Assistenzsystem zur Qualitätssicherung von IoT-Geräten basierend auf AutoML und SHAP

Jan Ewerszumrode, Marvin Schöne,
Stephan Godt, Martin Kohlhase

Center for Applied Data Science Gütersloh, FH Bielefeld
Schulstraße 10, 33330 Gütersloh
E-Mail: {jan.ewerszumrode, marvin.schoene,
stephan.godt, martin.kohlhase}@fh-bielefeld.de

Kurzfassung

Die Qualitätssicherung stellt einen wesentlichen Bestandteil der Produktentwicklung und Produktion dar, bei der die Potenziale der Daten aus IoT-Geräten bislang wenig Beachtung finden. IoT-Geräte ermöglichen eine Erfassung der tatsächlichen Gerätenutzung sowie auftretender Fehlerfälle, die in Summe als IoT-Gerätenutzungsdaten bezeichnet werden können. In diesem Beitrag wird ein Konzept und die Evaluation eines KI-basierten Assistenzsystems zur verbesserten Qualitätssicherung basierend auf IoT-Gerätenutzungsdaten vorgestellt. Das Konzept vereint eine kontinuierliche Fehlerüberwachung mittels deskriptiver Datenanalysen, eine automatisierte Modellbildung zum Erlernen von Zusammenhängen zwischen Gerätenutzung und auftretenden Fehlern mittels AutoML, und die Modellinterpretation mittels Shapley-Werten zur Bereitstellung hypothetischer Ursachen. Die Evaluation des Konzepts erfolgt anhand realer IoT-Gerätenutzungsdaten von über 40 Tsd. vernetzten Waschmaschinen. Als Ergebnis der Evaluation konnte eine zuvor unbekannte hypothetische Ursache für einen relevanten Fehlerfall auf Grundlage der Gerätenutzung identifiziert werden. Das Assistenzsystem unterstützt somit Domänenexpert:Innen des Qualitätsmanagements bei der explorativen Untersuchung von Kausalitäten zwischen Nutzung und Fehlern, wodurch

DOI: 10.58895/ksp/1000138532-17 erschienen in:
Proceedings - 31. Workshop Computational Intelligence : Berlin, 25. - 26. November 2021
DOI: 10.58895/ksp/1000138532 | https://www.ksp.kit.edu/site/books/m/10.58895/ksp/1000138532/

sich Verbesserungsmaßnahmen in Bezug auf die IoT-Geräte ableiten lassen können.

1 Einleitung

Zur Erfüllung bzw. Einhaltung aller Anforderungen an ein Produkt ist das Qualitätsmanagement (QM) bzw. die Qualitätssicherung (QS) einer der wesentlichen Bestandteile der Produktion. Quantitative Methoden der QS, wie z.B. *Six Sigma*, stützen sich dabei auf die Produktionsdaten aus MES- und ERP-Systemen. Hierbei wird jedoch der Großteil des Lebenszyklus von Produkten nicht berücksichtigt: Die tatsächliche Nutzung beim Endkunden. Veranschaulichen lässt sich dies anhand von Bild 1, in dem der Status quo und die Vision einer QS von IoT-Geräten gegenübergestellt werden. Neben den Produktionsdaten bieten sich Nutzungs- und Fehlerdaten der gefertigten Produkte bzw. Geräte als zusätzliche Datenquelle für die QS an. Das Internet of Things (IoT) ermöglicht es, diese Daten direkt aus den IoT-Geräten zu erheben. Ein Beispiel für IoT-Geräte sind vernetzte Waschmaschinen, die Daten zu den gewählten Waschprogrammen der Endkunden sowie aus ihrer geräte-internen Fehlerdiagnose erfassen (vgl. Bild 1b) und 1c)). Diese Daten lassen sich als IoT-Gerätenutzungsdaten bezeichnen, mit deren Hilfe eine kundenorientierte Qualitätssicherung (vgl. Bild 1a)) nach den Grundsätzen des Total-Quality-Managements erreicht werden kann.

Mithilfe dieser zusätzlichen Gerätenutzungsinformationen können hypothetische Ursachen für einen auftretenden Fehlerfall identifiziert werden. Die Domänenexpert:Innen des QM werden dabei besser und zuverlässiger unterstützt, unbekannte Kausalitäten aufzudecken, bekannte Kausalitäten zu bestätigen und neues Domänenwissen zu generieren. Der Vergleich einer QS auf Basis von IoT-Gerätenutzungsdaten mit dem Status quo zeigt jedoch Unterschiede in den Eigenschaften der Datenquellen und -basen (vgl. Bild 1c) und 1d)). Diese äußern sich in Form großer, heterogener und hochdimensionaler Datenmengen in einem kontinuierlichen Datenstrom. Durch Verfahren der *künstlichen Intelligenz* (KI) bzw. des *Machine Learnings* (ML) können diese Datenmengen ausgewertet werden (vgl. Bild 1e)).

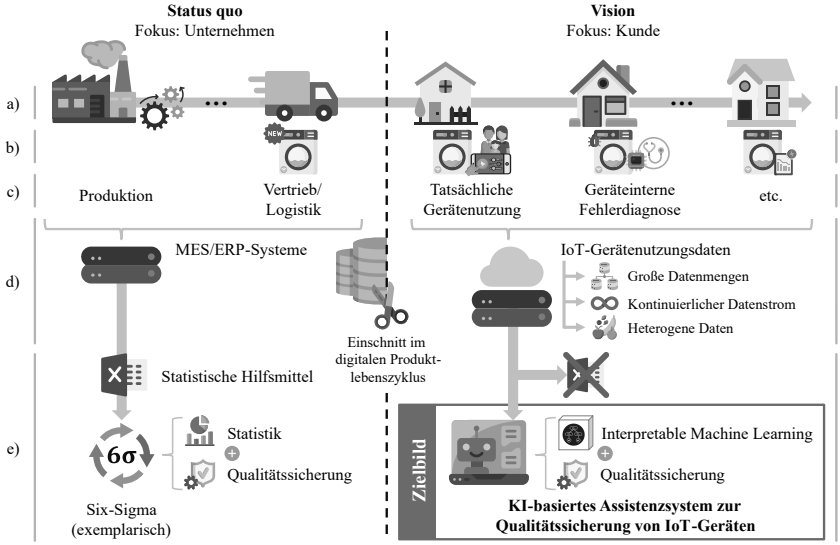

Bild 1: Status quo und Vision einer QS bzgl. des a) Lebenszyklus von b) IoT-Geräten als zusätzliche c) Datenquelle. Anhand von IoT-Gerätenutzungsdaten als d) Datenbasis ergeben sich neue Potenziale zur e) Auswertung mittels KI und IML.

Infolge der Anforderungen an Präzision in der QS werden nachvollziehbare Entscheidungen in der Datenauswertung benötigt, wodurch sich Verfahren des *Interpretable Machine Learnings* (IML) anbieten, um die zumeist intransparenten Entscheidungen des ML erklärbar zu machen. Vergleichbare technische Systeme in [1, 2, 3] lösten bereits ähnliche Aufgaben zur Generierung neuen Wissens mittels IML-Verfahren. Eine derartige Anwendung im Bereich der QS für IoT-Geräte steht jedoch noch aus. Das Potenzial von IML wird zudem im Gesundheitswesen ersichtlich [4, 5], wo ebenfalls ein hohes Maß an Präzision und Sorgfalt gefordert ist.

