T. Grust et al. (Hrsg.): Datenbanksysteme für Business, Technologie und Web (BTW 2019),
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 399

Konzept und Implementierung eines echtzeitfähigen Model Management Systems

am Beispiel zur Überwachung von Lastprognosen für den Intraday Stromhandel

Yvonne Hegenbarth[1], Gerald H. Ristow[2]

Abstract: Zur Gewährleistung der Stromnetzstabilität in Deutschland müssen Verteilernetzbetreiber darauf achten, dass zu jedem Zeitpunkt Energie-Erzeugung und -Verbrauch in ihrem Zuständigkeitsbereich in Einklang stehen. Dafür werden Vorhersagemodelle benötigt, um den zu erwartenden Überschuß oder zusätzlichen Bedarf an Energie für den Folgetag der Strombörse für den sogenannten *Day-Ahead* Handel zu melden. Neben dem Stromhandel für den Folgetag können Marktteilnehmer beim kontinuierlichen *Intraday* Strommengen bis zu fünf Minuten vor der tatsächlichen Auslieferung kaufen oder verkaufen. Bei Fehlprognosen und demnach Fehleinkäufen könnte mit einer Früherkennung und Modellanpassung diese im Intraday ausgeglichen werden. Dazu wird in dieser Arbeit ein System beschrieben, das automatisiert Fehlprognosen frühzeitig erkennt und eine Modelländerung durchführt. Das Modell wird dabei an den aktuellen Sachverhalt der Verbrauchszeitreihe angepasst. Durch diese Modellanpassung wird die Vorhersage verbessert, sodass der Intraday Handel besser betrieben werden kann und Fehleinkäufe ausgeglichen werden.

Keywords: data stream • model management • data mining • time series • forecast • concept drift • concept evolution

1 Motivation

Damit die Sicherheit des Stromnetzes in Deutschland gewährleistet wird, muss jeder Stromproduzent und Stromabnehmer seine Strommenge prognostizieren, die er ins Stromnetz einspeist oder entnimmt. Zur Prognose werden sogenannte *vorhersagende Modelle* entworfen.

In der Regel wird bei dem Prozess zur Berechnung von vorhersagenden Modellen mehrere zehn bis hundert Modelle erstellt, bis eines den Anforderungen (z.B. Vorhersagegenauigkeit) der Data Scientists genügt und eingesetzt wird. Zur Verwaltung und zum Reproduzieren von vorhersagenden Modellen muss ein *Model Management System* aufgesetzt werden. Die erzeugten Modelle werden persistiert und stehen für die Wiederverwendung zur Verfügung [Va16]. Durch sogenannte „Human-in-the-Loop Workflows“ werden die Benutzer mit Hilfe von Visualisierungstools in den Prozess zur Erstellung von vorhersagenden Modellen

[1] Software AG, Research, Uhlandstraße 12, 64297 Darmstadt, Yvonne.Hegenbarth@softwareag.com
[2] Software AG, Research, Uhlandstraße 12, 64297 Darmstadt, Gerald.Ristow@softwareag.com

doi:10.18420/btw2019-24

integriert [LWG14]. Dadurch wird ermöglicht, dass der Mensch als Überwacher agiert und aktiv die Parameter des Modells anpasst, falls die Vorhersagen zu stark von der Realität abweichen [Se16].

Im Hinblick auf den Wandel des Day-Ahead Energiemarktes und dem kontinuierlichen Intraday Handel, werden schnelle, detaillierte und flexible Planungsmöglichkeiten gefordert [Da15]. Eine Modellanpassung der Stromverbrauchsprognose des Folgetages ist bei einer zu hohen Abweichung sinnvoll. Basierend auf den Berechnungen des neuen Modells kann entsprechend im Intraday überflüssiger Strom verkauft oder auf Grund mangelnder Reserven Strom gekauft werden.

Im weiteren Verlauf wird ein *echtzeitfähiges Model Management System* (eMMS) vorgestellt, dass nicht den Persistenz-Ansatz von Liu u. a. verfolgt und den Menschen als Überwacher entlastet. Hier soll von der Hypothese ausgegangen werden, dass mit Hilfe von Data Mining Techniken und einer anwendungsspezifischen *Concept Drift* Änderungserkennung zur Laufzeit automatisiert eine Modellanpassung erfolgt.

Der Bericht fokussiert sich nicht auf die Erstellung eines vorhersagenden Modells mit minimalem Vorhersagefehler. In diesem Bericht soll vielmehr aufgezeigt werden, wie ein auf *Complex Event Processing* (CEP) basiertes Model Management aufgesetzt und betrieben werden kann.

2 Stand der Technik

Die Erstellung eines Vorhersagemodells ist ad hoc, das heißt für einen bestimmten Augenblick gemacht. Daher besteht das Bedürfnis bei Data Scientists die im Laufe der Zeit erstellten Modelle zu persistieren. In [Va16] sind diverse Problemen beschrieben, die beim iterativen Modellierungsprozess auftreten. Zusammenfassend beschreibt der Autor, dass unter anderem das Reproduzieren von Modellen und Ergebnissen übermäßig zeitaufwendig oder auf Grund mangelnder Dokumentation nicht durchführbar ist. Die Data Scientists müssen sich an Ergebnisse und Parameter früherer Versionen eines Modells „erinnern“ und haben keine Möglichkeit verschiedene Ausführungen zu kennzeichnen. Die genannte Herausforderung hebt ein wichtiges Problem für Machine Learning Tools hervor: das *Model Management*. Model Management bezeichnet die Angelegenheit des Verfolgens, Speicherns und Indexierens einer großen Anzahl von vorhersagenden Modellen, die anschließend reproduziert, geteilt, abgefragt und analysiert werden können.

Viele Model Management Systeme (MMS) sind Lösungen basierend auf einem Datenbankmanagementsystem (DBMS), das eine strukturierte Umgebung zum Speichern, Manipulieren und Abrufen von vorhersagenden Modellen bereitstellt [Be03; DK84; Va16].

3 Vorgehensweise

Zur Erläuterung der Vorgehensweise zeigt Abb. 1 den Datenfluss des von uns vorgeschlagenen echtzeitfähigen Model Management Systems (eMMS). Das Konzept baut auf einer *Offline-* und *Online-Analyse* auf und wird in Abschnitt 5 detailliert vorgestellt.

