DIPLOMARBEIT

Information aus Daten:
Maschinelles Lernen in der Gewässergüte

Surrogat-Parameter und Datenplausibilisierung

ausgeführt zur Erlangung des akademischen Grades
eines Diplom-Ingenieurs unter der Leitung von

ao. Univ.Prof. Dipl.-Ing. Dr.techn. Thilo Sauter

Univ.Ass. Dipl.-Ing. Marcus Meisel, Bakk.techn.

am

Institut für Computertechnik (E384)

der Technischen Universität Wien

durch

Andreas Winkelbauer, BSc
Matr.Nr. 0025977
1020 Wien, Schreygasse 19/1/22

Wien, im Juni 2017

_____

Kurzfassung

Diese Diplomarbeit befasst sich mit der messtechnischen Erfassung der wesentlichen Parameter der Gewässergüte und wie aus der Sammlung zusätzlicher Daten schließlich Informationen über Zusammenhänge der Messgrößen erhalten und genutzt werden können. Neben der Messung der Konzentration des gelösten Sauerstoffs im Gewässer werden ausgewählte Daten zu meteorologischen Umgebungsbedingungen mit vergleichsweise kostengünstiger Sensortechnik erfasst und durch Einsatz von Methoden des maschinellen Lernens einer mathematischen Modellierung unterzogen. Die Eignung der Methodik durch Vergleich eines generalisierten linearen Modells und der Modellierung eines künstlichen neuronalen Netzes mit der tatsächlich erfassten Sauerstoffkonzentration wird evaluiert. Das gewonnene Modell ist für die Verwendung zur Datenvorhersage als Surrogat-Parameter wenig geeignet, für die statistische Absicherung der Sauerstoffmessung, also der Plausibilisierung eines herkömmlich erfassten Parameters des Gewässerzustands, aber gut verwendbar.

Abstract

This thesis addresses the measurement of water quality and how the collection of additional data can be used to obtain information regarding the relationships between these measured variables. In addition to measuring the concentration of dissolved oxygen in the water body, selected data on meteorological, environmental conditions are compiled, using comparatively cost-effective sensor technology and fed into mathematical modeling algorithms by using methods of machine learning. The suitability of the methodology is evaluated by comparing a generalized linear model and the artificial neural network model with the detected oxygen concentration. The model obtained is not suitable as a surrogate parameter for use in data prediction, but is well-suited for the statistical validation of the measured oxygen concentration, that is, the plausibility check of the conventionally recorded parameter of the water state.

Danksagung

An erster Stelle bedanke ich mich bei Herrn Marcus Meisel und Herrn Professor Dr. Thilo Sauter für die gute Betreuung und fachliche Führung im Rahmen der Erstellung dieser Arbeit.

Großer Dank gebührt Herrn Professor Dr. Jörg Krampe für die Förderung und persönliche Teilnahme, besonders während dem Endspurt meines Studiums.

Meinen Kolleginnen und Kollegen danke ich für die gute Zusammenarbeit und vor allem für die tatsächlich gelebte Interdisziplinarität an unserem Institut.

An meine Eltern richte ich den aufrichtigen Dank für ihre Unterstützung, die es mir ermöglicht hat, meinen Weg in der Technik zu gehen.

Abschließend gilt ganz besonderer Dank meiner Frau Bettina für ihre Geduld und den Rückhalt, zu jeder Zeit.

Inhaltsverzeichnis

Abkürzungen

3G	dritte Generation des Mobilfunks
ADC	Analog Digital Converter
ADR	automatisierter Datenreport
AI	Artificial Intelligence
ANN	Artificial Neural Network
BD	Block Diagram
CMOS	complementary metal-oxide-semiconductor
CPS	cyber-physikalisches System
CPU	Central Processing Unit
CSI	Camera Serial Interface
DAC	Digital Analog Converter
DBL	Double (Datentyp)
DOC	Dissolved Organic Carbon
DWA	Deutsche Vereinigung für Wasserwirtschaft, Abwasser und Abfall
EM	Environmental Monitoring
ERP	Enterprise Resource Planning
EWM	Environmental Water Monitoring
FP	Front Panel
glm	generalized linear model
GND	ground
GPIO	general purpose input output
GPU	Graphics Processing Unit
HDMI	High Definition Multimedia Interface
I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IDE	Integrated Development Environment
IMW	innovative Messtechnik in der Wassergütewirtschaft
IoT	Internet of Things
ISE	Ionen-sensitiv
iTUWmon	intelligent information water monitoring networks
LabVIEW	Laboratory Virtual Instrumentation Engineering Workbench
LDO	Luminescent Dissolved Oxygen
LM pred	linear modeling prediction