In diesem Beitrag wird ein Konzept für ein Assistenzsystem zur QS basierend auf IoT-Gerätenutzungsdaten und Verfahren des ML sowie IML vorgestellt, welches anhand realer Daten vernetzter Waschmaschinen evaluiert wird. Ziel des Assistenzsystems ist es, Domänenexpert:Innen aus dem QM bei der explorativen Untersuchung von Fehlerfällen, die während der Gerätenutzung auftreten, zu unterstützen. In den Daten enthaltene Zusammenhänge zwischen

Gerätenutzung und auftretenden Fehlerfällen werden in ML-Modelle mittels *Automated Machine Learning* (AutoML) verdichtet. Anschließend werden die erlernten Zusammenhänge zur Bereitstellung neuer hypothetischer Fehlerursachen durch SHAP aus den ML-Modellen extrahiert und grafisch dargestellt.

2 Theoretische Grundlagen & Stand der Technik

Wesentlich für das in diesem Beitrag vorgestellte Konzept sind das AutoML zur Approximation der Funktion $f : \mathscr{X} \rightarrow \mathscr{Y}$ zwischen Gerätenutzung $\mathscr{X}$ und Fehlerfall $\mathscr{Y}$ der IoT-Geräte, sowie das IML zur Untersuchung des erlernten Zusammenhangs von f.

Automated Machine Learning (AutoML). Unter AutoML ist ein Prozess für die automatisierte Entwicklung von ML-Pipelines zur Approximation von f aus einem dedizierten Datensatz $\mathscr{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$ zu verstehen. Der Datensatz besteht aus N gelabelten Instanzen, für die $\mathbf{x}^{(i)} \in \mathscr{X}$ und $y^{(i)} \in \mathscr{Y}$ gilt. Der Prozess beinhaltet die Auswahl, Kombination sowie Parametrisierung von ML-Algorithmen. Formalisieren lässt sich dieser Prozess als Optimierungsproblem, das als *Combined Algorithm Selection and Hyperparameter optimization* (CASH) bezeichnet wird [6]:

$$A^*_{\lambda^*} \in \underset{A^{(k)} \in \mathscr{A}, \lambda \in \Lambda^{(k)}}{\operatorname{argmin}} \; E[L(A^{(k)}_{\lambda}, \mathscr{D}_{\text{train}}, \mathscr{D}_{\text{valid}})] \tag{1}$$

mit λ als Hyperparameter aus den Hyperparameterräumen $\Lambda^{(1)}, ..., \Lambda^{(K)}$ und den zugehörigen ML-Algorithmen $\mathscr{A} = \{A^{(1)}, ..., A^{(K)}\}$ für $k = 1, ..., K$. Zudem bezeichnet $L(A^{(k)}_{\lambda}, \mathscr{D}_{\text{train}}, \mathscr{D}_{\text{valid}})$ die Fehlerfunktion für einen auf den Trainingsdatensatz $\mathscr{D}_{\text{train}} \subset \mathscr{D}$ angewendeten und am Validierungsdatensatz $\mathscr{D}_{\text{valid}} \subset \mathscr{D}$ getesteten Algorithmus $A^{(k)}_{\lambda}$. Eine gängige Methode zur Lösung des CASH-Problems besteht darin, die Algorithmenauswahl als zusätzlichen Hyperparameter zu betrachten, wodurch etablierte Hyperparameteroptimierungen verwendet werden können, wie z.B. SMAC [7], *Hyperband* [8] und BOHB [9]. Unter Berücksichtigung der natürlichen Hierarchie bzgl. Auswahl und Parametrisierung von ML-Algorithmen ($A^{(k)}$ bedingt $\Lambda^{(k)}$) stellen hierarchische Suchansätze, wie z.B. *ML-Plan* [10], eine

weitere Lösung des CASH-Problems dar. Des Weiteren bieten sich Verfahren des *Reinforcement Learnings* an, in denen der Agent geeignete $A^{(k)}$ und λ auswählt [11].

Interpretable Machine Learning (IML). IML bezeichnet Methoden, die das Verhalten und die Vorhersagen von ML-Modellen für den Menschen verständlich machen [12]. Während sich die lokale Interpretierbarkeit auf die Erklärung einzelner Prädiktionen bezieht, ist unter einer globalen Interpretierbarkeit die Erklärung des gesamten Modellverhaltens zu verstehen. Wesentlich für die Interpretierbarkeit von ML-Modellen ist der Einfluss eingehender Merkmale auf die resultierenden Prädiktionen [13]. Gegenüber inhärent interpretierbaren ML-Modellen (z.B. *Decision Trees*) eignen sich besonders modellagnostische post-hoc Methoden, welche die Interpretation vom ML-Modell separieren und eine schwerpunktmäßige Betrachtung der Modellgenauigkeit ermöglichen. Shapley-Werte bieten dazu aufgrund hoher Übereinstimmungen mit der menschlichen Intuition und einer fundierten theoretischen Grundlage [14] gegenüber vergleichbaren Methoden, wie LIME [15] oder PFI [20], ein großes Potenzial. Ausgehend von einer lokalen Prädiktion $\hat{f}(\mathbf{x}^{(i)})$ lassen sich die Shapley-Werte $\phi_j^{(i)}$ für jedes Merkmal j über eine gewichtete Summe darstellen, die den Einfluss jedes zum ML-Modell hinzugefügten Merkmals wiedergibt, gemittelt über alle Kombinationen verfügbarer Merkmale [14]:

$$\phi_j^{(i)} = \sum_{\mathscr{S} \subseteq \mathscr{S}_{\text{all}} \setminus \{j\}} \frac{|\mathscr{S}|!(M - |\mathscr{S}| - 1)!}{M!} \cdot (\hat{f}_{\mathbf{x}}(\mathscr{S} \cup \{j\}) - \hat{f}_{\mathbf{x}}(\mathscr{S})) \qquad (2)$$

mit der Merkmalsanzahl M, der Menge aller Merkmale $\mathscr{S}_{\text{all}}$, und der Prädiktion $\hat{f}_{\mathbf{x}}(\mathscr{S}) = E[\hat{f}(\mathbf{x}^{(i)}) | \mathbf{x}_{\mathscr{S}}]$ einer ausgewählten und auf die Teilmenge $\mathscr{S}$ beschränkten Kombination an Merkmalen $\mathbf{x}_{\mathscr{S}}$ des Eingabevektors. Des Weiteren besitzen Shapley-Werte folgende Eigenschaften: a) Die Summe der Shapley-Werte aller Merkmale ist gleich der Differenz aus $\hat{f}(\mathbf{x}^{(i)})$ minus der mittleren Prädiktion $E[\hat{f}(\mathbf{x})]$ einer zufälligen Instanz $\mathbf{x} \in \mathscr{X}$; b) $\phi_j^{(i)} = 0$, wenn das Merkmal j keinen Einfluss auf die Prädiktion hat; c) wenn die Werte zweier Merkmale über alle $\mathscr{S}$ hinweg eine symmetrische Auswirkung haben, ergeben sich die gleichen Shapley-Werte und d) ihre lokalen Einflüsse sind über $\mathbf{x}^{(i)}$ hinweg additiv [13]. Die exakte Berechnung der Shapley-Werte äußert sich jedoch aufgrund 2^M möglicher Teilmengen für $\mathscr{S}$ als NP-schwer. Sampling-basierte Ap-