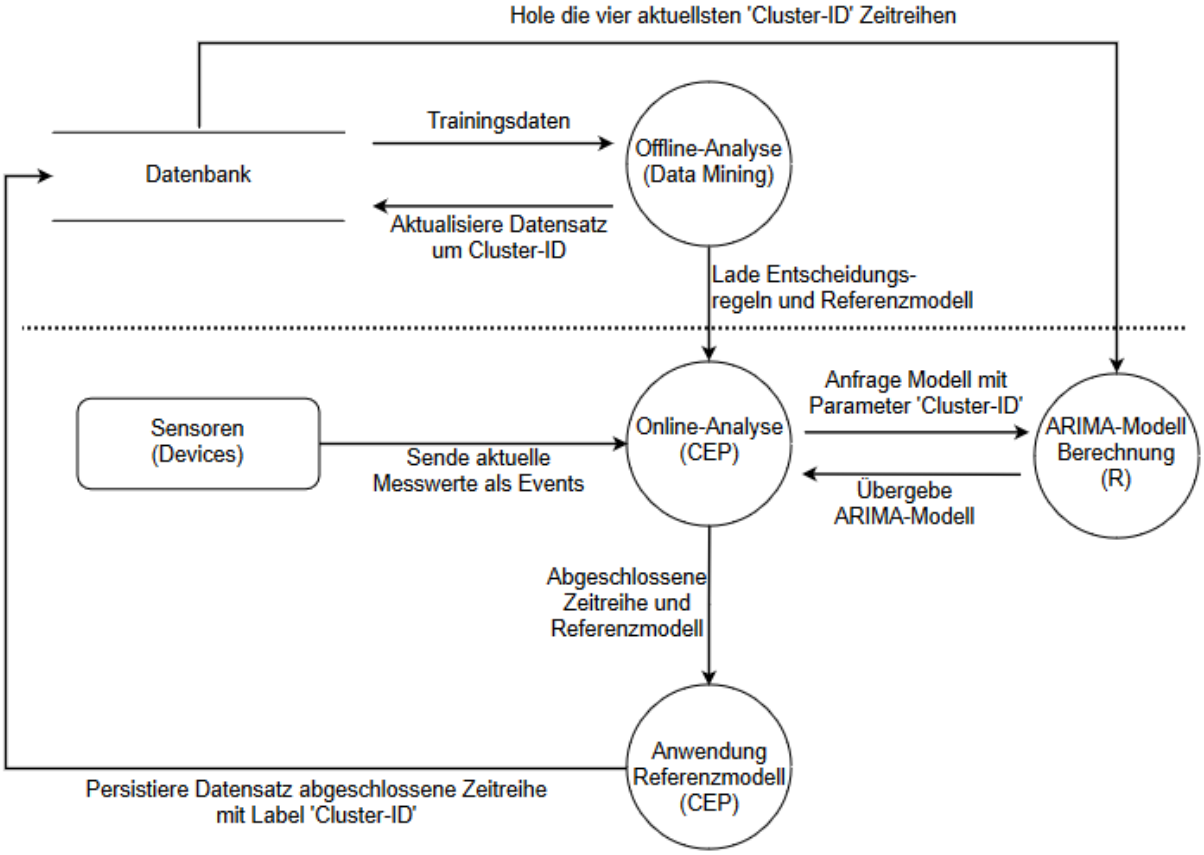

Abb. 1: Architektur eines echtzeitfähigen Model Management Systems

Das in diesem Bericht beschriebene eMMS nutzt ein DBMS lediglich zur Verwaltung von Messdaten in Form von Zeitreihen. Eine *Complex Event Processing* (CEP)-Engine ist für die Berechnung und Wartung von vorhersagenden Modellen zur Laufzeit verantwortlich (Abb. 1, Online-Analyse). CEP ist eine Technologie, die basierend auf Anfrageformulierungen Muster im Datenstrom in Form von Ereignissen (*Events*) erkennt [BK09]. Durch Aggregation und Zusammensetzung von auftretenden Ereignissen können neue komplexe Ereignisse (*Complex Events*) generiert und für weitere Analysen genutzt werden. Unter anderem senden verschiedene Sensoren (z.B. Smart Meters) ihre Messwerte in Form von Events an die CEP-Engine.

Als statistisches Modell zur Analyse und Prognose von Zeitreihen, wird ein *Auto Regressive Integrated Moving Average* (ARIMA)-Modell erstellt [BJ70]. Zu der Berechnung eines ARIMA-Modells wird die in [Da15] definierte *heuristische* Strategie verwendet, um automatisch eine Reihe vielversprechender Vorhersagemodelle für die vorliegenden Daten zu identifizieren und um den Menschen als Akteur zu entlasten. In [HK08] wird ein in *R*

implementierter, heuristischer Algorithmus zur ARIMA-Modellauswahl vorgestellt und in dem Bericht eingesetzt. Informationskriterien wie *Akaike Information Criterion* (AIC) [Ak74] und *Bayesian Information Criterion* (BIC) [Sc78] werten mehrere Modelle aus, indem schrittweise die Modellparameter erhöht werden.

Zur Wartung der Modelle in CEP wird eine Kombination der Strategien *periodisch* und *schwellwertbasiert* aus [Da15] angewendet, die bereits erfolgreich untersucht wurde [Da11]. Zum einen findet periodisch (z.B. täglich) eine Modellberechnung statt. Weiterhin wird ebenfalls erkannt, wenn eine Vorhersage sich von den tatsächlichen Messwerten entfernt. Dafür wird in [Tr13] zum einen eine *fensterbasierte* (manuell) und eine *clusterbasierte* (automatisiert) Änderungserkennung vorgestellt. Die fensterbasierte Methode basiert auf der Untersuchung eines gleitenden Fensters (*Sliding Window*), wodurch der Datenstrom segmentweise untersucht wird. Die clusterbasierte Methode hingegen nutzt ein Clustering-Modell und versucht jedes neu eingetroffene Ereigniss einem Cluster zuzuweisen. Unter Berücksichtigung des berechneten Vorhersagefehlers in CEP und der in dem Bericht eingesetzen fensterbasierten Änderungserkennung, wird segmentweise überprüft, ob ein vordefinierter Schwellwert überschritten wurde und ob ein neues Modell berechnet werden muss. Ein Ausblick zur Anwendung einer clusterbasierten Änderungserkennung wird in Abschnitt 7 vorgestellt.

Unterstützend zur Berechnung neuer vorhersagender Modelle werden vorab die historischen Daten mit Hilfe von Data Mining Techniken untersucht (Abb. 1, Offline-Analyse). Als Ergebnis wird ein Clustering-Modell generiert, das im Folgenden als *Referenzmodell* bezeichnet wird. Ein Referenzmodell ist die Zusammenfassung einer Menge von *formähnlichen Zeitreihen*. Das Referenzmodell stellt die repräsentativsten Zeitverläufe (*Referenzzeitreihe*) einer endlichen Menge von Zeitreihendaten dar. Während das Referenzmodell basierend auf den Messwerten erstellt wird, soll zusätzlich ein Klassifikator die Referenzzeitreihen mit Hilfe von z.B. kalendarischen Merkmalen beschreiben. Als Klassifikator wird ein Entscheidungsbaum gewählt, zur Erstellung von bedingten Anweisungen (*Entscheidungsregeln*). Für das eMMS soll zu Beginn eine Entscheidungsregel dabei helfen, die Referenzzeitreihe (basierenden auf der zuvor berechneten Clusteranalyse) für den folgenden Tag zu prognostizieren. Basierend auf der Entscheidungsregel werden die entsprechenden formähnlichen Trainingsdaten zur Zeitreihenvorhersage ausgewählt und ein initiales ARIMA-Modell erstellt.

