PROFESSUR FÜR
WIRTSCHAFTSINFORMATIK
DER FREIEN UNIVERSITÄT BERLIN

Masterarbeit

Eine umfassende Studie zur Vorhersage der Nutzerlast in Web-Anwendungen: Eine Untersuchung von algorithmischen Ansätzen durch statistische Evaluationsmethoden

Emanuele Luca Maurer

Gutachter(in):	Univ.-Prof. Dr. rer. pol. Natalia Kliewer
Semester:	Sommersemester 2023
Verfasser:	Emanuele Luca Maurer
Matrikel-Nr.:	4683195
Adresse:	Frommannstr. 06, 07743 Jena
Email:	E.L.Maurer@gmx.de
Telefon:	0176 43847978
Studienfach:	Master of Science Wirtschaftsinformatik

Abgabetermin: 03. Juli 2023

Abstract

Dieser Blindtext zeigt den ungefähren Umfang des deutschen Abstract.Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc, quis gravida magna mi a libero. Fusce vulputate eleifend sapien. Vestibulum purus quam, scelerisque ut, mollis sed, nonummy id, metus. Nullam accumsan lorem in dui. Cras ultricies mi eu turpis hendrerit fringilla. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In ac dui quis mi consectetuer lacinia. Nam pretium turpis et arcu. Duis arcu tortor, suscipit eget, imperdiet nec, imperdiet iaculis, ipsum.

Abstract

The following blindtext illustrates the length of the Abstract in english. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim. Donec pede justo, fringilla vel, aliquet nec, vulputate eget, arcu. In enim justo, rhoncus ut, imperdiet a, venenatis vitae, justo. Nullam dictum felis eu pede mollis pretium. Integer tincidunt. Cras dapibus. Vivamus elementum semper nisi. Aenean vulputate eleifend tellus. Aenean leo ligula, porttitor eu, consequat vitae, eleifend ac, enim. Aliquam lorem ante, dapibus in, viverra quis, feugiat a, tellus. Phasellus viverra nulla ut metus varius laoreet. Quisque rutrum. Aenean imperdiet. Etiam ultricies nisi vel augue. Curabitur ullamcorper ultricies nisi. Nam eget dui. Etiam rhoncus. Maecenas tempus, tellus eget condimentum rhoncus, sem quam semper libero, sit amet adipiscing sem neque sed ipsum. Nam quam nunc, blandit vel, luctus pulvinar, hendrerit id, lorem. Maecenas nec odio et ante tincidunt tempus. Donec vitae sapien ut libero venenatis faucibus. Nullam quis ante. Etiam sit amet orci eget eros faucibus tincidunt. Duis leo. Sed fringilla mauris sit amet nibh. Donec sodales sagittis magna. Sed consequat, leo eget bibendum sodales, augue velit cursus nunc, quis gravida magna mi a libero. Fusce vulputate eleifend sapien. Vestibulum purus quam, scelerisque ut, mollis sed, nonummy id, metus. Nullam accumsan lorem in dui. Cras ultricies mi eu turpis hendrerit fringilla. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; In ac dui quis mi consectetuer lacinia. Nam pretium turpis et arcu. Duis arcu tortor, suscipit eget, imperdiet nec, imperdiet iaculis, ipsum.

Sperrvermerk

Die vorgelegte Masterarbeit basiert auf internen, vertraulichen Daten und Informationen des Unternehmens ..... In diese Arbeit dürfen Dritte, mit Ausnahme der Gutachter und befugten Mitgliedern des Prüfungsausschusses ohne ausdrückliche Zustimmung des Unternehmens und des Verfassers keine Einsicht nehmen. Von diesem Verbot ausgenommen sind außerdem jene Personen, die auch ansonsten zur Einsichtnahme in die genannten Daten und Informationen befugt sind. Eine Vervielfältigung und Veröffentlichung der Masterarbeit ohne ausdrückliche Genehmigung – auch auszugsweise – ist nicht erlaubt.

Berlin, den 01. Januar 2099

.....

(Unterschrift des Verfassers)

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Im Zeitalter der Digitalisierung sind Web-Anwendungen ein wesentlicher Bestandteil unseres täglichen Lebens geworden. Sie unterstützen uns in den unterschiedlichsten Bereichen wie Kommunikation, Arbeit und Unterhaltung. Im Kontext dieser rasanten technologischen Entwicklung spielt die Vorhersage der Nutzerlast in Web-Anwendungen eine entscheidende Rolle, um eine hohe Leistungsfähigkeit und eine optimale Nutzererfahrung zu gewährleisten. In dieser Masterarbeit sollen verschiedene algorithmische Ansätze untersucht und miteinander verglichen werden, um ihre Leistungsfähigkeit bei der Vorhersage der Nutzerlast einer virtuellen Büroplattform zu evaluieren. Dabei wird der Fokus auf statistische Evaluationsmethoden gelegt. Im Folgenden wird zunächst der Hintergrund und die Motivation dieser Arbeit erläutert, bevor die Zielsetzung, der Aufbau der Arbeit, die Forschungsfragen und die Literatur im Detail vorgestellt werden.

1.1 Hintergrund und Motivation

Mit dem rasanten technologischen Fortschritt im letzten Jahrzehnt hat Cloud Computing sowohl in der Industrie als auch in der Wissenschaft immer mehr an Popularität gewonnen (Al-Dhuraibi et al. (2017)). Ein Kernmerkmal von Cloud Computing ist die Elastizität, die es den Anwendungsbesitzern ermöglicht, unvorhersehbare Arbeitslasten zu bewältigen, indem Ressourcen basierend auf der Nachfrage bereitgestellt oder freigegeben werden, um die Leistung zu verbessern und gleichzeitig die Kosten zu senken (Fernandez et al. (2014), Al-Dhuraibi et al. (2017)).

Die Autoskalierung, ein Prozess, der dynamisch Ressourcen akquiriert und freigibt, spielt in der Cloud-Computing-Umgebung eine wesentliche Rolle (Qu et al. (2018)). Es gibt sowohl reaktive als auch proaktive Ansätze zur Autoskalierung. Reaktive Ansätze führen Skalierungsaktionen durch, indem sie den aktuellen Zustand des Systems anhand von vordefinierten Regeln oder Schwellenwerten analysieren. Proaktive Ansätze hingegen analysieren historische Daten, prognostizieren die Zukunft und treffen auf dieser Basis Skalierungsentscheidungen (Wang et al. (2021)).

Die Auswahl einer geeigneten Autoskalierungsmethode kann die Qualität der Bereitstellung der Applikation erheblich beeinflussen. Parameter wie CPU-Auslastung, Speicher

und Antwortzeit können Über- und unterprovisioniert werden. Überprovisionierung kann zu Ressourcenverschwendung führen und ist aufgrund der Pay-per-Use-Preisgestaltung von Cloud-Anbietern, kostspielig, während Unterprovisionierung die Systemleistung beeinträchtigen kann und somit zu einer schlechten Nutzererfahrung führt (Mao & Humphrey (2011)).