proximationen der Shapley-Werte, wie *Kernel*-SHAP [14], ermöglichen zwar die Berechnung lokaler Interpretationen, scheitern jedoch an einer globalen Interpretation für große Datensätze [16]. Die Anpassung von *Kernel*-SHAP für baumartige ML-Modelle führt zu *Tree*-SHAP und ermöglicht eine Reduktion der ursprünglich exponentiellen auf eine polynomielle Berechnungszeit. Die zugrundeliegende Idee von *Tree*-SHAP besteht darin, den Anteil aller möglichen Teilmengen in jedes der Blätter des Baums rekursiv zu verfolgen. Da die hierdurch erzeugten Shapley-Werte auf bedingte Erwartungswerte beruhen, ist eine gesonderte Behandlung abhängiger Merkmale erforderlich, welche die Interpretation verfälschen können (z.B. die Schätzung von $\phi_j^{(i)} \neq 0$ für Merkmale ohne Einfluss) [12]. Unter Berücksichtigung eines Backgrounddatensatzes lässt sich diese Abhängigkeit nach den Regeln der zufälligen Inferenz beheben, wodurch die zuvor aufgeführten Eigenschaften weiterhin gültig sind und sich zusätzlich eine Berechnungszeit proportional zur Größe des Backgrounddatensatzes ergibt [17]. Aufgrund der Komplexitätsreduktion ermöglicht *Tree*-SHAP eine globale Interpretierbarkeit basierend auf vielen lokalen Interpretationen.

3 Konzept des KI-basierten Assistenzsystems

Der Nutzen des KI-basierten Assistenzsystems liegt in der Bereitstellung hypothetischer Ursachen für Fehlerfälle $F\#_l$, die aufgrund der Gerätenutzung auftreten. Diese Ursachen sollen den Domänenexpert:Innen helfen, unbekannte Kausalitäten zu identifizieren und bekannte Kausalitäten zu bestätigen. Die Anzahl und Art der Fehlerfälle $F\#_l$ werden von dem jeweiligen Fehlerdiagnosesystem der IoT-Geräte vorgegeben, wie z.B. *Kurzschluss im Gerät.* Das Konzept dieses KI-basierten Assistenzsystems zur QS von IoT-Geräten ist in Bild 2 dargestellt. Als Eingabe in das System dienen IoT-Gerätenutzungsdaten (bestehend aus Nutzungs- und Fehlerdaten der IoT-Geräte), während als Ausgabe Diagramme zur Erklärung des erlernten Zusammenhangs zwischen Gerätenutzung und Fehlerfällen dienen. Hierbei vereinigt das Konzept die drei Schritte 1) einer kontinuierlichen Fehlerüberwachung mittels Methoden der deskriptiven Datenanalyse, namens QS-Watchdog, 2) einer automatisierten Modellbildung zur Klassifikation eines Fehlerfalls anhand zugehöriger Nutzungsdaten mittels

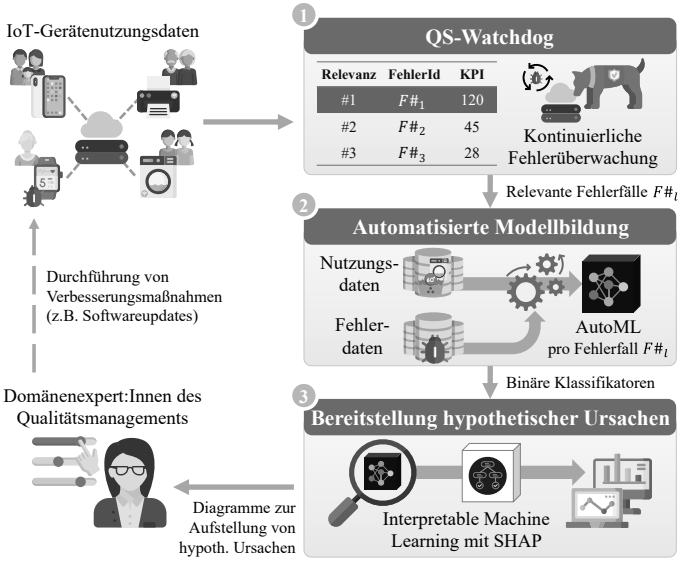

Bild 2: Assistenzsystem zur QS mittels IoT-Gerätenutzungsdaten als Eingabe, einer 1) kontinuierlichen Fehlerüberwachung, 2) automatisierten Modellbildung mittels AutoML und 3) Bereitstellung hypothetischer Ursachen durch IML für Domänenexpert:Innen des QM

AutoML und 3) die globale Interpretation zuvor trainierter Modelle mittels Shapley-Werten in Form von aussagekräftigen Diagrammen.

Qualitätssicherungs-Watchdog (QS-Watchdog). Aufgrund eines starken Ungleichgewichts zwischen intakten und defekten Geräten, bedarf es einer Selektion von Fehlerfällen, um eine stichhaltige Menge an Labels in Form von betroffenen Geräten für die weiteren Schritte zu gewährleisten. Gleichzeitig ist die Beobachtung von Trends bei relevanten Fehlerfällen, die den Kundennutzen beeinträchtigen, für die Domänenexpert:Innen des QM von Interesse, was den Einsatz einer kontinuierlichen Fehlerüberwachung der IoT-Geräte notwendig macht. Hierfür ist es erforderlich, eine Kennzahl zu definieren, die relevante von nicht-relevanten Fehlerfällen unterscheidet und somit eine Priorisierung ermöglicht. Aus dem Bereich der Zuverlässigkeitsanalyse bieten sich Kennzahlen an, welche die Häufigkeit relevanter Fehler mit der Betriebszeit in Bezug setzen. Eine der hierbei wichtigsten Kennzahlen stellt die Fehlerrate $\lambda(t)$ dar. Für

praktische Anwendungen kann $\lambda(t) = \lambda$ angenommen werden, wodurch sich λ über $\hat{\lambda} = n/T$ schätzen lässt, wobei T die kumulierte Betriebszeit und n die absolute Anzahl an Fehlern ist [18]. Als Kennzahl für den QS-Watchdog ergibt sich demnach die geschätzte Fehlerrate $\hat{\lambda}_{F\#_l}$ pro Fehlerfall $F\#_l$ für ein zuvor festgelegtes Zeitintervall:

$$\hat{\lambda}_{F\#_l} = n_{F\#_l}/T. \tag{3}$$

Wichtig zu erwähnen ist, dass sich n auf die Anzahl repräsentativer Fehler bezieht, die dadurch definiert werden, dass sie auf eine Unterbrechung der Gerätefunktionalität (z.B. Stopp eines Waschprogramms) folgen müssen.