Sobald eine Zeitreihe abschließt, wird das Referenzmodell auf dieser angewendet und mit einem entsprechenden Cluster-Label in der Datenbank persistiert. Dadurch wird sichergestellt, dass stets aktuelle Zeitreihen zur Berechnung von ARIMA-Modellen herangezogen werden. Außerdem können die historischen Daten für nachfolgende Datenanalyseprozesse verwendet werden.

4 Auswertung der historischen Daten mit Data Mining Techniken

Zur Auswertung des entwickelten eMMS wurden drei anonymisierte Stromverbrauchsdaten von der EWE AG[3] zur Verfügung gestellt. Tabelle Tab. 1 beschreibt die Menge an aufgezeichneten Messwerten pro Datensatz und die Partitionierung der Daten in *historische* (Analyse mithilfe von Data Mining Techniken) und *Simulationsdaten* (zur Simulation neuer Messwerte und Auswertung des eMMS).

Datensatz	Partitionierung	Menge an Daten
Verbraucher I	Historisch: 01.01.2014 - 31.12.2014	365 Tage
	Simulation: 01.01.2015 - 02.09.2015	245 Tage
Verbraucher II	Historisch: 01.05.2015 - 30.04.2016	365 Tage
	Simulation: 01.05.2016 - 31.12.2016	245 Tage
Verbraucher III	Historisch: 01.01.2015 - 31.12.2015	365 Tage
	Simulation: 01.01.2016 - 01.09.2016	245 Tage

Tab. 1: Übersicht der zur Verfügung stehenden Datensätze

Für die bevorstehende Datenanalyse wird für jeden Verbraucher ein historischer Datensatz im Zeitraum von einem Jahr gewählt (365 Tage). Zur Simulation und Auswertung des entwickelten eMMS steht ein zeitlich aktuellerer Datensatz von 245 Tagen zur Verfügung. Die Messwerte wurden in einem Intervall von fünfzehn Minuten aufgezeichnet. Daraus folgt, dass pro Tag 96 Messwerte für jeden Verbraucher aufgezeichnet wurden.

Im Folgenden werden die Analyseschritte zur Erstellung von *Referenzmodellen* und *Entscheidungsregeln* beschrieben. Die Ergebnisse der Datenanalyse werden im Anschluss in das eMMS übertragen.

4.1 Clusteranalyse

Im Dezenniumbericht [ASW15] beschreiben die Autoren eine spezielle Art von Clustering – das *Zeitreihen-Clustering*. Das Clustering von Zeitreihen wird hauptsächlich zur Entdeckung interessanter Muster in den Zeitreihendatensätzen verwendet. Zum einen dient das Zeitreihen-Clustering zum Auffinden von Mustern, die häufig im Datensatz vorkommen. Zum anderen können ebenfalls Muster erkannt werden, die in den Datensätzen „überraschend“ auftreten. Das Auffinden von Mustern, die „ungewöhnlich“ sind und/oder „überraschend“ auftreten, wird als *Anomalie-Erkennung* bezeichnet.

[3] Die EWE AG (ehemals Energieversorgung Weser-Ems) ist ein Versorgungsunternehmen im Bereich Strom, Erdgas, Telekommunikation, Informationstechnologie und Umwelt. Die Software AG und EWE AG sind Forschungspartner des Projektes *enera*, zur Erstellung einer neuen Modellregion für die Energiewende. Siehe `http://energie-vernetzen.de/`

Das Zeitreihen-Clustering dient zur Erstellung von sogenannten Referenzmodellen, die für das eMMS benötigt werden.

Definition 4.1 (Referenzmodell)
Ein Referenzmodell ist die Zusammenfassung einer Menge von formähnlichen Zeitreihen (als Resultat eines Clusterings). Das Referenzmodell stellt die repräsentativsten Zeitverläufe einer endlichen Menge von Zeitreihendaten dar. Jede der Zeitreihen aus der Grundgesamtheit der Daten wird einem der Cluster des Referenzmodells zugeordnet.

Definition 4.2 (Referenzzeitreihe)
Eine Referenzzeitreihe repräsentiert den Zeitverlauf einer Menge von formähnlichen Zeitreihen. Eine Referenzzeitreihe beschreibt das Zentrum eines Clusters, resultierend aus einem Clustering.

Zum Clustern von Zeitreihen und der Generierung von Referenzmodellen wird ein Trainingsdatensatz von einem Jahr gewählt (Tab. 1, historischer Datensatz). Zum Clustern der Messwerte (in Form von Zeitreihen) können sowohl partitionierende (zum Beispiel *k-Means*) als auch hierarchische Verfahren angewendet werden [ASW15, S. 26]. Für die vorliegende Arbeit wird der k-Means Algorithmus angewendet, der bereits in anderen Publikationen großen Zuspruch für Zeitreihen-Clustering genießt [FRB98; Ma67] und sehr leistungsstark ist [BFR98].

Aus dem k-Means Clustering resultiert ein Referenzmodell mit k Referenzzeitreihen. Um die geeignete Anzahl von k Clustern im Datensatz zu ermitteln, wird zur Orientierung die *Ellbogen-Methode* angewendet [HKP11, S. 486].

Nach der Untersuchung einer geeigneten Clustergröße k zeigt Abb. 2 folgende Referenzmodelle für Verbraucher I (links), II (mitte) und III (rechts).

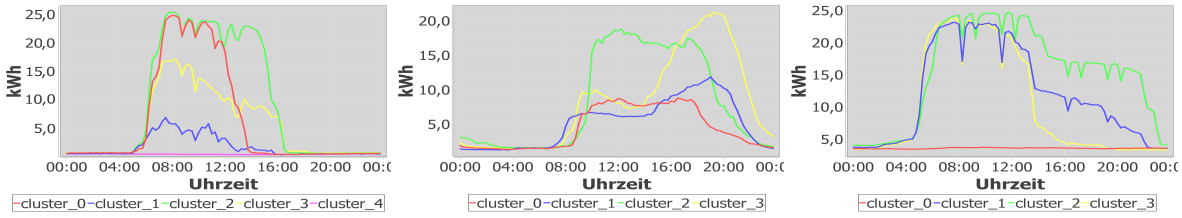

Abb. 2: Referenzmodelle basierend auf historischen Daten

Aus dem Referenzmodell von Verbraucher I und III ist anhand der durchgängigen Linie nahe Null zu erkennen, dass an manchen Tagen kein Strom bezogen wird. Lediglich Verbraucher II bezieht regelmäßig Strom.

Abb. 3 zeigt ebenfalls die Mengenverteilung der jeweiligen Cluster als Kuchendiagramm. Anhand der Clustergröße lässt sich herleiten, dass die größeren Cluster vermutlich einen normalen Zustand beschreiben und die kleineren Cluster auf eine mögliche Anomalie hinweisen. Demnach können für Verbraucher I die Referenzzeitreihen *cluster_1* und *cluster_3*

auf eine Anomalie des Verbrauchverhaltens hinweisen, bei Verbraucher II eventuell *cluster_2* und bei Verbraucher III schließlich die Referenzzeitreihe *cluster_3*.