Kubernetes ist ein weit verbreitetes Management-System für die Orchestrierung von Containern, das reaktive Autoskalierungsverfahren auf der Grundlage statischer Regeln bietet, um unter schwankenden Arbeitslasten zu operieren (Burns et al. (2022)).

Trotz ihrer Vorteile können diese reaktiven Autoskalierungsmethoden in bestimmten dynamischen Arbeitslastszenarien zu einer schlechten Ressourcennutzung oder einer unbefriedigenden Qualität des Dienstes (QoS) führen (Mao & Humphrey (2011)). Daher ist es eine wichtige Forschungsrichtung, Autoskalierungsmethoden zu entwickeln, die das System-QoS hinsichtlich der Antwortzeit ständig sicherstellen und gleichzeitig die optimale Menge an Ressourcen in Bezug auf die Anzahl der Containerinstanzen zuweisen.

Vor diesem Hintergrund besteht die Motivation dieser Masterarbeit, effektive Algorithmen zur Vorhersage der Nutzerlast von Web-Anwendungen zu untersuchen und zu evaluieren, um eine effiziente Ressourcenallokation und Leistungsoptimierung zu gewährleisten. Die Arbeit wird sich dabei auf die Analyse von Zeitreihendaten und den Einsatz von Algorithmen aus dem Bereich des maschinellen Lernens, befassen.

1.2 Aufbau der Arbeit

Diese Masterarbeit ist in sieben Hauptkapitel unterteilt, um die Untersuchung des algorithmischen Ansatzes zur Vorhersage der Nutzerlast in Web-Anwendungen strukturiert darzustellen.

Das erste Kapitel, die Einleitung bietet einen allgemeinen Überblick über das Forschungsthema und seine Bedeutung. Es skizziert den Hintergrund, die Motivation, die Zielsetzung und die Forschungsfragen, die diese Arbeit leiten. Weiterhin liefert es einen Ausblick auf die Literatur, die zur Untermauerung und Analyse des Themas herangezogen wird.

Das zweite Kapitel, “Grundlagen“beschäftigt sich mit den theoretischen Konzepten, die für das Verständnis dieser Arbeit unerlässlich sind. Es beleuchtet Aspekte der Zeitreihen, Web-Anwendungen und Nutzerlast sowie statistische Evaluationsmethoden. Darüber hinaus wird auf die Herausforderungen und Auswirkungen eingegangen, die mit der Vorhersage der Nutzerlast verbunden sind.

Im dritten Kapitel, "Daten“wird die Datengrundlage der Arbeit vorgestellt. Hier werden

die Datenquellen, die Auswahl der Daten und die Datenaufbereitung ausführlich erläutert. Zudem werden statistische Tests zur Beschreibung der Daten und Methoden zur Featurization vorgestellt.

TODO Das vierte Kapitel, MMethodik“beleuchtet die verschiedenen angewendeten Methoden. Hier wird auf die Bayesian Optimization, die Probabilistic Matrix sowie die Implementierung der Algorithmen eingegangen. Es werden auch die Evaluationsmethoden und Kriterien präsentiert, die zur Beurteilung der Leistungsfähigkeit der Algorithmen eingesetzt werden.

TODO Im fünften Kapitel, Älgorithmen und Hyperparameter“werden die spezifischen Algorithmen ARIMA, Regressionen und neuronale Netzwerke vorgestellt, die zur Vorhersage der Nutzerlast verwendet werden. Hier wird auch auf die Auswahl und Einstellung von Hyperparametern eingegangen.

Das sechste Kapitel, Ërgebnisse und Diskussionppräsentiert die Ergebnisse der Untersuchung. In diesem Kapitel werden die Ergebnisse der verschiedenen Algorithmen verglichen und diskutiert. Darüber hinaus werden mögliche Limitationen und Verbesserungen der Arbeit aufgezeigt.

Das siebte und letzte Kapitel, FFazit und Ausblickßieht ein abschließendes Resümee aus der Arbeit und gibt einen Ausblick auf mögliche zukünftige Forschungsrichtungen. Es bietet eine Zusammenfassung der wichtigsten Erkenntnisse und leitet daraus Empfehlungen für die Praxis und für zukünftige Forschung ab.

1.3 Forschungsfragen

Das folgende Kapitel stellt die zentralen Fragen vor, die diese Arbeit zu beantworten versucht und die das Rückgrat unserer Untersuchung bilden. Die Forschungsfragen dienen als Wegweiser und definieren den Rahmen und die Richtung der Forschung. Die Masterarbeit konzentriert sich auf zwei Forschungsfragen.

Bieten sich Algorithmen für die Nutzerlastvoraussage von digitalen Büros an?

Die Frage zielt darauf ab, zu untersuchen, ob verschiedene Algorithmen - einschließlich ARIMA, Regressionen, neuronale Netzwerke und weitere - effektiv zur Vorhersage der Nutzerlast von digitalen Büros eingesetzt werden können. In der Betrachtung werden sowohl die Anwendbarkeit dieser Algorithmen auf den spezifischen Kontext der Nutzerlastvorhersage als auch die Herausforderungen und potenziellen Einschränkungen ihrer Anwendung erörtert.

Welcher Algorithmus liefert die besten Ergebnisse für die Vorhersage der Nutzerlast?

Diese Frage zielt darauf ab, die verschiedenen untersuchten Algorithmen zu vergleichen und herauszufinden, welcher von ihnen die genauesten und zuverlässigsten Vorhersagen für die Nutzerlast liefert. Diese Frage beinhaltet die Durchführung einer gründlichen Evaluierung der verschiedenen Algorithmen unter Verwendung geeigneter statistischer Evaluationsmethoden und Kriterien, um ihre Leistung in Bezug auf die Nutzerlastvorhersage zu beurteilen.

Die Antworten auf diese Forschungsfragen werden dazu beitragen, das Verständnis für die Vorhersage der Nutzerlast in Web-Anwendungen zu vertiefen und wertvolle Erkenntnisse für die Praxis zu liefern.

1.4 Literatur

Die schwellenwertbasierte reaktive Skalierung, obwohl effizient in bestimmten Szenarien, kann aufgrund ihrer inhärenten reaktiven Natur nicht gut auf plötzliche oder unvorhergesehene Spitzen in der Arbeitslast reagieren. Es kann eine Verzögerung zwischen dem Zeitpunkt, an dem eine Arbeitslastspitze auftritt, und dem Zeitpunkt, an dem zusätzliche Ressourcen bereitgestellt werden, geben. Dies kann zu einer suboptimalen Benutzererfahrung, durch langsam reagierende- und ausfallende Systeme, führen. Deshalb wurden zahlreiche Anstrengungen unternommen, um die Herausforderungen der Elastizität der Ressourcenallokation zu lösen. Bei der von Roy et al. (2011) beschriebenen Methode handelt es sich um eine schwellenwertbasierte Skalierung, bei der Leistungsmetriken regressiert und bei Erreichen eines Schwellenwerts hochskaliert werden. Empirische Ergebnisse konnten zeigen, dass die Methode zu einer präziseren Ressourcenallokation führt.

