

Sebastian Steindl

Aus der Reihe: e-fellows.net stipendiaten-wissen

e-fellows.net (Hrsg.)

Band 3547

Integration von python-basierten Vorhersagemodellen in ein Monitoring-Webdashboard

Bachelorarbeit

BEI GRIN MACHT SICH IHR WISSEN BEZAHLT



- Wir veröffentlichen Ihre Hausarbeit, Bachelor- und Masterarbeit
- Ihr eigenes eBook und Buch - weltweit in allen wichtigen Shops
- Verdienen Sie an jedem Verkauf

Jetzt bei www.GRIN.com hochladen
und kostenlos publizieren



Bibliografische Information der Deutschen Nationalbibliothek:

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de/> abrufbar.

Dieses Werk sowie alle darin enthaltenen einzelnen Beiträge und Abbildungen sind urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsschutz zugelassen ist, bedarf der vorherigen Zustimmung des Verlanges. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen, Auswertungen durch Datenbanken und für die Einspeicherung und Verarbeitung in elektronische Systeme. Alle Rechte, auch die des auszugsweisen Nachdrucks, der fotomechanischen Wiedergabe (einschließlich Mikrokopie) sowie der Auswertung durch Datenbanken oder ähnliche Einrichtungen, vorbehalten.

Impressum:

Copyright © 2020 GRIN Verlag
ISBN: 9783346227003

Dieses Buch bei GRIN:

<https://www.grin.com/document/899477>

Sebastian Steindl

Aus der Reihe: e-fellows.net stipendiaten-wissen

e-fellows.net (Hrsg.)

Band 3547

Integration von python-basierten Vorhersagemodellen in ein Monitoring-Webdashboard

GRIN - Your knowledge has value

Der GRIN Verlag publiziert seit 1998 wissenschaftliche Arbeiten von Studenten, Hochschullehrern und anderen Akademikern als eBook und gedrucktes Buch. Die Verlagswebsite www.grin.com ist die ideale Plattform zur Veröffentlichung von Hausarbeiten, Abschlussarbeiten, wissenschaftlichen Aufsätzen, Dissertationen und Fachbüchern.

Besuchen Sie uns im Internet:

<http://www.grin.com/>

<http://www.facebook.com/grincom>

http://www.twitter.com/grin_com

Ostbayerische Technische Hochschule Amberg-Weiden
Fakultät Elektrotechnik, Medien und Informatik

Studiengang Angewandte Informatik

Bachelorarbeit

von

Sebastian Steindl

**Integration von Python-basierten Vorhersagemodellen in
ein Monitoring-Webdashboard**

Integration of Python-based predictive models into a
monitoring webdashboard

Inhaltsverzeichnis

1	Einleitung	1
1.1	Ausgangssituation	1
1.2	Zielsetzung	2
1.3	Gliederung der Arbeit	2
2	Einführung in die Zeitreihenanalyse in der Statistik	4
2.1	Definition einer Zeitreihe	4
2.1.1	Arithmetisches Mittel	4
2.1.2	Erwartungswert	5
2.1.3	Varianz	5
2.1.4	Kovarianz	5
2.2	Stationäre Prozesse	6
2.3	Das Komponentenmodell	7
2.3.1	Trendbereinigung	7
2.3.2	Saisonbereinigung	8
2.4	AR-, MA- und ARMA-Modelle	8
2.4.1	AR(p)-Modelle	9
2.4.2	MA(q)-Modelle	11
2.5	ARMA(p,q)-Modelle	11
2.6	ARIMA(p,d,q)-Modelle	12
2.6.1	Vorgehensweise zur Zeitreihenvorhersage mit ARIMA-Modellen	14
3	Python für Machine-Learning und Data Science	15
3.1	Relevante Python-Bibliotheken	16
3.1.1	Numpy	16
3.1.2	Pandas	16
3.1.3	statsmodels	16
3.1.4	Matplotlib	16
3.1.5	Tensorflow	16
3.1.6	Keras	17
3.1.7	Scikit-learn	17
4	Der Data Science Prozess	18
4.1	Fragestellung	19
4.2	Datenbeschaffung (1)	19