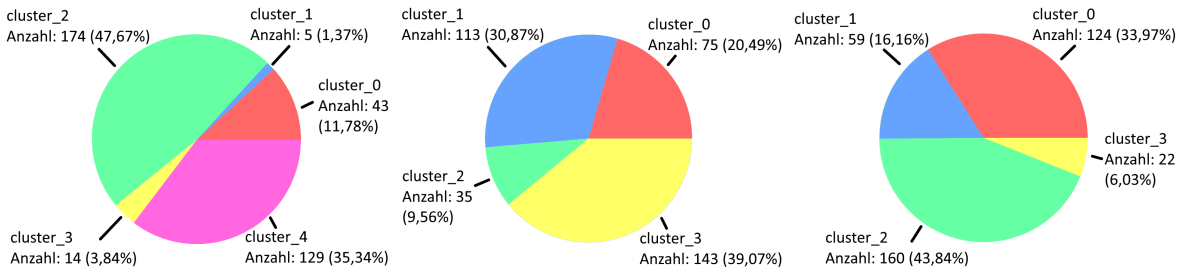

Abb. 3: Kuchendiagramme zur Darstellung der Clustergröße im Referenzmodell

Weiterhin wird mithilfe eines Klassifikators untersucht, welche Merkmale zur Beschreibung und Vorhersage einer Referenzzeitreihe geeignet sind. Die zuvor aufgestellte Hypothese der Anomalie-Cluster wird hierbei unter anderem bestätigt.

4.2 Klassifikation

Für das eMMS soll zu Beginn eine Entscheidungsregel (Klassifikator) dabei helfen, die Referenzzeitreihe (basierenden auf der zuvor berechneten Clusteranalyse) für den folgenden Tag zu prognostizieren. Basierend auf der Entscheidungsregel werden die entsprechenden formähnlichen Trainingsdaten zur Zeitreihenvorhersage ausgewählt.

Zur Erstellung eines Klassifikators werden zunächst Merkmale (*Features*) generiert. Im vorliegenden Bericht werden folgende kalendarische Merkmale herangezogen:

`Monat, Quartal, Jahreszeit, Wochentag, Arbeitstag und Feiertag.`

Mit einer Filterauswahlmethode (z.B. *Forward Selection* oder *Backward Selection*) werden für jeden Verbraucher die bedeutsamsten Merkmale zur Beschreibung und Vorhersage der Referenzzeitreihe ausgewählt [JBB15].

Schließlich wird ein Entscheidungsbaum-Algorithmus, wie z.B. *C4.5* [Qu93] oder *SPRINT* [SAM96], zur Generierung von Entscheidungsregeln eingesetzt. Zur Erstellung und Auswertung der Regeln wird mit der sogenannten *Holdout-Methode* der historische Datensatz in Trainings- und Testdaten partitioniert (75:25) [HKP11, S. 370].

Bei der Klassifikation von Verbraucher I und III konnten für manche Referenzzeitreihen keine Regeln definiert werden und sind in Tab. 2 als „/“ gekennzeichnet. Die Referenzzeitreihen ohne Entscheidungsregeln wurden in Abb. 3 unter Berücksichtigung ihrer Mengenverteilung im Referenzmodell bereits zuvor als mögliche Anomalien in der Zeitreihe erkannnt. Zellen mit „-“ wiederum kennzeichnen, dass die entsprechende Cluster-ID für den Datensatz nicht existiert.

Nachdem basierend auf einer Trainingsdatenmenge ein Klassifikator berechnet wurde, wird zur Auswertung der Vorhersagegenauigkeit (*Accuracy*) das Modell auf einer Testmenge ausgeführt. Im Anschluss erfolgt die Ausführung des Modells auf der Gesamtmenge (Training und Test), um die Robustheit[4] eines Modells zu bewerten. In Tab. 2 ist ebenfalls die Vorhersagegenauigkeit auf dem Testdatensatz sowie auf der Gesamtmenge aufgelistet[5].

Die Gesamtgenauigkeit in allen drei Verbrauchsdatensätze entspricht nahezu die der Testdatenmenge. Daher könnnen alle drei Klassifikatoren als relativ robust angesehen werden.

Klassifikatoren – Entscheidungsregeln			
Cluster-ID	Verbraucher I	Verbraucher II	Verbraucher III
cluster_0	Freitag	(Q2 OR Q3) AND NOT Arbeitstag	Mittwoch OR Donnerstag
cluster_1	/	(Q2 OR Q3) AND Arbeitstag	Dienstag
cluster_2	Montag OR Dienstag OR Mittwoch OR Donnerstag	(Q1 OR Q4) AND NOT Arbeitstag	Freitag OR Samstag OR Sonntag OR Montag
cluster_3	/	(Q1 OR Q4) AND Arbeitstag	/
cluster_4	Samstag OR Sonntag	–	–
Accuracy Testdaten	89,13%	88,04%	77,17%
Accuracy Gesamt	84,93%	86,3%	78,63%

Tab. 2: Entscheidungsregeln basierend auf der Ausgabe der Clusteranalyse und ihre Genauigkeit

5 Echtzeitfähiges Model Management System

Das eMMS ist in zwei voneinander losgelöste Prozesse unterteilt: *Offline-* und *Online-Analyse*. Die Offline-Analyse beschreibt den bereits in Abschnitt 4 vorgestellten Prozess zur Datenanalyse historischer Daten mit Data Mining Techniken. Die Online-Analyse wiederum umfasst sowohl die Berechnung von Modellen zur Zeitreihenvorhersage als auch deren Auswertung und Austausch in Echtzeit.

Im Folgenden wird zunächst die Architektur des eMMS vorgestellt und im Anschluss der Prozess zur Online-Analyse erläutert.

[4] Die Robustheit eines Modells beschreibt, dass auch bei verrauschten Daten oder Daten mit fehlenden Werten korrekte Vorhersagen getroffen werden [HKP11, S. 369].

[5] Bezeichnung 'Q' in Tab. 2 steht für 'Quartal'

5.1 Architektur

Die einzelnen Komponenten der Architektur werden in *Online* und *Offline* unterteilt und sind in Abb. 4 dargestellt.

