

# Automatisierte Hybride Zeitreihenprognose: Design, Benchmarking und Anwendungsfälle<sup>1</sup>

André Bauer<sup>2</sup>

**Abstract:** Moderne Cloud-Umgebungen unterliegen Lastschwankungen und entsprechend schnellen und unerwarteten Änderungen. Um ausreichend Rechenressourcen rechtzeitig bereitzustellen, müssen sogenannte Auto-Skalierer den zukünftigen Ressourcenbedarf vorhersagen. Allerdings haben bestehende Arbeiten zur Zeitreihenprognose und zur automatischen Skalierung der Cloud zwei große Probleme. Erstens gibt es keinen vollautomatischen und generischen Prognoseansatz, der die vorhandenen Prognosemethoden so kombiniert, dass ihre Stärken genutzt und ihre Schwächen vermieden werden, um genaue Vorhersagen mit einer verlässlichen Laufzeit zu liefern. Zweitens wird bestehenden Auto-Skalierern misstraut, ein zuverlässiges und kosteneffizientes autonomes Ressourcenmanagement für moderne Cloud-Umgebungen zu bieten, da die Sorge besteht, dass ungenaue oder verzögerte Anpassungen zu finanziellen Verlusten führen können. Um diese Probleme zu lösen, stellt die Dissertation drei Beiträge vor: (i) Einen Prognose-Benchmark, der das Problem der begrenzten Vergleichbarkeit zwischen bestehenden Prognosemethoden adressiert; (ii) Eine automatisierte hybride Zeitreihen-Prognosemethode; (iii) Einen neuartigen hybriden Auto-Skalierer für koordinierte Skalierung von Anwendungen.

## 1 Einleitung

Heutzutage leben wir in einer digitalisierten Welt. Sowohl unser berufliches als auch unser privates Leben ist von verschiedenen IT-Diensten durchzogen, welche typischerweise in verteilten Computersystemen (z.B. Cloud-Umgebungen) betrieben werden. Die Betreiber solcher Systeme sind aufgrund des hohen Digitalisierungsgrades mit schnellen und wechselnden Anforderungen konfrontiert. Insbesondere Cloud-Umgebungen unterliegen starken Lastschwankungen und entsprechenden schnellen und unerwarteten Änderungen des Bedarfs an Rechenressourcen. Um dieser Herausforderung zu begegnen, können sogenannte Auto-Skalierer, wie z.B. der schwellenwertbasierte Mechanismus von Amazon Web Services EC2, eingesetzt werden, um eine elastische Skalierung der Rechenressourcen zu ermöglichen. Doch trotz dieser Gelegenheit werden geschäftskritische Anwendungen nach wie vor mit deutlich überdimensionierten Rechenkapazitäten betrieben, um einen stabilen und zuverlässigen Dienstbetrieb zu gewährleisten. Diese Strategie wird aufgrund des mangelnden Vertrauens in Auto-Skalierer und der Sorge verfolgt, dass ungenaue oder verzögerte Anpassungen zu finanziellen Verlusten führen könnten.

Um die Ressourcenkapazität rechtzeitig anpassen zu können, müssen die zukünftigen Ressourcenanforderungen “vorhergesehen” werden. Denn die Reaktion auf Veränderungen,

---

<sup>1</sup> Englischer Titel der Dissertation: “Automated Hybrid Time Series Forecasting: Design, Benchmarking, and Use Cases”

<sup>2</sup> Universität Würzburg, Fakultät für Mathematik und Informatik, andre.bauer@uni-wuerzburg.de

sobald diese beobachtet werden, führt zu einer inhärenten Verzögerung. Mit anderen Worten, es sind genaue Prognosemethoden erforderlich, um Systeme proaktiv anzupassen. Ein wirksamer Ansatz in diesem Zusammenhang ist die Zeitreihenprognose, welche auch in vielen anderen Bereichen angewandt wird. Die Kernidee besteht darin, vergangene Werte zu untersuchen und vorherzusagen, wie sich diese Werte im Laufe der Zeit entwickeln werden. Nach dem “No-Free-Lunch Theorem” [WM97] gibt es keinen Algorithmus, der für alle Szenarien am besten funktioniert. Daher ist die Auswahl einer geeigneten Prognosemethode für einen gegebenen Anwendungsfall eine wesentliche Herausforderung. Denn jede Methode hat - abhängig vom spezifischen Anwendungsfall - ihre Vor- und Nachteile. Deshalb basiert üblicherweise die Wahl der Prognosemethode auf Trial-and-Error oder auf Expertenwissen, welches nicht vollständig automatisiert werden kann. Beide Ansätze sind teuer und fehleranfällig.