4.3	Datenexploration (1)	19
4.4	Datenbeschaffung (2)	19
4.5	Datenexploration (2)	20
4.5.1	Rücksprache mit Datenbankadministrator	22
4.6	Datenbeschaffung (3)	22
4.7	Datenexploration (3)	23
4.7.1	Vereinheitlichung der Timestamps	23
4.7.2	Merkmale als Spalten	24
4.7.3	Keine fehlenden Merkmalswerte	24
4.8	Datenmodellierung	25
4.9	Visualisierung	25
5	Grundlagen für die Prognosenerstellung mit Machine-Learning	27
5.1	Was ist Machine-Learning?	27
5.2	Statistische Methoden oder Machine-Learning?	27
5.3	Machine-Learning-Klassen	28
5.3.1	Supervised Learning	28
5.3.2	Unsupervised Learning	28
5.3.3	Reinforcement Learning	29
5.3.4	Entscheidung für eine Klasse	29
5.4	Künstliche Neuronale Netze	29
5.4.1	Das Perzeptron nach Rosenblatt	30
5.4.2	Aufbau von KNNs	31
5.4.3	Lernen mittels Gradientenabstiegsverfahren	34
5.4.4	Backpropagation	37
5.5	Über- und Unteranpassung eines Künstlichen Neuronalen Netzes	38
5.6	Regularisierung	39
5.6.1	Dropout-Regularisierung	40
5.6.2	Regularisierung durch Early Stopping	40
5.7	Standardisierung	41
5.8	Univariate und multivariate Vorhersagen	42
6	Zeitreihenprognose mit Neuronalen Netzen	43
6.1	Evaluation verschiedener Architekturen Neuronaler Netze	43
6.1.1	Multilayer Perceptron (MLP)	43
6.1.2	Convolutional Neural Network (CNN)	44
6.1.3	Recurrent Neural Network(RNN)	46
6.1.4	Long Short Term Memory (LSTM)	47
6.1.5	Gated Recurrent Unit (GRU)	50
6.2	Datenvorbereitung für multivariate KNNs	52
6.2.1	Featuregenerierung aus den Zeitstempeln	52
6.2.2	Hauptkomponentenanalyse (PCA)	52
6.2.3	Ablauf der Datenvorbereitung	53
6.3	Implementierung der KNNs	54
6.3.1	Grundlegende Designentscheidungen	55
6.3.2	Implementierung des MLP	57

6.3.3	Implementierung des CNN	57
6.3.4	Implementierung des LSTM- und GRU-Netzes	59
6.4	Modelloptimierung	59
6.4.1	Kreuzvalidierung	60
6.4.2	Rastersuche	61
6.4.3	Ergebnisse der Hyperparameteroptimierung	62
6.5	Trainings-, Validierungs- und Testdaten	62
6.6	Modellbewertung	63
6.6.1	Möglichkeiten zur Fehlerbestimmung	63
6.6.2	Fehlerauswertung der erstellten Modelle	65
6.6.3	Interpretation der Fehlerauswertung	66
6.6.4	Entscheidung für ein Modell	67
6.6.5	Modell für zweite Serverinstanz	68
7	Möglichkeiten zur Integration der Prognosen in die Webanwendung	72
7.1	Durchführen des Machine-Learning im Browser	72
7.1.1	Tensorflow.js	72
7.1.2	WebAssembly	73
7.1.3	Iodide	73
7.1.4	Pyodide	74
7.2	Bereitstellung der Prognosewerte	74
7.3	Entscheidung für einen Ansatz	75
8	Umsetzung des gewählten Ansatzes	77
8.1	Die Webanwendung DC-Cubes	77
8.2	Verwendete Technologien	77
8.2.1	React	78
8.2.2	D3.js	78
8.2.3	three.js	79
8.2.4	Solr	79
8.2.5	Node.js	80
8.3	Backend-Architektur	80
8.4	Konzept zur Bereitstellung der Prognosewerte	80
8.5	Visualisierung der Prognosen in der Webanwendung	80
8.5.1	Ausgangssituation	81
8.5.2	Erweiterung um Prognose	81
8.6	Technische Umsetzung	82
8.6.1	Speichern der historischen Daten	82
8.6.2	Erzeugen und Speichern der Prognosen	82
8.6.3	Laden der Prognose in die Webanwendung	83
8.6.4	Neue Prognosen erstellen	83
8.6.5	Anzeigen der Prognose aktivieren	84
8.6.6	Anzeigen der Prognose im 2d-Chart	84
8.6.7	Anzeigen der Prognose im 3d-Chart	85
8.6.8	Merge-Core	85

