

Kapitel 5

Experimentelle Erprobung und Evaluierung

Im Folgenden werden die ausgewählten Algorithmen aus Abs. 4.2 anhand realer und synthetischer Daten getestet, evaluiert und gegenübergestellt. Ziel ist es, die Leistungsfähigkeit der verschiedenen Verfahren hinsichtlich ihrer Erkennungsrate, Robustheit und Effizienz zu untersuchen. Dabei wird zwischen den drei Anomalietypen – Punkt-, Subsequenz- und Korrelationsanomalien – unterschieden, um herauszufinden, welche Algorithmen sich besonders für die jeweilige Art von Abweichung eignen.

Zunächst wird die experimentelle Umgebung beschrieben, einschließlich der verwendeten Datensätze und Metriken zur Bewertung der Algorithmen. Anschließend erfolgt die Durchführung der Experimente, wobei sowohl synthetische als auch reale Sensordaten auf Basis des SSP X1 Systems verwendet werden. Abschließend wird eine vergleichende Analyse vorgenommen, um Stärken und Schwächen der Algorithmen herauszuarbeiten und deren Eignung für den praktischen Einsatz in Predictive-Maintenance-Szenarien zu bewerten.

5.1 Detektion von Punktanomaliedetektion

Zur Erprobung der Punktanomaliedetektionsalgorithmen werden drei, im Original aus den Datenaufzeichnung der SSP X1 stammenden, Ausschnitte der Datensätze um einige synthetische Punktanomalien erweitert. Dabei handelt es sich um die RAM und CPU Auslastung sowie die CPU Temperatur. Die Zeitspanne, über die Daten entstanden sind, entspricht einer Dauer von genau 48 Stunden und umfasst ca. 17000 Datenpunkten, wie in Abb. 5.1 dargestellt ist.

Um eine aussagekräftige Vergleichbarkeit herzustellen, wird, wie in Abs. 3.2.4 erwähnt, der F1-Score als Metrik angewandt, um die Genauigkeit der implementierten Algorithmen zu testen. Ebenfalls