Obwohl Auto-Skalierung und Zeitreihenprognose etablierte Forschungsgebiete sind, können die bestehenden Ansätze die genannten Herausforderungen nicht vollständig bewältigen: (i) Bei unserer Untersuchung zur Zeitreihenvorhersage stellten wir fest, dass die meisten der überprüften Artikel nur eine geringe Anzahl von (meist verwandten) Methoden berücksichtigen und ihre Performanz auf einem kleinen Datensatz von Zeitreihen mit nur wenigen Fehlermaßen bewerten, während sie keine Informationen über die Ausführungszeit der untersuchten Methoden liefern. Daher können solche Artikel nicht als Hilfe für die Wahl einer geeigneten Methode für einen bestimmten Anwendungsfall herangezogen werden; (ii) Bestehende hybride open-source Prognosemethoden [Ce17, THA18, BHB16, TL18, Sm20], die sich mindestens zwei Methoden zunutze machen, um das “No-Free-Lunch Theorem” anzugehen, sind rechenintensiv, schlecht automatisiert, für einen bestimmten Datensatz ausgelegt oder haben eine unvorhersehbare Laufzeit. Methoden, die eine hohe Varianz in der Ausführungszeit aufweisen, können nicht für zeitkritische Szenarien angewendet werden (z.B. Auto-Skalierung), während Methoden, die auf einen bestimmten Datensatz zugeschnitten sind, Einschränkungen für mögliche Anwendungsfälle mit sich bringen (z.B. nur jährliche Zeitreihen vorhersagen); (iii) Auto-Skalierer skalieren typischerweise eine Anwendung entweder proaktiv oder reaktiv. Obwohl es einige hybride Auto-Skalierer [AETE12, Ji13, Ur08, Wu16, Iq11] gibt, fehlt es ihnen an ausgeklügelten Lösungen zur Kombination von reaktiver und proaktiver Skalierung. Beispielsweise werden Ressourcen nur proaktiv freigesetzt, während die Ressourcenzuweisung vollständig reaktiv (inhärent verzögert) erfolgt; (iv) Die Mehrheit der vorhandenen Mechanismen berücksichtigt bei der Skalierung einer Anwendung in einer öffentlichen Cloud-Umgebung nicht das Preismodell des Anbieters, was häufig zu überhöhten Kosten führt. Auch wenn einige kosteneffiziente Auto-Skalierer vorgeschlagen wurden, berücksichtigen sie nur den aktuellen Ressourcenbedarf und vernachlässigen ihre Entwicklung im Laufe der Zeit. Beispielsweise werden Ressourcen oft vorzeitig abgeschaltet, obwohl sie vielleicht bald wieder benötigt werden.

Um den genannten Herausforderungen und den Defiziten der bisherigen Arbeiten zu begegnen, wurden in der Dissertation [Ba20] drei Ziele formuliert:

**Ziel I:** *Einen Prognose-Benchmark bereitstellen, um gleiche Voraussetzungen für die Bewertung und den Vergleich der Leistung von Prognosemethoden in*

*einem breiten Rahmen zu schaffen, der eine Vielzahl von Bewertungsszenarien abdeckt.*

**Ziel II:** *Bereitstellung einer vollautomatischen und generischen hybriden Prognosemethode, die automatisch relevante Informationen aus einer gegebenen Zeitreihe extrahiert und diese nutzt, um bestehende Methoden so zu kombinieren, dass eine hohe Prognosegenauigkeit bei gleichzeitig geringer Laufzeit-Varianz erreicht wird.*

**Ziel III:** *Entwicklung eines hybriden Auto-Skalierers, der die koordinierte Skalierung von Anwendungen ermöglicht, indem er proaktive Skalierung (basierend auf der entwickelten Prognosemethode) mit reaktiver Skalierung als Fallback-Mechanismus kombiniert, um maximale Zuverlässigkeit der Ressourcenanpassungen zu gewährleisten.*

Die auf diesen Zielen basierenden Beiträge der Dissertation sind in Abschnitt 2 zusammengefasst. Anschließend wird im Abschnitt 3 die Grundidee des Forschungsbeitrag II (eine automatisierte hybride Zeitreihen-Prognosemethode namens *Telescope*) präsentiert. In Abschnitt 4 werden dann die wichtigsten Ergebnisse der Auswertung von *Telescope* vorgestellt. Abschließend erfolgt ein Resümee der Dissertation.

## 2 Übersicht der Dissertationsbeiträge

Im Rahmen der Dissertation [Ba20] wurden folgende drei Beiträge vorgestellt: (i) Der erste Beitrag - ein *Prognosebenchmark*<sup>3</sup> - behandelt das Problem der begrenzten Vergleichbarkeit zwischen bestehenden Prognosemethoden; (ii) Der zweite Beitrag stellt eine automatisierte hybride Zeitreihen-Prognosemethode namens *Telescope*<sup>4</sup> vor, die sich der Herausforderung des “No-Free-Lunch Theorem” [WM97] stellt; (iii) Der dritte Beitrag stellt *Chamulleon*, einen neuartigen hybriden Auto-Skalierer für die koordinierte Skalierung von Anwendungen bereit, der *Telescope* zur Vorhersage der Lastintensität als Grundlage für eine proaktive Ressourcenbereitstellung nutzt. Im Folgenden werden die drei Beiträge der Arbeit zusammengefasst.