LMS	least mean square
MEAS	measurement
MES	Manufacturing Execution System
MESZ	Mitteleuropäische Sommerzeit
MEZ	Mitteleuropäische Zeit
MSE	mean square error
MSElm	mean square error linear model
MSEnn	mean square error neural network model
MSR	Mess-, Steuer- und Regelungstechnik
NaN	not a number
NaWas	nachhaltige Wassergütewirtschaft
NIST	National Institute of Standards and Technology
NN pred	neural network modeling prediction
NoSQL	not only SQL
NTP	Network Time Protocol
ÖWAV	Österreichischer Wasser- und Abfallwirtschaftsverband
PAR	photosynthetically active radiation
PC	Personal Computer
PlausibilityASM	assumption of plausibility
PlausibilityDEF	definiton of plausibility
PLSR	Partial Least Square Regression
PoE	Power over Ethernet
Q	Volumenstrom
relHum	relative humidity
RPi	Raspberry Pi
RPROP	resilient backpropagation
SAC	spectral absorption coefficient
SCADA	Supervisory Control and Data Acquisiton
SCL	Serial Clock Line
SD	standard deviation
SDA	Serial Data Line
SiC	Siliziumcarbid
SignalSpec	signal specification
SOA	Service Oriented Architecture
SoC	System on Chip
SOP	Standard Operating Procedure
spc	spectrum
SPS	speicherprogrammierbare Steuerung
SQL	Structured Query Language
SSH	secure shell
T	temperature
TCP/IP	Transmission Control Protocol/Internet Protocol
UI	user interface

UID	universal identifier
UINT	unsigned Integer
UTC	Coordinated Universal Time
UV	ultraviolett
UV-A	nahes UV, Teilbereich der UV-Strahlung
ValueRAW	raw value
ValueSCAL	scaled value
VCC	voltage at the common connector
VI	Virtual Instrument
VPN	Virtual Private Network
WHO	World Health Organization

„It is vital to remember that information
– in the sense of raw data – is not knowledge,
that knowledge is not wisdom,
and that wisdom is not foresight.
But information is the first essential step to all of these.“

Arthur C. Clarke

1. Einleitung

Wasser ist eine zentrale und notwendige Ressource des Lebens und die Wasserinfrastruktur mit ihren Wasserversorgungs- und Abwasserentsorgungsanlagen wirkt als bestimmender Faktor der Lebensqualität einer Gesellschaft [Kroi04]. Für die Erfassung der Gewässergüte werden, neben der probebasierten chemisch-biologischen Analysemethodik in Laboratorien, sogenannte online-Messapparaturen verschiedener Hersteller eingesetzt. Die Mess- und Automatisierungstechnik umfasst hier den Bereich der Gütebestimmung zur Einhaltung gesetzlich vorgeschriebener Grenzwerte in der Wasserversorgung genauso wie eine ressourcenschonende und auf wirtschaftliche Gesichtspunkte ausgerichtete Steuerung und Regelung von Prozessen der Abwasserentsorgung, welche als eine Quelle der Rückführung der Ressource Wasser in den Wasserkreislauf fungiert.

Die Weiterentwicklungen der Computertechnik und hier im Speziellen die exponentiell wachsende Rechenleistung und Speicherkapazität [Moor65] führen zu neuen Fragestellungen und erweiterten Möglichkeiten der Informationsverarbeitung, welche unter anderem mit den Begriffen *Big Data*, *Internet of Things* und *maschinellem Lernen* umrissen werden können. Gleichzeitig steigen die Anforderungen an die Datenhaltung, Robustheit und Sicherheit der eingesetzten Systeme [SLCZ15]. Mit der zunehmenden Verdichtung von Messzeitreihen in der Datenerfassung treten zunehmend Fragestellungen im Hinblick auf generierbare Information aus Daten und einer möglichst automatisierten, zeitlich nahe zur Messung durchzuführenden Datenauswertung und Interpretation einer großen Anzahl an Messergebnissen hervor. Am Ende der Datenerfassung steht die Bewertung der Ergebnisse durch den Menschen und damit Fragestellungen der Datenpräsentation. Die Vermittlung der Ergebnisse einer automatisierten Vorprüfung von Messdaten ist ein wichtiger Einsatzbereich geeigneter Benutzerschnittstellen.

