombination einer statischen Analyse und einem dynamischen Profiling, welche eine automatische Partitionierung des Anwendungsprogrammes bewirkt, erreicht. Ebenso wie bei MAUI wird hier auf die Verwendung Laufzeitumgebungen wie die CLR oder Java VM gesetzt. Die automatisierte Partitionierung einer Anwendung läuft folgendermaßen ab: Der *Static Analyzer* untersucht den Ausführungscode der Anwendung und findet unter Verwendung bestimmter Einschränkungen(z. B. kann kein Code ausgelagert werden, der auf das lokale Dateisystem zugreift) mögliche Partitionierungspunkte. Der *Dynamic Profiler* zeichnet Daten für Ausführungsszenarios auf. Diese werden verwendet um ein Kostenmodell zu erstellen. Für die Ausführungsszenarios werden zufällige Eingabeparameter verwendet. Jede Ausführung wird sowohl auf dem Mobilen Gerät als auch auf dem Offloading-Server ausgeführt. Als Datenmodell wird ein sogenannter *profile tree* verwendet, der eine parametrisierte Ausführung darstellt. Das bedeutet, dass pro Ausführung zwei Profile Trees

entstehen. Für jeden Aufruf in einer Ausführung werden Ausführungskosten und Migrationskosten errechnet. Im letzten Schritt bestimmt der *Optimization Solver* Partitionen, welche die Zielfunktion, die durch das Kostenmodell hergeleitet wurde, minimieren. Die Autoren weisen selbst auf den Schwachpunkt in der Verwendung von nur wenigen Ausführungsszenarien im Dynamic Profiler hin.

Odessa [RSM+11] ist ein 2011 vorgestelltes Framework, welches adaptiv Entscheidungen zum Offloading und Parallelisieren von Anwendungen trifft. Die Autoren von Odessa kommen zum Schluss, dass durch statische Analysen die Entscheidung, welche Programmteile geoffloaded oder parallelisiert werden können, nicht getroffen werden kann. Als Grundlage nimmt Odessa Sprout [PMS+09]. Sprout ist ein stream processing System zum parallelen Ausführen von Anwendungen. Die Offloadingentscheidung wird in einem auf dem Mobilgerät laufenden *Application Profiler* anhand von einem durch historischen Ausführungen erstellten simplen Vorhersagemodell getroffen.

ThinkAir [KAH+12] ist ein Framework das Offloading auf Methodenebene ermöglicht. Die Ziele bei ThinkAir sind Skalierbarkeit und Elastizität und ermöglicht die parallele Ausführung von Methoden mit Zuhilfenahme mehrerer Virtual Machines. Im Gegensatz zu den vorherigen Ansätzen bietet ThinkAir eine API an, mit der man Applikationen für das Offloading integrieren kann. ThinkAir verwendet 3 Arten von Profilern: Einen Network Profiler, der die RTT, sowie den Datenverbrauch des ThinkAir Frameworks aufzeichnet. Dazu gibt es einen Hardware Profiler, der die CPU Auslastung, die Helligkeit des Displays und die Energieeinstellungen der Wifi und 3G Antenne aufzeichnet. Der Dritte ist der Software Profiler. Hier wird für jede auslagerbare Methode unter anderem die Ausführungszeit, die Anzahl Anweisungen und die Unteraufrufe von Methoden aufgezeichnet. Der *Execution Controller* trifft die Entscheidung, ob eine Methode ausgelagert wird oder nicht. Für die Entscheidungsfindung werden historische Daten des Energieverbrauches, der Ausführungszeit oder den Kosten einer Cloudausführung verwendet. Aus den Daten werden keine Vorhersagemodelle gebildet.

MALMOS [EFC+15] ist ein 2015 vorgestelltes Framework, welches die Offloadingentscheidung mit Hilfe von Machine Learning Algorithmen trifft. Dabei werden Klassifikatoren verwendet. Anhand eines Profilers werden die Ausführungszeiten der Ausführungen aufgezeichnet und als Grundlage für das Trainieren der Klassifikatoren verwendet. Im Gegensatz zu dieser Arbeit interpretieren die Autoren von MALMOS die Offloadingentscheidung als binäres Klassifikationsproblem. Dabei wird davon ausgegangen, dass sowohl die Modelle als auch die Ausführungsszenarien auf dem Mobilgerät erstellt werden.

Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Code Offloading führt einige Grundlagen zum Thema Offloading ein.

Kapitel 3 – Machine Learning führt Grundlagen des Machine Learnings und Regressionsalgorithmen ein.

Kapitel 4 – Kooperativer Systementwurf stellt einen in Offloadingframeworks integrierbaren Entwurf zur Verwaltung von Vorhersagemodellen und zur Vorhersage von Ausführungszeiten vor.

Kapitel 5 – Implementierung stellt eine prototypische Implementierung des Entwurfs vor.

Kapitel 6 – Evaluation bewertet die Regressionsalgorithmen anhand einer Auswahl an Testszenarien - und Anwendungen.

Kapitel 7 – Zusammenfassung und Ausblick fasst die Ergebnisse der Arbeit zusammen und stellt mögliche Ansatzpunkte für weitere Arbeiten vor.

2. Code Offloading

2.1. Einleitung

In diesem Kapitel soll zunächst eine kurze Einleitung in das Code Offloading gegeben werden. Dabei sollen die allgemeinen Konzepte des Code Offloadings mit einem Fokus auf die Offloadingentscheidung erläutert werden.

Im Allgemeinen lässt sich folgende Vorgehensweise in den meisten Frameworks beobachten [KLLB13; FHT+15]:

1. *Partitionierung des Programmes*: Das Programm wird aufgeteilt in einen lokalen Teil und einer Menge an Offloadingkandidaten.
2. *Profiling der Verbindungsqualität oder des Programmabschnittes*: Viele Frameworks führen ein Profiling der Verbindungsqualität, des Energieverbrauchs oder der Ausführungszeit durch, welche die Offloadingentscheidung beeinflussen.
3. *Offloadingentscheidung*: Abhängig vom Profiling wird die Offloadingentscheidung getroffen, also ob eine lokale oder ausgelagerte Ausführung stattfindet.
4. *Auslagern auf einen entfernten Server*: Soll ausgelagert werden, wird der für die Berechnung benötigte Zustand auf den Offloadingserver übertragen und dort ausgeführt. Anschließend werden die Ergebnisse der Berechnung wieder auf den Client übertragen.

Im nächsten Abschnitt soll näher auf diese Punkte eingegangen werden.

2.2. Grundlagen

Im Allgemeinen sind die Code Offloading Frameworks als Client-Server-Architektur konzipiert. Das Mobilgerät nimmt hierbei die Rolle des Clients ein, welcher den Offloading-Service nutzt. Der Client macht dem Server zunächst bekannt, was er auslagern möchte.

Offloading kann in unterschiedlicher Granularität durchgeführt werden[LSJ+13]:

- Programmkomponenten: Komplette Programmabschnitte werden ausgelagert.

- Thread-Ebene: Bei multithreaded Anwendungen bietet es sich an, einzelne Threads auszulagern.
- Methodenebene: Sollen einzelne Methoden ausgelagert werden, müssen Parameterwerte an den Offloading-Server übertragen werden. Der Rückgabewert der Methode wird schließlich vom Server wieder an das Mobilgerät geschickt.

Je nachdem welche Granularität im einzelnen Framework gewählt wurde, wird ein unterschiedlicher Ansatz zur Partitionierung der Anwendungen angewandt. Findet ein Offloading auf Komponentenebene statt, so wird oft eine statische Code Analyse verwendet, um mögliche Partitionierungspunkte zu identifizieren [CIM+11]. Wird auf Methodenebene Offloading gemacht, bedient man sich häufig Annotationen im Source Code, um Partitionierungspunkte zu definieren [CBC+10; BDR14b]. Abbildung 2.1 veranschaulicht diesen Prozess.

Nach der Identifikationsphase hat man eine Menge an *Offloadingkandidaten*. Ein Offloadingkandidat kann, muss aber nicht ausgelagert werden.

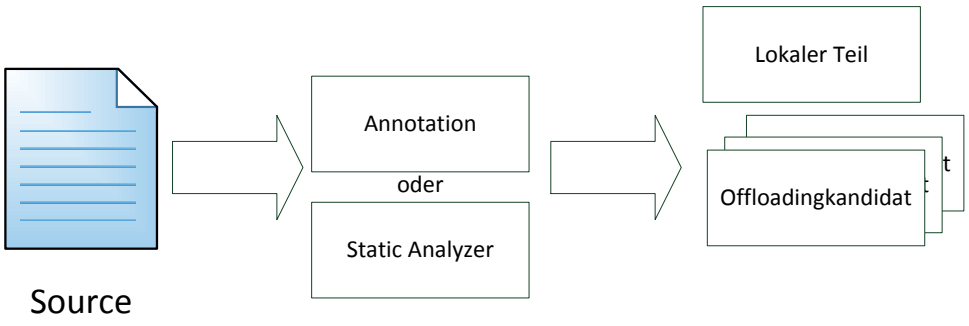


Abbildung 2.1.: Identifikation der Offloadingkandidaten.

Die Entscheidung, ob ein Offloadingkandidat ausgelagert wird, hängt von dem jeweiligen Optimierungsziel, nämlich der Leistung oder dem Energieverbrauch,ab [KLLB13].
In beiden Fällen muss ein Vergleich zwischen einer rein lokalen Ausführung und der Ausführung auf dem Offloading-Server gemacht werden.
Ebenfalls muss der Übertragungsweg der Daten betrachtet werden. Soll die Entscheidung leistungsoptimal sein, muss die Dauer der Übertragung der Ein-und Ausgabedaten und die Berechnungsdauer auf dem Server betrachtet werden. Für eine energieoptimale Entscheidung muss ein Vergleich zwischen dem Energieverbrauch für die lokale Ausführung, dem Energieverbrauch der Übertragung in beide Richtungen, sowie dem Energieverbrauch während der Wartezeit gezogen werden.

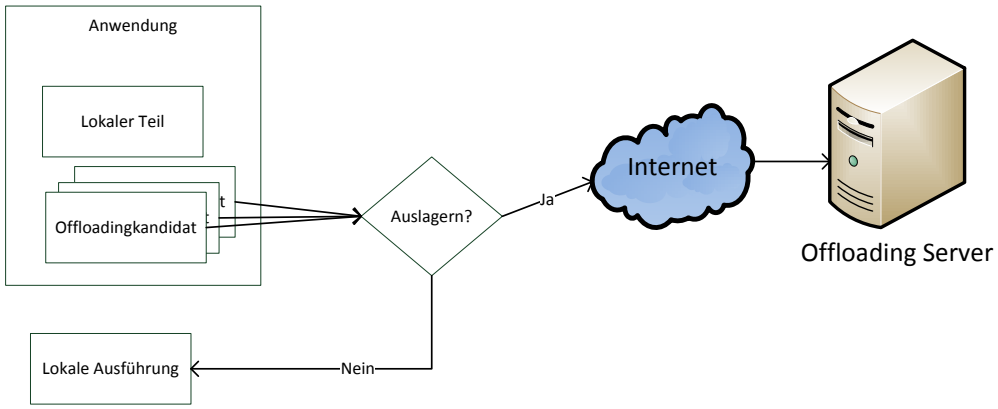


Abbildung 2.2.: Ausgehend von einer Menge an Offloadingkandidaten der Anwendung wird für jeden Offloadingkandidaten entschieden, ob eine lokale Ausführung oder eine entfernte Ausführung stattfinden soll.

Ist das Ziel eine Leistungsverbesserung, findet ein Offloading unter folgender Bedingung statt[KLLB13]:

$$\frac{w}{s_m} > \frac{d_i}{B} + \frac{w}{s_S} \tag{2.1}$$

wobei w der Code des Offloadingkandidaten ist, der ausgeführt werden muss, s_m die Geschwindigkeit des Mobilgerätes, s_S die Geschwindigkeit des Servers, d_i die Eingabedaten des Offloadingkandidaten, und B die Bandbreite ist.

Ist das Ziel eine Verbesserung des Energieverbrauches, findet ein Offloading unter folgender Bedingung statt [KLLB13]:

$$p_m * \frac{w}{s_m} > p_C * \frac{d_i}{B} + p_i * \frac{w}{s_S} \tag{2.2}$$

wobei p_m die Energie des Mobilgerätes, p_C die Energie zum Senden der Eingabedaten und p_i der Energieverbrauch bis zum Erhalt der Ergebnisse ist.

Die Entscheidung hängt stark von der Verbindungsqualität ab. Die Verbindungsqualität wird durch sogenannte *Profiler* [CBC+10; CIM+11] überwacht, welche in Frameworks wie MAUI oder CloneCloud verwendet werden. Der Energieverbrauch hängt direkt mit der benötigten Laufzeit des Offloadingkandidaten zusammen. Kennt man die Laufzeit - kann unter Annahme einer stabilen Verbindung zum Server - die Entscheidung stets korrekt getroffen werden.

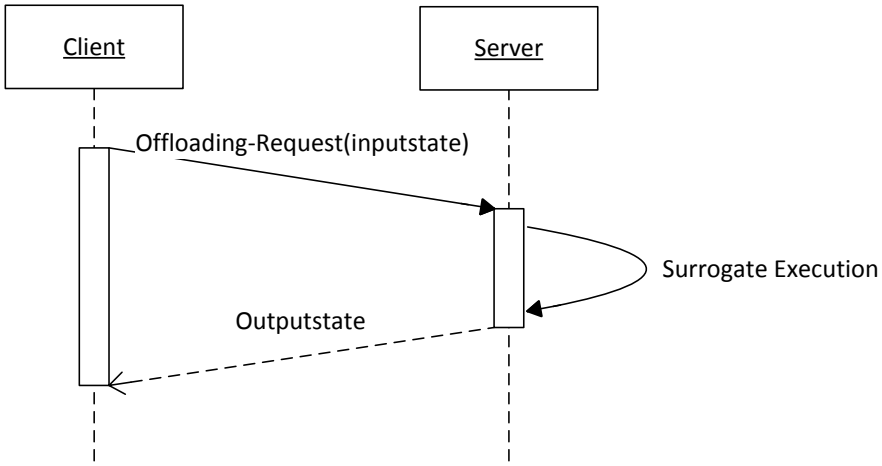


Abbildung 2.3.: Sequenzdiagramm: Stellvertretende Ausführung eines Offloadingkandidaten auf dem Server.

Abbildung 2.3 zeigt anschließend die Auslagerung des Offloadingkandidaten. Oft ist nötig einen Zustand, wie z.B. die Parameterwerte einer Methode, an den Server zu übertragen. Wurde der Zustand übertragen, führt der Offloading-Server die Berechnung stellvertretend aus. Der Ergebniszustand der Berechnung wird anschließend an den Client zurück geschickt.

2.3. Zusammenfassung

Code Offloading bietet sich an, um rechenintensive oder energieaufwändige Programmteile einer sonst lokal ausgeführten Anwendung auf Server auszulagern. Die Anwendung wird in einen lokalen Teil und eine Menge von Offloadingkandidaten aufgeteilt. Für jeden Offloadingkandidat, der ausgeführt werden soll, wird in Abhängigkeit der Verbindungsgeschwindigkeit und dem geschätzten Verhältnis der Ausführungszeit auf dem Server und dem Mobilgerät die Entscheidung getroffen, ob eine lokale Ausführung oder eine Ausführung auf dem Offloadingserver effizienter ist.
Dabei war festzustellen, dass das Wissen um die Ausführungszeit essentiell für eine korrekte Entscheidung ist. Wie bereits in der Einleitung erwähnt, soll diese Entscheidung mit Hilfe von Machine Learning Algorithmen verbessert werden. Diese werden im nächsten Kapitel vorgestellt.

3. Machine Learning

3.1. Einleitung

Unter Machine Learning versteht man eine Vielzahl von Algorithmen und Konzepten die zur Informationsgewinnung von Datensätzen genutzt werden können. Beispiele dafür sind das Finden von Mustern, Vorhersagen anhand von vergangenen Ergebnissen oder das Zuordnen von neuen Datensätzen zu Klassen.
Zunächst werden einige gängige Begriffe, Definitionen und Einordnungen eingeführt. Anschließend werden folgende, gängige Machine Learning Konzepte näher betrachtet:

- Linear Regression
- Polynomial Regression
- Support Vector Regression
- KNN Regression
- Regression Trees
- Neuronale Netze

3.2. Definitionen

Die folgenden Definitionen und Erklärungen basieren auf den Büchern von Bishop, Witten und Smola et. al [Bis06; WF05; SV08].

3.2.1. Samples und Attribute

Ein Feature-Vektor oder *Sample* kann als n-dimensionaler Vektor $x = (x_0, ..., x_n)$ formuliert werden. Ein Element des Vektors x wird als *Feature* oder *Attribut* bezeichnet. Die Dimension hängt von der Anzahl der Features ab, die definiert wurden. Ein Datensatz besteht aus mehreren Samples und kann als Matrix $H = [x_1, ...x_m]$ bestehend aus m Samples beschrieben werden. Das i-te Feature eines Feature-Vektors x wird oft auch mit $x^{(i)}$ benannt [Nil97, Kapitel 1.2].

3.2.2. Zielvariable

Zusätzlich kann zu jedem Sample eine Ergebnisvariable y definiert werden, die das Ergebnis zu einem Sample vorgibt. Dies kann verwendet werden, um auf Basis von vergangenen Ergebnissen Vorhersagen auf unbekannte Samples, deren Ergebnis nicht bekannt ist, zu machen. [Nil97, Kapitel 1.2].

3.2.3. Datenarten

Attribute können sowohl numerisch als auch kategorisch sein [SV08, Kapitel 1]. Numerische Attribute nehmen Werte aus dem reellen Raum $\mathbb{R}$ an. Bei kategorischen Werten wird zunächst eine Menge an *Klassen* (im englischen auch als *label* bezeichnet) $\{Klasse_1, Klasse_2, Klasse_n\}$ für das Attribut definiert. Für ein Sample kann ein Attribut eine oder mehrere Klassen annehmen.

3.2.4. Vorhersagemodell

Als Vorhersagemodell wird eine Funktion bezeichnet, die anhand von unbekannten Eingabewerten einen Wert für die Zielvariable erstellen kann.

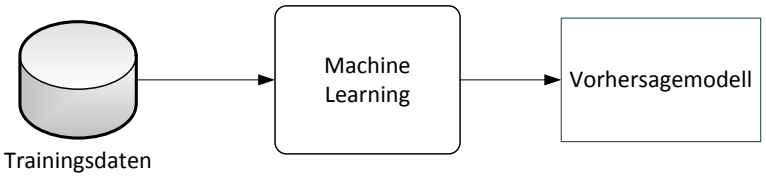


Abbildung 3.1.: Allgemeiner Ablauf der Erstellung eines Vorhersagemodelles.

Abbildung 3.1 zeigt die Erstellung eines Vorhersagemodelles. Das Vorhersagemodell wird dabei anhand eines Trainingsdatensatzes, der aus einer Menge an Feature-Vektoren besteht, für eine Zielvariable des Vektors optimiert. Dieser Ablauf wird auch Trainingsphase genannt. Ein optimiertes Modell wird oft als *trainiert* bezeichnet.

Das erstellte Vorhersagemodell kann dann verwendet werden, um für unbekannte Datensätze Vorhersagen über die Zielvariable zu treffen. Abbildung 3.2 stellt die Vorhersage der Zielvariablen eines unbekannten Datensatzes anhand eines trainierten Vorhersagemodells dar.

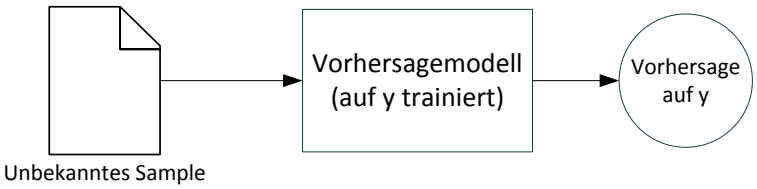


Abbildung 3.2.: Vorhersage der Zielvariablen anhand eines unbekannten Samples.

3.2.5. Lernarten

Modelle können auf unterschiedliche Arten eingelernt werden, die häufigsten Vertreter sind dabei das *Supervised Learning* und *Unsupervised Learning* [SV08]:

- *Supervised Learning*: Beim Supervised Learning hat man zu jedem Datensatz x ein Ergebnis y vorliegen, anhand derer das Vorhersagemodell erstellt und optimiert wird.
- *Unsupervised Learning*: Beim Unsupervised Learning liegen zu den Datensätzen keine Ergebnisse vor. Das Ziel ist, Strukturen in den Daten sichtbar zu machen.

3.2.6. Vorhersagearten

Die Art der Vorhersage kann sich dabei folgendermaßen unterteilen lassen [SV08]:

- *Klassifikation*: Klassifikation ist ein Beispiel für *Supervised Learning*. Der einfachste Fall des Klassifikationsproblems ist eine binäre Klassifikation, bei der man versucht anhand eines Datensatzes x festzustellen, ob der Datensatz zu einer Klasse gehört oder nicht. Beispielsweise wird das mit den Ergebniswerten $y = \{1, -1\}$ angegeben. Multiklassen-Klassifikation erweitert die binäre Klassifikation. Hier findet eine Zuordnung des Eingabedatensatzes zu einer Klasse aus einer Menge an Klassen $\{c_1, c_2, ..., c_n\}$ statt.
- *Regression*: Regression ist ebenso wie Klassifikation ein Beispiel für Supervised Learning. Unter Regression versteht man das Vorhersagen von einer Zielvariablen $y \in R$ (engl. *target variable* zu einem gegeben Datensatz $x = (x_i, ..., x_n)$. Die Werte, anhand derer die Zielvariable vorhergesagt wird, heißen Eingabevariablen (auch *input variables* oder *input features* genannt). Das Vorhersagemodell wird dabei mit einem Datensatz, der für jedes Sample den Wert der Zielvariable vorgibt, trainiert.
- *Clustering*: Clustering ist ein Beispiel für Unsupervised Learning, bei dem versucht wird, eine Gruppierung innerhalb eines Datensatzes vorzunehmen. Dieses Gruppierung wird anhand der Ähnlichkeit der Daten durchgeführt.

3.2.7. Einordnung dieser Arbeit

Ziel der Vorhersage ist die Ausführungszeit $y \in \mathbb{R}$ eines Offloadingkandidats zu einer Menge an Parameterwerten $x = (x^{(0)}, ..., x^{(n)})$. Das Vorhersagemodell wird anhand von vorhergegangen Ausführungen $H = [x_0, ..., x_m]$ erzeugt. Das Problem ist somit ein Regressionsproblem.

3.3. Methoden zum Einlernen,Validieren,Optimieren und Bewerten von Modellen

3.3.1. Unterteilung in Trainings-, Validations- und Testdatensatz

Klassisch können die Eingabedaten in 3 Datensätze unterteilt werden [WF05, Kapitel 5.1]. Anhand des Trainingsdatensatzes wird das Modell erstellt. Da viele Algorithmen parametrisiert werden können, ist es ratsam diese Parameter an die Daten anzupassen. Deshalb bedient man sich eines zweiten von der Trainingsphase unabhängigen Datensatz, dem Validierungsdatensatz. Anhand dieses werden die Parameter des Algorithmus optimiert. Um eine abschließende Aussage über die Leistungsfähigkeit treffen zu können (siehe Abschnitt 3.3.5) wird ein weiterer dem Modell unbekannter Datensatz, genannt Testdatensatz, verwendet, um Metriken zu erstellen.

3.3.2. k-fold Cross Validation

Anstatt eines dedizierten Validierungsdatensatzes kann auch eine alternative Validierungsmethode, die *k-Fold Cross Validation*, verwendet werden [WF05; RPL10, Kapitel 5.3]. Dabei wird ein gegebener Datensatz in k Teilmengen gespalten, von welcher $k - 1$ Teile für die Trainingsphase und 1 Teil für das Berechnen der Fehlerrate des trainierten Modelles verwendet werden. Aus den Fehlerraten aller erstellten Modelle wird das Mittel gebildet. Es hat sich herausgestellt, dass durch die Verwendung der k-Fold Cross Validation eine gute Abschätzung über die Leistung des Algorithmus mit einer bestimmten Parametrisierung getroffen werden kann.
Durch die Verwendung der k-fold Cross Validation ist nur noch eine Unterteilung in Trainings- und Testdatensätzen nötig. Im Weiteren lässt sich die k-fold Cross Validation für die Optimierung von Hyperparametern verwenden, wie in Abschnitt 3.3.4 beschrieben wird.

3.3.3. Transformationen zur Stabilisierung von Varianzen und Erzwingung positiver Vorhersagen

Transformationen der Zielvariablen können zur Stabilisierung der Varianz verwendet werden. Hierbei wird zuerst eine Hintransformation der Zielvariablen in den Trainingsdatensätzen durchgeführt und anschließend das Vorhersagemodell erstellt. Nach einer Vorhersage der Zielvariablen wird eine Rücktransformation gemacht. Gängige Hintransformationen sind beispielsweise die log-Transformation $y_{transformed} = log(y)$ oder Wurzel-Transformation $y_{transformed} = \sqrt[2]{y}$ Die entsprechenden Rücktransformationen sind $y' = e^{y_{transformed}}$ und $y' = y^2_{transformed}$. Außer der Verringerung der Varianz bieten die Transformationen den Vorteil, dass eine Vorhersage niemals negativ werden kann [BC64].

3.3.4. Optimierung von Hyperparametern

Parameter der Algorithmen, wie beispielsweise der C-Parameter der Support Vector Regression, werden oft als Hyperparameter bezeichnet. Diese müssen vor dem Erstellen eines Modelles festgelegt werden. Da sie sich aber stark auf die Vorhersagequalität auswirken können, macht es Sinn, diese automatisiert zu bestimmen.

Gittersuche

Um die optimale Parametrisierung eines Algorithmus finden zu können, bedient man sich oft der sogenannten Gittersuche (engl. Gridsearch) [HCL+03]. Bei der GridSearch werden für eine Menge an Parametern mit einem vordefinierten initialen Wertebereich Kombinationen gebildet, mit denen durch eine k-fold Cross Validation die Leistung des Modelles mit der aktuellen Parameterkombination bewertet wird. Die beste Kombination wird ausgewählt und eine feinere Unterteilung des Wertebereiches wird vorgenommen. Dies wird so lange fortgesetzt, bis keine bessere Parameterkombination mehr gefunden werden kann.

Das Problem bei dieser Methode ist, dass die Laufzeit dieser Suche mit der Anzahl der Parameter stark steigt, weswegen Methoden wie die RandomSearch [BB12] entwickelt wurden.

3.3.5. Metriken zur Qualitätsbewertung

Um die Qualität eines Vorhersagemodelles zu bewerten, kann man verschiedene Metriken verwenden. Zur Erstellung der Metriken wird das erstellte Vorhersagemodell auf einen Testdatensatz angewendet. Aus den tatsächlichen Ergebnissen y_i und den Vorhersagen $y_i^{'}$ werden die Differenzen $r_i = y_i^{'} - y_i$ gebildet. Diese Differenzen dienen als Grundlage für die folgenden Metriken [WF05, Kapitel 5.8]. In der Regel werden die Metriken anhand eines Testdatensatzes erstellt. Für die Metriken *Relative Root Squared Error* und *Relative Absolute Error* wird allerdings

auch der Datensatz benötigt, der für die Erstellung des Vorhersagemodelles verwendet worden ist.

Root Mean Squared Error

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} r_i^2}{n}}$$

Relative Root Squared Error

$$RSE = \frac{\sum_{i=1}^{n} r_i^2}{\sum_{i=1}^{n} (\overline{y} - y_i)^2},$$

wobei $\overline{y}$ der Mittelwert der Zielvariable y aus dem Trainingsdatensatz ist.

Mean Absolut Error

$$MAE = \frac{\sum_{i=1}^{n} |r_i|}{n}$$

Relative Absolute Error

$$RAE = \frac{\sum_{i=1}^{n} |r|}{\sum_{i=1}^{n} |\overline{y} - y_i|},$$

wobei $\overline{y}$ der Mittelwert der Zielvariable y aus dem Trainingsdatensatz ist.

Die oben aufgelisteten Metriken können zur Bewertung eines Algorithmus verwendet werden und geben einen Hinweis über die Qualität des erstellten Modelles.

3.4. Regressionsalgorithmen

Im Folgenden wird eine Auswahl an Machine Learning Ansätzen, die das Regressionsproblem lösen, beschrieben. Es soll drauf hingewiesen werden, dass auf die meisten Herleitungen verzichtet worden ist und dieses Kapitel nur zum groben Verständnis beiträgt.

3.4.1. Linear Regression

Linear Regression [Ng11] ist eine sehr bekannte Methode um Vorhersagemodelle zu erstellen. Im einfachsten Fall ist das Vorhersagemodell eine Mittelgerade, wie Abbildung 3.3 zeigt.

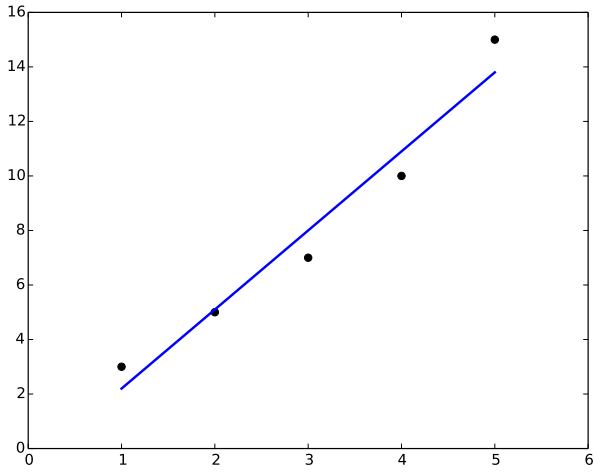


Abbildung 3.3.: Beispiel einer Mittelgerade eines Linear Regression Modelles.

Dabei wird eine lineare Kombination der Features $x_0, ...x_n$ gemacht, wobei jedes Feature unterschiedlich gewichtet wird.

(3.1) $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 ... + \theta_i x_i$

wobei y der vorhergesagte Wert ist, $\theta_1, ...\theta_i$ die Gewichtsfaktoren und x_i den Wert des Features i darstellt. Diese Funktion wird auch oft *hypothesis* genannt [Nil97, Kapitel 1.2]. Die Gleichung 3.1 wird meist auch folgendermaßen dargestellt:

(3.2) $h(x) = \sum_{i=1}^{n} \theta_i x_i = \theta^T x$

Das Ziel ist nun, die Parameter $\theta_1, ..., \theta_i$ so zu wählen, dass die entstehende Funktion die gegebenen Datenpunkte so gut wie möglich beschreibt. Hierzu wird eine *Kostenfunktion* definiert, nach welcher die Parameter $\theta_1, ..., \theta_i$ optimiert werden. Im Falle der Linear Regression wird die mittlere quadratische Abweichung (engl. *MSE*) verwendet.

(3.3) $J(\theta) = \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2,$

Mit Hilfe des *gradient descent* Algorithmus [Bis06, Abschnitt 3.1.3] werden nun die Parameter θ ausgehend von einem initialen Wert schrittweise aktualisiert, sodass die Kostenfunktion 3.3

minimiert wird. Der Parameter α stellt hierbei die Lernrate dar, also wie schnell man sich in Richtung Optimum bewegt[Ng11]:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta), \tag{3.4}$$

wobei die Least Mean Squares Updateregel für θ_j (nach partieller Ableitung) für ein Sample j aus dem Trainingsdatensatz wie folgt lautet[Ng11]:

$$\theta_j = \theta_j + \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \tag{3.5}$$

Wobei 3.5 $\forall_j$ bis zur Konvergenz ausgeführt wird.
Wurden die optimalen Parameter θ gefunden, können diese nun für die Linearkombination 3.1 eingesetzt werden und für die Vorhersage von unbekannten Datensätzen der Form $x = (x_i, ..., x_n)$ verwendet werden.

Ridge Regression

Anstatt der Least Mean Squares Updateregel wird bei der Ridge Regression eine andere Updateregel verwendet. Hierbei werden die Parameter θ mit einer vordefinierten Konstante γ versetzt. [Bis06, Kapitel 1].

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \frac{\gamma}{2}(h_\theta(x^{(i)})^2 \tag{3.6}$$

Durch die Ridge Regression werden große Parameterwerte bestraft. Dadurch kann eine hohe Varianz in den Daten ausgeglichen werden [WF05].

3.4.2. Polynomial Regression

Polynomial Regression ist eine Erweiterung der Linear Regression, bei der die Features in der Zielfunktion als Polynom abgebildet werden. Die Zielfunktion 3.1 wird folgendermaßen geändert [Ng11] :

$$y_i = \theta_0 + \theta_1 x_1^1 + \theta_1 x_1^2 ... \theta_1 x_1^n ... + \theta_2 x_2^1 + \theta_2 x_2^2 ... \theta_2 x_2^n ... + \theta_i x_i^1 + \theta_i x_i^2 ... \theta_i x_i^n \tag{3.7}$$

Abbildung 3.4 zeigt beispielhaft die Vorhersagefunktion die mit Hilfe der Polynomial Regression erstellt wurde.

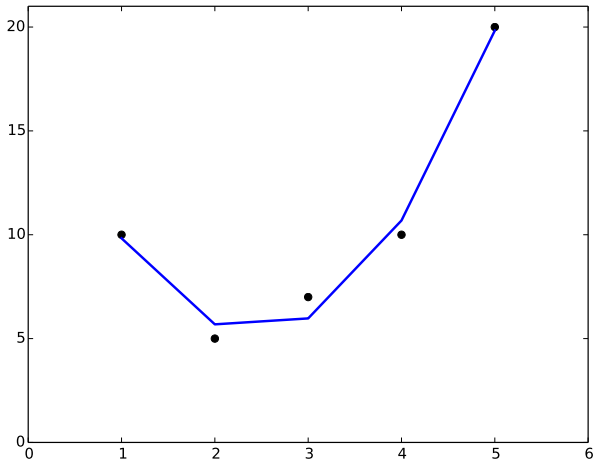


Abbildung 3.4.: Beispiel einer Vorhersage eines Polynomial Regression Modelles.

3.4.3. Support Vector Machines

Ursprünglich wurden *Support Vector Machines* (SVMs) zur Klassifikation entwickelt. Jedoch können mit der Verwendung einer anderen Kostenfunktion SVMs für Regression verwendet werden, was Support Vector Regression (SVR) genannt wird. Folgend wird zuerst die binäre Klassifikation, *Soft Margins* und der *Kernel-Trick* erläutert und anschließend auf die Änderungen eingegangen die zur *Support Vector Regression* führen. Die Herleitungen und mathematischen Definitionen beziehen sich in großen Teilen auf [Fle09; Gun+98].

Binäre Klassifikation

Wie in Abschnitt 3.2.6 beschrieben, hat man bei der binären Klassifikation Trainingsdaten, die $\{-1, 1\}$ als Ausgabewert haben. d.h. ob sie zu einer Klasse gehören oder nicht. Im einfachsten Fall sind die Trainingsdaten linear trennbar, d.h es gibt eine Hyperebene, welche zwischen beiden Klassen liegt.
Die Hyperebene H hat die folgende Form:

$$H := w * x + b = 0 \tag{3.8}$$

w is die Normale zur Hyperebene, b die Verschiebung und $\frac{b}{||w||}$ ist der (orthogonale) Abstand zum Ursprung. Da diese Definition keine eindeutige Hyperebene beschreibt, verwendet man hierbei die kanonische Form der Hyperebene, die eine Skalierung relativ zu den Trainingsdaten darstellt. Eine Hyperebene ist kanonisch, wenn folgende Bedingung erfüllt wird [Gun+98]:

$$\min_i |wx_i + b| = 1 \tag{3.9}$$

Punkte, die der Trennebene H am nächsten liegen, können durch einen Vektor beschrieben werden, auf welchem diese Punkte liegen. Diese werden als *Support Vektoren* bezeichnet. Aus den Support Vektoren lassen sich für die binäre Klassifikation folgende Hyperebenen ableiten[Fle09]:

(3.10) $H_1 := wx_i + b = 1 \quad für \quad y_i = 1$
$H_2 := wx_i + b = -1 \quad für \quad y_i = -1$

Dies kann zu folgender Gleichung zusammengefasst werden:

(3.11) $y_i(x_i * w + b) - 1 \geq 0 \forall i$

Abbildung 3.5 zeigt beispielhaft eine Trennebene.

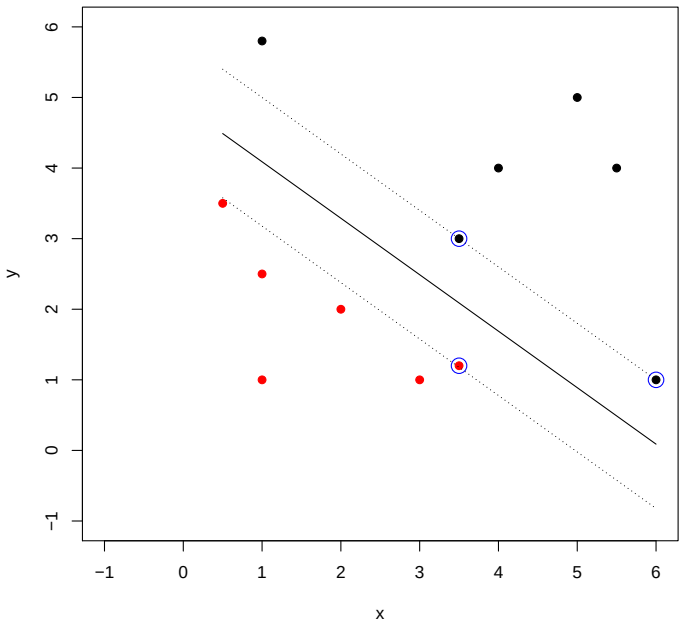


Abbildung 3.5.: Linear Separierbarer Raum. Die Trennebene wie auch die Hyperebenen, die durch die Support Vektoren definiert werden, wurden dargestellt.

Diese Hyperebenen haben denselben Abstand zur Trennebene H. Der Abstand wird in der SVM Terminologie als Rand (engl. *Margin*)bezeichnet. Der Rand ist $\frac{1}{||w||}$.
Das Ziel ist nun, diesen Rand zu maximieren, um die bestmögliche Trennung zu erzielen. Die Maximierung des Randes entspricht der Minimierung der Länge des Vektors w. Daraus ergibt sich folgendes Optimierungsproblem[Fle09]:

(3.12) $\min \frac{1}{2}||w||^2 \; s.t. \; y_i(x_i * w + b) - 1 \geq 0 \forall i$

Um die Nebenbedingungen aus 3.12 zu lösen, wird für jedes i ein Lagrange Multiplikator α_i eingeführt. Daraus ergibt sich folgende Gleichung[Fle09]:

$$L(w,b,\alpha) = L_P = \frac{1}{2}||w||^2 - \sum_{i=1}^{N} \alpha_i y_i (x_i * w + b) + \sum_{i=1}^{L} \alpha_i \tag{3.13}$$
$$s.t.\ \alpha_i \geq 0 \,\forall i$$

Man will nun das Minimum w und b und gleichzeitig das Maximum für α finden. Die Gleichung kann gelöst werden, indem bezüglich w und b abgeleitet wird und auf 0 gesetzt wird[Fle09]:

(3.14)
$$\frac{\partial L_p}{\partial w} = 0$$
$$w = \sum_{i=1}^{L} \alpha_i y_i x_i$$
$$und$$
$$\frac{\partial L_p}{\partial b} = 0$$
$$0 = \sum i = 1^L \alpha_i y_i$$

Setzt man nun 3.14 in 3.13 ein, erhält man das duale Problem zu 3.13 [Fle09]:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j x_i x_j \tag{3.15}$$
$$s.t. \quad \alpha_i \geq 0$$
$$\sum_{i=1}^{l} y_i \alpha_i = 0$$

3.15 ist nun konvexes quadratisches Problem, welches man mithilfe eines QP Solvers lösen kann. Löst man 3.15 erhält man w.
Die Verschiebung b kann man berechnen, indem man die Support Vektoren bestimmt. Diese können folgendermaßen berechnet werden[Fle09]:
Ausgehend von 3.16 werden diejenigen Vektoren genommen, für welche $\alpha_m > 0$ ist:

$$b = y_s - \sum_{m \in S} \alpha_m y_m x_m * x_s \tag{3.16}$$

Ein neuer Feature-Vektor x' kann folgendermaßen klassifiziert werden:

$$y' = sgn(w * x' + b) \tag{3.17}$$

Nichtlineare Klassifikation

Bisher wurde nur der Fall betrachtet, dass eine Hyperebene gefunden werden kann, wenn beide Klassen klar linear trennbar sind. In der Praxis treten jedoch häufig Fälle auf, wo eine lineare Trennbarkeit nicht gegeben ist. Das kann zum Einen passieren, wenn in den Trainingsdaten Rauschen vorzufinden ist und zum Anderen, wenn die Daten sich überlappen.

Ersteren Fall kann man durch die Einführung von Slack Variablen $\xi_i \geq 0$ lösen. Datenpunkte, welche auf der "falschen Seite" liegen, werden mit zusätzlichen Kosten belegt, welche von der Entfernung zur Hyperebene abhängen. Die Hyperebenen werden entsprechend der Zusatzkosten weg von den Daten auf der falschen Seite bewegt. Diese nicht mehr komplett strikte Bedingung für die Trennebene wird *Soft Margin* genannt. Folgende Änderungen in der Definition der kanonischen Hyperebenen 3.10 werden gemacht [Fle09]:

(3.18)
$$H_1 := wx_i + b = 1 - \xi; \quad f\ddot{u}r\ y_i = 1$$
$$H_2 := wx_i + b = -1 + \xi; \quad f\ddot{u}r\ y_i = -1$$
$$\xi_i \geq \forall_i$$

Zur Zielfunktion 3.12 werden nun die Slackvariablen und ein Parameter C hinzugefügt, welcher die Größe des Soft Margins bestimmt[Fle09]:

(3.19)
$$\begin{aligned} \underset{w,b,\xi_i}{\text{minimize}} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{N} \xi_i s.t. y_i(x_i * w + b) - 1 + \xi_i \\ \text{subject to} \quad & y_i(x_i * w + b) - 1 + \xi_i \geq 0\ \forall i \end{aligned}$$

Daraus ergibt sich letztlich folgendes Optimierungsproblem [Fle09]:

(3.20)
$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j x_i x_j$$
$$s.t. 0 \leq \alpha_i \leq 0 \leq C$$
$$\sum_{i=1}^{l} y_i\alpha_i = 0$$

Im zweiten Fall sind die Daten nicht durch ein eventuelles Rauschen nicht klar trennbar, sondern die zwei Klassen sind von Grund auf ineinander verschränkt.
Der SVM Algorithmus kann um den sogenannten *Kernel-Trick* erweitert werden. Dieser besteht

darin, dass man mit Hilfe einer Kernel-Funktion $k(x_i, x_j) = \phi x_i \phi x_j$, welche den Eingaberaum in einen höherdimensionalen Raum $\mathbb{R}^n - > \mathbb{R}^{n+m}$ projiziert, eine lineare Trennbarkeit erreichen kann. Dabei ist es nur notwendig, die Skalarprodukte x_i und x_j implizit durch die Kernel-Funktion zu transformieren. Daraus ergibt sich für das Optimierungsproblem folgende Gleichung[Fle09]:

$$\max \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{3.21}$$

$$s.t. \quad \alpha_i \geq 0$$

$$\sum_{i=1}^{l} y_i \alpha_i = 0$$

Typische verwendete Kernel sind[Fle09; Gun+98]:

- Polynomischer Kernel: $k(x, x^{'}) = (x * x^{'} + 1)^d$
- Gaussian Radial Basis Function (meist nur *RBF* genannt): $k(x, x^{'}) exp(-\frac{||x-x^{'}||^2}{2\phi^2})$
- Exponential Gaussian Radial Basis Function: $k(x, x^{'}) = exp(-\frac{||x-x^{'}||}{2\phi^2})$

Support Vector Regression

Im Falle einer numerischen Vorhersage berechnet sich der Wert y_i durch die folgende Ebene:

$$y_i = wx_i + b \tag{3.22}$$

Anstatt der einfachen Kostenfunktion aus 3.4.3 werden nun diejenigen Punkte mit **keinen** Kosten versetzt, die innerhalb des Randes liegen. Je nachdem ob die Punkte über dem oberen oder unter dem unteren Rand liegen, werden sie entsprechend mit den positiven Schlupfvariablen ξ^+ oder ξ^- versetzt. Als Kostenfunktion wird in der Regel die sogenannten ϵ-sensitive Kostenfunktion verwendet, welche folgende Eigenschaften hat[Fle09]:

$$L_\epsilon(y) = \begin{cases} 0, & \text{falls } |f(x) - y| < \epsilon \\ |f(x) - y| - \epsilon, & \text{sonst} \end{cases} \tag{3.23}$$

Die Zielfunktion für Support Vector Regression sieht nun folgendermaßen aus[Fle09]:

(3.24) $$C\sum_{i=1}^{N}(\xi^{+}\xi^{-}) + \frac{1}{2}||w||^2$$
$$s.t.\xi^{+} \geq 0 \quad und \quad \xi^{-} \geq 0 \forall_i$$

Wie zuvor werden nun Lagrange Multiplikatoren eingeführt [Fle09]:

(3.25)
$$L_P = C*\sum_{i=1}^{N}(\xi_i^{+}+\xi_i^{-}+\frac{1}{2}*||w||^2-\sum_{i=1}^{N}(\gamma_i^{+}\xi_i^{+}+\gamma_i^{-}\xi_i^{-})-\sum_{i=1}^{N}\alpha_i^{+})\epsilon+\xi_i^{+}-y_i-t_i)-\sum_{i=1}^{N}\alpha_i^{-}(\epsilon+\xi_i^{-}-y_i+t_i)$$
$$s.t.\alpha_i^{+} \geq 0, \alpha_i^{-} \geq 0, \gamma_i^{+} \geq 0, \gamma_i^{-} \geq 0 \forall_i$$

Die duale Form dieser Gleichung lautet:

(3.26) $$L_D = \sum_{i=1}^{N}(\alpha_i^{+} - \alpha_i^{-})t_i - \epsilon\sum_{i=1}^{N}(\alpha_i^{+} - \alpha_i^{-})(\alpha_j^{+} - \alpha_j^{-})x_i * x_j$$

Hierzu wird nach α^{+} und α^{-} optimiert sodass $0 \leq \alpha_i^{+} \leq C, 0 \leq \alpha_i^{-} \leq C$ sowie $\sum_{i=1}^{N}(\alpha_i^{+}-\alpha_i^{-}) = 0 \; \forall_i$

Vorhersagen zu einem neuen, unbekannten Datensatz x können nun mit folgender Gleichung gemacht werden [Fle09]:

(3.27) $$y' = \sum_{i=1}^{N}(\alpha_i^{+} - \alpha_i^{-})x_i * x' + b$$

Die Verschiebung b kann durch die Bestimmung der Support Vektoren berechnet werden. Die Support Vektoren. Diese sind diejenigen Vektoren, welche folgende Bedingungen erfüllen: $0 < \alpha < C$ und $\xi_i^{+} = 0$ beziehungsweise $\xi_i^{-} = 0$.
Das Ergebnis ist eine Menge S an Support Vektoren. Um die Verschiebung b zu berechnen wird nun der Mittelwert über allen Support Vektoren aus der Menge S gebildet [Fle09]:

(3.28) $$b = \frac{1}{N_s}\sum_{s\in S}[t_s - \epsilon - \sum_{m\in S}^{L}\alpha_m^{+} - \alpha_m^{-})x_m * x_s]$$

Abbildung 3.6 zeigt abschließend eine beispielhafte Veranschaulichung einer SVR mit RBF Kernel. Der Rand wurde dabei ebenfalls darsgestellt.

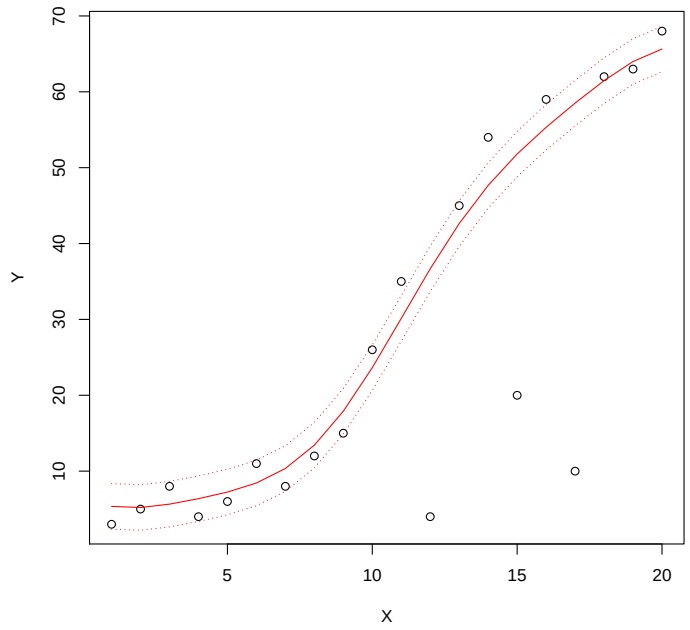


Abbildung 3.6.: Darstellung des Randes bei der Support Vector Regression mit Radial Basis Function(RBF) Kernel.

3.4.4. K-Nearest-Neighbor-Regression

Der K-Nearest-Neighbor-Algorithmus[SV08] kann zur Regression oder Klassifikation verwendet werden. Dabei werden zu einem unbekannten Sample s die k Nachbarn mit dem geringsten Abstand für die Entscheidung verwendet. Für den Abstand wird meist der Euklidsche Abstand verwendet:

$$\sum_{i=1}^{k}(x_i - y_i)^2 \tag{3.29}$$

Im Falle der Klassifikation wird s der Klasse zugewiesen, welche häufiger in den k Nachbarn vorkommt. Bei der Regression wird aus den k Nachbarn der Mittelwert der Zielvariablen gebildet und ausgegeben.

3.4.5. Künstliche Neuronale Netze

Für die Entwicklung von künstlichen neuronalen Netzen[Nil97, Kapitel 4] wurde als Grundlage die Funktionsweise von biologischen neuronalen Netzen genommen. Ein neuronales Netz besteht aus *Neuronen*, welche miteinander über Synapsen vernetzt sind. Jedes Neuron kann

mehrere Eingabe - und Ausgabestränge haben. Künstliche neuronale Netze übernehmen dieses Grundkonzept: Ein Neuron ist eine Berechnungseinheit, welche anhand von einer oder mehreren Eingaben eine oder mehrere Ausgaben erzeugen kann. Verbindungen zwischen Neuronen können verschieden gewichtet werden.
Die Netze sind dabei in Schichten aufgeteilt. Der *Input-Layer* stellt die Eingabedaten für das Netz dar. Anschließend werden die Daten im *Hidden-Layer* verarbeitet. Der Hidden-Layer kann ein - oder mehrschichtig sein. Abschließend werden die Ergebnisse im *Output-Layer* erstellt. Folgend sollen nur Feedforward Netze betrachtet werden. Ein Feedforward Netz ist formal gesehen ein gerichteter azyklischer Graph. Es werden keine Zustände gespeichert oder beispielsweise auf ein Neuron einer vorherigen Schicht wieder zurückgeführt. Abbildung 3.7 stellt ein typisches Feedworward Netz dar.

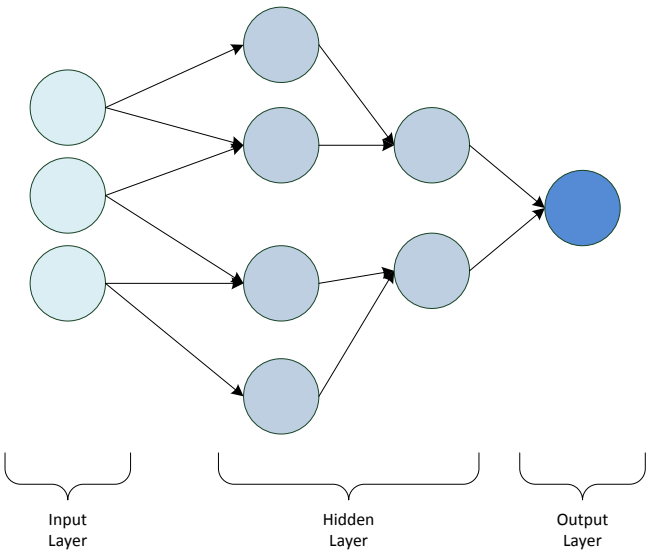


Abbildung 3.7.: Aufbau eines zwei-Schichtigen feedforward Netzes.

Im Folgenden soll das *Radial Basis Function Network*(kurz: RBF Network) betrachtet werden.

Radial Basis Function Network

Das Radial Basis Function Network (kurz: RBF-Network)[WF05, Kapitel 6.4] ist ein 3-schichtiges neuronales Netz, bestehend aus einem Input-Layer, einem Output-Layer und einer Schicht im Hidden-Layer. Abbildung 3.8 zeigt veranschaulicht den Aufbau. Um das RBF Netzwerk für die Regression zu nutzen, muss jedem Feature des Feature-Vektors ein Neuron zugewiesen werden.

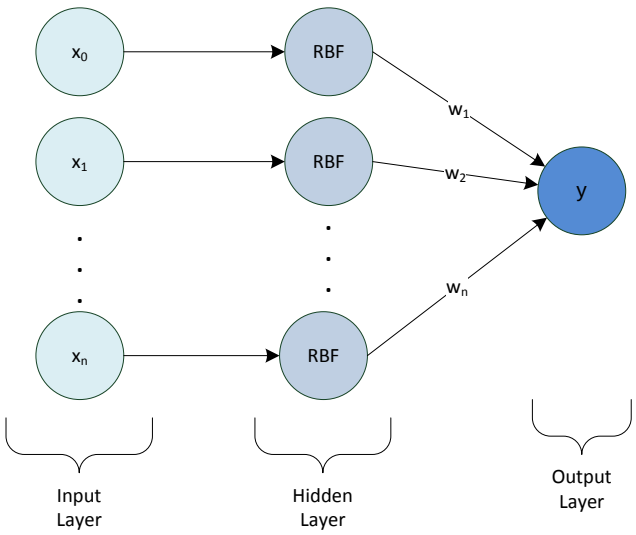


Abbildung 3.8.: Aufbau des RBF Network.

Die Berechnungseinheiten(Neuronen) des Hidden-Layers verwenden die Gauss'sche Form der Radial Basis Function [Orr+96] [SKP01]:

(3.30) $\phi(x) = \exp(-\beta||x - c||^2)$

Wobei c der Prototyp für das Zentrum der Funktion ist. β bestimmt die Breite der Funktion. Für jedes Feature des Inputvektors wird ein Neuron definiert, das mit β und c versetzt ist. Zusätzlich wird die Ausgabe des Neurons mit einem Gewicht w versetzt.

Diese drei Parameter müssen eingelernt werden. c und β können durch ein k-means Clustering[Bis06, Kapitel 9.1] bestimmt werden. β kann anschließend ausgehend durch das vom Algorithmus bestimmte Zentrum berechnet werden[SKP01]:

(3.31) $\sigma = \frac{1}{m} \sum_{i=1}^{m} ||x_i - c||$

wobei c das Zentrum und x_i ein Element des Clusters ist. Anschließend kann β folgendermaßen berechnet werden [SKP01]:

(3.32) $\beta = \frac{1}{2\sigma^2}$

Die Gewichte w können mit einem Algorithmus wie Gradient Descent [Bis06, Abschnitt 3.1.3] eingelernt werden.

Um anhand eines Eingabevektors x nun eine Vorhersage zu treffen, werden alle gewichteten Ausgaben der Neuronen kombiniert[SKP01]:

$$y = \sum_{i=1}^{n} w_i * \phi_i(x_i) \tag{3.33}$$

3.4.6. Regression Trees

Anstatt der Verwendung eines globalen Modelles, wie bei Linear Regression oder Support Vector Regression, unterteilen Regression Trees die Daten in mehrere Regionen, auf denen diese Modelle angewendet werden. Beispiele für Decision Trees sind C4.5[Qui93] und CART [BFSO84].

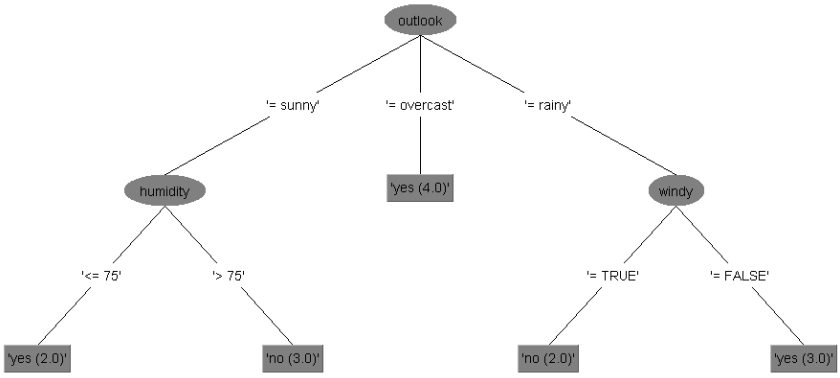


Abbildung 3.9.: Beispiel eines Decision Trees zur Klassifikation. Für die Erstellung des Baumes wurde eine Implementierung des C4.5 Algorithmus des WEKA Frameworks verwendet. Der Testdatensatz stammt von [Wei16].

Die Baumstruktur wird rekursiv aufgebaut. Hierbei gibt es ein Split-Kriterium wonach die Daten aufgeteilt werden. Im Falle der Regression wird für das Splitting ein anderes Kriterium verwendet. Es wird ein Binärbaum erstellt, d.h. ein Knoten hat immer zwei Kindknoten. Folgender Abschnitt bezieht sich auf den CART Algorithmus. Der Pseudocode für die Baumerstellung ist im Detail in [Loh11] und [WF05] beschrieben:

1. Am Wurzelknoten beginnen
2. Ausgehend vom aktuellen Datensatz soll das Attribut gefunden werden, das das Splitkriterium erfüllt. Nach diesem Attribut wird der Datensatz in zwei weitere Teile aufgeteilt.
3. Ist das Stopkriterium erreicht, bricht die Rekursion ab. Ansonsten wird 2. für jeden neu gesplitteten Knoten ausgeführt.

Das Splitkriterium ist für das Attribut erfüllt, dass die folgende Gleichung maximiert [WF05, Kapitel 6.6]:

(3.34) SDR = sd(T) - $\sum_i \frac{|T_i|}{|T|} * sd(T_i)$

wobei T_i das Set ist, das bei der Auswahl eines Attributes entsteht. $sd(T)$ ist die Standardabweichung der Variablen.
Anschließend wird der erstellte Baum wieder verkleinert (engl. *pruning*). CART verwendet das sogenannte *Cost complexity pruning* [BKK+98] zur Verkleinerung des Baumes.

Jedes Blatt des finalen Baumes stellt nun einen Wertebereich dar, auf dem ein Regressionsalgorithmus, wie Linear Regression ausgeführt werden kann. Alternativ kann auch beispielsweise der Mittelwert des Wertebereiches verwendet werden. Soll eine neue Vorhersage getätigt werden, wird der Entscheidungsbaum verwendet um das richtige Vorhersagemodell auszuwählen.