**Prognosebenchmark** Um gleiche Ausgangsbedingungen für die Bewertung von Prognosemethoden anhand eines breiten Spektrums zu schaffen, schlagen wir einen neuartigen Benchmark vor, der Prognosemethoden auf der Grundlage ihrer Performanz in einer Vielzahl von Szenarien automatisch bewertet und ein Ranking erstellt. Der Benchmark umfasst vier verschiedene Anwendungsfälle, die jeweils 100 heterogene Zeitreihen aus verschiedenen Bereichen abdecken. Der Datensatz wurde aus öffentlich zugänglichen Zeitreihen zusammengestellt und so konzipiert, dass er eine viel höhere Diversität aufweist als bestehende Prognosewettbewerbe. Neben dem neuen Datensatz führen wir zwei neue Maße ein, die verschiedene Aspekte einer Prognose beschreiben. Wir haben den entwickelten Benchmark zur Bewertung von *Telescope* angewandt.

<sup>3</sup> Prognosebenchmark: <https://github.com/DescartesResearch/ForecastBenchmark>

<sup>4</sup> Telescope: <https://github.com/DescartesResearch/telescope>

**Telescope** Um eine generische Prognosemethode bereitzustellen, stellen wir einen neuartigen, auf maschinellem Lernen basierenden Prognoseansatz vor, der automatisch relevante Informationen aus einer gegebenen Zeitreihe extrahiert. Genauer gesagt, Telescope extrahiert automatisch intrinsische Zeitreihenmerkmale und zerlegt die Zeitreihe dann in Komponenten, wobei für jede dieser Komponenten ein Prognosemodell erstellt wird. Jede Komponente wird mit einer anderen Methode prognostiziert und dann wird die endgültige Prognose aus den vorhergesagten Komponenten unter Verwendung eines regressionsbasierten Algorithmus des maschinellen Lernens zusammengestellt. In mehr als 1300 Experimentstunden, in denen 15 konkurrierende Methoden (einschließlich Ansätze von Facebook [TL18] und Uber [Sm20]) auf 400 Zeitreihen verglichen wurden, übertraf Telescope alle Methoden und zeigte die beste Prognosegenauigkeit in Verbindung mit einer niedrigen und zuverlässigen Ausführungszeit. Im Vergleich zu den konkurrierenden Methoden, die im Durchschnitt einen Prognosefehler (genauer gesagt, den symmetric mean absolute forecast error) von 29% aufwiesen, wies Telescope einen Fehler von 20% auf und war dabei 2556 mal schneller. Insbesondere die Methoden von Uber und Facebook wiesen einen Fehler von 48% bzw. 36% auf und waren 7334 bzw. 19-mal langsamer als Telescope.

**Chamulleon** Um eine zuverlässige Auto-Skalierung zu ermöglichen, stellen wir einen hybriden Auto-Skalierer vor, der proaktive und reaktive Techniken kombiniert, um verteilte Cloud-Anwendungen, die mehrere Dienste umfassen, koordiniert und kostengünstig zu skalieren. Genauer gesagt, werden proaktive Anpassungen auf der Grundlage von Prognosen von Telescope geplant, während reaktive Anpassungen auf der Grundlage tatsächlicher Beobachtungen der überwachten Lastintensität ausgelöst werden. Um auftretende Konflikte zwischen reaktiven und proaktiven Anpassungen zu lösen, wird ein komplexer Konfliktlösungsalgorithmus implementiert. Außerdem überprüft Chamulleon Anpassungen im Hinblick auf das Preismodell des Cloud-Anbieters, um die anfallenden Kosten in öffentlichen Cloud-Umgebungen zu minimieren. In mehr als 400 Experimentstunden, in denen fünf konkurrierende Auto-Skalierungsmechanismen unter fünf verschiedene Arbeitslasten, vier verschiedene Anwendungen und drei verschiedene Cloud-Umgebungen evaluiert wurden, zeigte Chamulleon die beste Auto-Skalierungsleistung und Zuverlässigkeit bei gleichzeitiger Reduzierung der berechneten Kosten. Die konkurrierenden Methoden lieferten während (durchschnittlich) 31% der Versuchszeit zu wenige Ressourcen. Im Gegensatz dazu reduzierte Chamulleon diese Zeit auf 8% und die SLO-Verletzungen (Service Level Objectives) von 18% auf 6%, während es bis zu 15% weniger Ressourcen verwendete und die berechneten Kosten um bis zu 45% senkte.