Die Modellierung der Äquivalentkonzentration von Gewässergüteparametern, beispielsweise auf Basis von Absorptionsspektren, ist ein erprobtes Verfahren in Forschung und Industrie; eine große Auswahl geeigneter, optischer Sondentechnik ist kommerziell verfügbar und im Einsatz. Basierend auf dem Konzept, aus der Messung eines umfassenden Zusammenhangs mehrere Einzelparameter zu generieren wird in dieser Arbeit die Tauglichkeit ausgewählter Verfahren des maschinellen Lernens unter zwei wesentlichen Gesichtspunkten untersucht: Modellierung ausgewählter Parameter des Gewässerzustands aus weiteren Messdaten sowie die automatisierte Plausibilitätsbewertung von Messergebnissen unter Einsatz der gefundenen Zusammenhänge. Die Zielsetzung umfasst die Abschätzung der Tauglichkeit dieser Algorithmen für den Einsatz im Bereich der kostengünstigen Erfassung von Leitparametern durch Einführung von Surrogat-Parametern unter Verzicht auf komplexe Sondentechnik. Als Surrogat-Parameter werden allgemein „Ersatzgrößen“ verstanden. Umgelegt auf diese Arbeit wird

der Ersatz der Datenerfassung einer bestehenden Sonde durch einen modellierten Parameter auf Basis weiterer Messgrößen verstanden. Eine weitere Zielsetzung ist die Abschätzung einer möglichen Verbesserung im Bereich der Datenplausibilisierung von Messwerten durch Einbeziehung einer größeren Zahl weiterer Messparameter und ihrer gewässerspezifischen, zeitlichen Entwicklung. Überlegungen zur Gestaltung ansprechender Benutzerschnittstellen für die intuitive Bedienung der Softwarewerkzeuge und die praktikable Datenpräsentation sind ebenfalls Teil der Arbeit.

In den folgenden drei Abschnitten werden Grundlagen und Begriffe zur Gewässergüte, der Automatisierungstechnik und schließlich des maschinellen Lernens dargestellt und aufbereitet.

1.1 Gewässergüte: Grundlagen und Definitionen

Ein Ökosystem wird nach [Scha12, S.204] definiert als ein „Beziehungsgefüge der Lebewesen untereinander (Biozönose) und mit ihrem Lebensraum (Biotop)" und ist nach [Kroi04] geprägt durch geologische Ressourcen, wie zum Beispiel dem Boden und klimatischen Bedingungen wie Niederschlag, Temperatur und Wind. Den Überbegriff der Betrachtungen bildet die Wissenschaft der Ökologie, auch *Haushaltslehre* genannt und wird unter Verwendung einer verkürzten Form nach [Bick93, S.8] wie folgt definiert: „Ökologie ist die Wissenschaft von den wechselseitigen Beziehungen zwischen Organismen und ihrer Umwelt".

Wasser, als eine wesentliche Voraussetzung für das Leben, spielt in allen Kreisläufen des Ökosystems unseres Planeten eine entscheidende Rolle und ist daher eine besonders schützenswerte, weil auch begrenzte, Ressource. Der Einfluss der menschlichen Zivilisation auf die Umwelt, speziell das ökonomische Konzept eines dauerhaften Wachstums, führt zu beschleunigtem Ressourcenverbrauch [KuSt92]. Von besonderem Interesse ist daher die Erfassung des Zustandes der Gewässer, wobei [Kroi04] speziell auf die Trennung der objektiven Beschreibung der Beschaffenheit von der Bewertung der Qualität bzw. der Güte hinweist. Die ersten Arbeiten zur objektiven Beschreibung dazu wurden 1908 von Kolkwitz und Marsson vorgelegt und Kolkwitz stellte 1950 das sogenannte Saprobiensystem vor [Kolk50]. Hierbei wird der Verschmutzungsgrad des Gewässers auf Basis einer beobachteten Artenzusammensetzung von Tieren und Pflanzen am Grund eines Fließgewässers erfasst. Als Fließgewässer ist laut [Scha12, S.93] „ein Gewässer mit mehr oder weniger starker Strömung" definiert. Die Beschaffenheit des Gewässers wird in einer sogenannten Beschaffenheitsskala mit dem Wertebereich 1 (unbelastet bis sehr gering belastet) bis 4 (übermäßig verschmutzt) abgebildet und dient als Maß für die Selbstreinigung eines belasteten Fließgewässers, im Detail dargestellt in [UhHo01, S.319]. Aufbauend auf dieser objektiven Festlegung kann in einem nächsten Schritt die Gewässerqualität als subjektive Festlegung als Vergleich mit einem im Voraus bestimmten Referenzzustand verstanden werden [KuSt92].