9 Zusammenfassung und Ausblick	86
9.1 Zusammenfassung dieser Arbeit	86
9.2 Lebenszyklus der Prognosemodelle	87
9.3 Ausblick	87
Literaturverzeichnis	89
Abbildungsverzeichnis	97
Tabellenverzeichnis	99
Listingverzeichnis	100
A Abbildungen	101
B Tabellen	107
C Listings	109

Glossar

ACF: Autocorrelationfunction. **10**, 11 API: Application Programming Interface. 17, 73, 75
AR: Autoregressive. 9, 14, 11
ARIMA: Autoregressive integrated moving average. 45, 4, 12, 14
ARMA: Autoregressive moving average. 8, 11, 12, 14, 28
AWS: Amazon Web Services. 75
CNN: Convolutional Neural Network. **43**, 44, 45, 53, 54, 57, 58, 62, 65, 66, 67, 68, 69, 71, 86
CPU: Central processing unit. 4, 19, 22, 25, 69, 73, 77
CRUD: Create, Read, Update, Delete. 79
CSS: Cascading Style Sheets. 73
DOM: Document Object Model. 78
DSV: Delimiter-separated values. 16, 20
GPU: Graphics processing unit. 37, 73
GRU: Gated recurrent unit. **50**, 51, 53, 54, 59, 62, 65, 66, 67
HTML: Hypertext Markup Language. 73, 74, 78
HTTP: Hypertext Transfer Protocol. 82, 83, 84
IT: Informationstechnologie. 1, 2, 25, 77, 87
KNN: Künstliches Neuronales Netz. **29**, 31, 32, 34, 35, 37, 38, 39, 42, 43, 44, 52, 53, 54, 55, 57, 59, 62
LSTM: Long Short Term Memory. 45, **47**, 48, 49, 50, 51, 53, 54, 58, 59, 60, 62, 65, 66, 67, 86
MA: Moving average. 8, **11**, 12, 14
OCR: Optical character recognition. 28, 75
PACF: Partial Autocorrelationfunction. **10**
PCA: Principal components analysis. **52**, 53, 54, 65, 82, 86
REST: Representational State Transfer. 80, 83
RNN: Rekursives Neuronales Netz. 38, 45, **46**, 47, 50, 74
SVG: Scalable vector graphics. 78, 79, 84

Kapitel 1

Einleitung

Dieses Kapitel stellt zunächst die Ausgangssituation dar, formuliert dann die daraus entstehende Zielsetzung dieser Arbeit und legt schließlich die Gliederung der Arbeit dar.

1.1 Ausgangssituation

Rechenzentren spielen seit eine zentrale Rolle bei Organisationen und Unternehmen, um diverse Dienste zur Verfügung zu stellen und Daten zu speichern. Mit der in den letzten Jahrzehnten steigenden Bedeutung des Internets und der IT wuchsen auch die Datenmengen und die Bedeutung der Daten nahm zu. Gleichzeitig erhöhten sich damit die Ansprüche an die Netzwerkstrukturen, die diese Daten speichern und zur Verfügung stellen. Diese Server sollen möglichst ununterbrochen erreichbar sein und immer höhere Geschwindigkeiten ermöglichen [42]. Bei der Erreichbarkeit besteht für den reibungslosen Ablauf eines Betriebs meist der Anspruch der sogenannten „Hochverfügbarkeit“. Das heißt nach [80], dass die Server nahezu stetig erreichbar sein und die Ausfallquote weniger als ein Prozent betragen soll. Weiter weist der Autor darauf hin, dass je größer die Anzahl der verwendeten Geräte ist, desto komplizierter und teurer ist es auch, dieses Ziel zu erreichen. Zudem ist der Betrieb eines Rechenzentrums an sich mit nicht vernachlässigbaren Kosten verbunden. Diese setzen sich hauptsächlich aus den Aufwendungen für die Hardware und den Stromkosten zusammen [36]. Daher ist es auch aus dieser Sicht für Großorganisationen mit großen Rechenzentren von wirtschaftlichem Interesse, die Infrastruktur effizient und mit geringen Ausfallzeiten zu betreiben. Nach [36] wird dies aber nur selten erreicht. Dementsprechend ist die Überwachung und Optimierung von Rechenzentren von großer Bedeutung.