3.5. Zusammenfassung

Bei der Vorhersage der Ausführungszeit handelt es sich um ein Regressionsproblem. Es wurden Algorithmen vorgestellt, die sich in ihrer Komplexität und Vorgehensweise teilweise stark unterscheiden. KNN Regression ist ein sehr einfacher Algorithmus wohingegen Support Vector Regression deutlich komplexer ist. Da es sich um *Supervised Learning* handelt, ist zur Erstellung der Vorhersagemodelle ein Trainingsdatensatz vonnöten. Um die Machine Learning Algorithmen für die Vorhersage der Ausführungszeiten zu verwenden, drängen sich folgende Fragen auf:

- Wie werden die Trainingsdatensätze erstellt?
- Wo werden die Vorhersagemodelle erstellt?
- Wo findet die Vorhersage statt?

Im Kapitel 4: *Kooperativer Systementwurf* wird ein Lösungsansatz für die Fragestellungen vorgestellt.

4. Kooperativer Systementwurf

4.1. Einleitung

Im vorherigen Kapitel wurden Machine Learning Algorithmen vorgestellt, welche das Regressionsproblem lösen können und für die Vorhersage von Ausführungszeiten in Frage kommen. In diesem Kapitel wird ein kooperativer Systementwurf vorgestellt, mit dessen Hilfe Samples und Vorhersagemodelle verwaltet und verwendet werden können. Wie Abbildung 4.1 zeigt, wird die Vorhersage verwendet, um anschließend die Offloadingentscheidung zu treffen. Wie in

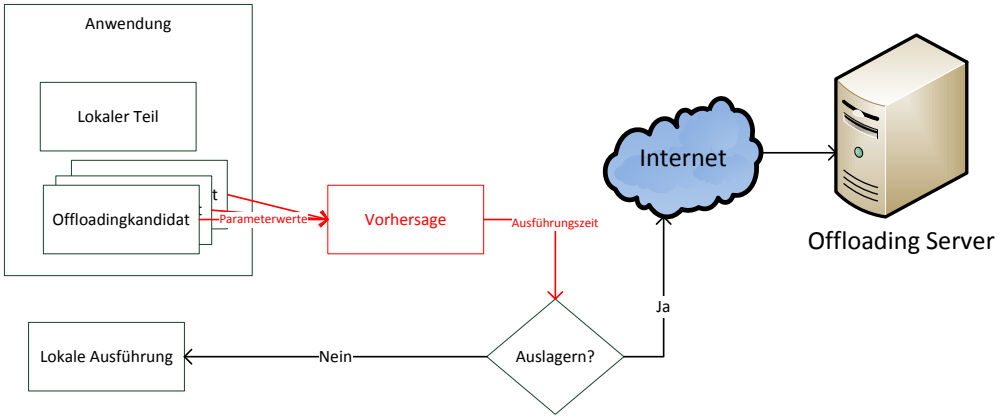


Abbildung 4.1.: Offloading: Die Entscheidung soll mit Hilfe einer Vorhersage (rot) verbessert werden.

Kapitel 3 beschrieben handelt es sich bei dem Problem der Vorhersage von Ausführungszeiten um *Supervised Learning* und ein Regressionsproblem. Supervised Learning bedeutet, dass man anhand eines Trainingsdatensatzes ein Vorhersagemodell erstellt. Der Systementwurf muss also auch eine Möglichkeit bereitstellen, Feature-Vektoren aufzuzeichnen und zu sammeln.
Der Systementwurf soll **kooperativ** gestaltet werden: jeder, der das Offloading nutzt, soll sich an der Erstellung der Vorhersagemodelle durch Bereitstellung von aufgezeichneten Ausführungsszenarios beteiligen.
Um sinnvolle Vorhersagen zu treffen, benötigen die Algorithmen einen ausreichend großen Trainingsdatensatz. Die Größe ist dabei von der Komplexität des jeweiligen Offloadingkandidaten abhängig. Es ist von Vorteil, wenn über mehrere mobile Geräte Samples gesammelt werden

können. Aus diesem gebündelten Datenbestand kann ein insgesamt besseres Vorhersagemodell abgeleitet werden, als wenn nur lokal aufgezeichnete Ausführungen für Vorhersagemodelle verwendet werden.
Die Erstellung der Vorhersagemodelle sollte ebenfalls nicht auf dem Mobilgerät stattfinden, da die Erstellung je nach Größe des Datenbestandes und Algorithmus lange dauern kann (vgl. Kapitel 6).

Um dies zu erreichen, muss das System folgende Aufgaben erfüllen:

1. Bündelung von Samples.
2. Ausgelagerte Erstellung des Vorhersagemodelles.
3. Update des Vorhersagemodelles für den Client.

Aus diesen Anforderungen an das System lassen sich zwei Komponenten ableiten. Zum Einen ein *Client*, der die Komponente auf dem Mobilgerät darstellt, welche die Offloadingentscheidung für einen Offloadingkandidaten trifft. Es wird angenommen, dass die Entscheidungskomponente in einer Art mit der tatsächlichen Anwendung des Offloadingkandidaten gekoppelt ist, sodass auf die Parameter zugegriffen und Ausführungszeit aufgezeichnet werden kann.
Für diesen Entwurf wird ein Offloading auf Methodenebene angenommen - ein Offloadingkandidat ist also eine Methode. Die Eingabe sind die Parameter der Methode und die Ausgabe dessen Rückgabewert. Der Client muss Methodenaufrufe aufzeichnen, sowie ein Vorhersagemodell zur Vorhersage der Ausführungszeit anhand der aktuellen Parameter benutzen können. Zum Anderen lässt sich eine Serverkomponente ableiten, welcher die Sammelstelle für die Samples darstellt und aus diesen die Vorhersagemodelle für die Offloadingkandidaten erstellt.

Zusätzlich lassen sich weitere Anforderungen aufgrund der begrenzten Ressourcen des Mobilgerätes und dem praktischen Fall, dass oft nur ein beschränktes Datenvolumen nutzbar ist, ableiten:

- *Zwischenspeicherung von Samples auf dem Mobilgerät:* Da bei Mobilgeräten nicht dauerhaft eine Verbindung bestehen kann, sollte es auf diesem möglich sein, Samples zwischenzuspeichern. Diese können bei einer ungedrosselten Verbindung an den Server geschickt werden.
- *Persistentes Speichern des Vorhersagemodelles auf dem Mobilgerät:* Ebenfalls ist es ratsam das Vorhersagemodell, das zuletzt für eine Methode einer Anwendung erhalten wurde, persistent bis zum nächsten Start der Anwendung zu speichern.

Aus diesen zusätzlichen Anforderungen an das System lässt sich eine dritte Komponente ableiten, welche ein Zwischenglied zwischen dem Client und dem Server darstellt. Diese Zusatzkomponente wird folgend als *Controller* bezeichnet. Abbildung 4.2 veranschaulicht, wo sich die drei Komponenten sich befinden. Der Client und der Controller befinden sich auf dem Mobilgerät. Der Server läuft auf einer entfernten Maschine.

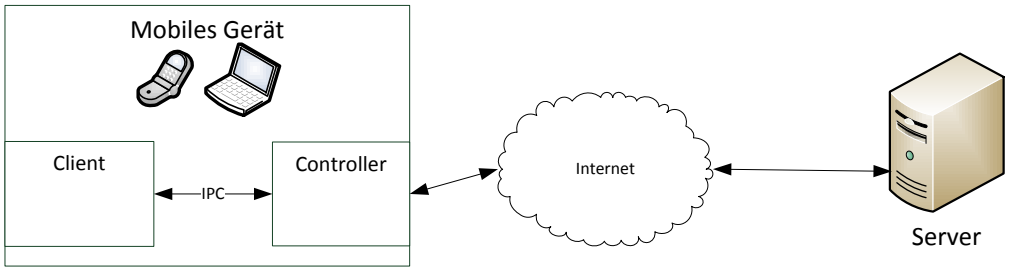


Abbildung 4.2.: Hauptkomponenten des Entwurfs: Der Client und der Controller laufen auf dem Mobilgerät, wohingegen sich der Server auf einer entfernten Maschine befinden kann. Es können auf dem Mobilgerät mehrere Clients gleichzeitig laufen, jedoch nur ein Controller.

In den folgenden Abschnitten werden diese Anforderungen weiter verfeinert und ein Systementwurf vorgestellt. Der Systementwurf soll dabei nicht dazu beitragen, **wie** Offloading stattfinden soll, sondern soll die Entscheidung, **ob** geoffloaded wird, verbessern.

Außerdem werden folgende Aspekte im Entwurf nicht weiter betrachtet:

- Security
- Reliability
- Scalability

4.2. Anforderungen an die Komponenten

Wie im Abschnitt 4.1 erläutert wurde, besteht der Systementwurf aus drei Komponenten, nämlich einem Client, einem Controller und einem Server. In diesem Abschnitt werden die Anforderungen an die jeweiligen Komponenten spezifiziert und verfeinert. Zunächst werden jedoch einige komponentenübergreifende Definitionen eingeführt.

4.2.1. Definitionen

Method-ID

Jede Methode muss eindeutig referenzierbar sein. Unter der Annahme, dass Namen für Anwendungen eindeutig vergeben werden und dass eine Anwendung mit unterschiedlichen Versionen existieren kann, ergeben sich folgende Attribute die für die Erstellung einer ID, die als Referenz verwendet werden kann:

- Methodensignatur
- Name der Anwendung
- Version der Anwendung

Die Methodensignatur umfasst den Namen, sowie die Typen der Parameter in der Reihenfolge, wie sie im Code definiert wurden. Da beispielsweise Java *Method-Overloading* [1] fähig ist, wird die komplette Signatur benötigt.
Aus diesen Attributen wird ein SHA-1 Hash gebildet, welcher im folgenden als *Method-ID* bezeichnet wird und den eindeutigen Identifizierer darstellt.

Sample

Ein Sample ist wie in Kapitel 3 beschrieben ein n-dimensionaler Vektor, dessen Elemente kategorisch oder numerisch sein können. Ein Sample besteht aus den Eingabeparametern einer Methode. Die Zielvariable, also der Wert der nach Erstellung des Modelles vorhergesagt werden soll, ist die Ausführungszeit. Ein Sample wird eindeutig anhand einer *Method-ID* referenziert.

Vorhersagemodell

Ein Vorhersagemodell gibt anhand eines Feature-Vektors eine Vorhersage auf die Zielvariable. Jedes Vorhersagemodell wird dabei mit der *Method-ID* und einem Zählerwert eindeutig identifiziert. Der Zählerwert ist beim erstmaligen Erstellen eines Vorhersagemodelles 1. Für jedes Neuerstellen (d.h. das Vorhersagemodell wird geupdated) des Vorhersagemodelles wird der Zähler um 1 erhöht. Somit ist jede Modellversion eindeutig identifizierbar.

Client

Als Client wird die Komponente bezeichnet, die die Vorhersagen für die Offloadingentscheidung nutzen möchte. Für jede Methode, die auslagerbar ist, muss vom Client eine separate Registrierung durchgeführt werden.

Controller

Als Controller wird die Zwischenschicht zwischen Client und Server bezeichnet. Der Controller befindet sich ebenfalls auf dem mobilen Gerät. Pro Mobilgerät gibt es einen dedizierten Controller.

[1] Beim Method-Overloading ist es möglich, dass Methoden mit denselben Namen existieren, so lange die Reihenfolge der Parametertypen oder die Typen der Parameter selbst verschieden sind [AGHH00].

Server

Der Server dient zum Verwalten und Erstellen von Vorhersagemodellen. Der Server läuft auf einer entfernten Maschine.

Nachrichten

Jede Nachricht hat immer eine *Method-ID* als Referenz. Folgende Nachrichten werden definiert:

- **Register**: Diese Nachricht besteht aus der Method-ID und den für die Vorhersage genutzten Parametern. Die Nachricht wird zum Bekanntmachen einer Methode, für die Vorhersagemodell erstellt werden sollen, verwendet.
- **RegisterCtrl**: Die Registrierungsnachricht, die vom Controller zum Server weitergeleitet wird. Hierbei schickt der Controller den aktuellen Zählerwert des lokal gespeicherten Vorhersagemodelles mit. Ein Zählerwert von 0 wird verwendet, wenn kein Modell lokal vorhanden ist.
- **Unregister**: Diese Nachricht besteht aus der Method-ID und wird verwendet um keine Updates mehr zu erhalten.
- **Sample**: Dient zum verschicken eines einzelnen Samples an den Server.
- **SampleBatch**: Dient zum Verschicken mehrerer Samples an den Server.
- **Modelupdate**: Dient zum Verschicken eines Vorhersagemodelles an den Controller oder Client.
- **ModelUpdateNotification**: Benachrichtigt den Controller, dass ein Update verfügbar ist.
- **ModelUpdateAcknowledgement**: Bestätigung des Controllers, dass ein Update erhalten werden kann.

4.2.2. Anforderungen an die Serverkomponente

Die Server-Komponente dient als zentrale Stelle für das Sammeln von Samples und das Erstellen der Vorhersagemodelle. Der Server muss folgende Anforderungen erfüllen:

Anforderung 1: Sammeln von Samples

Samples, welche von Anwendungen aufgezeichnet werden, sollen auf dem Server gebündelt werden. Hierbei ist es nötig, Samples eindeutig zu der jeweiligen Methode, für die aufgezeichnet wurde, über eine ID zu identifizieren. Dazu wird die in Abschnitt 4.2.1 beschriebene *Method-ID* verwendet.

Anforderung 2: Vorhersagemodelle erstellen

Die gesammelten Samples sollen verwendet werden, um ein Vorhersagemodell zu erstellen. Dabei sollen mehrere Machine Learning Algorithmen verwendet werden, aus welchen Vorhersagemodelle erstellt werden. Das beste Modell wird anhand einer Metrik ausgewählt. Als Metrik soll die in Kapitel 3.3.5 beschriebene Metrik *Root Mean Squared Error* verwendet werden.

Anforderung 3: Erstellungsentscheidung

Die Entscheidung, wann Vorhersagemodelle neu erstellt werden, soll folgendermaßen getroffen werden können:

- *Periodisch*: Die Vorhersagemodelle werden unabhängig von der Anzahl an dazugekommenen Samples innerhalb eines definierten Intervalls neu erstellt.
- *Schwellwert*: Die Vorhersagemodelle werden nur dann neu erstellt, wenn ein bestimmter Schwellwert an neuen Samples überschritten wurde.

Anforderung 4: Persistente Speicherung von Vorhersagemodellen und Samples

Die Samples, sowie die erstellten Vorhersagemodelle, werden in einer Datenbank verwaltet und persistent gespeichert. Jedes Sample besteht aus einer Menge an Attributwerten und der dazugehörigen *Method-ID*.
Zu jedem Datensatz einer Methode wird zudem eine Beschreibung der Attribute gespeichert. Vorhersagemodelle werden ebenfalls mit der Method-ID referenziert.

Anforderung 5: Registrierung von Controllern

Controller können sich beim Server anmelden. Bei einer Anmeldung werden im Server in einer in-memory Datenstruktur die *Method-ID*, sowie die Netzwerk-Adresse des Controllers gespeichert.

Anforderung 6: Updaten der Controller auf Basis der aktuellen Registrierungen

Der Server soll nur für diejenigen Methoden Updates verschicken, für welche Registrierungen vorliegen. Da die Übertragung der Modelle selbst auf dem Mobilgerät Energie benötigt, muss das Mobilgerät selbst entscheiden können, ob es ein Update erhalten soll. Daher soll der Server zunächst eine Benachrichtigung an den zuständigen Controller schicken, welcher dann überprüft, ob das Update empfangen werden soll.
Der Server überträgt anschließend nur an diejenigen Controller das Modell, für das er eine positive Bestätigung erhalten hat. Die Nachricht an den Controller enthält die Größe des Vorhersagemodelles in Byte.

4.2.3. Anforderungen an die Controllerkomponente

Anforderung 8: Überprüfung der vorhandenen Ressourcen

Da das Empfangen von Vorhersagemodellen auch Energie kostet und das System des Mobilgerätes beeinträchtigt, muss je nach den aktuellen Bedingungen entschieden werden, ob ein Modell empfangen werden sollte. Folgenden Ressourcen müssen überwacht werden:

- Aktuelle Internetverbindung: Für den Nutzer kann es von Relevanz sein, eine Übertragung der Modelle zu vermeiden, wenn nur gedeckelte Mobilverbindungen (2-4G) verfügbar sind.
- Aktuelle Energiekapazität: Soll energieoptimal ausgelagert werden, beeinflusst das Empfangen der Modelle die Gesamteinsparung. In Kapitel 6 wird untersucht, inwiefern diese das System beeinflussen.

Daraus lassen sich Bedingungen definieren, bei denen ein Modell erhalten werden soll:

- **Option 1**: Immer Übertragen.
- **Option 2**: Übertragung ausschließlich bei Laden des Akkus des Mobilgerätes.
- **Option 3**: Übertragung ausschließlich bei ungedrosselter Verbindung.
- **Option 4**: Übertragung ausschließlich bei Laden des Akkus des Mobilgerätes und ungedrosselter Verbindung.

Welche Bedingung an das System gestellt wird, ist vom konkreten Mobilgerät abhängig.

Anforderung 9: Bereitstellung von Vorhersagemodellen

Der Controller soll registrierten Anwendungen Vorhersagemodelle für die Offloadingentscheidung bereitstellen. Erhält der Controller Updates für die Vorhersagemodelle, so werden diese auch an die Anwendung weitergeleitet. Bei der ersten Registrierung einer Anwendung wird initial ein persistent gespeichertes Vorhersagemodell, falls vorhanden, an die Anwendung weitergeleitet.

Anforderung 10: Weiterleitung von Nachrichten zum Client und zum Server

Der Controller dient als Zwischenstation für die Client-Server Kommunikation. Nachrichten, die an bestimmte Clients oder den Server gerichtet sind, soll der Controller weiterleiten. Dies umfasst sowohl die in Abschnitt 4.2.1 Kontroll- als auch Datennachrichten.

Anforderung 11: Caching von Samples

Es soll die Möglichkeit geben, Samples erst lokal auf dem Gerät zu cachen und anschließend gebündelt auf den Server zu übertragen. Durch das Caching kann beispielsweise verhindert werden, dass bei einer langsamen Verbindung des Mobilgerätes die Bandbreite zu sehr belastet oder bei einem gedeckelten Datenvolumen dieses aufgebraucht wird. Ebenfalls wird der Overhead durch die Kapselung in UDP Pakete verringert. Die Entscheidung innerhalb des Controllers, wann die gesammelten Samples an den Server übertragen werden, wird in der Anforderung 13 definiert. Das Caching soll aktiviert und deaktiviert werden können.

Anforderung 12: Caching von Vorhersagemodellen

Ebenfalls sollen Vorhersagemodelle persistent gespeichert werden. Die persistent gespeicherten Modelle werden - sofern vorhanden - initial an den Client geschickt. Ist dies nicht der Fall, wird bei einer Registrierung beim Server eine Update-Benachrichtigung an den Controller geschickt. Anschließend kann je nach definierten Bedingungen ein Modell empfangen werden.

Anforderung 13: Leerung des Caches

Mobilgeräte haben je nach Tarif unterschiedliche Einschränkungen in Bezug auf ihr Datenvolumen und die Bandbreite. Praktisch ist oft der Fall gegeben, dass Nutzer nur ein bestimmtes Volumen mit voller Netzgeschwindigkeit verwenden können und anschließend meist auf 64 KBit/s Download und 16 KBit/s Upload gedrosselt wird [Tel].
Der Nutzer ist darauf bedacht, sein Volumen möglichst sinnvoll einzusetzen [Acc10, S. 47]. Deswegen sollte es möglich sein die Entscheidung, wann die Übertragung der Samples stattfindet, folgendermaßen zu treffen:

- *Ungedrosselte Verbindung*: Verbindet sich der Nutzer in ein für ihn ungedrosseltes Netz, beispielsweise WLAN, sollen die lokalen Samples automatisch an den Server übertragen werden.
- *Manuell*: Dem Nutzer soll ermöglicht werden, das Leeren des Caches manuell anzustoßen.
- *Anzahl gespeicherter Samples*: Wird ein Schwellwert einer bestimmten Anzahl an gespeicherten Samples überschritten, sollen diese automatisch an den Server geschickt werden.
- *Periodisch*: Die Samples werden periodisch an den Server geschickt.

Zusätzlich soll noch die Bedingung gestellt werden können, dass nur bei einem Aufladen des Akkus übertragen wird.

Anforderung 14: Persistentes Caching

Da nicht garantiert werden kann, ob das Mobilgerät bis zum Zeitpunkt der Übertragung angeschaltet ist, müssen Samples zwischenzeitlich persistent gespeichert werden.

Anforderung 15: Registrierung beim Server

Registriert sich eine Anwendung beim Controller, so registriert sich dieser stellvertretend für den Client beim Server.

Anforderung 16: Abmeldung beim Server

Meldet sich eine Anwendung von Updates ab, so schickt der Controller ebenfalls eine Abmeldung an den Server weiter, damit keine Updates für die Vorhersagemodelle der Methoden erhalten werden.

Anforderung 17: Registrierung eines Clients am Controller

Bei einer Registrierung im Client wird in einer in-memory Struktur die ID der Methode sowie der Netzwerk-Port gespeichert. Verschiedene Registrierungsmodi sollen unterstützt werden:

- *Normaler Modus*: Im normalen Modus wird beim Schließen der Anwendung und dem Erhalt einer Abmeldungs-Nachricht der in-memory Eintrag gelöscht.
- *Persistenter Modus*: Beim persistenten Modus werden vom Controller auch nach dem Beenden der Anwendung Updates für die registrierten Methoden erhalten. Der Client muss explizit eine Abmeldungs-Nachricht für den persistenten Modus schicken.

- *Timeout-basierter Modus*: Beim Timeout-basierten Modus hält der Controller die Registriertung nur eine gewisse Zeitspanne persistent. Nach Ablauf des Timeouts wird diese gelöscht.
- *Lokaler Modus*: In diesem Modus wird lediglich das lokal im Controller gespeicherte Vorhersagemodell verwendet. Falls keines vorhanden ist, wird einmalig ein Modell vom Server geladen.

Anforderung 18: Persistente Registrierungen

Um den persistenten und Timeout-basierten Modus aus Anforderung 17 zu unterstützen, sollen diese Arten der Registrierung persistent auf dem Mobilgerät gespeichert werden.

Anforderung 19: Abmeldung eines Clients am Controller

Möchte der Client keine weiteren Updates erhalten, schickt dieser an den Controller eine Abmeldung. Bei der Abmeldung wird der in-memory Eintrag von Method-ID und Netzwerk-Port gelöscht.

4.2.4. Anforderungen an die Clientkomponente

Der Client stellt die Komponenten dar, welche die Vorhersage zur Offloadingentscheidung verwendet. Pro Methode, für die ein Offloading stattfinden soll, wird eine eigene Registrierung beim Controller vorgenommen.

Anforderung 20: Erfassung der Ausführungszeit einer Methodenausführung

Es soll darauf hingewiesen werden, dass hier nicht im Detail erläutert wird, **wie** eine Aufzeichnung konkret gemacht werden kann. Das hängt vom tatsächlich verwendeten Offloadingframework ab. In Kapitel 5 wird beispielsweise eine Aufzeichnung auf JVM Ebene angenommen.

Zu einer Aufzeichnung gehören die aktuellen Parameter und die Ausführungszeit. Aufgezeichnet wird nur dann, wenn eine lokale Ausführung stattfindet. Die Aufzeichnungsstrategie, also in welchem Umfang Aufzeichnungen durchgeführt werden, hängt vom konkreten Offloadingframework ab.

Anforderung 21: Transformation von Methodenparametern

Werden komplexe Datentypen wie Listen oder ähnliches verwendet, sollten diese für das Vorhersagemodell insoweit aufbereitet werden, damit diese sinnvoll zur Vorhersagegenauigkeit beitragen können.
Beispiel: Es soll bei einer Anwendung zur Gesichtserkennung anhand von Bildern eine Methode ausgelagert werden, welche aus dem Rohbild Merkmale extrahiert. Es würde wenig Sinn machen, das komplette Bild für die Erstellung des Vorhersagemodelles zu verwenden, da jedes Pixel ein Feature darstellen würde. Dadurch hätte man allein schon bei einem kleineren Bild von 512x512 Pixeln circa 262.000 Features, die von den Algorithmen verarbeitet werden müssten. Da die Laufzeit der Merkmalsextraktion hauptsächlich von der Größe des Bildes abhängt, könnte man diese als Features verwenden. Wie man in Kapitel 6 sehen kann, reicht für dieses Beispiel ein einzelnes Feature aus, um eine ausreichend gute Vorhersage über die Ausführungszeit zu treffen.
Es wird angenommen, dass dem Client eine Methode zur Aufbereitung der Parameter bereitgestellt wird.

Anforderung 22: Erfassen eines Samples

Ein Sample besteht aus den aufgezeichneten Parameterwerten und der Ausführungszeit eines Offloadingkandidaten.
Es muss eine eindeutige Zuordnung der Aufzeichnung in Bezug auf die Methode der Anwendung möglich sein. Hierfür wird die in Abschnitt 4.2.1 beschriebene *Method-ID* verwendet.

Anforderung 23: Registrierung einer Methode beim Controller

Möchte ein Client eine Methode auslagern und dafür ein Vorhersagemodell erhalten, registriert sich dieser beim Controller. Für die Registrierung ist eine lokal generierte *Method-ID*, sowie die Namen der Features , die später für die Vorhersage verwendet werden, nötig.

Anforderung 24: Abmeldung einer Methode beim Controller

Sollen keine Modelle mehr für eine Methode empfangen werden, findet eine Abmeldung beim Controller statt. Hierzu wird der die *Method-ID* als Referenz verwendet.

Anforderung 25: Vorhersage der Ausführungszeit

Auf Seiten des Clients soll die Vorhersage der Ausführungszeit stattfinden. Konkret findet beispielsweise im Framework von Berg [BDR14a] die Entscheidung auf JVM-Ebene innerhalb der Anwendung statt. Denkbar wäre aber auch die Vorhersage in einer von der Anwendung unabhängigen Komponente, die abstrakt durch den Client dargestellt wird. Alternativ wäre es auch möglich, die Vorhersage in den Controller auszulagern. Diese Alternative wird im Abschnitt 4.4.4 diskutiert.

4.3. Metriken für die Ausführungszeit

4.3.1. Systemzeit

Naheliegend ist die Verwendung der lokalen Systemzeit zur Bestimmung der Ausführungszeit. Hierfür werden zwei Messpunkte t_{start}, t_{end} benötigt welche die aktuelle Systemzeit direkt vor und direkt nach der Methodenausführung aufzeichnen. Die Ausführungszeit berechnet sich durch $\delta_t = t_{end} - t_{start}$. Unterschiedliche CPU-Architekturen führen dazu, dass man pro Architektur für eine Methode ein separates Vorhersagemodell benötigen würde. Das Problem bei der Systemzeit ist, dass sie durch das Scheduling des Betriebssystems stark beeinflusst werden kann. Dieses Vorgehen funktioniert also nur unter der Annahme, dass während der Aufzeichnung keine Prozesse ausgeführt werden, die die CPU stark auslasten und die Ausführungszeit beeinträchtigen.

Verwendung eines Leistungsfaktors

Um nur noch einen Datensatz pro Architektur zu verwenden, kann eine konkrete Rechnerarchitektur als Referenz genommen werden. Datensätze anderer Architekturen werden anhand eines Benchmarks bewertet. Die relative Geschwindigkeit im Vergleich zur Referenzarchitektur kann verwendet werden, um Aufzeichnungen zusammenzuführen. Abbildung 4.3 veranschaulicht dieses Vorgehen. Um dies zu bewerkstelligen muss ein Benchmark bestehend aus mehreren Ausführungsszenarien sowohl auf dem Server als auch auf dem Client ausgeführt werden. Der Server muss dabei jedem neu registrierten Controller diesen Benchmark-Datensatz zur Verfügung stellen. Ein Faktor kann nun berechnet werden, indem der Client dieselben Testszenarien ausführt und die Dauer zur Ausführung mit dem Referenztestsatz vergleicht.

Im Entwurf wird nicht näher auf die Übertragung des Benchmarks und auf die Erstellung des Leistungsfaktors eingegangen. Es wird die Annahme getroffen, dass alle beteiligten Geräte dieselbe Architektur besitzen und die Ausführungszeiten nicht durch andere Prozesse beeinflusst werden.

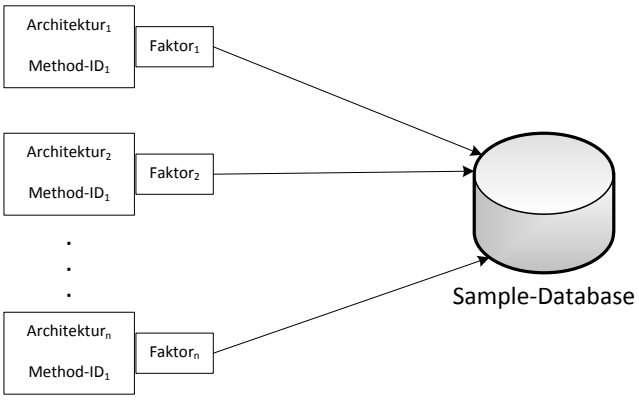


Abbildung 4.3.: Zusammenführung von Aufzeichnungen unterschiedlicher Architekturen anhand eines Leistungsfaktors.

4.4. Entwurf

Im diesem Abschnitt wird aus den definierten Anforderung an die Komponenten ein Entwurf abgeleitet. Die Komponenten werden einzeln erläutert und anschließend Ausführungsszenarien betrachtet, die die Interaktionen zwischen den Komponenten darstellen.
Auf die Alternativen aus Abschnitt 4.4.4 wird ebenfalls eingegangen.

4.4.1. Server

Der Server hält eine Datenstruktur zur Registrierung von Controllern. Jede registrierte Methode eines Controller bekommt Updates für Vorhersagemodelle. Beim Server werden Samples gesammelt und anhand dieser neue Vorhersagemodelle erstellt. Neue Vorhersagemodelle werden an die registrierten Controller zurückgeschickt.

Der *Prediction Server* besteht aus folgenden Komponenten:

- **Message Handler**: Dient zum Versenden und Empfangen von Nachrichten.
- **RegistrationDB**: Hier werden die aktuell registrierten Controller vermerkt. Eine Registrierung besteht aus <ID> : <IP/Port>.
- **Build-Scheduler**: Ist für des Scheduling der Modellerstellungen zuständig.
- **Model-Builder**: Erstellt die Vorhersagemodelle anhand der Daten aus der *Registration-DB*.
- **PredictionDB**: Zentrale Datenbank, die alle Samples und die dazugehörigen Vorhersagemodelle verwaltet.

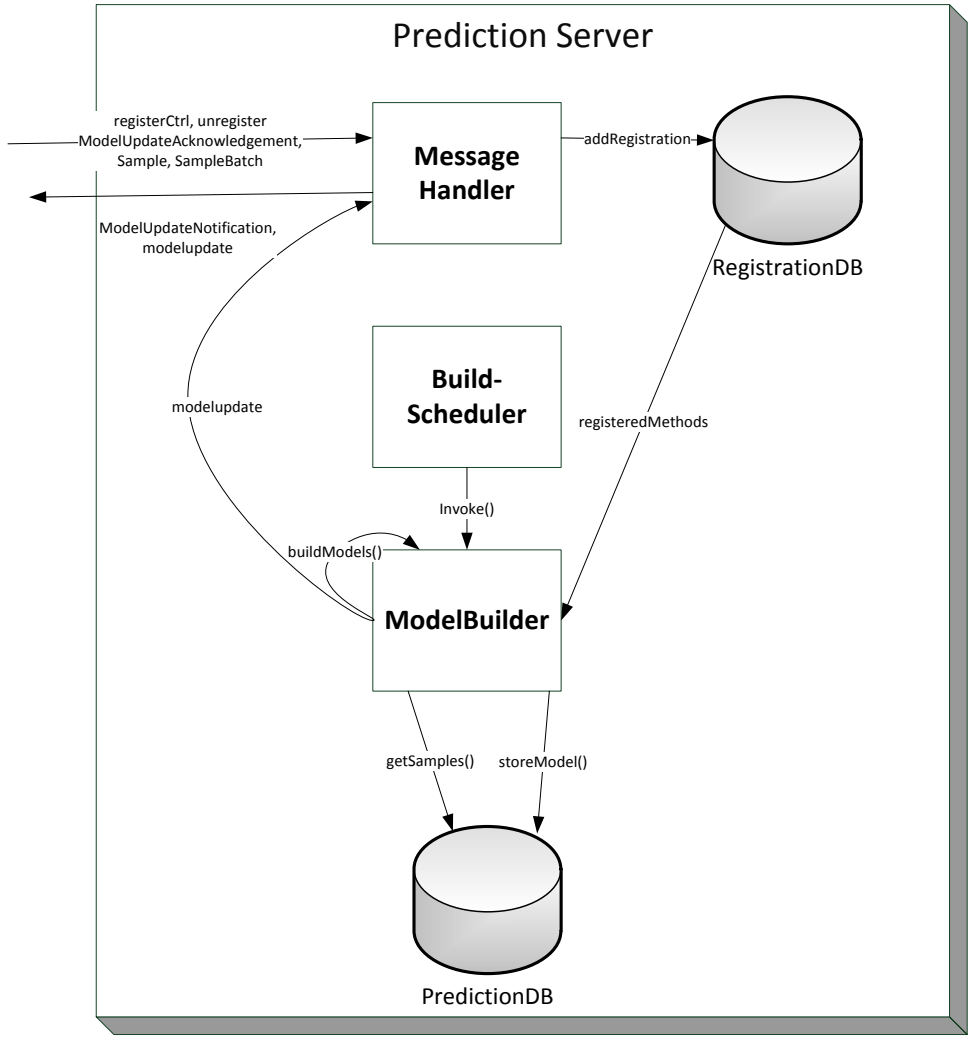


Abbildung 4.4.: Komponente “Server”.

Bei einer hineinkommenden Registrierungsnachricht nimmt der MessageHandler diese an und legt sie in der RegistrationDB ab. Falls bereits einer Registrierung zu dieser Methode für diesen Controller stattfand, wird die Nachricht ignoriert.
Bei einer erstmaligen Registrierung wird - sofern vorhanden - eine Benachrichtigung für den Erhalt des aktuell vorhanden Vorhersagemodelles verschickt. Der Build-Scheduler triggert anhand einer bestimmten Strategie, die in Anforderung 3 definiert wurde, den Start des Model-Builders.
Der Model-Builder liest aus der RegistrationDB alle Method-IDs aus, welche aktuell für ein Update registriert sind. Für diese Methoden werden im Anschluss aus der PredictionDB alle dazugehörigen Samples ausgelesen und daraus ein Vorhersagemodell erstellt. Dieses neu erstellte Modell wird wieder in der Datenbank abgespeichert. Anschließend werden alle

registrierten Controller benachrichtigt. Für jede positive Bestätigung wird das Modell an den jeweiligen Controller geschickt.

4.4.2. Client

Als *Client* wird die Komponente bezeichnet, die für einen Offloadingkandidaten eine Vorhersage der Ausführungszeit haben möchte, um die Offloadingentscheidung zu treffen. Die Integration des Clients kann dabei durch die Verwendung des Clients als Service oder Komponente stattfinden. Dieser stellt der Entscheidungskomponente die Vorhersage der Ausführungszeit bereit. Folgende Komponenten lassen sich im ClientService identifizieren:

- **Record Interface**: Ist zuständig, die Ausführungszeit aufzuzeichnen.
- **Messaging Interface**: Ist für das Versenden und Empfangen von Nachrichten zuständig.
- **Prediction Interface**: Trifft über das Vorhersagemodell anhand der aktuellen Parameter eine Vorhersage zur Ausführungszeit.

Wenn das *Record Interface* eine Aufzeichnung gemacht hat, wird dieses Sample an das Messaging Interface weitergeleitet, welches das Sample an den Controller schickt. Das *Prediction Interface* gibt die Vorhersage für die Ausführungszeit für die aktuelle Parametrisierung zurück.

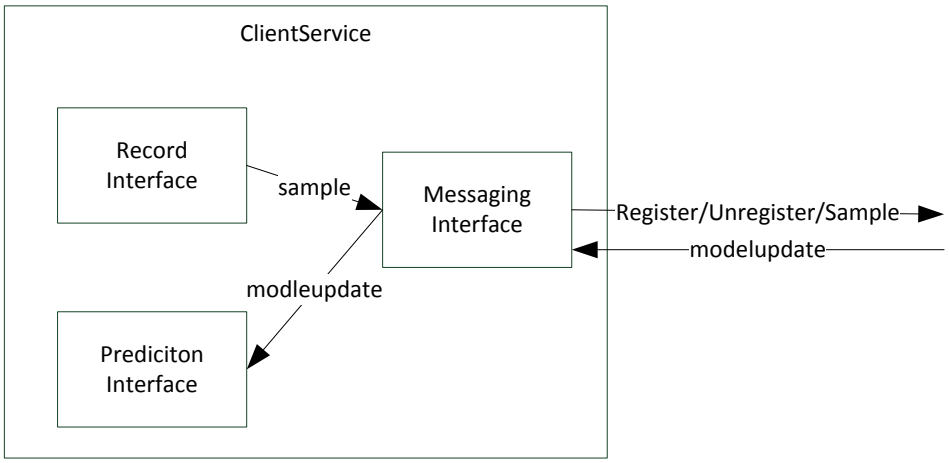


Abbildung 4.5.: Komponente “Client”.

Der Client stellt die Anwendung auf einem Gerät dar, welche eine oder mehrere Methoden auslagern möchte. Der Ablauf im Client zur Verwendung eines Vorhersagemodelles ist folgender:

- Pro Offloadingkandidat muss eine Registrierung beim Controller erfolgen.

- Für die lokalen Ausführungen der Offloadingkandidat werden die Ausführungszeiten und die Parameterwerte dem Controller bereitgestellt.
- Für die Vorhersage der Ausführungszeit verwendet der Client das Vorhersagemodell, das der Controller bereitgestellt hat.

4.4.3. Controller

Der Controller dient als Zwischenschicht zwischen Clients und der Serverseite. Der Controller leitet Registrierungen an den Server weiter und empfängt Updates von Vorhersagemodellen. An den Client werden die Updates weitergeleitet, welche dann für eine Vorhersage der Ausführungszeit verwendet werden können.

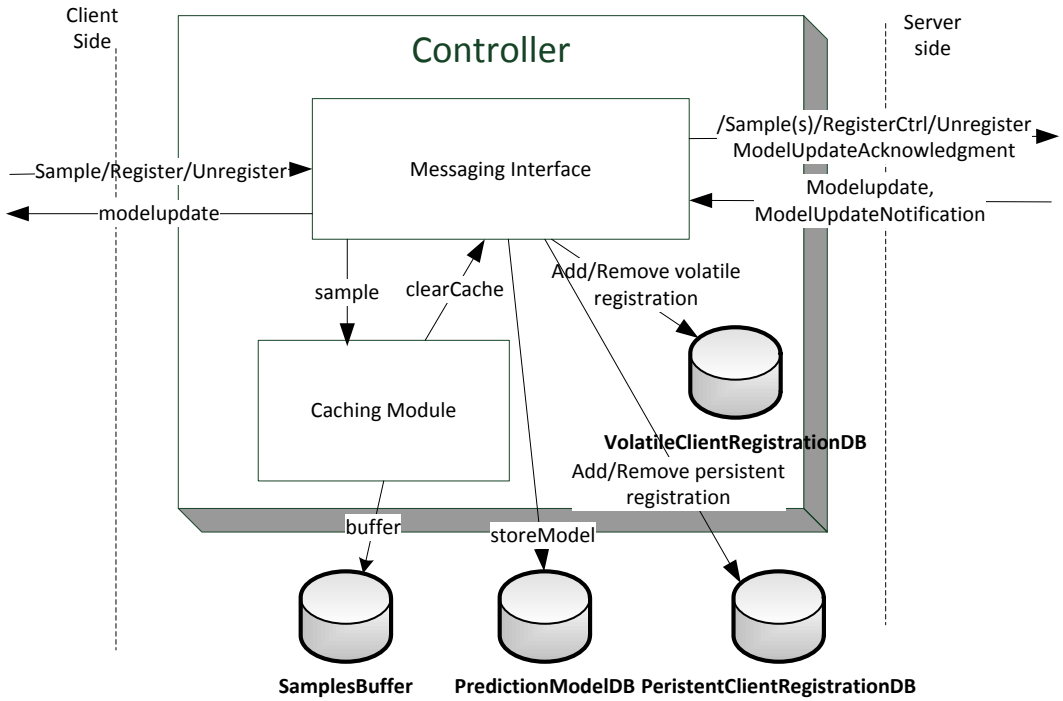


Abbildung 4.6.: Struktur der Komponente “Controller”

Auf dem *Controller* lassen sich folgende Komponenten identifizieren:

- **Messaging Interface**: Dient zum Erhalten von Nachrichten vom Client und zum Weiterleiten von Nachrichten an den Server.

- **Caching Module**: Das Caching Module sammelt Samples von Clients und schickt diese gebündelt an den Server weiter.
- **VolatileClientRegistrationDB**: Hier werden die registrierten Clients auf dem Mobilgerät gehalten.
- **PersistantRegistrationDB**: Stellt die persistente Haltung von Registrierungen dar. Registrierungen, die als persistent oder timeout-basiert markiert sind, werden in die *PersistantRegistrationDB* gespiegelt.
- **PredictionModelDB**: Hier werden die erhaltenen Vorhersagemodelle persistent gespeichert.
- **SamplesBuffer**: Hier werden der Sample-Cache persistent gehalten.

Das Messaging Interface verarbeitet Nachrichten (Neue Samples, eine Registrierung oder eine Abmeldung) vom Client, sowie auch Nachrichten vom Server (Updatebenachrichtigungen, Updates). Registrierungen werden an den Server weitergeleitet, wohingegen neue Samples im Caching Module gesammelt werden. Der Cache wird anhand eines in Anforderung 13 definierten Modus geleert.

4.4.4. Alternativen

Alternativ kann statt auf Clientseite die Vorhersage auch im Controller stattfinden. Abbildung 4.7 veranschaulicht das Alternative Design der Controller-Komponente.

Dazu muss der Client dem Controller die Rohparameter oder die aufbereiteten Parameter zur Verfügung stellen.

Abbildung 4.8 zeigt den Ablauf der Vorhersage des alternativen Designs.
Stellt der Controller die Möglichkeit zur Vorhersage bereit, muss vom Client lediglich über eine Kommunikationsschnittstelle die Übertragung der Rohparameter an den Controller stattfinden. Die Aufbereitung und die Vorhersage finden anschließend dort statt und das Ergebnis wird wieder an den Client zurückgegeben.
Durch diese Auskopplung allerdings entsteht bedingt durch die Interprozesskommunikation eine höhere Latenz bis zur Entscheidungsfindung.

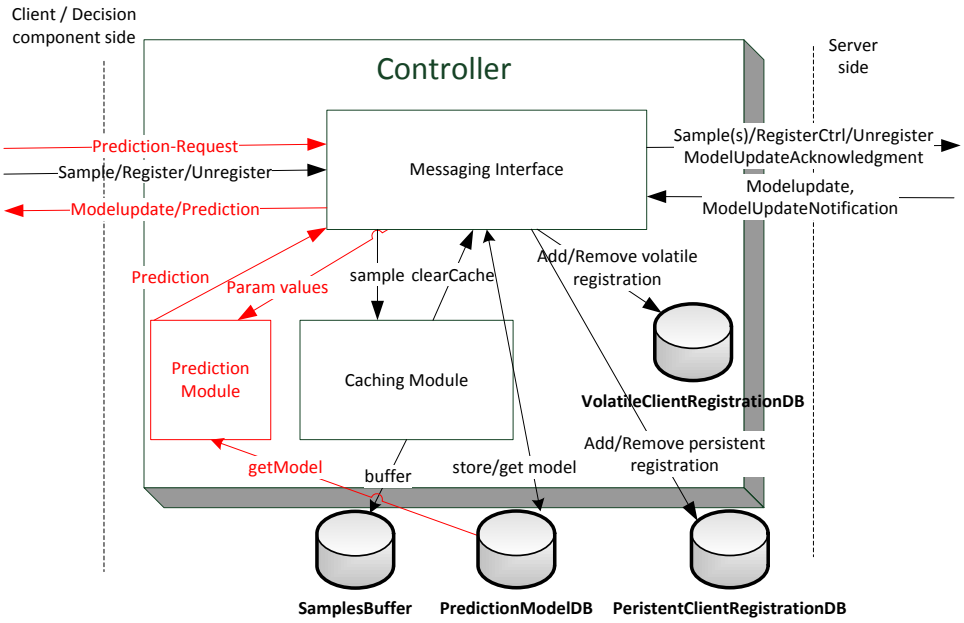


Abbildung 4.7.: Alternatives Design für die Controller-Komponente.

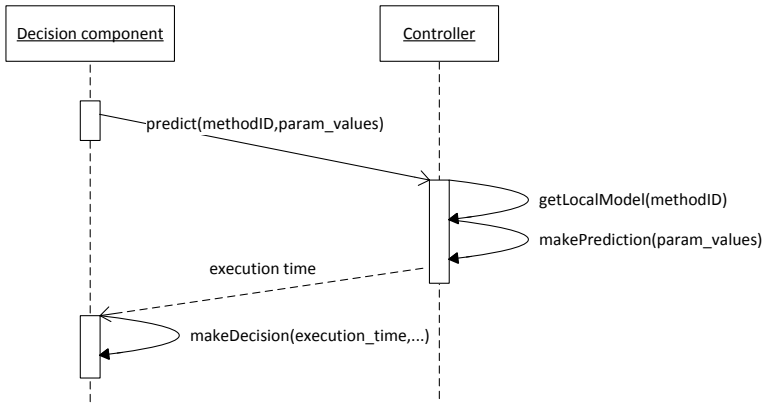


Abbildung 4.8.: Sequenzdiagramm: Erstellen der Vorhersage auf Serverseite und updaten der Clients.

4.4.5. Ausführungsszenarien

Client: registriert sich

Folgend werden zwei Registrierungsszenarios betrachtet: Abbildung 4.9 zeigt den Ablauf der Registrierung, wenn kein lokal gespeichertes Modell verfügbar ist.

Für die Registrierung schickt der Client zunächst eine *Register*-Nachricht an den Controller. Dieser überprüft zunächst, ob lokal ein Modell verfügbar ist. Da dies nicht der Fall ist, schickt der Controller an den Server eine *RegisterCtrl*-Nachricht an den Server. Der Zählerwert wird dabei auf 0 gesetzt. Das Zurückschicken des Modelles wird im nächsten Abschnitt dargestellt.

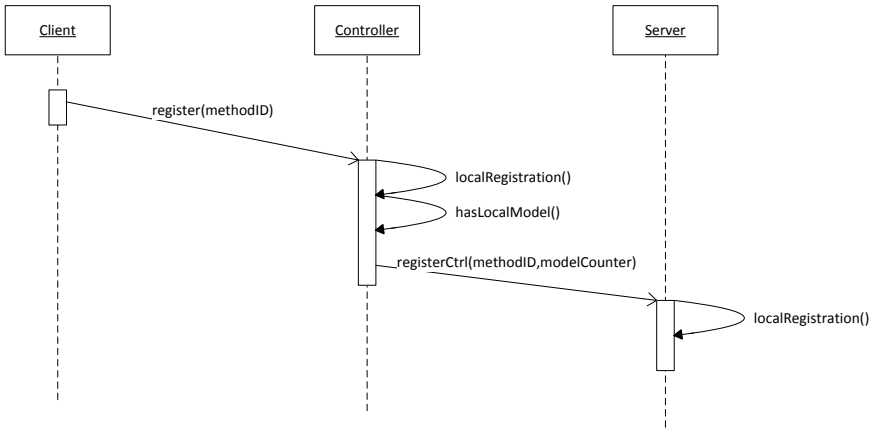


Abbildung 4.9.: Sequenzdiagramm: Registrierung eines Clients ohne lokal gespeichertes Modell.

Abbildung 4.10 zeigt den Ablauf mit lokal gespeichertem Modell. Für die Registrierung schickt der Client zunächst eine *Register*-Nachricht an den Controller. Dieser überprüft zunächst, ob lokal ein Modell verfügbar ist. Da dies der Fall ist, überträgt der Controller an den Client das persistent gespeicherte Modell. An den Server wird eine *RegisterCtrl*-Nachricht mit dem aktuellen Zählerwert des Modelles geschickt. Das Zurückschicken des Modelles wird im nächsten Abschnitt dargestellt.

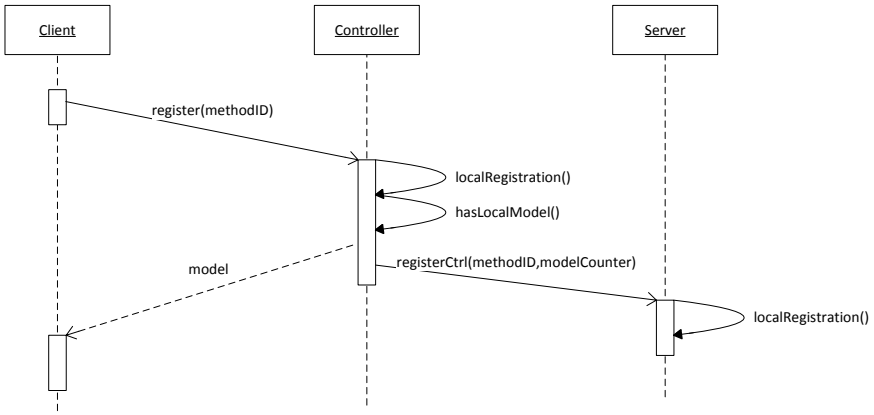


Abbildung 4.10.: Sequenzdiagramm: Registrierung eines Clients mit lokal gespeichertem Modell.

Server: Modellerstellung

Abbildung 4.11 zeigt den Ablauf zum Verschicken eines neuen Vorhersagemodelles an den Client.

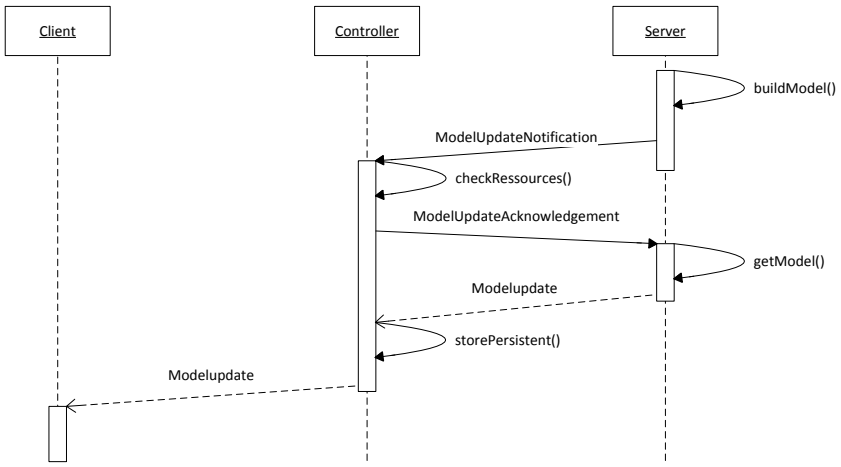


Abbildung 4.11.: Sequenzdiagramm: Erstellen der Vorhersage auf Serverseite und updaten der Clients.

Hat der Server ein neues Modell erstellt, schickt dieser zunächst eine *ModelUpdateNotification*-Nachricht an den Controller. Dieser entscheidet abhängig der aktuellen Bedingungen, ob ein Update erhalten werden soll - hier wird davon ausgegangen, dass es erhalten werden soll. Der Controller schickt nun eine *ModelUpdateAcknowledgement*-Nachricht an den Server. Dieser schickt nun das neue Modell an den Controller. Dieser speichert es zunächst persistent ab und überträgt es im Anschluss an den Client.

4.5. Zusammenfassung

Um Machine Learning sinnvoll für das Code Offloading verwenden zu können, wurde in diesem Kapitel ein kooperativer Entwurf vorgestellt, mit dessen Hilfe Samples in der Anwendung aufgezeichnet, auf dem Mobilgerät gesammelt und zum Server übertragen werden können. Der Entwurf ist dabei nicht als standalone-Lösung konzipiert worden, sondern soll bestehende Frameworks erweitern. Der Server ist für die Erstellung und Zusammenführung verschiedener Samples verantwortlich und schickt erstellte Vorhersagemodell an die Clients zurück. Dabei wurde während des Entwurfes festgestellt, dass es für einige Probleme unterschiedliche Lösungsalternativen gibt, die je nach verwendetem Framework Anwendung finden könnten. Im nächsten Kapitel wird eine prototypische Implementierung des Entwurfs zu einem konkreten Framework vorgestellt. Dabei wird eine Teilmenge der Designalternativen implementiert.

5. Implementierung

5.1. Einleitung

In Kapitel 3 wurden Machine Learning Methoden vorgestellt, welche für die Vorhersage der Ausführungszeit verwendet werden können.
Im Kapitel 4 wurde der kooperative Systementwurf vorgestellt der verwendet werden kann, um die Offloadingentscheidung bei bestehenden Frameworks mit der Vorhersage von Ausführungszeiten zu verbessern.
Dieses Kapitel wird sich um die Implementierung des Systementwurfs und die Anwendung von Machine Learning Algorithmen drehen. Zunächst wird der Ist-Zustand und die Rahmenbedingungen für die Implementierung betrachtet. Anschließend werden die verwendeten Frameworks und Werkzeuge vorgestellt. Danach werden die implementierten Komponenten, die in Kapitel 4 konzipiert wurden, beschrieben.

5.1.1. Ist-Zustand und Rahmenbedingungen

Die vorgestellte Implementierung erweitert das von Berg [BDR14a] entwickelte Offloading-System. In diesem Abschnitt wird das System kurz vorgestellt und die daraus resultierenden Rahmenbedingungen für die Implementierung erläutert.
Das Offloading-System besteht aus drei Hauptkomponenten, die alle auf Java basieren: Dem Mobilgerät, einem Offloading-Server und einem Code-Server. Java wurde aufgrund der plattformunabhängigkeit des Byte Codes verwendet, da somit Ausführungscode architekturunabhängig verwendet werden kann.
Offloading Server hosten JVMs , welche die ausgelagerten Methoden der Anwendungen ausführen. Der Code-Server speichert die auszulagernden Byte Codes und stellt sie dem Offloading Server bereit. Dadurch muss nicht bei jeder Auslagerung erneut der Code an den Server übertragen werden.
Offloading-Kandidaten werden mit Hilfe von Annotationen [AGHH00] identifiziert. Vereinfacht ist der Ablauf für das Offloading wie folgt: Startet die Anwendung, wird ein *Prepare Request* an den Offloading Server geschickt, welcher diesen dazu veranlasst, den Anwendungscode vom Code-Server herunterzuladen und eine JVM für die stellvertretende Ausführung vorzubereiten. Will das Mobilgerät einen Kandidaten auslagern, schickt er einen *Offloading-Request* an den Offloading Server. Dieser beinhaltet die Parameterwerte der Methode. Der Offloading Server startet die Ausführung der Methode und schickt anschließend die Ergebnisse an das Mobilgerät

zurück. Dieser Ablauf ist stark vereinfacht dargestellt und beschreibt weder den Safe-Point Mechanismus oder Fehlerfälle bei Verbindungsabbrüchen.

Der Entwurf, wie auch die Implementierung in diesem Kapitel, setzen an dem Punkt an, an dem das Mobilgerät den Offloading-Request an den Offloading-Server schicken möchte. Hier soll die Implementierung dafür verwendet werden können, die Ausführungszeit anhand der aktuellen Parameter des Offloadingkandidaten vorherzusagen. Abbildung 5.1 veranschaulicht

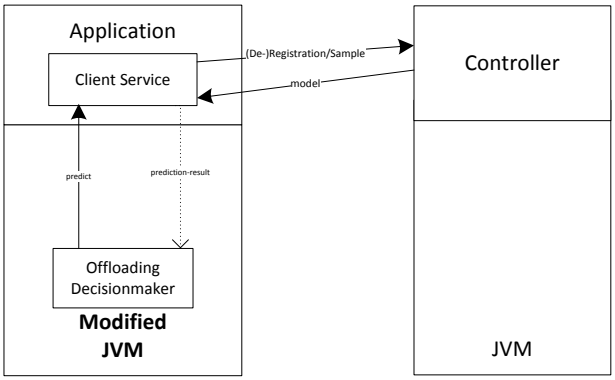


Abbildung 5.1.: Integration der Client-Komponente in das Offloadingframework von Berg.

die Integration der Client-Komponente des Entwurfs. Da das Framework komplett auf Java basiert, soll hierzu direkt auf Clientseite eine Vorhersage stattfinden. Da die Anwendung in einer modifizierten JVM ausgeführt in der Offloadingentscheidung getroffen wird, sollte an dieser Stelle auch die Vorhersage stattfinden.
Praktisch kann dies so umgesetzt werden, dass der ClientService, welcher die Machine Learning Modelle zur Vorhersage verwendet, über einen Reflection Call [FFI04] für die Vorhersage angesprochen werden kann. Der Anwendungsentwickler muss außerdem eine Java-Methode bereitstellen, welche die Parameterwerte aufbereitet. Anzumerken ist, dass die Integration in das Framework von Berg in dieser Implementierung nicht durchgeführt wurde.

5.1.2. Übersicht

Die Implementierung erfolgte komplett in Java mit Hilfe der aktuellen Eclipse Version Mars [Ecl].

Da es sich um eine prototypische Implementierung handelt wurden nicht komplett alle Anforderungen und Alternativen aus dem Systementwurf von Kapitel 4 implementiert.

Für die Clientkomponente wurden folgende Dinge nur eingeschränkt implementiert:

- **Vorhersage der Ausführungszeit**: Die Vorhersage kann lediglich im Client stattfinden.

- **Aufbereitung der Parameter**: Die Aufbereitung der Parameter findet direkt im Client statt.
- **Registrierungsmodi**: Es wurde der normale Modus implementiert. Bevor die Anwendung schließt, muss explizit eine Abmeldungsnachricht an den Controller geschickt werden.

Für die Serverkomponente wurden folgende Dinge nur eingeschränkt implementiert:

- **Erstellung der Vorhersagemodelle**: Es wurde eine periodische Erstellung unabhängig der gesammelten Samples implementiert.
- **Updaten der Vorhersagemodelle**: Hierfür wurde die Optimierung über Benachrichtigungen nicht implementiert. Der Server schickt immer direkt an den Controller das neue Vorhersagemodell.

Für die Controllerkomponente wurden folgende Dinge nur eingeschränkt implementiert:

- **Caching**: Das Caching für Samples und Vorhersagemodelle wurde nicht persistent implementiert.
- **Leeren des Caches**: Das Leeren des Caches wird ausschließlich bei Überschreitung eines Schwellwertes ausgeführt.

5.2. Verwendete Frameworks und Kommunikationsmechanismen

5.2.1. Weka

Als Machine Learning Framework wurde Weka [HFH+09] ausgewählt. Weka ist ein in der Waikato University entstandenes Open Source Projekt, welches komplett in Java geschrieben wurde. Neben bekannten Regressions - und Klassifikationalgorithmen werden ebenfalls Möglichkeiten für das Preprocessing der Daten über sogenannte Filter bereitgestellt. Das Datenmodell wurde im sogenannten ARFF-Format(*Attribute-Relation File Format*) definiert. Zusätzlich bietet Weka die Möglichkeit Parameter von Algorithmen mit Hilfe verschiedener Metaalgorithmen, die in Kapitel 3.3.4 beschrieben worden sind, zu optimieren. Zum Evaluieren können Testdatensätze und alle gängigen Metriken aus Kapitel 3.3.5 verwendet werden.