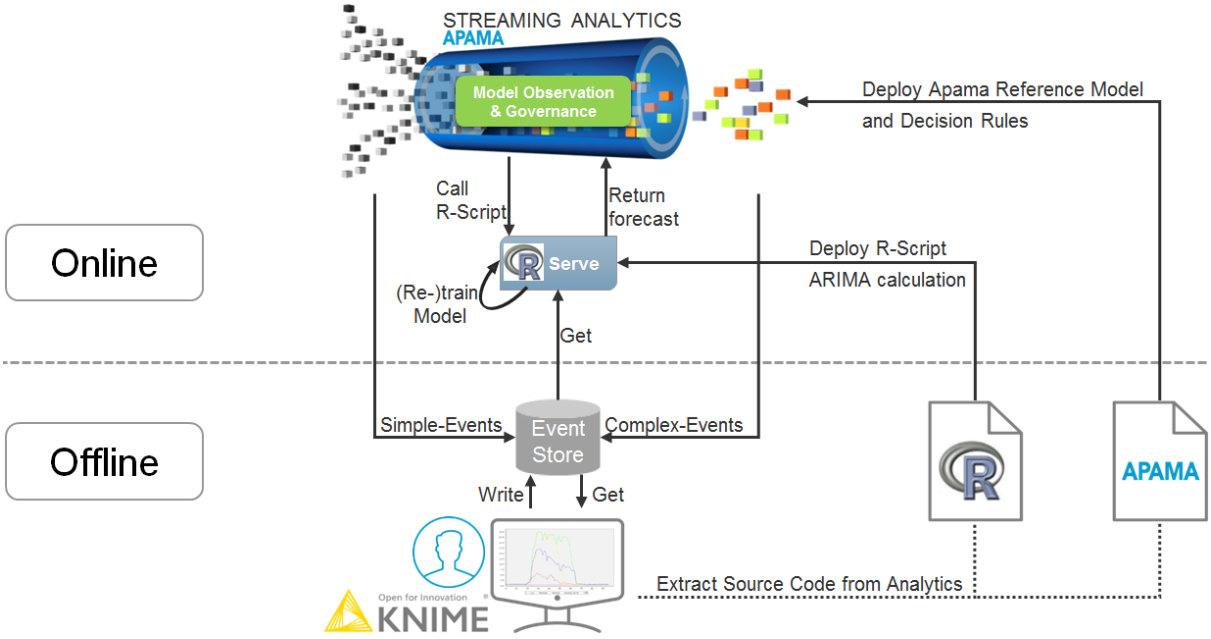

Abb. 4: Architektur eines echtzeitfähigen Model Management Systems

Zu den Online-Komponenten gehört eine CEP-Engine (hier *Apama*[6]) und *Rserve*[7]. Rserve ist ein TCP/IP-Server, mit dessen Hilfe andere Programme (hier CEP-Engine) statistische Berechnungen in der Programmiersprache *R* ausführen können [Ur03].

Die CEP-Engine hat die Aufgabe, die Flut an Datenströmen in Echtzeit zu überwachen und Berechnungen auf Basis von Events durchzuführen. Als wesentlichen Bestandteil des eMMS ist die CEP-Engine verantwortlich zur Auswertung von Zeitreihenprognosen und involviert im Prozess zur Berechnung neuer ARIMA-Modelle. Die Berechnungen eines ARIMA-Modelles laufen auf dem *Rserve*. Die Aufrufe des R-Skripts werden von der CEP-Engine via TCP realisiert. Das hinterlegte R-Skript umfasst die zuvor vorgestellte heuristische Methode nach [HK08]. Zur Modellgenerierung wird von der CEP die erwartete Referenzzeitreihe (Cluster-ID) übergeben. Basierend auf dieser Information werden zur Modellberechnung nur Trainingsdaten hinzugezogen, die der gleichen Referenzzeitreihe zugehören. Das Modell mit dem besten Informationskriterium wird daraufhin automatisiert ausgewählt und der CEP-Engine in Form einer prognostizierten Zeitreihe übergeben.

Das DBMS (hier *Event Store*) und eine Analytik-Plattform (hier *KNIME*[8]) gehören zu

[6] Informationen und eine kostenfreie Version zum Download zur Streaming Analytic Platform Apama der Software AG, siehe `http://www.apamacommunity.com/`.

[7] Information und Download, siehe `https://www.rforge.net/Rserve/`.

[8] Konstanz Information Miner (KNIME) ist eine freie Software für die interaktive Datenanalyse. Weitere Informationen und Download, siehe `http://knime.org/`.

den Offline-Komponenten. Der Event Store beinhaltet sowohl die historischen Messwerte, die analysiert werden sollen, als auch Messwerte für die Simulation und Auswertung des eMMS. Sowohl der Rserve als auch die Analytik-Plattform beziehen ihre Daten aus dem Event Store. Die historisch hinterlegten Zeitreihen werden visuell aufbereitet und mit Hilfe von statistischen Verfahren aus Abschnitt 4 analysiert. Die Erkenntnisse aus der Datenanalyse werden zur Erstellung der R-Skripte und CEP-Anfragen übertragen und zur Laufzeit ausgeführt. Weiterhin können von der CEP-Engine aggregierte, verarbeitete Ereignisse (z.B. die Beschriftung einer neuen Zeitreihe) im Event-Store gespeichert werden.

5.2 Online-Analyse

Die Online-Analyse beschreibt den Wartungsprozess des ausführenden vorhersagenden Modells. Zur Wartung wird eine Kombination der Strategien *periodisch* und *schwellwertbasiert* angewendet (vgl. Abschnitt 3). Neben der täglichen Modellberechnung wird ebenfalls unter Berücksichtigung des Vorhersagefehler ein neues Modell angefordert und gegebenenfalls neu berechnet. Der Vorhersagefehler (engl.: *Forecast Error* (FE)) e_i beschreibt allgemein die Differenz zwischen dem beobachteten und vorhergesagten Messwert. Entspricht y_i dem beobachteten und $\hat{y}_i$ dem vorhergesagten Wert, dann gilt:

$$e_i = y_i - \hat{y}_i \tag{1}$$

In der Echtzeit-Modellvalidierung wird der FE von der CEP-Engine stets beobachtet. Um eine Modellabweichung zu detektieren, wird eine Concept Drift Anfrage formuliert.

Definition 5.1 (Concept Drift) *Im Bereich des maschinellen Lernens bezeichnet der Ausdruck Concept Drift das Phänomen, dass die Trainingsdaten nicht mit dem aktuellen Anwendungsfall übereinstimmen [Ts04; Zl10]. Das zugrundeliegende Konzept zur Vorhersage der Daten weicht von der Realität ab, wodurch falsche Vorhersagen von ausführenden Modellen getroffen werden. Daher muss das vorhersagende Modell regelmäßig an das neueste Konzept angepasst werden [Ha16].*

Durch die Berechnung des Vorhersagefehlers FE wird von der CEP-Engine ein neuer Datenstrom erzeugt. Dieser wird nach einer fensterbasierten Änderungserkennung untersucht (vgl. Abschnitt 3). Ein gleitendes Fenster mit einer vordefinierten Fenstergröße und einem Grenzwert für FE beschreiben eine Concept Drift Erkennung, wann die Vorhersage nicht mehr den Anforderungen genügen und eine Modelländerung womöglich erforderlich ist.

Nachdem ein Concept Drift erkannt wurde, wird daraufhin validiert, ob eine Modelländerung angebracht ist. Hierfür wird die Vorhersagegenauigkeit *Mean Absolute Error* (MAE) der gesamten, bisherigen Zeitreihe mit den Referenzzeitreihen des entsprechenden Referenzmodells berechnet. Der MAE beschreibt den durchschnittlichen, absoluten Fehler und wird wie folgt definiert:

$$MAE = \frac{1}{n} \sum_{i=0}^{n} |e_i| \tag{2}$$

Die Referenzzeitreihe mit dem kleinsten Fehler wird als geeignetes Modell ausgewählt. Entspricht die Beschreibung des ausführenden Modells nicht der Referenzzeitreihe mit dem kleinsten Fehler, dann wird ein neues ARIMA-Modell berechnet. Andernfalls würde ein unnötiger Rechenaufwand entstehen, der damit vermieden wird. Zur Neuberechnung eines ARIMA-Modells wird nun ein Trainingsdatensatz der vier aktuellsten Zeitreihen[9] aus der Gruppe der Referenzzeitreihe mit dem kleinsten Fehler gewählt. Das neue Modell wird in das CEP-System geladen und weiterhin nach einem möglichen Concept Drift untersucht.