Der für diese Arbeit vorliegende Datensatz stammt von der Bundesagentur für Arbeit. Mit rund 9000 Servern, von denen die Hälfte virtuell ist, die in zwei redundanten Rechenzentren betrieben werden, liegt eine der größten IT-Landschaften Deutschlands vor [3].

1.2 Zielsetzung

Technologien aus den Bereichen der Künstlichen Intelligenz und Data Science können bei mehreren Aspekten eines Rechenzentrums eine Unterstützung sein. Laut [51] konnte Google durch deren Einsatz die Energiekosten für ihre Rechenzentren um 40% senken. Ein durch statistische Vorhersagen angereichertes Monitoring kann außerdem zu erhöhter Effizienz führen und dabei helfen, Ausfallzeiten zu minimieren [1]. Im Zuge dieser Arbeit sollen basierend auf realen Belastungsdaten mittels Machine Learning Prognosen über die zukünftige Auslastung eines Servers getroffen werden. Die Daten stammen dabei aus der IT-Struktur der Bundesagentur für Arbeit. Diese Vorhersagen können als Grundlage zur Optimierung und Verhinderung von Netzwerkausfällen dienen. Dafür ist es nötig, die Ergebnisse zu visualisieren. So können die Prognosen vom Menschen besser verstanden und analysiert werden und es wird eine einfachere Kommunikation ermöglicht [86, S. 1 f.]. Diese Visualisierung soll in eine prototypische Web-Anwendung, die zum Monitoring der oben genannten IT-Struktur konzipiert ist, integriert werden. Dabei soll kenntlich gemacht werden, ob der Benutzer gerade Monitoringdaten der Vergangenheit oder die prognostizierten Werte betrachtet. Nach dem Lesen dieser Arbeit kennt der Leser Grundlagen zur Zeitreihenvorhersage in der Statistik und für die Prognosenerstellung mit tiefen Künstlichen Neuronalen Netzen. Er kennt Architekturen beziehungsweise Arten von Neuronalen Netzen, die sich für die Verarbeitung von Sequenzen, zu denen Zeitreihen gehören, anbieten sowie deren besonderen Eigenschaften, die sie dafür geeignet machen. An einem konkreten Datensatz kann er den Prozess zur Prognosenerstellung nachverfolgen. Schließlich lernt er Leser Möglichkeiten kennen, mit denen man die Zeitreihenvorhersagen in eine Webanwendung integrieren kann. Dem Leser wird eine Implementierung der Integration sowie die verwendete Visualisierung in der Webanwendung vorgestellt.

1.3 Gliederung der Arbeit

Das folgende Kapitel gibt eine Einführung in die Zeitreihenanalyse in der Statistik und stellt eine Modellklasse vor, die für Zeitreihenvorhersagen verwendet werden kann. Anschließend wird in Kapitel 3 erläutert, warum sich Python als Programmiersprache für Machine-Learning und Data Science eignet. Dabei werden einige für diese Bereiche und diese Arbeit relevante Bibliotheken vorgestellt. Daraufhin wird in Kapitel 4 ein Prozess erläutert, der die Vorgehensweise bei einem Data Science Projekt allgemein aber auch in dieser Arbeit, beschreibt. Im darauf folgenden Kapitel werden einige Grundlagen zur Prognosenerstellung mittels Machine-Learning vorgestellt, bevor dann in Kapitel 6 auf konkrete Modell-Architekturen eingegangen wird. In diesem Kapitel wird auch die Hauptkomponentenanalyse und deren Auswirkungen auf die verschiedenen Netz-Architekturen dargestellt. Anschließend wird die Implementierung und Optimierung der Modelle beschrieben. Daraufhin wird das Ergebnis des Trainierens analysiert und darauf basierend eine Modell-Architektur ausgewählt. Für die ausgewählte Architektur werden die Vorhersagemodelle erstellt, deren Integration in das Monitoring-Webdashboard in den Kapiteln 7 und 8 behandelt wird. Dafür werden in Kapitel 7 zunächst zwei grundlegende Ansätze vorgestellt und verglichen. Danach