Datenhaltung

ARFF [BFH+15, Kapitel 10] ist ein plaintext Datenformat, welches in unterschiedliche Abschnitte zusammengefasst werden kann. Tags, die zeilenweise definiert werden, werden mit dem Sonderzeichen "@" eingeleitet. Folgende Tags werden verwendet:

- *@relation*: Wird verwendet um den Datensatz zu benennen.
- *@attribute*: In Weka wird ein Feature als *Attribute* bezeichnet und ein Feature-Vektor als *Instance*. Im ARFF Format werden zunächst in einem Header-Abschnitt mit Hilfe der @attribute Tags die Attribute und deren Datenart bestimmt, wie zum Beispiel numerisch oder kategorisch. Numerische Attribute werden mit *numeric* definiert. Kann ein Attribut verschiedene Klassen annehmen, kann man dies über durch die Verwendung von einer Liste $\{ Klasse_1, Klasse_2, ..., Klasse_n \}$ nach dem @attribute tag definieren.
- *@data*: Nach der Verwendung dieses Tags folgt der Datenabschnitt. Im Datenabschnitt werden die Feature-Vektoren kommasepariert in der Reihenfolge aufgelistet, in der sie zuvor über die @attribute tags definiert worden sind.

Preprocessing der Daten

Weka bietet die Möglichkeit Daten vor dem Erstellen des Vorhersagemodelles zu modifzieren. Dies kann durch die Verwendung sogenannter *Filter* [BFH+15, Kapitel 17.4] gemacht werden. Zum Einen ist es möglich die Menge und Reihenfolge eines Datensatzes zu ändern oder in mehrere Teilmengen aufzuteilen. Zum Anderen ist es möglich einzelne Attribute zu transformieren. In der Implementierung wird eine Transformation, wie sie in Abschnitt 3.3.3 beschrieben ist, verwendet.

Classifcator

Abstrakt wird ein Vorhersagemodell in Weka *Classificator*[BFH+15, S. 17.7] genannt. Die Schnittstelle bietet Methoden zum Vorhersagen einzelner Instanzen. Außerdem können die erstellten Modelle serialisiert auf persistentem Speicher abgelegt werden.

Optimierung von Hyperparametern

Um Parameter, wie in Abschnitt 3.3.4 beschrieben, zu optimieren, stellt das Framework die Gittersuche [WF05, Kapitel 11.5] zur Verfügung, sowie eine vereinfachte Version, welche für die Optimierung von einem Parameter verwendet werden kann

5.3. Implementierung: Nachrichten

Die Nachrichten welche über UDP verschickt werden, werden als Java-Klassen definiert. Jede Nachricht besitzt eine Method-ID und eine UUID, die für die Serialisierung verwendet wird, welche automatisch erzeugt wurde.

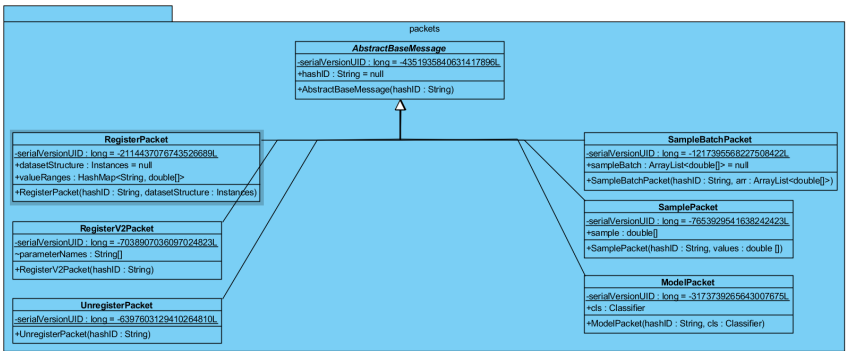


Abbildung 5.2.: Aufbau der Kontroll- und Datennachrichten. Konkrete Nachrichtentypen werden von der *AbstractBaseMessage* abgeleitet, welche als Referenz die Method-ID verwendet.

Abbildung 5.2 zeigt das zugehörige UML Diagramm. Folgende Nachrichten wurden implementiert:

- RegistrierPacket: Durch dieses Paket findet eine Anmeldung des Clients beim Controller oder des Controllers beim Server statt. Zusätzlich zur Method-ID aus der AbstractBaseMessage werden die Namen der für die Vorhersage benötigten Features übergeben.
- UnregisterPacket: Durch dieses Paket findet eine Abmeldung des Clients beim Controller oder des Controllers beim Server statt. Dadurch bekommt der abgemeldete Client/Controller keine Updates für das Modell mehr, das mit der Method-ID referenziert wurde.
- SamplePacket: Mit diesem Paket wird ein aufgezeichnetes Sample vom Client zum Controller geschickt.
- SampleBatchPacket: Ist ein Caching erwünscht nutzt der Controller dieses Paket, um gebündelt Samples an den Server zu schicken.
- ModelPacket: Mit diesem Paket wird ein Vorhersagemodell verschickt. Das Vorhersagemodell wird dabei als serialisiertes Java-Objekt übertragen.

5.4. Implementierung: Machine Learning

Die verwendeten Regressionsalgorithmen werden in der Klasse “Regression” zusammengefasst, welche in Abbildung 5.3 als UML Diagramm veranschaulicht ist.

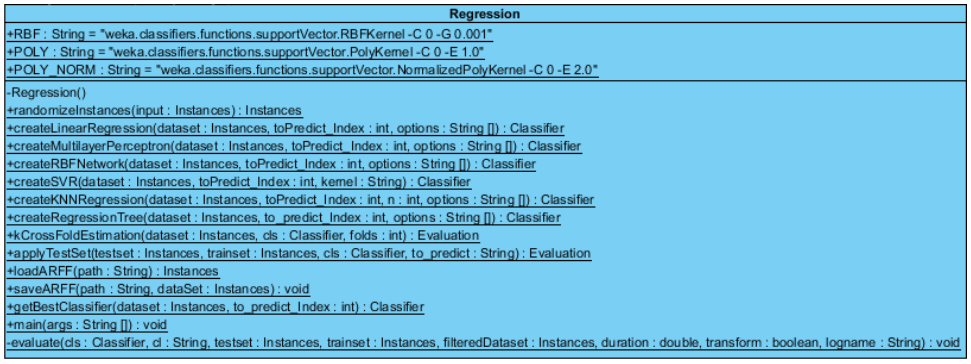


Abbildung 5.3.: Implementierung der Klasse “Regression”, welche die Erstellung der Vorhersagemodelle ermöglicht.

Es werden statische Methoden zur Erstellung und Validierung von Vorhersagemodellen bereitstellt.
Folgende Machine Learning Algorithmen wurden über das WEKA Framework verwendet:

- Linear Ridge Regression
- Support Vector Regression mit RBF Kernel (SVR RBF)
- K-Nearest-Neighbor (KNN) Regression
- Neurale Netze (RBF Network)
- Regression Trees

Listing 5.1 zeigt beispielhaft die Erstellung von eines Linear Regression Modelles.

```
public static Classifier createLinearRegression(Instances dataset, int
    toPredict_Index, String[] options ) throws Exception{
      String opts = "";
      if(options == null)
            opts = "-S 0 -R 0.000000001";

       LinearRegression lr = new LinearRegression();
       lr.setOptions(weka.core.Utils.splitOptions(opts));

  CVParameterSelection ps = new CVParameterSelection();
     ps.setClassifier(lr);
     ps.setNumFolds(10); // using 10-fold CV
     ps.addCVParameter("R 0.00000001 10000 5"); //test param "V" with startvalue
         0.0001 to 0.01 with 5 steps

     dataset.setClassIndex(toPredict_Index);

    // build best option
     ps.buildClassifier(dataset);
```

```
        return (Classifier) ps.getClassifier();

    }
```

Listing 5.1: Erstellung eines Linear Regression Vorhersagemodelles mit Hilfe des Weka Frameworks.

Modelle wurden mit Hilfe der Gittersuche, die in Abschnitt 3.3.4 beschrieben wurde, optimiert.

Zudem implementiert die Klasse die in Kapitel 3.3.3 erläuterte Transformation der Zielvariablen. Eine Methode zum Evaluieren eines Modelles anhand eines Testdatensatzes wurde ebenso implementiert. Hierzu kann ein auf dem Datenträger gespeichertes Modell verwendet werden. Tabelle 5.1 zeigt eine Übersicht der verwendeten Algorithmen und deren Parameter.

Tabelle 5.1.: Zusammenfassung der verwendeten Algorithmen und der verwendeten Parameter.

Algorithmus	Gittersuche	Parameter
Linear Ridge Regression	Ja	**Ridge**: Penalty-Parameter für die Koeffizienten (vgl. Kapitel 3.4.1)
SVR RBF	Ja	**C**: bestimmt die Größe des Soft-Margins. **Gamma**: Breite der Radial Basis Function.
KNN Regression	Nein	**k**: Anzahl der Nachbarn, die für die Vorhersage verwendet wurde. Es wurde k = 20 gewählt.
RBF Network	Nein	Standardparameter für k-means wurden gewählt.
Regression Trees (REPTREE)	Ja	**V**: Threshold für Splitting

5.5. Implementierung: Persistente Datenhaltung

Als Basis wurde eine abstrakte Klasse erstellt, welche die Aktionen für das Speichern und Laden von Daten definiert. Das Datenmodell muss in der Lage sein Samples, sowie Vorhersagemodelle, zu speichern und zu laden. Ebenfalls muss es möglich sein, die einzelnen Samples zu einer spezifischen Methode einer Anwendung zuzuordnen. Hierzu wird die *Method-ID*, die in Kapitel 4.2.1 definiert wurde, verwendet.
Abbildung 5.4 zeigt das UML Diagramm der implementierten Klassen.
Für die Datenhaltung wurden zwei unterschiedliche Implementierungen gewählt. Zum Einen wurde ein ARFF-Datenmodell implementiert und zum Anderen ein SQL-Datenmodell. Beim ARFF-Datenmodell wird für jede registrierte Methode eine ARFF-Datei angelegt. Als Dateiname

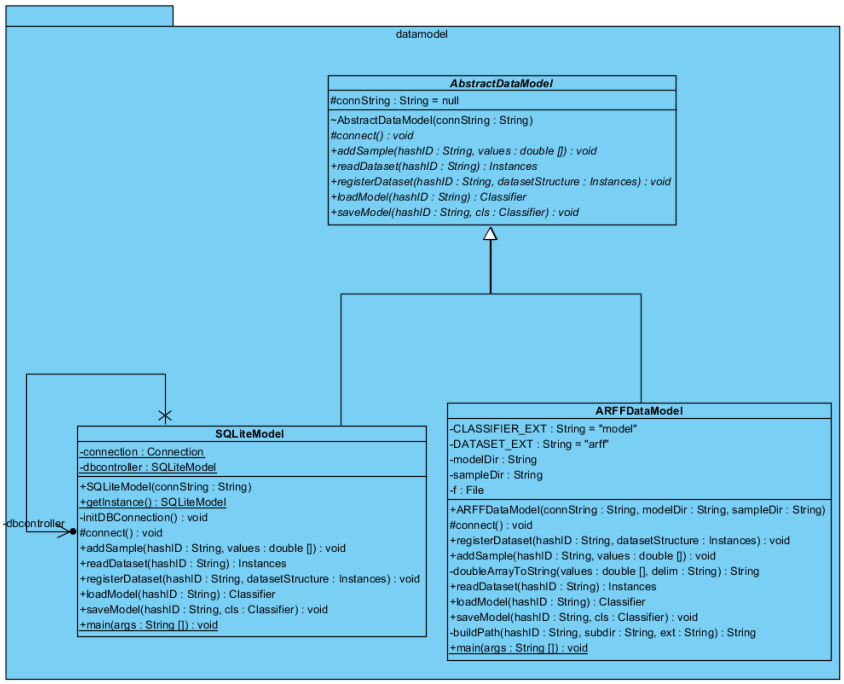


Abbildung 5.4.: Implementierung der persistenten Datenhaltung.

wurde jeweils der Hashwert gewählt und die Endung ".arff". Vorhersagemodelle werden als serialisierte Binärdateien gespeichert. Die Dateinamen der Vorhersagemodelle sind ebenfalls die jeweiligen Hashwerte und haben die Endung ".model".

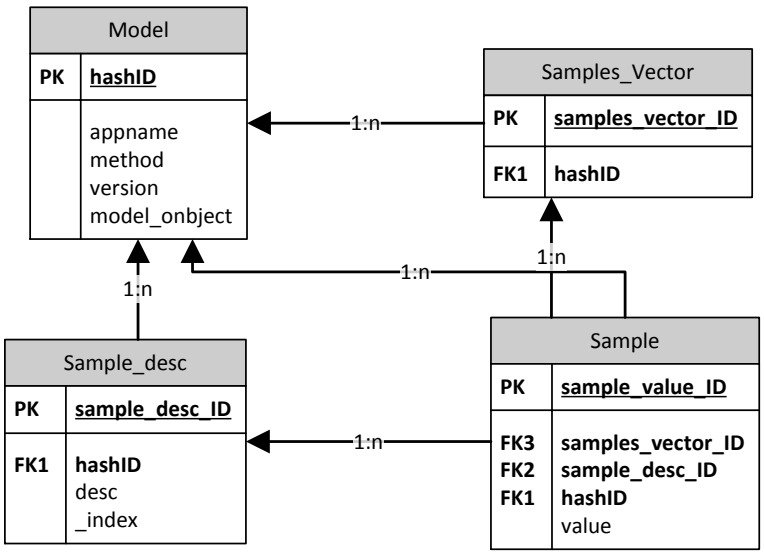


Abbildung 5.5.: Entity-Relationship Modell der SQLite Datenbank.

Für das SQL Datenmodell wurden folgende Tabelle, die im ER-Diagramm in Abbildung 5.5 modelliert sind, verwendet:

- Model: Diese Tabelle enthält neben textuellen Beschreibungen auch das Vorhersagemodell im Binärformat. Jedes Model besitzt n Einträge der Tabelle Samples_Vector. Für jeden Eintrag in der Model-Tabelle gibt es einen Eintrag in der Tabelle Sample_Dest.
- Samples_Vector: Diese Tabelle stellt ein Sample dar. Die Werte der Features werden in der Tabelle Sample gespeichert.
- Sample_Desc: Diese Tabelle modelliert die Features, die zur Erstellung des Vorhersagemodelles verwendet werden. Jeder Eintrag stellt die Beschreibung eines Features dar. Um die einzelnen Features später in der korrekten Reihenfolge auszulesen, wird zu jeder Featurebeschreibung ein - in Bezug auf jeweils eine Method-ID - eindeutiger Index versehen, der die Reihenfolge vorgibt, in der die Features gelesen werden.
- Sample: Jeder Eintrag in dieser Tabelle stellt einen Wert eines Features dar. Jeder Eintrag hat eine Referenz zu einem Eintrag in der Tabelle Samples_Vector.

5.6. Implementierung: Komponenten

Die Komponenten wurden jeweils als eigenständig lauffähige Java-Anwendungen implementiert. Als Kommunikationsmechanismus wurde das UDP Protokoll gewählt. Jede Komponente agiert sowohl als Empfänger, als auch als Sender von Nachrichten. Im Folgenden werden die Komponenten in Bezug auf den Entwurf des vorherigen Kapitels implementiert.

5.6.1. Client

Der Anwendungsentwickler kann mit Hilfe dieser Klasse Samples aufzeichnen und abschicken, sowie Vorhersagen über die Ausführungszeit anhand der aktuellen Methodenparameter treffen. Die Komponente Client wurde mit Hilfe von drei Klassen implementiert. Die Klasse “ClientService” stellt die Schnittstelle zum Anwendungsentwickler dar und realisiert das *Prediction Interface*. Innerhalb dieser Klasse wird das Vorhersagemodell gehalten und kann für Vorhersagen verwendet werden. Das *Record Interface* wurde durch die Bereitstellung von zwei Methoden *startExecRecord* und *endExecRecord* implementiert und ermöglicht es zu Testzwecken direkt in der Anwendung die Ausführungszeiten aufzuzeichnen.
Das *Messaging Interface* wurde mit Hilfe von zwei weiteren Klassen implementiert: “ModelReceiver” und “MessageSender”. Über “MessageSender” verschickt der ClientService Kontrollnachrichten und Samples. Die Klasse "ModelReceiver"wurde als Thread implementiert und erhält Nachrichten vom Controller. Wird ein neues Vorhersagemodell erhalten, wird dieses Update an die ClientService-Klasse propagiert.

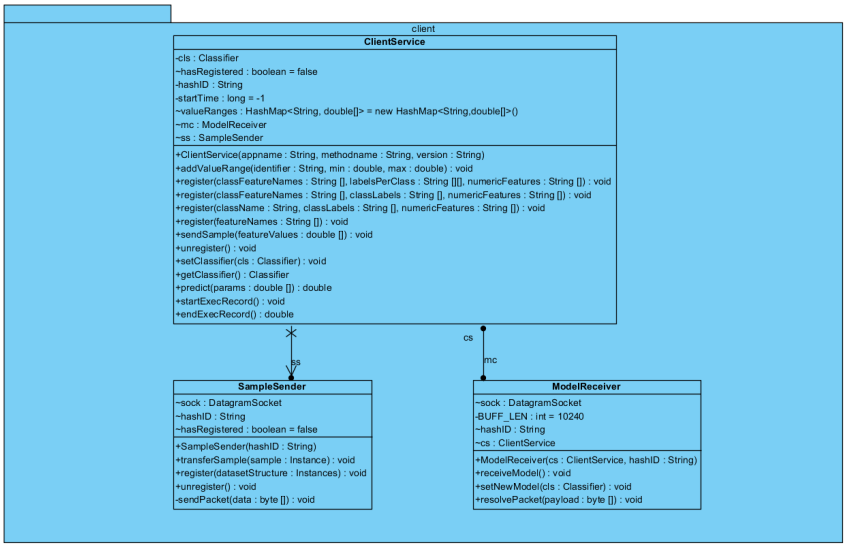


Abbildung 5.6.: Implementierung der Komponente “Client”.

5.6.2. Controller

Das *Messaging Interface* wurde direkt in der der Klasse *Prediction_Controller* als UDP-Server implementiert. Der Controller empfängt sowohl vom Client als auch vom Server Nachrichten und leitet diese Nachrichten weiter. Der Controller cached vom Client empfangene Samples und schickt diese an den Server erst bei Überschreiten eines benutzerdefinierten Schwellwertes. Dieses implementierte Verhalten realisiert den im Systementwurfs definierten Schwellwert-Modus zur Leerung von Caches.
Die in-memory Datenstruktur für die Registrierungen eines Clients wurde mit der Klasse “ClientRegistration” implementiert. Für die prototypische Implementierung wurde auf persistentes Caching verzichtet. Das *Caching* wurde ebenfalls in der Klasse *Prediction_Controller* realisiert. Für das Caching wurde der Schwellwert-Modus für die Implementierung gewählt. Der Controller erstellt pro registrierter Methode einen Puffer. Wird eine gewisse Pufferlänge gemessen in der Byte-Länge überschritten, wird er geleert und die Samples an den Server geschickt.

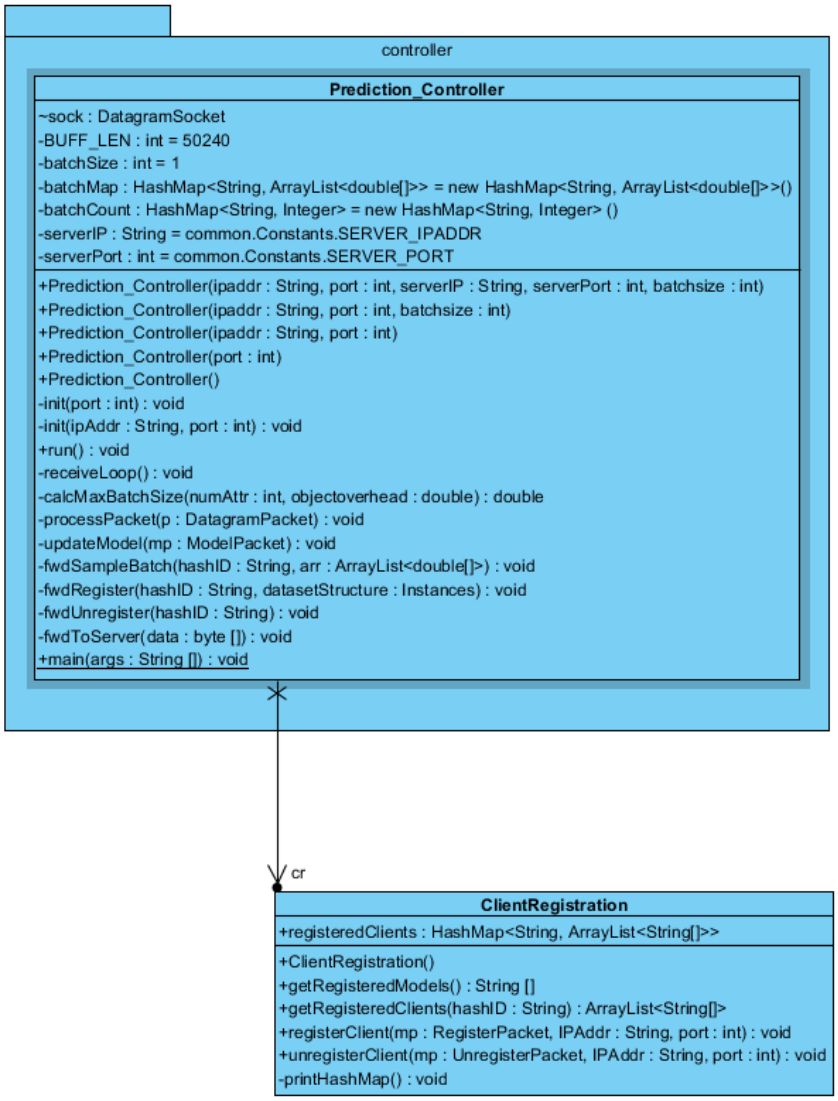


Abbildung 5.7.: Implementierung der Komponente “Controller”.

5.6.3. Server

Der *MessageHandler* wurde als UDP-Server in der Klasse *UDPPredictionService* implementiert. Erhaltene Nachrichten werden über die Methode *processPacket* verarbeitet. Samples und Modelle werden persistent über ein Datenbank-Controller-Objekt des Types *AbstractDataModel* verwaltet, welches im Abschnitt 5.5 erläutert wurde.
Der *Build-Scheduler* wurde mit der abstrakten Klasse *AbstractModelCreation* realisiert. Mit der konkreten Klasse *PeriodicModelCreator* ist es möglich, eine Modellerstellung periodisch anzustoßen. Für jede beim Server registrierte Methode werden die Samples über das verwendete Datenmodell ausgelesen und daraus ein Vorhersagemodell erstellt. Die Erstellung mit Hilfe des *Model-Builder* erfolgt über die Erzeugung eines Threadpools mit Objekten der Klasse

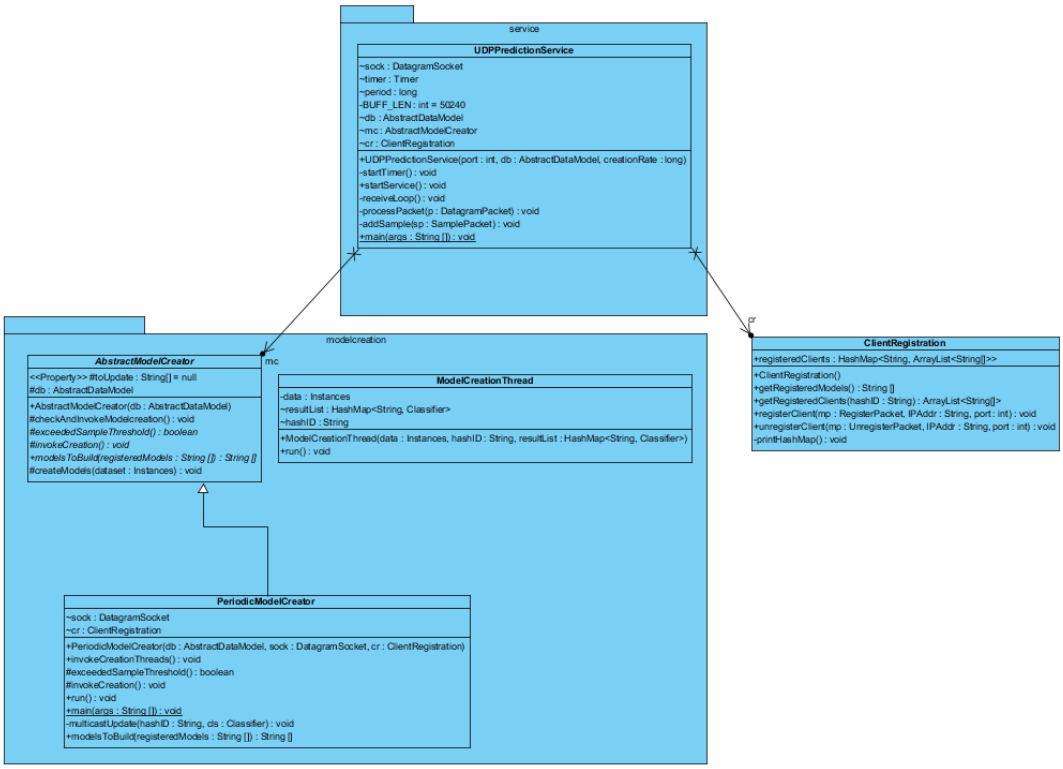


Abbildung 5.8.: Implementierung der Komponente “Server”.

ModelCreationThread. Diese erstellen mit Hilfe der Klasse *Regression* die Modelle. Dieses wird anschließend wieder persistent gespeichert und an alle registrierten Controller wird ein Update geschickt.

5.7. Zusammenfassung

Dieses Kapitel stellte die Implementierung einer Teilmenge des Systementwurfs aus dem vorherigen Kapitel vor. Als Grundlage für die Implementierung wurde dabei das Framework von Berg genommen. Die Implementierung kann dabei so in das Framework integriert werden, dass eine Vorhersage der Ausführungszeiten innerhalb der JVM der Client-Anwendung stattfinden kann. Die Machine Learning Algorithmen, die in Kapitel 3 eingeführt wurden, wurden mit Hilfe des Weka Frameworks angewandt und in einer Hilfsklasse für die Nutzung zusammengefasst. Die Kommunikation der Komponenten wurde über UDP realisiert. Für die Datenhaltung wurden zwei Alternativen implementiert: Zum Einen eine Textformat-Implementierung, welche das Textformat *ARFF* des Weka Frameworks verwendet und zum Anderen eine SQLite Implementierung. Auf Basis des Weka Frameworks soll nun im nächsten Kapitel untersucht werden, wie gut die Machine Learning Algorithmen für eine Offloadingentscheidung geeignet sind.

6. Evaluation

6.1. Einleitung

Das letzte Kapitel beschäftigte sich mit der Implementierung des kooperativen Systems zur Verwaltung und Erstellung von Vorhersagemodellen. Die in Kapitel 3 vorgestellten und in Kapitel 5.4 implementierten Machine Learning Algorithmen sollen in diesem Kapitel auf ihre Tauglichkeit für die Verbesserung der Offloadingentscheidung getestet werden.

Für die Evaluation wurden bei vier verschiedenen Testanwendungen Methoden als Offloadingkandidaten manuell ausgewählt. Die Methoden wurden anhand vorgefertigter Testfälle ausgeführt. Für jede Ausführung wird ein Sample erstellt. Ein Sample besteht aus den Parametern des Offloadingkandidaten, sowie der Zeit zum Ausführen des Kandidaten. Die Parameter der Methoden wurden teilweise aufbereitet.
Die Methoden haben dabei eine der folgenden Eigenschaften bezüglich ihrer Ein-und Ausgabe haben:

1. Konstante Eingabe und Ausgabe.
2. Kleinere Ausgabe als Eingabe.
3. Größere Ausgabe als Eingabe.

Anhand dieser Datensätze wurden mit verschiedenen Machine Learning Algorithmen Vorhersagemodelle erstellt. Um die Qualität der Vorhersagemodelle zu Bewerten wurden $\frac{2}{3}$ des Datensatzes als Trainingsdatensatz und $\frac{1}{3}$ als separater Testdatensatz verwendet. Anhand des Testdatensatzes werden Metriken aus Kapitel 3.3.5 erstellt. Die Leistung der Algorithmen in Bezug auf die Offloadingentscheidung wird anhand einer Variation der vorgestellten Gleichungen aus Kapitel 2 bestimmt.

Folgende Anwendungen wurden für die Evaluation verwendet.

- **Schach**: Konstante Eingabe und Ausgabe.
- **Gesichtserkennung**: Größere Eingabe als Ausgabe.
- **Rauschreduzierung**: Konstante Ein- und Ausgabe.
- **Text-To-Speech**: Größere Ausgabe als Eingabe.

Zusätzlich wird bei der Text-To-Speech Anwendung angenommen, dass die Ausgabegröße **nicht** bekannt ist. Dies ist beeinflusst die Offloadingentscheidung, da in diesem Fall das Herunterladen der Ausgabe nicht betrachtet wird.

6.2. Bewertungskriterien und Vorgehen

6.2.1. Leistung der Vorhersagemodelle anhand gängiger Metriken

Alle in Kapitel 3.4 vorgestellten Algorithmen werden anhand der Metriken aus 3.3.5 bewertet. Hierbei sollen die Metriken *Root Relative Squared Error* und *Relative Absolute Error* dargestellt werden.

Zum Einen soll dargestellt werden, wie sich die Metriken mit steigender Größe der Sampledatensätze verhalten. Der Testdatensatz wird dabei stets in Abhängigkeit der aktuellen Gesamtdatensatzgröße erstellt. Das heißt es werden jeweils unterschiedliche Testdatensätze verwendet: Praktisch ist der Fall gegeben, dass anfangs nur ein sehr kleiner Datensatz zum Einlernen der Modelle vorhanden ist. Es soll untersucht werden inwieweit man bei diesem Szenario eine Verbesserung der Metriken beobachten kann.

Zum Anderen soll für jedes Modell, das mit unterschiedlichen Trainingsdatensätzen eingelernt wurde, der komplette Testdatensatz - also $\frac{1}{3}$ vom kompletten Datensatz - verwendet werden. Es soll untersucht werden, ob die erstellten Metriken im ersten Fall auch bei wechselnden Testdatensätzen sinnvoll sind.

Zusammengefasst werden die Testfälle also folgendermaßen erzeugt:

1. Laden eines Testfalles aus der vorgenerierten Testmenge.
2. Ausführen des Testfalles und Aufzeichnen der Ausführungszeit.
3. Speichern der Parameterwerte und der Ausführungszeit.

Das Ergebnis ist der Datensatz zur Erstellung der Modelle.

Anschließend werden die Modelle folgendermaßen trainiert:

1. Laden des Datensatzes mit dem Weka Framework.
2. Ausgehend von einer initialen Größe eine Teilmenge des kompletten Datensatzes verwenden.
3. Aufteilung im Verhältnis 2:1 in Trainings- und Testdatensatz.
4. Erstellen des Modelles anhand des Trainingsdatensatzes.
5. Evaluieren des Modelles anhand des Testdatensatzes.

Das Ergebnis ist eine Menge an Modellen, die mit unterschiedlich großen Trainingsdatensätzen eingelernt wurden.

6.2.2. Leistung der Vorhersagemodelle anhand der Offloadingentscheidung

Die Gleichung 2.1 und 2.2 aus Kapitel 2 sollen für die Offloadingentscheidung verwendet werden.

Die Gleichung 2.1 wird dahingehend modifiziert, dass auch die Zeit zum Empfangen der Ergebnisse mit einbezogen wird:

$$\frac{w}{s_m} > \frac{d_i}{B_{up}} + \frac{w}{s_S} + \frac{d_o}{B_{down}} \tag{6.1}$$

wobei d_o die Größe des Ergebnisses in Byte und B_{down} der Downstream und B_{up} der Upstream der Verbindung ist.

Die Gleichung 2.2 wird dahingehend modifiziert, dass auch die Energie zum Empfangen der Ergebnisse zum Client betrachtet wird. Die modifizierte Gleichung lautet wie folgt:

$$p_m * \frac{w}{s_m} > p_C * \frac{d_i}{B_{up}} + p_i * \frac{w}{s_S} + p_r * \frac{d_o}{B_{down}} \tag{6.2}$$

wobei p_r die Energie für das Empfangen von Daten, d_o die Größe des Ergebnisses in Byte und B_{down} der Downstream und B_{up} der Upstream der Verbindung ist.

Für Gleichung 6.1 wird für den Energieverbrauch folgende Annahme getroffen [CH10; HQG+12]:

- Für p_m wurde aus [CH10] der Energieverbrauch der CPU und des RAMs unter Last (Abbildung 5, Testfall “equaqe”) mit dem Verbrauch de restlichen Komponenten im Idle-Modus (Abbildung 3) addiert.
- Für p_i wurde der Energieverbrauch des Idle-Modus übernommen.
- Für die benötigte Leistung in mW pro Mbps von 3G,LTE und 802.11 g wurden die Werte der Tabelle 4 aus [HQG+12] übernommen.

Tabelle 6.1 stellt eine Übersicht über die einzelnen Werte dar.

Um die Leistung eines Vorhersagemodelles zu bewerten, wird der komplette Testdatensatz (d.h. ausgehend vom kompletten Datensatz wird $\frac{1}{3}$ als Testdatensatz verwendet) der jeweiligen Testanwendung benutzt.

p_m	426 mW
p_i	268.8 mW
p_C (3G)	817,88 mW + n * 868,98 mW/Mbps
p_r (3G)	817,98 mW + n * 122,12 mW/Mbps
p_C (LTE)	1288,04 mW + n * 438,39 mW/Mbps
p_r (LTE)	1288,04 mW + n * 51,97 mW/Mbps
p_C (Wifi 802.11 g)	132,86 mW + n * 283,17 mW/Mbps
p_r (Wifi 802.11 g)	132,86 mW + n * 137,01 mW/Mbps

Tabelle 6.1.: Übersicht: Leistung von Smartphone Komponenten für 3G,LTE und Wifi für das Senden/Empfangen mit einer Bandbreite von n Mbps sowie die benötigte Leistung für einen Idle-Betrieb und unter Last. Der Faktor n stellt die Datenrate in Mbps dar.

Anhand der tatsächlichen Ausführungszeit wird bestimmt, ob ein Offloading stattfinden sollte. Dieses Ergebnis wird als Referenzwert benutzt. Nun wird das Vorhersagemodell auf den Testdatensatz angewendet und die vorhergesagte Ausführungszeit wird für die Offloadingentscheidung verwendet. Folgende Fälle können nun auftreten [WF05, Kapitel 5.7]:

1. **True**: Die Offloadingentscheidung anhand der vorhergesagten Ausführungszeit stimmt mit der Referenzentscheidung überein Dies lässt sich noch weiter in *true negative* und *true positive* unterteilen: Dies bedeutet, dass beide Entscheidungen übereinstimmen und negativ oder positiv sind.
2. **False positives**: Das Ergebnis ist, dass geoffloaded werden sollte, obwohl es sich tatsächlich nicht lohnen würde.
3. **False negatives**: Das Ergebnis ist, dass nicht geoffloaded werden sollte, obwohl es sich tatsächlich lohnen würde.

Die Punkte 2 und 3 können zu einem Fall (**False**) zusammengefasst werden.

Daraus soll die Genauigkeit (engl. accuracy) bestimmt werden:

$$accuracy = \frac{|True|}{|True| + |False|} \tag{6.3}$$

Zusätzlich soll für jede falsche Entscheidung der *loss* und für jede richtige der *benefit* in Bezug auf die eingesparte Energie/Zeit berechnet werden.
Für die Berechnung des *loss* werden nur die *false positives* herangezogen, da der direkte Vergleich zwischen Offloading und kein Offloading gezogen werden soll. Der *loss* ist dabei die zusätzliche Zeit/Energie, die im Vergleich zu einer rein lokalen Ausführung anfällt. Der *benefit* ist die Einsparung im Vergleich zu einer rein lokalen Ausführung.

Einbezug des Energieverbrauchs für das Downloaden der Vorhersagemodelle

Zudem soll untersucht werden, wie stark das Herunterladen der Vorhersagemodelle die Effizienz des Systems in Bezug auf den Energieverbrauch beeinflusst. Hierzu wird die Übertragung des Linear Regression, des SVR und des KNN Regression Modelles betrachtet. Zum Einen wird der *benefit*, der durch das Offloading entsteht und zum Anderen der *loss*, der durch Übertragung der Modelle entsteht, betrachtet.

6.2.3. Berechnung eines Leistungsfaktors, Dauer zur Erstellung und Speicherbedarf

Leistungsfaktor

Es soll untersucht werden, inwiefern man die Ausführungszeit von einer CPU-Architektur zur anderen Umrechnen kann. Die Idee ist, die Ausführungszeiten einer Architektur als Referenz zu nehmen und auf eine andere Architektur anhand eines Faktors umzurechnen. Hierzu sollen für eine Testanwendung für die gleichen Testfälle Samples $u_1, u_2, ..., u_n$ für zwei verschiedene Architekturen u' und u'' aufgezeichnet werden, wobei u' die Referenzarchitektur sein soll. Für jedes Paar u'_i und u''_i soll der relative Unterschied $\frac{u''_i}{u'_i}$ berechnet werden. Anhand des Mittelwertes der Standardabweichung und der Varianz soll untersucht werden, ob es ausreichend ist, anhand der Ausführungszeit einen Leistungsfaktor zu berechnen.

Dauer zur Erstellung der Modelle

Es soll die Dauer zur Erstellung der Modelle untersucht werden. Das Ziel dabei ist, festzustellen, ob die Annahme des Entwurfs, dass der Server für das Erstellen der Modelle zuständig ist, richtig ist.

Speicherbedarf

Es soll untersucht werden, wie sich der Speicherbedarf der Modelle bei steigender Datensatzgröße verhält.

6.3. Testszenarien

6.3.1. Szenarios zum Bewerten der Machine Learning Algorithmen

Für jede aus 6.4 vorgestellte Anwendung sollen anhand der beschriebenen Testfälle Vorhersagemodelle erstellt werden. Die Anzahl der Samples wird schrittweise erhöht, bis alle Samples aus dem Datenbestand zur Modellerstellung verwendet werden. Es soll dargestellt werden, wie sich die Metriken aus 3.3.5 bei steigender Sampleanzahl verändern.
Anhand der Metriken für den jeweils gesamten Datensatz wird schließlich bewertet, wie gut die Modelle für Vorhersagen verwendet werden können. Für die Testfälle wurde jeweils eine Aufzeichnung in Nanosekunden durchgeführt. Die Vorhersage der Machine Learning Modelle ist ebenfalls in Nanosekunden.

6.3.2. Szenarien zum Bewerten der Offloadingentscheidung

Ausgehend von den Messergebnissen von [HQG+12] und Tabelle 6.1 wurden fünf Testkonfigurationen für die Datenraten erstellt.

Uplink	5 Mbps
Downlink	15 Mbps
p_m	426 mW
p_i	268.8 mW
p_C	3479,99 mW
p_r	2067,59 mW

Tabelle 6.2.: Testkonfiguration 1: LTE mit mittleren Datenraten.

Uplink	2,5 Mbps
Downlink	7,5 Mbps
p_m	426 mW
p_i	268.8 mW
p_C	2990,33 mW
p_r	1733,88 mW

Tabelle 6.3.: Testkonfiguration 2: 3G mit mittleren Datenraten.

Für die Gleichung 6.1 werden lediglich die Up- und Downstream Raten verwendet und für Gleichung 6.2 noch zusätzlich die Leistungswerte. Für jedes Vorhersagemodell, das in 6.3.1 für den jeweils gesamten Datensatz erstellt worden ist, soll nun die Leistung bezüglich der Offloadingentscheidung für Gleichung 6.1 und 6.2 getestet werden.

Uplink	0,5 Mbps
Downlink	1 Mbps
p_m	426 mW
p_i	268.8 mW
p_C	1252,37 mW
p_r	940,1 mW

Tabelle 6.4.: Testkonfiguration 3: 3G mit niedrigen Datenraten.

Uplink	1 Mbps
Downlink	6 Mbps
p_m	426 mW
p_i	268.8 mW
p_C	416,03 mW
p_r	954,92 mW

Tabelle 6.5.: Testkonfiguration 4: Wifi (802.11 g) mit niedrigen Datenraten.

Uplink	10 Mbps
Downlink	50 Mbps
p_m	426 mW
p_i	268.8 mW
p_C	2964,56 mW
p_r	6983,36 mW

Tabelle 6.6.: Testkonfiguration 5: Wifi (802.11 g) mit hohen Datenraten.

Für den Leistungsunterschied wird eine moderate Annahme von einem 10-mal[1] schnelleren Server getroffen, also $s_m = 1$ und $s_S = 10$. Für den Eingabezustand d_i wird für jede Anwendung bei jedem Ausführungsszenario zusätzlich dieser in Byte aufgezeichnet - im Falle der Gesichtserkennung und der Rauschreduzierung wäre das die Größe des Bildes und bei der Text-To-Speech Anwendung der umzuwandelnde String. Für den Ausgabezustand wird bei der Schachanwendung, der Rauschreduzierung und Bilderkennung die gleiche Größe wie beim Eingabezustand angenommen. Bei der Schachanwendung (siehe Abschnitt 6.4.1) wird ein konstanter Eingabe - und Ausgabezustand angenommen, welcher 64 Byte beträgt.

[1] http://cpuboss.com/cpus/Qualcomm-Snapdragon-800-vs-Intel-Core-i5-2500K

Einbezug des Energieverbrauchs für das Downloaden der Vorhersagemodelle

Hierfür wird die Energieeinsparung bei der Schachanwendung mit allen "difficulty"-Parametern betrachtet. Es wird dabei die Einsparung mit dem kompletten Testdatensatz(circa 6300) und dem Herunterladen von 25 Modellen untersucht. Dem Gegenüber wird die mittlere Einsparung von 1% des Testdatensatzes und dem Herunterladen der 25 Modelle gestellt.

6.3.3. Szenario:Berechnen eines Leistungsfaktors, Dauer zur Erstellung und Speicherbedarf

Leistungsfaktor

Für die Bewertung des Leistungsfaktors werden für die Schachanwendung jeweils 1000 Samples mit "difficulty"-Paramter 2 und 100 Samples mit "difficulty"-Parameter 3 auf zwei verschiedenen Architekturen aufgezeichnet. Die Referenzarchitektur kann aus Tabelle 6.9 und die Testarchitektur aus Tabelle 6.10 entnommen werden.

Dauer zur Erstellung

Anhand der Architektur aus 6.8 wird für die Schachanwendung und die Rauschreduzierung untersucht, wie viele Sekunden zur Modellerstellung benötigt werden.

Speicherbedarf

Es werden die Größen der Vorhersagemodelle aller Machine Learning Algorithmen für die Schachanwendung und die Gesichtserkennung untersucht. Beide Testanwendungen haben ähnlich viele Testfälle, unterscheiden sich aber in der Anzahl ihrer Features. Bei Schach sind es 65 und bei der Gesichtserkennung ein Feature.

6.4. Testanwendungen

6.4.1. Schachspiel

Beschreibung

Schach ist ein Brettspiel für zwei Spieler, das vermutlich zwischen dem 3. und 6. Jahrhundert entstanden ist. Es besteht aus einem 8x8 Felder großen Spielfeld und jede Seite besitzt zu Beginn des Spiels 8 Bauern, 2 Läufer, 2 Türme, 2 Springer eine Dame und einen König, welche

unterschiedliche Zugmöglichkeiten besitzen. Der Spieler kann gegnerischen Figuren aus dem Spielverlauf ausschließen, indem er eine eigene Figur auf die Position der gegnerischen Spielfigur setzt. Geschlagene Spielfiguren werden vom Spielfeld entfernt und können nicht mehr eingesetzt werden. Gewonnen hat der Spieler, welcher den gegnerischen König geschlagen hat.
Bereits Shannon hat sich algorithmisch mit Schach auseinandergesetzt und festgestellt, dass Schach eine Spiel-Komplexität von 10^{120} hat [Sha50]. Deswegen ist es effektiv nur mit Heuristiken möglich, computergesteuerte Spielzüge zu machen. Als Beispielanwendungen wurde eine Implementierung einer Heuristik namens Alpha-Beta Pruning verwendet [KM76]. Als Offloadingkandidat wurde die Methode *execute*, welche den nächsten Spielzug des Computergegners entscheidet, genommen. Listing 6.1 zeigt die execute-Methode und deren Parameter. Als Features wurden die Parameter *difficulty* und das 64-elementige Byte-Array *myBoard* verwendet. Dieses Byte-Array stellt das Spielfeld dar. Der Wert eines Elementes des Byte-Arrays bestimmt, ob und welche Spielfigur an dieser Stelle steht. Daraus wurden 64 Features abgeleitet. Jedes dieser Features stellt eine Kachel auf dem Spielfeld dar. Außerdem soll für die die Evaluation der Schachanwendung der Testdatensatz nach den Werten des Parameters "difficulty" aufgeteilt werden, da dieser einen großen Einfluss auf die Ausführungszeit ausübt. Der Parameter kann die Werte 1,2 und 3 annehmen. Es soll untersucht werden, inwieweit sich die Qualität der Modelle unterscheidet, wenn eine komplette Unterteilung stattfindet im Vergleich zu den Modellen, die ohne Aufteilung erstellt worden sind. Hierzu werden die MAE und RMSE Werte anteilig aufaddiert und durch 3 geteilt.

```
public byte[] execute(byte[][] myboard, byte[][][] undo, byte difficulty) {
      short steps = (short) ((difficulty + 4) / 2);
      short maxsteps = (short) (4 * difficulty + 2);

      // Exec
/*    System.out.println("Calculating next chess move (difficulty=" + difficulty +
    ", steps=" +
                steps + ", maxsteps=" + maxsteps + ") .."); */
      int[] ply = search(myboard, undo, steps, maxsteps, ALPHA, BETA);
      byte[] plymov = new byte[6];
      plymov[0] = (byte) ply[1];
      plymov[1] = (byte) ply[2];
      plymov[2] = (byte) ply[3];
      plymov[3] = (byte) ply[4];
      plymov[4] = (byte) ply[5];
      plymov[5] = (byte) ply[0];

      //
      //System.out.println("Set returns (s:" + invoke_counter_search + "|m:" +
          invoke_counter_move
      //          + "|e:" + invoke_counter_isEnPassant + "|c:" +
          invoke_counter_isCastleing + "|n:" +
      //          invoke_counter_isNormal + "|c:" + invoke_counter_copy + ")");
```

```
            return plymov;
    }
```

Listing 6.1: Search Methode, welche die Alpha-Beta Heuristik verwendet

Testfallerstellung

Für die Testfallerstellung wurde eine bestehende Schachdatenbank verwendet [vKem]. Die Datenbank verwendet das *Portable Game Notation* Format [Edw94]. Die Datenbank enthält gespielte Züge eines Spiels. Die Testfälle wurden folgendermaßen erstellt: Aus jedem Zug wurden individuelle Spielfelder abgeleitet, welche als Eingabe für die Methode *execute* dient. Für diese Spielkonfiguration wird ein nächster Zug mit Hilfe des Alpha-Beta Prunings berechnet. Jeder Zug wird anschließend wieder verworfen und ein neues Spielbrett wird aus der Datenbank abgeleitet.

6.4.2. Gesichtserkennung

Beschreibung

Für die Gesichtserkennung wurde eine Open-Source Bildbearbeitungsbibliothek namens JJIL [jji]verwendet. Mit Hilfe dieser werden sogenannte Haar-Features [WF06] in Bildern erkannt. Über Haar Features kann eine Gesichtserkennung innerhalb eines Bildes gemacht werden. Die in Listing 6.2 gezeigte Methode wurde als Offloadingkandidat für Aufzeichnungen gewählt.

```
public List<Rect> pushAndReturn(Image image) throws jjil.core.Error
{
   List<Rect> result = new ArrayList<Rect>();
   Gray8Image imGray;

         [...]
      // now run Haar detection on each scaled image
      int nxLastFound = -hcc.getWidth();
      int nyLastFound = -hcc.getHeight();
      while (!mgsi.isEmpty()) {
         Gray8OffsetImage imSub = (Gray8OffsetImage) mgsi.getFront();
         // if we've found a feature recently we skip forward until
         // we're outside the masked region. There's no point rerunning
         // the detector
         if (imSub.getXOffset() > nxLastFound + hcc.getWidth() &&
            imSub.getYOffset() > nyLastFound + hcc.getHeight()) {
            if (hcc.eval(imSub)) {
               // Found something.
               nxLastFound = imSub.getXOffset();
               nyLastFound = imSub.getYOffset();
               // assign Byte.MAX_VALUE to the feature area so we don't
               // search it again
```

```
                result.add(new Rect(nxLastFound*nScale, nyLastFound*nScale,
                      this.hcc.getWidth()*nScale,
                      this.hcc.getHeight()*nScale));

                Gray8Rect gr = new Gray8Rect(nxLastFound,
                      nyLastFound,
                      this.hcc.getWidth(),
                      this.hcc.getHeight(),
                      Byte.MAX_VALUE);
                gr.push(imMask);
                imMask = (Gray8Image) gr.getFront();
              }
           }
        }
        nScale = nScale * 256 / this.nScaleChange;
     }
     // Stretch imMask to original image size; this is the result
     Gray8RectStretch grs = new Gray8RectStretch(image.getWidth(), image.getHeight());
     grs.push(imMask);
     super.setOutput(grs.getFront());
     return result;
  }
```

Listing 6.2: Offloadingkandidat der Gesichtserkennung.

Als Feature zur Vorhersage wurde die Anzahl der Pixel des Bildes verwendet, welche durch die Multiplikation der Höhe und Breite des Bildes bestimmt wurde.

Testfallerstellung

Für die Testfallerstellung wurde als Basis eine Bilddatenbank mit Portraitfotos Names *Yalefaces* [Yal] verwendet. Die Datenbank besteht aus 165 Graustufenbildern. Von 15 Personen wurden jeweils 11 Portraitfotos in unterschiedlichen Positionen aufgenommen.
Um unterschiedliche Bildgrößen zu simulieren, wurden die aufgeteilten Datensätze zusätzlich mit unterschiedlichen Faktoren vergrößert. Ein Faktor von 1.5 würde beispielsweise das Basisbild 1.5 fach vergrößern.
Es wurden Faktoren von 1 bis 9 mit jeweils Unterschritten von 0.2, 0.4, 0.5, 0.6, und 0.8 gewählt.

6.4.3. Rauschreduzierung

Beschreibung

Für die Rauschreduzierung wurde die Bildbearbeitungsbibliothek Marvin [Mar] verwendet. Als Beispielanwendung wurde ein Algorithmus zur Rauschreduzierung eines Bildes verwendet.

Listing 6.3 zeigt die Methode *denoise*, welche als Offloadingkandidat gewählt wurde. Als Features wurden die Bildgröße in Pixel gewählt.

```
public double[][] denoise(double mat[][],int iter)
{
      img_org=new double[width][height];
      double img_res[][]=new double[width][height];
      double l_currentNum;
      double l_currentDen;
      int val=1;double lam=0;
      double dt=0.4;
      img_org=mat;

      //Perform iterations

      for (int it = 0;it<iter;it++)
      {
            //compute derivatives
            img_x=diff_x(mat);
            img_y=diff_y(mat);
            img_xx=diff_xx(mat);
            img_yy=diff_yy(mat);
            img_xy=diff_xy(mat);

            for(int i = 0;i<width;i++)
            {
                  for (int j=0;j<height;j++)
                  {
                        double a= img_xx[i][j]*(val+Math.pow(img_y[i][j],2));
                        double b=(2*img_x[i][j]*img_y[i][j]*img_xy[i][j]);
                        double c=(img_yy[i][j]*(val+Math.pow(img_x[i][j],2)));
                        l_currentNum= a-b+c;
                        l_currentDen=Math.pow((val+Math.pow(img_x[i][j],2)+Math.pow(img_y[i][j],
                        img_res[i][j]=(l_currentNum/l_currentDen)+
                            lam*(img_org[i][j]-mat[i][j]);
                        mat[i][j]=mat[i][j]+dt*img_res[i][j];//evolve image by dt.
                  }
            }
      }// end of iterations.
      return mat;
}
```

Listing 6.3: Offloadingkandidat der Rauschreduzierung.

Testfallerstellung

Als Testfälle wird eine Teilmenge von der bei der Gesichtserkennung erstellten Bilder verwendet. Die Testmenge wurde auf jeweils 125 Bilder von Faktor 1-4 und jeweils 16 Bilder von Faktor 1-8 reduziert. Faktor 9 wurde nicht mehr verwendet.

6.4.4. Text-To-Speech

Beschreibung

FreeTTS ist ein Sprachsyntheseframework, welches anhand einer Texteingabe eine Audiausgabe erzeugt [WLK02]. FreeTTS bietet mehrere verschiedene Stimmen für die Synthese. Für die Erstellung des Vorhersagemodelles wurde die Stimme *alan* verwendet. Als Offloadingkandidat wurde die Methode *speak* der Klasse *Voice* aus dem Java Package *com.sun.speech.freetts.Voice* gewählt. Die Methode hat als Eingabe einen String. Als Feature für die Vorhersage wurde die Anzahl der Wörter pro synthetisiertem Satz, sowie die Gesamtlänge des Strings gewählt.

Testfallerstellung

Für die Testfallerstellung wurden insgesamt elf Forschungspapiere aus verschiedenen Disziplinen gewählt. Die PDF-Dateien wurden in ein Plaintext-Format umgewandelt. Die Texte wurden satzweise aufgeteilt und als Eingabe für die Methode *speak* verwendet. In Anhang A ist eine detaillierte Auflistung der Testfälle beschrieben.

6.4.5. Zusammenfassung

Tabelle 6.7 fasst nochmals die Anzahl der Parameter und deren Beschreibung zusammen, die für die einzelnen Testanwendungen verwendet worden sind.

Testanwendung	Anzahl Features	Beschreibung	Datensatzgröße
Schach	65	8x8 Schachbrett + difficulty	circa 19.800
Schach mit difficulty=1	64	8x8 Schachbrett	circa 7.000
Schach mit difficulty=2	64	8x8 Schachbrett	circa 10.500
Schach mit difficulty=3	64	8x8 Schachbrett	circa 1.800
Gesichtserkennung	1	Pixel (Höhe * Breite des Bildes)	circa 21.000
Rauschreduzierung	1	Pixel (Hohe * Breite des Bildes)	circa 2.400
Text-To-Speech	2	Wordcount und Stringlänge	circa 840

Tabelle 6.7.: Zusammenfassung: Anzahl der Parameter und Parameterbeschreibungen sowie die Datensatzgrößen.

6.5. Testhardware

CPU	Intel Core I-5 2500K @ 3,3 GHz, Sandy Bridge
CPU Kerne	4
Arbeitsspeicher	8 GB
Grafikkarte	NVIDIA GeForce GTX 560 Ti
Festplatte	WDC WD6400AAKS-65A7B0, 7200 RPM

Tabelle 6.8.: Hardware für die Testfallerstellung.

CPU	Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz, Sandy Bridge
CPU Kerne	4
Arbeitsspeicher	8 GB

Tabelle 6.9.: Hardware für die Testfallerstellung der Schachanwendung.

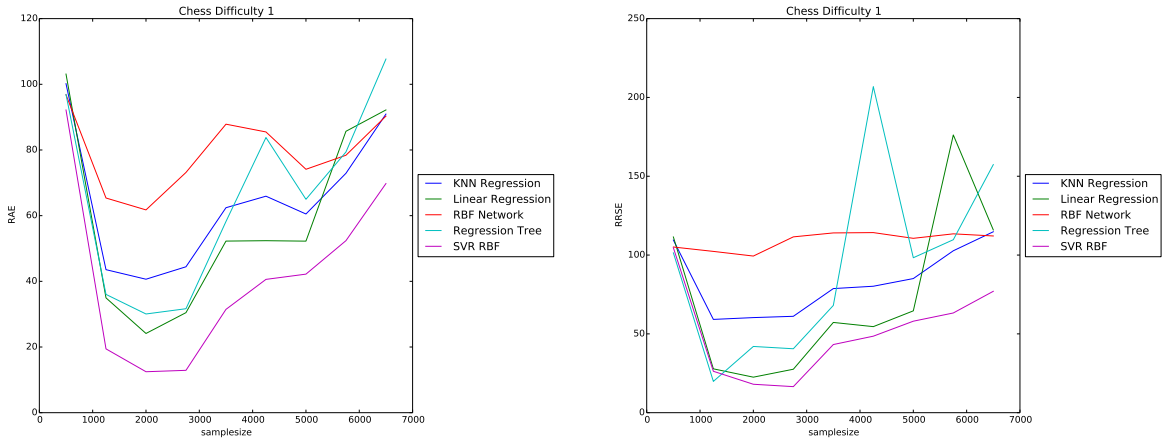
CPU	Intel(R) Xeon(R) CPU E5620 @ 2.40GHz
CPU Kerne	6
Arbeitsspeicher	8 GB

Tabelle 6.10.: Hardware für die Testfallerstellung der Schachanwendung zur Berechnung eines Leistungsfaktors.

6.6. Testergebnisse: Machine Learning

Im Folgenden werden die Metriken *Relative Absolute Error*(RAE) und *Root Relative Squared Error*(RRSE) für die jeweiligen Testanwendungen als Diagramm aufgeführt. Diese beiden Metriken wurden gewählt, da sie sich im Gegensatz zu dem *Mean Absolute Error*(MAE) und *Root Mean Squared Error*(RMSE) lesbarer darstellen lassen. Die vollständigen Daten mit allen Metriken sind in Anhang B aufgelistet. Außerdem wird jeweils der Fall unterschieden, wenn ausgehend von der Datensatzgröße, mit der das Modell erstellt worden ist, der Testdatensatz abgeleitet wird. Dies wird als *dynamic* in den Grafiken vermerkt. Der Fall, dass das erstellte Modell immer mit dem kompletten Testdatensatz bewertet wird, wird mit *static* in den Grafiken vermerkt.