Folgend veranschaulicht Abb. 5 das Konzept der Modelländerung nach einem Concept Drift Alarm und das Heranziehen einer Referenzzeitreihe zur Unterstützung der Neuberechnung eines ARIMA-Modells.

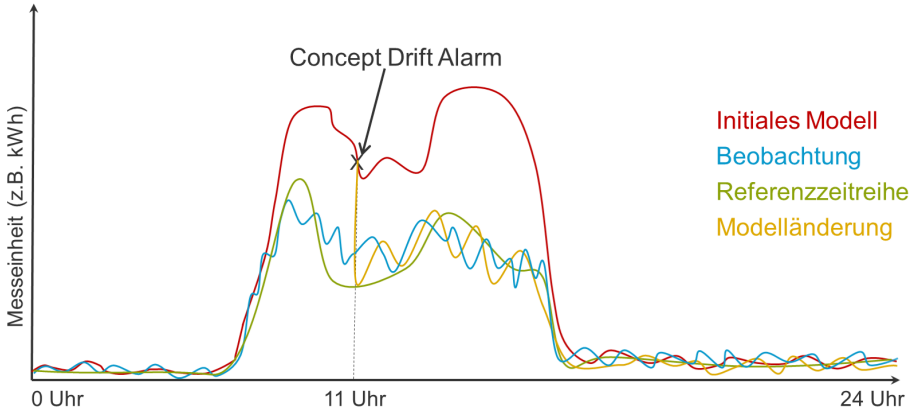

Abb. 5: Illustration einer Modelländerung nach einem Concept Drift

Die Illustration zeigt das initiale Modell mit der Zeitreihenvorhersage des gesamten Tages (rote, obere Kurve). Im Verlauf des Tages trifft der tatsächlich beobachtete Verbrauch ein (blaue, stark schwankende Kurve) und berechnet die Differenz von der Vorhersage und der Beobachtung. Der Concept Drift Alarm wird ausgelöst, wenn die Differenz zu hoch ist und eine vorgegebene Zeitspanne überdauert. Aus dem Referenzmodell wird eine Referenzzeitreihe ausgewählt, die zum Zeitpunkt des Alarms (hier 11 Uhr) den momentanen Verbrauch am besten beschreibt (grüne Kurve). Basierend auf der vorzeitigen Clustering-Anwendung um 11 Uhr, wird ein neuer Trainingsdatensatz zur Neuberechnung eines Modells herangezogen, das den zu erwartenden Verbrauch ab sofort besser beschreiben soll (orangene Kurve). Während die Illustration zeigt, dass das initiale Modell sich im Laufe des Tages immer mehr vom tatsächlichen Verbrauch entfernt (engl.: *drift*), weist das neu generierte Modell ab 11 Uhr tatsächlich eine höhere Übereinstimmung mit dem aktuellen Anwendungsfall auf.

[9] Das Auswahlkriterium „vier aktuellste Zeitreihen“ basiert auf den Erfahrungswerten von Experten der EWE AG.

Damit die neue abgeschlossene Zeitreihe ebenfalls im DBMS eine Clusterbezeichnung erhält, wird das Referenzmodell angewendet. Dafür wird die Vorhersagegenauigkeit MAE der Referenzzeitreihen auf die abgeschlossene Zeitreihe berechnet. Die Referenzzeitreihe mit dem kleinsten Fehler wird für das Clustering ausgewählt und die entsprechende Cluster-ID im DBMS eingetragen. Dadurch wird gewährleistet, dass für die nächste ARIMA-Modellberechnung stets die aktuellsten Zeitreihen als Trainingsdatensatz ausgewählt werden.

6 Auswertung

Zur Auswertung des eMMS wird im vorliegenden Bericht die Genauigkeit der Zeitreihenvorhersage des Folgetages über den gesamten Simulationsdatensatz berechnet. In [He18] sind die Ergebnisse zusätzlich gruppiert nach der Cluster-ID aufgelistet. Weiterhin wird die Vorhersage der auf historischen Daten basierenden Entscheidungsregeln auf dem Simulationsdatensatz validiert.

Das eMMS, das zur Laufzeit auf Änderungserkennungen mit einer Modelländerung (im Folgenden „*Change*-Modell“ genannt) reagieren kann, wird neben dem initialen ARIMA-Modell (im Folgenden „*Day-Ahead*-Modell“ genannt) und dem erzeugten „*Referenzmodell*“ mit zwei weiteren Modellen verglichen: „*Mean*-“ und „*Naiv*-Modell“.

Im Buch [HA18] werden diese Verfahren als Benchmark zum Vergleich von Vorhersagemodelle verwendet. Die untersuchten Modelle werden wie folgt zusammengefasst:

Referenzmodell Ergebnis des Zeitreihen-Clustering aus Abschnitt 4.1.

Day-Ahead Initiales ARIMA-Modell zur Vorhersage des Stromverbrauchs für den Folgetag, berechnet nach dem Verfahren aus [HK08].

Change Modellanpassung durch Beobachtung des Vorhersagefehlers und Concept Drift Erkennung.

Mean Die Vorhersage aller zu prognostizierenden Werte gleicht dem Durchschnitt der historischen (Trainings-) Daten. Wenn $\{y_1 + \cdots + y_T\}$ die historischen Daten bezeichnet, dann wird die Vorhersage wie folgt definiert:

$$y_{T+h} = \overline{y} = (y_1 + \cdots + y_T)/T \tag{3}$$

Die Prognose wird als y_{T+h} bezeichnet und basiert auf den Daten $\{y_1 + \cdots + y_T\}$.

Naiv Die gesamte Vorhersage gleicht der letzten Beobachtung. Demnach gilt:

$$\hat{y}_{T+h} = y_T. \tag{4}$$

Tab. 3 zeigt die Auswertung des eMMS mit Modelländerung (*Change*) im Vergleich mit dem initialen Modell (*Day-Ahead*). Die zwei einfachen Modelle (*Mean* und *Naiv*) und das Referenzmodell dienen als Vergleichsgröße zur Orientierung der Auswertung. In Abschnitt 4.2 wurden bereits die Entscheidungsregeln auf ihre Robustheit geprüft. Zur Bestätigung und finalen Untersuchung des Klassifikators wird ein dritter, eigenständiger Datensatz zur Validierung empfohlen[10] [Bi95; RH96]. Zu diesem Zweck wird die Vorhersagegenauigkeit des Klassifikators aus Abschnitt 4.2 auf dem Simulationsdatensatz validiert und ebenfalls in Tab. 3 präsentiert.