### 3 Grundidee von Telescope

Die Annahme der Datenstationarität<sup>5</sup> ist eine inhärente Einschränkung für die Zeitreihenprognose. Jede Zeitreiheneigenschaft, welche die Stationarität verletzt, wie z. B. ein nicht konstanter Mittelwert, eine nicht konstante Varianz oder ein multiplikativer Effekt, stellt eine Herausforderung für die richtige Modellbildung dar [MSA18]. Daher transformiert Telescope automatisch die gegebene Zeitreihe, leitet intrinsische Merkmale aus

<sup>5</sup> Bei einer stationären Zeitreihe ändern sich die statistischen Eigenschaften (wie z. B. Mittelwert, Varianz und Autokovarianz) im Laufe der Zeit nicht.

der Zeitreihe ab, wählt einen geeigneten Satz von Merkmalen aus und behandelt jedes Merkmal separat. Mit anderen Worten, wir integrieren verschiedene Methoden, um nicht-stationäre Zeitreihen zu behandeln.

---

**Algorithm 1:** Telescope Prognose
 

---

```

Input: Time series  $ts$ , horizon  $h$ 
Result: Forecast of  $ts$ 
1  $[ts, freqs] = \text{Preprocessing}(ts)$ ;
2 if  $freqs[1] > 1$  then //  $ts$  ist saisonal
3    $features = \text{FeatureExtraction}(ts, freqs)$ ;
4    $model = \text{ModelBuilding}(ts, features)$ ;
5    $forecast = \text{Forecasting}(model, h)$ ;
6 else //  $ts$  ist nicht saisonal und kann daher nicht zerlegt werden
7    $forecast = \text{ARIMA}(ts, h)$ ; // Fallback: ARIMA Prognose
8 end
9  $forecast = \text{Postprocessing}(forecast)$ ;
10 return  $forecast$ 

```

---

Der Arbeitsablauf von Telescope kann in fünf Phasen geteilt werden und ist in Algorithmus 1 vereinfacht dargestellt und erhält als Eingabe eine univariate Zeitreihe  $ts$  und den Horizont  $h$ . Der Horizont gibt an, wieviele Werte auf einmal vorhergesagt werden müssen. In der ersten Phase (Zeile 1) wird die Zeitreihe vorverarbeitet und die Frequenzen<sup>6</sup> der zugrunde liegenden Muster in der Zeitreihe werden extrahiert. Falls die Zeitreihe saisonal ist, umfassen die zweite und dritte Phase von Telescope die Extraktion relevanter intrinsischer Zeitreihenmerkmale (Zeile 3) und die Erstellung eines Modells, das die Zeitreihe auf der Grundlage dieser Merkmale beschreibt (Zeile 4). Anschließend wird das Modell verwendet, um das Verhalten der zukünftigen Zeitreihe vorherzusagen (Zeile 5). In dem Fall, dass die Zeitreihe nicht saisonal ist (Zeile 7), wird die Zeitreihe mit ARIMA (Auto-Regressive Integrated Moving Average) [BJ70] modelliert und vorhergesagt. Schließlich wird die Prognose entsprechend der Preprocessing-Phase nachbearbeitet und zurückgegeben. Im Folgenden wird jede Phase im Detail erläutert.

**Preprocessing** Da Vorhersagemethoden, insbesondere Methoden des maschinellen Lernens, mit sich ändernder Varianz und multiplikativen Effekten innerhalb einer Zeitreihe Schwierigkeiten haben [SK12], wird die Zeitreihe transformiert. Genauer gesagt wendet Telescope die Box-Cox-Transformation [BC64] an, da sie sowohl die Varianz als auch die multiplikativen Effekte der Zeitreihe reduziert, was zu einem verbesserten Vorhersagemodell [MSA18] führt. Parallel zur Transformation extrahiert Telescope die dominantesten Frequenzen<sup>7</sup> aus der Zeitreihe durch Anwendung eines Periodogramms [Sc99].

**Feature Extraction** In dieser Phase ermittelt Telescope intrinsische Zeitreihenmerkmale zur Bewältigung typischer Probleme oder Schwierigkeiten, die bei der Modellierung einer Zeitreihe auftreten können: (i) Die Zeitreihe hat mehrere sich überlagernde wiederkehrende Muster und (ii) die Zeitreihe verletzt die stationäre Eigenschaft. Um das erste Problem anzugehen, bestimmt Telescope für jede dominante Frequenz die zugehörigen Fourier-

<sup>6</sup> In der Zeitreihenanalyse bezeichnet die Frequenz die Länge eines wiederkehrenden Musters.

<sup>7</sup> Mit dominant meinen wir die häufigste Periode wie z.B. Tage in einem Jahr.