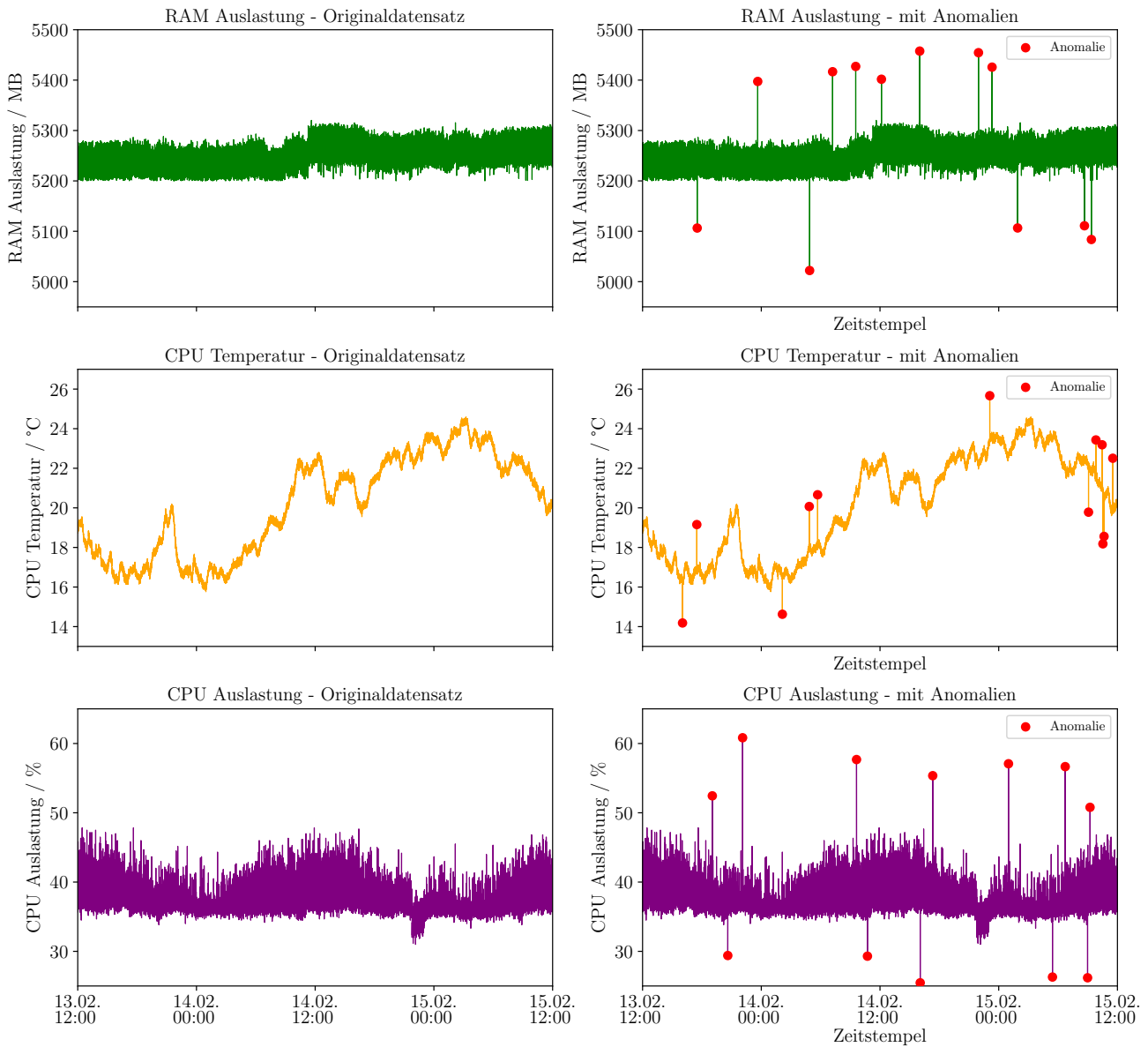


Abbildung 5.1: Drei Datensätze zur Gegenüberstellung und Evaluierung der beiden Algorithmen aus Tab. 4.2 mit synthetisch eingefügten Anomalien, die plausible Anomaliefälle darstellen.

wird die Rechendauer der Algorithmen gegenübergestellt. Jedoch soll die absolute Aussagekraft dieser Metrik nicht zu hoch bewertet werden, da die Algorithmen performancetechnisch nicht optimiert sind. Trotzdem soll es einen Überblick verschaffen und als zusätzlicher Vergleichsparameter herangezogen werden.

Für die Parametrisierung der beiden Algorithmen **History Based Outlier Score** (SWZ) und **Sliding Window Z-Score** (HBOS) wurden jeweils identische Fenstergrößen und Kontaminationsparameter übergeben. Die Kontamination gibt an, welcher relative Anteil eines Datensatzes anomal ist und ist

daher bereits eine Unsicherheit in der Analyse. Jedoch kann dies etwas eingedämmt werden, indem mehrere Durchläufe mit variabler Kontamination durchgeführt werden. Für die Tests der drei Datensätze wurden jeweils Kontaminationswerte von 0,02 %, 0,069 % und 0,1 % verwendet. Die genaue Kontamination entspricht den 0,069 %, da 12 aus den insgesamt jeweils 17281 Datenpunkten eine Anomalie darstellen.

Zudem werden mehrere verschiedene Fenstergrößen benutzt, um mögliche Fehlerquellen zu eliminieren. Insgesamt werden 200 verschiedene Fenster mit Größen zwischen 5 und 1000 Punkten eingesetzt.

5.2 Detektion von Subsequenzanomalien

5.3 Detektion von Korrelationsanomalien

5.4 Ergebnisse und Diskussion

Nach mehreren Testläufen sowie der Auswertung der Tests werden nun die Ergebnisse vorgestellt, diskutiert und eingeordnet.

5.4.1 Diskussion von HBOS und SWZ

Punktanomalien sind im Rahmen dieser Arbeit diejenigen Anomalien, denen im Feld wohl die geringste Bedeutung zugesprochen werden kann. Trotzdem wurden die beiden Algorithmen HBOS und SWZ Tests unterzogen und auf ihre Tauglichkeit überprüft. Gemessen werden sie jeweils am F1-Score sowie anhand der Rechendauer, die sie für die durchgeführten Tests benötigen.

Kontamination	RAM Auslastung	CPU Temperatur	CPU Auslastung
0,02 %	0,5	0,667	0,5
0,069 %	0,706	0,706	0,615
0,1 %	0,545	0,511	0,606

Tabelle 5.1: F1-Score von SWZ über drei verschiedene Kontaminationsparameter mit variabler Fenstergröße

Die Ergebnisse in Tab. 5.1 zeigen, dass der Kontaminationsparameter von 0,069 % die besten Ergebnisse liefern konnte. Trotz der Tatsache, dass es sich bei 0,069 % um die tatsächliche Kontamination im Datensatz handelt, konnte aber kein besseres Ergebnis als ein F1-Score von 0,706 für RAM Auslastung und CPU Temperatur erzielt werden. Das liegt auch an der Vielzahl der unterschiedlichen

Fenstergrößen. Im SWZ Algorithmus wurde festgelegt, dass ein Punkt als Anomalie gilt, wenn er von mehr als einem Drittel aller Fenster als solche erkannt wird.