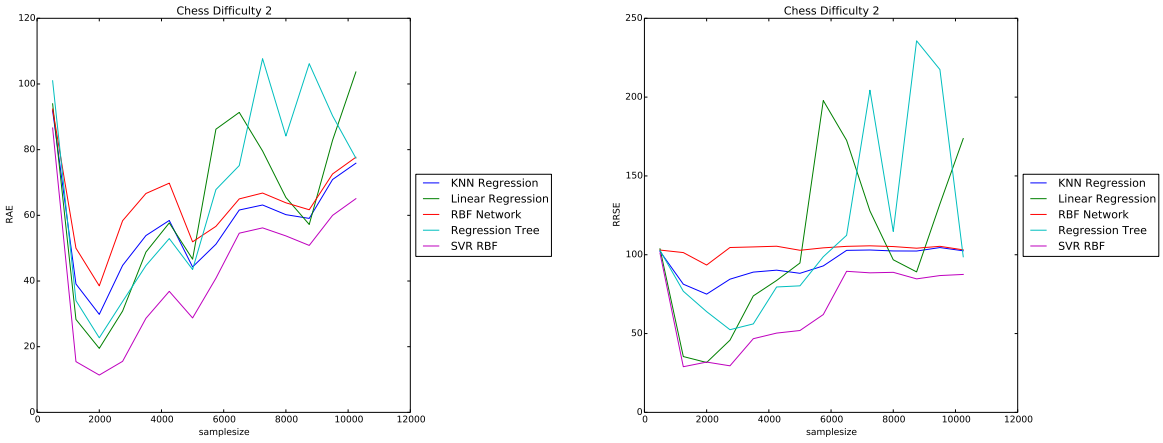
6.6.1. Schachspiel



(a) Relative Absolute Error (*dynamic*)

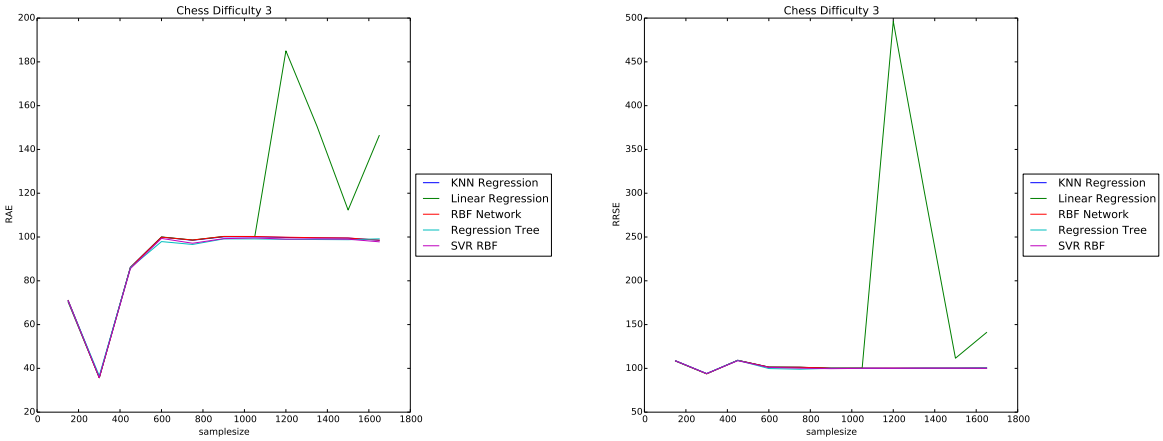
(b) Root Relative Squared Error (*dynamic*)

Abbildung 6.1.: Schach mit difficulty 1: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.



(a) Relative Absolute Error (*dynamic*) (b) Root Relative Squared Error (*dynamic*)

Abbildung 6.2.: Schach mit difficulty 2: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

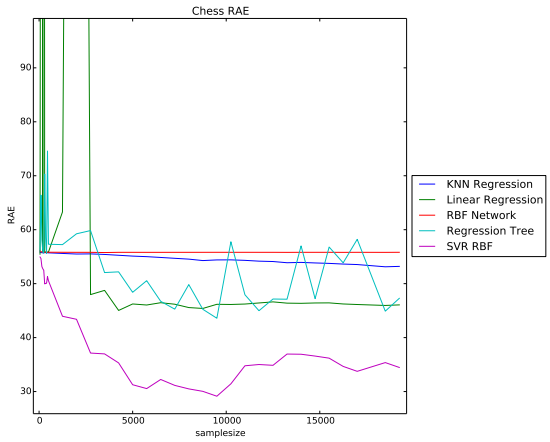


(a) Relative Absolute Error (*dynamic*) (b) Root Relative Squared Error(*dynamic*)

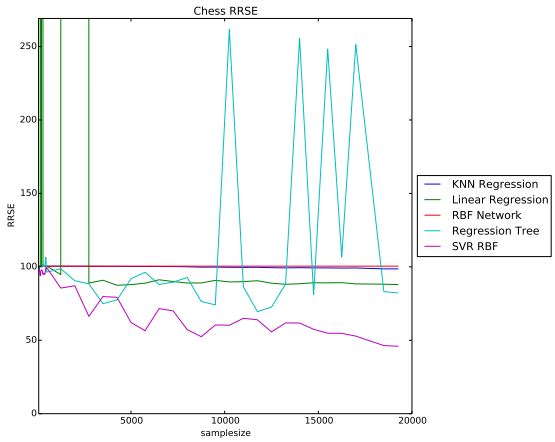
Abbildung 6.3.: Schach mit difficulty 3: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

Das Modell, das mit dem “difficulty”-Parameter 1 erstellt wurde, hat einen RRSE von 69%, bei 2 einen RRSE von 65% und bei 3 einen Wert von 97%. Das beste Modell für den kompletten Datensatz ist Support Vector Regression mit RBF Kernel mit einem RRSE von 33,9%.

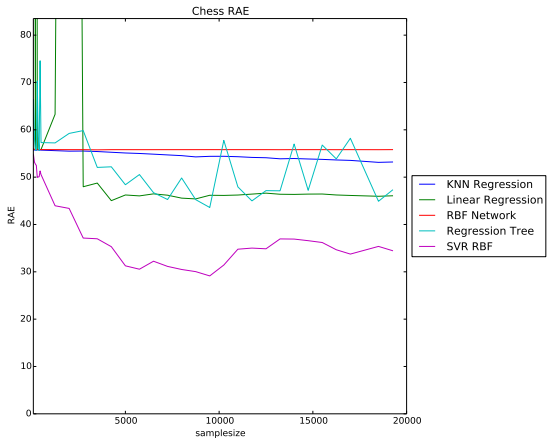
Für jeden Parameter ist das SVR Modell das, welches den besten RAE und RRSE hat. Betrachtet man den RAE und RMSE (siehe Anhang B) der einzelnen Modelle, lässt sich feststellen, dass



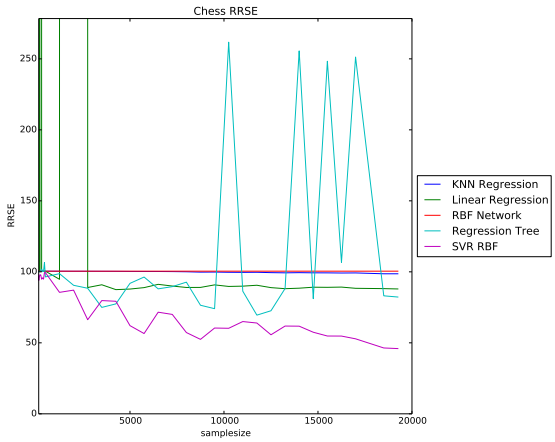
(a) Relative Absolute Error (*dynamic*)



(b) Root Relative Squared Error (*dynamic*)



(c) Relative Absolute Error (*static*)



(d) Root Relative Squared Error (*static*)

Abbildung 6.4.: Schach (alle difficulty parameter):Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

geringere Fehlerraten durch die Aufteilung nach dem Parameter “difficulty” auftreten. Summiert man die RMSE und RAE Fehlerraten anteilig nach der jeweiligen Datensatzgröße auf, erhält man ebenfalls eine geringere Fehlerrate, als mit dem SVR Modell, welches mit allen “difficulty”-Parametern eingelernt wurde.

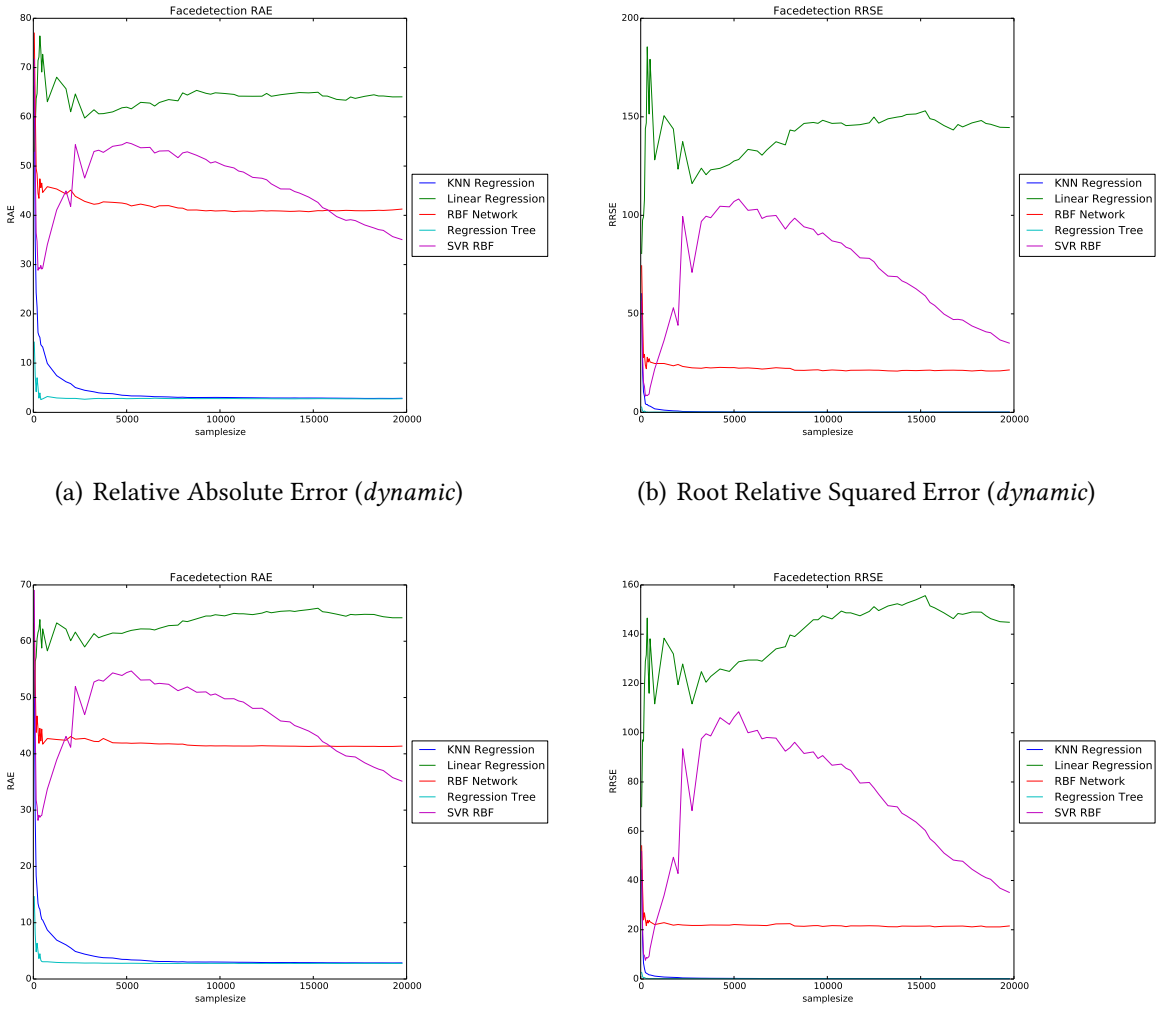
Tabelle 6.11 fasst eine Übersicht der jeweils besten Modelle für den jeweiligen Datensatz zusammen. Außerdem ist der RMSE und MAE aufgelistet, der entsteht, wenn die Modelle für die einzelnen “difficulty”-Parameter zusammengeführt werden.

Difficulty	RRSE	RAE	RMSE	RME	Model
1	69,00%	77%	425348341,7	175924329,9	SVR
2	65,00%	87%	3,19247E+12	55190705991	SVR
3	97,00%	100%	4,72771E+12	1,00476E+12	SVR
Kombiniert	-	-	7,37808E+11	40465705012	SVR
Alle	**34%**	**45%**	7.930258026759402E11	4.845774349391204E10	SVR

Tabelle 6.11.: Schachanwendung: Beste Modelle für den jeweils kompletten Datensatz mit verschiedene Datensatzaufteilungen nach Parameter “difficulty”.

6.6.2. Gesichtserkennung

Bei der Gesichtserkennung ist die Vorhersagequalität im Vergleich zur Schachanwendung deutlich besser. Am Besten schneidet der Regression Tree mit einem RRSE von 0.12% ab.



(a) Relative Absolute Error (*dynamic*)

(b) Root Relative Squared Error (*dynamic*)

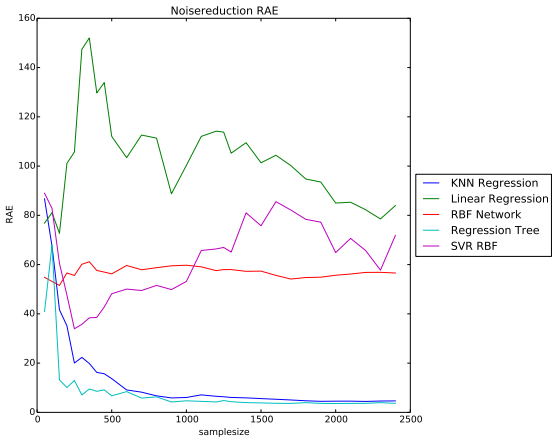
(c) Relative Absolute Error (*static*)

(d) Root Relative Squared Error(*static*)

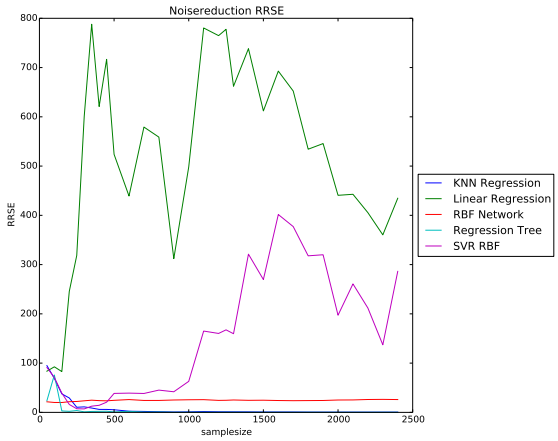
Abbildung 6.5.: Gesichtserkennung: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

6.6.3. Rauschreduzierung

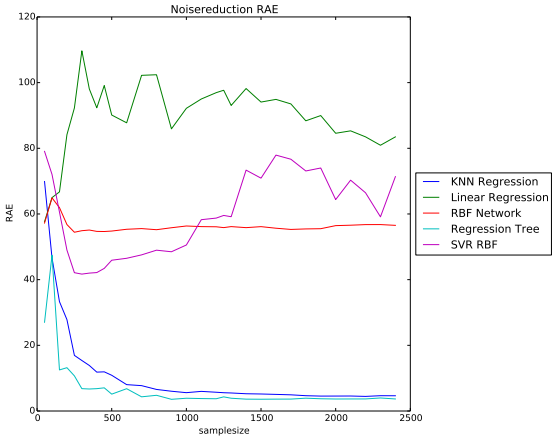
Der beste Algorithmus ist auch hier, wie bei der Gesichtserkennung, der Regression Tree mit einem RRSE von 0.16%.



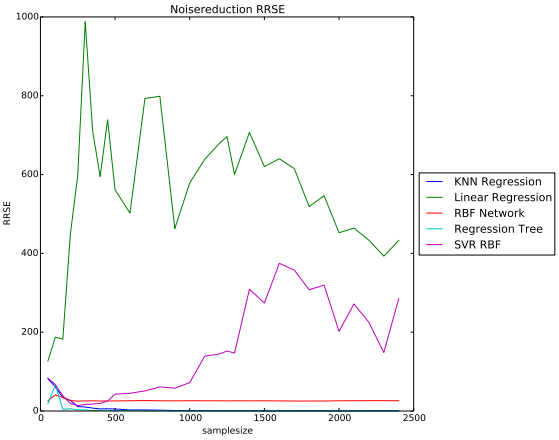
(a) Relative Absolute Error (*dynamic*)



(b) Root Relative Squared Errorr (*dynamic*)



(c) Relative Absolute Error (*static*)

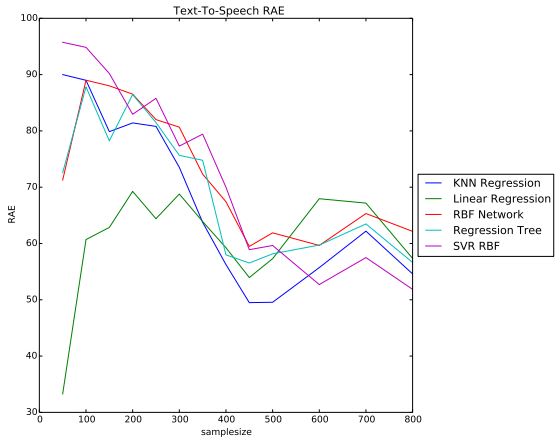


(d) Root Relative Squared Error(*static*)

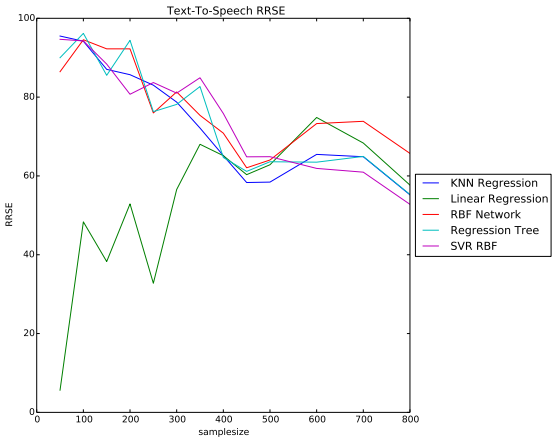
Abbildung 6.6.: Rauschreduzierung: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

6.6.4. Text-To-Speech

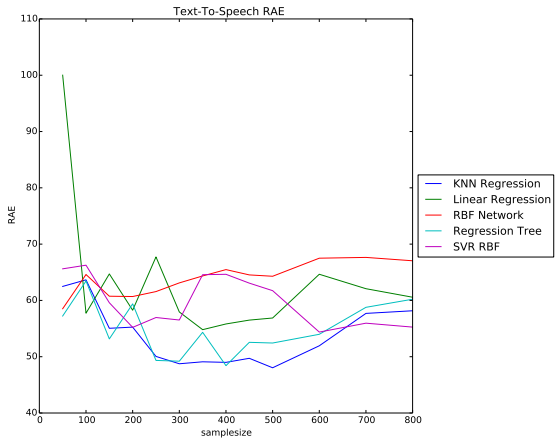
Bei der Text-To-Speech Anwendung ist der beste Algorithmus Support Vector Regression mit RBF Kernel mit einem RRSE von 51%.



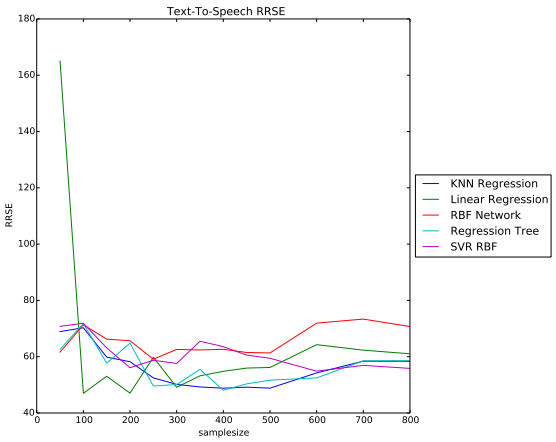
(a) Relative Absolute Error (*dynamic*)



(b) Root Relative Squared Error(*dynamic*)



(c) Relative Absolute Error (*static*)



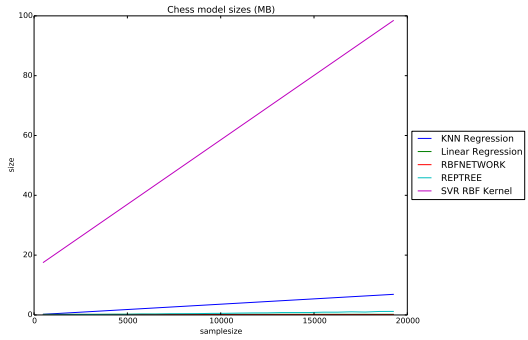
(d) Root Relative Squared Error (*static*)

Abbildung 6.7.: Text-To-Speech: Vergleich der Metriken der Machine Learning Algorithmen in Abhängigkeit der Anzahl der Samples.

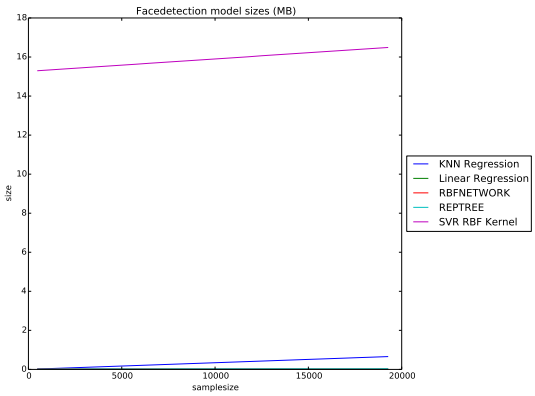
6.7. Testergebnisse: Leistungsfaktor, Speicherbedarf und Zeit zum Erstellen

6.7.1. Speicherbedarf der Vorhersagemodelle

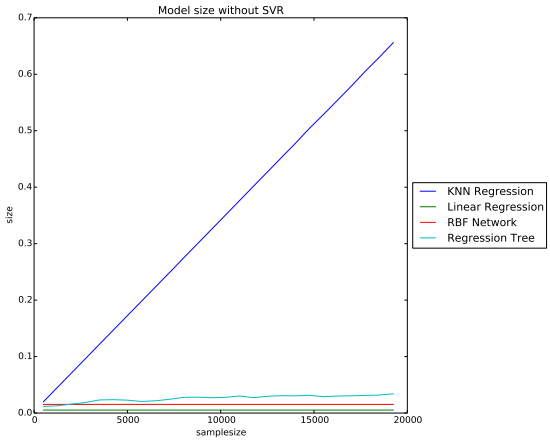
Für die Schachanwendung und der Gesichtserkennung soll die Größe der Vorhersagemodelle in Abhängigkeit der Anzahl der Samples, mit denen sie erstellt wurden, dargestellt werden. Die Größen wurden dabei in Megabyte angegeben. Die Linear Regression und RBF Network



(a) Schach



(b) Gesichtserkennung



(c) Gesichtserkennung ohne SVR

Abbildung 6.8.: Vergleich der Größen der Vorhersagemodelle in Megabyte.

Modelle haben konstante Größen von circa 16 und 30 Kilobyte.
Die Regression Tree Modelle steigen bei steigender Trainingsdatensatzgröße leicht an: Anfangs hat das Modell 10 Kilobyte und bei circa 19000 Samples 30 Kilobyte.
Das KNN Regression Model steigt linear zur Größe des Datensatzes an. Das liegt an der Konzeption des Algorithmus: Es werden ausgehend vom gesamten Datensatz bei jeder Vorhersage die k Nachbarn ausgewählt, die dem zu vorhersagenden Sample am nächsten sind. Bei der Schachanwendung hat das KNN Modell anfangs eine Größe von 20 KB und beim kompletten Datensatz ca 670 KB.
Die Vorhersagemodelle, die durch die Support Vector Regression erstellt worden sind, liegen

im Bereich von circa 18-100 Megabyte. Anzumerken ist jedoch, dass durch Kompression[2] der Speicherbedarf stark verringert werden kann. Tabelle 6.12 zeigt die Kompressionsrate: diese liegt im Mittel bei circa 2800%.

Tabelle 6.12.: Vergleich des Speicherbedarfs zwischen komprimierten (LZMA) und unkomprimierten SVR Vorhersagemodellen der Schachanwendung.

Compressed (bytes)	Uncompressed (bytes)	Compression
148587	18434729	12407%
739836	21821737	2950%
1602926	25215999	1573%
1994165	28602491	1434%
2186846	31997777	1463%
2302264	35384269	1537%
2332474	38777507	1663%
2360979	42163999	1786%
2410395	45561333	1890%
2369276	48947825	2066%
2480742	52341063	2110%
2520868	55727555	2211%
2475476	59120793	2388%
2547759	62507289	2453%
2603474	65900527	2531%
2610114	69287019	2655%
2671957	72688449	2720%
2746975	76074941	2769%
2677530	79468179	2968%
2788755	82854671	2971%
2756458	86247909	3129%
2812129	89634401	3187%
2832335	93027639	3284%
2859306	99807369	3491%
3004576	103193861	3435%
	Mean	2843%

Ebenfalls scheint der Speicherbedarf des Support Vector Regression Modelles stark von der Implementierung abzuhängen. Beispielsweise ist das Vorhersagemodell mit der Implementierung

[2]Mit 7zip (http://www.7-zip.de/) und LZMA-Kompression

der LibSVM-Bibliothek [CL] für den Schachdatensatz mit circa 19000 Samples 23 Megabyte groß. Mit Kompression sind es noch lediglich 450 Kilobyte.

6.7.2. Leistungsfaktor

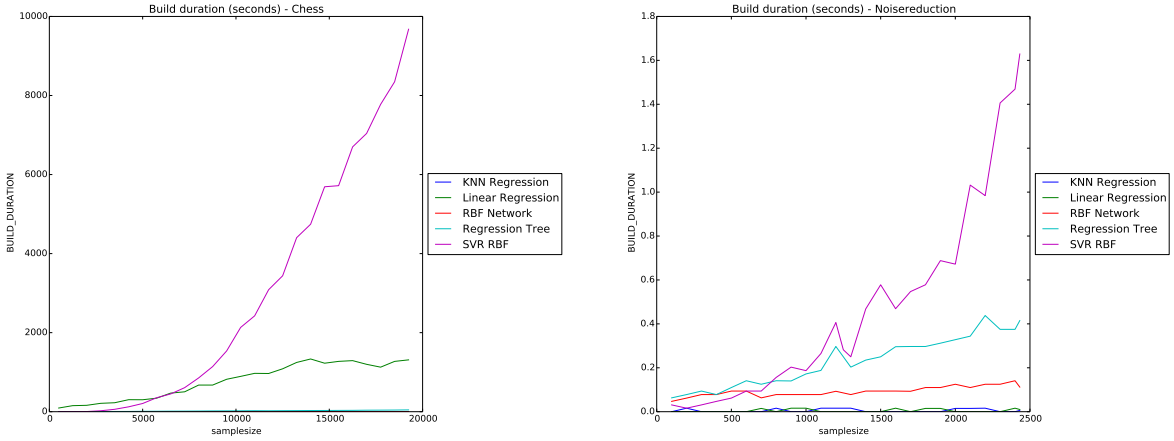
Es soll untersucht werden, inwiefern die Verwendung einer Referenzarchitektur zum Umrechnen der Ausführungszeiten auf andere Rechnerarchitekturen verwendet werden kann. Die Hardwarekonfiguration aus Tabelle 6.9 wurde als Referenzarchitektur verwendet. Die Konfiguration aus Tabelle 6.10 wurde als Vergleichsarchitektur benutzt. Hierzu wurden bei der Schachanwendung jeweils 1000 zufällige Schachkonfigurationen der Schwierigkeitsstufe 2 und 100 Schachkonfigurationen der Schwierigkeitsstufe 3 verwendet. Für beide Architekturen wurden dieselben Konfigurationen verwendet. Tabelle 6.13 fasst die Ergebnisse zusammen. In Anhang J sind die vollständigen Daten aufgelistet.

Testdatensatz	difficulty = 3	difficulty =2
Mittelwert	0,53	0,55
Standardabweichung	0,03	0,057

Tabelle 6.13.: Mittlere relative Differenz der Ausführungszeiten und die Standardabweichung der Differenz auf zwei verschiedenen Architekturen für zwei Testdatensätze.

6.7.3. Dauer zur Erstellung der Modelle

Die Dauer zum Erstellen der Modelle ist stark vom jeweiligen Algorithmus abhängig. Anzumerken ist auch, dass die Optimierung der Parameter mit der Gittersuche die Dauer stark beeinflusst, da hierfür eine 10-Fold-Cross-Validation für jede zu testende Parameterkonfiguration verwendet wird. Das SVR Modell dauert zur Erstellung beim kompletten Datensatz von circa 19000 Samples über 2.5 Stunden bei der Schachanwendung. Bei der Rauschreduzierung mit 2500 Samples werden knapp 2 Sekunden benötigt - die Schachanwendung benötigt 23 Sekunden. Die Anzahl der Features beeinflusst die Dauer also stark.



(a) Schachanwendung

(b) Rauschreduzierung

Abbildung 6.9.: Dauer zum Erstellen der Modelle der Schachanwendung (alle *difficulty*-Werte) und der Rauschreduzierung.

6.8. Testergebnisse: Offloadingentscheidung

Ausgehend von den Ergebnissen der Metriken sollen für die Offloadingentscheidung jeweils die Modelle genommen werden, die aus den kompletten Datensätzen erstellt worden sind. Folgend werden die Ergebnisse aus den Tests tabellarisch aus Übersichtsgründen zusammengefasst. Für jeweils das Leistungs- und Energieszenario wird jeweils aus allen Testkonfigurationen pro Modell alle Werte aufsummiert. Die Tabellen wurden jeweils absteigend nach der Anzahl der *false positive* Entscheidungen sortiert. In Anhang C - I sind die Rohtabellen aufgelistet.
Folgend soll auch eine Korrelation zwischen den Ergebnissen der Metriken und der Anzahl der *false positive* Entscheidungen untersucht werden.

6.8.1. Schachspiel

Tabellen 6.14 und 6.15 fassen jeweils die Ergebnisse für das Energieszenario und das Leistungsszenario. Es kann eine Korrelation zwischen den Metriken und der *accuracy* festgestellt werden: Der Support Vector Regression Algorithmus hat sowohl den besten Root Relative Squared Error und auch die höchste *accuracy*.

Tabelle 6.14.: Energieszenario für die Schachanwendung.

model	True positives	True positives (True)	True negatives	False	False negatives	False True positives	Accuracy	loss (mWs)	benefit (mWs)	overall benefit (mWs)
SVR RBF	31381	24282	7099	1494	695	799	**95,5%**	200,334993	100969591	100969391
REPTree	30796	24142	6654	2079	835	1244	**93,7%**	365,344619	100966311	100965946
Linear Regression	30530	24024	6506	2345	953	1392	**92,9%**	460,462395	100969216	100968756
KNN	29413	23461	5952	3462	1516	1946	**89,5%**	668,186295	100965053	100964385
RBF Network	24977	24977	0	7898	0	7898	**76%**	4562,51737	100969980	100965418

Tabelle 6.15.: Leistungsszenario für die Schachanwendung.

model	True positives	True positives	True negatives	False	False negatives	False positives	Accuracy	loss (sec)	benefit (sec)	overall benefit (sec)
SVR RBF	31914	28614	3300	961	411	550	**97,1%**	0,09876089	227708,413	227708,315
REPTree	31462	28507	2955	1413	518	895	**95,7%**	0,19812827	227707,874	227707,676
Linear Regression	31154	28106	3048	1721	919	802	**94,8%**	0,22380413	227708,066	227707,842
KNN	30689	28244	2445	2186	781	1405	**93,4%**	0,38153105	227707,724	227707,342
RBF Network	29025	29025	0	3850	0	3850	**88,3%**	1,9282323	227708,63	227706,702

Tabelle 6.16.: Leistungsszenario für die Schachanwendung mit aufgeteiltem “difficulty”-Parameter.

model	True positives	True positives	True negatives	False	False negatives	False positives	Accuracy	loss (sec)	benefit (sec)	overall benefit (sec)
SVR RBF	31253	30225	1028	1012	250	762	**96,8%**	0,28387968	352790,814	352790,53
Linear Regression	30887	29798	1089	1378	677	701	**96%**	0,26169675	352788,892	352788,63
RBF Network	30138	29183	955	2127	1292	835	**93%**	0,32914147	352767,676	352767,347
REPTree	30422	29489	933	1843	986	857	**94%**	0,35653304	351421,133	351420,776
KNN	28672	27081	1591	3593	3394	199	**89%**	0,05451868	348374,96	348374,906

6.8.2. Gesichtserkennung

Auf eine tabellarische Darstellung bei der Gesichtserkennung kann verzichtet werden. Es wurde für jede Konfiguration und Modell bei beiden Szenarien jeweils die korrekten Entscheidungen getroffen.

Tabelle 6.17.: Energieszenario für die Schachanwendung mit aufgeteiltem “difficulty”-Parameter.

model	True positives	True positives (True)	True negatives	False	False negatives	False True positives	Accuracy	loss (mWs)	benefit (mWs)	overall benefit (mWs)
SVR RBF	30215	27221	2994	2050	565	1485	**94%**	495,550796	156437831	156437336
Linear Regression	29440	26400	3040	2825	1386	1439	**91%**	530,204423	156435595	156435065
RBF Network	28111	26211	1900	4154	1575	2579	**87%**	1165,67354	156421732	156420566
REPTree	28314	25419	2895	3951	2367	1584	**88%**	542,124358	155378318	155377776
KNN	24743	20619	4124	7522	7167	355	**77%**	119,17375	144050898	144050779

6.8.3. Rauschreduzierung

Bei dem Energieszenario kann auf die tabellarische Darstellung verzichtet werden, da alle Entscheidungen korrekt getroffen worden sind . Beim Leistungsszenario sind der KNN Algorithmus und der Regression Tree die Algorithmen mit den besten Ergebnissen. Diese Ergebnisse stimmen auch mit den Ergebnissen der Metriken überein.

Tabelle 6.18.: Leistungsszenario der Rauschunterdrückung.

model	True positives	True positives	True negatives	False	False negatives	False positives	Accuracy	loss (sec)	benefit (sec)	overall benefit (sec)
REPTree	4015	0	4015	0	0	0	**100%**	0	0	0
KNN	4015	0	4015	0	0	0	**100%**	0	0	0
Linear Regression	3815	0	3815	200	0	200	**95%**	1060,88284	0	-1060,88284
SVR RBF	3635	0	3635	380	0	380	**90,5%**	1321,93471	0	-1321,93471
RBF Network	3625	0	3625	390	0	390	**90,3%**	448,097331	0	-448,097331

6.8.4. Text-To-Speech

Bei der Text-To-Speech sind keine eindeutigen Ergebnisse zu beobachten. Sowohl im Energieszenario als auch im Leistungsszenario ist keine Korrelation der Metriken mit der Offloadingentscheidung zu beobachten.

Tabelle 6.19.: Text-To-Speech: Energieszenario.

model	True positives	True positives (True)	True negatives	False	False negatives	False True positives	Accuracy	loss (mWs)	benefit (mWs)	overall benefit (mWs)
REPTree	1343	808	535	47	20	27	**96,6%**	0,02707946	9,42892528	9,40184582
KNN	1341	810	531	49	18	31	**96,5%**	0,03302567	9,36872538	9,33569972
Linear Regression	1336	797	539	54	31	23	**96,1%**	0,05288966	9,27112241	9,21823275
SVR RBF	1331	808	523	59	20	39	**95,8%**	0,09238217	9,3513706	9,25898843
RBF Network	1323	796	527	67	32	35	**95,2%**	0,11631731	9,26777357	9,15145626

Tabelle 6.20.: Text-To-Speech: Leistungsszenario.

model	True positives	True positives	True negatives	False	False negatives	False positives	Accuracy	loss (sec)	benefit (sec)	overall benefit (sec)
REPTree	1296	32	1264	94	80	14	**93,2%**	7,25225252	29,447163	22,1949105
KNN	1293	30	1263	97	82	15	**93%**	10,5546159	23,1234152	12,5687994
SVR RBF	1283	66	1217	107	46	61	**92,3%**	84,7789885	102,607647	17,828658
Linear Regression	1283	39	1244	107	73	34	**92,3%**	37,1180345	42,8659338	5,7478993
RBF Network	1275	69	1206	115	43	72	**91,7%**	112,328625	107,793595	-4,53503055

6.8.5. Energieoptimales Offloading mit Einbezug des Energieverbrauchs für das Downloaden der Vorhersagemodelle

Tabellen 6.22 und 6.23 zeigen die Differenz und den übriggebliebenen relativen Anteil der Einsparung. Zu beobachten ist, dass bei einem realistischen Übertragungsszenario von 1% des Testdatensatzes (entspricht also circa 63 Ausführungen) die Größe des Vorhersagemodelles stark ins Gewicht fällt. Bei der KNN Regression ist von der Einsparung nach Übertragung nur noch 21% der Gesamteinsparung übrig. Bei der Übertragung des SVR Modelles wird deutlich mehr Energie verbraucht, als eingespart.

Tabelle 6.21.: Energieverbrauch von verschiedenen Vorhersagemodellen, die aus unterschiedlich großen Datenbeständen erstellt worden sind. Als Grundlage wurden die Modelle der Schachanwendung mit allen Difficulty-Parameterwerten verwendet.

Samplesize	LR	KNN	SVR	SVR(compressed)
500	388,0734714	397,0159067	32515,06339	262,076851
1250	391,1354177	890,1427094	38489,0476	1304,918257
2000	394,6912263	1384,320733	44475,82636	2827,231174
2750	395,5166819	1877,530434	50448,90045	3517,296153
3500	396,3386099	2371,80723	56437,48534	3857,145734
4250	395,8376924	2864,833497	62410,55942	4060,719304
5000	396,3244995	3359,209065	68395,53206	4114,003519
5750	396,636691	3852,193001	74368,60615	4164,280465
6500	396,2680581	4346,371025	80360,80329	4251,440107
7250	396,7195894	4839,623057	86333,87737	4178,914664
8000	396,3615392	5333,688198	92318,85001	4375,517719
8750	396,5784858	5827,067223	98291,9241	4446,291715
9500	396,6049427	6321,033591	104276,8967	4366,229581
10250	396,7037152	6814,484932	110249,9779	4493,721898
11000	396,5996514	7308,394859	116234,9505	4591,991678
11750	397,1517189	7801,59045	122208,0246	4603,70327
12500	397,1517189	8296,05068	128207,4462	4712,781579
13250	396,9665205	8789,147499	134180,5203	4845,097873
14000	396,9912136	9283,339633	140165,493	4722,611202
14750	397,1922861	9776,267127	146138,5671	4918,789184
15500	397,1199706	10271,02367	152123,5397	4861,823931
16250	397,176412	10763,64074	158096,6138	4960,016104
17000	397,3810121	11257,79054	164081,5864	4995,655324
18500	397,578557	12245,12183	176039,6331	5043,226611
19250	397,4303983	12738,30331	182012,7072	5299,452957
	Summe	**Summe**	**Summe**	**Summe**
	9902,53008	159009,9909	2618862,432	103774,9369

Tabelle 6.22.: Tatsächliche Einsparung nach Übertragung der Modelle mit zwischenzeitlicher Ausführung von 6300 Offloadingkandidaten.

	LR	KNN	SVR	SVR(compressed)
Benefit(best model)	20192991	20192697	20192695	20192695
Differenz	20183088,5	20033687	17573832,6	20088920,06
Relativ	100%	99%	87%	99%

Tabelle 6.23.: Tatsächliche Einsparung nach Übertragung der Modelle mit zwischenzeitlicher Ausführung von 63 Offloadingkandidaten.

	LR	KNN	SVR	SVR(compressed)
Benefit(best model)	201929,91	201926,97	201926,95	201926,95
Differenz	192027,38	42916,9791	-2416935,48	98152,01315
Relativ	95%	21%	-1197%	49%

6.9. Schlussfolgerung

Folgende Fragestellungen sollen beantwortet werden:

- Eignen sich Machine Learning Algorithmen zur Vorhersage von Ausführungszeiten?
- Lohnt sich eine Aufteilung des Parameterraumes zur separaten Modellerstellung?
- Eignet sich ein dynamisch gewählter Testdatensatz zum Vergleich von Modellen, die aus unterschiedlichen Datensatzgrößen erstellt worden sind?
- Eignen sich die Vorhersagen für Offloadingentscheidung?
- Korrelieren die Metriken mit der Anzahl der richtigen Entscheidungen?
- Kann man Ausführungszeiten von unterschiedlichen Architekturen zusammenführen?
- Wie verhält sich der Speicherbedarf der Vorhersagemodelle bei steigender Datensatzgröße?
- Wie sehr beeinträchtigt die Übertragung der Vorhersagemodelle die Effizienz des Systems?

Eignen sich Machine Learning Algorithmen allgemein zur Vorhersage?

Die gewählten Algorithmen eignen sich zur Vorhersage.
Je nach Komplexität der Parameter werden unterschiedlich viele Samples benötigt, um Vorhersagemodelle zu erhalten, die sich in ihrer Qualität nur noch geringfügig verbessern. Die Leistung eines Modelles ist von der jeweiligen Anwendung abhängig. Dabei gibt es keinen klaren Gewinner: Je nach Anwendung kann sich ein anderer Algorithmus mehr oder weniger eignen.

Bei der Gesichtserkennung und der Rauschunterdrückung erbrachten jeweils die Regression Tree Modelle die besten Ergebnisse.
Bei der Gesichtserkennung ergab sich dabei bereits nach 500 Samples einen RAE von 2,74%. Bei 20.000 Samples liegt der RAE bei 2,77%. Bei der Rauschunterdrückung verhält es sich ähnlich: Mit den Regression Trees hat man bereits nach 500 Samples einen RAE von 6,72% und bei 2400

Samples 3,68%.
Bei der Text-To-Speech Anwendung stellt sich das SVR Modell als Bestes heraus. Bei 100 Samples liegt der RAE bei 82,9% und bei 800 bereits bei 51,8%. Auch bei der Schachanwendung ist das SVR Modell das Beste. Hier liegt der RAE bei 100 Samples bei 54% und bei circa 19.000 Samples bei 34%.

Tabelle 6.24 fasst nochmals pro Testanwendung die besten Modelle, welche anhand des kompletten Datensatzes erstellt wurden, dar.

Tabelle 6.24.: Zusammenfassung: Beste Algorithmen pro Testanwendung.

Testanwendung	Model	RRSE	RAE
Schach(alle Parameter)	SVR	34%	45,00%
Text-To-Speech	SVR	51%	52,00%
Noisereduction	Regression Tree	0,16%	3,60%
Facedetection	Regression Tree	0,12%	2,70%

Lohnt sich eine Aufteilung des Parameterraumes zur separaten Modellerstellung?

Wie aus Tabelle 6.11 hervorgeht, kann durch eine Aufteilung des Parameters “difficulty” bei der Schachanwendung eine leicht verbesserte Vorhersage erreicht werden. Dies spiegelt sich bei der Offloadingentscheidung allerdings nicht in der *accuracy* wieder.

Eignet sich ein dynamisch gewählter Testdatensatz zum Vergleich von Modellen, die aus unterschiedlichen Datensatzgrößen erstellt worden sind?

Ja, auch dynamisch gewählte Testdatensätze sind ausreichend zum Vergleich zweier Modelle, die mit unterschiedlich großen Datensätzen erstellt worden sind. Dabei korrelieren die Verläufe der dynamischen und statischen Metriken. Einzig bei der Rauschreduzierung ist eine Abweichung bei der Verwendung von 50 Samples zu beobachten: Dort ist die Linear Regression deutlich besser, als mit dem statischen Testdatensatz.

Eignen sich die Vorhersagen für Offloadingentscheidung?

Grundsätzlich eignen sich die Vorhersagen der Ausführungszeit, um eine Offloadingentscheidung zu treffen. Abschnitt 6.8 hat anhand der Modelle, die aus den jeweils gesamten Datensätzen der Testanwendungen erstellt worden sind, für unterschiedliche Testszenarien eine Offloadingentscheidung getroffen.
Bei der Anwendung zur Gesichtserkennung hat jedes Modell immer die richtige Entscheidung

getroffen. Bei der Anwendung zur Rauschreduzierung waren beim Energie- und Leistungsszenario der KNN und der Regression Tree Algorithmus ebenfalls immer korrekt. Bei der Schachanwendung waren für das SVR Modell und das Energieszenario noch 95% der Entscheidungen richtig und 2,4% *false positive* Entscheidungen. Beim Leistungsszenario waren 96% richtige Entscheidungen und 1,6% *false positive* Entscheidungen. Bei einer Aufteilung des Datensatzes nach dem "difficulty"-Parameter ist der Anteil der richtigen Entscheidungen beim Leistungsszenario 96,8% und der Anteil der *false positive* Entscheidungen bei 2.3%. Beim Energieszenario beträgt der Anteil der richtigen Entscheidungen 94% und der Anteil der *false positive* Entscheidungen 4,6%.
Bei der Text-To-Speech Anwendung waren beim Leistungsszenario für das Regression Tree Modell 93% waren richtige Entscheidungen und 1% *false positive* Entscheidungen. Beim Energieszenario waren 96,6% richtige Entscheidungen und ebenfalls 1,9% *false positive* Entscheidungen.

Insgesamt lässt sich festhalten, dass der Anteil der *false positive* Entscheidungen gering ausfällt. Betrachtet man den *loss*, der durch die *false positive* Entscheidungen entsteht, wirkt sich dieser nur gering auf das Gesamtergebnis der Verbesserung, d.h. gesparte Zeit oder Energie, aus.

Auffallend ist auch, dass der Anteil des *loss* bezogen auf den *benefit* nur einen sehr geringen Anteil ausmacht. Bei der Schachanwendung im Leistungsszenario umfasst der *loss* bei 550 *false positive* Entscheidungen 98 Millisekunden, wohingegen der *benefit* bei 227708 Sekunden liegt. Der *loss* beträgt also 0,0000434%. Beim Energieszenario der Schachanwendung liegt der *loss* bei 0,0001984%.
Wird für jeden "difficulty"-Parameter ein separates Vorhersagemodell verwendet liegt der *loss* beim Energieszenario bei 0,0003168% und beim Leistungsszenario bei 0,0000805%. Beim der Text-To-Speech Anwendung beträgt der *loss* des Energieszenarios 0,28%. Beim Leistungsszenario ist ein deutlich höherer *loss* von 24,6% zu beobachten.

Das legt den Schluss nahe, dass gerade Entscheidungen falsch getroffen werden, die nur einen geringen Verlust mit sich bringen.

Korrelieren die Metriken mit der Qualität der Entscheidungen?

Die Metriken korrelieren mit der Evaluation der Offloadingentscheidungen. Die Algorithmen, die die besten Metriken hatten, schnitten auch bei der Offloadingentscheidung am besten ab. Einzig bei der Text-To-Speech Anwendung ist jedoch zu beobachten, dass ein Algorithmus, der im Vergleich zu den anderen Algorithmen einen schlechteren Relativen Absoluten Fehler hatte, besser abschnitt. Der Regression Tree und KNN Regression sind dort leicht besser als der Support Vector Regression Algorithmus, der den besten MAE und RMSE hat.

Kann man Ausführungszeiten von unterschiedlichen Architekturen zusammenführen?

Die Ergebnisse aus Abschnitt 6.7.2 legen nahe, dass Ergebnisse unterschiedlicher Architekturen zusammengeführt werden können. Die Bedingung dafür ist allerdings, dass auf jeder Architektur der gleiche Testdatensatz unter möglichst gleichen Bedingungen ausgeführt wird. Das heißt, es sollten möglichst keine auslastenden Prozesse neben der Testausführung laufen, da diese die Ergebnisse beeinflussen können.

Wie verhält sich der Speicherbedarf der Vorhersagemodelle bei steigender Datensatzgröße?

Der benötigte Speicherplatz hängt stark von der Art des Machine Learning Algorithmus ab. Der KNN Algorithmus steigt linear zur Datensatzgröße an. Beim SVR Modell ist die Modellgröße von Grund auf im Vergleich zu den anderen Algorithmen hoch und steigt ebenfalls linear zur Datensatzgröße an. Die Vorhersagemodelle der anderen Algorithmen verändern sich in ihrem Speicherbedarf nur geringfügig.

Wie sehr beeinträchtigt die Übertragung der Vorhersagemodelle die Effizienz des Gerätes?

Die Übertragung der Vorhersagemodelle während einer mobilen Nutzung des Gerätes kann die Energieeffizienz des Gerätes beeinträchtigen. Beispielhaft wurde hier die Schachanwendung untersucht und die Übertragung der Modelle für Linear Regression, Support Vector Regression und KNN Regression betrachtet. Dabei war festzustellen, dass sich die Übertragung des Linear Regression Modelles nicht merkbar auf die Gesamteffizienz auswirkt. Bei der Übertragung der Support Vector Regression war festzustellen, dass eine Verschlechterung von 14% bei der Übertragung der unkomprimierten Modelle auftritt. Man sollte dabei allerdings beachten, dass in diesem Testszenario mehr als 6000 Ausführungen betrachtet worden sind. Betrachtet man den Fall, dass alle erstellten Vorhersagemodelle während der Ausführung von 60 Offloadingkandidaten übertragen werden, fällt der steigende Speicherbedarf bei der KNN - und Support Vector Regression stark ins Gewicht. Deswegen ist es vorteilhaft, die Übertragung der Modelle auf einen Zeitpunkt zu verschieben, bei dem der Akku des Mobilgerätes geladen wird und eventuell auch eine ungedrosselte Verbindung besteht.

Dauer zur Erstellung der Modelle

Die SVR Modelle eignen sich bei der Schach - und Text-To-Speech-Anwendung gut für eine Vorhersage. Die Modelle benötigen jedoch sehr lange zum Erstellen. Deshalb macht es Sinn, die Erstellung auf einen Server auszulagern.

7. Zusammenfassung und Ausblick

Kapitel 2 – *Code Offloading* führte einige Grundlagen zum Thema Offloading ein. Dabei wurde der Fokus auf die Offloadingentscheidung gelegt. Für das Offloading kann es mehrere Optimierungsziele, wie die Verbesserung der Leistung auf dem Mobilgerät oder der Einsparung von Energie geben. Die Entscheidung hängt dabei maßgeblich von der Ausführungszeit des Offloadingkandidaten ab.
Es wurde festgestellt, dass bei komplexen Parameterkombinationen von Offloadingkandidaten keine zuverlässige Aussage über Ausführungszeit mehr getroffen werden kann, wenn nur einfache, historienbasierte Verfahren verwendet werden.
Die Schlussfolgerung war, dass man mit Hilfe von Näherungsverfahren wie Machine Learning, Aussagen über die Ausführungszeit von unbekannten Parameterkonfigurationen eines Offloadingkandidaten machen kann.

Kapitel 3 – *Machine Learning* führte Grundlagen des Machine Learnings und eine Auswahl an Regressionsalgorithmen ein. Dabei wurde festgestellt, dass es sich bei der Vorhersage von Ausführungszeiten um *Supervised Learning* handelt, also der Erstellung eines Vorhersagemodelles anhand eines vorgefertigten Datensatzes. Gängige Metriken wurden vorgestellt, mit deren Hilfe die Leistung der Vorhersage bewertet werden kann. Da die Qualität der Vorhersage auch von der Größe des Datensatzes abhängt, mit dem die Modelle trainiert werden, wurde geschlussfolgert, dass aufgezeichnete Samples an einer zentralen Stelle gesammelt werden sollen, da der praktische Fall gegeben ist, dass Anwendungen, die Offloading benutzen, mehrere Anwender haben.

Kapitel 4 – *Kooperativer Systementwurf* stellte einen Entwurf zur Einbindung von Machine Learning gestützten Offloadingentscheidungen in bestehende Frameworks ein. Der Kern des Entwurfs besteht aus 3 Komponenten: Einem Client, einem Controller und einem Server.
Der Client stellt dabei die Komponente dar, welche die Vorhersage der Ausführungszeit nutzen möchte. Die Clientkomponente ist für die Aufzeichnung und Vorhersage der Ausführungszeit anhand eines Vorhersagemodelles zuständig.
Der Controller stellt ein Zwischenglied zwischen Client und Server dar, um beispielsweise persistent gespeicherte Vorhersagemodelle auf dem Mobilgerät zu verwalten oder zur Zwischenspeicherung von Samples. Auf einem Mobilgerät gibt es einen dedizierten Controller und mehrere Clients.
Der Server verwaltet die Samples pro Offloadingkandidat und leitet daraus Vorhersagemodelle ab. Die Vorhersagemodelle werden regelmäßig neu erstellt und an den Client zurück übertragen. Der Server befindet sich auf einer entfernten Maschine.

Beim Entwurf wurde darauf geachtet, dass so wenig Overhead wie möglich in Bezug auf die Übertragung der Samples und Vorhersagemodelle entsteht. Der Server benachrichtigt den zuständigen Controller über ein neues Vorhersagemodell. Dieser kann je nach aktuellen Bedingungen auf dem Mobilgerät entscheiden, ob das Modell empfangen werden soll. Dabei wurde festgestellt, dass die Bedingungen von dem jeweiligen Nutzer abhängen und entschieden werden sollten.

Kapitel 5 – *Implementierung* beschrieb das Vorgehen bei der Umsetzung des Entwurfs für das Offloadingframework von Berg [BDR14a]. Dabei wurde eine Teilmenge aus dem vorgestellten Entwurf implementiert. Die Komponente des Frameworks, die die Offloadingentscheidung trifft, befindet sich in der JVM der Anwendung, die Offloading nutzen möchte. Die Clientkomponente kann über Reflection-Calls eingebunden werden. Als Machine Learning Framework wurde Weka [WF05] verwendet. Es wurden verschiedene Algorithmen verwendet: Linear Regression, Support Vector Regression mit RBF Kernel, KNN Regression, Regression Trees und das RBF Network.

Kapitel 6 – *Evaluation* bewertete die Regressionsalgorithmen, den Speicherbedarf und die Dauer zur Erstellung der Vorhersagemodelle anhand einer Auswahl an Testszenarien- und Anwendungen. Außerdem wurde untersucht, ob ein Leistungsfaktor für eine Zusammenführung von Samples, die auf verschiedenen Architekturen erstellt worden sind, möglich ist. Zudem wurde der Einfluss der Übertragung von Vorhersagemodellen auf die Energieeinsparung untersucht.
Das Ergebnis war, dass sich Machine Learning Algorithmen grundsätzlich zur Vorhersage von Ausführungszeiten eignen. Anhand der erstellten Metriken war festzustellen, dass ein kooperativer Entwurf durchaus Sinn macht, da die Vorhersagequalität steigt, je mehr Samples zusammenkommen. Auch werden bei komplexeren Offloadingkandidaten viele Samples benötigt, bis sich die Vorhersagequalität nur noch wenig ändert.
Das Umrechnen von Ausführungszeiten auf andere Architekturen war nur geringfügig mit Fehlern belastet und kann als erster Ansatz für plattformübergreifende Vorhersagemodelle verwendet werden.
Die Leistung der Vorhersagemodelle bei der Offloadingentscheidung wurde ebenfalls untersucht. Dabei wurde ein Vergleich der Entscheidung mit der aufgezeichneten Ausführungszeit und der vorhergesagten Ausführungszeit gezogen. Daraus ergaben sich vier Fälle: Die Entscheidung war *true positive*, wenn sie mit der Entscheidung der aufgezeichneten Ausführungszeit übereinstimmt und sich ein Offloading lohnen würde. Die Entscheidung war *true negative*, wenn sie mit der Entscheidung der aufgezeichneten Ausführungszeit übereinstimmt und sich ein Offloading nicht lohnen würde.
Die Entscheidung war *false positive*, wenn die eigentliche Entscheidung negativ gewesen wäre, die Entscheidung mit Vorhersage allerdings positiv war. Die Entscheidung war *false negative*, wenn sich eigentlich Offloading gelohnt hätte, man mit der Vorhersage aber eine negative Entscheidung getroffen hat.
Der Anteil der richtigen Entscheidungen lag über alle Testszenarien hinweg zwischen 93% und 100%. Der Anteil der *false positive* Entscheidungen lag dabei zwischen 1.6% und 4.6%. Zusätzlich wurde ein *loss* und *benefit* errechnet, der sich aus den *false positive* Entscheidungen

und *true positive* Entscheidungen ergibt. Dabei war zu beobachten, dass der *loss*-Anteil bis auf ein Testszenario einer Testanwendung bei unter 1% lag.

Abschließend lässt sich festhalten, dass Machine Learning grundsätzlich für die Vorhersage von Ausführungszeiten und insbesondere für die Verbesserung der Offloadingentscheidung verwendet werden kann. Der Systementwurf kann dabei von bestehenden Offloadingframeworks integriert werden, um Vorhersagemodelle zu verwalten und zur Vorhersage zu verwenden.

Ausblick

Damit die Vorhersagemodelle auch plattformübergreifend, d.h. über Geräte mit unterschiedlichen Architekturen verwendet werden können, ist es nötig die aufgezeichneten Samples in geeigneter Weise zusammenzuführen. Untersucht wurde die Berechnung eines Leistungsfaktors anhand von reinen Ausführungszeiten auf Basis eines identischen Datensatzes.
Bei diesem Vorgehen ließ sich eine Standardabweichung von 3%-5% beobachten.

Ein alternativer Ansatz, der hier nicht weiter untersucht wurde, ist die Berechnung eines Leistungsfaktors anhand von Byte Codes, die bei Programmiersprachen wie Java oder C# plattformübergreifend verwendet werden. Anhand eines vorgefertigten Benchmarks könnte hier ein Leistungsfaktor erstellt werden.
Bei diesem Ansatz würde nicht mehr die Ausführungszeit aufgezeichnet werden, sondern die erzeugte Länge des Byte Codes. Dieser könnte verwendet werden, um Vorhersagemodelle zu erstellen. Um nun eine Vorhersage zu treffen, wird nun anstatt der Ausführungszeit der Länge des Byte Codes vorhergesagt. Diese kann nun anhand des individuellen Leistungsfaktors berechnet werden.

Für diesen Ansatz müsste untersucht werden, inwiefern sich diese Art der Vorhersage von dem in dieser Arbeit verwendeten Vorgehen unterscheidet und wie die Berechnung eines Leistungsfaktors anhand eines Benchmarks stattfinden kann. Der Vorteil dieses Ansatzes ist, dass keine Umrechnung mehr für die Erstellung der Samples gemacht werden muss, sondern nur noch ein individueller Leistungsfaktor für das jeweilige Gerät lokal berechnet werden muss.

Die Erstellung mehrere Vorhersagemodelle durch Aufteilung des Parameterraumes kann von Vorteil sein. Eine manuelle Aufteilung ist aber nur bei sehr kleinen Wertebereichen praktisch umsetzbar.
Gupta et. al [GMD08] haben einen eigenen Ansatz zur Vorhersage von Ausführungszeiten entwickelt. Der Kontext war die Vorhersage von Anfragen in Data Warehouses, woraus auch der Name des Algorithmus *Predicting Query Runtime*(PQR) entstand. Der PQR Algorithmus erzeugt einen Binärbaum bei dem jeder Knoten eine Zeitspanne darstellt. Kindknoten stellen Teilmengen des Elternknotens dar. Jeder Knoten bekommt einen binären Klassifikator zugewiesen, der entscheidet, zu welchem Kindknoten eine neues Sample zugewiesen wird.
Das Ergebnis einer Vorhersage ist die Zuweisung zu einem Blatt des Baumes - also einer

Zeitspanne.
Matsunga und Fortes [MF10] haben den PQR Algorithmus aufgegriffen und ihn so modifiziert, dass er für numerische Vorhersagen verwendet werden kann - er wird von den Autoren PQR2 genannt. Jedes Blatt bekommt einen Regressionsalgorithmus zugewiesen, der für diesen Wertebereich die geringste Fehlerrate hat. Durch diese Vorgehensweise erreicht meine ein deutlich feineres Vorhersagemodell.
Der PQR2 Algorithmus kann hierfür ein Ansatz zur automatisierten Aufteilung des Parameterraumes sein, auf welchen unterschiedlichste Machine Learning Modelle eingelernt werden können.