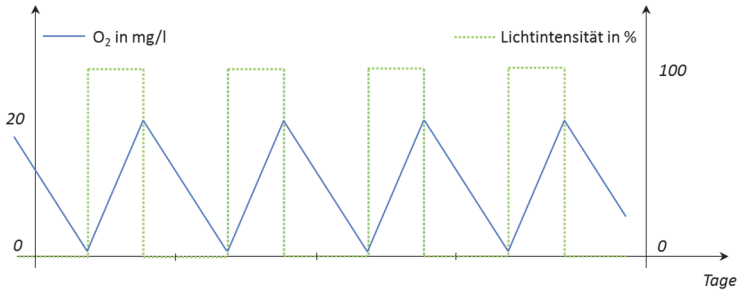

Abbildung 1: Lichtintensität und Sauerstoffgehalt, eigene Darstellung nach [UhHo01, S.11]

Die vollständige Beschreibung des Gewässerzustandes ist aufgrund der Komplexität der biologischen und chemischen Teilsysteme, sowie der ständigen Änderung der Parameter durch im Einzugsgebiet wirkende Prozesse sehr schwierig und eine Modellierung nur unter stark vereinfachenden Annahmen möglich [Kroi04]. Ein besonders wichtiger Faktor für die biologischen Prozesse eines Gewässers ist die Sonnenstrahlung und dem damit, neben dem Eintrag aus der Atmosphäre, direkt verbundenen Sauerstoffgehalt des Gewässers durch die Photosynthese [DoHK01, S.50].

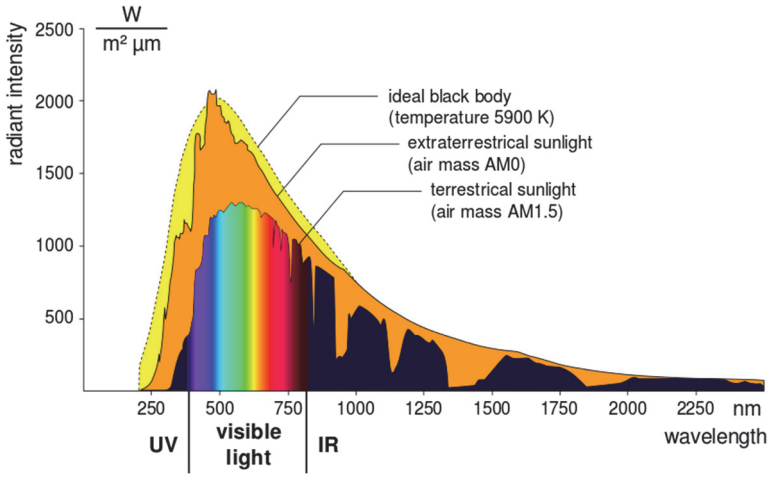

Abbildung 2: Sonnenstrahlung, Intensität über der Wellenlänge [Degr06][1]

Unter Photosynthese wird die Assimilation von Kohlenstoffdioxid und Wasser zu Glucose und Sauerstoff durch Organismen als Produzenten von organischen Substanzen [UhHo01] verstanden; die vereinfachte Zusammenhänge sind in Gleichung **1.1** dargestellt.

$$6\,CO_2 + 6\,H_2O \rightarrow C_6H_{12}O_6 + 6\,O_2 \qquad \textbf{1.1}$$

Unter Einwirkung von Licht wird Kohlenstoffdioxid CO_2 und Wasser H_2O durch in Organismen stattfindende Umbauvorgänge zu energiereicher Glucose $C_6H_{12}O_6$ umgesetzt und dabei Sauerstoff O_2 frei. Abbildung 1 zeigt die exemplarischen Zusammenhänge für Phytoplankton in einem Labormodell.

