Proctor: A Semi-Supervised Performance Anomaly Diagnosis Framework for Production HPC Systems

Seminar Supercomputer

Jan Mägdefrau

Arbeitsbereich Wissenschaftliches Rechnen
Fachbereich Informatik
Fakultät für Mathematik, Informatik und Naturwissenschaften
Universität Hamburg

2022-01-14

Gliederung (Agenda)

Einleitung / Motivation

- Rechenzeit auf HPC-Systemen sehr kostbar
- Komplexe HPC-Systeme können signifikanten Laufzeitschwankungen unterliegen
- Grund dafür: Anomalien
- Lösung: Monitoring und Analyse von Metriken
- Problem: Full supervised Ansätze benötigen viele gelabelte Daten
- Proctor: semi-supervised Ansatz zur Erkennung und Klassifizierung von Performanceproblemen

Was sind Monitoring Frameworks?

- Software, welche auf Rechenknoten läuft
- Durchgehende Erfassung von Metriken
- Bestenfalls nur mit geringem CPU-Overhead
- Beispiele: Lightweight Distributed Metric Service (LDMS), Ganglia, ExaMon

[AAB+14]

Welche Daten können erfasst werden?

- Hunderte Metriken sind möglich
- Beispiele:
  - Memory (z.B. aktuell freier, aktiver, inaktiver Speicher)
  - CPU (z.B. pro-Kern Idle-Zeit, I/O Wartezeit)
  - Netzwerk (z.B. Anzahl empfangener Pakete, durchschnittliche Paketgröße)
  - Shared File System (z.B. Anzahl Lese-, Schreib- & Öffnenoperationen)

Anomalien

- Unnormale Performance-Schwankungen
- Mögliche Gründe:
  - Hardware-Abweichungen
  - Software-Probleme
  - Ressourcen-Stau zwischen oder innerhalb von Jobs
- Beispiele:
  - Memory leaks
  - Network contention
  - Orphan-Prozesse

[AZA+19]

Was ist supervised, semi-supervised und unsupervised?

- Kategorien für Machine-Learning Algorithmen
- supervised:
  - benötigt Datensatz mit Daten und Label
- unsupervised:
  - benötigt nicht zwangsläufig Datensatz mit Label
- semi-supervised
  - Mischform
  - benutzt sowohl Datensatz mit und ohne Label

[Lou20]

Autoencoder

- unsupervised künstliches Neuronales Netz [Bad19]
- Lernt Daten zu encodieren und codierte Daten zu decodieren
- Nutzung: Reduzierung von Daten-Dimension, Anomalie-Erkennung
- Anomalie-Erkennung durch großen Reconstruction Loss
- Deep-Autoencoder: besteht aus mehren Hidden Layern

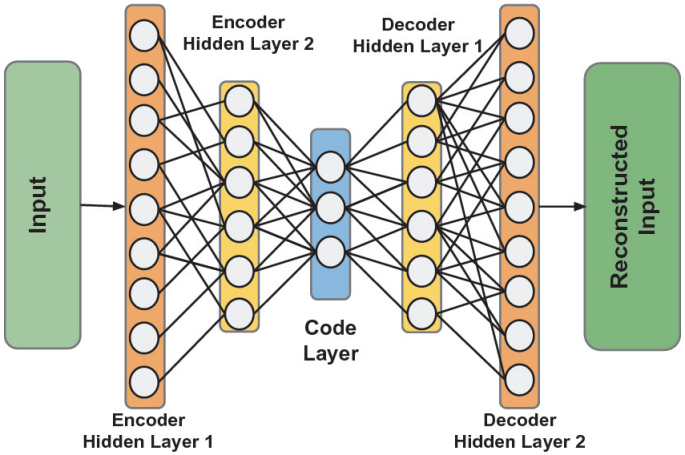

Abbildung: Aufbau Autoencoder

Was ist Proctor?