Literaturverzeichnis

[Acc10] Accenture. *Accenture Mobile Web Watch Studie 2010*. 2010. URL: http://www.iab-austria.at/wp-content/uploads/2012/01/Accenture-Mobile-Web-Watch-Studie-2010.pdf (Zitiert auf S. 48).

[AGHH00] K. Arnold, J. Gosling, D. Holmes und D. Holmes. *The Java programming language*. Bd. 2. Addison-wesley Reading, 2000 (Zitiert auf S. 44, 63).

[BB12] J. Bergstra und Y. Bengio. „Random search for hyper-parameter optimization". In: *The Journal of Machine Learning Research* 13.1 (2012), S. 281–305 (Zitiert auf S. 25).

[BC64] G. E. Box und D. R. Cox. „An analysis of transformations". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1964), S. 211–252 (Zitiert auf S. 25).

[BDR14a] F. Berg, F. Durr und K. Rothermel. „Increasing the Efficiency and Responsiveness of Mobile Applications with Preemptable Code Offloading". In: *Mobile Services (MS), 2014 IEEE International Conference on*. Juni 2014, S. 76–83 (Zitiert auf S. 52, 63, 110).

[BDR14b] F. Berg, F. Dürr und K. Rothermel. „Optimal Predictive Code Offloading". In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. MOBIQUITOUS '14. London, United Kingdom: ICST (Institute for Computer Sciences, Social-Informatics und Telecommunications Engineering), 2014, S. 1–10. URL: http://dx.doi.org/10.4108/icst.mobiquitous.2014.258023 (Zitiert auf S. 18).

[BFH+15] R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald und D. Scuse. *WEKA manual for version 3-7-12*. 2015 (Zitiert auf S. 66).

[BFSO84] L. Breiman, J. Friedman, C. J. Stone und R. A. Olshen. *Classification and regression trees*. CRC press, 1984 (Zitiert auf S. 38).

[Bis06] C. M. Bishop. „Pattern Recognition". In: *Machine Learning* (2006) (Zitiert auf S. 13, 21, 27, 28, 37).

[BKK+98] J. P. Bradford, C. Kunz, R. Kohavi, C. Brunk und C. E. Brodley. „Pruning decision trees with misclassification costs". In: *Machine Learning: ECML-98*. Springer, 1998, S. 131–136 (Zitiert auf S. 39).

[CBC+10] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra und P. Bahl. „MAUI: making smartphones last longer with code offload“. In: *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM. 2010, S. 49–62 (Zitiert auf S. 12, 14, 18, 19).

[CH10] A. Carroll und G. Heiser. „An Analysis of Power Consumption in a Smartphone.“ In: *USENIX annual technical conference*. Bd. 14. Boston, MA. 2010 (Zitiert auf S. 77).

[CIM+11] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik und A. Patti. „Clonecloud: elastic execution between mobile device and cloud“. In: *Proceedings of the sixth conference on Computer systems*. ACM. 2011, S. 301–314 (Zitiert auf S. 14, 18, 19).

[CL] C.-C. Chang und C.-J. Lin. *LibSVM*. URL: http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (Zitiert auf S. 98).

[dCP16] A. M. R. da Cruz und S. Paiva. „Modern Software Engineering Methodologies for Mobile and Cloud Environments“. In: (2016) (Zitiert auf S. 12).

[Ecl] Eclipse-Foundation. *Eclipse Mars*. URL: https://eclipse.org/ (Zitiert auf S. 64).

[Edw94] S. J. Edwards. „Portable game notation specification and implementation guide“. In: *Retrieved April* 4 (1994), S. 2011 (Zitiert auf S. 84).

[EFC+15] H. Eom, R. Figueiredo, H. Cai, Y. Zhang und G. Huang. „MALMOS: Machine Learning-Based Mobile Offloading Scheduler with Online Training“. In: *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2015 3rd IEEE International Conference on*. IEEE. 2015, S. 51–60 (Zitiert auf S. 15).

[FFI04] I. R. Forman, N. Forman und J. V. Ibm. „Java reflection in action“. In: (2004) (Zitiert auf S. 64).

[FHT+15] H. Flores, P. Hui, S. Tarkoma, Y. Li, S. Srirama und R. Buyya. „Mobile code offloading: from concept to practice and beyond“. In: *Communications Magazine, IEEE* 53.3 (2015), S. 80–88 (Zitiert auf S. 11, 17).

[Fle09] T. Fletcher. „Support vector machines explained“. In: *Online]. http://sutikno. blog. undip. ac. id/files/2011/11/SVM-Explained. pdf.[Accessed 06 06 2013]* (2009) (Zitiert auf S. 29–34).

[FPS02] J. Flinn, S. Park und M. Satyanarayanan. „Balancing performance, energy, and quality in pervasive computing“. In: *Distributed Computing Systems, 2002. Proceedings. 22nd International Conference on*. 2002, S. 217–226 (Zitiert auf S. 14).

[GMD08] C. Gupta, A. Mehta und U. Dayal. „PQR: Predicting query execution times for autonomous workload management“. In: *Autonomic Computing, 2008. ICAC'08. International Conference on*. IEEE. 2008, S. 13–22 (Zitiert auf S. 111).

[Gun+98] S. R. Gunn et al. „Support vector machines for classification and regression“. In: *ISIS technical report* 14 (1998) (Zitiert auf S. 29, 33).

[HCL+03] C.-W. Hsu, C.-C. Chang, C.-J. Lin et al. „A practical guide to support vector classification“. In: (2003) (Zitiert auf S. 25).

[HFH+09] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann und I. H. Witten. „The WEKA data mining software: an update“. In: *ACM SIGKDD explorations newsletter* 11.1 (2009), S. 10–18 (Zitiert auf S. 65).

[HQG+12] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen und O. Spatscheck. „A close examination of performance and power characteristics of 4G LTE networks“. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, S. 225–238 (Zitiert auf S. 77, 80).

[jji] jjil. *jjil*. URL: https://code.google.com/archive/p/jjil/ (Zitiert auf S. 84).

[KAH+12] S. Kosta, A. Aucinas, P. Hui, R. Mortier und X. Zhang. „Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading“. In: *INFOCOM, 2012 Proceedings IEEE*. IEEE. 2012, S. 945–953 (Zitiert auf S. 15).

[KLLB13] K. Kumar, J. Liu, Y.-H. Lu und B. Bhargava. „A survey of computation offloading for mobile systems“. In: *Mobile Networks and Applications* 18.1 (2013), S. 129–140 (Zitiert auf S. 11, 17–19).

[KM76] D. E. Knuth und R. W. Moore. „An analysis of alpha-beta pruning“. In: *Artificial intelligence* 6.4 (1976), S. 293–326 (Zitiert auf S. 83).

[KZP07] S. B. Kotsiantis, I. Zaharakis und P. Pintelas. *Supervised machine learning: A review of classification techniques*. 2007 (Zitiert auf S. 13).

[LGZ+09] M. Lv, N. Guan, Y. Zhang, Q. Deng, G. Yu und J. Zhang. „A survey of WCET analysis of real-time operating systems“. In: *Embedded Software and Systems, 2009. ICESS'09. International Conference on*. IEEE. 2009, S. 65–72 (Zitiert auf S. 12).

[Loh11] W.-Y. Loh. „Classification and regression trees“. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.1 (2011), S. 14–23 (Zitiert auf S. 38).

[LSJ+13] F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu und B. Li. „Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications“. In: *IEEE Wireless Communications* 20.3 (Juni 2013), S. 14–22 (Zitiert auf S. 17).

[Mar] Marvin-Project. *Marvin Project*. URL: http://marvinproject.sourceforge.net (Zitiert auf S. 85).

[MF10] A. Matsunaga und J. A. B. Fortes. „On the Use of Machine Learning to Predict the Time and Resources Consumed by Applications“. In: *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*. CCGRID '10. Washington, DC, USA: IEEE Computer Society, 2010, S. 495–504. URL: http://dx.doi.org/10.1109/CCGRID.2010.98 (Zitiert auf S. 112).

[NFS00] D. Narayanan, J. Flinn und M. Satyanarayanan. „Using history to improve mobile application adaptation“. In: *Mobile Computing Systems and Applications, 2000 Third IEEE Workshop on*. IEEE. 2000, S. 31–40 (Zitiert auf S. 14).

[Ng11] A. Ng. *Cs229 lecture notes on machine learning*. Techn. Ber. Technical report, Stanford University, Department of Computer Science, Stanford, USA, 2011. 39, 116, 140, 2011 (Zitiert auf S. 27, 28).

[Nil97] N. J. Nilsson. „Introduction to machine learning“. In: (1997) (Zitiert auf S. 21, 22, 27, 35).

[Orr+96] M. J. Orr et al. *Introduction to radial basis function networks*. 1996 (Zitiert auf S. 37).

[PMS+09] P. S. Pillai, L. B. Mummert, S. W. Schlosser, R. Sukthankar und C. J. Helfrich. „SLIPstream: scalable low-latency interactive perception on streaming data“. In: *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*. ACM. 2009, S. 43–48 (Zitiert auf S. 15).

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993 (Zitiert auf S. 38).

[RPL10] J. D. Rodriguez, A. Perez und J. A. Lozano. „Sensitivity Analysis of k-Fold Cross Validation in Prediction Error Estimation“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.3 (März 2010), S. 569–575 (Zitiert auf S. 24).

[RSM+11] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall und R. Govindan. „Odessa: Enabling Interactive Perception Applications on Mobile Devices“. In: *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services*. MobiSys '11. Bethesda, Maryland, USA: ACM, 2011, S. 43–56. URL: http://doi.acm.org/10.1145/1999995.2000000 (Zitiert auf S. 15).

[Sha50] C. E. Shannon. „A chess-playing machine“. In: *Scientific American* 182.2 (1950), S. 48–51 (Zitiert auf S. 12, 83).

[SKP01] F. Schwenker, H. A. Kestler und G. Palm. „Three learning phases for radial-basis-function networks“. In: *Neural networks* 14.4 (2001), S. 439–458 (Zitiert auf S. 37, 38).

[SV08] A. Smola und S. Vishwanathan. „Introduction to machine learning“. In: *Cambridge University* (2008), S. 32–34 (Zitiert auf S. 21–23, 35).

[Tel] Telekom. *Telekom Smartphone-Tarife*. URL: https://www.t-mobile.de/telefonieren-und-surfen/0,21919,25250-_,00.html (Zitiert auf S. 48).

[vKem] H. van Kempen. *CEGT*. URL: http://www.husvankempen.de/nunn/downloads/40_120_new/cegt40120.rar (Zitiert auf S. 84).

[Wei16] G. M. Weiss. „Sample Weka Data Sets“. In: (2016). URL: http://storm.cis.fordham.edu/~gweiss/data-mining/datasets.html (Zitiert auf S. 38).

[WF05] I. H. Witten und E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005 (Zitiert auf S. 12, 13, 21, 24, 25, 28, 36, 38, 39, 66, 78, 110).

[WF06] P. I. Wilson und J. Fernandez. „Facial Feature Detection Using Haar Classifiers“. In: *J. Comput. Sci. Coll.* 21.4 (Apr. 2006), S. 127–133. URL: http://dl.acm.org/citation.cfm?id=1127389.1127416 (Zitiert auf S. 84).

[WLK02] W. Walker, P. Lamere und P. Kwok. „FreeTTS: a performance case study“. In: (2002) (Zitiert auf S. 87).

[Yal] Yale. *Yale Faces*. URL: http://vision.ucsd.edu/content/yale-face-database (Zitiert auf S. 85).

Alle URLs wurden zuletzt am 24. 04. 2016 geprüft.

A. Testfälle der Text-To-Speech Anwendung

Name	Autoren	Anzahl Seiten
Coherent XRay Scattering for the Hydrogen Atom in the Hydrogen Molecule	Robert F. Stewart, Ernest R. Davidson, und William T. Simpson	14
Direct Expression Cloning of Vascular Cell Adhesion Molecule 1, a Cytokine-Induced Endothelial Protein That Binds to Lymphocytes	Laurelee Osborn, Catherine Hession, Richard lizard, Cornelia Vassallo, Stefan Luhowskyj, Gloria Chi-Rosso, und Roy Lobb	9
Technical Note Q-Learning	CHRISTOPHER J.C.H. WATKINS und PETER DAYAN	14
Geochemical Surveillance of Active Volcanoes: Data on the Fumaroles of Vulcano (Aeolian Islands, Italy)*	M. MARTINI G. PICCARDI P. CELLINI LEGITTIMO	9
Genetic Algorithms and Machine Learning	David E. Goldberg und John H. Holland	5
Survey radiography and computerized tomography imaging of the thorax in female dogs with mammary tumors	Carolina C Otoni1 , Sheila C Rahal , Luiz C Vulcano , Sérgio M Ribeiro , Khadije Hette , Tatiana Giordano, Danuta P Doiche, Renée L Amorim	10
COMPARATIVE STUDY OF THE SEISMIC PERFORMANCE OF FRAMES USING DIFFERENT DISSIPATIVE BRACES	Alfonso VULCANO und Fabio MAZZA	8
Criminal Law	Keine Angaben, Creative Commons license (CC BY-NC-SA)	526
Principles and Realization Strategies of Multilevel Transaction Management	GERHARD WEIKUM	59
Single Molecule Detection Using Surface-Enhanced Raman Scattering (SERS)	Katrin Kneipp, Yang Wang, Harald Kneipp, Lev T. Perelman, Irving Itzkan, Ramachandra R. Dasari, und Michael S. Feld	4
Sources of American Law An Introduction to Legal Research	Beau Steenken und Tina M. Brooks	183

Testfälle für die Text-To-Speech Anwendung

B. Machine Learning Ergebnisse

samplesize	RMSE	RRSE	MAE	RAE
50	2.990235126794489E8	44.28710194334373	2.011270671693653E8	54.82036263639238
100	1.800558670751974E8	16.057618612232304	1.055166194973314E8	28.76022320824156
150	1.1440082306306218E8	6.482236651940671	6.732197710012598E7	18.34966944016547
200	9.3245810163902E7	4.306507792348997	5.875289094885371E7	16.01402950424547
250	7.416302829105039E7	2.724215716038881	4.9243288919776306E7	13.422037093185239
300	6.902249182851177E7	2.3596514232418078	4.659077091240609E7	12.699051365256597
350	6.5942186775620885E7	2.153740001548451	4.536416494808139E7	12.364720513011743
400	6.235647697577819E7	1.925882224908624	4.1459354967580654E7	11.300402804958079
450	5.807907584283879E7	1.6707288749517355	3.895128911255288E7	10.616789795413244
500	5.778357912715161E7	1.6537713400981515	3.8423950660202876E7	10.473055361272673
750	4.882977644014825E7	1.1809623601728179	3.194331047698143E7	8.706654399133603
1250	4.008109807016756E7	0.7956934923254992	2.5352567485217568E7	6.910243175439347
1750	3.641125395187377E7	0.6566559070944228	2.2265355081282202E7	6.068774615781039
2000	3.4497755544151925E7	0.5894517546746951	2.027779969253587E7	5.527034964800906
2250	2.985889072885983E7	0.4415846038045415	1.796007213826559E7	4.895301669001083
2750	2.7822537951144293E7	0.3834070294280306	1.6216637969223082E7	4.420101116809342
3250	2.557448126540706E7	0.323951690922005 76	1.4868777440537352E7	4.052720415615059
3500	2.5202311274838544E7	0.31459174801937034	1.4296435870054016E7	3.896719669980172
3750	2.428651023975562E7	0.2921438941713626	1.3900202228351869E7	3.7887199251792465
4250	2.369404716199374E7	0.278064201884472	1.3699901379982447E7	3.7341247615420112
4750	2.2858681584723916E7	0.2588027818395765	1.2812344899688838E7	3.4922072076555253
5000	2.2628394522812076E7	0.2536144930175323	1.273554143555055E7	3.4712732089896643
5250	2.1540379292524327E7	0.22981230389269644	1.2447053745426593E7	3.3926413271086786
5750	2.122238786649095E7	0.2230771457915771	1.2277115297121871E7	3.346321915738325
6250	2.06770305421494E7	0.21175950960782486	1.1771583117959544E7	3.208531126183842
6500	1.9566179633693468E7	0.18961760499909464	1.1512063829237368E7	3.137795040190462
6750	1.9308027271869875E7	0.18464705749634763	1.141355261560597E7	3.1109442511294523
7250	1.9324003637697965E7	0.1849527551605963	1.141078524761289E7	3.1101899612217627
7750	1.89625873202007E7	0.17809911909887752	1.1155838911001856E7	3.0407003056396644
8000	1.8951288210294895E7	0.17788693687493684	1.1265556904640183E7	3.070605679807556
8250	1.8090255553586476E7	0.16208991259134536	1.1043146751300309E7	3.009984275479911
8750	1.8288505812116627E7	0.16566205080556334	1.1093616628935738E7	3.0237406387239427
9250	1.812313567257834E7	0.1626796642365988	1.1049183358834945E7	3.0116296483220637
9500	1.8247307404606797E7	0.1649165196598867	1.108983593499908E7	3.0227101508063208
9750	1.8257500364636466E7	0.16510081611045874	1.1064813703780912E7	3.0158899460041666
10250	1.8048673747913532E7	0.16134561740547199	1.101436056452071E7	3.0021381450689915
10750	1.7963828190629423E7	0.15983223393818968	1.0943638990255194E7	2.9828618616626046
11000	1.7941501292475276E7	0.15943517599209642	1.0903503010331333E7	2.9719221656527175
11250	1.793655605936165E7	0.15934729756552102	1.089993725376104E7	2.970950262313164
11750	1.7838155109280776E7	0.15760371703118636	1.085530230078435E7	2.95878430005419
12250	1.7702384523715638E7	0.15521372588109267	1.0745328971205065E7	2.9288093300378915
12500	1.770659312650634E7	0.1552875363408666	1.0738269041736057E7	2.926885034620445
12750	1.7470094252202537E7	0.15116702986429353	1.0697077822620934E7	2.915657716482183
13250	1.7565768574373808E7	0.15282728485886246	1.0739716465769578E7	2.927279552929308
13750	1.7515254222649004E7	0.15194956966823572	1.0730700400175571E7	2.924822081678059
14000	1.759410643224246E7	0.15332077777648653	1.0742630709712774E7	2.9280738761997926
14250	1.7388520968646225E7	0.1497586342668834	1.0690297594704865E7	2.9138096581459902
14750	1.7427406873834413E7	0.15042919291910015	1.0682206230105303E7	2.9116042287733355
15250	1.7325832967215214E7	0.14868077987575873	1.0648515225740068E7	2.902421212768179
15500	1.7321669015109167E7	0.14860932296480558	1.0643810080451101E7	2.901138751015874
15750	1.71467913435344E7	0.14562378339945828	1.059226484403659E7	2.8870892817316496
16250	1.7112355714639254E7	0.14503946286443412	1.0574061301395534E7	2.882127618327045
16750	1.7048234846154712E7	0.14395455876521818	1.05351967678722E7	2.871534475139364
17000	1.7035484895967036E7	0.14373931922398697	1.0559140022497095E7	2.878060587808858
17250	1.7004660729001712E7	0.14321962326263776	1.0557125301412838E7	2.877511443717988
17750	1.6703899662598612E7	0.1381981817874023	1.0509412913438076E7	2.864506677885937
18250	1.672795137963636E7	0.13859644765972517	1.0514884700577026E7	2.8659980999975807
18500	1.6710965460899152E7	0.13831512298374388	1.0516449459918281E7	2.8664245999000233
18750	1.6707328179664029E7	0.13825491865245498	1.051450237037799E7	2.8658938898559376
19250	1.6658390352986766E7	0.13744617339977963	1.0481202629527811E7	2.8568175189089704
19750	1.6750315587283242E7	0.13896728466756875	1.0516795224223197E7	2.866518843428997

KNN Regression für Gesichtserkennung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	2.6006133911025774E8	89.99526122420846	9.531439872272445E7	72.57525706745642
100	2.128549335296847E8	96.17653009620243	8.781763842314777E7	87.81931061562463
150	1.6237504670407227E8	85.53089710535863	5.8598617251435556E7	78.23242677785083
200	1.4973254544684377E8	94.4097409267683	5.8147096568156384E7	86.45785678949093
250	1.211367252667686E8	76.31308848850294	4.818564451246298E7	81.48179540330429
300	1.1401949247520529E8	78.15068272530333	4.623290073958758E7	75.65164656438887
350	1.1400965004976197E8	82.67620271771237	5.2758421343542635E7	74.77941397875608
400	9.965295942965606E7	64.67547872418051	4.482248503615463E7	57.97780263593534
450	9.744052470021346E7	61.147131261102984	4.83229324813067E7	56.5287670920911
500	9.505808612357953E7	63.600960357126425	4.716243211042681E7	58.166565969492936
550	9.254227558845878E7	63.491174304791656	4.8952655305227734E7	59.71768223409716
700	9.407648206237406E7	64.95347615078873	5.208429286167487E7	63.47315576289123
800	8.872429950171077E7	55.29174702593965	4.9851548216960825E7	56.630197212990105

REPTREE für Text-To-Speech(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.757961596669753E8	69.94734899140316	1.846898896576116E8	50.34014997981197
100	4.4276809171369445E8	97.09999677009336	2.069081467598732 7E8	56.39608729663724
150	4.415306161328384E8	96.55799311883125	2.097774281370650 2E8	57.17815530875142
200	4.892289928905228E8	118.54710617579023	2.217507528726850 7E8	60.44167430302607
250	5.103850656107795E8	129.02162280555308	2.258250872846726 2E8	61.55219857561941
300	5.155138538548486E8	131.6276918935483	2.277662250599227E8	62.08128637202625
350	5.439477152501217E8	146.54833814175527	2.342218410724874 4E8	63.84086660073852
400	5.051205083167754 4E8	126.3736668805798	2.237991717513763 6E8	61.000003260642345
450	4.8398941026567566E8	116.02145371369099	2.156826807715919 6E8	58.787725295727235
500	5.280824754360980 4E8	138.1243016128037	2.281916921067821 7E8	62.197254143678926
750	4.749189684163445E8	111.71348896531823	2.138435843330192 6E8	58.28645047924043
1250	5.285719171670744E8	138.38045525913833	2.320801629738249 8E8	63.2571183679846
1750	5.162503377298500 5E8	132.00405780204412	2.280331849010555E8	62.15405049823871
2000	4.911624993233482E8	119.48598974579407	2.204468153977603 3E8	60.086265524702064
2250	5.081799176961667E8	127.9091406546675	2.260777211782976E8	61.62105793829834
2750	4.749131023950821E8	111.71072929599335	2.164178350409841 5E8	58.98810321705314
3250	5.019222493492663E8	124.77841920377787	2.25105193390912E8	61.35598011098854
3500	4.932432163752345 4E8	120.50049368345542	2.224073627706342 3E8	60.62064371387075
3750	4.979779659580511 5E8	122.82501842381109	2.235931838482937 2E8	60.94385800031201
4250	5.041286363204049 5E8	125.87785080561933	2.254937525644960 7E8	61.46188805814534
4750	5.021329629696102E8	124.88320846728678	2.252401165638843 8E8	61.39275555536058
5000	5.06188738291482E8	126.90874269642127	2.263277738638720 8E8	61.68921375190573
5250	5.099485712765872 5E8	128.80103200390406	2.272051305703861E8	61.92835119615667
5750	5.113257910587576 6E8	129.4976781934073	2.281666672341444 8E8	62.19043321891612
6250	5.113604902988788E8	129.51525455499814	2.280673706506960 4E8	62.16336836489219
6500	5.103940477235 06E8	129.02616409540596	2.274763254573292 4E8	62.00226965107473
6750	5.137614169418776 6E8	130.73430306981345	2.285707765259468 6E8	62.300579596691755
7250	5.202221703779514 4E8	134.0430488195339	2.302962910926635 6E8	62.770895877901836
7750	5.219145154492866E8	134.91658357581906	2.306245940832341 3E8	62.860380049530185
8000	5.310671319662684E8	139.69003659197998	2.333720771055468 3E8	63.60924999399013
8250	5.298427165760501 6E8	139.0466472758719	2.329198372479458 5E8	63.48598486940379
8750	5.362396037868669E8	142.4243857335153	2.347376465603486E8	63.98145754303361
9250	5.426765508668139E8	145.86419386331136	2.365787680145019 3E8	64.48328430953907
9500	5.4267524976803E8	145.8634944282706	2.365630157334612E8	64.47899077624612
9750	5.457359256218156E8	147.5134677176033	2.374009996128396E8	64.70739653384757
10250	5.43320187747898E8	146.21040100845286	2.366686500831380 2E8	64.50778308867649
10750	5.491873426482068E8	149.38521713384657	2.383019826872448E8	64.9529737182777
11000	5.478785938850192E8	148.6740763290102	2.380125693327074E8	64.87408953192305
11250	5.478697407808573E8	148.66927155487147	2.380210933503345 8E8	64.87641288845566
11750	5.456619369475774E8	147.47347186782983	2.374694918798771 5E8	64.72606518431631
12250	5.489508712935325E8	149.25659902246932	2.384606710656867 3E8	64.99622674520641
12500	5.52493774254768E8	151.18940614533537	2.394923123019445 5E8	65.27741687778408
12750	5.495411568283767E8	149.57776211740966	2.386950762900466 6E8	65.06011759582204
13250	5.529589432898 43E8	151.44409946441573	2.396331205111932E8	65.31579638185495
13750	5.546269706254531E8	152.35915440761366	2.399689566915363 E8	65.40734446744625
14000	5.534452069878092E8	151.7105718615474	2.395392580580161 2E8	65.29021268596794
14250	5.550492511174978E8	152.59124839137522	2.400318913567084 4E8	65.42448768167667
14750	5.575422702154497E8	153.96506238423171	2.407239411756919E8	65.61311680342077
15250	5.606018243780 86E8	155.65948798433857	2.416228996329477 4E8	65.85814214643099
15500	5.530669661559258E8	151.50327573352385	2.393563010648306E8	65.24034486432438
15750	5.515983263361789E8	150.69972631273993	2.389936926829556 5E8	65.14151021581513
16250	5.478321662189419E8	148.64887987862252	2.377766295409009E8	64.8097804106833
16750	5.435157483418932E8	146.31567277762593	2.364341718767021 6E8	64.44387234564198
17000	5.473782835362 77E8	148.4026686949541	2.376078538109936 7E8	64.76377791659858
17250	5.467616461977646E8	148.06849730112995	2.373531227178106E8	64.69434692901649
17750	5.484742855384712E8	148.9975496450239	2.376742268119686E8	64.7818689275894
18250	5.48391421429532E8	148.95253161283625	2.375706093225901 4E8	64.75362634232654
18500	5.456889120673798E8	147.48805310442407	2.367950403622706 5E8	64.54223275789943
18750	5.435123291423382E8	146.3138318708586	2.361117657594693 3E8	64.35599546009077
19250	5.412207539257324E8	145.08264610286025	2.354228963947058 6E8	64.16823322143776
19750	5.40767440871701E8	144.83971264491908	2.354287674825031 5E8	64.16983347925807

Linear Regression für Gesichtserkennung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.305035684789531E8	54.10269387554645	2.3156049578816324E8	63.11547485888783
100	2.7329952223964554E8	36.99511051701248	1.936900512901687E8	52.79328635487504
150	2.199943436270299E8	23.971191053750136	1.607038538144213E8	43.802376612759744
200	2.329235821586028E8	26.871598701564142	1.7138173235922506E8	46.712801262469235
250	2.2131546348238215E8	24.259961259648914	1.6283668137310457E8	44.38371249088623
300	2.090646459906067E8	21.648498447890493	1.5361995184217826E8	41.871547110473
350	2.1939651632029712E8	23.841086244305046	1.6336847163084453E8	44.5286603349835
400	2.142093159390709E8	22.72706179011321	1.5510398107145452E8	42.27604274428754
450	2.1933995059801394E8	23.828794213084837	1.628721672453077E8	44.39338472650337
500	2.165390376845997E8	23.224105135467088	1.5303776315752822E8	41.71286237815677
750	2.1122517025180382E8	22.098251996427077	1.5670914460593554E8	42.71355544851676
1250	2.148840190611407E8	22.870455820819775	1.5608483478531313E8	42.54338993451778
1750	2.10068457532221E8	21.85688550045718	1.555941851732231E8	42.409655623961115
2000	2.1134895200527093E8	22.12415953256621	1.5802876645096445E8	43.07323924993406
2250	2.1038675075422606E8	21.92317026396293	1.5631847015298462E8	42.60707094851901
2750	2.095289546413359E8	21.744762900132205	1.568000084722862E8	42.73832183214673
3250	2.095744582136088E8	21.754208581119514	1.5487544130472755E8	42.21375061690192
3500	2.100542790132402E8	21.85393514978278	1.5473515747211906E8	42.17551403997643
3750	2.1045641069056574E8	21.93769037480755	1.5674712301609272E8	42.723907064768476
4250	2.1027616657904923E8	21.900129663323316	1.5397215350943157E8	41.96754524434779
4750	2.1012790231588987E8	21.869257292768456	1.5377505102384162E8	41.913821845062714
5000	2.1129762105919525E8	22.11341411727712	1.5380463498070458E8	41.921885420325225
5250	2.1094069050368118E8	22.03876787531881	1.5360326747721758E8	41.86699952296837
5750	2.1008857152674237E8	21.861071281869243	1.537964267128358E8	41.91964812711792
6250	2.0992971779262185E8	21.828024268668596	1.535671579369015E8	41.857157296744816
6500	2.095521028639817E8	21.74956777703724	1.533148493587492E8	41.78838660394227
6750	2.0928410332608 86E8	21.693971610605942	1.531860717580876E8	41.75328623248521
7250	2.1253499474781042E8	22.373167872974474	1.5328099907936382E8	41.77916017501171
7750	2.1277748878236443E8	22.42425079990897	1.530373839617241E8	41.712759022344024
8000	2.1297792237407663E8	22.46651739558464	1.5309004463241816E8	41.727112520879814
8250	2.0842086650458458E8	21.515377890574992	1.5245695048212364E8	41.55455269891961
8750	2.078303142667352E8	21.393624691236266	1.521129843240 3907E8	41.4607992832992
9250	2.0912753787404466E8	21.66152522841981	1.5190886917194515E8	41.40516447086519
9500	2.092384490234146E8	21.684507775537796	1.5199364377058482E8	41.42827112828922
9750	2.075849071314949E8	21.34313110982556	1.5189930609669563E8	41.40255790348231
10250	2.0920541579941115E8	21.677661494054192	1.5194152591681004E8	41.41406558308849
10750	2.0878988328222224E8	21.591632866108217	1.5184458625660846E8	41.3876431457662
11000	2.072173128009491E8	21.267608591955643	1.5185484839884564E8	41.39044025491111
11250	2.08721285090228E8	21.577447277830274	1.5182503186290225E8	41.38231328654239
11750	2.0859860361452654E8	21.552089297636474	1.5183548483142143E8	41.38516240840806
12250	2.0895665375471368E8	21.62613918262756	1.5205055047740418E8	41.44378195111473
12500	2.0877383441499883E8	21.588313663801223	1.5195458386735412E8	41.41762473399836
12750	2.0859216593079913E8	21.55075905485243	1.5191290300761044E8	41.406263956563954
13250	2.070629703859033E8	21.23593873221221	1.5182420205965486E8	41.3820871105454
13750	2.0685866539280057E8	21.194053233341574	1.5175796542283642E8	41.364033267761805
14000	2.0830305495386848E8	21.491061288505286	1.5175326851253703E8	41.36275305058613
14250	2.081869772347324E8	21.467116000670014	1.516749748934296E8	41.34141288661919
14750	2.079746765594407E8	21.423355727906518	1.5158423907109886E8	41.316681404311636
15250	2.0829513152736887E8	21.489426366675186	1.517144060701856E8	41.352160477380416
15500	2.0850137047399157E8	21.532002021018886	1.5180633182982332E8	41.3772162968178
15750	2.0685221613225004E8	21.192731714221978	1.517956960601424E8	41.37431734960482
16250	2.0798691964836448E8	21.42587810970579	1.5158805446109912E8	41.31772134829006
16750	2.0810974124256206E8	21.45119063909803	1.5163209209392852E8	41.32972450149729
17000	2.0824674658625394E8	21.479443955102752	1.5168529213105163E8	41.344225012882475
17250	2.0835348390240553E8	21.501468268002956	1.5167820240988317E8	41.34229259726624
17750	2.067574151569682E8	21.173310783794765	1.51600888456343E8	41.32121945664138
18250	2.084115937195875E8	21.513463465800303	1.5163755573686 71E8	41.33121370377834
18500	2.0665840270683354E8	21.15303659525248	1.5154829897296205E8	41.30688536133348
18750	2.0666218369598362E8	21.153810627495595	1.5156143742979923E8	41.3104664555066
19250	2.0668736900074434E8	21.158966845447306	1.5156162441686368E8	41.31051742178789
19750	2.0854229654675257E8	21.540455747393594	1.517781363046526E8	41.36953116056868

RBFNETWORK für Gesichtserkennung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	7.36092104688764E7	2.6836810719111193	5.370162978171308E7	14.637228396115187
100	4.574052227768961E7	1.0362602096895623	2.943689554181719E7	8.023491373904038
150	2.590451638675724E7	0.3323667433265193	1.7741488913070004E7	4.835723354455465
200	3.467804361382207E7	0.5956288991340265	2.3274327569945946E7	6.343786022734596
250	2.761799517186904 7E7	0.3777903694326751	1.8122667515589572E7	4.939619610257571
300	1.9289906262950093E7	0.1843006294759505	1.3313843291962484E7	3.6288985248060825
350	2.3751573825883552E7	0.27941606247684947	1.6445355539395055E7	4.482441707335548
400	1.790520039143405E7	0.15879066075251735	1.2270779563594593E7	3.344595011376713
450	1.742506711642573E7	0.15038880318194306	1.1435598750568075E7	3.1169532738351613
500	1.7194167808250647E7	0.1464296101031863	1.1249195656049034E7	3.066146162778972
750	1.7312544696335156E7	0.14845280205407735	1.1239453913174197E7	3.063490896709491
1250	1.6373189355043197E7	0.13278014939385577	1.0800032208539585E7	2.9437195624112285
1750	1.6284466079552796E7	0.13134502629921352	1.0606337190140132E7	2.890924922192172
2000	1.6225879406322487E7	0.13040164555665135	1.0578142617484624E7	2.8832400455661356
2250	1.6319345076829238E7	0.131908273373001	1.0563390971984742E7	2.8792192513131867
2750	1.6046515952418283E7	0.12753461881567382	1.0394181685214002E7	2.8330985844493575
3250	1.6066987240427492E7	0.12786023009215708	1.042667647828407E7	2.8419555541499877
3500	1.607488568496454 1E7	0.1279859717947816	1.0391322473320115E7	2.83231926103415
3750	1.6009576173056157E7	0.12694811416700733	1.0298206092207342E7	2.8069389188861 08
4250	1.5992256003949927E7	0.12667358179947882	1.0287790551091507E7	2.80409999845108
4750	1.5945547260696666E7	0.12593470878094468	1.0228924317390647E7	2.788055075587454
5000	1.5875174067377836E7	0.12482557517907494	1.0258513428235253E7	2.796120055649157
5250	1.588583670148897E7	0.12499331084056568	1.0274964604297716E7	2.8006040838320962
5750	1.5927109490963502E7	0.12564364160207078	1.020898009187964E7	2.782618961540732
6250	1.5868728355449121E7	0.12472423123740822	1.0122720490904504E7	2.759107543247413
6500	1.6028187165866222E7	0.1272434378761562	1.0307276000099799E7	2.809411065707224
6750	1.577586143919882E7	0.12326868140382842	1.014860985722054E7	2.7661640994327437
7250	1.5829140198235258E7	0.12410270147432297	1.0148089482175702E7	2.766022262985435
7750	1.5842568056571256E7	0.12431334340323788	1.0140932614519117E7	2.7640715455320537
8000	1.5858615811149577E7	0.12456531775645176	1.0185453914995473E7	2.7762065300048415
8250	1.5956228024400692E7	0.12610347430901775	1.0239332682261683E7	2.790892040023482
8750	1.5848159326230377E7	0.12440110595247707	1.0165079511565648E7	2.7706531641638206
9250	1.58331346337501E7	0.12416534325761594	1.0170811683007305E7	2.7722155581346977
9500	1.5857643112027826E7	0.12455003762598278	1.019134321752758E7	2.7778117525391792
9750	1.5850000527798546E7	0.12443001288251795	1.0178245133511957E7	2.7742416626170026
10250	1.5858817608999379E7	0.12456848791625819	1.0179586735888552E7	2.7746073375598823
10750	1.5907224987751033E7	0.12533011304460792	1.0234222683864623E7	2.789499229154509
11000	1.5812322406315053E7	0.12383913382143431	1.0140685359426498E7	2.7640041522466454
11250	1.5862289428649314E7	0.12462303506776225	1.0137863932123292E7	2.7632351276191263
11750	1.583132071785077E7	0.12413689499464567	1.015314458290807E7	2.767400110627784
12250	1.5883572837600864E7	0.12495768820571833	1.0130897483274328E7	2.761336311822909
12500	1.5879342195045091E7	0.12489113127740264	1.0159534861691276E7	2.7691418821614384
12750	1.5861088518961716E7	0.12460416574306853	1.0117397407133767E7	2.7575666525902644
13250	1.5871249705877127E7	0.1247638687521227	1.013921938511985E7	2.763604577767501
13750	1.5856951545645382E7	0.12453917437976886	1.0172314730165765E7	2.7726252374058467
14000	1.5816759191442387E7	0.12390863970397985	1.0133555849472402E7	2.762060891567693
14250	1.5854136650377382E7	0.12449496239919393	1.0136860827932175E7	2.762961715709466
14750	1.5840838909170704E7	0.1242862083626219	1.0122085317940442E7	2.7589344168118872
15250	1.5894838024161596E7	0.12513499980547937	1.0171886366424413E7	2.772508480094304
15500	1.5899116449063402E7	0.12520237422638808	1.0185389261582607E7	2.7761889076947823
15750	1.5893312480890926E7	0.1251109807251198	1.0161951013281029E7	2.7698004424845166
16250	1.5855084169284044E7	0.12450984367065324	1.0124691185209323E7	2.7596446871433393
16750	1.5848963628128573E7	0.12441373310778797	1.0116040720095098E7	2.757286866084198
17000	1.589022234193422E7	0.12506233476392495	1.0161705923027275E7	2.7697336392601737
17250	1.592525817628335E7	0.12561443449457585	1.0194098913751937E7	2.7785628611217343
17750	1.586972509542592E7	0.12473989998273662	1.0172132539788276E7	2.772575578537434
18250	1.5877401484822113E7	0.12486060574541973	1.0191597547534073E7	2.7778810742042417
18500	1.585659172539631E7	0.12453352244746778	1.0176400492332129E7	2.7737388764936055
18750	1.5851815912624296E7	0.12445851777662895	1.016343011099782E7	2.77020359395662
19250	1.5864320686174681E7	0.12465495450626025	1.0160791342330502E7	2.7694843558287445
19750	1.587597495178339E7	0.12483817011225283	1.0197950208458021E7	2.7796125923954933

REPTREE für Gesichtserkennung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.234189047812657E8	51.808066943817145	2.532436712037094 5E8	69.02556720060028
100	2.0611991310412493E8	21.04294332280026	1.6537837390929198E8	45.07649098412938
150	1.4250990676353917E8	10.05904372475152	1.165978786050228E8	31.780595609137368
200	1.386285267023069E8	9.518572004007895	1.1260301453596103E8	30.691732235195534
250	1.238672262495914 4E8	7.599401831863172	1.0338383723678155E8	28.178899676837645
300	1.3287779491653675E8	8.745233484657582	1.0667521095448853E8	29.076015631025186
350	1.3082255817032182E8	8.476798478025318	1.0539116636850935E8	28.726028974155188
400	1.3323977681470734E8	8.792945415148404	1.0631812576306775E8	28.978686415414
450	1.3698481093063164E8	9.294187151012729	1.0659262589072181E8	29.053505765954522
500	1.5740423264340582E8	12.271549315523691	1.1005843811575174E8	29.998167693758553
750	2.0725431543530273E8	21.275204750759794	1.237373876357448E8	33.726581694542745
1250	2.6162517215308458E8	33.902024882612494	1.4276884994219947E8	38.9139076960805
1750	3.1595920718757105E8	49.44570971492161	1.581206915654998E8	43.09829489353859
2000	2.9405484617552394E8	42.82755351661145	1.5102592570883393E8	41.164504267758716
2250	4.344287691264875E8	93.47677909147181	1.9069834527265 53E8	51.97784957110563
2750	3.71469413158201E8	68.3459372427008	1.7228537444026512E8	46.95910320120533
3250	4.4370609246892595E8	97.51184373019615	1.9356206372786552E8	52.758401320865666
3500	4.48376098551134E8	99.57526973584838	1.949433527071448E8	53.1348934748828
3750	4.4636041931212795E8	98.68199886967012	1.941139717137551E8	52.908832569897456
4250	4.62882745103149E8	106.12276766839115	1.994981958141043E8	54.37638799071107
4750	4.5686592010410875E8	103.38180493780665	1.9780807595055765E8	53.91571909555647
5000	4.6328955830816466E8	106.30938561871312	1.996909793213669E8	54.4289341841558
5250	4.680815945279998E8	108.51998201900868	2.0074051381431246E8	54.71500140679485
5750	4.4929657934713745E8	99.98452975782989	1.948343493730389E8	53.10518289247158
6250	4.51506826940487E8	100.97066738512707	1.9498338620258576E8	53.145805237126254
6500	4.436858948204516E8	97.50296638932207	1.922349838311331E8	52.396684709520294
6750	4.449710055653862E8	98.06860787768878	1.926868590855679E8	52.519850455748106
7250	4.4435587089523834E8	97.79765223055666	1.9206877584605768E8	52.351382094893964
7750	4.3213406075056094E8	92.4918745466334	1.8789653556328982E8	51.214172029006136
8000	4.354845167215 99E8	93.931665172961	1.890645131751831E8	51.53252279668903
8250	4.4052589798491216E8	96.11905101328814	1.903582171570645E8	51.88514227465896
8750	4.29985028619573E8	91.5742252112312	1.8688563532990938E8	50.938635184746076
9250	4.314718170616359E8	92.20860492541459	1.8712024991628835E8	51.00258310029068
9500	4.2517717865278846E8	89.5378116041012	1.8500259635908595E8	50.42538313621859
9750	4.2798496088866585E8	90.72429477013846	1.8578429049146816E8	50.63844623315048
10250	4.18619454377834E8	86.79713669025146	1.8259914785234514E8	49.77028523929318
10750	4.197299376106119E8	87.25824573732103	1.8266513219603512E8	49.788270315595035
11000	4.156506378413593E8	85.57038512927025	1.8121325107208297E8	49.3925371562537
11250	4.1356342818646526E8	84.71315126418027	1.805092671104533E8	49.2006551952134
11750	4.007881681676035E8	79.56029196155907	1.7636103727723 77E8	48.0699895570351
12250	4.013662131881808E8	79.78995240995982	1.7645062412410527E8	48.09440786881108
12500	3.9571532513932294E8	77.55902192604678	1.7457748634692642E8	47.583854547 18182
12750	3.892193169648862E8	75.03352616823973	1.7240931879470268E8	46.99288619497453
13250	3.7691176820355195E8	70.36326429748426	1.681593512606223E8	45.8344903376184
13750	3.756430339598142E8	69.89035769519018	1.675563260433814E8	45.67012627885053
14000	3.6861069488127846E8	67.29804488343805	1.6521998204338852E8	45.033318776381876
14250	3.656131148485747E8	66.20794614975726	1.6408753588392815E8	44.7246526679285
14750	3.586869124475895E8	63.72321008164735	1.6161361752361706E8	44.050347098052725
15250	3.488766537702873E8	60.285157573281346	1.581164979425235E8	43.09715185528145
15500	3.391487706526896E8	56.970112674916194	1.5468526799532026E8	42.16191587415846
15750	3.342628207114744E8	55.34045575628763	1.5308022008530587E8	41.72443468520601
16250	3.213440187029776E8	51.145452502893654	1.485595773137013E8	40.492262011598335
16750	3.122882597740964E8	48.303422032943175	1.453709928938623E8	39.62316290598104
17000	3.113362498471267E8	48.00936522641345	1.4500317292221704E8	39.52290775626754
17250	3.107321150246126E8	47.82322571512119	1.447141443757765E8	39.44412845544476
17750	2.9997946563989323E8	44.57071883193565	1.4094569768039283E8	38.41697871710107
18250	2.9174132105613536E8	42.156298832693906	1.380230367342206E8	37.62036125935499
18500	2.8822813617472464E8	41.147109358394516	1.3676935376597878E8	37.27865014151619
18750	2.858970893685614E8	40.48424570351855	1.3586957335072005E8	37.033400760854406
19250	2.7290020216823053E8	36.88708180433895	1.3127969218397252E8	35.78235606776412
19750	2.663981075169924E8	35.15028532378993	1.289439733604351E8	35.14571896702449

SVR RBF für Gesichtserkennung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	8.580051492802E10	82.12869126227596	4.382591331930331E10	69.88492699630936
100	7.3831714774560 93E10	60.81364355117642	2.924348791337339E10	46.63174964669607
150	5.5615544200742966E10	34.507023223481866	2.088441661102977E10	33.30235058853823
200	4.9263627064431885E10	27.074963607648527	1.745479202648384E10	27.8334613957562
250	3.154654494692622E10	11.102447193700177	1.0622027971413252E10	16.937916248924203
300	3.007751893143937E10	10.092508676378955	9.6459707266491E9	15.381492568769714
350	2.5603224017763763E10	7.313149069321333	8.69694276879321E9	13.868169866992652
400	2.1812984797025 79E10	5.308175915344798	7.425343726553219E9	11.840474389477587
450	2.2343272244568233E10	5.569403306752921	7.479134510275471E9	11.926249327380548
500	2.1591371295046947E10	5.200864814963554	6.823886232934756E9	10.88138854620787
600	1.585683145496048E10	2.805097323796005	5.0369635402002735E9	8.031956498552303
700	1.5663387467061762E10	2.737073729526964	4.844300536059656E9	7.724735519923516
800	1.3992192422173819E10	2.18417107657091	4.1189378674976134E9	6.568070129541816
900	1.1106806002497704E10	1.376237813732436	3.776310325033565E9	6.021715268262097
1000	9.898752024470297E9	1.0931405466059032	3.502326366938949E9	5.584819663369624 5
1100	1.0724665352087482E10	1.2831653182097338	3.7486440809881067E9	5.9775985962080025
1200	9.530324306498333E9	1.0132823378925166	3.5838803374068265E9	5.7148658584342344
1250	9.368566492708216E9	0.9791774443635407	3.4927710258207107E9	5.569582689034196
1300	9.175596141471895E9	0.9392553856920977	3.44422112448812E9	5.492164877211626
1400	8.862894892248016E9	0.8763272399496232	3.297203010382158E9	5.257729371063147
1500	7.9162097697318E9	0.6991167407791342	3.249156094871434E9	5.181113621873823
1600	7.7912279643617115E9	0.6772155738989369	3.172452363590551E9	5.058801632118939
1700	7.7232417146170 65E9	0.6654483739231838	3.091033059884655E9	4.928970177058979
1800	6.788318523003172E9	0.5140904522493767	2.9242642043216825E9	4.663040081971991
1900	6.678898323570984E9	0.49765088167662563	2.8488551356982756E9	4.542792565001585
2000	6.484137132897495E9	0.46905038429444473	2.854381580149522E9	4.55160504916388
2100	6.61044373900396E9	0.48750193120431184	2.8636755535647054E9	4.566425245810865
2200	6.437651113294878E9	0.4623490683391999	2.7901788327343154E9	4.449227164112527
2300	6.475653448922147E9	0.4678237994906606	2.9227167216949344E9	4.660572461739859
2400	6.525284598003343E9	0.4750223346939507	2.9117446613092346E9	4.643076382799824

KNN Regression für Rauschreduzierung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.065838680354693E11	126.7354374295326	3.6155982373039536E10	57.65443312522582
100	1.2958306030116872E11	187.3318311424212	4.074672426347249E10	64.97484330206287
150	1.2790181976656226E11	182.50239200014605	4.184759175598094E10	66.73028976101156
200	2.0015083917461612E11	446.9201588234524	5.277101344758757E10	84.14880930003498
250	2.3162642646849084E11	598.537448096794	5.787109387385563E10	92.28141216602165
300	2.975430579694585E11	987.6765850597479	6.878363012769434E10	109.68257375477964
350	2.5223591785714508E11	709.7885223441123	6.149808036122794E10	98.06501521474516
400	2.309434860765051E11	595.01312842231	5.789924302550092E10	92.32629888391858
450	2.5737732125411 72E11	739.0191065869477	6.215194117474038E10	99.10766354212039
500	2.2426411945314 67E11	561.0928111745018	5.6520076309194786E10	90.12707568501855
600	2.1221966043025 41E11	502.442472833778	5.5034740171077 32E10	87.75855442178685
700	2.6660013199584 25E11	792.9317976424164	6.4101224171324 99E10	102.21599579566552
800	2.6757416993846744E11	798.7364201251938	6.4216592673931305E10	102.39996274059999
900	2.037009237016169E11	462.91484746806	5.388045836164519E10	85.91793333997127
1000	2.2778134565953 11E11	578.8305203890728	5.7804747422951 81E10	92.17561592524291
1100	2.3922473646879855E11	638.4505765120783	5.9583569338985016E10	95.01213045115152
1200	2.4683642976396 73E11	679.7255990358317	6.0795025865827 31E10	96.9439225012281
1250	2.4982452442067 35E11	696.2821352458866	6.1254535685677 53E10	97.67665817704518
1300	2.3197419131043 16E11	600.3360903591704	5.8351601787533 13E10	93.04763146244211
1400	2.5178371531771246E11	707.2458197588082	6.1581836622255936E10	98.1985731886493
1500	2.3576755532155 78E11	620.1306464343082	5.8998958263575 92E10	94.07990795464501
1600	2.3954910053543 05E11	640.1830965267526	5.9509148135751 58E10	94.89345818716346
1700	2.3483904970332358E11	615.2558367190456	5.8642973117613 38E10	93.51225302053997
1800	2.1562698261473 09E11	518.7060638122483	5.5418069970448 84E10	88.3698132185953
1900	2.2130408643341 86E11	546.3789767996739	5.6421178999677 34E10	89.96937375179526
2000	2.0138478119557486E11	452.44772493503075	5.3038372630245 64E10	84.57514101902788
2100	2.0395792563541 56E11	464.0836694729392	5.3500904725840 16E10	85.31269602440949
2200	1.9713190367043872E11	433.53977260388587	5.2354808668649 08E10	83.48512796658014
2300	1.8768552355419083E11	392.98562637239155	5.0750009235469 78E10	80.92611019062755
2400	1.9700867067272678E11	432.99790489403546	5.2350436509842476E10	83.4781561096153

Linear Regression für Rauschreduzierung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.8042211058769005E10	25.749042701574993	3.589096360777824E10	57.23183344252211
100	6.066133590957739E10	41.05244750222176	4.070295727860954E10	64.90505234255197
150	5.5260948450656815E10	34.06840353769092	3.8843513883045334E10	61.93997857429946
200	4.876593800242805E10	26.530673694870792	3.560802225022731E10	56.780654342782185
250	4.746829709768993E10	25.137519331073428	3.4136549529795715E10	54.43423972512882
300	4.794406889093831E10	25.6439482178665	3.4432558406019806E10	54.90625632760013
350	4.811261752048699E10	25.8245690923194	3.4559929204129074E10	55.10936216734558
400	4.76513828462058E10	25.33180467211676	3.4300371468521637E10	54.69547065818075
450	4.77614302404946E10	25.448943689874522	3.427683690450526E10	54.65794237494358
500	4.790681283161814E10	25.60410924411725	3.436521021849137E10	54.798862714728514
600	4.826759866838739E10	25.991210100020258	3.470528026183556E10	55.341139380581204
700	4.884633468850578E10	26.618224004421755	3.4834903292291336E10	55.547836636481904
800	4.820988078100798E10	25.92908723323845	3.462326414529006E10	55.21035624605658
900	4.805954320020959E10	25.767624966767073	3.499774206202281E10	55.80749980543727
1000	4.8360905398441864E10	26.091795127695615	3.5331272628990364E10	56.33934860351578
1100	4.828027273673061E10	26.004861395336544	3.521368975557055E10	56.15185061653508
1200	4.8212826078087875E10	25.93225551312155	3.518076394723707E10	56.09934702819193
1250	4.794477130438155E10	25.64469962617754	3.502188216626213E10	55.84599368484958
1300	4.825832139148679E10	25.981219775796415	3.521451707363109E10	56.15316986028574
1400	4.798602444068314E10	25.68884956767817	3.5016542030290344E10	55.83747828872264
1500	4.825174363817927E10	25.974137622815174	3.520728004980404E10	56.141629681347794
1600	4.788953573594288E10	25.585644860793238	3.490699559489415E10	55.66279522884909
1700	4.76276321651942E10	25.306558912689564	3.467528427340098E10	55.2933076914465
1800	4.762705484527539E10	25.30594540782589	3.47717799724353E10	55.44718000971035
1900	4.766120051557934E10	25.3422440292959	3.4814536425008884E10	55.515359571536095
2000	4.823942160882256E10	25.960873305460836	3.5400872759058846E10	56.450333170416464
2100	4.8367622947813E10	26.099044168779944	3.548251489199046E10	56.580519949598454
2200	4.851390113599806E10	26.257145544346756	3.559668393510542E10	56.76257423299004
2300	4.8505068981578125E10	26.2475859728087	3.560224392874367E10	56.771440214782075
2400	4.836931695953962E10	26.100872369481802	3.545373071160023E10	56.53462061304465

RBFNETWORK für Rauschreduzierung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.081667924348062E10	18.586193357100104	1.6947097835502495E10	27.02389078362353
100	7.714047346860205E10	66.38649105550645	2.974552020023013E10	47.43229177028304
150	1.9915977639813038E10	4.425051951508114	7.843778160745767E9	12.507711137591363
200	2.2749968103436733E10	5.773998915605333	8.274451454621164E9	13.194646006435414
250	1.627244589694207E10	2.9540700280988346	6.710281124527 1E9	10.700233514716945
300	1.5155134760884497E10	2.5623279788916613	4.268015122616597E9	6.805789147851734
350	1.038815555296859E10	1.2039044011560502	4.203367545318612E9	6.702701935794296
400	1.3848289718700043E10	2.1394758833684144	4.269659924912646E9	6.808411954307478
450	1.3840129006438799E10	2.1369550662860046	4.424130052728006E9	7.054730462876522
500	7.008586996650874E9	0.5479943122815147	3.221441225004235E9	5.136919410945004
600	1.3810267805143791E10	2.1277437062574487	4.25893174900069E9	6.791304774248315
700	5.585863373722633E9	0.34809335412281867	2.7091088413895955E9	4.319952723544924
800	6.836425093635795E9	0.5214026487049813	2.9999975014682436E9	4.783804614674748
900	3.927739534194207E9	0.17210776860128965	2.2334314638725305E9	3.561436213931556
1000	4.36243086437604E9	0.21231086927387235	2.4382645586901917E9	3.8880636540368463
1100	4.1464323941829457E9	0.19180692874729574	2.3775627677641187E9	3.791268404237788
1200	4.0981477618987994E9	0.18736580794171354	2.3421595978663874E9	3.7348144080434595
1250	4.913139518834183E9	0.26929811147228977	2.7026869324580946E9	4.309712329155762
1300	4.384537136755599E9	0.21446805813142938	2.4387994313236346E9	3.8889165634712293
1400	3.8561887746137853E9	0.1658943846729406	2.2657875353840218E9	3.613031298305348
1500	3.816517198084074E9	0.16249857665745432	2.2493396537434807E9	3.5868034590971036
1600	3.8750400200694895E9	0.16752032157494443	2.268939337808719E9	3.6180571714857197
1700	3.902129117097988E9	0.16987066425271752	2.276722920160541E9	3.6304688942138625
1800	4.387468535898784E9	0.21475493066846268	2.448439856112207E9	3.904289212471245
1900	4.059956592154586E9	0.18388990748552037	2.339543330032967E9	3.730642499857331
2000	3.846000812570061E9	0.16501896432587546	2.292812604685017E9	3.6561255512741364
2100	3.840861902275167E9	0.16457827221574914	2.302013763191655E9	3.6707977450019205
2200	3.8517860094209037E9	0.16551578439938475	2.30486663865576E9	3.6753469483941217
2300	4.1591492451141357E9	0.19298525280772616	2.489808762200811E9	3.9702561886950103
2400	3.82300644967647E9	0.1630516414411534	2.299436612181288E9	3.6666882126139515

REPTREE für Rauschreduzierung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	8.653095645596494E10	83.53300830398314	4.958433000760976E10	79.06731475271866
100	7.760348297978346E10	67.18580745382913	4.518728898548985E10	72.05578053569457
150	5.885632137377478E10	38.64571463331589	3.800422110961676E10	60.60163991214199
200	4.160630279932335E10	19.312271782786354	3.073080193704277 4E10	49.00342485189803
250	3.504725741495676E10	13.703237681002477	2.6413615861989094E10	42.11922756235239
300	3.86748437741038E10	16.68676884600609	2.6144006996630146E10	41.68930849287646
350	3.976130379479232E10	17.6374724025787	2.633506597044779E10	41.99397168015463
400	4.140786350108619E10	19.128493153919266	2.643561069879632E10	42.15430059216857
450	4.7663431908544785E10	25.34461702150686	2.7234472635561382E10	43.4281681262963
500	6.19875144901228E10	42.867046241894066	2.8794138740913105E10	45.915216168330254
600	6.3427641904202415E10	44.88200450823931	2.9165748733348446E10	46.50778652740602
700	6.7640366046557045E10	51.041921136131386	2.981649437251918E10	47.54546738884702
800	7.405450297675618E10	61.181209273916316	3.0712317415378193E10	48.973949380669865
900	7.204328833636658E10	57.90314710384057	3.0410524479020138E10	48.492709499330196
1000	8.040174456553076E10	72.11839588529021	3.1703353702525066E10	50.55425868474198
1100	1.1163052923479538E11	139.02121546271385	3.653637062612424E10	58.261001323894014
1200	1.1393252847937857E11	144.81401185226218	3.684146139598842E10	58.74749993999507
1250	1.1686330050826903E11	152.36015618671803	3.734652881818362E10	59.55288189909229
1300	1.1475517581802393E11	146.9128146075216	3.7103439353246475E10	59.165250741541
1400	1.6647247808500906E11	309.1718912761939	4.6003146029483284E10	73.35674851652139
1500	1.566713612510209E11	273.83839928719937	4.446490320428982E10	70.90386209843291
1600	1.8327388175932117E11	374.72810814287794	4.886843953489763E10	77.92575375298921
1700	1.7905113628377383E11	357.6590977992142	4.8081480504315796E10	76.6708666271783
1800	1.66073371598008E11	307.6912309744268	4.583784179644281E10	73.0931539127068
1900	1.6920010095883047E11	319.3863487095083	4.64089359856763E10	74.0038223437745
2000	1.3458882109764384E11	202.08454081365525	4.037497837617938E10	64.38205624465739
2100	1.5608479824067178E11	271.7917866213624	4.408416371729295E10	70.29673382116167
2200	1.4204400003733905E11	225.0924360621465	4.167624330422106E10	66.45705702870256
2300	1.1540209026624225E11	148.57387971221664	3.707799915555839E10	59.124683729388835
2400	1.5986745094388733E11	285.1249448908354	4.4758456930312256E10	71.37196371135553