Prädiktive Lösungen, wie Modelle aus der Statistik und aus dem Bereich des maschinellen Lernens, lassen bessere Ergebnisse zu. Zu den Modellen gehören die Exponentielle Glättung, Weighted Moving Average, Autoregressive Modelle (AR) und ihre Abwandlungen z. B. ARMA, ARIMA (Calheiros et al. (2014)), Random Forest, Support Vector Machine, Gradient Boosting Tree und andere (Kumar & Thenmozhi (2006), Masini et al. (2023)). Obwohl diese statistischen und maschinellen Lernmethoden Zeitreihen von Arbeitslasten mit zyklischen oder saisonalen Trends modellieren können, machen sie suboptimale Vorhersagen für Arbeitslasten, die sich kontinuierlich und dynamisch verändern. Außerdem kann ein Modell für eine bekannte Art von Arbeitslast gut funktionieren, aber es kann oft nicht genau vorhersagen, wie sich andere bisher unbekannte Muster in der Zukunft entwickeln werden (Kim et al. (2016)).

Der Einsatz neuronaler Netze (NN) bei Zeitreihenprognosen kann die Genauigkeit in verschiedenen Bereichen erhöhen. Mit einer Abfolge nichtlinearer Schichten lernen die Modelle, wesentliche historische Daten aus einer Zeitreihe zu abstrahieren, um eine endgültige Prognose zu erstellen. LLong short-erm memoryModelle (LSTM) und seine Variationen werden erforscht, um die Ressourcennutzung oder Benutzeranfragen für Anwendungen vorherzusagen (Dang-Quang & Yoo (2022). Das LSTM ist jedoch von der Verwendung von Rekursionen abhängig, um lange/kurze Abhängigkeiten zu erfassen. Mit zunehmender Länge einer Eingabesequenz steigt auch die Schwierigkeit, solch lange Eingabesequenzen zu kodieren. Traditionelle vergessen bereits Informationen, die vor zehn Schritten auftraten. LSTM können sich jedoch an Information erinnern, welche über 1000 Schritte her ist (Gers et al. (2002)).

Aufgrund der jüngsten Entwicklungen auf dem Gebiet der Verarbeitung natürlicher Sprache kann der Aufmerksamkeitsmechanismus des Transformer-Netzwerks als Ersatz für Rekursionen oder Faltungen dienen (Vaswani et al. (2017)). Bei Zeitreihendaten ermöglicht der Aufmerksamkeitsmechanismus des Transformer-Netzwerks dem Modell, zeitliche Informationen in den Eingabesequenzen zu erkennen (Lim & Zohren (2021)).

Es ist Annehmbar, das Google Cloud Platform (GCP) und Amazon Web Service (AWS) „state of the art“ Lösungen für das Autoskalierungsproblem haben. Google verwendet ein eigens entwickeltes System zur Autoskalierung „Autopilot“. Rzadca et al. (2020) gehen leider nicht auf das verwendete Modell von „Autopilot“ ein.

1.5 Zielsetzung

In dieser Masterarbeit liegt das Hauptziel darin, verschiedene Algorithmen zu untersuchen und ihre Anwendung auf die Vorhersage der Nutzerlast in Web-Anwendungen zu evaluieren. Dabei soll ein breites Spektrum an Algorithmen verwendet werden, welche für die Voraussage von Zeitreihen geeignet sind. Durch diese Untersuchung soll ein tiefgreifendes Verständnis für die Leistung dieser Techniken und ihre möglichen Auswirkungen in der Praxis entwickelt werden.

Zusätzlich zur Analyse der Algorithmen wird ein weiteres Ziel dieser Arbeit darin bestehen, statistische Evaluationsmethoden anzuwenden, um die Leistung und Wirksamkeit der untersuchten Algorithmen zu bewerten. Dadurch soll ein umfassendes Bild der Stärken und Schwächen jedes Ansatzes entstehen, um letztlich fundierte Empfehlungen für ihre praktische Anwendung abgeben zu können.

Schließlich ist ein zentrales Anliegen dieser Arbeit, sich auf die Nutzerlast in digitalen

Büroanwendungen zu konzentrieren. Dadurch erhält die Untersuchung eine spezifische Anwendungsperspektive und ermöglicht eine realitätsnahe Evaluation. Das Ziel besteht darin, durch diese spezifische Betrachtung, wertvolle Erkenntnisse für die optimale Allokation von Ressourcen solcher Plattformen zu gewinnen und somit einen Beitrag zur Forschung und der Anwendung in der Praxis in diesem Bereich zu leisten.

2 Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte und Technologien erörtert, die im Rahmen dieser Arbeit relevant sind. Die Ausarbeitung stützt sich auf die Themen Zeitreihenanalyse, Web-Anwendungen, Nutzerlast und maschinelles Lernen, sowie das Konzept des "Digital Office“. Jeder dieser Aspekte spielt eine entscheidende Rolle in der Analyse und Vorhersage der Systemlast in einer Web-Anwendung, die in einem "Digital OfficeÜmfeld eingesetzt wird. Ziel ist es, ein solides Verständnis der theoretischen Grundlagen zu vermitteln, auf dem die nachfolgenden Abschnitte und die praktische Anwendung aufbauen.

Im Folgenden beginnen wir mit einer Diskussion über Zeitreihen und ihre Voraussage, da dies ein zentraler Punkt in unserer Untersuchung ist.

2.1 Zeitreihen und ihre Voraussage

Zeitreihen, als zentrales Element dieser Arbeit, stellen ein Thema dar, das einer detaillierten Betrachtung bedarf. Im Allgemeinen handelt es sich bei einer Zeitreihe um eine Sequenz von Datenpunkten, die in zeitlicher Abfolge gesammelt wurden.

Die Analyse von Zeitreihen besteht aus der Untersuchung von Datensequenzen, um Muster und Strukturen aufzudecken, die im Zeitverlauf verborgen sind. Dies beinhaltet das Identifizieren von Trends, also von allgemeinen Richtungen, in die sich die Daten bewegen, und saisonalen Schwankungen, die sich periodisch wiederholen. Eine weitere wichtige Komponente ist die Autokorrelation, das bedeutet, die Korrelation der Zeitreihe mit sich selbst, verschoben um eine bestimmte Anzahl an Zeiteinheiten (Box et al. (2015)).

In der Behandlung von Zeitreihen können wir zwischen stationären und nichtstationären Zeitreihen unterscheiden. Eine stationäre Zeitreihe hat über die Zeit hinweg konstante statistische Eigenschaften wie Mittelwert und Varianz, was sie relativ einfach in der Analyse macht. Eine nichtstationäre Zeitreihe hingegen zeigt wechselnde statistische Eigenschaften, was eine zusätzliche Herausforderung für die Analyse darstellt.

Die Modellierung von Zeitreihen befasst sich mit der Erstellung von mathematischen Modellen, die dazu dienen, die beobachteten Muster und Strukturen zu erklären und zukünftige Datenpunkte vorherzusagen. Es gibt eine Vielzahl von Modellen zur Zeitreihenanalyse,

von einfachen linearen Modellen bis hin zu komplexen nichtlinearen Modellen (Box et al. (2015)).