[1] Autor: Degreen, Lizenz: Creative Commons Attribution-Share Alike 2.0 Germany

Hohe Lichtintensität bewirkt steile, ansteigende Flanken im Sauerstoffgehalt [UhHo01]. Die photosynthetische Produktion als sogenannte Primärproduktion wird durch die photosynthetisch aktive Strahlung (*PAR*, engl. *photosynthetically active radiation*) im Wellenlängenbereich 400 bis 700 nm bestimmt; dieser Bereich deckt sich weitestgehend mit dem vom Menschen visuell wahrnehmbaren Bereich. In Abbildung 2 ist der typische Verlauf der Strahlungsenergie über der Wellenlänge des Sonnenlichts dargestellt. Die Absorptionsspektren der für die Photosynthese relevanten Pigmente, sind für Chlorophylle (zum Beispiel bei Landpflanzen) und für Carotinoide (zum Beispiel bei Algen) unterschiedlich ausgeprägt [HeHe98, S.14]. Im Gewässer werden, durch den vertikalen Abfall der Lichtintensität bedingt, zwei Bereiche unterschieden: Die Produktionsschicht mit der Bildung von Sauerstoff und organischen Substanzen unter Verbrauch von Kohlenstoffdioxid und die Abbauschicht mit Verbrauch von Sauerstoff und Bildung von Kohlenstoffdioxid [UhHo01, S.35].

Bereits [Kolk50] wies auf die große Rolle des Sauerstoffes der Gewässer bei der Verteilung der Organismen hin, obgleich er auf weitere beteiligte, komplizierte Prozesse hinweist. Für die Beschreibung des Gewässerzustands sind laut [KuSt92] „objektive, messbare Daten chemischer, physikalischer und biologischer Größen wie beispielsweise Stoffkonzentrationen, Temperatur, Biomasseproduktivität oder Fließgeschwindigkeiten“ vonnöten. Für die Erreichung des Ziels eines *guten* ökologischen und chemischen Zustandes der Gewässer wurde auf europäischer Basis die sogenannte EU-Wasserrahmenrichtlinie formuliert [Euro14], welche mit Änderungen seit dem Jahr 2000 in Kraft ist. Der *gute* Zustand ist hier als Zustand definiert, der von einem weitgehend anthropogenen, also vom Menschen unbeeinflussten Zustand [Scha12, S.16], nur geringfügig abweicht [Umwe17].

Die möglichst objektive Erfassung des Gewässerzustandes ist eine wichtige Grundlage der Qualitätsbewertung eines Gewässers. Abgesehen von den Laboranalysen entsprechender Stichproben ergibt sich erst durch möglichst kontinuierliche Messdatenerfassung ein umfassendes Bild. Vor allem die Verhältnisse vom im Gewässer gelösten Sauerstoff gehen als wichtiger Faktor in die Bewertung der Gewässergüte ein. Im folgenden Abschnitt werden für die Messdatenerfassung wichtige Grundbegriffe der Automatisierungstechnik dargestellt. Die Spezialisierung auf die Gewässergütemesstechnik und die Herausforderungen beim Betrieb werden in Abschnitt 2.1 skizziert.

1.2 Automatisierungstechnik: Begriffe und Anwendungen

Das Hauptziel der Automatisierungstechnik ist die Sicherung eines zuverlässigen, wirtschaftlichen Betriebs einer Anlage und durch Einsatz technischer Gewerke in der gesamten Signalkette beginnend bei der messtechnischen Erfassung über Eingriffe ohne Rückwirkung bis hin zu geschlossenen Wirkungsabläufen zur Beeinflussung von Maschinen [Wink16].