SVR RBF für Rauschreduzierung(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.123076136559996E12	100.4589973418883	7.69783658034776E10	55.863714114133856
100	1.123039231004499E12	100.45239505720853	7.683533403285323E10	55.759915003045975
150	1.1230784648018596E12	100.45941386411332	7.680779197399803E10	55.739927547012826
200	1.1230582068852417E12	100.45578975388605	7.676452766349237E10	55.70853034796597
250	1.1230675466883738E12	100.45746062222649	7.678559177130232E10	55.723816711666764
300	1.1230720794431448E12	100.45827152617287	7.682464853333835E10	55.75216046469179
350	1.123058609563635E12	100.45586179179654	7.681948680542749E10	55.7484145642776
400	1.12304984382672E12	100.4542936346015	7.68221315323013E10	55.75033386023104
450	1.1230261624063992E12	100.45005717923475	7.679528286258937E10	55.7308496013295
500	1.1230005081205698E12	100.44546789159274	7.676848527346777E10	55.71140241195348
1250	1.1228707917555903E12	100.42226457445548	7.661323217040224E10	55.59873419830074
2000	1.122702148159842 8E12	100.39210206574778	7.646572123280853E10	55.49168452583197
2750	1.1226711466999556E12	100.38655783701375	7.649923712400015E10	55.51600723188408
3500	1.1225685430540 26E12	100.36790755240419	7.635635371400237E10	55.41231578709365
4250	1.122356414124541E12	100.33028046446968	7.61450723641649E10	55.25898750060987
5000	1.122030204521326E12	100.27196753585193	7.593437188530447E10	55.106080756073105
5750	1.121906934428166E12	100.2499363004284	7.579259057665115E10	55.00318911358802
6500	1.1218247113512673E12	100.2352424681425	7.558809057192053E10	54.854782094533135
7250	1.1214662803701665E12	100.17120095363028	7.537073697358379E10	54.697047136763835
8000	1.1207638612397056E12	100.04575780159965	7.516059762567123E10	54.544547608703084
8750	1.1193476250197332E12	99.7930749492393	7.479318999897505E10	54.27791744051393
9500	1.1196358262938638E12	99.8444695173345	7.495929317670424E10	54.39845962580534
10250	1.118750712317524E12	99.68667034373772	7.497151490001949E10	54.40732901203014
11000	1.118435960709563E12	99.63058613182582	7.484815294238403E10	54.317804415580525
11750	1.1187589984540393E12	99.68814702734406	7.464332230812366E10	54.16915745646801
12500	1.1173136850626064E12	99.43074119476776	7.45495698258775E10	54.10112065403626
13250	1.116559018359565E12	99.29646962982876	7.424124609250026E10	53.87736806178247
14000	1.1172510778424805E12	99.41959856226033	7.432396996884277E10	53.93740133126248
14750	1.1166002452457834E12	99.3038024438832	7.418685128734889E10	53.83789338844337
15500	1.1164238091453137E12	99.27242256286615	7.40722374960962E10	53.754717394756746
16250	1.1158532167234275E12	99.17097431990666	7.390181311300652E10	53.63103928187882
17000	1.1161946593956147E12	99.23167474645741	7.377702582957704E10	53.540480316982375
18500	1.1127801321599634E12	98.62548836033888	7.320364498878079E10	53.12437401186104
19250	1.112789194039429E12	98.6270946721357	7.33085597756639E10	53.20051137330809

KNN Regression für Schach (alle Parameter)(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1164079573833 12E12	99.26960350502976	7.641647979556674E10	55.455949685998306
100	3.016299469537279E12	724.6350823160808	2.9396227409980615E11	213.3303853523784
150	3.731920639765279E12	1109.264821234183	2.7320653766279004E11	198.26780881618757
200	1.1231071297151663E12	100.46454208485484	7.698889694395723E10	55.871356633105385
250	9.407677982551072E12	7049.126876236403	4.8908936988720905E11	354.9354217962184
300	1.123121929575167E12	100.46718986602406	7.694788487255016E10	55.8415938990125
350	1.1231231674885427E12	100.46741133751955	7.694562851782274E10	55.8399564473192
400	1.1231225966978464E12	100.46730921897951	7.694663372606476E10	55.84068593365307
450	1.1231195970565671E12	100.4667725625425	7.695270347471194E10	55.84509079077925
500	1.1231202775411245E12	100.46689430578009	7.695123969670581E10	55.8440285173053
1250	1.0906655502116737E12	94.74441793367016	8.720027897391132E10	63.28182476239618
2000	4.3346601708508414E13	149651.3197465837	6.633981848712029E11	481.43249283951377
2750	1.05686124486465E12	88.96237851874872	6.6104223158546036E10	47.97227618073303
3500	1.0682297147787456E12	90.88657750339185	6.717883992639689E10	48.75213274529994
4250	1.0479254328252817E12	87.46437597161344	6.205483241497 39E10	45.03360627686877
5000	1.0503631794314515E12	87.87177897351907	6.370871276357849E10	46.23383829022825
5750	1.05636203919609E12	88.87835606970341	6.3443536189929375E10	46.04139787991974
6500	1.0700191222974763E12	91.19132345357556	6.403002792738576E10	46.46701885030003
7250	1.0629433729723201E12	89.9892635846748	6.3645131446136475E10	46.18769690357361
8000	1.0570312209505352E12	88.99099664239836	6.279068525928691E10	45.56761959990708
8750	1.057232960801089E12	89.02496866370016	6.256481138793646E10	45.403701423114555
9500	1.0677644477878921E12	90.80742352574012	6.362455463074822E10	46.17276417123731
10250	1.0612963212494883E12	89.71059937202006	6.358007682145318E10	46.140486328015434
11000	1.0625233684799034E12	89.91816209662885	6.3691118460051605E10	46.22106998666654
11750	1.0662577742488657E12	90.55133593216432	6.396420710877938E10	46.41925224269534
12500	1.0563482816831604E12	88.87604107328131	6.4242711457815056E10	46.62136470860286
13250	1.0518964439761935E12	88.12850735505782	6.3919646356810 89E10	46.38691420742146
14000	1.0539944942465437E12	88.48040973684367	6.387395744596945E10	46.353757459719795
14750	1.058123427536933E12	89.17499644532067	6.3964609937146484E10	46.41954457793138
15500	1.0575870434813214E12	89.08461016051744	6.398809363563694E10	46.43658685474669
16250	1.058591006794272E12	89.2538257959097	6.370865724787197E10	46.2337980021093
17000	1.0534882651833893E12	88.39543662052358	6.357143993260307E10	46.134218480101
18500	1.0521678200727736E12	88.17398532101292	6.332973788473236E10	45.95881368990941
19250	1.0507919094996141E12	87.94352741850754	6.347442516883787E10	46.06381421818583

Linear Regression für Schach (alle Parameter)(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1230660118122842E12	100.45718603562761	7.727709146146898E10	56.080501318985554
100	1.1230608423228018E12	100.45626122592634	7.70940978346486E10	55.94770162199666
150	1.1231063663511765E12	100.46440551553646	7.69915892960198E10	55.87331049097634
200	1.1231091145812405E12	100.46489718693228	7.697213568692755E10	55.859192876949656
250	1.123117735976192E12	100.46643960319544	7.694764411125322E10	55.84141917693155
300	1.1231190753036929E12	100.4666792175144	7.693095869213268E10	55.82931045685292
350	1.12312257919308E12	100.46730608725234	7.69359772301303E10	55.83295244339087
400	1.123121435391908E12	100.46710145319713	7.693458289986742E10	55.83194056860796
450	1.1231163828823032E12	100.4661975262712	7.693171116685085E10	55.82985653278081
500	1.1231140039026777E12	100.46577191270616	7.692370761807643E10	55.82404830398866
1250	1.1230916200992422E12	100.46176736165759	7.690944922581E10	55.813700893505406
2000	1.1230756799667249E12	100.4589156575013	7.68986836585696E10	55.80588824946812
2750	1.123082018470989E12	100.46004961681234	7.68983833330239E10	55.80567030121407
3500	1.1230179869507153E12	100.4485946629921	7.68429223162774E10	55.765421871000385
4250	1.1230884423715237E12	100.46119886005818	7.689877904259576E10	55.80595747029287
5000	1.1230885954345356E12	100.46122624328846	7.689764618402449E10	55.80513534726263
5750	1.1230922921612407E12	100.4618875950475	7.689485290419463E10	55.803108245436974
6500	1.1230924830296074E12	100.46192174183699	7.689481316257419E10	55.80307940467763
7250	1.123098145342793E12	100.46293474531925	7.689958083882153E10	55.8065393391682
8000	1.123097821335831E12	100.46287677944028	7.690010360685307E10	55.80691871541965
8750	1.1230976726169417E12	100.46285017316224	7.690227467999382E10	55.808494277696006
9500	1.1230987352036008E12	100.46304027334216	7.69056469157421E10	55.81094153169784
10250	1.1230962394425605E12	100.4625937739286	7.690235239508691E10	55.80855067605803
11000	1.1230989376546355E12	100.46307649250554	7.690449457397295E10	55.81010526958079
11750	1.1230967454739705E12	100.46268430444174	7.690237594110751E10	55.80856776356242
12500	1.1230958756713054E12	100.46252869420236	7.69022143942804E10	55.808450527951194
13250	1.1230754741017566E12	100.4588788283366	7.688867558258072E10	55.79862532189048
14000	1.1230943578775469E12	100.4622571567508	7.690169066134062E10	55.80807045145939
14750	1.123093476330874E12	100.46209944581015	7.690182012732573E10	55.80816440578934
15500	1.123091416510742E12	100.46173093923517	7.690267903604356E10	55.808787721581545
16250	1.1230721886093914E12	100.45829105591304	7.689015915859042E10	55.79970196290783
17000	1.123092441940629E12	100.4619143909238	7.690274643517154E10	55.80883663357879
18500	1.123072743489382E12	100.45839032345782	7.689009758135883E10	55.79965727589966
19250	1.123101053876206E12	100.46345509197292	7.690860175104355E10	55.81308587540808

RBFNETWORK für Schach (alle Parameter)(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.123052266133298E12	100.45472697486689	7.725082440342693E10	56.06143913955677
100	1.1188354388421782E12	99.7017700882379	9.151026854671191E10	66.40961297687336
150	1.1216610734547356E12	100.20600244911134	8.004801507873424E10	58.09137909186774
200	1.1217857367451606E12	100.22827781280677	7.699303181085443E10	55.87435733362474
250	1.122484624113301E12	100.35320380446589	7.667309770814699E10	55.64217901881562
300	1.1312289022163423E12	101.92281870658857	9.690178780021516E10	70.32227450293745
350	1.1213330854751355E12	100.14740799523739	8.052946448611067E10	58.44077014185132
400	1.122240376091931E12	100.30953567006674	7.663294564142015E10	55.61304039586218
450	1.1563146472289294E12	106.49335134768603	1.027780595765263E11	74.58672417191143
500	1.1002287152128945E12	96.41317650071713	7.896397849135231E10	57.30468650136294
1250	1.1133351546451301E12	98.72389596714697	7.886558001062408E10	57.233278066807436
2000	1.0660638168258923E12	90.51839548046162	8.162972898685008E10	59.23923943744735
2750	1.0537487077185679E12	88.43914812278605	8.245381995101558E10	59.8372877533042
3500	9.700520488637803E11	74.94808059939996	7.173155455074327E10	52.05606815056718
4250	9.859154787283594E11	77.41940168080046	7.191114080965535E10	52.186395097911095
5000	1.0740897507615111E12	91.88647379398624	6.668568571871089E10	48.39424744356873
5750	1.0997392592226377E12	96.32741340532506	6.965646798458508E10	50.55016397237399
6500	1.0512118431582688E12	88.01383215962171	6.438370593866542E10	46.72368534489701
7250	1.0604502626711371E12	89.56762296291494	6.238712708790557E10	45.27475474003172
8000	1.0790485450261858E12	92.73686440427362	6.865709176322309E10	49.824909974839684
8750	9.798287553214989E11	76.46642775521867	6.2350082374308876E10	45.24787114399528
9500	9.64218929389296E11	74.04943463551136	6.006353105505832E10	43.588505582351495
10250	1.8128274982785012E12	261.748017579137	7.959714508013399E10	57.76417972303799
11000	1.041428325512152E12	86.38318487843271	6.606224558064576E10	47.94181277213052
11750	9.340973034664994E11	69.49517891152044	6.1986009114530136E10	44.98366074814454
12500	9.546217528338179E11	72.58269562629923	6.4970716689479904E10	47.149682966737174
13250	1.0525926421289542E12	88.24520174691276	6.492178829776248E10	47.11417530613101
14000	1.7914102986180806E12	255.5998377370499	7.854381924640222E10	56.99977413153913
14750	1.008913695644662E12	81.0734166116098	6.506556103465703E10	47.21851214138922
15500	1.765748752229116E12	248.32946714652633	7.824506000114441E10	56.78296255218389
16250	1.157303001860708E12	106.67547855614741	7.421874170090631E10	53.861036474521626
17000	1.7762460204508774E12	251.2908506361034	8.021734483758992E10	58.2142628286293
18500	1.021474880040697E12	83.10474525630067	6.186937480665753E10	44.899018452054115
19250	1.0162262625525961E12	82.25290952583704	6.5154694341336685E10	47.28319677726616

REPTREE für Schach (alle Parameter)(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.121844114545438E12	100.23870985600112	7.570950529241911E10	54.94289356798916
100	1.115032995091975E12	99.02523421476306	7.53180217785386E10	54.658791367694995
150	1.086229930850277E12	93.97535427637665	7.325274846726772E10	53.16000873410503
200	1.1084333123746555E12	97.85647768399083	7.26928207854555E10	52.75366547629668
250	1.1086958385504912E12	97.90283668246673	7.231404200223459E10	52.47878318388257
300	1.0911367036763436E12	94.82629234976419	6.88383484532361E10	49.95644916518845
350	1.0965403983020988E12	95.76784467423039	6.89405368522126E10	50.03060796872526
400	1.0912547330163325E12	94.84680836760539	6.900353492409264E10	50.07632609018074
450	1.117312688027241E12	99.43056374071439	7.071578358273862E10	51.318916375912515
500	1.117540330559608E12	99.47108405587606	6.964942313500061E10	50.545051477986036
1250	1.036349908875807E12	85.54276181728534	6.056744756783025E10	43.95420116086357
2000	1.0458135688270267E12	87.11220064154166	5.979551168195427E10	43.39400213349473
2750	9.122144531864618E11	66.27722837153217	5.117096572923099E10	37.13511154211538
3500	1.0006611358537988E12	79.75253679838322	5.095670600957132E10	36.979621832762504
4250	9.975976953654591E11	79.26497280042062	4.865580892565688E10	35.309845453903165
5000	8.829899413188435E11	62.0986205848094	4.308193230828757E10	31.264845971125006
5750	8.42530328172219E11	56.538141240295495	4.209842044292396E10	30.551104842678644
6500	9.479513314051864E11	71.5718960248338	4.442360284111487E10	32.238505236282286
7250	9.375043002099005E11	70.00305242909451	4.29155291705457E10	31.14408610284931
8000	8.477372403070499E11	57.23912208170999	4.2019595773562225E10	30.493901253742933
8750	8.113610364384458E11	52.43227918711134	4.13931463820961E10	30.039282747016255
9500	8.709131974829404E11	60.411578563217624	4.014724306957336E10	29.13512239315384
10250	8.696633453629755E11	60.23830903289736	4.33124969915039E10	31.432168301413178
11000	9.032243033405741E11	64.97730134324505	4.792262094404151E10	34.777766039520095
11750	8.960071883107059E11	63.943061863837514	4.824509863856543E10	35.011790255896166
12500	8.360394904704419E11	55.67035931380539	4.804043966507731E10	34.8632678721524
13250	8.809358591505405E11	61.810039032216835	5.091715197669779E10	36.95091720697
14000	8.803275482303777E11	61.7247053737051	5.0870346527868935E10	36.91695018019478
14750	8.493287168603379E11	57.45423644378393	5.040329547926733E10	36.57800811532429
15500	8.294352533401682E11	54.79430476227649	4.986882571544443E10	36.19013983860514
16250	8.291354988833165E11	54.75470705030925	4.776158655330975E10	34.66090230680137
17000	8.143731045822101E11	52.82229728479611	4.65030937623902E10	33.74760568439564
18500	7.630804576356536E11	46.377902536495476	4.873663541223285E10	35.3685017749591
19250	7.595048762573066E11	45.94429307549111	4.74935981426722E10	34.4664213276133

SVR RBF für Schach (alle Parameter)(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	9.452000610180761E7	68.93184630626885	5.111858782310379E7	62.48660358852647
100	9.544378200588425E7	70.28581892028953	5.2074131501333244E7	63.65464600862752
150	8.80740385667086E7	59.85056452020291	4.5033251152743965E7	55.04797829748043
200	8.68709448085243E7	58.22661174386068	4.520458499600469E7	55.257414246340765
250	8.247435411942585E7	52.48198772290649	4.094465887951708E7	50.05014374273219
300	8.06258327487601E7	50.155764626256186	3.987843128122396E7	48.746802940356254
350	7.99031360023405E7	49.26064321754204	4.016461786281651E7	49.09663317311255
400	7.95400371550907E7	48.81395630524323	4.008240913073474E7	48.99614243829637
450	7.984141559460177E7	49.18457079089913	4.067177824709016E7	49.716578504888496
500	7.95519991998155E7	48.8286396938093	3.929711649441575E7	48.03621236674467
600	8.3846661661565953E7	54.24297668742955	4.249106194923O5E7	51.940443919639456
700	8.701472887462938E7	58.41951830195141	4.719768659540198E7	57.69375208071473
800	8.698102485104436E7	58.37427098531601	4.758004557886653E7	58.1611420311363

KNN Regression für Text-To-Speech(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.4626028351777092E8	165.05369490164605	8.181781264533238E7	100.01288069500495
100	7.805870665861924E7	47.01269323821746	4.722418960272565E7	57.72614896378726
150	8.293287005952273E7	53.067156763355136	5.293377409638259E7	64.7054603670082
200	7.809354732258777E7	47.05466982376309	4.765889018203455E7	58.25752050466346
250	8.804231306338361E7	59.80745426144636	5.5403273602033526E7	67.72413993628082
300	7.981224577832933E7	49.14863849184829	4.740595921225739E7	57.94834143855899
350	8.304791094141096E7	53.21448379376142	4.4834558143216506E7	54.805098909553706
400	8.428401321726853E7	54.81038366656442	4.566280889032619E7	55.8175403386018
450	8.516738449099523E7	55.96532747991447	4.6222426979551405E7	56.501609190882974
500	8.532799892374018E7	56.17661294469458	4.652880909133679E7	56.87612614021654
600	9.126811970288777E7	64.27034566520258	5.288544931860351E7	64.64638887538102
700	8.988068475302406E7	62.33115491271442	5.078694013493579E7	62.08120237334973
800	8.894171863600063E7	61.03563444888697	4.954818691963023E7	60.566968815555185

Linear Regression für Text-To-Speech(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	8.937297150403607E7	61.62895794705414	4.787546938254766E7	58.52226370717162
100	9.615292109871146E7	71.3341341330891	5.284786654971653E7	64.60044825610322
150	9.2660233853733E7	66.24593065984445	4.9694733022338234E7	60.74610459800378
200	9.225119785933872E7	65.66235536327034	4.964891566407331E7	60.69009813880187
250	8.754169110942088E7	59.12923945189872	5.0369157856151484E7	61.570511512110656
300	9.005726933950655E7	62.57631400231182	5.162955759720209E7	63.11120546192504
350	8.993037403164661E7	62.400091776089305	5.263966943630145E7	64.3459511925446
400	9.00929740836346E7	62.625942737534245	5.3562193659465216E7	65.47363112050212
450	8.928051771136467E7	61.501517095913385	5.2790850241306916E7	64.53075236634193
500	8.916460334001073E7	61.34192389274903	5.259477455875423E7	64.29107236009021
600	9.655193125482394E7	71.92739954542834	5.522067507382697E7	67.50093420361513
700	9.751758517302586E7	73.37334283380847	5.533473946675114E7	67.64036482577714
800	9.57322684227798E7	70.71134999847271	5.484406066979447E7	67.04056634189044

RBFNETWORK für Text-To-Speech(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	9.008972132948087E7	62.62142067307616	4.681798519418157E7	57.229610719412406
100	9.64544328997574E7	71.78220799355176	5.190004726752174E7	63.441848023150406
150	8.647381616050364E7	57.695465191637126	4.349030508707716E7	53.16190391104711
200	9.165505621978791E7	64.81645660938632	4.8582419413231145E7	59.38642893953487
250	8.016171380419415E7	49.57998786380537	4.036564786009533E7	49.342369260205444
300	8.05967136244963E7	50.11954228523777	4.025131362810842E7	49.202608790775265
350	8.482479273568761E7	55.515984177816804	4.444820101563804E7	54.332821687065845
400	7.896363442248489E7	48.109039625877635	3.960595815463557E7	48.413735831605656
450	8.078314252752796E7	50.351674276675155	4.299325133138801E7	52.55431277215617
500	8.1835482394933E7	51.67205366805494	4.289372344362593E7	52.43265135830582
600	8.247261111097385E7	52.4797694434783	4.4161035079447135E7	53.98179430577385
700	8.711194498406479E7	58.55012816378273	4.8080592837608576E7	58.77300357634514
800	8.713622257332587E7	58.582767867527906	4.925775014458192E7	60.211943209355425

REPTREE für Text-To-Speech(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	9.578322140526828E7	70.78664149765119	5.3674290280018665E7	65.61065636690321
100	9.651555408733062E7	71.87321063125606	5.420368669052398E7	66.25778268735066
150	9.044914964904943E7	63.12209513952954	4.8747759311213546E7	59.58853797858594
200	8.526294484533517E7	56.090987482321786	4.516828226815888E7	55.21303833024673
250	8.724812019710173E7	58.73332480844865	4.65996878743918E7	56.96276731338279
300	8.640505901443477E7	57.60375191723778	4.6237262685971506E7	56.519744138374804
350	9.213720857574786E7	65.50018553225144	5.2822528088870965E7	64.56947489737354
400	9.077181475661936E7	63.57325751058841	5.288003019260011E7	64.63976461612835
450	8.860079883870551E7	60.56862354116729	5.1596387270671956E7	63.07065854674101
500	8.777215373305212E7	59.440977021462324	5.0501863111540854E7	61.732728448077765
600	8.432655441325913E7	54.86572720400208	4.4483951222866655E7	54.37652221015201
700	8.589626785099111E7	56.92735657096431	4.577716412565704E7	55.9573264822081
800	8.507659832156584E7	55.8460759684871	4.520644510528367E7	55.259688016335126

SVR RBF für Text-To-Speech(static): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.668951457843232E8	60.240847746712376	2.6117089140105793E8	66.23157109372781
100	2.074807475093369E8	20.61568147994956	1.3072753690511735E8	34.894133611894596
150	1.5450034853242153E8	10.39625875013621	9.346895005919115E7	24.309518419022417
200	1.3640840027358076E8	8.249687905906235	8.185413742249185E7	21.37004060254352
250	9.982191976817511E7	4.391967592920864	6.2199688063162886E7	16.19357744971711
300	9.219674839932425E7	3.9868713071358886	5.747317609316241E7	15.573320891320133
350	9.361003095538439E7	4.020130278309736	5.7040130587266E7	15.22327851220644
400	8.773257188988793E7	3.4934089509799335	5.204709193881652E7	13.754860276632627
450	8.56794137239436E7	3.2989602041396027	5.1731857429833114E7	13.48798513566746
500	8.319929631115876E7	3.1711546824819146	5.0570859269992456E7	13.271621016951793
750	6.034054477274445E7	1.791310107604451	3.6396878319016546E7	9.926641585549282
1250	4.8164292346764624E7	1.1419998967624307	2.7095878531585783E7	7.45958351267584
1750	3.911270055496792E7	0.7696276529563624	2.2503815519693796E7	6.214605146298383
2000	3.6701873740695156E7	0.6915429304933605	2.106671172593218E7	5.8372151210607255
2250	3.0149141689991925E7	0.44899295181404103	1.8468180999520298E7	5.040388909607266
2750	2.8218682698131368E7	0.39435296783609997	1.6603713163315443E7	4.483493670323126
3250	2.5503128934336852E7	0.3222449029108073	1.5353059505768018E7	4.168916618622864
3500	2.478837299487742E7	0.3067140243910176	1.4560722421725098E7	3.9622861191808565
3750	2.429472495105672E7	0.2958352391523238	1.4208531118340675E7	3.876607741584812
4250	2.3146575704191014E7	0.2726541616266524	1.3755007226805482E7	3.7866146617997627
4750	2.185910625434068E7	0.2394430018421757	1.2717434124986622E7	3.4807024161838407
5000	2.1140185451100685E7	0.22353933580453023	1.2549925079096274E7	3.4286111901935055
5250	2.034921313616682E7	0.20772668702719405	1.2222334808124624E7	3.339517743396756
5750	2.0431690261030108E7	0.2077124617081101	1.2202119526111575E7	3.3270876880726497
6250	2.0854495988420818E7	0.21374354153857644	1.1987424788871085E7	3.2447788763856145
6500	1.9575334380026594E7	0.18887913572279016	1.1741078135997972E7	3.1811610924108322
6750	1.9438746803400464E7	0.18629719012910087	1.1701331617535856E7	3.1742553159323297
7250	1.9396104994572435E7	0.1837497674568533	1.1606685866311055E7	3.13342014380248
7750	1.886846018181413E7	0.17240409202317755	1.137632576343658E7	3.050153135200988
8000	1.892654596121179E7	0.17167395973217234	1.1524696937249238E7	3.0800880763224243
8250	1.8238111315775793E7	0.15993003612303516	1.1305960375117315E7	3.024034846288225
8750	1.8191471144508533E7	0.15875255517429626	1.1274422320697496E7	3.0165862458966974
9250	1.8286263221599374E7	0.1625451859375459	1.1245084770132652E7	3.0226677317802246
9500	1.8273557828454737E7	0.1630455310654095	1.1241568148169875E7	3.028629484546391
9750	1.8484041764497146E7	0.1664081515167155	1.1272493431455541E7	3.0361569529692694
10250	1.846762056103418E7	0.16553392902115738	1.1236141394839127E7	3.020443958876523
10750	1.83840977904046E7	0.16495828131462384	1.1152198821692297E7	3.003157310888502
11000	1.82863100255073E7	0.16425457239694155	1.1081337207160495E7	2.9924364343041425
11250	1.821915220200726E7	0.1631385542680214	1.1037822675133308E7	2.984096506946141
11750	1.8089851705563523E7	0.16069477500983864	1.0991126448541678E7	2.9697075472718706
12250	1.825295019042203E7	0.16476968126265673	1.0851441040549416E7	2.9449300236567795
12500	1.8281489077354454E7	0.1643989518019748	1.085650692723021E7	2.9390813860299136
12750	1.7887356443386793E7	0.15798293585832585	1.0777555713743906E7	2.9219321423484783
13250	1.7914551650441058E7	0.1584655876546737	1.0781397369187439E7	2.9219310093516344
13750	1.7793490480195373E7	0.15642976881734252	1.0746195115314964E7	2.916322165292111
14000	1.7997990662662324E7	0.15963901302505273	1.0806619318864068E7	2.9284413560736136
14250	1.770381671343097E7	0.15460562531129282	1.0735537765633814E7	2.9112925392661944
14750	1.7743731813034493E7	0.15520677392887733	1.0757497753652014E7	2.9162602394078005
15250	1.760190372237924E7	0.15328026244194987	1.0717625454988865E7	2.911227400725447
15500	1.7561958585845314E7	0.153016486900246	1.0699629711134886E7	2.9062374321886386
15750	1.735664401003899E7	0.14957515080324862	1.0644105099931749E7	2.8935125993042026
16250	1.724687999139877E7	0.14808003601580413	1.0613350667398874E7	2.8839686630085333
16750	1.710088645772917E7	0.1452678951313303	1.0549104442074442E7	2.86475551362814
17000	1.703357122358287E7	0.14333310522233814	1.0559977142945765E7	2.8626646066679218
17250	1.695226622347667E7	0.14220703130469153	1.0528560861012239E7	2.8547233540438826
17750	1.6617957757969117E7	0.13605565890023996	1.0453910386561772E7	2.830178703306745
18250	1.6661425136179242E7	0.1360866007862917	1.046607956794894E7	2.828090333829676
18500	1.6721485309846384E7	0.1375228650115591	1.0492185442995796E7	2.840724813409348
18750	1.6676779848404823E7	0.13645052369495714	1.0491703617732607E7	2.838013735252157
19250	1.6552023149514288E7	0.13479815068287387	1.043424552912612E7	2.827874704013753
19750	1.6755840633338816E7	0.1387011556426391	1.051966338507186E7	2.8580617066334577

KNN Regression für Gesichtserkennung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.2451194691291326E8	80.64676983003014	2.0899759838580418E8	53.000697059502755
100	4.5169439055464476E8	97.70830559689927	2.1654622130774596E8	57.80107969866902
150	4.771186571582861E8	99.14503079371177	2.4481493641666263E8	63.67176695901533
200	4.934462716666638E8	107.95307843005773	2.479276332255107E8	64.72762105078193
250	5.714777187155668E8	143.94808639165632	2.7436310385712534E8	71.4299429788686
300	5.597349851112751E8	146.94875568184676	2.662074008613746E8	72.13335957870783
350	6.358762000970812E8	185.4985375935208	2.8634416199773103E8	76.42158044794117
400	6.080265939593621E8	167.79262048967013	2.783611020699608E8	73.56449559031493
450	5.804981880264156E8	151.43485336047036	2.6498473112310418E8	69.08915109833507
500	6.253961598151984E8	179.17959090331854	2.7700658959066254E8	72.69653965375711
750	5.1032941015742147E8	128.13079404012356	2.311913596217566E8	63.05358785239411
1250	5.530488260605084E8	150.5714548445441	2.4715073966741353E8	68.04140270334972
1750	5.346966847557607E8	143.833497259457	2.3788351200630605E8	65.69339748809132
2000	4.903538113185017E8	123.44166836140424	2.2022412771006542E8	61.02023063757095
2250	5.2747074844992864E8	137.43152909274434	2.3685475411255774E8	64.64307859272668
2750	4.8425861099215126E8	116.13586131627274	2.2128429272947878E8	59.753304338367585
3250	5.0011936846328014E8	123.92144581244895	2.262535763641741E8	61.43611272874296
3500	4.9158324361034554E8	120.62344597389689	2.2273664834285033E8	60.611438388864144
3750	4.956047802483322E8	123.11101298248126	2.2228895398817283E8	60.648568998593056
4250	4.933345717155483E8	123.85733497473979	2.2159217847328773E8	61.00209022877942
4750	5.0108982029877996E8	125.82545410109213	2.2592366300213647E8	61.83433876289909
5000	5.050219807179346E8	127.57246063411498	2.2679642322539583E8	61.96027065226508
5250	5.0579554642918533E8	128.335573478108	2.2553272369598916E8	61.62247592814217
5750	5.1792240881537414E8	133.4699089842833	2.3082114374588042E8	62.93678601167506
6250	5.1953023107505511E8	132.65244288721553	2.3195162024086602E8	62.785104470452325
6500	5.1459091089963186E8	130.52383646715467	2.29526537650192E8	62.18857270099397
6750	5.195783068912171E8	133.0980827046989	2.3189111782007688E8	62.9057988882874
7250	5.3030315476537967E8	137.35553565263965	2.352199278061469E8	63.50157732370847
7750	5.2943747909577996E8	135.7388026987936	2.3588334168512794E8	63.243645544590755
8000	5.468422209610888E8	143.3130616037067	2.427080312375989E8	64.8660973137095
8250	5.448769346428636E8	142.7470556961296	2.4084657641533643E8	64.41986487871799
8750	5.528748041435118E8	146.63534908764763	2.4431912663947868E8	65.37006474177822
9250	5.501756687391055E8	147.13870784027517	2.409773078334773E8	64.77446345394952
9500	5.48104630645127E8	146.6863216579908	2.3974291695742336E8	64.5899626669627
9750	5.516982602024018E8	148.24632734626564	2.408633167095448E8	64.87462939674528
10250	5.4968575581381 98E8	146.65406445852454	2.408246791740336E8	64.73729920252585
10750	5.485841192596091E8	146.8858973701029	2.3970352781300902E8	64.54936945682668
11000	5.443285650475819E8	145.5417398251059	2.3771384871809846E8	64.192937056644
11250	5.444701317320517E8	145.69625954342976	2.3743222473498207E8	64.1902568397217
11750	5.453140570337852E8	146.0242985074253	2.3750690976509908E8	64.17231807502476
12250	5.45125830563879E8	146.96187593269116	2.365057608271515E8	64.18437083379334
12500	5.519737362827038E8	149.8693201600733	2.3916324154882544E8	64.74644525815846
12750	5.452238724520563E8	146.78023667870238	2.3667755044836944E8	64.16628782957704
13250	5.493137042808864E8	148.99234184251557	2.379454607675143E8	64.48702302153086
13750	5.507786509104681E8	149.88262780974227	2.3846040846160823E8	64.71382356999165
14000	5.520836577493864E8	150.21066654556125	2.3912609963073757E8	64.79979897624814
14250	5.536178857402371E8	151.18604532982368	2.3942129426529256E8	64.92692242835687
14750	5.542855337415928E8	151.456470074079	2.3925199168480346E8	64.85904868654532
15250	5.561972549533648E8	153.04693450054333	2.3924839030324337E8	64.98701343458937
15500	5.481251391769348E8	149.05679244489346	2.364588235248607E8	64.2270342668099
15750	5.467798160978131E8	148.44066335185357	2.3620599312041688E8	64.21066033343381
16250	5.407378213190436E8	145.56230439934558	2.3383411428498867E8	63.539807462664236
16750	5.371406743264389E8	143.3205535805474	2.3335920203145126E8	63.37192548868173
17000	5.438229628227925E8	146.0999850109735	2.362044441412548E8	64.0317770604774
17250	5.411004303924565E8	144.884291336715	2.350809993232176E8	63.74007119482455
17750	5.460738062266271E8	146.9142757859844	2.3698802272653082E8	64.1595757049528
18250	5.496866706402618E8	148.12265431955817	2.3859807316776678E8	64.47274741370846
18500	5.459872290398486E8	146.6189742943954	2.3722802611720878E8	64.22871039485362
18750	5.459022207269526E8	146.21116501061113	2.3744717678068864E8	64.2296402618048
19250	5.422968535828125E8	144.6958026628313	2.3630654943175298E8	64.0434722055842
19750	5.409299510612674E8	144.55386065420873	2.3577243438739413E8	64.05643807563311

Linear Regression für Gesichtserkennung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.0780340080047137E8	74.42328396109986	3.0362992732566184E8	76.99896037407913
100	3.186184708028342E8	48.61653308179659	2.3458114113175452E8	62.61500732950566
150	2.5278994396912086E8	27.831572207444182	1.9032539208594158E8	49.50005987647121
200	2.5780233213007393E8	29.466528302036576	1.853169007778124E8	48.38154574299123
250	2.3331333280676877E8	23.99310231528843	1.7086393935391626E8	44.4841207641052
300	2.172688062576045E8	22.140925186526133	1.6037694766620597E8	43.45682350193661
350	2.4693203476384032E8	27.973719754532834	1.776409190760135E8	47.41008055935622
400	2.3710910670762476E8	25.516680485955163	1.7262996163850558E8	45.62216472515732
450	2.4638695826687145E8	27.28095294800777	1.785033761323473E8	46.54097114539513
500	2.3625262211646807E8	25.570039271999345	1.7011912449097097E8	44.64540608111652
750	2.2453879696782103E8	24.8047643643893	1.6800625021663263E8	45.82090297456274
1250	2.2448080181587127E8	24.80699201764361	1.6470933539438263E8	45.34501589455658
1750	2.1677912670351437E8	23.641769692945925	1.605321755022087E8	44.332219269635935
2000	2.17587630978494E8	24.30586490857169	1.6286104522169572E8	45.12592986354026
2250	2.1720890111996643E8	23.30477181290008	1.6072014021235207E8	43.86420147701097
2750	2.1363012131767714E8	22.601466297785525	1.5863459364368832E8	42.83603248863505
3250	2.1255060633912095E8	22.38328476856817	1.5558701766477647E8	42.24756006064923
3500	2.1348521477442178E8	22.749570141107395	1.557188614299582E8	42.37445541973806
3750	2.121229871051847E8	22.552848366975258	1.5664324527203918E8	42.738015086211945
4250	2.1180698869867083E8	22.83069861521464	1.5488092442659873E8	42.63715529890455
4750	2.128189196614874E8	22.6964437891768	1.55265251562844E8	42.49543423502678
5000	2.131938252581164E8	22.734510859933483	1.5487994128739464E8	42.31285019533468
5250	2.1145635385433537E8	22.430478192337922	1.5340049376960057E8	41.91373242769224
5750	2.1299443964684 97E8	22.57306294259202	1.550669686824787E8	42.28129393636826
6250	2.1280395912392 84E8	22.256302298686016	1.547497751919624E8	41.88796263685172
6500	2.1119140122887862E8	21.984574861515778	1.535015034592064 6E8	41.59013378284519
6750	2.119943459423506E8	22.15737637531924	1.5459740856216565E8	41.93810260213428
7250	2.153953691048623E8	22.660551188132157	1.5545547414788178E8	41.96782094130677
7750	2.1482979506082174E8	22.34928730711094	1.5477137325774986E8	41.4963846146827
8000	2.158505719741921E8	22.3289398910887	1.5511592342028394E8	41.45624902554662
8250	2.1084543126228905E8	21.37463468516153	1.535574012608542E8	41.072400478271454
8750	2.1075040858275145E8	21.30698160299308	1.5356668367932737E8	41.08832653577313
9250	2.1053565744211793E8	21.546438764682453	1.5220805818108588E8	40.91337641160958
9500	2.1032613679716653E8	21.599771938044533	1.521070164787053E8	40.97967372895289
9750	2.0834320012613827E8	21.14167417138212	1.5179518826170564E8	40.884833428423406
10250	2.105872806549577E8	21.524318542000483	1.523379628901067E8	40.950737762193015
10750	2.08914187667872E8	21.30244768741855	1.513348469908054E8	40.75271248290606
11000	2.0706247980911106E8	21.060473575508677	1.5119031464217454E8	40.82787100430464
11250	2.0849601660988334E8	21.364651130143915	1.511828217211917E8	40.872565494717975
11750	2.0873111211384916E8	21.394695836969273	1.511368170179346E8	40.83586411912856
12250	2.0835785459666717E8	21.469944321952624	1.5087717153672165E8	40.94596382937259
12500	2.0858746864473575E8	21.401900315113487	1.5100023475359455E8	40.87889246745619
12750	2.0810940505452868E8	21.38460697953013	1.5095734254807097E8	40.92645150998652
13250	2.0657200474792588E8	21.07007139035285	1.50847433691473E8	40.882065570081906
13750	2.059247277461034E8	20.95145091524181	1.5035166312571812E8	40.802710453039545
14000	2.0771363421236262E8	21.26282664416366	1.506434532653544E8	40.82225028450227
14250	2.078459464160122E8	21.30956577111004	1.5073853720211056E8	40.877689438270565
14750	2.0730845015766403E8	21.186282171287854	1.5028275386147583E8	40.740293866772774
15250	2.078582755237321E8	21.37477701059857	1.5076104355943432E8	40.951205359381895
15500	2.0784821595517933E8	21.433097111030097	1.506807826621068E8	40.927970659430144
15750	2.0646686070587945E8	21.165505544576032	1.5096780551089278E8	41.03935870925172
16250	2.0714839086514485E8	21.361835017056265	1.505666266569713E8	40.913510406043315
16750	2.0763453738359308E8	21.415666054975517	1.5101766382477108E8	41.01093959898069
17000	2.0790873932914534E8	21.354065877028074	1.5119469276453108E8	40.986802322975116
17250	2.0764617380894515E8	21.33601991924342	1.5099281660178703E8	40.94032655898471
17750	2.0650886095424452E8	21.01061143234351	1.5116708040546104E8	40.92534140665279
18250	2.0857300448546493E8	21.325895093374104	1.5164154523989758E8	40.97580049107095
18500	2.0674645968781307E8	21.023323398959 49	1.5153918984632516E8	41.028738878017016
18750	2.06831084844185E8	20.988549952692903	1.5155126979599297E8	40.99473269040787
19250	2.0686752308885196E8	21.055540433335146	1.5164368001628044E8	41.09825914528871
19750	2.087743377442627E8	21.532867442467126	1.5190474904344785E8	41.27063104632582

RBFNETWORK für Gesichtserkennung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	7.791605307020193E7	2.716821044691389	5.623675258204663E7	14.261346112260945
100	4.259009354570755E7	0.8686789055839302	2.8168981659209937E7	7.518937731083364
150	2.283720241808591E7	0.22714540494911628	1.6143076040517747E7	4.198510886215486
200	3.979150882089124 6E7	0.7019983667620072	2.6900756098563254E7	7.023105589619888
250	3.0483894470966283E7	0.4095889108404544	1.9621864638387356E7	5.108517335439406
300	1.4285250939282477E7	0.09571428413306546	1.0638425680418156E7	2.8826598451955596
350	2.065166075825392E7	0.19566147099021716	1.4792453788800256E7	3.9479159950612464
400	1.3706541165043926E7	0.0852674855538911	9825357.22422697	2.5966180002163255
450	1.4478908957586538E7	0.09420973431651987	1.0285471163072238E7	2.681718558994764
500	1.466011423112423E7	0.09845841247249547	1.0447587229377422E7	2.7418244469521595
750	1.7128487394224025E7	0.1443412119945154	1.1806006519899176E7	3.2198913943250433
1250	1.5014490944699103E7	0.1109780071215757	1.0633876524766626E7	2.9275408032078056
1750	1.4856679388827363E7	0.11104232848054829	1.0277384094995951E7	2.8381802202098774
2000	1.5111539792780299E7	0.11723576197119906	1.021436421241186E7	2.830220587265085
2250	1.5122754188474916E7	0.1129669501656504	1.0385645991829297E7	2.8344802813922634
2750	1.4676456870513834E7	0.1066728962124439	9855707.398713417	2.6613325166516493
3250	1.5665777749391943E7	0.1215914522032253	1.0307521977415146E7	2.798868828217756
3500	1.5743102244419048E7	0.12371382126128455	1.0408400036602747E7	2.832349782753876
3750	1.5870623157640494E7	0.12624491397898077	1.0242758537310574E7	2.7945997 16902985
4250	1.5826405709310317E7	0.1274686008137082	1.023814129559172E7	2.8184569662686276
4750	1.5967855497304568E7	0.1277704821981229	1.0351266884163717E7	2.8330974039426113
5000	1.5645945401962321E7	0.12244469996315951	1.0167781416144034E7	2.777814921054763
5250	1.561649571452842E7	0.12233880282137959	1.0211243078348624E7	2.7900256356596147
5750	1.6493724385656621E7	0.13536033692040061	1.039179824112633E7	2.8334769144819383
6250	1.6603466499491645E7	0.13548497792205885	1.0429246533168152E7	2.823008230997114
6500	1.6685057137689821E7	0.13722113421553217	1.0628397736283163E7	2.879688301338224
6750	1.638801398842 05E7	0.132410501562242	1.0470368193109276E7	2.8403281765759307
7250	1.61940009291389E7	0.12808729824087103	1.0391807060633799E7	2.805443168649132
7750	1.6253837259450765E7	0.1279341430559239	1.0420555180183766E7	2.7938975829548465
8000	1.6353366339866366E7	0.12816694376853588	1.0493355862490166E7	2.8044520778850393
8250	1.6530951844231658E7	0.1313911254563383	1.0597120836691402E7	2.8344396775890264
8750	1.6281857245089553E7	0.12717242877853774	1.041673333454492E7	2.787102843085432
9250	1.6351107397366084E7	0.12996263853763712	1.0400437459134758E7	2.795627364910838
9500	1.6292021324662182E7	0.1296022721539343	1.0388907014583403E7	2.798911120038041
9750	1.6303563670239115E7	0.12946304034366152	1.038762361134926E7	2.7978242652525065
10250	1.6287132872248268E7	0.12875212819899406	1.0379629002071971E7	2.7902005335290374
10750	1.6321215446686732E7	0.13001654878008445	1.0426989638696527E7	2.8078669206561355
11000	1.6224779199014412E7	0.12930725770171717	1.034326569504963E7	2.793125462832636
11250	1.6233178919277288E7	0.1295112312534101	1.0296378114214748E7	2.783645549411206
11750	1.6091112041852124E7	0.1271463391214198	1.0283611446124751E7	2.778543097265529
12250	1.6145449228289675E7	0.1289173681349443	1.0248997055117376E7	2.7814351132904385
12500	1.6156236638885299E7	0.12839742698102277	1.026311803798754E7	2.778438717928651
12750	1.6060089880734343E7	0.12735435497950467	1.0197460323437728E7	2.7646609194864813
13250	1.6004297457703006E7	0.12647255951563088	1.018072144650314E7	2.7591382335210892
13750	1.5848472603069592E7	0.12410004235240636	1.0140176745373795E7	2.751859787132339
14000	1.594106826039657E7	0.12523503245237322	1.014516576936871E7	2.749196777143864
14250	1.6036458933238242E7	0.12685524694991207	1.0190641326570014E7	2.7635260302798135
14750	1.6071217984723588E7	0.12732636068067446	1.0186981226448143E7	2.7615983745103767
15250	1.6061898124048395E7	0.1276323268534775	1.0235570153742433E7	2.7802867732938057
15500	1.6083511957368568E7	0.12833765572128233	1.0239056617563294E7	2.7811363958972897
15750	1.6033074020187162E7	0.12763257535231243	1.02108942621524E7	2.7757477890640527
16250	1.5983629550786195E7	0.1271821725462832	1.017837509716706E7	2.7657726330237575
16750	1.5914824938033141E7	0.12581607377445547	1.0144971789217094E7	2.7550076907804795
17000	1.5901076658435985E7	0.12490739489956068	1.0161834075511223E7	2.7547334954442166
17250	1.5868632450422322E7	0.12460760373020233	1.0175354884682178E7	2.758954771544531
17750	1.5782142169694092E7	0.12271374527183443	1.012796203207592E7	2.74193496894017
18250	1.5776690738690011E7	0.12201771812625141	1.0135624314501189E7	2.7387964103532303
18500	1.582821227694625E7	0.12322219487040538	1.0144446213865327E7	2.7465755570743675
18750	1.5814170742868949E7	0.12269974566956879	1.0142231980390662E7	2.7434813939857494
19250	1.5764261481176954E7	0.12227256525741276	1.0122393638137165E7	2.7433570384609496
19750	1.5857493751839725E7	0.12422721106708685	1.019828161438637E7	2.770746276625525

REPTREE für Gesichtserkennung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.549876927647808E8	56.39410859878916	2.827142444547502E8	71.69485267047062
100	2.438245776872663E8	28.47062813752696	1.9069979107639253E8	50.90208342569592
150	1.86966725119093E8	15.22463169364004	1.4010385756611302E8	36.43838198591819
200	1.7267039308049774E8	13.218769899732049	1.3288721785792778E8	34.69348441759576
250	1.370348565135816E8	8.276932049832384	1.1081076220617642E8	28.84938358896098
300	1.3682888994785467E8	8.781255535908047	1.0840000372690168E8	29.372798884872793
350	1.3618555559039444E8	8.508586991303627	1.0922157764355068E8	29.149836770912763
400	1.4031686723128363E8	8.9360886108042	1.1284176540564923E8	29.82150699885108
450	1.439253368332953E8	9.308905326510011	1.1159093392829895E8	29.09497035055525
500	1.6437986668412948E8	12.378709413488936	1.1162986145365247E8	29.29570975802666
750	2.1149291690339583E8	22.006145337956927	1.248456491846912E8	34.0495688149208
1250	2.723026306363676E8	36.50222778903575	1.492471027222222E8	41.0882129355401
1750	3.2488803037985295E8	53.1021958334848	1.6277290204518592E8	44.95101347781131
2000	2.933681859620615E8	44.18437366864569	1.5082705078780925E8	41.79152176083792
2250	4.48665803382745E8	99.43414805031613	1.9917503962926129E8	54.35942288219899
2750	3.7874181515565777E8	71.03921499137431	1.7615636493885547E8	47.56742900952295
3250	4.4227361369812745E8	96.91280936749101	1.9504741555330968E8	52.96249987268947
3500	4.46583933500207E8	99.55057637449994	1.955570801914226E8	53.21902660900053
3750	4.439407653406979E8	98.78158364281167	1.934546253298337E8	52.781507951302174
4250	4.53386425586794E8	104.61059102595229	1.9620474628024343E8	54.01318637853005
4750	4.5623876235979223E8	104.30898206006212	1.9848549419182175E8	54.32462949780733
5000	4.6259136956160754E8	107.03638242818106	2.0056172226439652E8	54.79300959537758
5250	4.646441575331179E8	108.30238357680435	1.9965670970841143E8	54.552353140926726
5750	4.5396387909689E8	102.5407659588422	1.9700140823011962E8	53.715336786597845
6250	4.5796073731539667E8	103.07422931094604	1.9873835775324053E8	53.7948755902903
6500	4.468218151791565E8	98.40889273502398	1.9432992658925655E8	52.65230282910903
6750	4.4923416520922256E8	99.49822527114745	1.9557125201554823E8	53.05319998140742
7250	4.523205126522242E8	99.92871715492552	1.9672850233100736E8	53.11016935965306
7750	4.382872165292986E8	93.02341049354222	1.9283672780493295E8	51.702242193742954
8000	4.47884398022158E8	96.13766400445634	1.9709075369571784E8	52.67443332496647
8250	4.527753875174967E8	98.56799738887247	1.977516374083338E8	52.89314862180829
8750	4.430805237976032E8	94.17820338758527	1.9504369761528146E8	52.18592304236951
9250	4.371458502524675E8	92.89169721368064	1.9098397250181717E8	51.33630406252906
9500	4.2942058740648885E8	90.0385306659767	1.8791597185978246E8	50.6270873858805
9750	4.324982560399921E8	91.10651521230476	1.8893354126028538E8	50.887755085828644
10250	4.2348856592152256E8	87.04593920221345	1.8621606121644318E8	50.05768060246446
10750	4.196263109645253E8	85.94469218270717	1.8426728520638335E8	49.621034701132885
11000	4.131472538572889E8	83.84459123051246	1.8140985198018333E8	48.98844250100434
11250	4.108974869847478E8	82.97881944088762	1.804857979059198E8	48.79468124613061
11750	3.9962926904295915E8	78.42351790251976	1.7656672378989106E8	47.706805548168276
12250	3.975566789180011E8	78.1644492479202	1.751261175900327E8	47.52679019221054
12500	3.942626346581349E8	76.46226329514663	1.744151075930489E8	47.21778373146749
12750	3.851541942000825E8	73.24654105682515	1.7109531055385906E8	46.38611022672695
13250	3.742778166595787E8	69.1689611817339	1.6735787243571943E8	45.35665836105795
13750	3.733042509734731E8	68.85301681569345	1.670725521189154E8	45.34045601516457
14000	3.679854698084473E8	66.73479502173282	1.6544979128518432E8	44.834558973269225
14250	3.6485343610790515E8	65.66415555641811	1.642200778707918E8	44.53365056694237
14750	3.567111310511407E8	62.72693475256759	1.6125055982355547E8	43.71356675729178
15250	3.4562107936987793E8	59.09726702840737	1.5689514296100518E8	42.61741009210226
15500	3.353909350525041E8	55.807808679573824	1.5301273746475846E8	41.561377096905765
15750	3.306369965844482E8	54.27887425956356	1.5140758628197953E8	41.158909501932214
16250	3.1654989807112473E8	49.88382931298674	1.4629244106951764E8	39.75208479405794
16750	3.079135237527563E8	47.09663999890779	1.43547804442668E8	38.98239575732937
17000	3.091815866371893E8	47.223952258748334	1.442848085778627E8	39.11362772897861
17250	3.076561061765572E8	46.83781430118726	1.4359063388264874E8	38.93329215436119
17750	2.9837629290891474E8	43.86215207462206	1.4054808076278344E8	38.050468222570274
18250	2.9226358493508345E8	41.873618145979954	1.3854943324967617E8	37.43811714664586
18500	2.883021227795369E8	40.8809414262508	1.3702047350515887E8	37.09784402362464
18750	2.870398002023534E8	40.423606131005144	1.3659516933838394E8	36.94909624555388
19250	2.7335987693733007E8	36.766416654665186	1.3168168945439748E8	35.68818824038851
19750	2.6671671730204386E8	35.143750903475265	1.2910495955040969E8	35.07621180646427

SVR RBF für Gesichtserkennung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.2231762732754962E11	94.9335025709393	6.753511783315667E10	86.70077183451097
100	1.0823473224683574E11	68.92982558258389	5.493814084240626E10	67.53452590589409
150	6.555924122112642E10	37.540319922714694	2.794022175680682E10	41.656176000819016
200	5.746120445497536E10	29.23438763825348	2.3128843454725872E10	35.1407186003293
250	3.29464246252329E10	10.348648903116311	1.3045037684429174E10	20.033196348294716
300	3.6130236381026794E10	11.167880573628253	1.4398394421448696E10	22.310108419861983
350	3.3728981244122734E10	8.802310945369472	1.3152737162811121E10	19.848699017354583
400	2.7232394097110657E10	6.020654595930111	1.0943873086228989E10	16.21564422830668
450	2.6746909171400063E10	5.99005959459933	1.0479152556339422E10	15.68588718389262
500	2.448785347138938E10	5.309229653192774	9.238837065232925E9	13.691020717682278
600	1.5322991479390656E10	2.350027518891237	5.7289975398773775E9	9.106051433362945
700	1.4061346696150484E10	1.9061775535498149	5.207162014807975E9	8.207646081300783
800	1.162259879379671E10	1.3470792586490579	4.1939503208734207E9	6.712603525502578
900	8.318820771102177E9	0.7438836664969047	3.616354089531474E9	5.864859201350926
1000	1.007798001205372E10	1.0406502137243712	3.738536604463941E9	6.044256612063092
1100	1.2861539614203194E10	1.6502123578228542	4.431948642858115E9	7.103680102192618
1200	1.1135324915804544E10	1.1840510134002615	4.162727634076056E9	6.5244215026417915
1250	1.074960030162415 3E10	1.1417976213538863	3.969039736423941E9	6.333586302525387
1300	1.0332180475969503E10	1.0823045782116874	3.7883982735824375E9	6.053224224176351
1400	9.849920740707033E9	0.9812390754051119	3.679971104964431E9	5.886572413143134
1500	8.554790071529581E9	0.7446283911671339	3.514791327728161E9	5.591959271256261
1600	8.7581717332477E9	0.7619065738794168	3.4083040826657944E9	5.321411156852316
1700	8.463848882083886E9	0.7256502120344765	3.2342945974072356E9	5.032605711398345
1800	7.3181151595201845E9	0.5382414407756388	3.044915041847408E9	4.707685048972462
1900	7.036857410369626E9	0.5089993231171329	2.8858858337095246E9	4.4836324306893465
2000	6.775107646714135E9	0.4841283114499285	2.922927660604547E9	4.541708175678942
2100	6.764157685604117E9	0.4921814499556636	2.890847032553582E9	4.554327939117017
2200	6.4821373135921335E9	0.463249293819895	2.79881930544341E9	4.438574595984049
2300	6.414495579549788E9	0.4643129822460043	2.8871414262381544E9	4.605026316899978
2400	6.564568387317144E9	0.47612378393092036	2.927211117948629E9	4.654691015087818

KNN Regression für Rauschreduzierung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1477800987310242E11	83.59085908583945	5.9860958467057724E10	76.84877835956718
100	1.253677708252801E11	92.47954678897061	6.5959810065241104E10	81.08327717854698
150	9.737270915366422E10	82.8141672732615	4.87638410663563E10	72.70218409920936
200	1.6657962692146713E11	245.69047575184712	6.6526917648842384E10	101.07741431262845
250	1.8281220875807233E11	318.6232489752699	6.883434430226859E10	105.70854360654222
300	2.6501815586071176E11	600.8698465966703	9.51298151630904E10	147.40230251564645
350	3.1912761465341846E11	787.9869597371451	1.0073918721285867E11	152.02476727768862
400	2.7654867718900653E11	620.8908169503269	8.75439508145687E10	129.71473166439839
450	2.9257571119116003E11	716.7380753147148	8.945234737957785E10	133.89817752787593
500	2.432312278639488E11	523.8039148159494	7.561202068446658E10	112.04935582125172
600	2.0947569067836246E11	439.1906540829301	6.506003006735792E10	103.41075832652238
700	2.4511102529740894E11	579.2088802872504	7.143329184079202E10	112.59476393939568
800	2.367207637920529E11	558.803816030889	6.957446073379852E10	111.35701061651466
900	1.7039808400813593E11	312.1123130619391	5.471311807586135E10	88.73155836999031
1000	2.20527758848476E11	498.2924652688341	6.2024026484811485E10	100.2769724763341
1100	2.79691667268568E11	780.3922921454871	6.99113467296923E10	112.05631714200874
1200	2.8300846562537946E11	764.8274733576208	7.284646480099141E10	114.17538765889459
1250	2.8053660771773315E11	777.6489385289127	7.131091926838446E10	113.79424029290985
1300	2.555342596691428E11	662.0088534521373	6.586141149122128E10	105.23547491222666
1400	2.701950986814895E11	738.3534291087065	6.844868765781203E10	109.49220659335089
1500	2.4524290895281198E11	611.9469791614579	6.3698198381489334E10	101.34249740280418
1600	2.6410062540222607E11	692.8092325585748	6.685821340587467E10	104.38623846818335
1700	2.537604047414241E11	652.2877790493184	6.438358357745896E10	100.18171835427738
1800	2.3055514930694052E11	534.2300416804125	6.127459321934391E10	94.73547945218192
1900	2.3040491214480362E11	545.6864509884989	6.020163181152922E10	93.53176262819566
2000	2.0439796456277774E11	440.63725605461974	5.470423907580482E10	85.00062906226403
2100	2.028729389237758E11	442.73722241003037	5.415520434377405E10	85.31774854014863
2200	1.917625316422659E11	405.42060922152945	5.1890226145181206E10	82.29135732339813
2300	1.78707661776303E11	360.3895183941454	4.9231487261380455E10	78.52483165474207
2400	1.9835044259424707E11	434.68421555738723	5.2806855615781845E10	83.970573513698