2.2 Web-Anwendungen und Nutzerlast

Web-Anwendungen kennzeichnen sich als grundlegende Technologien unserer modernen, digitalen Welt (Varshney et al. (2000)). Die Handhabung dieser Plattformen erfordert ein vertieftes Verständnis ihrer Natur und ihres Funktionierens. Grundsätzlich sind Web-Anwendungen Programme, die auf einem Webserver laufen und über einen Webbrowser zugänglich sind (Souders (2008)). Sie umfassen vielfältige Funktionen, die von einfachen Informationsseiten bis hin zu komplexen interaktiven Systemen reichen, einschließlich digitaler Büros, die eine zentrale Rolle in der modernen Arbeitswelt spielen.

In dem Kontext der Nutzerlast wird der Begriff verwendet, um den Grad der Beanspruchung einer Web-Anwendung durch ihre Nutzer zu beschreiben (Menascé (2002)). Sie umfasst Aspekte wie die Anzahl der gleichzeitigen Benutzer, die Häufigkeit und Art der von ihnen durchgeführten Aktionen und die Menge der durch diese Aktionen verarbeiteten Daten. Ein hohes Maß an Nutzerlast kann die Leistung einer Web-Anwendung beeinträchtigen, was zu langsameren Antwortzeiten und möglicherweise sogar zu Systemausfällen führen kann (Menascé (2002))).

Das Management der Nutzerlast ist von entscheidender Bedeutung, um die Leistungsfähigkeit und Zuverlässigkeit einer Web-Anwendung zu gewährleisten. Dies erfordert den Einsatz von Techniken wie dem Load Balancing, das die Nutzerlast gleichmäßig auf mehrere Server verteilt (Bourke (2001)), und der Autoskalierung, die die Ressourcen dynamisch anpasst, um die aktuelle Nutzerlast zu bewältigen (Han et al. (2014)).

Das Verständnis der Nutzerlast und ihre korrekte Vorhersage ist eine Schlüsselkomponente bei der Verwaltung von Web-Anwendungen. Eine genaue Vorhersage der Nutzerlast ermöglicht es, Ressourcen proaktiv bereitzustellen und freizugeben, um eine hohe Leistungsfähigkeit zu gewährleisten und gleichzeitig die Kosten zu optimieren (Han et al. (2014)). Dies ist besonders relevant in Cloud-Umgebungen, wo die Ressourcen auf Pay-per-Use-Basis bereitgestellt werden und eine Überallokation von Ressourcen zu erheblichen Kosten führen können.

2.3 Konzept und Nutzung des 'Digital Office'

Beginnend mit der Einführung des "Digital Office“Konzepts, wurde eine Plattform geschaffen, welche darauf abzielt, die Erfahrung von Remote-Mitarbeitern zu verbessern. Die Schaffung einer persönlicheren Umgebung steht dabei im Vordergrund.

Ermöglicht durch das "Digital Office“, können Gesprächsräume zu jeder Zeit betreten werden. Dieser kontinuierliche Zugang erleichtert die spontane Kommunikation und Interaktion mit Kollegen. Im Gegensatz zu anderen Produkten wie Microsoft Office, die auf vorher geplante Meetings angewiesen sind, eröffnet das "Digital Officenneue Möglichkeiten der Zusammenarbeit und Kommunikation.

Die Abschaffung des Prozesses der Meetingplanung steht dabei im Zentrum des "Digital Office“. Mit der Eliminierung dieses oft zeitaufwendigen und bürokratischen Schritts soll die Kommunikation zwischen Mitarbeitern vereinfacht und gefördert werden.

Die Nutzung des "Digital Office“hat sich als besonders wertvoll erwiesen, um die Herausforderungen der Remote-Arbeit zu bewältigen und eine stärkere Verbindung zwischen den Teammitgliedern zu schaffen. Infolgedessen ist es zu einem integralen Bestandteil vieler Unternehmen geworden, die die Vorteile der Digitalisierung nutzen und gleichzeitig ein Gefühl der Gemeinschaft und Zusammengehörigkeit unter ihren Mitarbeitern aufrechterhalten wollen.

Schlussendlich zeigt der Einsatz und die Nutzung des "Digital Office“, dass es Möglichkeiten gibt, die persönliche Interaktion und Zusammenarbeit in einem zunehmend digitalisierten Arbeitsumfeld zu fördern. Mit seiner einzigartigen Mischung aus Zugänglichkeit und Personalisierung bietet es eine praktikable Lösung für viele der Herausforderungen, die Remote-Arbeit mit sich bringt.

2.4 Verwendete Algortihmen

2.4.1 Naive Methode

Die naive Prognosemethode ist eine der einfachsten Prognosetechniken. Sie geht davon aus, dass der nächste Punkt in der Zeitreihe mit dem zuletzt beobachteten Punkt identisch sein wird. Trotz ihrer Einfachheit kann die Naive Methode als nützlicher Maßstab für komplexere Modelle dienen.

2.4.2 Saisonale Naive Methode

Die saisonale naive Vorhersagemethode erweitert die naive Methode durch Berücksichtigung der Saisonalität. Anstatt vorherzusagen, dass der nächste Punkt mit dem letzten Punkt übereinstimmt, sagt die saisonale Naive-Methode voraus, dass der nächste Punkt mit dem Punkt der vorherigen Saison oder des vorherigen Zyklus übereinstimmt. Dies kann bei Zeitreihendaten mit einem klaren saisonalen Muster nützlich sein.

2.4.3 Durchschnittsmethode

Die Durchschnittsvorhersagemethode sagt voraus, dass der nächste Punkt in der Zeitreihe der Durchschnitt aller zuvor beobachteten Punkte ist. Bei dieser Methode wird davon ausgegangen, dass die zugrunde liegende Zeitreihe stationär ist, d. h., dass sich ihre Eigenschaften im Laufe der Zeit nicht ändern. Trotz ihrer Einfachheit kann die Durchschnittsmethode für bestimmte Datentypen effektiv sein.

2.4.4 Saisonale Durchschnittsmethode

Die Methode der saisonalen Durchschnittsprognose ähnelt der Durchschnittsmethode, berücksichtigt aber die Saisonalität. Sie prognostiziert den nächsten Punkt in der Zeitreihe als Durchschnitt aller Punkte der entsprechenden Saison in früheren Zyklen. Bei Zeitreihendaten mit einem klaren saisonalen Muster kann diese Methode effektiver sein als die einfache Durchschnittsmethode.

2.4.5 Elastic Net

Elastic Net ist eine erweiterte lineare Regressionsmethode, die zwei wichtige Techniken kombiniert: Ridge Regression und Lasso. Es ist besonders nützlich für die Regression mit hohen Dimensionen, wo es viele korrelierende Merkmale gibt. Elastic Net verwendet einen Regularisierungsparameter, um die Anpassung zu glätten und gleichzeitig variable Auswahl durchzuführen. (Zou & Hastie (2005)) In der linearen Regression kann Overfitting ein Problem sein, wenn das Modell zu komplex ist im Vergleich zur Menge der verfügbaren Daten. Eine Methode zur Bekämpfung des Overfittings ist die Regularisierung, welche die Komplexität des Modells begrenzt. Bei der Ridge Regression wird eine L2-Regularisierung verwendet, während das Lasso eine L1-Regularisierung verwendet.