Terme der Zeitreihe zur Modellierung der verschiedenen Muster. Genauer gesagt werden für jede dominante Frequenz sowohl ein Sinus als auch ein Kosinus mit der Periodenlänge der entsprechenden Frequenz extrahiert. Um die Nicht-Stationarität zu behandeln, besteht die Kernidee von Telescope darin, die Zeitreihe zu zerlegen und dann jeden Teil separat zu behandeln. Zu diesem Zweck wird die Zeitreihe mit STL (Seasonal and Trend decomposition using Loess) [CI90] in ihre Komponenten Trend (langfristige Entwicklung der Zeitreihe), Saisonalität (wiederkehrendes Muster innerhalb einer regelmäßigen Periode) und Unregelmäßigkeit (verbleibende Teil der Zeitreihe, der nicht durch Trend oder Saisonalität beschrieben wird) zerlegt.

**Model Building** Um ein geeignetes Vorhersagemodell zu erstellen, das die Zeitreihe beschreibt, implementiert Telescope XGBoost (eXtreme Gradient Boosting) [CG16], um die Beziehung zwischen der Zeitreihe und den intrinsischen Merkmalen zu finden. Da ein starker Trend sowohl die Varianz erhöht als auch die Stationarität verletzt, entfernt Telescope den Trend aus der Zeitreihe. Die resultierende enttrentete Zeitreihe ist nun trend-stationär. Obwohl auch die Saisonalität die Stationarität verletzen kann, eignen sich Methoden des maschinellen Lernens zur Mustererkennung. Folglich lernt XGBoost während seines Trainingsverfahrens, wie die enttrentete Zeitreihe durch die intrinsischen Merkmale Fourier-Terme und Saisonalität beschrieben werden kann.

**Forecasting** Um die Zeitreihe vorhersagen zu können, müssen die verschiedenen Komponenten getrennt voneinander vorhergesagt werden. Die Saisonalität und die Fourier-Terme sind per Definition wiederkehrende Muster, so dass sie einfach fortgesetzt werden können. Die resultierende vorhergesagte Saisonalität und die vorhergesagten Fourier-Terme werden in Verbindung mit dem Vorhersagemodell als Merkmale für die Vorhersage der zukünftigen enttrenteten Zeitreihen verwendet. Genauer gesagt, regressiert das maschinelle Lernverfahren für jeden Zeitpunkt der Vorhersage einen neuen Wert auf Basis der entsprechenden Werte der Merkmale. Parallel zur Vorhersage der wiederkehrenden Muster wird auch der Trend vorhergesagt. Da der Trend keine wiederkehrenden Muster enthält, ist eine fortschrittliche Prognosemethode erforderlich. Zu diesem Zweck verwenden wir ARIMA [BJ70], da es in der Lage ist, den Trend auch aus wenigen Punkten zu schätzen. Nachdem der Trend vorhergesagt wurde, werden im letzten Schritt dieser Phase die vorhergesagte enttrentete Zeitreihen und der vorhergesagte Trend aufsummiert.

**Postprocessing** Da die Zeitreihe in der Preprocessing-Phase transformiert wurde, wird die vorhergesagte Zeitreihe mit der inversen Box-Cox-Transformation zurücktransformiert.

## 4 Evaluation von Telescope

Um die Qualität von Telescope zu evaluieren, haben wir unsere hybride Prognosemethode auf 400 verschiedenen Zeitreihen mit 15 unterschiedlichen Prognosemethoden verglichen. Aus Platzgründen vergleichen wir Telescope in diesem Abschnitt mit vier aktuellen hybriden Prognosemethoden (BETS [BHB16], ES-RNN [Sm20], FFORMS [THA18] und Prophet [TL18]) und die nach unseren Experimenten [Ba20] besten Methoden aus dem Bereich der "klassischen" Zeitreihenprognose (sARIMA [BJ70]) als auch des maschinellen

Lernens (XGBoost [CG16]). Um die Vorhersagegenauigkeit zu messen, berücksichtigen wir den sMAPE (symmetrical mean absolute percentage error) [Ma93], wobei  $\bar{e}_{sM}$  den durchschnittlichen Fehler und  $\sigma_{e_{sM}}$  die Standardabweichung des Fehlers widerspiegelt. Neben der Genauigkeit messen wir auch die Laufzeit der einzelnen Methoden. Die Laufzeit wird durch die Zeit normalisiert, die eine naive Prognosemethode benötigt (im Schnitt 0,1 Sekunde pro Vorhersage), wobei  $\bar{t}_{sN}$  die durchschnittliche normierte Zeit und  $\sigma_{t_{sN}}$  die Standardabweichung der normierten Zeit reflektiert. Als Beispiel für die Vorhersagen zeigt Abbildung 1 die konkurrierenden Prognosemethoden auf der Zeitreihe Air Passengers, welche oft als Referenz verwendet wird.