Linear Regression für Rauschreduzierung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	5.819073454209747E10	21.4856791838347	4.270851748151781E10	54.828680964222166
100	5.847188622727826E10	20.117225914073774	4.3251089383459816E10	53.167831521716856
150	4.7882181268436424E10	20.025251788383013	3.456938024539667E10	51.539611971428215
200	4.9430165853493195E10	21.633594568981763	3.7262734484608826E10	56.61499111988111
250	4.8181882256946625E10	22.132710021044545	3.618898875334431E10	55.57524132010197
300	5.238608566879364E10	23.477997543408513	3.880118280215205E10	60.1218837181846
350	5.657187015577517E10	24.762319917800355	4.053256093001514E10	61.16738990095559
400	5.417976123330547E10	23.831217991416636	3.888219467329416E10	57.61212969760007
450	5.289079495694081E10	23.42308673163311	3.805018860794974E10	56.95603366982832
500	5.254722391617368E10	24.44722838567187	3.7952480624594635E10	56.24173203293312
600	5.0957081291535355E10	25.989320027109468	3.7519890426495316E10	59.636620476733704
700	5.010525545001469E10	24.203383362838252	3.673399780894487E10	57.90095493662422
800	4.9280917494026764E10	24.21833185582399	3.6696902727806305E10	58.73502063190805
900	4.8332044559731995E10	25.11029542994513	3.669084277113122E10	59.50374921927739
1000	4.997209584974589E10	25.586620105030207	3.695620714869636E10	59.74872605194409
1100	5.085687593979166E10	25.801994174809252	3.686769564766975E10	59.09281381409486
1200	5.0482375750791084E10	24.335719806363738	3.671280072275066E10	57.541546676493695
1250	4.983405488150742E10	24.539005249916283	3.633196974261303E10	57.976659642341765
1300	4.9791493165283666E10	25.134822865964324	3.6280507528904 3E10	57.970158145938946
1400	4.917346349691733E10	24.455185418895464	3.580310457113931E10	57.271527863092984
1500	4.924875484118948E10	24.678081085510016	3.6046084652705 63E10	57.348564529568236
1600	4.920707884477402E10	24.0508248464554	3.561801699712466E10	55.610681569588095
1700	4.837855817627739E10	23.70813366703506	3.479006070765703E10	54.1338004143488
1800	4.879658391082086E10	23.93081517817169	3.541522606166713E10	54.75480528853633
1900	4.845250350201185E10	24.131934965049478	3.532492626199538E10	54.882276751885605
2000	4.877741472950381E10	25.093754076829246	3.582842685647693E10	55.670984051011565
2100	4.844327441825285E10	25.244383904670105	3.565513011755844E10	56.17217149113862
2200	4.86828882977638E10	26.12949316051891	3.58325904334313E10	56.82597132898225
2300	4.8358959288527695E10	26.39001873018469	3.564730974279037E10	56.85790034405252
2400	4.851944849694099E10	26.00993302653702	3.558175573334123E10	56.5801618125514

RBFNETWORK für Rauschreduzierung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	6.104961107033668E10	23.648696684465992	3.1917722821888214E10	40.97558859220764
100	1.1362387592362129E11	75.96492002000974	5.513183656929702E10	67.77263278182731
150	1.8297802937066948E10	2.9243410767952565	8.831181857546713E9	13.166440444001992
200	1.5572237374132114E10	2.1470718676877394	6.636751472089339E9	10.08352260922754
250	1.980959971824302E10	3.7412649921133094	8.419106805226136E9	12.929178419137678
300	1.1154703754861692E10	1.0644989487276104	4.571631672993276E9	7.0836785890668015
350	1.5559147009489744E10	1.8731037218719275	6.2889439506219015E9	9.4905991101116
400	1.5899701369767973E10	2.052343265239489	5.770454430581093E9	8.550138999665766
450	1.6624399167029497E10	2.3140665981033854	6.101999944143825E9	9.133876256248945
500	9.851915359817806E9	0.8593526482619518	4.538277848873804E9	6.7252680843726225
600	1.4517994650070667E10	2.1095950210489764	5.308169355363516E9	8.437159002162145
700	7.610735956117912E9	0.5584223969148794	3.678247925416494E9	5.797737248283167
800	8.800254288677101E9	0.7722841104425813	3.930327318403788E9	6.290663216154159
900	4.49845465744416E9	0.21752450176193042	2.6052973183853164E9	4.225167550439348
1000	5.030017975994106E9	0.25923692817460303	2.9019995705548553E9	4.691790384394407
1100	4.808093154751057E9	0.2306214308511175	2.7875865960411773E9	4.468039914529873
1200	4.668429354679052E9	0.20811635393076472	2.6923559742946496E9	4.2198449563835085
1250	5.3483412131717415E9	0.2826459158289255	3.0068925303553605E9	4.798242045463473
1300	4.8773454542077159E9	0.24117515267776876	2.7101292546000004E9	4.330331414463606
1400	4.172887882004833E9	0.1761095044928751	2.478933752421795E9	3.9653634837865317
1500	4.0627853329717484E9	0.16794559966000078	2.4115568575661874E9	3.836736372211691
1600	4.050232872164658E9	0.16294265765924884	2.358549861338854E9	3.6824218854042154
1700	4.0469287793600893E9	0.1658985721244497	2.3382890359252257E9	3.6384090572735577
1800	4.667842729402 1225E9	0.2189833443118898	2.5589560754057903E9	3.956353163093868
1900	4.158152017015038E9	0.17772987201138762	2.3475708853089714E9	3.6472839056088766
2000	3.9127062222374E9	0.16146654880226283	2.3063062302653084E9	3.5835884694624247
2100	3.875191383573519E9	0.16154149821205205	2.31490624777748E9	3.6469733894486525
2200	3.8612746641920156E9	0.1643764903741859	2.3096780119728 03E9	3.6628581662656776
2300	4.121042358523922E9	0.1916460901060192	2.472539794119807E9	3.9437315810122517
2400	3.8415528562993827E9	0.16305012967765725	2.310648367172064E9	3.6742666518841416

REPTREE für Rauschreduzierung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1985289143578502E11	91.14617325513696	6.932129898670374E10	88.99384971188097
100	1.097201679394676E11	70.83482294174172	6.736879728914465E10	82.81532130517199
150	6.688784326608317E10	39.07729721751252	4.0578680088119316E10	60.49889848207175
200	4.256173884624479E10	16.039221180154822	3.1335088532774033E10	47.6088452326666
250	2.778294502175365E10	7.359081777902721	2.2107841306094334E10	33.95089661186426
300	2.8698012466695293E10	7.04584029569183	2.306555358614123E10	35.73974890591802
350	4.034664298186709E10	12.595200028725323	2.5424206391844852E10	38.36748307064744
400	4.205295391956145E10	14.357045617464312	2.601867545681451E10	38.55212694584157
450	4.967411414810535E10	20.66066040148911	2.8577002753553238E10	42.77594384050168
500	6.593419469685594E10	38.490317005847494	3.2496665182140766E10	48.15676617336867
600	6.249385505528573E10	39.089530961014205	3.149583961605588E10	50.0615916631889
700	6.30805760810866E10	38.361962830739735	3.1393547140592316E10	49.483216276710365
800	6.731701478631028E10	45.18937697040166	3.220586274032872E10	51.54691191658192
900	6.2381721489305275E10	41.83080272842443	3.0750017479696465E10	49.86915509174176
1000	7.838701329765723E10	62.95729020760018	3.2875047180727 89E10	53.15053517377355
1100	1.2863671120697289E11	165.07593731772198	4.100139025992144E10	65.71844207197475
1200	1.2953326556745415E11	160.2238838906862	4.23428640330844E10	66.36578629824574
1250	1.3020232627163995E11	167.5104821113324	4.19593040339294E10	66.95646578036877
1300	1.254877495564998E11	159.64973069405988	4.072136523722636E10	65.06590297392803
1400	1.7820743039893848E11	321.1892335086882	5.0641973665042404E10	81.0081483307302
1500	1.6280242111294E11	269.67618508829287	4.763177399394586E10	75.78115323392215
1600	2.0108940824715506E11	401.6550202118631	5.479425859296454E10	85.55069381513893
1700	1.9295500032907425E11	377.140871540911	5.279157912230401E10	82.14440416702482
1800	1.7782173425199728E11	317.7959341505204	5.0693578213013466E10	78.37637403752538
1900	1.7643413682592285E11	319.98154952935874	4.971063588480453E10	77.23251439811922
2000	1.36744539212749E11	197.21868786786715	4.175874836991965E10	64.88564579752801
2100	1.5570231767326053E11	260.78820745766336	4.482877104743742E10	70.62460315559434
2200	1.3860775986908594E11	211.81338562362885	4.146469586077221E10	65.75778054691916
2300	1.1024178858047621E11	137.1444722996472	3.620666376973661E10	57.75007694168725
2400	1.6095918592603653E11	286.2457558708391	4.518527712501968E10	71.85115626216793

SVR RBF für Rauschreduzierung(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.123076136559996E12	100.4589973418883	7.69783658034776E10	55.863714114133856
100	1.123039231004499E12	100.45239505720853	7.683533403285323E10	55.759915003045975
150	1.123078464801859 6E12	100.45941386411332	7.680779197399803E10	55.739927547012826
200	1.123058206885241 7E12	100.45578975388605	7.676452766349237E10	55.70853034796597
250	1.123067546688373 8E12	100.45746062222649	7.678559177130232E10	55.723816711666764
300	1.123072079443144 8E12	100.45827152617287	7.682464853333835E10	55.75216046469179
350	1.123058609563635E12	100.45586179179654	7.681948680542749E10	55.7484145642776
400	1.12304984382672E12	100.4542936346015	7.68221315323013E10	55.75033386023104
450	1.123026162406 3992E12	100.45005717923475	7.679528286258937E10	55.7308496013295
500	1.123000508120569 8E12	100.44546789159274	7.676848527346777E10	55.71140241195348
1250	1.122870791755590 3E12	100.42226457445548	7.661323217040224E10	55.59873419830074
2000	1.122702148159842 8E12	100.39210206574778	7.646572123280853E10	55.49168452583197
2750	1.122671146699955 6E12	100.38655783701375	7.649923712400015E10	55.51600723188408
3500	1.122568543054026 E12	100.36790755240419	7.635635371400237E10	55.41231578709365
4250	1.122356414124541E12	100.33028046446968	7.61450723641649E10	55.25898750060987
5000	1.122030204521326E12	100.27196753585193	7.593437188530447E10	55.106080756073105
5750	1.121906934428166E12	100.2499363004284	7.579259057665115E10	55.00318911358802
6500	1.121824711351267 3E12	100.2352424681425	7.558809057192053E10	54.854782094533135
7250	1.121466280370166 5E12	100.17120095363028	7.537073697358379E10	54.697047136763835
8000	1.120763861239705 6E12	100.04575780159965	7.516059762567123E10	54.544547608703084
8750	1.119347625019733 2E12	99.7930749492393	7.479318999897505E10	54.27791744051393
9500	1.119635826293863 8E12	99.8444695173345	7.495929317670424E10	54.39845962580534
10250	1.118750712317524E12	99.68667034373772	7.497151490001949E10	54.40732901203014
11000	1.118435960709563E12	99.63058613182582	7.484815294238403E10	54.317804415580525
11750	1.118758998454039 3E12	99.68814702734406	7.464332230812366E10	54.16915745646801
12500	1.117313685062606 4E12	99.43074119476776	7.45495698258775E10	54.10112065403626
13250	1.116559018359565E12	99.29646962982876	7.424124609250026E10	53.87736806178247
14000	1.117251077842480 5E12	99.41959856226033	7.432396996884277E10	53.93740133126248
14750	1.116600245245783 4E12	99.3038024438832	7.418685128734889E10	53.83789338844337
15500	1.116423809145313 7E12	99.27242256286615	7.40722374960962E10	53.754717394756746
16250	1.115853216723427 5E12	99.17097431990666	7.390181311300652E10	53.63103928187882
17000	1.116194659395614 7E12	99.23167474645741	7.377702582957704E10	53.540480316982375
18500	1.112780132159963 4E12	98.62548836033888	7.320364498878079E10	53.12437401186104
19250	1.112789194039429E12	98.6270946721357	7.33085597756639E10	53.20051137330809

KNN Regression für Schach - alle Parameter(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1164079573833 12E12	99.26960350502976	7.641647979556674E10	55.455949685998306
100	3.016299469537279E12	724.6350823160808	2.9396227409980615E11	213.3303853523784
150	3.731920639765279E12	1109.264821234183	2.7320653766279004E11	198.26780881618757
200	1.1231071297151663E12	100.46454208485484	7.698889694395723E10	55.871356633105385
250	9.407677982551072E12	7049.126876236403	4.8908936988720905E11	354.9354217962184
300	1.123121929575167E12	100.46718986602406	7.694788487255016E10	55.8415938990125
350	1.1231231674885427E12	100.46741133751955	7.694562851782274E10	55.8399564473192
400	1.1231225966978464E12	100.46730921897951	7.694663372606476E10	55.84068593365307
450	1.1231195970565671E12	100.4667725625425	7.695270347471194E10	55.84509079077925
500	1.1231202775411245E12	100.46689430578009	7.695123969670581E10	55.8440285173053
1250	1.0906655502116737E12	94.74441793367016	8.720027897391132E10	63.28182476239618
2000	4.3346601708508414E13	149651.3197465837	6.633981848712029E11	481.43249283951377
2750	1.05686124486465E12	88.96237851874872	6.6104223158546036E10	47.97227618073303
3500	1.0682297147787456E12	90.88657750339185	6.717883992639689E10	48.75213274529994
4250	1.0479254328252817E12	87.46437597161344	6.20548324149739E10	45.03360627686877
5000	1.0503631794314515E12	87.87177897351907	6.370871276357849E10	46.23383829022825
5750	1.05636203919609E12	88.87835606970341	6.3443536189929375E10	46.04139787991974
6500	1.0700191222974763E12	91.19132345357556	6.403002792738576E10	46.46701885030003
7250	1.0629433729723201E12	89.9892635846748	6.3645131446136475E10	46.18769690357361
8000	1.0570312209505352E12	88.99099664239836	6.279068525928691E10	45.56761959990708
8750	1.057232960801089E12	89.02496866370016	6.256481138793646E10	45.403701423114555
9500	1.0677644477878921E12	90.80742352574012	6.362455463074822E10	46.17276417123731
10250	1.0612963212494883E12	89.71059937202006	6.358007682145318E10	46.140486328015434
11000	1.0625233684799034E12	89.91816209662885	6.3691118460051605E10	46.22106998666654
11750	1.0662577742488657E12	90.55133593216432	6.396420710877938E10	46.41925224269534
12500	1.0563482816831604E12	88.87604107328131	6.4242711457815056E10	46.62136470860286
13250	1.0518964439761935E12	88.12850735505782	6.3919646356810 89E10	46.38691420742146
14000	1.0539944942465437E12	88.48040973684367	6.387395744596945E10	46.353757459719795
14750	1.058123427536933E12	89.17499644532067	6.3964609937146484E10	46.41954457793138
15500	1.0575870434813214E12	89.08461016051744	6.398809363563694E10	46.43658685474669
16250	1.058591006794272E12	89.2538257959097	6.370865724787197E10	46.2337980021093
17000	1.0534882651833893E12	88.39543662052358	6.357143993260307E10	46.134218480101
18500	1.0521678200727736E12	88.17398532101292	6.332973788473236E10	45.95881368990941
19250	1.0507919094996141E12	87.94352741850754	6.347442516883787E10	46.06381421818583

Linear Regression für Schach - alle Parameter(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1230660118122842E12	100.45718603562761	7.727709146146898E10	56.080501318985554
100	1.1230608423228018E12	100.45626122592634	7.70940978346486E10	55.94770162199666
150	1.1231063663511765E12	100.46440551553646	7.69915892960198E10	55.87331049097634
200	1.1231091145812405E12	100.46489718693228	7.697213568692755E10	55.859192876949656
250	1.123117735976192E12	100.46643960319544	7.694764411125322E10	55.84141917693155
300	1.1231190753036929E12	100.4666792175144	7.693095869213268E10	55.82931045685292
350	1.12312257919308E12	100.46730608725234	7.69359772301303E10	55.83295244339087
400	1.123121435391908E12	100.46710145319713	7.693458289986742E10	55.83194056860796
450	1.1231163828823032E12	100.4661975262712	7.693171116685085E10	55.82985653278081
500	1.1231140039026777E12	100.46577191270616	7.692370761807643E10	55.82404830398866
1250	1.1230916200992422E12	100.46176736165759	7.690944922581E10	55.813700893505406
2000	1.1230756799667249E12	100.4589156575013	7.68986836585696E10	55.80588824946812
2750	1.123082018470989E12	100.46004961681234	7.68983833330239E10	55.80567030121407
3500	1.1230179869507153E12	100.4485946629921	7.68429223162774E10	55.765421871000385
4250	1.1230884423715237E12	100.46119886005818	7.689877904259576E10	55.80595747029287
5000	1.1230885954345356E12	100.46122624328846	7.689764618402449E10	55.80513534726263
5750	1.1230922921612407E12	100.4618875950475	7.689485290419463E10	55.803108245436974
6500	1.1230924830296074E12	100.46192174183699	7.689481316257419E10	55.80307940467763
7250	1.123098145342793E12	100.46293474531925	7.689958083882153E10	55.8065393391682
8000	1.1230978213358 31E12	100.46287677944028	7.690010360685307E10	55.80691871541965
8750	1.1230976726169417E12	100.46285017316224	7.690227467999382E10	55.808494277696006
9500	1.1230987352036008E12	100.46304027334216	7.69056469157421E10	55.81094153169784
10250	1.1230962394425605E12	100.4625937739286	7.690235239508691E10	55.80855067605803
11000	1.1230989376546355E12	100.46307649250554	7.690449457397295E10	55.81010526958079
11750	1.1230967454739705E12	100.46268430444174	7.690237594110751E10	55.80856776356242
12500	1.1230958756713054E12	100.46252869420236	7.69022143942804E10	55.808450527951194
13250	1.1230754741017566E12	100.4588788283366	7.688867558258072E10	55.79862532189048
14000	1.1230943578775469E12	100.4622571567508	7.690169066134062E10	55.80807045145939
14750	1.123093476330874E12	100.46209944581015	7.690182012732573E10	55.80816440578934
15500	1.123091416510742E12	100.46173093923517	7.690267903604356E10	55.808787721581545
16250	1.1230721886093914E12	100.45829105591304	7.689015915859042E10	55.79970196290783
17000	1.123092441940629E12	100.4619143909238	7.690274643517154E10	55.80883663357879
18500	1.1230727434893 82E12	100.45839032345782	7.689009758135883E10	55.79965727589966
19250	1.123101053876206E12	100.46345509197292	7.690860175104355E10	55.81308587540808

RBFNETWORK für Schach - alle Parameter(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.123052266133298E12	100.45472697486689	7.725082440342693E10	56.06143913955677
100	1.1188354388421782E12	99.7017700882379	9.151026854671191E10	66.40961297687336
150	1.1216610734547356E12	100.20600244911134	8.004801507873424E10	58.09137909186774
200	1.1217857367451606E12	100.22827781280677	7.699303181085443E10	55.87435733362474
250	1.122484624113301E12	100.35320380446589	7.667309770814699E10	55.64217901881562
300	1.1312289022163423E12	101.92281870658857	9.690178780021516E10	70.32227450293745
350	1.1213330854751355E12	100.14740799523739	8.052946448611067E10	58.44077014185132
400	1.122240376091931E12	100.30953567006674	7.663294564142015E10	55.61304039586218
450	1.1563146472289294E12	106.49335134768603	1.027780595765263E11	74.58672417191143
500	1.1002287152128945E12	96.41317650071713	7.896397849135231E10	57.30468650136294
1250	1.1133351546451301E12	98.72389596714697	7.886558001062408E10	57.233278066807436
2000	1.0660638168258923E12	90.51839548046162	8.162972898685008E10	59.23923943744735
2750	1.0537487077185679E12	88.43914812278605	8.245381995101558E10	59.8372877533042
3500	9.700520488637803E11	74.94808059939996	7.173155455074327E10	52.05606815056718
4250	9.859154787283594E11	77.41940168080046	7.191114080965535E10	52.186395097911095
5000	1.0740897507615111E12	91.88647379398624	6.668568571871089E10	48.39424744356873
5750	1.0997392592226377E12	96.32741340532506	6.965646798458508E10	50.55016397237399
6500	1.0512118431582688E12	88.01383215962171	6.438370593866542E10	46.72368534489701
7250	1.0604502626711371E12	89.56762296291494	6.238712708790557E10	45.27475474003172
8000	1.0790485450261858E12	92.73686440427362	6.865709176322309E10	49.824909974839684
8750	9.798287553214989E11	76.46642775521867	6.2350082374308876E10	45.24787114399528
9500	9.64218929389296E11	74.04943463551136	6.006353105505832E10	43.588505582351495
10250	1.8128274982785012E12	261.748017579137	7.959714508013399E10	57.76417972303799
11000	1.041428325512152E12	86.38318487843271	6.606224558064576E10	47.94181277213052
11750	9.340973034664994E11	69.49517891152044	6.1986009114530136E10	44.98366074814454
12500	9.546217528338179E11	72.58269562629923	6.497071668947990 4E10	47.149682966737174
13250	1.0525926421289542E12	88.24520174691276	6.492178829776248E10	47.11417530613101
14000	1.7914102986180806E12	255.5998377370499	7.854381924640222E10	56.99977413153913
14750	1.008913695644662E12	81.0734166116098	6.506556103465703E10	47.21851214138922
15500	1.765748752229116E12	248.32946714652633	7.824506000114441E10	56.78296255218389
16250	1.157303001860708E12	106.67547855614741	7.421874170090631E10	53.861036474521626
17000	1.7762460204508774E12	251.2908506361034	8.021734483758992E10	58.2142628286293
18500	1.021474880040697E12	83.10474525630067	6.186937480665753E10	44.899018452054115
19250	1.0162262625525961E12	82.25290952583704	6.5154694341336685E10	47.28319677726616

REPTREE für Schach - alle Parameter(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.121844114545438E12	100.23870985600112	7.570950529241911E10	54.94289356798916
100	1.115032995091975E12	99.02523421476306	7.53180217785386E10	54.658791367694995
150	1.086229930850277E12	93.97535427637665	7.325274846726772E10	53.16000873410503
200	1.1084333123746555E12	97.85647768399083	7.26928207854555E10	52.75366547629668
250	1.1086958385504912E12	97.90283668246673	7.231404200223459E10	52.47878318388257
300	1.0911367036763436E12	94.82629234976419	6.88383484532361E10	49.95644916518845
350	1.0965403983020988E12	95.76784467423039	6.89405368522126E10	50.03060796872526
400	1.0912547330163325E12	94.84680836760539	6.900353492409264E10	50.07632609018074
450	1.117312688027241E12	99.43056374071439	7.071578358273862E10	51.318916375912515
500	1.111754033055968E12	99.47108405587606	6.964942313500061E10	50.545051477986036
1250	1.036349908875807E12	85.54276181728534	6.056744756783025E10	43.95420116086357
2000	1.0458135688270267E12	87.11220064154166	5.979551168195427E10	43.39400213349473
2750	9.122144531864618E11	66.27722837153217	5.117096572923099E10	37.13511154211538
3500	1.0006611358537988E12	79.75253679838322	5.095670600957132E10	36.979621832762504
4250	9.975976953654591E11	79.26497280042062	4.865580892565688E10	35.309845453903165
5000	8.829899413188435E11	62.0986205848094	4.308193230828757E10	31.264845971125006
5750	8.42530328172219E11	56.538141240295495	4.209842044292396E10	30.551104842678644
6500	9.479513314051864E11	71.5718960248338	4.442360284111487E10	32.238505236282286
7250	9.375043002099005E11	70.00305242909451	4.29155291705457E10	31.14408610284931
8000	8.477372403070499E11	57.23912208170999	4.2019595773562225E10	30.493901253742933
8750	8.113610364384458E11	52.43227918711134	4.13931463820961E10	30.039282747016255
9500	8.709131974829404E11	60.411578563217624	4.014724306957336E10	29.13512239315384
10250	8.696633453629755E11	60.23830903289736	4.33124969915039E10	31.432168301413178
11000	9.032243033405741E11	64.97730134324505	4.792262094404151E10	34.777766039520095
11750	8.960071883107059E11	63.943061863837514	4.824509863856543E10	35.011790255896166
12500	8.360394904704419E11	55.67035931380539	4.804043966507731E10	34.8632678721524
13250	8.809358591505405E11	61.810039032216835	5.091715197669779E10	36.95091720697
14000	8.803275482303777E11	61.7247053737051	5.0870346527868935E10	36.91695018019478
14750	8.493287168603379E11	57.45423644378393	5.040329547926733E10	36.57800811532429
15500	8.294352533401682E11	54.79430476227649	4.986882571544443E10	36.19013983860514
16250	8.291354988833165E11	54.75470705030925	4.776158655330975E10	34.66090230680137
17000	8.143731045822101E11	52.82229728479611	4.65030937623902E10	33.74760568439564
18500	7.630804576356536E11	46.377902536495476	4.873663541223285E10	35.3685017749591
19250	7.595048762573066E11	45.94429307549111	4.74935981426722E10	34.4664213276133

SVR RBF für Schach - alle Parameter(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.6020395846618545E8	42.672617839436654	4.365266926437343E8	63.329531214137255
100	4.280539092959894E8	93.75709091263629	2.784490151890213 5E8	87.68910554396457
150	9.129452229069 72E8	105.20005987511165	5.416055797570984E8	104.48362679626212
200	7.948314959630525E8	119.46312539366065	4.272046472880 6585E8	112.29639496268209
250	7.022580358367159E8	114.94673981703116	3.528364774153042 4E8	110.12144509814996
300	6.490937948962753E8	113.43816949649121	3.094985866258314E8	107.21347298742462
350	6.005352100363278E8	113.05106400805809	2.762402334267376 7E8	107.18200445969683
400	5.622815668909987E8	110.30154314944609	2.420153763321077E8	100.39595109238053
450	5.348760827919909 4E8	110.65403412463517	2.179230738118418E8	97.36990993765879
500	5.139076512665077 4E8	109.51861821624729	2.151912595139120 8E8	100.19169045416932
1250	2.756532335610989E8	59.17769581388086	1.002217645281201 5E8	43.55092672971882
2000	2.664019773226778E8	60.32860340876559	9.097302380471233E7	40.643000541503014
2750	2.835106669954774 4E8	61.14594784113132	1.054247742517033E8	44.432524158503
3500	3.644440604840249E8	78.73509881316309	1.394149068884786E8	62.41795207912567
4250	3.449224038480164E8	80.21988130363326	1.413194385056251 3E8	65.9272734298103
5000	3.452114262563713E8	85.08677676617421	1.393311118841282 7E8	60.505599719139994
5750	3.835492385022861E8	102.72008529788312	1.686800612663217 5E8	72.89669450421542
6500	5.144790869480133E8	114.78160485752052	2.271522241627316 8E8	90.92688697564259

null für Schach (difficulty=1)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.5589735851484376E8	41.87769197998449	4.3204228478783315E8	62.67895141667458
100	5.247420496429714E8	140.89608405696956	3.6023403225138617E8	113.44482598789673
150	9.160947865286022E8	105.92716990802396	5.434193910869126E8	104.83353749354134
200	7.83765885830834E8	116.1599587867688	4.1723371906998086E8	109.67540453003915
250	7.017016246033596E8	114.76466338179434	3.545059478569036E8	110.64249240857636
300	6.493958282451469E8	113.5437631158922	3.1083719179699725E8	107.67717949711904
350	6.004252358886251E8	113.00966241886547	2.729999934926535E8	105.92478205310773
400	5.648240997322309E8	111.30132465830023	2.470390856260275E8	102.47995120930946
450	5.352304205968049E8	110.80069199965676	2.2441824070298076E8	100.27200655440494
500	5.185108960236513E8	111.48939598017675	2.214796854824512E8	103.11954184407594
1250	1.8900389112091506E8	27.82103989744576	8.044667921880785E7	34.957750432786256
2000	1.628218238432485E8	22.53583290399558	5.400003941299852E7	24.1249937543573
2750	1.903748607711147E8	27.570717702592134	7.228912691943304E7	30.467111749047955
3500	3.10775453867099E8	57.253236502092264	1.167134723755682E8	52.254210746301574
4250	2.84545652032976E8	54.59377359709683	1.1226912073596641E8	52.37493934841686
5000	3.0080399894604856E8	64.60393313986495	1.2029136434236094E8	52.237443899973776
5750	5.023566590823237E8	176.21279860679644	1.9827997161836547E8	85.68857758800408
6500	5.170840703084997E8	115.94690451099854	2.3031109398201752E8	92.19135268839389

null für Schach (difficulty=1)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	3.6087554584129643E8	26.239979657510016	3.3531917127995646E8	48.64675237993707
100	5.00413653511939171E8	128.13432622932697	3.209465870169539E8	101.07243196317297
150	9.089893948334626E8	104.29036291797757	5.361167507221677E8	103.42475150791776
200	7.787823660023915E8	114.68746536177576	4.21591023193023E8	110.8207795812701
250	7.144250572187394E8	118.96427977963691	3.6873707900344451E8	115.08407605298461
300	6.487120258859652E8	113.30476984814375	3.105116554480009E8	107.5644103793662
350	5.981440696455096E8	112.15258946478363	2.7105129095590454E8	105.16867986844953
400	5.591677226497692E8	109.08325387582406	2.4116494372622618E8	100.0431636306852
450	5.235143184823981E8	106.00296747707607	2.1462331405696133E8	95.89554880403219
500	5.0344007838559055E8	105.1025761478381	2.0792793819969347E8	96.80993395333748
1250	3.624723409242649E8	102.32496708220683	1.5043583563543814E8	65.37123035231275
2000	3.4190449068370754E8	99.37061862331993	1.3822008224680355E8	61.75104050254111
2750	3.8291160874376595E8	111.53879291502129	1.7358261244997218E8	73.15845517264498
3500	4.386173322910695E8	114.04551264146536	1.962326142563059E8	87.8560132942561
4250	4.116269214329087E8	114.24751136976079	1.8325668621938735E8	85.49152040216745
5000	3.9361376500226384E8	110.61965590877175	1.7068231164762673E8	74.12009771570065
5750	4.0310923225830626E8	113.46413619502206	1.814000959830265E8	78.39377861638354
6500	5.084549895144339E8	112.10935861657833	2.256033481353653E8	90.30688655082292

0.094 für Schach (difficulty=1)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	8.504039358281565E8	145.71327169102068	8.251546627558832E8	119.71010903138108
100	5.201121195042527E8	138.42073011351167	3.301990790579714E8	103.98622481885505
150	9.147019487369246E8	105.60530975213263	5.392270897291855E8	104.02478133066231
200	7.778596021136382E8	114.41584453796357	4.176977414284226E8	109.79737894760663
250	6.88928682282159E8	110.62461073177168	3.402065574013434E8	106.1796048054459
300	6.371395070942138E8	109.29828950236318	3.113711090951293E8	107.86213390497228
350	6.336931693510 69E8	125.87971604111155	3.6405184067611074E8	141.25316043528568
400	5.537320057416509E8	106.97274704255018	2.4339628015763873E8	100.96879549190764
450	5.227305094666027E8	105.6857884423497	2.2048989088966632E8	98.51678595827703
500	4.9448868617179656E8	101.3982610991411	2.0815536512194994E8	96.91582249103709
1250	1.5965941266557258E8	19.852761400178114	8.288987478513142E7	36.01943031436006
2000	2.223361490452978E8	42.021208858468675	6.728159274417323E7	30.058644815464202
2750	2.3089915802148184E8	40.55772373000713	7.512165996452025E7	31.66091646761544
3500	3.388025830625534E8	68.04560017223939	1.3003382852703092E8	58.21791556449557
4250	5.538300955858915E8	206.81989376243672	1.7959690370918325E8	83.78418640200964
5000	3.7105223953419745E8	98.30188872662904	1.4963542885061038E8	64.9803280789596
5750	3.964293724996681E8	109.73489988280265	1.8369220417808488E8	79.38433499638136
6500	6.024249808365651E8	157.37755216664632	2.690238140850311E8	107.68768840908072

0.11 für Schach (difficulty=1)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	8.268217982635157E7	1.3774391487416737	5.934366059022996E7	8.609338830918519
100	4.519642186771303E8	104.52382537670877	2.879944812679054E8	90.69516172229004
150	9.272253026700561E8	108.51682905753862	5.562608116301669E8	107.31083507267074
200	8.066491711994091E8	123.04192587855907	4.3588520605882E8	114.5781947099589
250	7.149564505783522E8	119.14131819166496	3.6072972161522084E8	112.58495302163571
300	6.52286325716932E8	114.55679202595344	3.1001730447491974E8	107.39316215081509
350	6.035327072437761E8	114.18244134429982	2.7271787368320084E8	105.81531875625261
400	5.645940688693678E8	111.21068572748408	2.402980142397153E8	99.68353271942915
450	5.3074268134247386E8	108.9504234956739	2.1464928915639746E8	95.90715470261134
500	5.0233762828016293E8	104.64276580407204	1.9798771273616248E8	92.18182779820638
1250	1.8366384004386923E8	26.27115634740342	4.4717313305042766E7	19.431711709200407
2000	1.4566359463776624E8	18.036423734376157	2.7917185445165854E7	12.472248758058127
2750	1.474054711841031E8	16.529361609450373	3.0607275107771292E7	12.899799883897053
3500	2.7006177304539984E8	43.23474010481853	7.027490560368367E7	31.463032097741227
4250	2.683016195609725E8	48.53843787651413	8.701676981040289E7	40.594403975366376
5000	2.8512775385166353E8	58.04579097705363	9.716106337413348E7	42.19293400665805
5750	3.009896607567888E8	63.25811940075883	1.2121096020538612E8	52.38247052035423
6500	4.2142204931415796E8	77.0142961026011	1.743004129721446E8	69.77080681707126

null für Schach (difficulty=1)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.3017746052091408E11	23.82527476215528	1.2967879618207347E11	48.67160953007753
100	1.8217835048325888E11	104.45352432348317	9.294833931127496E10	79.8598910876466
150	4.582495780538763E11	101.04699919395763	1.8092016278528168E11	101.74114158859588
200	3.99709156111386E11	108.09116427134167	1.4406805590594403E11	108.56362922685672
250	3.5574509325752734E11	106.47588644135573	1.1505688535288394E11	101.76607996424936
300	3.2593783217523126E11	105.43062778743737	9.729419333226653E10	97.54688096451295
350	3.010882563725136E11	104.95389513375808	8.343508023845737E10	93.1923851228618
400	2.822439464077869E11	104.231670098836	7.33448191203839E10	90.41645759094764
450	2.6598453899661514E11	103.93934265556082	6.494708236197182E10	88.103442338887
500	2.5435579683550912E11	102.19129283851582	6.3863044048299965E10	91.86118422948104
1250	1.6166039168436157E11	81.31902378514162	3.3598517856869495E10	39.099331492267005
2000	1.2952083753346353E11	75.01085918355425	2.4503643936887966E10	29.84914276682766
2750	1.4618356967140756E11	84.49563839368493	3.1707906031441586E10	44.74423491451813
3500	2.424055008947914E11	88.98107945526937	5.022032280064006E10	53.84204633877603
4250	2.2583834458834458E11	90.21493660359823	4.7997358897878586E10	58.436071665954344
5000	2.0893308941033087E11	88.22293511968061	4.314245349526509E10	44.28881596460061
5750	2.10217221494588E11	92.91543978040507	4.773619312641151E10	51.197104006615966
6500	2.5272914147721286E11	102.77853571712572	5.634383983826025E10	61.62073376506443
7250	2.479917431114918E11	103.00897635707076	5.664096878343022E10	63.14726288195456
8000	2.362306018182436E11	102.40366113087951	5.200744368303246E10	60.2050177368483
8750	2.269966101408315E11	102.35477447837754	4.8835321415396706E10	59.06423728763338
9500	2.5620035743880975E11	104.55343777686717	5.938598474126305E10	70.9050873987684
10250	3.4676950771849615E11	102.50311674110864	6.458493307620942E10	75.87566441206292

null für Schach (difficulty=2)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	1.1791876960351466E11	19.549344753331653	1.1732957494654788E11	44.036646130724726
100	2.0640290766583072E11	134.0791317807708	1.1227642384392197E11	96.46630640552395
150	4.585800344626198E11	101.19278729668497	1.812922299087049E11	101.95037495043842
200	3.9904261720714026E11	107.73096789464822	1.4341975297866797E11	108.07509540038909
250	3.557624708129756E11	106.4862890349398	1.1547721804061147E11	102.13785788766239
300	3.260445834993157E11	105.49970048230573	9.788689085196971E10	98.14111780868929
350	3.0118533591364197E11	105.02158637283718	8.38150914410335E10	93.61683668736276
400	2.8252115230811005E11	104.43651291840092	7.409878500009642E10	91.34591552411341
450	2.6610183291817548E11	104.03103325716745	6.595952047217861E10	89.47685711627699
500	2.563258767794427E11	103.7804422695502	6.5333229637578835E10	93.97591257168366
1250	1.0668052152016273E11	35.41238386211049	2.43190781329932E10	28.300644140284053
2000	8.412745851116846E10	31.6461141274401	1.5996255793776838E10	19.48585786477028
2750	1.0763141472669574E11	45.805250604547176	2.1863427222240692E10	30.85231559278504
3500	2.208742992654823E11	73.87595317176276	4.553178341132438E10	48.81538499964393
4250	2.1751069131255592E11	83.68436060157426	4.739022881875279E10	57.696899810897975
5000	2.1653810606848984E11	94.76232729790662	4.544324369517405E10	46.65074175883269
5750	3.067387332850414E11	197.82835657134092	8.040238899173943E10	86.23162430840472
6500	3.2748437860439435E11	172.57304365904574	8.352153816408868E10	91.34376502616604
7250	2.7620174921076373E11	127.77723271347965	7.139755440293405E10	79.598923426777
8000	2.2967704776243192E11	96.80067047626851	5.65184201115607E10	65.42702821568004
8750	2.118089388050739E11	89.11646091700749	4.731329578102774E10	57.2234122327196
9500	2.8811979716584863E11	132.2284951310205	6.93954287073914E10	82.85606374590363
10250	4.514263863223424E11	173.71173097716115	8.823895532691753E10	103.66488039177753

null für Schach (difficulty=2)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
500	2.5532177232744336E11	102.96895728230977	6.421693676037315E10	92.37022672981409
1250	1.8050146071781863E11	101.37861286430861	4.2965905760871635E10	50.00036606871774
2000	1.445810093534339E11	93.46894837969216	3.1627207003180706E10	38.52671951916426
2750	1.626375852200354E11	104.58731189924033	4.135927238035147E10	58.363645882037815
3500	2.6327401783249792E11	104.96118627681335	6.216870834801102 4E10	66.65210992337151
4250	2.441091877986266E11	105.40260299935676	5.733070210522582E10	69.79927841464063
5000	2.2557263547101672E11	102.83474549194322	5.0587590005872734E10	51.931781397380625
5750	2.227995873001644E11	104.37105363236545	5.281504786488545E10	56.64417963728825
6500	2.5583909799192532E11	105.32357993957035	5.9437291060063446E10	65.00390279829222
7250	2.5116451884832352E11	105.66160544963124	5.9907413857339874E10	66.78892138819077
8000	2.3942625966845798E11	105.19297387772102	5.512683394112363E10	63.81609593098383
8750	2.289793193221981E11	104.15062546377625	5.101367292805927E10	61.69886048475259
9500	2.5716735825552994E11	105.34417876136763	6.080278572350501E10	72.59670534031385
10250	3.4776854038262213E11	103.09458439270098	6.613166169315458E10	77.69279196621525
50	9.6598798340319 85E10	13.119273180693252	9.587867736428403E10	35.9856019975902
100	2.046568202061 61E11	131.8202136531027	1.0875732559052278E11	93.44274724001774
150	4.5814696968709454E11	101.00175264162759	1.8073654451434296E11	101.63788315562618
200	3.980845781008654E11	107.21429834799139	1.4381226553269955E11	108.37087635692946
250	3.5650546694813586E11	106.93153847004326	1.166371142299313E11	103.1637685751499
300	3.259857860020274E11	105.46165317623226	9.779865501639424E10	98.0526528113971
350	3.0108006678549426E11	104.94818572896096	8.371186625049347E10	93.50153983999719
400	2.81927121582848E11	103.99779690905555	7.340516426486787E10	90.4908485876464
450	2.6512535833865344E11	103.26893947470796	6.522364734565167E10	88.47861434363992
500	2.5532177232744336E11	102.96895728230977	6.421693676037315E10	92.37022672981409

0.263 für Schach (difficulty=2)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	7.564681020216418E10	8.045398629995832	4.153685262146577E10	15.589792097251362
100	1.656042635144044E11	86.31228102958235	9.120303399195425E10	78.36034958159696
150	4.58818940110452E11	101.29825122415413	1.8165512836423062E11	102.15445227702
200	3.99390167620862E11	107.91870848212581	1.4381201613592206E11	108.37068842200443
250	3.546099750129014E11	105.7974799190538	1.1253640785907686E11	99.53675563136679
300	3.2532569544300476E11	105.03498573586747	9.734631192154555E10	97.59913491359579
350	3.00421970594592E11	104.489898866573	8.422036105657492E10	94.06950050673214
400	2.807128261823449E11	103.10386314954818	7.218672303746425E10	88.98880466844632
450	2.6499505528629916E11	103.16745577364179	6.74749285880186E10	91.53257181042875
500	2.550872379653840 6E11	102.77987300090767	7.020779728385297E10	100.98753507208929
1250	1.5715474446333826E11	76.8492965656431	2.9247456281440887E10	34.03590578980955
2000	1.1954299830396744E11	63.89886446861944	1.863658789898662E10	22.702181533332976
2750	1.1516321051938083E11	52.44024273713096	2.392614352848185E10	33.763093203799215
3500	1.9250689632298898E11	56.11840413781894	4.169718266351045E10	44.7042455317815
4250	2.1206850220557938E11	79.5491279166881	4.3471882038623436E10	52.92637079610627
5000	1.9929810445149 75E11	80.27509377287753	4.2374658010549675E10	43.50061895293217
5750	2.1668669845128073E11	98.72242360864672	6.325307957228629E10	67.83897670736387
6500	2.6421240105126495E11	112.33062555257641	6.871757394508648E10	75.15333248863733
7250	3.4933528189105225E11	204.40218301370123	9.660436279535313E10	107.70121387412534
8000	2.500896890845712E11	114.77168146591075	7.26921138102556E10	84.15006951597687
8750	3.4447889669363214E11	235.71900833781177	8.779170409566862E10	106.1803197420431
9500	3.695053843181876E11	217.48053714277424	7.564751502750346E10	90.32086758576983
10250	3.402697771896678E11	98.69656103762141	6.589554595718199E10	77.41539850164477

0.093 für Schach (difficulty=2)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	9.447946379517841E9	0.1254992606169679	6.725900192766372E9	2.5243940995640712
100	1.8559793478697806E11	108.41162287888363	9.768331447760478E10	83.92811440274505
150	4.58749269368118E11	101.26748968443775	1.81521507104803E11	102.07930985361413
200	3.999230515044386E11	108.20688035060888	1.4385865887079352E11	108.40583642579244
250	3.5638610475650836E11	106.85994660248015	1.1593653703964023E11	102.5441186154832
300	3.262716169342667E11	105.64667607883234	9.797778705482605E10	98.23225008263573
350	3.012929921719408E11	105.09667802134356	8.364955015047238E10	93.43193619155672
400	2.824163227010734E11	104.35902486214292	7.351384730442166E10	90.62482854089356
450	2.65871237685136E11	103.85081152759273	6.489273631536421E10	88.02971964635782
500	2.530013016833982E11	101.10581287842359	6.021427002464842E10	86.61275444049541
1250	9.646577998766862E10	28.955526316138002	1.3276078012872612E10	15.44966290935278
2000	8.447337747911041E10	31.906896932275536	9.326851231938538E9	11.361514830371522
2750	8.642538975887025E10	29.533836045829982	1.1014027518699318E10	15.54231408920458
3500	1.7568085682537534E11	46.737083694483815	2.671615062594535E10	28.64283102492397
4250	1.6862062953029755E11	50.29264740972563	3.027531009967272E10	36.85974043816943
5000	1.6029649681263174E11	51.92961618145887	2.8007880070310574E10	28.75209325145888
5750	1.7178950648093448E11	62.050397463700776	3.807867219343583E10	40.83940534509009
6500	2.358242763421514E11	89.48882437137952	4.9891154723007E10	54.563721096098064
7250	2.2989723241710416E11	88.52544431768614	5.039334595454441E10	56.181981573447594
8000	2.2003238446958832E11	88.84160205593564	4.6417505448932495E10	53.73397615705089
8750	2.0648149247503955E11	84.68990071138789	4.203217528460646E10	50.83612235513384
9500	2.3342674300273874E11	86.79207577586806	5.023852021300389E10	59.98328868058651
10250	3.204076725252959E11	87.51068462165567	5.539139946763115E10	65.07491820671518

null für Schach (difficulty=2)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	7.485878321623315E11	8.739739045720459	7.45622644353458E11	29.457066721707058
100	1.5935336319653877E12	115.51059210854207	7.461329188589939E11	72.21860876133267
150	6.018836244395211E12	108.6555238687944	1.7030726522776406E12	71.01012752326157
200	5.241961098919054E12	97.50886524231562	1.3340675974317095E12	43.02741507120902
250	4.6745175721851E12	97.85505143672111	1.0662041922602778E12	41.47078099827551
300	4.2792515524982534E12	94.02528629185689	8.942287601656273E11	35.844019462172504
350	5.675398437322109E12	107.23936160114853	1.7372876282898352E12	71.98030572694286
400	5.759408480223702E12	110.26336630029851	1.978532724094946E12	86.67227227976329
450	5.4245734741875 29E12	109.22608871159973	1.7741289899064001E12	86.15830761579501
500	5.1548090378275E12	108.25510264006462	1.6022619001585273E12	84.33784811824671
600	7.746323506240054E12	101.40411988995434	2.412846988580506E12	99.94128908264175
750	6.921439426632404E12	101.20193856203481	1.9278994284190852E12	98.52302458038932
900	6.450505185155664E12	100.19730213898302	1.8396697417261235E12	100.08711735115905
1050	5.969095850792642E12	100.23642228538507	1.5880702764361697E12	100.08321467546098
1200	5.5873762421907 22E12	100.2497759957166	1.3915202744436846E12	99.80047119816251
1350	5.26528802287556E12	100.30501326264616	1.2400755161814456E12	99.61290383236273
1500	4.997984200694071E12	100.31883941329913	1.1203233329302961E12	99.52130996676416
1650	4.763012016544502E12	100.48025388881912	1.017213616525886E12	98.50129533876736

null für Schach (difficulty=3)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	5.933051149069036E11	5.489959278002965	5.892130080079901E11	23.277843050540913
100	1.690516815841686E12	129.9984974523338	8.276929522506556E11	80.11284850494435
150	6.008717450854684E12	108.29049030855988	1.6856281603189646E12	70.28277417347498
200	5.230284803116465E12	97.07495346781958	1.3158898975509363E12	42.44113335706227
250	4.671607530557627E12	97.73325335519428	1.0611903510203546E12	41.27576402729532
300	4.273068154108475E12	93.7537548119153	8.907690403248488E11	35.70534100445486
350	5.750926671769267E12	110.11263807465896	1.7831026319775244E12	73.87853945555531
400	5.760403503198084E12	110.30146884519927	1.978691947052139E12	86.6792472543649
450	5.423488723156687E12	109.18240923472852	1.7723231098863826E12	86.07060735996969
500	5.153548550711781E12	108.20216664272235	1.6006554363628713E12	84.25328909603438
600	7.749886379414234E12	101.49742172656401	2.4152220911262793E12	100.0396669786507
750	6.924484296883209E12	101.29099938675878	1.9299595380277092E12	98.62830405016706
900	6.451547595329075E12	100.2296887888534	1.8429309753177153E12	100.26454456093907
1050	5.969523235531082E12	100.25077657013344	1.5905520946637483E12	100.23962358893404
1200	1.243212840634191E13	496.3163148290573	2.579750621088439E12	185.02089569719607
1350	9.128825356784883E12	301.51447815406766	1.8747933338051606E12	150.59857696485858
1500	5.271458954023527E12	111.59748307791972	1.264477039105519E12	112.32686819574269
1650	5.642038984898762E12	140.99039742647633	1.511207879814927E12	146.33694562240902

null für Schach (difficulty=3)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	5.147248785121618E11	4.13202801895578	5.100026568464978E11	20.148505956390323
100	1.6869010865036667E12	129.44300257716029	8.198042676667443E11	79.34929845748327
150	6.018846881027588E12	108.65590790645135	1.7030415131021306E12	71.0088291659537
200	5.23887655034716E12	97.3941439258266	1.3299786671161257E12	42.895535620554064
250	4.673165626938199E12	97.79845712992262	1.0618299185970173E12	41.300640469442065
300	4.273067298045428E12	93.75371724681497	8.907659320016423E11	35.705216411283104
350	5.751839168958433E12	110.14758390159425	1.7828530179375176E12	73.86819730229246
400	5.760546923982195E12	110.30696141879292	1.9786363921471301E12	86.67681359744593
450	5.423651361591609E12	109.18895761099179	1.7723140800803103E12	86.07016883897505
500	5.153866115746127E12	108.21550203021646	1.6024471405277546E12	84.34759856798513
600	7.744741762917156E12	101.36271213990594	2.4128209051038604E12	99.94020869242553
750	6.924460470611145E12	101.2903023290076	1.9297832982050576E12	98.61929752206544
900	6.451525008427662E12	100.2289869807669	1.8424830694846228E12	100.24017627207601
1050	5.969506735066033E12	100.25022236100031	1.5905176019940679E12	100.23744979516962
1200	5.587359006925577E12	100.24915751980585	1.3929533915699797E12	99.90325501462081
1350	5.264805620169752E12	100.28663432763625	1.240992153480862E12	99.68653556040225
1500	4.99734025977994E12	100.29299089475055	1.1206783255485652E12	99.55284490793713
1650	4.762231051750464E12	100.44730620477466	1.0180548747755817E12	98.58275809738876

0.063 für Schach (difficulty=3)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	5.933051149068993E11	5.489959278002887	5.892130080079855E11	23.277843050540728
100	1.5026791390482021E12	102.71452382613171	6.981481470626445E11	67.5741367466657
150	6.019426431821833E12	108.67683372520669	1.6975946053564248E12	70.78171870586104
200	5.327957545444974E12	100.73445154085677	1.701705903296883E12	54.88477973023386
250	4.73973835985392E12	100.60472853380385	2.035217223202273E12	79.16124168337161
300	4.279181913964947E12	94.02222606991248	9.125143869556534E11	36.576975492817645
350	4.959815445154509E12	81.90162917645202	1.5662002251322432E12	64.89171349571006
400	5.714880848696076E12	108.56500190789653	1.9433428765931196E12	85.1307339432888
450	5.423566896769157E12	109.18555674560073	1.7698666591957747E12	85.95131297075191
500	5.153472520218313E12	108.19897404509409	1.5965360131963542E12	84.03645607683937
600	7.684936839329988E12	99.80330988025146	2.3637460965117124E12	97.90750638871725
750	6.846630805169533E12	99.02612985076186	1.8902754336052434E12	96.60029473711307
900	6.446325841659688E12	100.06750663551558	1.8222410942757888E12	99.13891287561178
1050	5.962982991088524E12	100.03122623637988	1.5729270043013892E12	99.12885681205752
1200	5.577221357613084E12	99.88570536735817	1.3794729863372998E12	98.93643417925676
1350	5.261915040871183E12	100.17654217584604	1.230681668987972E12	98.85831398288256
1500	4.994409907928984E12	100.17540531154552	1.1118758131116238E12	98.77089424871825
1650	4.762414422114876E12	100.45504182850105	1.0239998012630897E12	99.15843163360553

0.109 für Schach (difficulty=3)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	4.652183980966257E10	0.03375411310336279	2.823157879010448E10	1.1153356277160127
100	1.5629060213642925E12	111.11305057511842	7.824863885174916E11	75.73728075130312
150	6.015361365847038E12	108.53009903806961	1.6958981629285461E12	70.71098502754236
200	5.230888572197178E12	97.09736686997893	1.309398406387851E12	42.23176459250876
250	4.672442715293571E12	97.76820176098566	1.0626712822733069E12	41.333365916417975
300	4.27942768466793E12	94.03302653254862	8.936060389717972E11	35.81905847725964
350	4.876758618941939E12	79.18155526288076	1.4126344328272974E12	58.52908677845693
400	5.688066539288563E12	107.54861462837817	1.923444874119647E12	84.25907533122661
450	5.412817385937995E12	108.75317406500562	1.7619627309802112E12	85.5674687956981
500	5.138859521022598E12	107.58623386132473	1.5879340717844275E12	83.58367789603798
600	7.734215214933323E12	101.0873577484597	2.3987907651517095E12	99.35907351084788
750	6.88810689673846E12	100.22954141686004	1.901073715562471E12	97.15212819015285
900	6.433033897062626E12	99.65526546082661	1.8243095426783887E12	99.25144668171298
1050	5.962673608669452E12	100.02084649860295	1.5800608829347673E12	99.57844743617986
1200	5.579626386674561E12	99.97187007369334	1.380911943709434E12	99.03963686082064
1350	5.25992168933918E12	100.10065754490248	1.2340911515892415E12	99.13219122507196
1500	4.992715388220971E12	100.10744116556785	1.1145218953389282E12	99.00595279101631
1650	4.752618523741O3E12	100.04221117533835	1.0100572363082573E12	97.80831137756103

null für Schach (difficulty=3)(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	2.678945201672587E8	95.49831595308318	1.1820133218509118E8	90.00205828296912
100	2.1063699077649432E8	94.1826587667439	8.89859356620989E7	88.98763010087944
150	1.6380458774087733E8	87.04354524555627	5.9822492369725265E7	79.86636841449393
200	1.426485078593518E8	85.6877732315407	5.475548736497498E7	81.41493495708491
250	1.2635448935997719E8	83.02879330999392	4.7782335910142235E7	80.79980164040688
300	1.144424832940342E8	78.73160688514828	4.493681265805998E7	73.5308365807729
350	1.0650322111919111E8	72.14774839403985	4.4981080587357804E7	63.75586609294012
400	1.0002201488396306E8	65.15540498687375	4.3487003602168016E7	56.250359837054894
450	9.516710392495522E7	58.32712438256149	4.2306271742034756E7	49.49040256953232
500	9.111543902463166E7	58.43452060745125	4.0183580225492954E7	49.559379478222475
600	9.396888381823973E7	65.46379033279236	4.5673666789688155E7	55.717621501960835
700	9.401725414750989E7	64.87171612169188	5.103431628500964E7	62.19358906175436
800	8.865604028904667E7	55.206703372096925	4.8063806186730884E7	54.599363921769914

KNN Regression für Text-To-Speech(dynamic): Rohdaten.

samplesize	RMSE	RRSE	MAE	RAE
50	6.502277664984411E7	5.62599154496447	4.3694842769418225E7	33.270570753299644
100	1.5091596128882936E8	48.34727620651643	6.069758075660683E7	60.69873653851724
150	1.0862092763102245E8	38.274670796987074	4.707761137110927E7	62.85124047660352
200	1.1208510717286849E8	52.90298939181335	4.657708459379809E7	69.2546171202625
250	7.938249618745843E7	32.77154184527595	3.808432046898439E7	64.40048358645566
300	9.699657219377278E7	56.55712738292219	4.203729833465278E7	68.78631418884433
350	1.0342026568692367E8	68.03127252315338	4.508215409045368E7	63.89912696272758
400	1.0001076538587658E8	65.14074972550607	4.580610455990773E7	59.250112695741485
450	9.678206569281882E7	60.32351418161939	4.611684900950967 5E7	53.94806322418806
500	9.448631496298613E7	62.838146193905445	4.646396016616707E7	57.305123660315616
600	1.0046540106097192E8	74.8283304245949	5.570183594174848E7	67.95105429694469
700	9.650069648781571E7	68.3441200841104	5.5121558773492195E7	67.17455673661851
800	9.061629376917878E7	57.67501881520566	5.047997945661890 5E7	57.344080458536595

Linear Regression für Text-To-Speech(dynamic): Rohdaten.