- Semi-supervised Machine-Learning Framework
- Ziel: Erkennen ob Rechenknoten in anomalen Zustand ist & Klassifizierung der Anomalie
- Training mit begrenzter Anzahl an gelabeten Daten
- Erstes semi-supervised Framework, welches entdeckt und analysiert
- 2021 von Forschern der Boston University und Sandia National Laboratories entwickelt

[AZA+21]

Übersicht

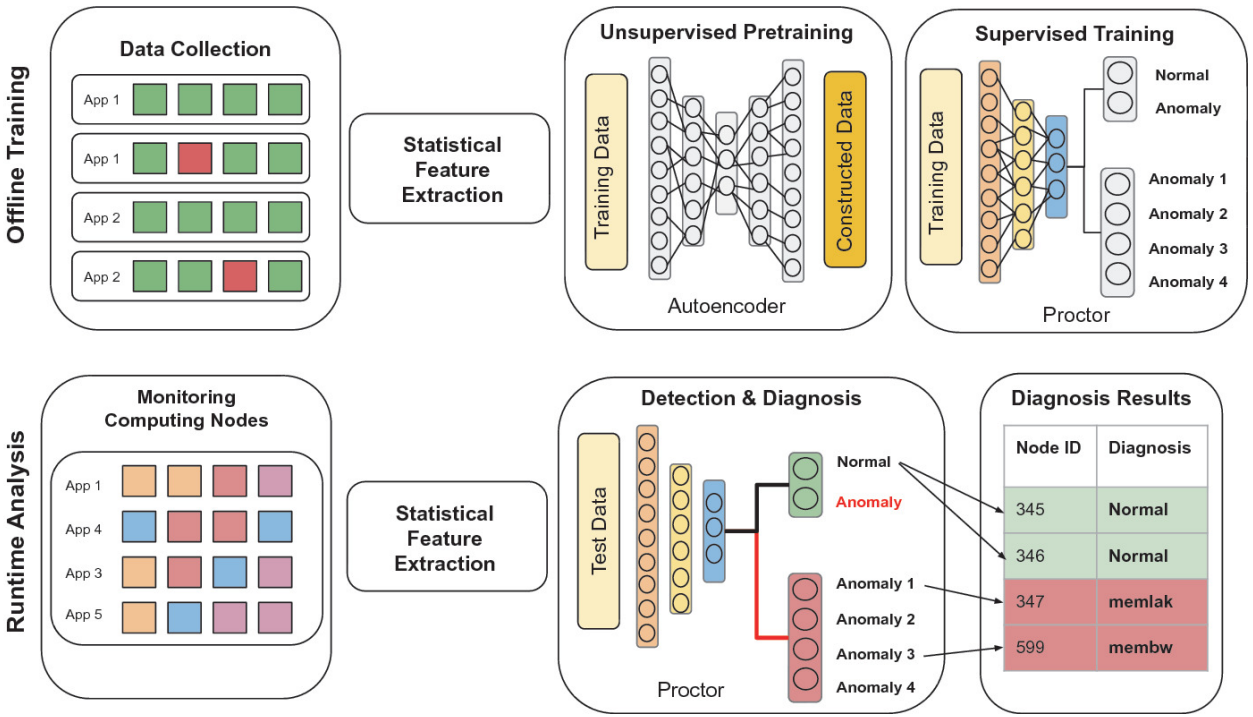

Abbildung: Architektur von Proctor

Feature Extraction

- Ziel: tausende Datenpunkte in nützliches Format bringen
- Ein Datensample = alle während Programmlaufs erfassten Daten
- Time series ⇒ statistische Eigenschaften
- Beispiele:
  - Ordnungsstatistik: Minimum, Maximum, Perzentile, Standardabweichung
  - Skewness / Schiefe
  - Kurtosis / Wölbung
  - ...

Unsupervised Pretraining

- Training des Autodecoders mit ungelabelten Daten
- Ziel: Gewichte lernen, damit reconstructed input und original input möglichst ähnlich sind
- Verwendung von Deep Autoencoder mit 2000 Neuronen im Code-Layer

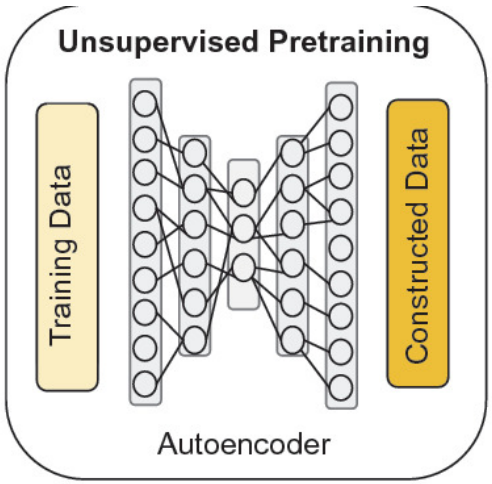

Abbildung: Aufbau unsupervised Pretraining

Supervised Training

- Support Vector Machine trainiert
- Encoded Features als Input
- Ziel: Klassifizierung