Messen, Steuern und Regeln als Teilgebiete der Automatisierungstechnik befassen sich mit der Erfassung und der gezielten Beeinflussung der physikalischen Wirklichkeit durch Maschinen beziehungsweise Anlagen. *Messen* ist dabei als Vorgang der Informationsgewinnung durch experimentelle Erfassung einer physikalischen Größe (siehe DIN 1319) definiert. *Steuern* bezeichnet die gezielte Beeinflussung von Ausgangsgrößen eines Systems auf Basis der (messtechnisch) erfassten Eingangsgrößen und ist durch einen offenen Wirkungsweg gekennzeichnet. Als *Regeln* wiederum wird der Vorgang des (kontinuierlichen) Vergleichs einer oder mehrerer zu regelnder Größen, der Regelgrößen

(IST-Wert), mit einer oder mehreren Führungsgrößen (SOLL-Wert) im Sinne eines geschlossenen Wirkungsweges durch Rückkopplung verstanden, siehe dazu die Normen [Dine08, Dini14] und [BeTV09]. Die vereinfachten Zusammenhänge sind in Abbildung 3 dargestellt.

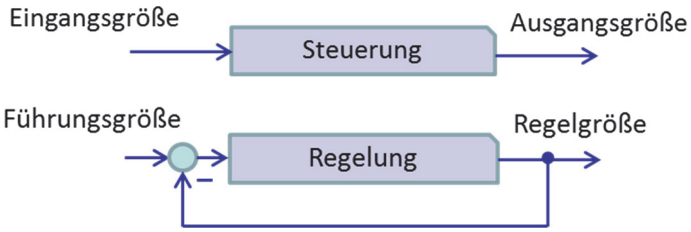

Abbildung 3: Steuern und Regeln [Wink17]

Nach [Litz13] zählt neben Messen, Steuern und Regeln auch die Überwachung und automatisierte Meldung bei Abweichung von einem SOLL-Bereich, sowie das Anzeigen und Bedienen der technischen Prozesse über geeignete Schnittstellen zu den wichtigsten Funktionen der Automatisierung.

Die Leittechnik wird in [Dini14] als „zweckmäßige Maßnahmen an oder in einem Prozess, um vorgegebene Ziele zu erreichen“ definiert. Dazu zählen nach [BeTV09, S.9] sämtliche Aspekte der Mess-, Steuer- und Regelungstechnik genauso wie die Informationstechnik, die Ordnung von Prozessen der Produktion sowie Methoden und Werkzeuge für das Design und den Betrieb von Leitsystemen. Das Informationsmodell der Leittechnik wird durch die hierarchisch aufgebaute Automatisierungspyramide geprägt (siehe Abbildung 4); *Daten* werden hierbei von jeder Schicht und über definierte Schnittstellen erfasst, ausgetauscht, eingegeben, bearbeitet, übertragen und ausgegeben [BeTV09]. Die Ebenen sind gegliedert nach der mit Planung der Produktion befassten *Unternehmensleitebene* (engl. *ERP*, *Enterprise Resource Planning*), der *Betriebsebene* mit Überwachungsaufgaben der Produktion (engl. *MES*, *Manufacturing Execution System*), der *Prozessleitebene* mit den SCADA-Systemen (engl. *Supervisory Control and Data Acquisiton*) zur Überwachung und Steuerung der technischen Prozesse, der Steuerungsebene zur Steuerung und Regelung der Anlagen mittels SPS (speicherprogrammierbare Steuerungen) und schließlich der untersten Feldebene mit den zugehörigen Sensoren und Aktoren (in Anlehnung an [Dini14]).

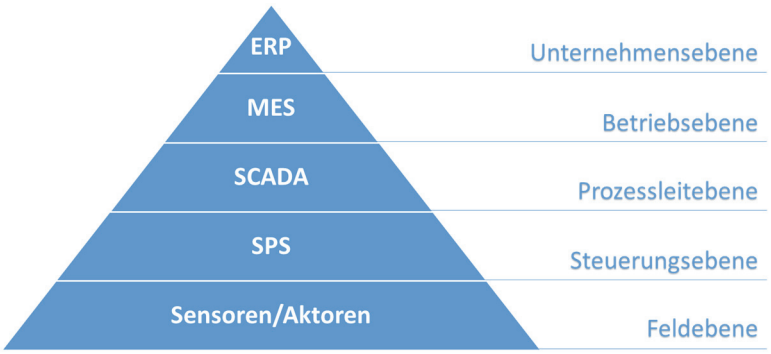

Abbildung 4: Automatisierungspyramide [Wink16] und eigene Darstellung nach [Dine14]