Ridge Regression und Lasso

Ridge Regression und Lasso sind zwei populäre Methoden zur Regularisierung in der linearen Regression. Ridge Regression hilft dabei, die Koeffizienten nahe Null zu halten, was die Modellkomplexität reduziert (Hoerl & Kennard (1970)). Lasso geht einen Schritt weiter und kann einige Koeffizienten genau auf Null setzen, was eine Art Feature-Auswahl bewirkt (Tibshirani (1996)).

Kombination von Ridge und Lasso: Elastic Net

Elastic Net kombiniert die Vorteile von Ridge Regression und Lasso, indem es eine Mischung aus L1- und L2-Regularisierung verwendet. Es behält die Feature-Auswahl-Eigenschaften von Lasso bei und behält gleichzeitig die Fähigkeit von Ridge Regression bei, mit korrelierenden Merkmalen umzugehen. Dies wird durch die Einführung eines Mischungsparameters erreicht, der bestimmt, in welchem Ausmaß jede Art von Regularisierung angewendet wird. (Zou & Hastie (2005))

Die Elastic-Net-Regression bietet so einen flexiblen Ansatz für die lineare Regression mit hohen Dimensionen und korrelierenden Merkmalen. Durch die Anpassung des Mischungsparameters kann der Benutzer zwischen Ridge und Lasso interpolieren, um das Modell auf die spezifischen Anforderungen der Daten anzupassen.

2.4.6 Prophet

Prophet ist ein maschinelles Lernverfahren, speziell entwickelt für die Vorhersage von Zeitreihen, das von Facebook entwickelt wurde. Seine Robustheit und Flexibilität macht es besonders geeignet für eine breite Palette von Geschäftsproblemen, da es verschiedene Funktionen zur Berücksichtigung von Trends, saisonalen Schwankungen sowie speziellen Ereignissen und Feiertagen bietet. (taylor2018forecasting)

Trend- und Saisonmodellierung

Das Kernstück von Prophet ist die additive Zeitreihenzerlegung, die es ermöglicht, sowohl Trend- als auch Saisonkomponenten zu modellieren. Der Trend, der sowohl linear als auch nicht linear sein kann, wird durch eine Change Point Analysisëingefangen, die in der Lage ist, Trendänderungen über die Zeit hinweg zu erkennen. Auf der anderen Seite ermöglicht die saisonale Modellierung, tägliche und wöchentliche Muster sowie jährliche Saisonalitäten bei Zeitreihen, die länger als ein Jahr sind, zu erfassen. Diese Modellierung nutzt Fourier-Reihen zur Erfassung von Saisonalitäten. (taylor2018forecasting)

Modellierung von Feiertagen und besonderen Ereignissen

Ein wesentliches Merkmal von Prophet ist seine Fähigkeit, Feiertage und spezifische Ereignisse in das Vorhersagemodell zu integrieren. Dies ist besonders relevant in Geschäftskontexten, in denen solche Ereignisse einen erheblichen Einfluss auf das Verhalten der Zeitreihen haben können.

2.4.7 Random Forest

Random Forest ist ein vielseitiges maschinelles Lernverfahren, das sowohl für Klassifikations- als auch für Regressionsaufgaben eingesetzt werden kann. Es basiert auf dem Konzept des Ensemble-Lernens, indem es mehrere Entscheidungsbäume erstellt und deren Vorhersagen kombiniert.

Funktionsweise von Random Forests

Ein Random Forest besteht aus einer Anzahl von Entscheidungsbäumen, die unabhängig voneinander erstellt werden. Jeder dieser Bäume wird auf einem Teil des Datensatzes trainiert, wobei sowohl die Beobachtungen als auch die Merkmale zufällig ausgewählt werden. Die Vorhersage des Random Forests ergibt sich dann durch Aggregation der Vorhersagen der einzelnen Bäume, normalerweise durch Mehrheitsvotum bei Klassifikation oder durch Mittelung bei Regression.

Stärken und Schwächen von Random Forests

Die Hauptstärken von Random Forests liegen in ihrer Fähigkeit, mit komplexen Datensätzen mit vielen Merkmalen und Klassen umzugehen, sowie in ihrer Robustheit gegenüber Overfitting. Sie können Merkmalsinteraktionen gut erfassen und benötigen in der Regel keine Vorverarbeitung der Daten, wie z.B. Skalierung oder Kodierung von kategorischen Variablen.

2.4.8 Extremely Randomized Trees (Extra Trees)

Extremely Randomized Trees, auch bekannt als Extra Trees, sind eine Erweiterung des Konzepts der Random Forests. Sie sind ein leistungsstarkes maschinelles Lernverfahren, das häufig für Klassifikations- und Regressionsaufgaben eingesetzt wird. Dabei zeichnen sie sich durch ihre Fähigkeit aus, komplexe Strukturen in Daten zu erfassen und dabei gegen Overfitting robust zu sein.

Grundprinzipien und Unterschiede zu Random Forests

Extra Trees verwenden das gleiche Grundprinzip wie Random Forests: Sie bauen mehrere Entscheidungsbäume auf und aggregieren ihre Ergebnisse. Der Hauptunterschied zu Random Forests besteht jedoch darin, wie diese Entscheidungsbäume aufgebaut werden. Während bei Random Forests für jede Aufteilung im Baum die bestmögliche Aufteilung aus einer zufälligen Teilmenge der Merkmale gewählt wird, werden bei Extra Trees sowohl die Merkmale als auch die Aufteilungspunkte zufällig gewählt.

Vorzüge der Extra Trees

Durch die zusätzliche Randomisierung können Extra Trees in einigen Fällen besser performen als Random Forests. Insbesondere können sie weniger anfällig für Overfitting sein, da sie weniger varianzbehaftet sind aufgrund der erhöhten Diversität der einzelnen Bäume. Zudem neigen sie dazu, weniger Zeit für das Training zu benötigen, da sie nicht nach dem besten Aufteilungspunkt suchen müssen. Extra Trees stellen daher eine attraktive Option für maschinelles Lernen mit hohen Dimensionen und komplexen Strukturen dar. Durch ihre erhöhte Randomisierung bieten sie eine robuste und effiziente Möglichkeit, aus Daten zu lernen und Vorhersagen zu treffen.