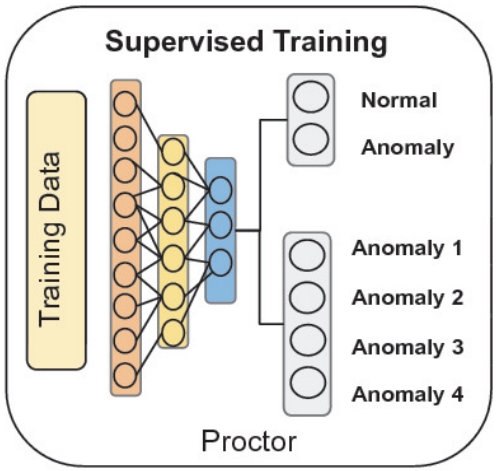

Abbildung: Aufbau supervised Training

Runtime Analysis

- Zwei-Level-Klassifikation:
  1. Entscheidung ob normal oder anomal
  2. Wenn anomal: Klassifikation

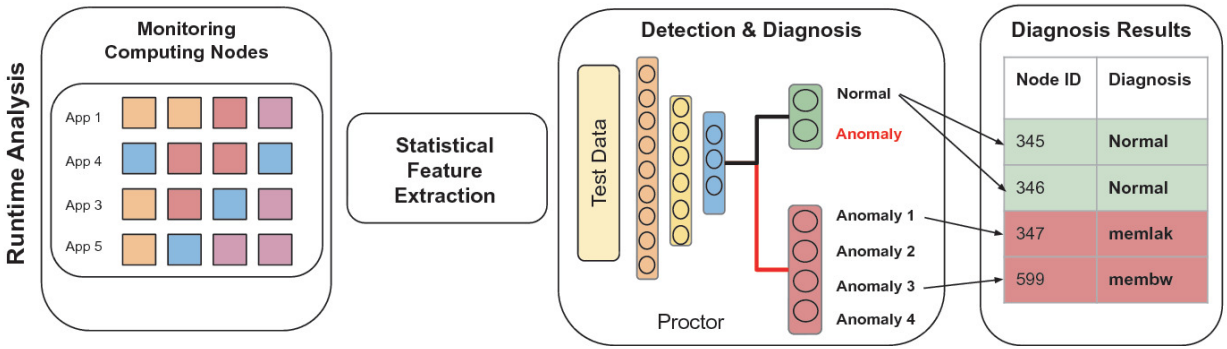

Abbildung: Aufbau Runtime Analysis

Testdatensätze

- Zwei Testsysteme, auf denen Daten erhoben werden
  - Volta
    - Cray XC30m supercomputer bei Sandia National Laboratories
    - 52 Rechenknoten, 64GB RAM, Intel Xeon E5-2695 v2 CPU
    - Programme über 4 oder 32 Knoten 10-15 Minuten Laufzeit
  - Eclipse
    - Production supercomputer bei Sandia National Laboratories
    - 1488 Rechenknoten, 128GB RAM, 2 Sockets mit je 18 E5-2695 v4 CPU cores
    - Programme über 4 Knoten 20-45 Minuten Laufzeit
- Unterschiedliche Programme ausgeführt
- LDMS als Monitoring Framework, jede Sekunde (Eclpise 806 Metriken, Volta 721 Metriken)
- Synthetische Erzeugung von Anomalien

Referenzframeworks

- RF-Tuncer
  - Statistische Feature Extraction & Selection
  - RF Classifier
  - supervised offline Traniningsphase und Runtime-Analyse
- AE-Borghesi
  - Autoencoder nur mit normalen Samples trainiert
  - Anomalie-Erkennung durch Threshold
  - Kann nur Erkennen, nicht klassifizieren