C. Offloadingentscheidung: Rauschreduzierung

C.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

C.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	798	0	798	5	0	5	446.74585896056857	0.0	-446.74585896056857
RBFNETWORK	676	0	676	127	0	127	203.11789000513798	0.0	-203.11789000513798
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	710	0	710	93	0	93	570.9859111459085	0.0	-570.9859111459085

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	803	0	803	0	0	0	0.0	0.0	0.0
RBFNETWORK	803	0	803	0	0	0	0.0	0.0	0.0
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	803	0	803	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	803	0	803	0	0	0	0.0	0.0	0.0
Linear Regression	608	0	608	195	0	195	614.1369825104084	0.0	-614.1369825104084
RBFNETWORK	540	0	540	263	0	263	244.9794405288833	0.0	-244.9794405288833
REPTREE	803	0	803	0	0	0	0.0	0.0	0.0
SVR RBF	516	0	516	287	0	287	750.9487961240133	0.0	-750.9487961240133

Offloadingentscheidung für Rauschreduzierung mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

D. Offloadingentscheidung: Gesichtserkennung

D.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

D.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

E. Offloadingentscheidung: Text-To-Speech

E.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

E.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
Linear Regression	6937	0	6937	0	0	0	0.0	0.0	0.0
RBFNETWORK	6937	0	6937	0	0	0	0.0	0.0	0.0
REPTREE	6937	0	6937	0	0	0	0.0	0.0	0.0
SVR RBF	6937	0	6937	0	0	0	0.0	0.0	0.0

Offloadingentscheidung für Gesichtserkennung mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

F. Offloadingentscheidung: Schach(alle Parameter)

F.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5893	4811	1082	682	318	364	73.26529596260072	2.019476136454829E7	2.0194688099252325E7
Linear Regression	6115	4951	1164	460	178	282	42.75696481951488	2.0194836281457637E7	2.0194793524492815E7
RBFNETWORK	5129	5129	0	1446	0	1446	407.92197763387236	2.0194919943302747E7	2.019451202132511E7
REPTREE	6173	4978	1195	402	151	251	41.24979235921068	2.019479094378543E7	2.019474969399307E7
SVR RBF	6271	4990	1281	304	139	165	21.528751548499834	2.019486426926111E7	2.019484274050956 2E7

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5762	4299	1463	813	368	445	159.10819872807016	2.0192856936152898E7	2.019269782795417E7
Linear Regression	5945	4420	1525	630	247	383	114.60736768811009	2.0193106466977723E7	2.0192991859610036E7
RBFNETWORK	4667	4667	0	1908	0	1908	958.6837071198014	2.019333518825285E7	2.019237650454573 3E7
REPTREE	6071	4489	1582	504	178	326	93.74282328239964	2.0192384519799788E7	2.0192290776976503E7
SVR RBF	6221	4490	1731	354	177	177	37.514808301134956	2.019323333315097E7	2.019319581834267E7

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5752	3661	2091	823	404	419	362.48483413533916	2.0185397209339947E7	2.0185034724505812E7
Linear Regression	5990	3850	2140	585	215	370	276.96368644170354	2.0189200702738337E7	2.0188923739051897E7
RBFNETWORK	4065	4065	0	2510	0	2510	2919.5308108907666	2.0189568565457545E7	2.0186649034646653E7
REPTREE	6051	3827	2224	524	238	286	198.99187608546092	2.018707110811931E7	2.0186872116243225E7
SVR RBF	6180	3889	2291	395	176	219	127.73233269942081	2.0189370461690698E7	2.0189242729358E7

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5958	5220	738	617	245	372	44.420981574024005	2.0195786121547315E7	2.019574170056574E7
Linear Regression	6215	5301	914	360	164	196	16.802036884023984	2.01958134850871E7	2.0195796683050215E7
RBFNETWORK	5465	5465	0	1110	0	1110	177.44055529346423	2.0195862752049852E7	2.019568531149456E7
REPTREE	6230	5330	900	345	135	210	19.398678210959982	2.019580716397282E7	2.019578776529461E7
SVR RBF	6339	5356	983	236	109	127	8.637370300455993	2.0195844407002255E7	2.0195835769631956E7

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6048	5470	578	527	181	346	28.90698500753777	2.0196251323469773E7	2.0196222416484766E7
Linear Regression	6265	5502	763	310	149	161	9.332339312820372	2.019625939061355E7	2.0196250058274236E7
RBFNETWORK	5651	5651	0	924	0	924	98.94032157855447	2.019629396745458E7	2.0196195027133E7
REPTREE	6271	5518	753	304	133	171	11.96144862023212	2.0196257559038077E7	2.0196245597589456E7
SVR RBF	6370	5557	813	205	94	111	4.921729823945625	2.0196278833697144E7	2.019627391196732E7

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

F.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6312	6224	88	263	48	215	0.007758354346562495	45543.68883286229	45543.68107450794
Linear Regression	6264	6054	210	311	218	93	0.0029310358621874983	45543.658736365796	45543.65580532994
RBFNETWORK	6272	6272	0	303	0	303	0.0129595675490625	45543.6921952854	45543.67923571786
REPTREE	6341	6235	106	234	37	197	0.007595971109687495	45543.68425852276	45543.67666255165
SVR RBF	6402	6217	185	173	55	118	0.0036265912906249978	45543.676760055416	45543.67313346412

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6144	5812	332	431	101	330	0.03386644798874999	45542.8597729036	45542.82590645561
Linear Regression	6291	5726	565	284	187	97	0.008965623671874999	45542.85134332201	45542.84237769833
RBFNETWORK	5913	5913	0	662	0	662	0.07984422363125016	45542.896271713886	45542.81642749025
REPTREE	6294	5767	527	281	146	135	0.012811857193125	45542.843375421675	45542.83056356448
SVR RBF	6409	5827	582	166	86	80	0.00577124789	45542.870632138525	45542.86486089064

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5796	4536	1260	779	370	409	0.23329102059093731	45536.5250993471	45536.29180832651
Linear Regression	5991	4678	1313	584	228	356	0.16771566201375004	45536.860517072746	45536.69280141073
RBFNETWORK	4906	4906	0	1669	0	1669	1.3992969769321875	45537.212932108945	45535.81363513201
REPTREE	6102	4716	1386	473	190	283	0.12717920197781254	45536.64586217225	45536.51868297027
SVR RBF	6238	4758	1480	337	148	189	0.0686648567446875	45537.08210330786	45537.01343845112

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	5935	5173	762	640	262	378	0.1054043645990625	45540.5342311724	45540.4288268078
Linear Regression	6189	5265	924	386	170	216	0.04366988607374998	45540.594991358856	45540.55132147278
RBFNETWORK	5435	5435	0	1140	0	1140	0.43480323122812525	45540.71296930806	45540.27816607683
REPTREE	6215	5302	913	360	133	227	0.04973851840046873	45540.585495136336	45540.53575661794
SVR RBF	6342	5329	1013	233	106	127	0.020182936184843743	45540.67211958592	45540.65193664974

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	6502	6499	3	73	0	73	0.0012108607626562498	45544.115981193114	45544.11477033235
Linear Regression	6419	6383	36	156	116	40	5.219190862499999E-4	45544.100572763564	45544.10005084448
RBFNETWORK	6499	6499	0	76	0	76	0.0013282998768749993	45544.115981193114	45544.11465289324
REPTREE	6510	6487	23	65	12	53	8.027224645312499E-4	45544.11480060876	45544.1139978863
SVR RBF	6523	6483	40	52	16	36	5.15261480625E-4	45544.11186667306	45544.11135141158

Offloadingentscheidung für Schach(alle Parameter) mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

G. Offloadingentscheidung: Schach (difficulty = 1)

G.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1591	945	646	723	671	52	7.772708082165812	16888.553080307793	16880.78037222563
Linear Regression	1849	1385	464	465	231	234	46.38126294385416	21894.312297414537	21847.931034470683
RBFNETWORK	1511	993	518	803	623	180	47.720108933927804	16793.460172450024	16745.740063516096
REPTREE	1780	1392	388	534	224	310	64.27935089517035	20268.21516607569	20203.93581518052
SVR RBF	1990	1592	398	324	24	300	56.85675624972771	22306.591930852537	22249.73517460281

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1537	744	793	777	723	54	18.947040625184634	14509.650913314956	14490.703872689772
Linear Regression	1795	1204	591	519	263	256	109.78768237017604	21013.720454003884	20903.932771633707
RBFNETWORK	1579	947	632	735	520	215	110.89854504686465	16262.855585917056	16151.957040870191
REPTREE	1719	1096	623	595	371	224	90.710473340394	16838.821894257508	16748.111420917114
SVR RBF	1960	1391	569	354	76	278	113.69783162846666	21768.342977182125	21654.64514555366

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1422	433	989	892	820	72	59.672632773842246	9789.841054479659	9730.168421705817
Linear Regression	1885	1011	874	429	242	187	204.64726562090777	19486.11191919043	19281.46465356952
RBFNETWORK	1571	821	750	743	432	311	347.8189656379196	14903.483527403234	14555.664561765314
REPTREE	1525	652	873	789	601	188	163.03737787786983	11292.400734860496	11129.363356982625
SVR RBF	2009	1109	900	305	144	161	149.45489498753503	20220.86293896401	20071.408043976473

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1597	1126	471	717	689	28	2.205078637503998	19183.536225532804	19181.3311468953
Linear Regression	1955	1657	298	359	158	201	19.727150427359998	22445.706951403994	22425.979800976635
RBFNETWORK	1815	1815	0	499	0	499	65.94394167315997	22624.85448764252	22558.91054596936
REPTREE	1883	1612	271	431	203	228	23.679824146295985	20766.44039731045	20742.760573164152
SVR RBF	2084	1789	295	230	26	204	19.570498378871992	22616.06223073792	22596.49173235905

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1548	1222	326	766	728	38	2.3175819729982496	19533.761115837413	19531.443533864414
Linear Regression	2047	1847	200	267	103	164	11.520790247557496	22710.03636775335	22698.515577505794
RBFNETWORK	1950	1950	0	364	0	364	32.18011473875251	22770.479556118305	22738.299441379553
REPTREE	2018	1823	195	296	127	169	12.747389669415375	22196.037659568225	22183.29026989881
SVR RBF	2133	1924	209	181	26	155	10.562264546842124	22764.0350091881	22753.472744641258

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

G.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2074	2034	40	240	208	32	8.263663299999996E-4	50.90483657464314	50.90401020831314
Linear Regression	2242	2220	22	72	22	50	0.0016395790768749998	51.99570599485875	51.99406641578188
RBFNETWORK	2242	2242	0	72	0	72	0.002422109047499999	51.99746485950813	51.99504275046063
REPTREE	2239	2219	20	75	23	52	0.0016830751887499994	51.98365635611469	51.98197328092594
SVR RBF	2253	2230	23	61	12	49	0.0013819817909375	51.98843958199939	51.98705760020845

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1611	1435	176	703	679	24	0.0016461321449999994	45.846419438219456	45.84477330607446
Linear Regression	2167	2063	104	147	51	96	0.00846190886999998	51.693544148562005	51.68508223969201
RBFNETWORK	2114	2114	0	200	0	200	0.02126217036499998	51.71264891496638	51.69138674460138
REPTREE	2088	1976	112	226	138	88	0.008115567084999999	50.40932063789513	50.40120507081013
SVR RBF	2171	2071	100	143	43	100	0.0091999742575	51.694823828947015	51.68562385468952

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1566	843	723	748	701	47	0.027200906976562494	35.0908028888372	35.06360198186064
Linear Regression	1816	1290	526	498	254	244	0.16207925422875	48.323179977755636	48.161100723526886
RBFNETWORK	1552	972	580	762	572	190	0.163964976793125	37.07975513491874	36.91579015812562
REPTREE	1659	1237	422	655	307	348	0.23884108763624995	41.555840606132854	41.3169995184966
SVR RBF	1967	1491	476	347	53	294	0.18456877220062493	49.77240387418967	49.58783510198904

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	1590	1106	484	724	695	29	0.005099809779531249	42.50161227148721	42.49651246170768
Linear Regression	1949	1631	318	365	170	195	0.04551079783546872	50.541119354530124	50.495608556694656
RBFNETWORK	1456	1081	375	858	720	138	0.04740917747156251	40.35912264610611	40.311713468634544
REPTREE	1878	1602	276	436	199	237	0.06058293482203123	46.77795618175973	46.7173732469377
SVR RBF	2078	1774	304	236	27	209	0.04971495540765624	50.943140903442796	50.89342594803514

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2308	2308	0	6	2	4	6.311298562499987E-5	52.14760384541449	52.147540732428865
Linear Regression	2308	2306	2	6	4	2	2.7331442812499897E-5	52.147915394517305	52.14788806307449
RBFNETWORK	2310	2310	0	4	0	4	6.311298562499987E-5	52.14834621989168	52.14828310690606
REPTREE	2310	2310	0	4	0	4	6.311298562499987E-5	52.14834621989168	52.14828310690606
SVR RBF	2310	2308	2	4	2	2	3.5781542812499974E-5	52.14104263481449	52.141006853271676

Offloadingentscheidung für Schach(difficulty = 1) mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

H. Offloadingentscheidung: Schach (difficulty = 2)

H.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2817	2707	110	733	710	23	3.61316677075011	9052684.224986747	9052680.611819977
Linear Regression	3397	3332	65	153	85	68	8.464739309818066	9076904.10955983	9076895.64482052
RBFNETWORK	3417	3417	0	133	0	133	16.192333487107337	9076947.435034623	9076931.242701136
REPTREE	3341	3323	18	209	94	115	14.07219802439055	8865925.089529738	8865911.017331714
SVR RBF	3454	3373	81	96	44	52	6.3163817442698145	9076934.42035734	9076928.103975594

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2392	2128	264	1158	1128	30	9.239082742450126	8725270.590552883	8725261.35147014
Linear Regression	3335	3173	162	215	83	132	33.69231439902375	9075787.594296256	9075753.901981857
RBFNETWORK	3256	3256	0	294	0	294	85.55917485820164	9075863.785211885	9075778.226037027
REPTREE	3182	2997	185	368	259	109	27.831851747187212	8860958.509388953	8860930.677537207
SVR RBF	3384	3221	163	166	35	131	33.89530848403784	9075823.542935992	9075789.647627508

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2304	1820	484	1246	1237	9	7.642101998562284	8342471.4450214915	8342463.802919493
Linear Regression	3329	2976	353	221	81	140	83.37432095689233	9072985.49398068	9072902.119659722
RBFNETWORK	3057	3057	0	493	0	493	431.3168052719051	9073117.144645909	9072685.827840637
REPTREE	3064	2750	314	486	307	179	129.76831941148356	8847334.943819182	8847205.17549977
SVR RBF	3337	3003	334	213	54	159	94.36137838853892	9073059.210919397	9072964.849541008

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3468	3463	5	82	60	22	1.3055634199600001	9077202.243926628	9077200.938363208
Linear Regression	3460	3451	9	90	72	18	1.111014242328	9077535.508162703	9077534.39714846
RBFNETWORK	3523	3523	0	27	0	27	1.7776539128159998	9077566.507823028	9077564.730169116
REPTREE	3481	3477	4	69	46	23	1.5721661730799992	8883740.390893377	8883738.818727205
SVR RBF	3470	3455	15	80	68	12	0.8640489695999996	9077549.330549791	9077548.466500822

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3533	3533	0	17	6	11	0.5299424509481253	9077825.122379342	9077824.592436891
Linear Regression	3490	3488	2	60	51	9	0.4220949703713752	9077825.935038125	9077825.512943154
RBFNETWORK	3539	3539	0	11	0	11	0.5299424509481253	9077840.467794668	9077839.937852217
REPTREE	3497	3497	0	53	42	11	0.5299424509481253	8884019.364995552	8884018.835053101
SVR RBF	3499	3495	4	51	44	7	0.26173702752262507	9077830.842343688	9077830.58060666

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

H.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3550	3550	0	0	0	0	0.0	20471.280460106387	20471.280460106387
Linear Regression	3550	3550	0	0	0	0	0.0	20471.280460106387	20471.280460106387
RBFNETWORK	3550	3550	0	0	0	0	0.0	20471.280460106387	20471.280460106387
REPTREE	3550	3550	0	0	0	0	0.0	20471.280460106387	20471.280460106387
SVR RBF	3550	3550	0	0	0	0	0.0	20471.280460106387	20471.280460106387

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3548	3548	0	2	0	2	9.896411375000014E-5	20470.81470824042	20470.81460927631
Linear Regression	3529	3528	1	21	20	1	7.521931187500001E-5	20470.804860970213	20470.8047857509
RBFNETWORK	3548	3548	0	2	0	2	9.896411375000014E-5	20470.81470824042	20470.81460927631
REPTREE	3514	3514	0	36	34	2	9.896411375000014E-5	20033.8342487283	20033.83414976419
SVR RBF	3543	3543	0	7	5	2	9.896411375000014E-5	20470.81132138541	20470.811222421296

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	2565	2406	159	985	951	34	0.015054687076875004	20307.850273030424	20307.83521834335
Linear Regression	3381	3275	106	169	82	87	0.039549062624062506	20467.01373039788	20466.974181335256
RBFNETWORK	3357	3357	0	193	0	193	0.08775930840843751	20467.121393242913	20467.033633934505
REPTREE	3246	3147	99	304	210	94	0.04157342023312503	19985.59409344208	19985.552520021847
SVR RBF	3422	3316	106	128	41	87	0.03529459909406252	20467.05742687828	20467.022132279188

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3452	3443	9	98	74	24	0.003301097373750001	20468.338279966676	20468.3349788693
Linear Regression	3454	3444	10	96	73	23	0.003125994248593749	20469.3803446791	20469.37721868485
RBFNETWORK	3517	3517	0	33	0	33	0.00493404365015625	20469.449075180302	20469.444141136653
REPTREE	3465	3461	4	85	56	29	0.004347274239531248	20032.36916945414	20032.3648221799
SVR RBF	3467	3450	17	83	67	16	0.0023570461324999973	20469.412251558508	20469.409894512377

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	3550	3550	0	0	0	0	0.0	20471.51555226942	20471.51555226942
Linear Regression	3550	3550	0	0	0	0	0.0	20471.51555226942	20471.51555226942
RBFNETWORK	3550	3550	0	0	0	0	0.0	20471.51555226942	20471.51555226942
REPTREE	3550	3550	0	0	0	0	0.0	20471.51555226942	20471.51555226942
SVR RBF	3550	3550	0	0	0	0	0.0	20471.51555226942	20471.51555226942

Offloadingentscheidung für Schach(difficulty = 2) mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

I. Offloadingentscheidung: Schach (difficulty = 3)

I.1. Energieszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	562	562	0	27	26	1	0.23448694712026544	2.195348886529664E7	2.195348863080969E7
Linear Regression	587	587	0	2	1	1	0.23448694712026544	2.2189530325378187E7	2.218953009089124E7
RBFNETWORK	588	588	0	1	0	1	0.23448694712026544	2.218953129191093E7	2.2189531057423983E7
REPTREE	569	569	0	20	19	1	0.23448694712026544	2.2189292213424172E7	2.2189291978937224E7
SVR RBF	588	588	0	1	0	1	0.23448694712026544	2.218953129191093E7	2.2189531057423983E7

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 1(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	432	430	2	157	150	7	1.2244377809255622	1.7979796818965845E7	1.7979795594528064E7
Linear Regression	578	577	1	11	3	8	1.2275089927035001	2.2189339091966547E7	2.2189337864457555E7
RBFNETWORK	580	580	0	9	0	9	1.3657383210174372	2.218934111983409E7	2.218933975409577E7
REPTREE	555	554	1	34	26	8	1.3258312069034985	2.21885071198516E7	2.2188505794020392E7
SVR RBF	582	579	3	7	1	6	0.8953240043156244	2.2189338489076823E7	2.218933759375282E7

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 2(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	365	331	34	224	217	7	4.413885837070216	1.5382953921201158E7	1.538294950731532E7
Linear Regression	556	535	21	33	13	20	9.557751505808623	2.2188819266195413E7	2.2188809708443906E7
RBFNETWORK	548	548	0	41	0	41	24.07968539877927	2.2188850757009983E7	2.2188826677324586E7
REPTREE	523	500	23	66	48	18	12.279106747812557	2.2187860859935634E7	2.2187848580828886E7
SVR RBF	548	525	23	41	23	18	8.523844620036561	2.2188770232892442E7	2.2188761709047824E7

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 3(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	586	586	0	3	2	1	0.05603977304799976	2.2189618027100354E7	2.218961797106058E7
Linear Regression	588	588	0	1	0	1	0.05603977304799976	2.2189636218849283E7	2.2189636162809051E7
RBFNETWORK	588	588	0	1	0	1	0.05603977304799976	2.2189636218849283E7	2.2189636162809051E7
REPTREE	588	588	0	1	0	1	0.05603977304799976	2.2189636218849283E7	2.2189636162809051E7
SVR RBF	588	588	0	1	0	1	0.05603977304799976	2.2189636218849283E7	2.2189636162809051E7

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 4(Energieentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	589	589	0	0	0	0	0.0	2.218968182402792E7	2.218968182402792E7
Linear Regression	589	589	0	0	0	0	0.0	2.218968182402792E7	2.218968182402792E7
RBFNETWORK	589	589	0	0	0	0	0.0	2.218968182402792E7	2.218968182402792E7
REPTREE	589	589	0	0	0	0	0.0	2.218968182402792E7	2.218968182402792E7
SVR RBF	589	589	0	0	0	0	0.0	2.218968182402792E7	2.218968182402792E7

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 5(Energieentscheidung): Rohdaten.

I.2. Leistungsszenario

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	589	589	0	0	0	0	0.0	50037.04874599239	50037.04874599239
Linear Regression	589	589	0	0	0	0	0.0	50037.04874599239	50037.04874599239
RBFNETWORK	589	589	0	0	0	0	0.0	50037.04874599239	50037.04874599239
REPTREE	589	589	0	0	0	0	0.0	50037.04874599239	50037.04874599239
SVR RBF	589	589	0	0	0	0	0.0	50037.04874599239	50037.04874599239

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 1(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	589	589	0	0	0	0	0.0	50036.97145412234	50036.97145412234
Linear Regression	589	589	0	0	0	0	0.0	50036.97145412234	50036.97145412234
RBFNETWORK	589	589	0	0	0	0	0.0	50036.97145412234	50036.97145412234
REPTREE	589	589	0	0	0	0	0.0	50036.97145412234	50036.97145412234
SVR RBF	589	589	0	0	0	0	0.0	50036.97145412234	50036.97145412234

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 2(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	506	506	0	83	81	2	0.0010703883168749987	45810.91345411074	45810.912383722425
Linear Regression	586	586	0	3	1	2	0.0010703883168749987	50036.34333010459	50036.34225971627
RBFNETWORK	587	587	0	2	0	2	0.0010703883168749987	50036.34520212279	50036.34413173448
REPTREE	568	568	0	21	19	2	0.0010703883168749987	50035.81193151157	50035.81086112325
SVR RBF	587	587	0	2	0	2	0.0010703883168749987	50036.34520212279	50036.34413173448

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 3(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	585	585	0	4	3	1	1.5721814515624953E-4	50036.64819722609	50036.648040007945
Linear Regression	588	588	0	1	0	1	1.5721814515624953E-4	50036.74422944334	50036.7440722252
RBFNETWORK	588	588	0	1	0	1	1.5721814515624953E-4	50036.74422944334	50036.7440722252
REPTREE	588	588	0	1	0	1	1.5721814515624953E-4	50036.74422944334	50036.7440722252
SVR RBF	588	588	0	1	0	1	1.5721814515624953E-4	50036.74422944334	50036.7440722252

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 4(Leistungsentscheidung): Rohdaten.

model	true all	true positives	true negative	false all	false negative	false positive	loss	benefit	overall benefit
KNN Regression	589	589	0	0	0	0	0.0	50037.087751424595	50037.087751424595
Linear Regression	589	589	0	0	0	0	0.0	50037.087751424595	50037.087751424595
RBFNETWORK	589	589	0	0	0	0	0.0	50037.087751424595	50037.087751424595
REPTREE	589	589	0	0	0	0	0.0	50037.087751424595	50037.087751424595
SVR RBF	589	589	0	0	0	0	0.0	50037.087751424595	50037.087751424595

Offloadingentscheidung für Schach(difficulty = 3) mit Testkonfiguration 5(Leistungsentscheidung): Rohdaten.

J. Leistungsfaktor: Rohdaten

J.1. Schach (difficulty 1)

Leistungsfaktorberechnung mit Schach(difficulty=1) anhand unterschiedlicher Architekturen. Die Einträge sind in Nanosekunden.

Architektur A	Architektur B	Verhältnis (A/B)	Mittelwert (Verhältnis)	Standardabweichung (Verhältnis)
741803257	1313914366	0,56457504	0,53036815	0,03046147
537806529	1036072569	0,51908191		
469670428	889311798	0,52812796		
334295218	632567394	0,52847368		
392324432	746724703	0,52539367		
438420586	768407438	0,57055745		
143128779	271709089	0,52677214		
122841298	232242210	0,52893614		
299660611	513598833	0,58345267		
137392023	262602970	0,52319295		
167604747	320493543	0,52295826		
290768725	600028272	0,48459171		
170857232	398645527	0,42859438		
151505324	298933272	0,50681988		
265576414	508121034	0,52266369		
126691145	242669390	0,52207304		
197227453	364848783	0,54057314		
102245435	193654350	0,52797903		
124497300	235490902	0,52867138		
153476448	399588753	0,38408601		
237252471	451069792	0,5259773		
178173765	335510122	0,53105332		
1106886	1769216	0,62563644		
3299759	5829825	0,56601339		
4137998	19289630	0,2145193		
4659469	8305368	0,56101897		
3062589	5367695	0,57055943		
4867003	20896962	0,23290481		
6007047	10936493	0,5492663		
57814295	195880959	0,29515015		
58962714	108785816	0,54200737		
21291852	37954789	0,56097933		
133211529	246364198	0,54070977		
49845633	91834569	0,54277636		
70012879	137989077	0,50737986		
6545572	23888368	0,27400666		
6485922	11914520	0,54437124		
6416781	11759362	0,54567425		
8693358	23948512	0,36300201		
106128907	217330528	0,4883295		
8967219	16403793	0,54665522		
76692969	146488331	0,5235432		
49407191	93505292	0,52838925		
7269445	13510414	0,53806234		
6104014	11251298	0,54251643		
7045438	12418463	0,56733575		
8223778	15175268	0,54191979		
8597038	15792509	0,54437442		
8403485	15714043	0,53477549		
7779968	13897758	0,55980022		
33577851	62975297	0,53319083		

80041452	158864670	0,50383419
101976707	177953866	0,57305137
110792672	211609734	0,52357077
75922218	145252264	0,52269215
205133024	395537984	0,51861776
25333721	47738730	0,53067438
53701597	101098168	0,53118269
198366340	330739485	0,59976613
328861359	594661686	0,55302261
362066601	671423666	0,53925207
381754020	709842322	0,53780115
381258870	694491636	0,54897547
307006360	579918790	0,52939543
493797413	934042030	0,52866723
429621309	817411566	0,52558751
480917710	911210766	0,52777878
654883266	1242633121	0,52701256
645475597	1229301009	0,5250753
397464843	753558962	0,52745022
819647420	1523902361	0,53786085
823433388	1539877117	0,53473967
878753264	1656128912	0,5306068
630870639	1188733916	0,53070803
592226289	1115625786	0,53084672
474556683	892236153	0,53187341
514239742	974312368	0,52779761
667142671	1267436620	0,52637162
444986174	840890166	0,52918466
444522258	849654692	0,5231799
384759320	731333840	0,52610627
292090603	555579641	0,52574029
216175777	410637202	0,52643983
224448031	424999050	0,52811419
241169670	456215733	0,52863076
273895458	518984280	0,5277529
419983345	790095773	0,53156005
334015952	628941745	0,53107614
134208859	252230877	0,53208735
130167019	246339196	0,52840563
57597184	108399541	0,53134159
59350649	111054963	0,5344259
73024029	136430044	0,53524889
59280839	110987515	0,53412169
83743302	155615115	0,53814375
411958905	725872067	0,56753652
161776856	313270250	0,51641308
165375403	308660358	0,53578439
253804861	475487083	0,53377867
125087931	234944624	0,53241453
192002035	362447734	0,52973716
64860626	122001466	0,53163809
69058511	129086988	0,53497655
348145710	669351330	0,52012403
577885497	1098376210	0,52612711
602095489	1147537643	0,52468474
566507732	1070434436	0,5292316
522049802	989806201	0,52742628
530139224	1004906723	0,52755068
856593706	1609703876	0,53214366
46867852	87763482	0,53402453
45490665	86149726	0,5280419
43844693	82623474	0,53065661
43215074	81924338	0,52749983
142508210	274499646	0,51915626
145907315	274213703	0,53209345
160442883	307861825	0,52115225
131048276	250551906	0,52303843
499279764	957408535	0,52149082
94576067	180680740	0,52344299
36575445	70120719	0,52160682
51545942	98955628	0,52089955
67357179	129040127	0,5219863
1672813	2750733	0,60813354
2649400	4482328	0,59107678
3500607	6061704	0,57749554
4470816	7780631	0,57460841
8298667	14678727	0,56535332

3794787	6529244	0,58119853
84370961	150719696	0,55978723
101473725	181125954	0,56023846
145672263	270947954	0,53763928
104486560	190433176	0,54867835
114690479	209122887	0,54843581
74041580	133701608	0,55378227
108149237	195798014	0,55235104
51517430	93023265	0,55381232
56235383	100684904	0,55852845
52850234	93842670	0,56317914
85300489	156102504	0,54643895
326963078	600817404	0,54419708
601006167	1118265748	0,53744485
277863199	521481273	0,53283447
938057368	1782028656	0,52639859
1353668000	2577036485	0,52528088
975701773	1858222490	0,52507263
524061633	988589426	0,5301105
639765151	1199454781	0,53337997
883988072	1657140709	0,53344177
917512521	1731489456	0,52989784
876058721	1645593138	0,53236654
1539729392	2913432665	0,52849321
566436154	1072879856	0,52795861
215900131	410155996	0,52638541
160174913	306177950	0,5231432
205894060	394038500	0,5225227
210195206	400513316	0,52481453
195136515	370302697	0,52696488
275981579	526492861	0,52418864
312634228	599827398	0,52120698
316164128	607234382	0,52066243
399262095	767876932	0,51995584
395643778	752427004	0,52582347
472432895	901877308	0,52383278
439955021	838166330	0,52490181
573469054	1090045944	0,52609622
701957008	1335592574	0,5255772
205695063	392518792	0,52403877
22158349	42541819	0,5208604
54057607	104165436	0,51895916
45648886	87809307	0,51986387
50031170	96259483	0,51975316
53114036	103628458	0,51254295
59398197	114730190	0,51772072
58324310	112461133	0,51861749
41306960	80012934	0,51625353
41541329	81006585	0,51281422
381913289	741559233	0,51501387
255704981	496490397	0,51502503
451585354	878939648	0,51378426
193830778	378765821	0,51174305
160946797	314290942	0,51209493
111868215	214501117	0,52152742
141471198	273451783	0,51735336
208741379	405113111	0,51526691
69097127	133836019	0,516282
70169040	136949392	0,51237204
101373400	197084281	0,51436573
100342136	194732082	0,51528302
147130332	285852185	0,51470774
80830973	157807896	0,51221121
85185919	166997012	0,51010445
66789288	128986613	0,51780015
292260215	559419781	0,52243454
69946752	133298170	0,52473903
94795406	180514849	0,5251391
59197557	112677942	0,52536953
81468937	164436777	0,49544231
106570847	205304698	0,51908626
82693530	158859136	0,52054627
82787622	158570292	0,52208784
43839238	83365790	0,52586604
30828848	58487577	0,52710079
21268678	40342174	0,52720704
81349748	164037867	0,49592054

84577815	162045017	0,52194024
45286606	118903857	0,38086743
31225406	59613721	0,52379562
29546282	57008309	0,51828027
25604534	77533369	0,33023889
53528755	104098544	0,51421233
27278541	52919803	0,51546944
1678800	2745419	0,61149136
2036916	3453208	0,58986195
1550308	2550411	0,60786595
2578641	4342857	0,59376604
6104008	10825404	0,5638596
11212305	19635303	0,57102786
30438114	53793652	0,56583096
81157413	151731065	0,53487671
84373651	151223388	0,55794049
94240063	169802020	0,55499966
97041640	176368915	0,55021963
112546185	203816542	0,55219358
193123419	352398658	0,54802541
166261834	299950180	0,55429816
559511313	1033014165	0,54162986
970895017	1790343790	0,5422953
624008401	1140880479	0,54695335
995654857	1845061615	0,53963231
430138297	802741877	0,53583637
294463815	545482571	0,53982259
234104371	442348211	0,52923097
182134173	342855256	0,53122759
820389962	1564719954	0,52430466
1088892321	2056864336	0,52939433
516737495	991644872	0,52109128
515275767	984450694	0,5234145
381358882	732507019	0,52062147
202337673	385986145	0,52420968
205999927	390811880	0,52710764
669024317	1271414918	0,52620455
586359683	1122087079	0,52256166
154840718	293120613	0,52824916
276083073	526241806	0,52463159
455475743	859339176	0,53003023
427773401	804977651	0,53141028
387852166	730698548	0,53079641
301202418	566611688	0,53158525
361329356	671657369	0,53796679
321733368	599581395	0,53659665
274944210	510150456	0,5389473
539778096	1011528970	0,53362594
617092739	1144317785	0,53926693
233558713	435393287	0,53643159
83163482	155062501	0,53632233
77814077	144798948	0,53739394
76561061	143563004	0,53329241
404230795	764870509	0,52849573
145705560	269477484	0,5406966
116873068	215132647	0,54326049
422053623	779224667	0,54163278
149464924	275087730	0,54333548
146182343	269040606	0,54334677
342266828	633123247	0,54060063
194167458	357853374	0,54258943
674877802	1254685286	0,53788612
603561874	1133842728	0,53231534
555176001	1047089616	0,53020868
636805149	1207587157	0,5273368
652963079	1234328282	0,52900277
1038337984	1954137032	0,53135372
960807990	1797100013	0,53464358
345468130	643916533	0,53651073
557389344	1054533512	0,52856485
409072284	769669738	0,53149067
455849894	860255161	0,5299008
430841533	817177701	0,52723114
461236144	871662570	0,52914529
1156372155	2183653044	0,52955856
1030854373	1950007327	0,52864128
551490283	1041578993	0,52947524

476863768	904256880	0,52735432
2091673304	3949756574	0,52957018
1511171791	2842517906	0,53163141
1487707515	2828784938	0,5259175
367751539	693064937	0,53061628
585720539	1104414861	0,53034467
325384101	612292016	0,5314198
346942326	655346649	0,52940276
405545086	763121577	0,5314292
383571645	721969093	0,53128541
389973772	729541363	0,53454649
179222939	336279540	0,5329582
246570618	462219383	0,53344933
200476211	376258519	0,53281507
187867919	363384018	0,51699555
172249365	324358832	0,53104571
30174501	55999651	0,53883373
114286068	213093276	0,53631945
175264938	327391465	0,53533753
943198	1479626	0,63745703
1347126	2236682	0,60228767
1590484	2607337	0,61000323
2323856	3977431	0,58426054
3236495	5674970	0,5703105
5783814	10553884	0,54802706
6155122	10792908	0,5702932
4058473	7101865	0,57146581
4277461	7490625	0,57104194
5009070	8845609	0,56627757
4953275	8921766	0,55518997
8110987	14529585	0,55823941
8013129	14411677	0,55601642
7799851	13961965	0,55864995
73440265	133653897	0,54948091
7411353	13355489	0,55492936
15499786	28038621	0,5528013
16697882	30048522	0,55569728
28015366	51171341	0,54748157
141816242	264483238	0,53620125
203897143	386915636	0,52698088
243862487	456024644	0,53475726
400784712	765017659	0,52388949
406266177	775220384	0,5240654
378269333	722034921	0,5238934
369816026	705938225	0,52386457
637981677	1207772620	0,52822996
570561506	1081310177	0,52765758
830032552	1736219077	0,47806902
802866524	1554874797	0,51635445
897176175	1723242005	0,52063272
1718569800	3310778873	0,51908323
1775394435	3411153521	0,52046747
1509832072	2901859957	0,52029805
820843210	1577564033	0,52032323
902518005	1730854983	0,52142901
585171240	1121671696	0,52169565
684560367	1340720538	0,51059139
781690835	1502140765	0,52038454
1284094000	2461433139	0,52168551
1104429176	2122203633	0,52041621
570614020	1093960907	0,52160367
566624028	1105531505	0,5125354
222298377	422563931	0,5260704
218958892	420301354	0,5209569
302284861	565515758	0,53452951
261467922	500223356	0,52270235
175428403	335570089	0,52277723
21368841	40216461	0,53134563
71910957	136744841	0,52587693
131461061	250437232	0,52492619
171306236	324635718	0,52768758
171379249	326196001	0,52538734
79153297	150376218	0,52636845
114519013	218415637	0,52431692
122499456	230875894	0,53058574
129141322	243403897	0,5305639
95739380	180398243	0,53071127

73436940	138939636	0,52855285
68522593	129492910	0,52916096
68171523	128300156	0,53134404
70603055	133512500	0,52881232
213981928	405164512	0,52813591
230047590	430647997	0,53418939
127546682	238963222	0,53375026
49251478	91641696	0,53743525
54666724	102801960	0,53176733
68126016	130858231	0,52060933
31062279	59377583	0,52313141
29328164	55953327	0,52415407
83042953	158472081	0,52402261
89541726	171172885	0,52310695
98723863	190254504	0,51890421
70505780	135773961	0,51928794
82219524	156920454	0,5239567
78492350	150256325	0,52238966
109546432	207843957	0,52706094
134591942	256416345	0,52489611
182906231	346552089	0,52778857
114121729	216328142	0,52753991
19485812	44298032	0,43987986
77622036	148483703	0,52276468
31440210	60228590	0,52201471
46669112	88782363	0,52565747
72784395	138362719	0,52604051
123096878	233958258	0,52614889
244555911	474211355	0,51571079
215088721	412428757	0,52151727
94739916	184095710	0,51462316
44446713	85690223	0,5186906
32073005	62015986	0,51717318
11369986	21908867	0,51896732
5458829	10666189	0,51178814
5858607	11372072	0,51517498
17326747	33210069	0,52173174
5173133	9877492	0,5237294
5878456	11174321	0,5260683
4128370	7637130	0,54056563
11699637	22065313	0,53022756
12646651	23949786	0,5280486
7537914	14116271	0,53398762
9665380	18517075	0,52197121
9969010	19352464	0,51512872
8267822	15971009	0,51767687
13478921	25462507	0,52936347
3397946	6215440	0,54669436
15974275	30092494	0,53083919
3528022	6523714	0,54079961
4151305	7853923	0,52856451
5299735	10037375	0,5280001
7227950	13713415	0,52707148
13606763	25867463	0,52601846
1104363	1768953	0,62430319
3313132	5759382	0,57525825
4130047	7300958	0,56568563
4683742	8203376	0,57095298
3073075	5321961	0,57743283
4912119	8767840	0,56024277
5003376	8997153	0,55610658
37286926	67383778	0,55335167
37075730	67380913	0,55024084
37892546	68900962	0,5499567
116673060	211629828	0,55130726
248172006	454572161	0,54594634
139311458	253724163	0,54906658
141451238	258429984	0,5473484
123646804	225127845	0,54922928
291954714	534562805	0,54615606
601089438	1100287657	0,54630208
493012486	902565520	0,54623457
192062683	351333493	0,54666773
106168526	194209678	0,54666959
12206462	22367726	0,54571761
56696300	103635391	0,5470747
119970429	221466985	0,54170796

85037350	158178415	0,53760401
11157265	20820004	0,53589159
25772759	48249657	0,53415424
31241011	59930281	0,52128925
36226828	69250007	0,52313104
54601177	102647155	0,53193074
234123612	446177771	0,52473168
231836567	440824009	0,52591638
298763915	564453122	0,52929801
306555025	574770143	0,53335238
412833373	778295940	0,53043239
485913470	917919623	0,52936385
579210871	1088836990	0,53195371
841395565	1577619618	0,53333234
762681852	1430513229	0,5331526
744826333	1403888651	0,53054516
252734075	476858613	0,52999792
74925623	141432547	0,52976224
74977660	140507525	0,53362025
171811504	325153388	0,5284014
209740303	398511053	0,52630988
166710239	316419115	0,52686526
64939673	121649430	0,53382636
13922385	26249163	0,53039348
12388920	23457165	0,52815078
17048294	31897613	0,53446927
23227318	43593838	0,53281195
22712151	43782287	0,51875205
22082757	42860270	0,51522674
20740465	39923304	0,51950773
6561184	12411407	0,52864143
4410247	8190521	0,53845744
18623020	34795786	0,53520906
20428038	38164792	0,53525873
14296764	27011825	0,52927797
6863123	12970772	0,52912217
20598007	39160083	0,52599498
22685723	43314028	0,52375002
17128028	32864660	0,52116857
137627604	260893810	0,52752345
61595320	115475535	0,5334058
109830401	206509809	0,53184109
86148394	162754925	0,52931359
85789118	161941505	0,52975374
112386134	212383393	0,5291663
133233910	250686343	0,53147654
26868173	50019023	0,53715909
50533434	94873587	0,53263965
12668045	23589210	0,5370271
17335696	32712001	0,52994912
26520590	49534140	0,53540023
33526415	62541775	0,53606433
42111594	79127393	0,53219994
8459956	15873589	0,53295798
10975572	20263628	0,54163904
6424807	11832144	0,54299601
6374350	11581969	0,55036842
21824503	40933263	0,53317281
17226040	32184282	0,53523145
7773335	14225631	0,54643165
8625913	16124190	0,53496721
6393607	11846567	0,53970125
6230089	11611188	0,53655914
5408964	10025682	0,53951083
5074233	9366783	0,54172633
9828173	18084266	0,54346541
3883958	7162563	0,54225813
3503026	6441259	0,54384182
3765437	6801534	0,55361585
6454586	11974595	0,53902332
5933147	11024805	0,53816344
6760496	12456358	0,54273456
7189237	13521245	0,53169934
8391953	15828316	0,53018609
2298499	4251541	0,54062727
1680808	2754960	0,61010251
2609940	4483351	0,58214046

3477308	6060464	0,57376927
4418532	7784294	0,56762142
3710028	6477613	0,57274616
9746457	17318335	0,56278257
111670958	200766427	0,55622327
88297132	160537852	0,55000818
160915613	292785408	0,54960257
161469172	293018722	0,55105411
109105272	196949797	0,55397504
74714502	135370709	0,55192517
39074670	71402623	0,54724418
283597447	521808054	0,54348998
170790410	311870298	0,54763282
1170840456	2143638183	0,54619313
942835161	1709021322	0,55168133
410109734	749537573	0,54715033
655227103	1196067684	0,54781775
635970014	1167486401	0,54473441
344417181	629898349	0,54678216
616770594	1127476618	0,54703626
540180536	1001167524	0,5395506
787541407	1474342525	0,53416448
1240811561	2285688854	0,5428611
1734706497	3213288695	0,53985392
1950969253	3635302793	0,53667311
755294462	1427186440	0,5292192
832731550	1562867386	0,53282291
741932699	1390156943	0,53370427
2605629984	4885196943	0,53337256
3166016155	5942063896	0,53281422
3516959655	6659701676	0,52809568
1879999053	3539434065	0,53115809
855985268	1607069751	0,53263728
1558690604	2950948897	0,5281998
631812149	1202140433	0,52557266
581389029	1095454424	0,53072863
649262098	1227577871	0,52889687
1197857646	2261252845	0,52973185
1217520739	2310592724	0,52693005
516743191	971978331	0,53164065
628966206	1177241407	0,53427122
300890239	562637780	0,53478499
164911621	308069996	0,53530569
121407476	227155637	0,53446825
179917412	339060126	0,53063571
136097303	254287859	0,5352096
155252945	289710707	0,53588957
127227287	237769237	0,53508725
253854184	477266238	0,53189219
248489913	464638884	0,53480223
612150633	1158262796	0,52850755
253501893	477594112	0,5307894
378150582	705199444	0,5362321
314280913	587488596	0,53495662
108444998	203463898	0,53299381
114265410	211566606	0,5400919
442631461	827915236	0,53463379
329388506	618154152	0,5328582
144248986	268392246	0,53745586
127049134	239552737	0,53035977
237038695	449083840	0,52782726
169399550	318496344	0,53187282
109230660	204387157	0,53443016
197813103	373370824	0,52980332
275290097	523434475	0,52593039
193599148	365843311	0,52918597
186735454	352948435	0,52907291
165552555	312245136	0,53020059
102907526	193625762	0,53147642
49868400	93198824	0,53507542
69027829	130866125	0,52746904
77307455	146220192	0,52870574
101498983	190047179	0,53407256
80167074	155098189	0,5168795
100630565	194727876	0,51677534
118049282	229683790	0,51396436
312686175	603950137	0,51773508

96001695	184811030	0,51945869
24124798	46725720	0,51630661
16410725	31371311	0,5231125
58495665	112865012	0,51827988
34474277	66430343	0,51895377
35486728	67345251	0,52693735
129758215	248228546	0,52273688
257823879	497157606	0,51859587
26026851	49278033	0,52816335
50590447	98150173	0,5154392
44461505	87877621	0,50594798
26493314	86546035	0,30611817
35437516	69184877	0,51221477
24341144	46539266	0,52302381
38566638	73794271	0,52262374
36053506	69211854	0,52091519
54692311	103718045	0,52731722
48622895	92433651	0,52603023
200329788	384133491	0,52151086
112053757	212569094	0,5271404
70817335	134194237	0,52772262
17751708	33679993	0,52706982
18478216	35190539	0,52509045
3609249	6700407	0,53866116
6847859	13193445	0,51903495
9016610	16968210	0,53138251
9298501	17430480	0,53346213
31385034	60046670	0,52267734
9778420	18293441	0,53453147
10175883	19061071	0,53385683
54990201	104332947	0,52706458
43793073	83054481	0,52728128
20416221	40365989	0,5057778
12507821	24885307	0,50261871
5069249	9829173	0,51573505
8399642	16203747	0,51837652
33382219	65036482	0,51328451
31203090	62484470	0,49937352
30082042	61208168	0,49147104
28608197	57275002	0,49948836
16608727	31770211	0,52277673
19042293	36705323	0,51878832
940120	1833045	0,51287339
13862304	28427410	0,48763866
1442139	2716434	0,53089418
11484408	23036032	0,49854107
11180601	22449721	0,49802851
6380651	12865906	0,49593484
7185405	14535348	0,49434007
4459156	8770805	0,50840898
13604395	28070960	0,48464303
17632374	37005435	0,47648066
18292268	37392719	0,48919331
17282509	35596623	0,48550979
16176474	33088445	0,48888589
14229612	28974898	0,49110137
9261125	19097208	0,48494654
7874168	16316958	0,48257574
8039814	16823456	0,47789313
6330031	13087208	0,48368078
12834455	27019247	0,47501157
9373490	19567040	0,47904486
11481910	23100632	0,49703878
11547431	22985447	0,50238009
18736830	38068038	0,49219321
9578299	19453029	0,49238085
16803473	35231037	0,47695085
19048549	39088716	0,4873158
15238724	30669822	0,49686379
11106913	22424688	0,49529844
16936009	34960962	0,48442629
11557429	23493926	0,49193264
12945307	26473084	0,48899883
13863956	28318309	0,4895757
16355490	34098174	0,47965882
29879865	59981500	0,49815135
9455235	19415932	0,48698332

11284190	22898161	0,49279896
10162770	20494325	0,49588215
9780254	19761465	0,49491543
9633947	19718853	0,48856528
10943919	22540567	0,48552102
12224377	25094832	0,48712727
8652239	17505246	0,49426549
15896517	33371546	0,47634943
16547539	34303355	0,48238836
14180892	28578353	0,49621096
17954987	37172990	0,48301164
7115523	13768872	0,51678329
3669525	6902161	0,53164871
1117968	1769142	0,63192666
3302291	5803450	0,56902205
4147380	7282721	0,5694822
4656993	8227328	0,56603955
3082615	5340121	0,57725565
4853043	8781257	0,55265926
6050590	10889162	0,55565249
5788582	10418699	0,55559547
6763544	12099380	0,55899922
9790473	17244344	0,56774981
26588876	48053971	0,55331277
14704110	26531805	0,55420692
5191638	9159118	0,56682729
5275215	9018844	0,58491033
5617956	10091720	0,55668964
38843120	72958260	0,53240195
8523577	15430071	0,55240037
6564652	11778478	0,55734298
51266173	96862962	0,52926497
68794815	130269795	0,5280949
71313259	134773306	0,52913489
68279770	151032631	0,45208621
71728194	136610344	0,52505683
96689312	181869936	0,53163989
98448449	184931260	0,53235158
202259934	380552449	0,5314903
226658280	426406988	0,53155386
307516613	584060469	0,52651503
897637472	1695132896	0,52953811
732041370	1391437428	0,52610441
267106718	506491476	0,52736666
164057256	311319154	0,5269745
197120393	377899275	0,52162152
310292660	594729475	0,52173748
83696378	159391647	0,5250989
208055694	393931475	0,52815199
59384365	111994998	0,53024123
57299027	107159578	0,53470747
57606939	107520961	0,53577403
20302485	37624946	0,5396017
28824444	54030705	0,53348266
20061850	36874019	0,54406464
15415632	28532309	0,54028687
14992657	27400711	0,54716306
9946437	18440638	0,53937597
12319661	20519726	0,60038136
19897071	37060626	0,53687898
49221161	92453506	0,53238826
99946509	187727516	0,53240202
77989208	147196128	0,52983193
73156696	137156646	0,53338061
77690823	144663582	0,53704479
89399285	168024233	0,53206185
68533716	127703956	0,53666087
49902930	91963522	0,54263831
63304999	117596525	0,53832372
63727390	116649179	0,54631666
68349945	127361776	0,5366598
53788004	101689247	0,52894485
69454217	130207158	0,5334132
68343539	129634924	0,52720005
53018536	100427041	0,52793088
46465339	87848419	0,5289263
79910359	151223840	0,52842435

80572384	152110718	0,52969564
94057450	177805219	0,5289915
56069393	105520562	0,53135988
53743557	101599587	0,52897417
66154352	124051272	0,53328234
21862308	40571064	0,53886455
32341292	60774386	0,53215333
2964668	5388905	0,55014293
12005515	22308681	0,53815441
94133677	172773807	0,54483766
60611477	111523629	0,54348552
21211941	39464162	0,53749883
8669897	15963869	0,54309497
18511917	35159711	0,52650936
10728458	20235946	0,53016834
15769076	29193235	0,54016199
12608845	22966088	0,54902015
7133475	13219469	0,53961888
4812755	8751895	0,54991005
6647434	12233234	0,54339139
12640899	23553056	0,53669889
9149069	16914061	0,54091498
12603475	23279999	0,54138641
9724440	17849034	0,54481604
10513471	19500535	0,53913757
13340108	24829257	0,53727375
11588519	21361749	0,54248924
1013336	1769094	0,57279941
13301957	24279330	0,54787167
10710334	19487913	0,54958856
11467930	21194000	0,54109323
9869645	18333692	0,53833374
8671997	16278691	0,53272078
9206464	17184797	0,53573307
7426181	13901748	0,53419045
7352411	13802476	0,53268783
7423116	13977868	0,5310621
8348404	15376292	0,54294
16485680	30725150	0,53655328
5185948	9687617	0,5353172
2160247	3897546	0,55425824
1085316	1943852	0,55833263
1124069	1938099	0,57998534
1573483	2855115	0,5511102
1593924	2834846	0,56226123
1755735	3104241	0,56559236
6166035	11233970	0,54887408
6422668	11829958	0,54291554
5904309	10912529	0,54105781
6011820	11057600	0,54368217
3990483	7296548	0,54690012
7185838	13283765	0,5409489
1824141	3320828	0,54930307
1142792	2000502	0,57125262
3406695	6270541	0,54328566
3389253	6311983	0,53695534
5207950	9715930	0,53602177
8716153	16642631	0,52372446
9002505	17376488	0,51808542
16663267	31684357	0,52591463
3304799	6178945	0,53484842
1792992	3306224	0,54230808
3610802	6909590	0,52257833
4828653	9286162	0,51998371
1434940	2501114	0,57372035
2654907	4784072	0,55494712
2646289	4820317	0,54898651
2481045	4533466	0,54727332
3333622	6142689	0,54269751
2713122	4906521	0,55296248
2689282	4889309	0,55003314
3838890	7154537	0,53656722
6295714	11847180	0,53141034
3481387	6442981	0,54033793
1996426	3488423	0,57230043
3074797	5629112	0,54623127
3861055	7225462	0,53436791

2149377	3911015	0,54957012
1876041	3423363	0,54801112
6672522	12347331	0,54040197
4546723	8433313	0,53913841
3137071	5745866	0,54597009
4751882	9267783	0,51273125
5160554	10026915	0,51467017
3784285	7280178	0,51980666
5629689	10650423	0,5285883
3571687	6736937	0,53016482
4117538	7811435	0,52711672
4334649	8149249	0,53190779
4078956	7738876	0,52707344
2774919	5205547	0,53306963
3250053	6108213	0,53207919
3697132	6943796	0,53243672
3908798	7497745	0,52132981
5023253	9923767	0,50618409
3876796	7665691	0,5057334
5560735	11121433	0,50000166
521496	1000334	0,52132188
5891995	12151993	0,48485833
918346	1843534	0,49814432
6430158	13210411	0,48674928
2412248	4899188	0,4923771
1347838	2759613	0,48841559
2397521	4840664	0,49528763
2002922	3890634	0,51480607
782296	1535014	0,50963444
1439733	2843540	0,50631713
1833582	3657419	0,50133222
1605734	3307206	0,48552585
2401222	5004610	0,47980202
1697115	3364130	0,50447367
1663775	3406598	0,48839781
1448061	2801392	0,51690767
1305983	2455427	0,53187613
1637476	3044047	0,53792731
1542165	2905129	0,53084218
4798331	9128213	0,5256594
1526346	3000044	0,50877454
1700833	3180540	0,53476234
3695255	6820406	0,54179399
3008986	5744844	0,52377158
2652056	5021554	0,52813452
3611623	6852395	0,52705995
4735616	8953194	0,52893035
9523238	18015948	0,52860044
7000150	13814221	0,50673505
8851679	17043937	0,51934474
8444586	15895530	0,53125539
4575271	8583137	0,53305347
624446	979298	0,63764656
973264	1621158	0,60035111
1806229	2975356	0,60706315
2690387	4733027	0,56842841
2887046	4954245	0,58274187
2742651	4841349	0,56650553
3336769	5832682	0,57208142
3884941	6851094	0,56705411
6744199	12033369	0,56045809
4158521	7333043	0,5670935
4799235	8598638	0,55813898
80843618	146523942	0,5517434
6114845	10910759	0,56044176
29773820	54657300	0,54473638
28655377	53830458	0,53232646
28512958	53107206	0,53689433
88247230	164918006	0,5350976
103147724	194952191	0,52909241
191009869	364442373	0,52411542
178270701	342342292	0,52073818
125863871	241270824	0,5216705
219763167	421525720	0,52135174
155885428	293584427	0,53097308
151114910	283295860	0,53341729
335437331	633204035	0,52974604

145296756	274570902	0,52917755
238444068	450782346	0,52895609
176616167	334399081	0,52815985
156042238	293896529	0,53094277
41103858	77199672	0,53243566
96083546	179080483	0,53653835
96478205	180578701	0,53427234
8651211	16120079	0,53667299
75511066	141000439	0,53553781
260790429	498909480	0,52272093
353012884	676504409	0,52181904
331920793	632421575	0,52484103
460209182	876809891	0,52486769
284339726	541349691	0,52524224
108888480	209788398	0,51903957
283034493	544464630	0,51984
497169892	949375766	0,52368083
381664584	726226705	0,52554468
476992642	917106363	0,52010613
420165415	812751792	0,51696646
592857790	1142453561	0,51893382
630420951	1210531132	0,52078045
646113201	1245138623	0,51890865
566836901	1089820959	0,52011929
132554965	249474860	0,53133596
169397530	320948063	0,52780356
805202224	1559668299	0,51626504
1227029303	2371972557	0,51730333
942048371	1820958089	0,51733666
1588979398	3091523128	0,51397946
244041504	475761872	0,51294885
278723795	546082278	0,51040623
213449108	416540291	0,51243328
236775237	456236551	0,51897472
193526162	373715680	0,5178433
174532478	335095555	0,52084391
119077215	226469748	0,52579745
116967451	222122096	0,52659079
106688945	205253886	0,51979013
282760831	540494301	0,52315229
189303724	364106370	0,51991324
184324796	357671782	0,51534621
184688787	359932157	0,51312111
445085527	864101842	0,51508457
418573225	804624936	0,52020911
358711353	695827757	0,51551745
45598751	87830703	0,51916641
149485830	288036893	0,51898154
102335935	196392705	0,52107809
99812028	192464001	0,51860102
255900263	494641903	0,51734449
262457616	505933416	0,5187592
49772756	95028478	0,52376674
18175246	34531718	0,52633483
3297882	5998507	0,5497838
32763834	61032744	0,53682387
42499158	79470220	0,53478093
47765872	90122216	0,53001218
13757968	25597032	0,53748294
13838555	25583419	0,54091891
1109766	1770821	0,62669575
2088865	3563787	0,58613632
42363313	75880081	0,55829293
64083005	114537551	0,55949341
59501816	107169751	0,55521092
75909541	137750750	0,55106445
78012983	141707296	0,55052199
37818660	69112801	0,54720196
38786876	70244447	0,55217
46824325	85077404	0,55037322
57461728	103717131	0,5540235
173751754	322812379	0,53824378
209327825	388223786	0,53919371
185870198	346939567	0,53574229
188088379	343472353	0,5476085
121449347	221738317	0,54771475
135515930	247291638	0,54800045

69698490	128397022	0,54283572
88142535	161917564	0,54436673
30795657	56746282	0,5426903
11941252	21997696	0,54284103
12081342	22370912	0,54004691
11256658	20932719	0,53775422
11041027	20561352	0,53697962
17324953	32184003	0,53830945
74685695	139307336	0,53612177
358714349	672913133	0,53307675
146511124	273498942	0,53569174
38607485	71451842	0,54032876
19215973	35601953	0,53974491
72552941	136343772	0,53213242
219911026	422810787	0,52011688
46707694	88646735	0,52689695
40927615	77687074	0,52682657
123993258	234811871	0,52805362
9597638	18105461	0,53009631
16765633	31574102	0,53099319
74898069	140575445	0,53279624
90601785	168906412	0,53640228
69650046	129766753	0,53673259
105772047	199134320	0,5311593
112337157	210313036	0,53414262
298677421	558443300	0,5348393

J.2. Schach (difficulty 3)

Architektur A	Architektur B	Verhältnis (A/B)	Mittelwert (Verhältnis)	Standardabweichung (Verhältnis)
264672017	567642049	0,4662657	0,55158319	0,05797891
522334674	954213755	0,54739797		
495102868	920950156	0,53760007		
3,3925E+10	59568771980	0,56950849		
2,2003E+10	38412825682	0,57281541		
2,0106E+10	35136592722	0,57223174		
2,2786E+10	39670635481	0,57438746		
1,7694E+10	30432538056	0,58140145		
1,744E+10	30129717382	0,57881916		
1,9018E+10	32922608150	0,57765466		
2,5956E+10	45430515181	0,57133071		
1340531632	2246404508	0,59674543		
1311728074	2317987818	0,56589084		
2561082113	4436311339	0,5772999		
3186652676	5545036429	0,57468562		
2,909E+10	50674794630	0,5740533		
5,4148E+10	94014432666	0,57595065		
4,5937E+10	79497073516	0,5778445		
3,5398E+10	61318779555	0,57727367		
3,684E+10	63948252596	0,57609306		
2,8942E+11	5,06198E+11	0,57175243		
2,9764E+11	5,1913E+11	0,57334618		
2,8634E+11	4,97602E+11	0,57543332		
3,2629E+11	5,65961E+11	0,57652808		
3,32E+11	577.956.030.523.000.000	5,7444E-07		
3,3734E+11	5,87346E+11	0,57434965		
8717619224	15111796711	0,5768751		
832628429	1449919975	0,5742582		
1385361022	2513847753	0,55109186		
753347550	1300533499	0,5792604		
220731159	371267763	0,5945336		
282071138	564444398	0,49973237		
276297945	578934341	0,47725264		
420952279	748036361	0,56274307		
505694325	902379903	0,5604007		
974917369	1770504093	0,55064395		
942017841	1743018405	0,54045203		
479766829	862505634	0,55624776		
767541345	1387261807	0,55327793		
362387023	663252111	0,546379		
376543560	669543128	0,56238881		
893047116	1607836376	0,55543408		
2377666891	4350994489	0,54646516		
1560836227	2845701452	0,5484891		
1598562078	2909034531	0,54951636		
1054789496	1884845558	0,55961587		
1473564322	2637600907	0,558676		
1295342989	2300146702	0,56315668		
1819472529	3244068016	0,5608614		
2688332301	4854682541	0,55376068		
1530583292	2764530277	0,5536504		
1502391110	2730469808	0,55023172		
2209223879	3979523516	0,55514784		
2231581196	4014936500	0,5558198		
936145853	1685333136	0,55546635		
998985088	1788433835	0,55858096		
2541721725	4556581127	0,55781334		
2286017518	4086534510	0,55940247		
2568092819	4602903867	0,55792884		
3029952597	5450924597	0,5558603		
1203558065	2139507039	0,56253989		
901248569	1592670889	0,56587244		
1226356498	2182009468	0,56203079		
1427084428	2576479718	0,55388925		
863736222	1564763451	0,55199156		
569515434	1037414518	0,54897577		
870590191	1576433937	0,55225289		
164697614	298420760	0,55189731		
405184719	743457108	0,5450008		
364928529	668410576	0,54596462		
414625457	772874079	0,5364722		

85272365	156148357	0,54609838
174553391	317169540	0,55034727
156926547	284787040	0,55103121
87924960	159375131	0,55168557
66815146	121142245	0,55154291
44986792	81333945	0,55311214
80965547	146958430	0,55094183
193743925	353448957	0,54815249
233204638	422101232	0,55248509
79876991	145036925	0,55073555
122022853	222572735	0,5482381
198763851	360812775	0,55087809
181711814	331005863	0,54896857
93794664	170220180	0,55101965
95029529	171724018	0,55338519
95515643	172850540	0,55259094
97142683	176396190	0,55070738
62354292	111034969	0,56157346
66924141	119319556	0,56088158
179663719	326958345	0,54950033
71810560	129557858	0,55427406
138084663	249053995	0,55443665
269130582	487436855	0,55213425
98291455	176848443	0,55579486
43018454	77398983	0,55580128
67468827	122778599	0,54951618
248849200	453239805	0,54904533
70762755	126776202	0,55817065
114313363	205360922	0,55664613
110299034	198649475	0,55524453

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