2.4.9 LightGBM

LightGBM ist ein maschinelles Lernverfahren, das für seine Effizienz und Genauigkeit bekannt ist. Es wurde von Microsoft entwickelt und stellt eine fortschrittliche Implementierung von Gradient Boosting Decision Trees (GBDT) dar. LightGBM zeichnet sich durch zwei Hauptinnovationen aus, die es von herkömmlichen GBDT-Methoden abheben: Gradient-based One-Side Sampling (GOSS) und Exclusive Feature Bundling (EFB). (Ke et al. (2017))

Gradient Boosting Decision Trees

GBDTs sind eine effektive Methode im maschinellen Lernen, die das Beste aus Entscheidungsbäumen und dem Boosting-Prinzip kombiniert. Entscheidungsbäume, die aufgrund ihrer klaren Struktur und Interpretierbarkeit beliebt sind, bilden eine Baumstruktur aus Entscheidungen, die auf den Datenmerkmalen basieren. Das Boosting-Prinzip verstärkt diese Entscheidungsstrukturen, indem es mehrere schwache Lernmodelle zu einem starken zusammenfügt, wobei jeder folgende Baum versucht, die Fehler des vorherigen zu korrigieren. GBDTs sind besonders nützlich bei komplexen Vorhersageaufgaben wie Zeitreihenprognosen, da sie durch diese iterative Fehlerkorrektur komplexere Datenbeziehungen erfassen können.

Mit Techniken wie GOSS und EFB, die in LightGBM verwendet werden, können GBDTs noch effizienter und genauer arbeiten. (Friedman (2001))

Gradient-based One-Side Sampling (GOSS)

In maschinellem Lernen und insbesondere beim Gradienten-Boosting kann das Arbeiten mit großen Datenmengen sowohl zeit- als auch rechenintensiv sein. GOSS adressiert dieses Problem, indem es die Menge der Daten, die zur Anpassung des Modells verwendet wird, reduziert, ohne die Modellgenauigkeit signifikant zu beeinträchtigen.

Bei der GOSS-Methode werden alle Beobachtungen mit hohen Gradienten (also Datenpunkten, die schwer vorherzusagen sind) beibehalten, während eine zufällige Stichprobe von Beobachtungen mit niedrigen Gradienten durchgeführt wird. Durch diese Technik wird die Geschwindigkeit der Modellanpassung erhöht, ohne dass signifikante Informationen verloren gehen, die zur Genauigkeit des Modells beitragen.

Exclusive Feature Bundling (EFB)

EFB ist eine weitere Innovation von LightGBM, die zur Reduzierung der Dimensionalität der Daten dient. Bei vielen realen Datensätzen treten viele Features nicht gleichzeitig auf, d.h., sie sind exklusiv, wie beispielsweise One-Hot-Kodierte Merkmale. Mit EFB können solche exklusiven Features gebündelt werden, wodurch die Anzahl der Features effektiv reduziert wird.

Diese Bündelung von exklusiven Features ermöglicht es LightGBM, die Rechenlast erheblich zu reduzieren, ohne dass die Modellgenauigkeit signifikant beeinträchtigt wird.

Zusammenfassend bietet LightGBM eine effiziente und hochpräzise Methode für maschinelles Lernen, insbesondere für Zeitreihenprognosen. Durch seine innovativen Ansätze, wie GOSS und EFB, kann es effizient große Mengen von Daten handhaben und dabei eine hohe Vorhersagegenauigkeit aufrechterhalten.

2.4.10 TCNForecaster

Der TCNForecaster ist ein speziell für Zeitreihendaten entwickeltes Modell, das auf einem temporalen Faltungsnetzwerk (Temporal Convolutional Network, TCN) basiert. Dieses Netzwerk verwendet historische Daten und zugehörige Merkmale, um Vorhersagen für eine Zielmenge bis zu einem bestimmten Vorhersagehorizont zu treffen. Der TCNForecaster besteht aus mehreren Hauptkomponenten, darunter eine Vor-Mischschicht, die die Eingabezeitreihen und Merkmalsdaten vermischt, eine Stapelung von dilatierten Faltungsschichten,

die die Kanäle verarbeiten, und eine Sammlung von Prognosekopfeinheiten, die die Ausgangssignale der Faltungsschichten zusammenführen und Prognosen für die Zielmenge aus dieser latenten Darstellung erzeugen. Dilatierte Faltungen erlauben es, Informationen über einen größeren Bereich der Eingangsdaten zu erfassen, ohne die Anzahl der Parameter zu erhöhen. Sie erreichen dies durch SSprüngeßwischen den gefalteten Eingangsdaten, die durch einen "Dilatationsfaktor“bestimmt werden. (Bai et al. (2018); Chang et al. (2017))

Die zentrale Operation eines TCN ist eine dilatierte, kausale Faltung entlang der Zeitdimension eines Eingangssignals. Diese Faltung mischt Werte von nahegelegenen Zeitpunkten im Eingangssignal. Die Proportionen in der Mischung sind der Kern oder die Gewichte der Faltung, während die Trennung zwischen den Punkten in der Mischung die Dilatation ist. Ein kausaler Faltung mischt nur Eingangswerte aus der Vergangenheit im Verhältnis zu jedem Ausgangspunkt, wodurch verhindert wird, dass die Ausgabe in die Zukunft ßieht“. (Bai et al. (2018))

Die Architektur des TCNForecaster besteht aus einem Stapel von Faltungsschichten zwischen der Vor-Mischschicht und den Prognoseköpfen. Dieser Stapel ist logisch in wiederholende Einheiten unterteilt, die als Blöcke bezeichnet werden und aus residualen Zellen bestehen. Eine residuale Zelle wendet kausale Faltungen zusammen mit Normalisierung und nicht-linearer Aktivierung an. Jede residuale Zelle fügt ihrem Eingang ihren Ausgang hinzu, was als residuale Verbindung bezeichnet wird. Diese Verbindungen sind bekannt dafür, dass sie das Training von tiefen neuronalen Netzwerken (DNN) verbessern, möglicherweise weil sie einen effizienteren Informationsfluss durch das Netzwerk ermöglichen. (Bai et al. (2018))

2.5 Statistische Evaluationsmethoden

Die Verwendung statistischer Evaluationsmethoden ist entscheidend für die Messung der Leistungsfähigkeit der Algorithmen zur Vorhersage der Nutzerlast. Diese Methoden ermöglichen eine objektive und quantifizierbare Bewertung der Leistung, was besonders wichtig ist, wenn mehrere Algorithmen miteinander verglichen werden. In den folgenden Formeln stellt y_i die tatsächlichen Werte und $\hat{y}_i$ die vorhergesagten Werte dar.