Automatisierte Modellbildung. Für eine präzise Erklärung des Zusammenhangs zwischen Gerätenutzung und Fehlerfällen basieren die zugrundeliegenden ML-Modelle auf einer pro Fehlerfall $F\#_l$ konzentrierten binären Klassifikation $y_{F\#_l} \in \{0 := F\#_l \text{ trat nicht auf}, 1 := F\#_l \text{ trat auf}\}$. Anstelle eines universellen Klassifikators ist es somit erforderlich, mehrere binäre Klassifikatoren je Fehlerfall zu trainieren. Um das Training der Klassifikatoren dennoch domänenunabhängig und automatisiert zu gestalten, erfolgt dieser Schritt mit Hilfe von AutoML. In diesem Beitrag wird das AutoML von Databricks [19] verwendet, welches die folgenden ML-Algorithmen beinhaltet: *Logistic Regression*, *Decision Trees*, *Random Forests* [20], *XGBoost* (XGB) [21] und *LightGBM* (LGBM) [22]. Zusätzlich ergibt sich durch die Beschränkung auf binäre Klassifikationen eine deutlich verringerte Komplexität des Suchraums nach (1), die sich positiv auf die angestrebte Generalisierbarkeit auswirkt [23]. Zur Evaluation der AutoML-Modelle werden die IoT-Gerätenutzungsdaten in Testdaten $\mathscr{D}_{\text{test}}$ und Trainingsdaten aufgeteilt (25%, 75%), wobei die Trainingsdaten jeweils vom AutoML in separate Trainingsmengen $\mathscr{D}_{\text{train}}$ und Validierungsmengen $\mathscr{D}_{\text{valid}}$ stratifiziert unterteilt (75%, 25%) werden. In Anbetracht der stark ungleich verteilten Klassen findet die Beurteilung erfolgreicher Klassifikatoren mittels des *F1-Scores* F_1 für $\mathscr{D}_{\text{test}}$ statt. Ab einem $F_1 > 0.9$ für $\mathscr{D}_{\text{test}}$ wird dem Klassifikator eine ausreichende Generalisierbarkeit des Datensatzes für eine anschließende Interpretation zugesprochen. Andernfalls wird der Klassifikator für weitere Interpretationen zurückgewiesen.

Bevor AutoML angewendet werden kann, müssen die binären Klassen aus der Gerätemenge extrahiert werden, die keine Fehlerfälle beinhalten und somit das Normalverhalten charakterisieren, sowie aus defekten Geräten bzgl. des betrachteten Fehlerfalls bestehen. Aufgrund des Ungleichgewichts zwischen intakten und defekten Geräten, bieten sich Under- und Oversampling-Verfahren zur Bereinigung der ungleichen Verteilung an. Während beim Undersampling eine zufällige Untermenge aus der Klasse intakter Geräte ausgewählt wird, werden beim Oversampling neue synthetische Daten aus der Klasse defekter Geräte, z.B. mittels SMOTE bzw. SMOTE-NC [25] generiert, wodurch sich bei beiden Verfahren ein ausbalancierter Trainingsdatensatz ergibt. Um möglichst signifikante Merkmale zu verwenden, werden diese mittels FRESH [26] anhand von p-Werten auf ihre Signifikanz gegenüber der Klasse getestet und selektiert. Angesichts der Betrachtung von Geräten über die gesamte Betriebszeit kann durch eine gesonderte Betrachtung der Nutzungsverlauf defekter Geräte bis zum *ersten relevanten Fehlerfall* hergestellt werden. Hierdurch ließen sich bspw. Nutzungsmuster in den Anfangsphasen des Gerätes bis zur ersten Reparatur identifizieren.

Bereitstellung hypothetischer Ursachen. Zur Interpretation des erlernten Zusammenhangs zwischen Nutzung und Fehler für eine Menge an intakten und defekten Geräten bedarf es einer globalen Interpretation des erlernten Modellverhaltens. Dieses lässt sich über den Einfluss eingehender Merkmale auf das Prädiktionsergebnis mittels Shapley-Werten beschreiben und interpretieren. Die Ermittlung der Shapley-Werte erfolgt durch *Tree*-SHAP, wodurch die globale Interpretation über eine Vielzahl lokaler Interpretationen der zuvor separierten Testmengen $\mathscr{D}_{\text{test}}$ erfolgt. Da es sich bei der automatisierten Modellbildung um binäre Klassifikatoren handelt, dient hierbei die prädizierte Wahrscheinlichkeit $\hat{f}(\mathbf{x}^{(i)})$ für ein defektes Gerät als Ausgabe. Für einen idealen binären Klassifikator ($F_1 = 1.0$) mit ausbalancierten Daten ergibt sich $E[\hat{f}(\mathbf{x})] = 0.5$. Wie in Bild 3a) schematisch dargestellt, kann $\hat{f}(\mathbf{x}^{(i)})$ für eine lokale Interpretation ausgehend von $E[\hat{f}(\mathbf{x})]$ durch Aufsummieren aller $\phi_j^{(i)}$ über j nachvollzogen werden. Zur Bestimmung des globalen Merkmalseinflusses (*Feature Importance*) I_j eignet sich der Mittelwert der absoluten Shapley-Werte pro

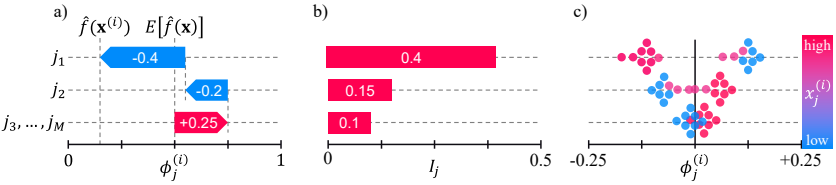

Bild 3: Schematische Diagramme zur Interpretation mittels Shapley-Werten für a) lokale Interpretationen, b) globale Merkmalseinflüsse und c) einer Übersichtsabbildung.

Merkmal:

$$I_j = \sum_{\{i \in \mathbb{N} \mid (\mathbf{x}^{(i)}, y^{(i)}) \in \mathscr{D}_{\text{test}}\}} \frac{|\phi_j^{(i)}|}{|\mathscr{D}_{\text{test}}|} \tag{4}$$

für alle Instanzen aus $\mathscr{D}_{\text{test}}$. Für eine erste globale Interpretation lässt sich I_j, wie in Bild 3b), über ein Balkendiagramm darstellen. In Verbindung mit den tatsächlichen Merkmalswerten $x_j^{(i)}$ zur Gerätenutzung jeder lokalen Interpretation und den zugehörigen Shapley-Werten $\phi_j^{(i)}$, lassen sich globale wechselseitige Beziehungen zwischen der Gerätenutzung und auftretenden Fehlerfällen in einer sogenannten Übersichtsabbildung identifizieren. Diese Übersichtsabbildung ist in Bild 3c) schematisch dargestellt. Hierbei gibt jeder Punkt eine lokale Interpretation pro Merkmal auf der y-Achse wieder, während die Shapley-Werte auf der x-Achse aufgetragen sind und der Merkmalswert über die Farbe der Punktfüllung beschrieben wird. Für einen Überblick über die Verteilung werden überlappende Punkte in Richtung der y-Achse gestapelt. Außerdem wird die Beschreibung dieser Merkmale in allen Diagrammen auf maximal drei beschränkt, da eine für den Menschen gute Erklärung weniger durch eine allumfassende als vielmehr durch eine präzise Erklärung ausgewählter Merkmale erreicht wird [24]. Der Einfluss aller restlichen Merkmale wird über diese Beschränkung hinaus zusammengefasst.

4 Evaluation

Zur Evaluation des Konzepts dienen reale IoT-Gerätenutzungsdaten von vernetzten Waschmaschinen, die im nachfolgenden Abschnitt 4.1 vorgestellt werden. Darauf aufbauend erfolgt in Abschnitt 4.2 eine schrittweise Evaluation der

einzelnen Bestandteile des Konzepts sowie in Abschnitt 4.3 eine Vorstellung und Diskussion der Evaluationsergebnisse.