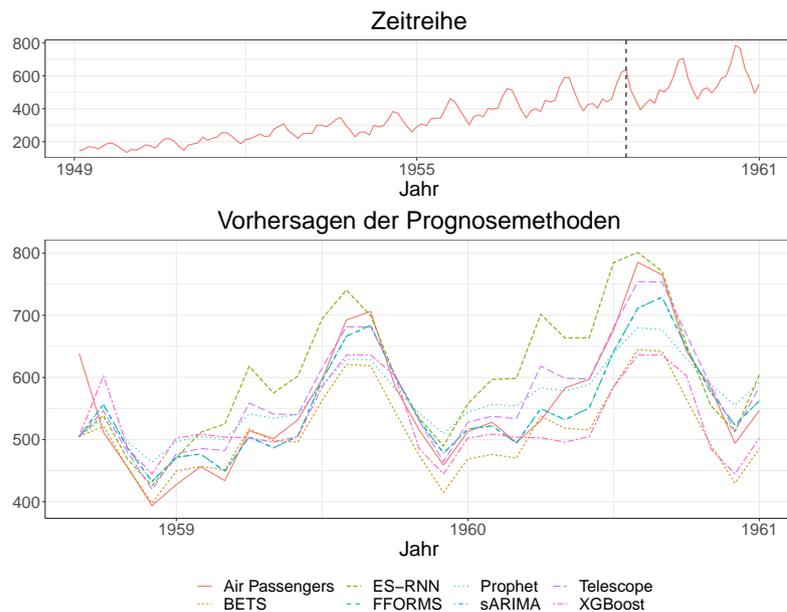


Abb. 1: Vorhersagen für alle konkurrierenden Methoden auf der Zeitreihe Air Passengers.

Tab. 1: Vorhersagefehler- und Laufzeitvergleich über alle Zeitreihen.

Maß	BETS	ES-RNN	FFORMS	Prophet	sARIMA	Telescope	XGBoost
$\bar{e}_{sM}$ [%]	25,52	47,87	21,15	35,56	20,63	<b>19,95</b>	23,85
$\sigma_{e_{sM}}$ [%]	34,25	66,99	36,87	$2,30 \cdot 10^2$	35,63	<b>31,35</b>	34,67
$\bar{t}_{sN}$	$6,14 \cdot 10^4$	$1,61 \cdot 10^6$	$5,23 \cdot 10^5$	$2,04 \cdot 10^3$	$8,48 \cdot 10^5$	$1,43 \cdot 10^2$	<b>6,73</b>
$\sigma_{t_{sN}}$	$1,07 \cdot 10^6$	$4,92 \cdot 10^6$	$7,93 \cdot 10^6$	$6,15 \cdot 10^3$	$9,08 \cdot 10^6$	98,67	<b>15,59</b>

Tabelle 1 zeigt die Ergebnisse, die über alle Zeitreihen gemittelt wurden, für alle konkurrierenden Methoden. Jede Zeile steht für ein Maß, jede Spalte für eine Methode, und die besten Werte (je niedriger, desto besser) sind fett hervorgehoben. Die genaueste Prognosemethode ist Telescope (19,95%), gefolgt von sARIMA (20,63%). Den höchsten Fehler weist ES-RNN mit 47,87% auf. Die mit Abstand schnellste Methode ist XGBoost (6,73). Telescope ( $1,43 \cdot 10^2$ ) hat die zweitniedrigste Laufzeit. Die langsamste Methode ist ES-RNN ( $1,61 \cdot 10^6$ ). Obwohl sARIMA die zweitbeste Vorhersagegenauigkeit hat, ist es im

Durchschnitt fast 6000 Mal langsamer als Telescope. Beispielsweise betrug die maximale Laufzeit von sARIMA für eine Zeitreihe 465.574 Sekunden, was fast 5,5 Tagen entspricht.

Obwohl unser Datensatz mit 400 verschiedenen Zeitreihen ein breites Spektrum an Anwendungsfällen umfasst, lassen sich die Evaluierungsergebnisse möglicherweise nicht auf alle Zeitreihen aus allen Bereichen verallgemeinern. Neben dem Datensatz versuchten wir auch, repräsentative Vorhersagemethoden zu untersuchen, die auf verschiedenen Techniken basieren, wie beispielsweise die Methoden von Facebook (Prophet) und Uber (ES-RNN). Wir verwenden jedoch alle Methoden mit ihren Standardeinstellungen. Folglich können die beobachteten Ergebnisse abweichen, wenn die Prognosemethoden auf die einzelnen Zeitreihen angepasst werden.