2.5.1 Mittlerer absoluter Fehler (MAE)

Der mittlere absolute Fehler (MAE) ist eine gängige Metrik zur Messung der Vorhersagegenauigkeit eines Modells. Der MAE berechnet den durchschnittlichen absoluten Unterschied zwischen den tatsächlichen und den vorhergesagten Werten (Chai & Draxler (2014)). Ein

kleinerer MAE-Wert weist auf eine höhere Vorhersagegenauigkeit hin. Die Formel für den MAE ist:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.1}$$

2.5.2 Root Mean Square Error (RMSE)

Der Root Mean Square Error (RMSE) ist eine weitere gängige Metrik zur Messung der Vorhersagegenauigkeit eines Modells. Im Gegensatz zum MAE berücksichtigt der RMSE die Quadratwurzel des durchschnittlichen quadratischen Fehlers, wodurch er stärker auf größere Fehler reagiert und diese stärker bestraft (Chai & Draxler (2014)). Die Formel für den RMSE ist:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.2}$$

2.5.3 Mean Absolute Percentage Error (MAPE)

Der Mean Absolute Percentage Error (MAPE) ist eine Metrik, die den mittleren absoluten prozentualen Fehler zwischen den tatsächlichen und den vorhergesagten Werten berechnet. Im Gegensatz zum MAE und RMSE, die absolute Fehlermaße sind, ist der MAPE ein relatives Fehlermaß, das den Fehler in Bezug auf die tatsächlichen Werte ausdrückt (De Myttenaere et al. (2016)). Die Formel für den MAPE ist:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{2.3}$$

2.5.4 Median Absolute Error

Der Median Absolute Error ist ähnlich zum Mean Absolute Error (Chai & Draxler (2014)), verwendet aber den Median anstelle des Durchschnitts, was ihn robuster gegenüber Ausreißern macht. Die Formel ist:

$$MedAE = median(|y_1 - \hat{y}_1|, ..., |y_n - \hat{y}_n|) \tag{2.4}$$

2.5.5 Root Mean Squared Logarithmic Error

Der Root Mean Squared Logarithmic Error (RMSLE) ist eine Variante des RMSE, bei der die logarithmischen Werte der tatsächlichen und vorhergesagten Werte verwendet werden.

Dadurch wird der Einfluss von großen Fehlern verringert (developers (o. J.)). Die Formel ist:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(y_i + 1) - log(\hat{y}_i + 1))^2} \tag{2.5}$$

2.5.6 Explained Variance Score

Der Explained Variance Score ist ein statistisches Maß, das erklärt, wie gut ein Modell die Struktur eines Datensatzes erfasst. Ein höherer Score zeigt an, dass das Modell besser in der Lage ist, die Varianz in den Daten zu erklären (Achen (1990)). Die Formel für den Explained Variance Score ist:

$$\text{Explained Variance Score} = 1 - \frac{Var(y - \hat{y})}{Var(y)} \tag{2.6}$$

2.5.7 R2 Score

Der R2 Score, auch als Bestimmtheitsmaß bekannt, gibt den Anteil der Varianz in den abhängigen Variablen an, der vorhersehbar ist aus den unabhängigen Variablen (Achen (1990)). Die Formel ist:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}i)^2}{\sum i = 1^n(y_i - mean(y))^2} \tag{2.7}$$

2.5.8 Spearman Korrelation

Die Spearman-Korrelation misst die stärke und die Richtung des Zusammenhangs zwischen zwei Variablen (Spearman (1961)). Im Gegensatz zur Pearson-Korrelation, die einen linearen Zusammenhang misst (Cohen et al. (2009)), misst die Spearman-Korrelation monotone Beziehungen, ob linear oder nicht. Der Wertebereich liegt zwischen -1 und +1. Ein Wert nahe -1 bedeutet eine starke negative Korrelation, während ein Wert nahe +1 eine starke positive Korrelation bedeutet. Die Formel ist:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2 - 1)} \tag{2.8}$$

Bei der Berechnung der Spearman'schen Rangkorrelation wird jedes Paar von Datenpunkten mit seinen Rängen anstelle seiner tatsächlichen Werte betrachtet (Spearman (1961)). Hier ist d_i die Differenz zwischen den Rängen von y_i und $\hat{y}_i$.

2.6 Herausforderungen und Auswirkungen

Die Herausforderungen und Auswirkungen bei der Vorhersage der Nutzerlast und der Ressourcenverwaltung sind vielfältig und komplex. Eine der größten Herausforderungen besteht darin, dass die Nutzerlast hochdynamisch und schwer vorhersehbar ist. Sie kann von zahlreichen Faktoren beeinflusst werden, einschließlich des Nutzerverhaltens, der Tageszeit und spezieller Ereignisse. Ein fehlender Einblick in diese Dynamik kann zu ungenauen Vorhersagen führen, die wiederum die Effizienz der Ressourcenverwaltung beeinträchtigen können (Mao & Humphrey (2011)).

Die Unvorhersehbarkeit der Nutzerlast führt auch zu technischen Herausforderungen. Die Bereitstellung ausreichender Ressourcen zur Bewältigung der Spitzenlast kann zu Überprovisionierung führen, die zu unnötigen Kosten führt. Andererseits kann die Nichtbereitstellung ausreichender Ressourcen zur Bewältigung der Last zu Unterprovisionierung führen, die wiederum die Leistung und Zuverlässigkeit der Web-Anwendung beeinträchtigen kann (Mao & Humphrey (2011)). Darüber hinaus können herkömmliche Methoden zur Ressourcenverwaltung, oft nicht mit der Dynamik und Unvorhersehbarkeit der Nutzerlast umgehen (Fernandez et al. (2014)).

Die Auswirkungen dieser Herausforderungen können erheblich sein. Eine schlechte Nutzererfahrung, die durch eine langsame Reaktionszeit oder Systemausfälle verursacht wird, kann das Vertrauen der Nutzer in die Web-Anwendung untergraben und sie dazu veranlassen, zu Konkurrenzprodukten überzugehen. Auf der anderen Seite können hohe Kosten durch Überprovisionierung die finanzielle Nachhaltigkeit der Web-Anwendung beeinträchtigen (Mao & Humphrey (2011)). Daher ist die Entwicklung effizienter Algorithmen zur Vorhersage der Nutzerlast und zur Optimierung der Ressourcenverwaltung von entscheidender Bedeutung.

3 Daten

3.1 Datenquelle

3.2 Auswahl der Daten und Datenaufbereitung

The most common practice is to implement fixed infrastructure level Central Processing Unit (CPU) based autoscaling rules to scale up or scale down the number of container instances(replicas) assigned to a specific service depending on demand and workload Evangelidis, Alexandros, Parker, David, and Bahsoon, Rami. “Performance Modelling and Verification of CloudBased AutoScaling Policies”. In: 2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID) (2017). DOI: 10.1109/ccgrid.2017.39. URL: https://core.ac.uk/ download/pdf/185505413.pdf

3.2.1 Nonstationary and Seasonal Time Series Models

`https://link.springer.com/chapter/10.1007/0-387-21657-X_6`

3.3 Statistische Tests zur Beschreibung der Daten

Augmented-Dickey Fuller Test (stationary or not)

Kwiatkowski-Phillips-Schmidt-Shin (stationary or not)

The Ljung-Box Test (evidence of autocorrelation)

3.4 Featurization

https://learn.microsoft.com/en-us/azure/machine-learning/concept-automl-forecasting-calendar-features?view=azureml-api-2

3.5 Weitere Verarbeitungsschritte

quelle dafür, weshalb 3-5 minuten in die zukunft voraussgesagt wird -> figure 8 Mao, M., & Humphrey, M. (2011). Ä performance study on the VM startup time in the cloud.Ïn 2011 IEEE Fifth International Conference on Cloud Computing.