Durchschnitt MAE in kWh & Accuracy der Entscheidungsregeln						
Datensatz	Day-Ahead	Change	Referenzmodell	Mean	Naiv	Accuracy
Verbraucher I	2,41	1,91	1,76	7,82	6,27	85,71%
Verbraucher II	1,81	1,71	1,29	2,9	3,17	54,69%
Verbraucher III	6,34	2,95	4,14	7,62	5,66	11,43%

Tab. 3: Vorhersagegenauigkeit der Modelle und Entscheidungsregeln auf Simulationsdatensatz

Während die Validierung des Klassifikators von Verbraucher I positiv ausfällt (> 80%, ähnlich zu dem Ergebnis aus Abschnitt 4.2), fallen hingegen die Klassifikatoren von Verbraucher II und III deutlich schlechter aus. Wie in Abschnitt 3 beschrieben, haben die Vorhersagen des Klassifikators einen Einfluss auf die Berechnung der Zeitreihenvorhersagen des initialen ARIMA-Modells durch die Auswahl des Trainingsdatensatzes. Eine ARIMA-Modelländerung wird hierbei vermutlich dringend benötigt.

Bei allen drei Datensätzen konnte eine bessere Vorhersagegenauigkeit mit der Modelländerung erzielt werden. Die Auswertungen bei zwei der Verbrauchsdaten ergaben eine geringe Verbesserung, wohingegen beim dritten Datensatz eine erhebliche Besserung nachgewiesen wurde (Vorhersagefehler halbiert).

Aufgrund des Nachweises einer Verschlechterung der Genauigkeit der Entscheidungsregeln für Verbraucher II und III, zeigen weiterführende Untersuchungen in [He18] Erkenntnisse einer Entwicklung des Datenstroms (*Concept Evolution*).

Definition 6.1 (Concept Evolution) *Concept Evolution tritt als Resultat vollkommen neuer Klassenobjekte auf, die sich im Lauf der Zeit im Datenstrom entwickelt haben [Mo10]. Durch die Entwicklung des Konzepts im Datenstrom sind die Anzahl der vorhandenen Klassenobjekte dynamisch. Bei einer unentdeckten Concept Evolution werden bei einer Klassifikation die Instanzen der neuen Klasse fälschlicherweise bestehenden Klassen zugewiesen, wodurch der Klassifikationsfehler zunimmt [Ha16].*

[10] In der Literatur zu maschinellem Lernen wird die Terminologie zu „Validierung“ und „Test“ gegebenenfalls umgekehrt beschrieben.

Bei einer Concept Drift Erkennung müssen *vorhersagende Modelle* (prädiktiv), wie zum Beispiel ARIMA, an das neue Konzepte im Datenstrom angepasst werden. Davon abweichend wird bei einer Concept Evolution Erkennung eine Anpassung von *beschreibenden Modellen* (deskriptiv), wie zum Beispiel dem Referenzmodell oder die Entscheidungsregeln[11], benötigt. Sowohl bei der Erkennung eines Concept Drift als auch Concept Evolution muss das ausführende Modell neu trainiert werden.

Zur Untersuchung einer Concept Evolution wird in [He18] auf dem Simulationsdatensatz erneut eine Clusteranalyse und Klassifikation durchgeführt und mit den Ergebnissen basierend auf den historischen Datensatz verglichen. Die Idee liegt hierbei, dass im Zuge einer Concept Evolution Beobachtung geprüft wird, inwiefern sich die Daten im Zeitraum von knapp zwei Jahren (610 Tagen) verändert haben.

Durch den Vergleich stellte sich heraus, dass bei allen drei Datensätzen eine Verschiebung der Clusterzentren festgestellt wurde und dass neu entstandene Klassenobjekte fälschlicherweise den bestehenden Klassen zugeteilt wurden. Weiterhin zeigte sich bei der Untersuchung in [He18], dass nicht die Vorhersagegenauigkeit des Klassifikators entscheidend ist, sondern ob und wie stark der Klassifikator im Zuge einer Concept Evolution sich verändert. Trotz einer Verschiebung der Clusterzentren hat sich der Klassifikator von Verbraucher I und II kaum verändert. Daher sind die Ergebnisse der Zeitreihenvorhersage für Verbraucher II weiterhin positiv, trotz schlechter Accuracy des Klassifikators ($54,69\%$ und nur $1,81$ kWh MAE). Die neuen Klassenobjekte werden zwar einer falschen Klasse zugeordnet, jedoch mit der richtigen Regel zur Modellberechnung abgefragt. Anders waren die Ergenisse bei Verbraucher III. Hier haben sich auch die Entscheidungsregeln im Zuge der Concept Evolution grundliegend verändert. Dies hatte Auswirkungen auf die Findung des initialen *Day-Ahead* Modells, das zu einem mittleren absoluten Fehler von 6,43 kWh führte. Mit dem im Bericht vorgestellten Verfahren konnte hier der Vorhersagefehler halbiert werden.

7 Zusammenfassung und Ausblick

Zusammenfassend ist das in der Arbeit implementierte eMMS im Stande drei wesentliche Aspekte zu behandeln:

1. Zum einen werden unvorhergesehene Zeitreihen, für die keine Regeln bei der Klassifikation gefunden wurden (Anomalien), frühzeitig mit der Concept Drift Erkennung erfasst. Ein neues vorhersagendes Modell zur Zeitreihenprognosen wird mit Hilfe eines Referenzmodells berechnet.
2. Zum anderen ist das System in dem untersuchten Beispiel bei der Verschiebung von Clusterzentren im Referenzmodell robust, solange die Entscheidungsregeln

[11] Der Klassifikator wird im Rahmen des Berichts als beschreibendes Modell bezeichnet, weil das Modell Merkmale extrahiert, die signifikant das Konzept in den Daten beschreiben. Diese Erkenntnisse werden zur anschließenden Zeitreihenvorhersage genutzt.

zur initialen ARIMA-Modellberechnung sich nicht verändert haben. Durch die zur Laufzeit durchgeführte Anwendung des Referenzmodells werden neue Zeitreihen einer bestehenden Referenzzeitreihe zugeordnet und im DBMS vermerkt. Die Auswahl der vier aktuellsten Zeitreihen einer Referenzzeitreihe bei gleichbleibender Regel führt daher zu einer Relativierung des Vorhersagefehlers spätestens nach der vierten Prognose.

3. Schließlich zeigte sich in [He18], dass das System auch im Zuge einer Concept Evolution mit sich ändernden Entscheidungsregeln angemessene Vorhersagen treffen kann.

Der Bericht zeigt, dass die Verwendung des eMMS die Vorhersagequalität des Stromverbrauchs für den Folgetag von allen drei untersuchten Verbrauchern verbessert. Insbesondere wenn die Verbrauchswerte von Tag zu Tag stark schwanken oder eine Concept Evolution in den Daten vorliegt, ist dieses Konzept von Vorteil.