4.1 Rahmenbedingungen & Versuchsaufbau

Der Evaluationsdatensatz des verwendeten Versuchsaufbaus umfasst reale IoT-Gerätenutzungsdaten von $>$ 40 Tsd. vernetzten Waschmaschinen, die Daten zu $>$ 10 Mio. Waschprogrammen beinhalten und über einen Zeitraum vom August 2019 bis Mai 2021 erhoben wurden. Hinter jedem dieser Waschprogramme verbergen sich ereignisdiskrete Daten, welche sich in Nutzungs-[1] und Fehlerdaten[2] aufteilen lassen. Die Vielzahl dieser Ereignisse führt zu einem Datenstrom mit fortlaufend neu hinzukommenden Geräten, gerätespezifischen Zeitreihen vereinzelter Nutzungs- und Fehlerdaten sowie Inkonsistenzen realer Datenproduzenten. Um diesen Datenstrom effizient zu verwalten und über alle Geräte hinweg vergleichbar zu machen, bietet sich eine aggregierte Sicht der IoT-Gerätenutzungsdaten an [26]. In dieser aggregierten Sicht werden die gerätespezifischen Zeitreihen der Nutzungs- und Fehlerdaten zusammengefasst, sodass jedes Element die bisherige Betriebszeit eines Gerätes beschreibt, inklusive einer Auflistung bereits aufgetretener Fehlerfälle. Ein Gerät wird als intakt bezeichnet, wenn während der bisherigen Betriebszeit keinerlei Fehlerfälle aufgetreten sind. Die Aggregation der Nutzungsdaten erfolgt über die Extraktion etablierter deskriptiver Merkmale[3] aus den dynamischen Variablen eines Waschprogramms. Die Extraktion der booleschen Programmextras erfolgt anhand der aktiven Nutzung dieser Extras im Verhältnis zur Anzahl durchgeführter Waschprogramme pro Gerät[4]. Für jeden Fehlerfall $F\#_l$ wird nach diesem

[1] Nutzungsdaten lassen sich in vier Gruppen clustern: 1) *Geräteeigenschaften* beinhaltet den Gerätetyp und die Softwareversion; 2) *Programmanwahl* beinhaltet das Waschprogramm (z.B. Feinwäsche, etc.), zusätzliche Einstellparameter (Temperatur, Schleuderdrehzahl) und Programmextras (z.B. Stärken/Weichspülen, etc.); 3) *Programmzustand* beinhaltet den Energie-, Wasserverbrauch und die Programmdauer abgeschlossener Waschprogramme; 4) Die *automatische Dosierung* beinhaltet die Dosierungsmenge und den verbleibenden Waschmittelinhalt.

[2] Fehlerdaten beinhalten Informationen zu aufgetretenen Fehlerfällen während eines Waschprogramms, die von der internen Gerätefehlerdiagnose bestimmt werden.

[3] Folgende deskriptive Merkmale werden berechnet: Arithmetischer Mittelwert, Median, Varianz, Standardabweichung, Minimum, Maximum, RMS und Summe aller Werte.

[4] Bsp.: Das Programmextra Stärken/Weichspülen wird zum Merkmal Stärken/Weichspülen%.

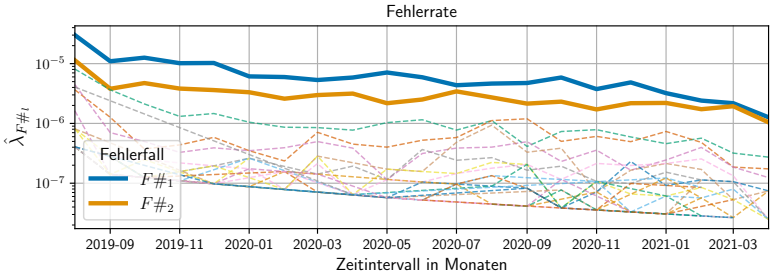

Bild 4: Fehlerrate aller spezifizierten Fehlerfälle pro Monate vom 08.2019 bis zum 04.2021, inklusive der hervorgehobenen Fehlerfälle $F\#_1$ und $F\#_2$.

Schema eine aggregierte Sicht der Nutzungsdaten aus intakten und defekten Geräten erzeugt, woraus sich die entsprechenden Datensätze $\mathscr{D}_{F\#_l}$ ergeben.

4.2 Schrittweise Evaluation des Konzepts

Evaluation des QS-Watchdogs. Mittels der Fehlerdaten des Evaluationsdatensatzes lässt sich die kontinuierliche Fehlerüberwachung für jeden Fehlerfall $F\#_l$ nachbilden, wobei ein Zeitintervall von einem Monat betrachtet wird. Die Fehlerrate $\hat{\lambda}_{F\#_l}$ lässt sich nach (3) bestimmen. In Bild 4 ist die Fehlerrate über den betrachteten Zeitraum pro Fehlerfall $F\#_l$ dargestellt, wobei $\hat{\lambda}_{F\#_l}$ logarithmisch abgebildet wird. Insgesamt ist ein deutlicher Rückgang von $\hat{\lambda}_{F\#_l}$ zu erkennen, welches jedoch auf die anfänglich geringe Betriebszeit weniger IoT-Geräte im Evaluationsdatensatz zurückzuführen ist. Auffällig sind hierbei die Fehlerfälle $F\#_1$ und $F\#_2$, die sich von der Gesamtmenge an Fehlern absetzen, eine demnach hohe Relevanz aufweisen und zur weiteren Betrachtung der automatisierten Modellbildung herangezogen werden.

Evaluation der automatisierten Modellbildung. Basierend auf den identifizierten Fehlerfällen $F\#_1$ und $F\#_2$ des QS-Watchdogs findet im Folgenden die Modellbildung mittels AutoML zur Generierung von $\hat{f} : \mathscr{X} \rightarrow \mathscr{Y}$ anhand der Evaluationsdatensätze $\mathscr{D}_{F\#_1}$ und $\mathscr{D}_{F\#_2}$ statt. Durch die stratifizierte Aufteilung dieser Datensätze ergibt sich für jeden der Fehlerfälle ein Testumfang von > 10 Tsd. Geräte, wobei $F\#_1$ einen Anteil von $4,7\%$ und $F\#_2$ von $2,2\%$ defekter

Tabelle 1: Evaluation der automatisierten Modellbildung anhand der besten AutoML-Runs für alle Datensatzkonfigurationen der Fehlerfälle $F\#_1$ und $F\#_2$ mittels separatem Testdatensatz. Erfolgreiche Runs mit $F_1 > 0.9$ sind fett markiert.