Da für unterschiedlichen Fenstergrößen unterschiedliche Datenpunkte herausstehen können, ist diese Schwelle ebenfalls nicht optimiert, soll aber in jedem Fall verhindern, dass nur vereinzelt - und falsch - identifizierte Anomalien das Gesamtergebnis verfälschen. Die gleiche Schwelle von einem Drittel wurde auch für die Erkennung mit HBOS eingesetzt.

Kontamination	RAM Auslastung	CPU Temperatur	CPU Auslastung
0,02 %	1,0	1,0	0,667
0,069 %	1,0	1,0	0,667
0,1 %	1,0	1,0	0,667

Tabelle 5.2: F1-Score von HBOS über drei verschiedene Kontaminationsparameter mit variabler Fenstergröße

Auffällig ist vor allen Dingen der viel höhere F1-Score des HBOS Algorithmus mit perfekten Werten für die Datensätze der RAM Auslastung und der CPU Temperatur. Für die CPU Auslastung ist der F1-Score zwar ebenfalls besser, aber in geringerem Maße. Das liegt an der Natur der Daten, die sehr stark um mehrere Prozentpunkte zwischen zwei benachbarten Datenpunkten schwanken können und so die statistischen Größen, auf denen beide Algorithmen basieren, stark verfälschen.

Ebenfalls sticht die Unabhängigkeit des Algorithmus vom Kontaminationsparameter hervor, im Gegensatz zu SWZ, bei dem sich klare Unterschiede in der Genauigkeit zeigen. Ein deutlicher Nachteil von HBOS liegt jedoch in der Rechendauer. So haben die selben Auswertungen mit den selben Datensätzen mit HBOS 150 Minuten gedauert, während die Analyse mit SWZ bereits in 10 Sekunden abgeschlossen war, allerdings auf Kosten der Genauigkeit. Dazu sei nochmals erwähnt, dass die Implementierungen noch nicht performancetechnisch optimiert sind. Demnach ist nicht auszuschließen, dass HBOS auch in dieser Hinsicht viel besser werden kann.

Im Kontext eines Predictive Maintenance Systems eignet sich desweiteren auch eine Abwandlung des F1-Scores, um die beinhalteten Größen unterschiedlich zu gewichten. Der F1-Score leitet sich allgemein vom sog. $F\beta$ -Score ab, mit einem Gewichtungsfaktor von $\beta = 1$.

$$F_{\beta} = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \quad (5.1)$$

Der F1-Score gewichtet nach Gl. 5.1 False Positives und False Negatives mit $\beta = 1$ gleich, während ein $\beta = 2$ bzw. F2-Score die False Negatives stärker gewichtet. Dementsprechend wäre im Vergleich F1- zu F2-Score ein höherer F2-Score ein Indiz für ein besseres False Positive zu False Negative Verhältnis im Sinne dessen, dass False Negatives, also das fälschliche Verfehlen einer Anomalie, seltener vorkommen.

Kontamination	RAM Auslastung	CPU Temperatur	CPU Auslastung
0,02 %	0,385	0,556	0,385
0,069 %	0,857	0,857	0,645
0,1 %	0,75	0,723	0,725
Kontamination	Δ_{F_1, F_2}	Δ_{F_1, F_2}	Δ_{F_1, F_2}
0,02 %	− 0,115	+ 0,111	− 0,115
0,069 %	+ 0,151	+ 0,151	+ 0,029
0,1 %	+ 0,205	+ 0,212	+ 0,118

Tabelle 5.3: F2-Score von SWZ mit direktem Vergleich zwischen F1- und F2-Score

Kontamination	RAM Auslastung	CPU Temperatur	CPU Auslastung
0,02 %	1,0	1,0	0,797
0,069 %	1,0	1,0	0,797
0,1 %	1,0	1,0	0,797
Kontamination	Δ_{F_1, F_2}	Δ_{F_1, F_2}	Δ_{F_1, F_2}
0,02 %	0,0	0,0	+ 0,13
0,069 %	0,0	0,0	+ 0,13
0,1 %	0,0	0,0	+ 0,13

Tabelle 5.4: F2-Score von HBOS mit direktem Vergleich zwischen F1- und F2-Score

Es zeigt sich aus Tab. 5.3 und 5.4, dass eine Betrachtung des F2-Scores ein anderes Bild ergibt mit Blick auf die mögliche Vermeidung von False Negatives. HBOS agiert wesentlich robuster als SWZ und weist eindeutig die besseren Ergebnisse auf, auch wenn SWZ durch eine Betrachtung des F2-Scores einen ebenfalls robusten Eindruck macht, so ist die absolute Genauigkeit für HBOS trotzdem höher.

5.5 Diskussion von GrammarViz und SWIFD