Zum Abschluss soll auf die clusterbasierte Änderungserkennung im Datenstrom aufmerksam gemacht werden. Die in dem Bericht verwendete fensterbasierte Änderungserkennung benötigt Expertenwissen zur Definition der Fenstergröße und des Schwellwertes. Um auch hier den Menschen im eMMS Prozess zu entlasten, empfiehlt sich eine automatisierte Änderungserkennung durchzuführen (vgl. Abschnitt 3). Zum einen muss kein Schwellwert des Vorhersagefehlers vordefiniert werden, sondern ein Concept Drift wird automatisiert erkannt. Zum anderen ließe sich das allgemeine Datenstromverhalten untersuchen und eine Concept Evolution Erkennung implementieren. Des Weiteren könnte eine Änderungsrate für einen Modellwechsel mit hinzugezogen werden, zur Abschätzung der Parametereinstellungen der Concept Drift Erkennung (Fenstergröße und ggf. Schwellwert bei fensterbasierter Strategie). Diese Aspekte werden in dem hier vorgestellten eMMS nicht berücksichtigt und könnten für die Weiterentwicklung aufgenommen werden.

Literatur

[Ak74] Akaike, H.: A new look at the statistical model identification. Automatic Control, IEEE Transactions on 19/6, S. 716–723, 1974.

[ASW15] Aghabozorgi, S.; Shirkhorshidi, A. S.; Wah, T. Y.: Time-series clustering – A decade review. Information Systems 53/, S. 16–38, 2015, ISSN: 0306-4379.

[Be03] Bernstein, P. A.: Applying Model Management to Classical Meta Data Problems. Proceedings of the Conference on Innovative Data Systems Research (CIDR)/, 2003.

[BFR98] Bradley, P. S.; Fayyad, U.; Reina, C.: Scaling Clustering Algorithms to Large Databases. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. KDD'98, AAAI Press, New York, NY, S. 9–15, 1998.

[Bi95] Bishop, C. M.: Neural Networks for Pattern Recognition. Oxford University Press, Inc., New York, NY, USA, 1995, ISBN: 0198538642.

[BJ70] Box, G. E.; Jenkins, G. M.: Time Series Analysis: Forecasting and Control. Holden-Day, Inc., San Francisco, CA, USA, 1970.

[BK09] Buchmann, A. P.; Koldehofe, B.: Complex Event Processing. it - Information Technology 51/, S. 241–242, 2009.

[Da11] Dannecker, L.; Böhm, M.; Lehner, W.; Hackenbroich, G.: Forcasting Evolving Time Series of Energy Demand and Supply. In: Advances in Databases and Information Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 302–315, 2011, ISBN: 978-3-642-23737-9.

[Da15] Dannecker, L.: Energy Time Series Forecasting – Efficient and Accurate Forecasting of Evolving Time Series from the Energy Domain. Springer Fachmedien Wiesbaden, 2015.

[DK84] Dolk, D. R.; Konsynski, B. R.: Knowledge Representation for Model Management Systems. IEEE Transactions on Software Engineering SE-10/6, S. 619–628, Nov. 1984.

[FRB98] Fayyad, U.; Reina, C.; Bradley, P. S.: Initialization of Iterative Refinement Clustering Algorithms. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. KDD'98, AAAI Press, New York, NY, S. 194–198, 1998.

[Ha16] Haque, A.; Khan, L.; Baron, M.; Thuraisingham, B.; Aggarwal, C.: Efficient handling of concept drift and concept evolution over Stream Data. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE). S. 481–492, Mai 2016.

[HA18] Hyndman, R. J.; Athanasopoulos, G.: Forecasting: Principles and Practice. OTexts, 2018, ISBN: 9780987507112.

[He18] Hegenbarth, Y.: Konzept und Implementierung eines echtzeitfähigen Model Management Systems, Magisterarb., Hochschule Darmstadt h_da, Germany, 2018.

[HK08] Hyndman, R.; Khandakar, Y.: Automatic Time Series Forecasting: The forecast Package for R. Journal of Statistical Software, Articles 27/3, S. 1–22, 2008, ISSN: 1548-7660.

[HKP11] Han, J.; Kamber, M.; Pei, J.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2011, ISBN: 0123814790, 9780123814791.

[JBB15] Jović, A.; Brkić, K.; Bogunović, N.: A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). S. 1200–1205, 2015.

[LWG14] Liu, J.; Wilson, A.; Gunning, D.: Workflow-based Human-in-the-Loop Data Analytics. In: Proceedings of the 2014 Workshop on Human Centered Big Data Research. HCBDR '14, ACM, Raleigh, NC, USA, 49:49–49:52, 2014, ISBN: 978-1-4503-2938-5.

[Ma67] Macqueen, J.: Some methods for classification and analysis of multivariate observations. In: In 5-th Berkeley Symposium on Mathematical Statistics and Probability. S. 281–297, 1967.

[Mo10] Mohammad, M. M.; Chen, Q.; Khan, L.; Aggarwal, C.; Gao, J.; Han, J.; Thuraisingham, B.: Addressing Concept-Evolution in Concept-Drifting Data Streams. In: Proceedings of the 2010 IEEE International Conference on Data Mining. ICDM '10, IEEE Computer Society, Washington, DC, USA, S. 929–934, 2010, ISBN: 978-0-7695-4256-0.

[Qu93] Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993, ISBN: 1-55860-238-0.

[RH96] Ripley, B.; Hjort, N.: Pattern Recognition and Neural Networks. Cambridge University Press, 1996, ISBN: 9780521460866.

[SAM96] Shafer, J. C.; Agrawal, R.; Mehta, M.: SPRINT: A Scalable Parallel Classifier for Data Mining. In: Proceedings of the 22th International Conference on Very Large Data Bases. VLDB '96, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, S. 544–555, 1996, ISBN: 1-55860-382-4.

[Sc78] Schwarz, G.: Estimating the Dimension of a Model. The Annals of Statistics 6/2, S. 461–464, 1978, URL: https://projecteuclid.org/euclid.aos/1176344136.

[Se16] Self, J. Z.; Vinayagam, R. K.; Fry, J. T.; North, C.: Bridging the Gap Between User Intention and Model Parameters for Human-in-the-loop Data Analytics. In: Proceedings of the Workshop on Human-In-the-Loop Data Analytics. HILDA '16, ACM, San Francisco, California, 3:1–3:6, 2016, ISBN: 978-1-4503-4207-0.

[Tr13] Tran, D.: Change detection in streaming data, Diss., Technische Universität Ilmenau, Germany, 2013.

[Ts04] Tsymbal, A.: The Problem of Concept Drift: Definitions and Related Work. In: Department of Computer Science Trinity College Dublin. Ireland, Mai 2004.

[Ur03] Urbanek, S.: Rserve – A Fast Way to Provide R Functionality to Applications. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003). 2003.

[Va16] Vartak, M.; Subramanyam, H.; Lee, W.-E.; Viswanathan, S.; Husnoo, S.; Madden, S.; Zaharia, M.: ModelDB: a system for machine learning model management. In. HILDA '16 – Proceedings of the Workshop on Human-In-the-Loop Data Analytics, San Francisco, California, USA, S. 10–12, Juni 2016.

[Zl10] Zliobaite, I.: Learning under Concept Drift: an Overview. CoRR abs/1010.4784/, 2010.