Vergleich Anomalie-Erkennung

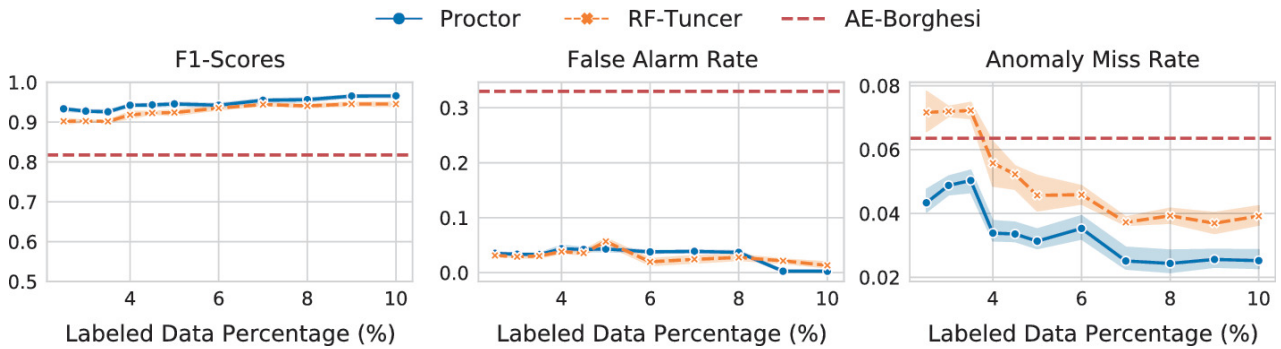

Abbildung: Vergleich Anomalie-Erkennung Proctor vs. RF-Tuncer vs. AE-Borghesi (Eclipse Datensatz)

Vergleich Anomalie-Klassifizierung

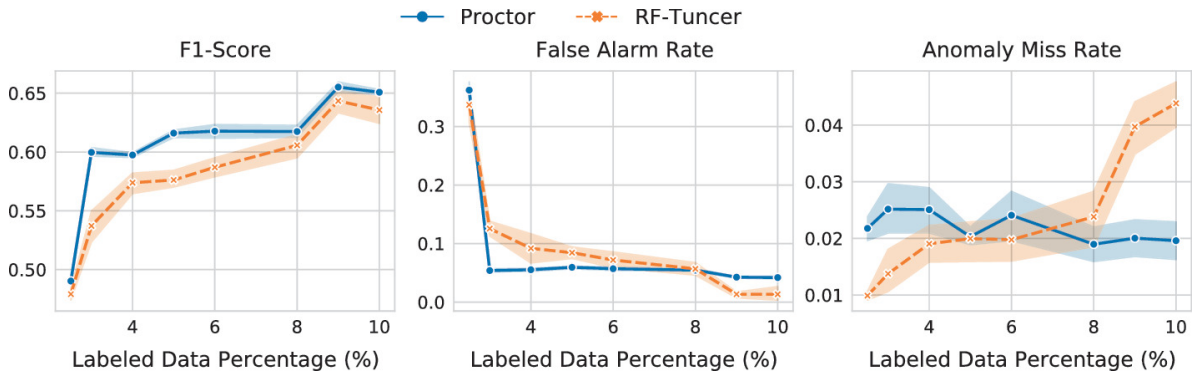

Abbildung: Klassifizierung Proctor vs. RF-Tuncer (Eclipse Datensatz)

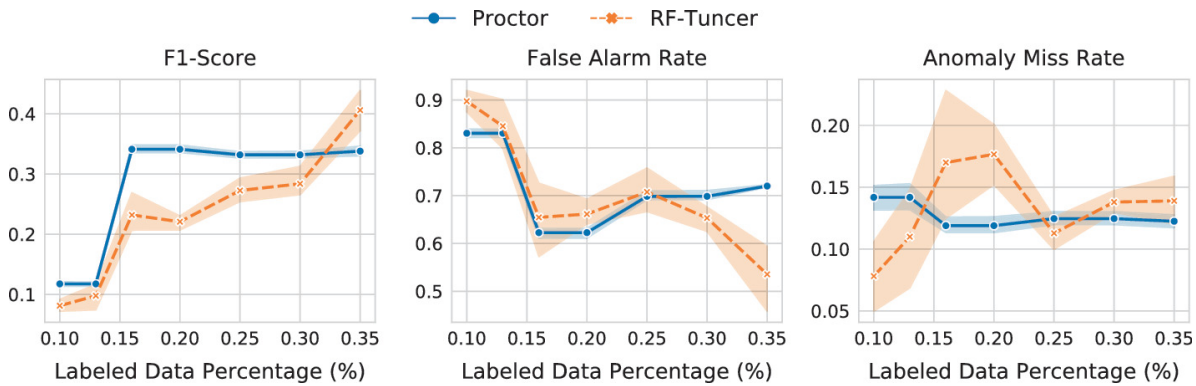

Abbildung: Klassifizierung Proctor vs. RF-Tuncer (Volta Datensatz)