## 5 Resümees

Heutzutage leben wir in einer schnelllebigen Welt, und so sind viele Bereiche Trends und unterschiedlichen Anforderungen unterworfen. So müssen Cloud-Umgebungen beispielsweise mit Lastschwankungen und entsprechend schnellen und unerwarteten Änderungen des Bedarfs an Rechenressourcen zurechtkommen. Da die Reaktion auf Änderungen, sobald sie beobachtet werden, eine inhärente Verzögerung mit sich bringt, muss der zukünftige Ressourcenbedarf vorhergesagt werden, um notwendige Schritte im Voraus zu identifizieren. Eine nützliche und etablierte Technik in diesem Zusammenhang ist die Zeitreihenprognose, die auch in vielen anderen Bereichen eingesetzt wird. Obwohl die Zeitreihenprognose die proaktive automatische Skalierung der benötigten Ressourcen in Cloud-Umgebungen ermöglicht, werden geschäftskritische Anwendungen immer noch mit stark überprovisionierten Ressourcen betrieben, um einen stabilen und zuverlässigen Servicebetrieb zu gewährleisten. Diese Strategie wird vor allem aufgrund von zwei Hauptproblemen der bestehenden Arbeiten verfolgt: Erstens gibt es keinen vollautomatischen und generischen Prognoseansatz, der die vorhandenen Prognosemethoden so kombiniert, dass ihre Stärken genutzt und ihre Schwächen vermieden werden, um genaue Prognosen mit einer zuverlässigen Laufzeit zu liefern. Zweitens wird bestehenden Auto-Skalierer misstraut, ein zuverlässiges und kosteneffizientes autonomes Ressourcenmanagement für moderne Cloud-Umgebungen zu ermöglichen, da die Sorge besteht, dass ungenaue oder verzögerte Anpassungen zu finanziellen Verlusten führen können.

Um den genannten Herausforderungen zu begegnen, wurden in der Dissertation [Ba20] drei Beiträge vorgestellt, welche als wichtige Meilensteine auf dem Gebiet der Zeitreihenprognose und der automatischen Skalierung in Cloud-Umgebungen angesehen werden können. (i) In dieser Arbeit wird zum ersten Mal ein Prognosebenchmark präsentiert, der eine Vielzahl verschiedener Bereiche mit einer hohen Diversität zwischen den analysierten Zeitreihen abdeckt. Auf der Grundlage des zur Verfügung gestellten Datensatzes und des automatischen Auswertungsverfahrens trägt der vorgeschlagene Benchmark dazu bei, die Vergleichbarkeit von Prognosemethoden zu verbessern. Die Benchmarking-Ergebnisse von verschiedenen Prognosemethoden ermöglichen die Auswahl der am besten geeigneten Prognosemethode für einen gegebenen Anwendungsfall. (ii) Telescope bietet den ersten generischen und vollautomatischen Zeitreihen-Prognoseansatz, der sowohl genaue als

auch zuverlässige Prognosen liefert, ohne Annahmen über die analysierte Zeitreihe zu treffen. Dementsprechend macht es teure, zeitaufwändige und fehleranfällige Verfahren überflüssig, wie z.B. Trial-and-Error oder das Hinzuziehen eines Experten. Dies eröffnet neue Möglichkeiten, insbesondere in zeitkritischen Szenarien, in denen Telescope genaue Vorhersagen mit einer kurzen und zuverlässigen Antwortzeit liefern kann. Obwohl Telescope für diese Arbeit im Bereich des Cloud Computing eingesetzt wurde, gibt es, wie die Auswertung zeigt, keinerlei Einschränkungen hinsichtlich der Anwendbarkeit von Telescope in anderen Bereichen. Darüber hinaus wird Telescope, das auf GitHub zur Verfügung gestellt wurde, bereits in einer Reihe von interdisziplinären datenwissenschaftlichen Projekten eingesetzt, z.B. bei der vorausschauenden Wartung im Rahmen von Industry 4.0, bei der Vorhersage von Herzinsuffizienz in der Medizin oder als Bestandteil von Vorhersagemodellen für die Entwicklung von Bienenstöcken. (iii) Im Kontext der elastischen Ressourcenverwaltung ist Chamulteon ein wichtiger Meilenstein für die Stärkung des Vertrauens in Auto-Skalierern. Der komplexe Konfliktlösungsalgorithmus ermöglicht ein zuverlässiges und genaues Skalierungsverhalten, das Verluste durch übermäßige Ressourcenzuweisung oder SLO-Verletzungen (Service Level Objectives) reduziert. Mit anderen Worten, Chamulteon bietet zuverlässige Ressourcenanpassungen, die die berechneten Kosten minimieren und gleichzeitig die Benutzerzufriedenheit maximieren.