Datensatzkonfiguration					Testergebnisse der besten AutoML-Runs				
Fehlerfall		#S	#B	#nM	Klassif.	F_1	*PPV*	*TPR*	*MMC*
$F\#_1$	$\mathscr{D}_{F\#_{1.1}}$	-	-	126	LGBM	**0.984**	0.998	0.971	0.984
	$\mathscr{D}_{F\#_{1.2}}$	U	-	100	LGBM	0.541	0.374	0.979	0.578
	$\mathscr{D}_{F\#_{1.3}}$	O	-	133	LGBM	**0.987**	0.992	0.981	0.986
	$\mathscr{D}_{F\#_{1.4}}$	-	✓	112	XGB	**0.958**	0.993	0.926	0.957
	$\mathscr{D}_{F\#_{1.5}}$	U	✓	120	LGBM	0.206	0.115	0.984	0.263
	$\mathscr{D}_{F\#_{1.6}}$	O	✓	131	XGB	**0.957**	0.996	0.922	0.956
$F\#_2$	$\mathscr{D}_{F\#_{2.1}}$	-	-	123	LGBM	0.105	0.341	0.062	0.138
	$\mathscr{D}_{F\#_{2.2}}$	U	-	91	LGBM	0.045	0.023	0.712	0.012
	$\mathscr{D}_{F\#_{2.3}}$	O	-	141	LGBM	0.192	0.578	0.115	0.251
	$\mathscr{D}_{F\#_{2.4}}$	-	✓	100	XGB	0.281	0.621	0.181	0.329
	$\mathscr{D}_{F\#_{2.5}}$	U	✓	112	XGB	0.064	0.033	0.854	0.087
	$\mathscr{D}_{F\#_{2.6}}$	O	✓	134	LGBM	0.446	0.723	0.323	0.476

#S-Samplingart: kein Sampling (-), Undersampling (U), Oversampling (O);
#B-Betriebszeit bis Fehler: nein (-), ja (✓); #nM-Merkmalsanzahl

IoT-Geräte aufweist. Zur Adressierung dieses Ungleichgewichts kommen die zusätzlichen Maßnahmen des Under- und Oversampling zum Einsatz. Hinzu kommt die zusätzliche Betrachtung der Gerätebetriebszeit bis zum ersten Fehlerfall, wodurch sich insgesamt sechs verschiedene Datensatzkonfigurationen in Form der aggregierten IoT-Gerätenutzungsdaten ergeben: Kein Sampling, Undersampling, Oversampling jeweils pro gesamter Betriebszeit sowie bis zum ersten Fehlerfall. Aufgrund der Merkmalsauswahl mittels FRESH variiert die Anzahl der Merkmale in jeder Datensatzkonfiguration. Für jede der insgesamt 12 Datensatzkonfiguration wird ein AutoML-Run gestartet, welcher 200 Versuche beinhaltet. Die Evaluationsergebnisse der besten AutoML-Runs mittels der separierten Testmengen sind in Tabelle 1 aufgeführt. Neben F_1 als primäre Bewertungsmetrik werden hier *Precision PPV* und *Recall TPR* als zusätzliche Metriken aufgeführt. Um Fehlinterpretationen aufgrund des starken Klassenungleichgewichts zu vermeiden, wird auf die Evaluation mittels *Accuracy* verzichtet und der *Matthews-Correlation-Coefficient MCC* zur Betrachtung der gesamten Konfusionsmatrix verwendet [27]. Die Evaluation aus

Tabelle 1 zeigt, dass vier Klassifikatoren für den Fehlerfall $F\#_1$ in der Lage sind, eine ausreichende Generalisierbarkeit von $\hat{f}$ für $\mathscr{D}_{F\#_1}$ zu erlernen. Auffällig ist zudem, dass beim Undersampling zwar ein guter *Recall* in allen Fehlerfällen erzielt werden konnte, der jedoch zu Lasten der *Precision* fällt. Für den Fehlerfall $F\#_2$ konnte diese Generalisierbarkeit jedoch nicht erzielt werden, wodurch dieser von einer darauffolgenden Interpretation zurückgewiesen wird. Zur weiteren Evaluation der Bereitstellung hypothetischer Ursachen werden demnach die erfolgreichen Klassifikatoren der Datensatzkonfigurationen $\mathscr{D}_{F\#_{1.1}}$, $\mathscr{D}_{F\#_{1.3}}$, $\mathscr{D}_{F\#_{1.4}}$ und $\mathscr{D}_{F\#_{1.6}}$ für den Fehlerfall $F\#_1$ mittels Verfahren des IML untersucht.

Evaluation der Bereitstellung hypothetischer Ursachen. Zur Bereitstellung hypothetischer Ursachen des Fehlerfalls $F\#_1$ werden die Shapley-Werte der vier erfolgreichen Klassifikatoren mittels *Tree*-SHAP für die zuvor bestimmten Testdaten berechnet. Aufgrund des Klassenungleichgewichts wird ein zusätzliches Undersampling durchgeführt, um die Klassen der intakten Geräte zu reduzieren und eine Fehlinterpretation der Modelle zu vermeiden. Der für die Berechnung verwendete Backgrounddatensatz entspricht hierbei den reduzierten Trainingsdaten der Datensatzkonfigurationen, während für die tatsächliche Schätzung der Shapley-Werte die jeweiligen reduzierten Testdaten verwendet werden. Für diese ausbalancierten Testdaten ergibt sich für die betrachteten Klassifikatoren eine Genauigkeit von ≥ 0.961, die eine stichhaltige Interpretation dieser Shapley-Werte erlaubt.

In Bild 5 sind drei lokale Interpretationen des Klassifikators aus $\mathscr{D}_{F\#_{1.1}}$ für verschiedene Waschmaschinen gegenübergestellt. Der Klassifikator besitzt einen Erwartungswert von $E[\hat{f}(\mathbf{x})] = 0.497$ und gibt für die Waschmaschine in 5a) eine Wahrscheinlichkeit von $\hat{f}(\mathbf{x}^{(1)}) = 0.001$ bzgl. des Fehlerfalls $F\#_1$ aus. Der Shapley-Wert des Merkmals Stärken/Weichspülen% gibt hierbei den größten Einfluss mit einem Merkmalswert von 0% wieder. Genau gegensätzlich verhält es sich mit der Waschmaschine aus 5b), die zu 100% Stärken/Weichspülen verwendet hat und zu $\hat{f}(\mathbf{x}^{(2)}) = 0.999$ als defekt klassifiziert wird. Bei der Waschmaschine in 5c) stellt sich die Ausgabe als weniger eindeutig heraus. Während ein Großteil der Merkmale dazu führt, dass der Klassifikator zu einem defekten Gerät tendiert, senkt der Verzicht auf Stärken/Weichspülen diese Tendenz von ursprünglich 0.915 auf $\hat{f}(\mathbf{x}^{(3)}) = 0.585$.

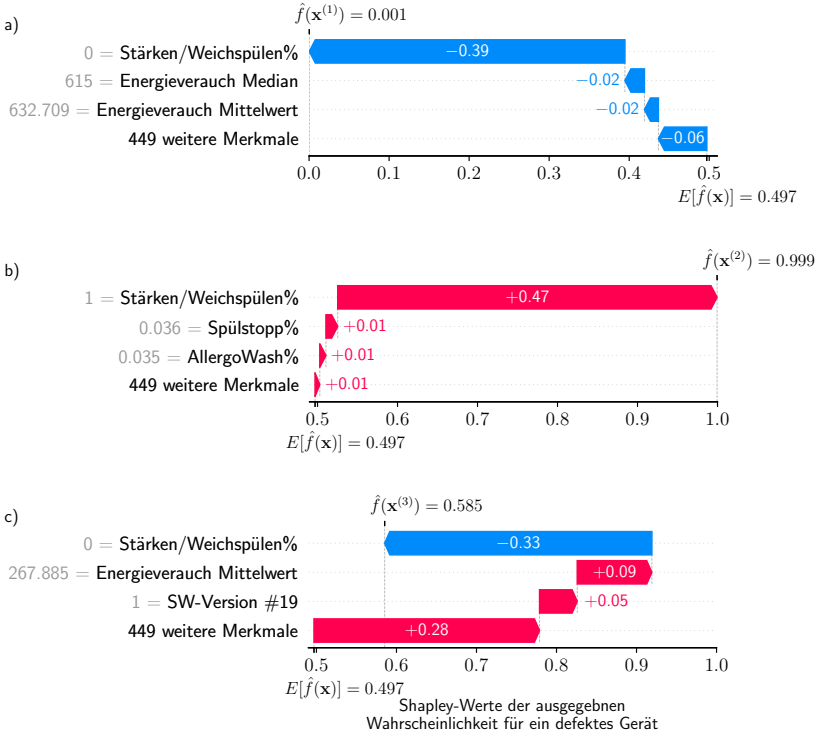

Bild 5: Lokale Interpretationen der ausgegebenen Wahrscheinlichkeit für ein defektes Gerät bzgl. des Fehlers $F\#_1$ mittels Shapley-Werten für drei verschiedene IoT-Geräte mit a) einem intakten Gerät, b) einem defekten Gerät und c) einer unsicheren Prädiktion.