Vergleich Anomalie-Klassifizierung unbekannter Anomalien

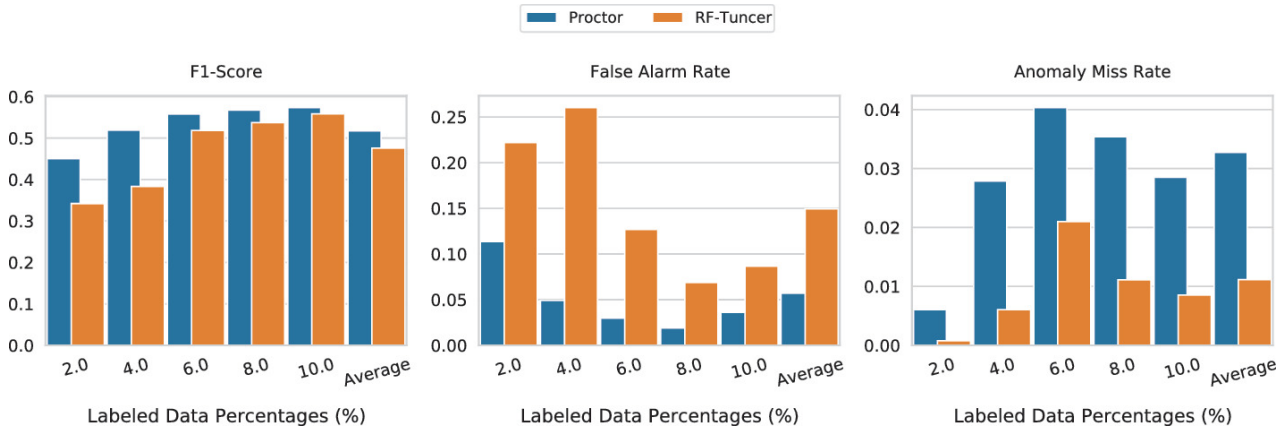

Abbildung: Vergleich Anomalie-Klassifizierung unbekannter Anomalien Proctor vs. RF-Tuncer (Eclipse Datensatz)

Zusammenfassung

- Proctor erstes Framework, welches Anomalien erkennt und klassifiziert
- semi-supervised Ansatz, funktioniert mit wenig gelabelten Daten
- Besteht aus Zwei-Level-Klassifikation mit Autoencoder und Support Vector Machine
- Im Vergleich mit RF-Tuncer und AE-Borghesi erzielt es bessere Ergebnisse

Literatur I

[AAB+14] Anthony Agelastos, Benjamin Allan, Jim Brandt, Paul Cassella, Jeremy Enos, Joshi Fullop, Ann Gentile, Steve Monk, Nichamon Naksinehaboon, Jeff Ogden, Mahesh Rajan, Michael Showerman, Joel Stevenson, Narate Taerat, and Tom Tucker. The lightweight distributed metric service: A scalable infrastructure for continuous monitoring of large scale computing systems and applications. In *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 154–165, 2014.

Literatur II

[AZA+19] Emre Ates, Yijia Zhang, Burak Aksar, Jim Brandt, Vitus J Leung, Manuel Egele, and Ayse K Coskun. Hpas: An hpc performance anomaly suite for reproducing performance variations. In *Proceedings of the 48th International Conference on Parallel Processing*, pages 1–10, 2019.

[AZA+21] Burak Aksar, Yijia Zhang, Emre Ates, Benjamin Schwaller, Omar Aaziz, Vitus J Leung, Jim Brandt, Manuel Egele, and Ayse K Coskun. Proctor: A semi-supervised performance anomaly diagnosis framework for production hpc systems. In *International Conference on High Performance Computing*, pages 195–214. Springer, 2021.

Literatur III

[Bad19] Will Badr. Auto-encoder: What is it? and what is it used for? (part 1), Jul 2019.

[Lou20] Serafeim Loukas. What is machine learning: A short note on supervised, unsupervised, semi-supervised and..., Jun 2020.