## Literaturverzeichnis

- [AETE12] Ali-Eldin, Ahmed; Tordsson, Johan; Elmroth, Erik: An Adaptive Hybrid Elasticity Controller for Cloud Infrastructures. In: IEEE NOMS 2012. IEEE, S. 204–212, 2012.
- [Ba20] Bauer, André: Automated Hybrid Time Series Forecasting: Design, Benchmarking, and Use Cases. Dissertation, University of Würzburg, Germany, 2020.
- [BC64] Box, George EP; Cox, David R: An Analysis of Transformations. *Journal of the Royal Statistical Society. Series B (Methodological)*, S. 211–252, 1964.
- [BHB16] Bergmeir, Christoph; Hyndman, Rob J; Benítez, José M: Bagging Exponential Smoothing Methods Using STL Decomposition and Box–Cox Transformation. *International journal of forecasting*, 32(2):303–312, 2016.
- [BJ70] Box, G.E.P.; Jenkins, G.M.: *Time Series Analysis: Forecasting and Control*. Holden-Day, 1970.
- [Ce17] Cerqueira, Vítor; Torgo, Luís; Pinto, Fábio; Soares, Carlos: Arbitrated Ensemble for Time Series Forecasting. In: *Joint European conference on machine learning and knowledge discovery in databases*. Springer, S. 478–494, 2017.
- [CG16] Chen, Tianqi; Guestrin, Carlos: Xgboost: A Scalable Tree Boosting System. In: *ACM SIGKDD 2016*. ACM, S. 785–794, 2016.
- [CI90] Cleveland, Robert B; Cleveland, William S; McRae, Jean E; Terpenning, Irma: STL: A Seasonal-Trend Decomposition Procedure based on Loess. *Journal of Official Statistics*, 6(1):3–73, 1990.
- [Iq11] Iqbal, Waheed; Dailey, Matthew N; Carrera, David; Janecek, Paul: Adaptive Resource Provisioning for Read Intensive Multi-Tier Applications in the Cloud. *Future Generation Computer Systems*, 27(6):871–879, 2011.

- [Ji13] Jiang, Jing; Lu, Jie; Zhang, Guangquan; Long, Guodong: Optimal Cloud Resource Auto-Scaling for Web Applications. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing. IEEE, S. 58–65, 2013.
- [Ma93] Makridakis, Spyros: Accuracy measures: theoretical and practical concerns. International journal of forecasting, 9(4):527–529, 1993.
- [MSA18] Makridakis, Spyros; Spiliotis, Evangelos; Assimakopoulos, Vassilios: Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. PloS one, 13(3):e0194889, 2018.
- [Sc99] Schuster, Arthur: The Periodgram of Magnetic Declination as Obtained from the Records of the Greenwich Observatory during the Years 1871-1895. Transactions of the Cambridge Philosophical Society, 18:107–135, 1899.
- [SK12] Sugiyama, Masashi; Kawanabe, Motoaki: Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation. The MIT Press, 2012.
- [Sm20] Smyl, Slawek: A Hybrid Method of Exponential Smoothing and Recurrent Neural Networks for Time Series Forecasting. International Journal of Forecasting, 36(1):75–85, 2020.
- [THA18] Talagala, Thiyanga S; Hyndman, Rob J; Athanasopoulos, George: Meta-Learning How to Forecast Time Series. Bericht, Monash University, Department of Econometrics and Business Statistics, 2018.
- [TL18] Taylor, Sean J; Letham, Benjamin: Forecasting at Scale. The American Statistician, 72(1):37–45, 2018.
- [Ur08] Urgaonkar, Bhuvan; Shenoy, Prashant; Chandra, Abhishek; Goyal, Pawan; Wood, Timothy: Agile Dynamic Provisioning of Multi-tier Internet Applications. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 3(1):1–39, 2008.
- [WM97] Wolpert, D. H.; Macready, W. G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation, 1(1):67–82, Apr 1997.
- [Wu16] Wu, S.; Li, B.; Wang, X.; Jin, H.: HybridScaler: Handling Bursting Workload for Multi-tier Web Applications in Cloud. In: 15th International Symposium on Parallel and Distributed Computing (ISPDC), 2016. S. 141–148, 2016.



**André Bauer** wurde 1989 in Deutschland geboren. Nach dem Erwerb der mittleren Reife, trat er 2006 eine Ausbildung zum Bauzeichner an. Nach dem Abschluss der Ausbildung entschied er sich sein Abitur nachzuholen und erlangte 2011 seine fachgebundene Hochschulreife. Er studierte Informatik an der Universität Würzburg. Nach seinem Masterabschluss fing er seine Promotion bei Prof. Dr. Samuel Kounev an und schloss diese 2020 mit Auszeichnung ab. Während seiner Promotion beschäftigte er sich mit Elastizität im Cloud Computing, Autoskalierung und Ressourcenmanagement, sowie der Datenanalyse und Modellbildung mittels Kombinationen von maschinellem Lernen und Verfahren aus der Zeitreihenanalyse. Seit der Disputation leitet er die

Forschergruppe für “Data Science Engineering Group” am Lehrstuhl für Software Engineering der Universität Würzburg. Darüber hinaus ist er gewählter Release Manager der RG-Cloud Arbeitsgruppe und Newsletter Editor der SPEC Research Group.