4 Methodik

4.1 Bayesian Optimization

https://medium.com/microsoftazure/a-review-of-azure-automated-machine-learning-automl-5d2f98512406

4.2 Probabilistic Matrix

4.3 Evaluationsmethoden und Kriterien

5 Algorithmen und Hyperparameter

5.1 ARIMA

5.1.1 Hyperparameter

5.2 Regressionen

5.2.1 Hyperparameter

5.3 Neuronale Netzwerke

5.3.1 Hyperparameter

6 Ergebnisse und Diskussion

6.1 Vergleich der Ergebnisse

6.2 Diskussion der Ergebnisse

6.3 Limitationen und mögliche Verbesserungen

7 Fazit und Ausblick

(...)

Literaturverzeichnis

Achen, C. H. (1990). What does “explained variance “explain?: Reply. *Political Analysis*, *2*, 173–184.

Al-Dhuraibi, Y., Paraiso, F., Djarallah, N. & Merle, P. (2017). Elasticity in cloud computing: state of the art and research challenges. *IEEE Transactions on Services Computing*, *11* (2), 430–447.

Bai, S., Kolter, J. Z. & Koltun, V. (2018). An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*.

Bourke, T. (2001). *Server load balancing*. Ö'Reilly Media, Inc.
.

Box, G. E., Jenkins, G. M., Reinsel, G. C. & Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

Burns, B., Beda, J., Hightower, K. & Evenson, L. (2022). *Kubernetes: up and running*. Ö'Reilly Media, Inc.
.

Calheiros, R. N., Masoumi, E., Ranjan, R. & Buyya, R. (2014). Workload prediction using arima model and its impact on cloud applications' qos. *IEEE transactions on cloud computing*, *3* (4), 449–458.

Chai, T. & Draxler, R. R. (2014). Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. *Geoscientific model development*, *7* (3), 1247–1250.

Chang, S., Zhang, Y., Han, W., Yu, M., Guo, X., Tan, W., ... Huang, T. S. (2017). Dilated recurrent neural networks. *Advances in neural information processing systems*, *30*.

Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., ... Cohen, I. (2009). Pearson correlation coefficient. *Noise reduction in speech processing*, 1–4.

Dang-Quang, N.-M. & Yoo, M. (2022). An efficient multivariate autoscaling framework using bi-lstm for cloud computing. *Applied Sciences*, *12* (7), 3523.

De Myttenaere, A., Golden, B., Le Grand, B. & Rossi, F. (2016). Mean absolute percentage error for regression models. *Neurocomputing*, *192*, 38–48.

developers, S. (o. J.). *3.3.4.5. mean squared logarithmic error.* `https://scikit-learn.org/0.22/modules/model_evaluation.html`. (Accessed: 2023-05-25)

Fernandez, H., Pierre, G. & Kielmann, T. (2014). Autoscaling web applications in heterogeneous cloud infrastructures. In *2014 ieee international conference on cloud engineering* (S. 195–204).

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Gers, F. A., Schraudolph, N. N. & Schmidhuber, J. (2002). Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, *3* (Aug), 115–143.

Han, R., Ghanem, M. M., Guo, L., Guo, Y. & Osmond, M. (2014). Enabling cost-aware and adaptive elasticity of multi-tier cloud applications. *Future Generation Computer Systems*, *32*, 82–98.

Hoerl, A. E. & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, *12* (1), 55–67.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... Liu, T.-Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, *30*.

Kim, I. K., Wang, W., Qi, Y. & Humphrey, M. (2016). Empirical evaluation of workload forecasting techniques for predictive cloud resource scaling. In *2016 ieee 9th international conference on cloud computing (cloud)* (S. 1–10).

Kumar, M. & Thenmozhi, M. (2006). Forecasting stock index movement: A comparison of support vector machines and random forest. In *Indian institute of capital markets 9th capital markets conference paper.*

Lim, B. & Zohren, S. (2021). Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, *379* (2194), 20200209.

Mao, M. & Humphrey, M. (2011). Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In *Proceedings of 2011 international conference for high performance computing, networking, storage and analysis* (S. 1–12).

Masini, R. P., Medeiros, M. C. & Mendes, E. F. (2023). Machine learning advances for time series forecasting. *Journal of economic surveys*, *37* (1), 76–111.

Menascé, D. A. (2002). Load testing of web sites. *IEEE internet computing*, *6* (4), 70–74.

Qu, C., Calheiros, R. N. & Buyya, R. (2018). Auto-scaling web applications in clouds: A taxonomy and survey. *ACM Computing Surveys (CSUR)*, *51* (4), 1–33.

Roy, N., Dubey, A. & Gokhale, A. (2011). Efficient autoscaling in the cloud using predictive models for workload forecasting. In *2011 ieee 4th international conference on cloud computing* (S. 500–507).

Rzadca, K., Findeisen, P., Swiderski, J., Zych, P., Broniek, P., Kusmierek, J., ... others (2020). Autopilot: workload autoscaling at google. In *Proceedings of the fifteenth european conference on computer systems* (S. 1–16).

Souders, S. (2008). High-performance web sites. *Communications of the ACM*, *51* (12), 36–41.

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

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58* (1), 267–288.

Varshney, U., Vetter, R. J. & Kalakota, R. (2000). Mobile commerce: A new frontier. *Computer*, *33* (10), 32–38.

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

Wang, T., Ferlin, S. & Chiesa, M. (2021). Predicting cpu usage for proactive autoscaling. In *Proceedings of the 1st workshop on machine learning and systems* (S. 31–38).

Zou, H. & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology)*, *67* (2), 301–320.

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe. Zudem versichere ich, dass ich die von mir vorgelegte Arbeit selbständig abgefasst habe, dass ich keine weiteren Hilfsmittel verwendet habe als diejenigen, die im Vorfeld explizit zugelassen und von mir angegeben wurden und dass die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Des Weiteren habe ich die Stellen der Arbeit, die dem Wortlaut oder dem Sinn nach anderen Werken (dazu zählen auch Internetquellen und KI-basierte Tools) entnommen sind, unter Angabe der Quelle kenntlich gemacht und ich bestätige, dass ich die vorliegende Arbeit noch nicht für andere Prüfungen eingereicht habe.

Mir ist bewusst, dass ich diese Prüfung nicht bestanden habe, wenn ich die mir bekannte Frist für die Einreichung meiner schriftlichen Arbeit versäume und dass ich im Falle eines Täuschungsversuchs diese Prüfung nicht bestanden habe. Ebenso ist mir bewusst, dass ich im Falle eines schwerwiegenden Täuschungsversuchs gegebenenfalls die Gesamtprüfung endgültig nicht bestanden habe und in diesem Studiengang bzw. Studienangebot nicht mehr weiter studieren darf. Sollte ich zur Erstellung dieser Arbeit KI-basierter Tools verwendet haben, trage ich die Verantwortung für eventuell durch die KI generierte fehlerhafte oder verzerrte (bias) Inhalte, fehlerhafte Referenzen, Verstöße gegen das Datenschutz- und Urheberrecht oder Plagiate.

Berlin, den 01. Januar 2099

. . . . . .

(Unterschrift des Verfassers)