Für eine erste globale Interpretation erfolgt die Ermittlung des Einflusses pro Merkmal nach (4) unter Verwendung der zuvor berechneten Shapley-Werte der Testdatensätze. In Bild 6 sind diese globalen Merkmalseinflüsse für jeden erfolgreichen Klassifikator gegenübergestellt. Dieses zeigt den signifikanten Einfluss des Merkmals Stärken/Weichspülen% und unterstützt die Beobachtungen der zuvor aufgeführten lokalen Interpretationen aus Bild 5. Insbesondere der Klassifikator aus $\mathscr{D}_{F\#_{1.1}}$ zeigt einen hohen Einfluss von 0.42, während der Klassifikator aus $\mathscr{D}_{F\#_{1.3}}$ mit Oversampling auf mehrere Merkmale angewiesen ist. Die Klassifikatoren aus $\mathscr{D}_{F\#_{1.4}}$ und $\mathscr{D}_{F\#_{1.6}}$ mit Betriebszeit bis zum Fehler äußern eine Relevanz des Merkmals Waschmittelverstärker%, das bei Betrach-

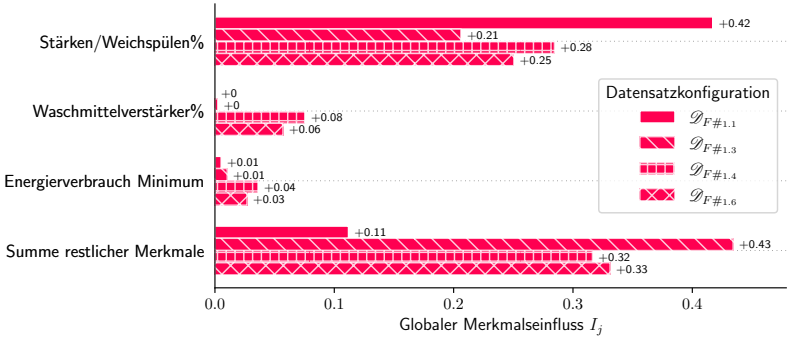

Bild 6: Balkendiagramm zur Darstellung des globalen Merkmalseinflusses mittels Shapley-Werten für die erfolgreichen Klassifikatoren der AutoML-Runs des Fehlerfalls $F\#_1$.

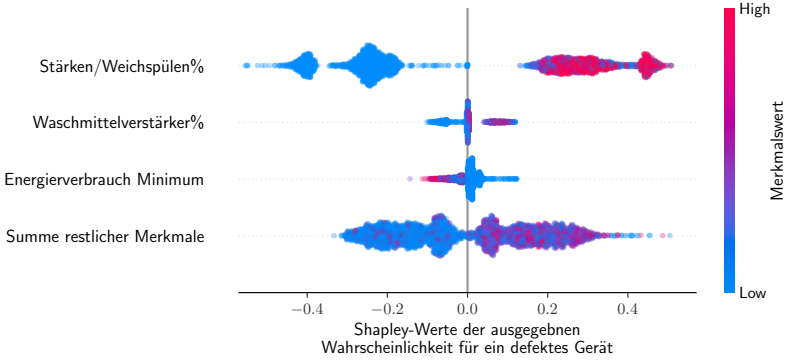

Bild 7: Übersichtsabbildung der Shapley-Werte für erfolgreiche Klassifikatoren aus $\mathscr{D}_{F\#_{1.1}}$, $\mathscr{D}_{F\#_{1.3}}$, $\mathscr{D}_{F\#_{1.4}}$ und $\mathscr{D}_{F\#_{1.6}}$ des Fehlerfalls $F\#_1$ zur Darstellung des Merkmalseinflusses gegenüber ihrer Merkmalswerte.

tung der gesamten Betriebszeit keine Bedeutung hat. Zusätzlich zum reinen Einfluss sind in Bild 7 die gesamten lokalen Interpretationen inklusive der jeweiligen Merkmalswerte in einer Übersichtsabbildung dargestellt. Zu erkennen ist hierbei, dass alle Waschmaschinen, die Stärken/Weichspülen verwendet haben (rote Punkte), positive und hohe Shapley-Werte aufweisen, während die Mehrzahl der Waschmaschinen, die kein Stärken/Weichspülen verwendet haben (blaue Punkte), negative Shapley-Werte aufweisen. Die Shapley-Werte der restlichen Merkmale äußern ein deutlich geringeren Einfluss. So tendieren

bspw. Prädiktionen für Waschmaschinen mit einem hohen Wert des Merkmals minimaler Energieverbrauch oder einem geringen Wert des Merkmals Waschmittelverstärker%, geringfügig zu einem intakten Gerät.

4.3 Ergebnisse & Diskussion

Anhand der Ausgabe des Assistenzsystems, in Form der Bilder 5, 6 und 7, lässt sich Stärken/Weichspülen% aufgrund seiner Signifikanz als alleinige hypothetische Ursache basierend auf der Gerätenutzung für den Fehlerfall $F\#_1$ identifizieren. Hypothese: *Der Fehlerfall $F\#_1$ tritt bei Waschmaschinen auf, bei denen das Programmextra Stärken/Weichspülen verwendet wurde*. Die restlichen Merkmale bieten hingegen einen zu geringen Einfluss, um weitere hypothetische Ursachen abzuleiten. Durch die Aufdeckung des unbekannten Zusammenhangs zwischen dem Stärken/Weichspülen und dem Auftreten des Fehlerfalls $F\#_1$, ist die Evaluation insgesamt positiv zu bewerten. Hinzu kommt die erfolgreiche Bestimmung der Relevanz von $F\#_1$ und $F\#_2$, sowie die Feststellung einer unzureichenden Grundlage des $F\#_2$ für weitere Interpretationen.

Trotz einer erfolgreichen Evaluation des Konzepts ist nicht sichergestellt, dass die identifizierten hypothetischen Ursachen einen kausalen Zusammenhang zum untersuchten Fehlerfall aufweisen. Eine eindeutige Fehlschlussfolgerung wäre z.B., dass der Energieverbrauch der Geräte erhöht werden muss, um die Wahrscheinlichkeit für den Fehler $F\#_1$ zu verringern. Demnach bedarf es weiterhin Domänenexpert:Innen, welche die vom Assistenzsystem aufgezeigten Korrelationen kritisch überprüfen und in kausale Zusammenhänge überführen müssen. Anhand von kausal widerlegbaren Hypothesen können Verbesserungen an der Datenbasis von IoT-Gerätenutzungsdaten, anstatt an den IoT-Geräten selbst, vorgenommen werden. Bspw. ließen sich durch widerlegbare Hypothesen unbekannte Inkonsistenzen oder bislang unbeobachtete *Confounder* in den Daten identifizieren. Eine weitere Beschränkung ergibt sich aus der Begrenzung des eingehenden Datensatzes auf die Nutzung der Geräte, während ein ganzheitlicher Datensatz über die Lebenszeit (Produktion + Nutzung) umfassendere Zusammenhänge sowie Schlussfolgerungen zulässt. Des Weiteren äußerte sich die Durchführung mehrerer AutoML-Runs je Fehlerfall für die binären Klassifikationen zwar

als präzise, was jedoch zu Lasten der Skalierbarkeit des Konzepts auf die Anwendung mehrere Fehlerfälle geht.

5 Zusammenfassung & Ausblick

In dieser Arbeit wurde das Konzept eines KI-basierten Assistenzsystems zur QS von IoT-Geräten vorgestellt, das relevante Fehlerfälle identifiziert, Zusammenhänge zwischen Gerätenutzung und Fehlerfällen mittels AutoML approximiert, und diese Zusammenhänge den Domänenexpert:Innen mittels SHAP zugänglich macht. Als Ergebnis der Evaluation basierend auf realen IoT-Gerätenutzungsdaten vernetzter Waschmaschinen konnte die Verwendung eines Programmextras als hypothetische Ursache für einen Fehlerfall identifiziert werden, so dass sich eine insgesamt positive Evaluation ergab. Im Anschluss an die Evaluation erfolgte abschließend eine Diskussion über das erarbeitete Konzept, welches insbesondere die Diskrepanz zwischen Korrelationen und tatsächlichen Kausalitäten sowie die Grenzen des Konzepts thematisiert.

Zukünftig gilt es, die Potenziale und Grenzen des Konzepts mithilfe der Domänenexpert:Innen des QM weiter auszuarbeiten, die identifizierten Hypothesen auf Kausalitäten zu überprüfen und in Verbesserungsmaßnahmen zu überführen. Zur Unterstützung bei der Schätzung von Kausalitäten bietet sich zudem die Verwendung von *Double/Debiased Machine Learning* an, wodurch *Confounder* in den Daten sichtbar werden. Nach dem ersten Erfolg in der Evaluation bietet sich die Betrachtung eines Mehrklassen-Problems zur verbesserten Skalierbarkeit und die Einbeziehung von Produktionsdaten an.

Literatur

[1] J. Chatterjee, N. Dethlefs. „Temporal Causal Inference in Wind Turbine SCADA Data Using Deep Learning for Explainable AI“. In: *Jour. of Phys.: Conf. Ser. 1618*. 2020.

[2] B. Steurtewagen, D. Van den Poel. „Adding interpretability to predictive maintenance by machine learning on sensor data“. In: *Comp. & Chem. Eng. 152*, 107381. 2021.

[3] R. Chen, F. Jankovic, N. Marinsek, et al. „Developing Measures of Cognitive Impairment in the Real World from Consumer-Grade Multimodal Sensor Streams“. In: *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, S. 2145-2155. 2019.

[4] S.M. Lundberg, B. Nair, M.S. Vavilala, et al. „Explainable machine-learning predictions for the prevention of hypoxaemia during surgery“. In: *Nat Biomed Eng 2*, S. 749-760. 2018.

[5] S.M. Lauritsen, M. Kristensen, M.V.Olsen, et al. „Explainable artificial intelligence model to predict acute critical illness from electronic health records“. In: *Nat Commun 11*, 3852. 2020.

[6] C. Thornton, et al. „Auto-WEKA: Combined Selection and Hyperparameter Optimization of Classification Algorithms“. In: *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, S. 847–855. 2013.

[7] F. Hutter, H.H. Hoos, K. Leyton-Brown. „Sequential modelbased optimization for general algorithm configuration“. In: *International conference on learning and intelligent optimization*, S. 507–523. 2011.

[8] L. Li, K. Jamieson, G. DeSalvo, et al. „Hyperband: A novel bandit-based approach to hyperparameter optimization“. In: *The Journal of Machine Learning Research 18.1*, S. 6765-6816. 2017.

[9] S. Falkner, A. Klein, F. Hutter. „BOHB: Robust and efficient hyperparameter optimization at scale“. In: *International Conference on Machine Learning*, S. 1437-1446. 2018.

[10] F. Mohr, M. Wever, E. Hüllermeier. „ML-Plan: Automated machine learning via hierarchical planning“. In: *Machine Learning 107*, S. 1495–1515. 2018.

[11] B. Zoph, Q.V. Le. „Neural architecture search with reinforcement learning“. In: *ICLR 2017*.

[12] C. Molnar. „Interpretable machine learning. A Guide for Making Black Box Models Explainable“. 2019.

[13] E. Štrumbelj, I. Kononenko. „Explaining prediction models and individual predictions with feature contributions“. In: *Knowl. and info. sys. 41.3*, S. 647-665. 2014.

[14] S.M. Lundberg, L. Su-In. „A Unified Approach to Interpreting Model Predictions“. In: *Adv. Neural Inf. Process Syst. 30*, S. 4765-4774. 2017.

[15] M.T. Ribeiro, S. Singh, C. Guestrin. „Why should I trust you?: Explaining the predictions of any classifier“. In: *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*. 2016.

[16] S.M. Lundberg, E.G. Gabriel, L. Su-In, et al. „From local explanations to global understanding with explainable AI for trees“. In *Nature Machine Intelligence volume 2*, S. 56-67. 2020.

[17] D. Janzing, et al. „Feature relevance quantification in explainable AI: A causal problem“. In: *Int. Conf. on AI. and Stat.*, S. 2907-2916. 2020

[18] Birolini, Alessandro. „Reliability engineering“ Springer Berlin. 2017.

[19] Databricks AutoML. URL: https://databricks.com/product/automl. [Zugriff am: 01.09.2021].

[20] L. Breiman. „Random Forests“. In: *Machine Learning 45*, S. 5–32. 2001.

[21] T. Chen, C. Guestrin. „"Xgboost: A scalable tree boosting system“. In: *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*. 2016.

[22] G. Ke, Q. Meng, et al. „LightGBM: A Highly Efficient Gradient Boosting Decision Tree“. In: *Adv. Neural Inf. Process Syst. 30*, S 3149-3157. 2017.

[23] M. Wever, A. Tornede, F. Mohr, E. Hüllermeier. „AutoML for Multi-Label Classification: Overview and Empirical Evaluation“. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence*. 2021.

[24] T. Miller. „Explanation in artificial intelligence: Insights from the social sciences“. In: *Artificial Intelligence 267*, S. 1-38. 2019.

[25] N.V. Chawla, K.W. Bowyer, L. O.Hall, W.P. Kegelmeyer. „SMOTE: synthetic minority over-sampling technique“. In: *Journal of artificial intelligence research*, S. 321-357. 2002.

[26] M. Christ, A.W. Kempa-Liehr, M. Feindt. „Distributed and parallel time series feature extraction for industrial big data applications“. *arXiv preprint arXiv:1610.07717*. 2016.

[27] D. Chicco, G. Jurman. „The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation“. In: *BMC genomics 21.1*, S. 1-13. 2020.