Die fortwährende Weiterentwicklung und Vernetzung der leittechnischen Anlagen führte zu einer zunehmenden Auflösung der starr hierarchisch aufgebauten Ebenen hin zu einer neuen Sicht der Leittechnik als verteiltes Netzwerk aus miteinander kommunizierenden *Services*, den sogenannten service-orientierten Architekturen, kurz *SOA* (siehe dazu [CoBK14, KCJD10]). Kennzeichnend für diesen Übergang ist die zunehmende Anwendung von Vernetzung auf Basis der Internet-Protokolle über Virtualisierung der Rechnereinheiten und Visualisierung auf mobilen Zugriffspunkten bis hin zur Datenhaltung und Datenverarbeitung in verteilten Rechenzentren, also dem Einsatz von *Cloud Computing*. Das National Institute of Standards and Technology (NIST) [MeGr11] definiert *Cloud Computing* als

> „model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (…) that can be rapidly provisioned and released with minimal management effort or service provider interaction“,

ein Modell für den ubiquitären Zugang zu Rechenressourcen, welche kurzfristig zugewiesen und wieder freigegeben werden können. Als *Smart Devices* beschreibt [Thom05] verteilte Systeme, gekennzeichnet durch Speicher, Sensorik, Aktorik, Rechenleistung und Vernetzung, auf relativ kleinem Raum vereint.

Die typischen Aufgaben und die Domäne einer serviceorientierten Architektur sind in Abbildung 5 dargestellt. Die Kapselung der Dienste und die standardisierten Datenschnittstellen, wie aus der klassischen Automatisierungspyramide bekannt, bleiben im Wesentlichen erhalten, die Ordnung ist jedoch nicht mehr streng hierarchisch geprägt. Die Herausforderungen der Modularisierung werden unter anderem in [FMJB15] beschrieben. Eine Herangehensweise zur Modularisierung kann beispielsweise die Trennung Datenerfassung von der Bewertung durch automatisierte Untersuchung auf Plausibilität hin sein, welche nicht mehr am selben Gerät stattfindet.

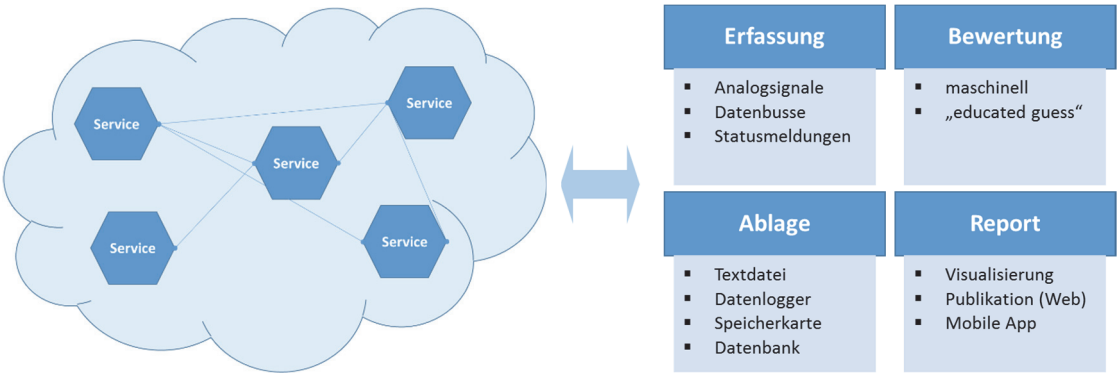

Abbildung 5: Serviceorientierte Architektur [Wink16]

Kennzeichnend einer *SOA* ist auch der Einsatz einer großen Zahl verteilter Datenerfassungssysteme anstelle einer zentral eingerichteten speicherprogrammierbaren Steuerung. Ein in diesem Zusammenhang oft gebräuchlicher Begriff zur Beschreibung dieser verteilten Datenerfassungs- und Rechensysteme ist das sogenannte *IoT* (engl. *Internet of Things*). Basis ist wiederum die Vernetzung von Dingen bzw. Objekten und deren Interaktion über standardisierte (Software-)Schnittstellen, beschrieben bei

[AtIM10]. Eine große Zahl von dezentralen, eng miteinander vernetzten Datenerfassungsgeräten, welche untereinander Services anbieten und Daten austauschen, wird auch *CPS*, cyber-physikalisches System, genannt [SLCZ15]. Typische *CPS* zeichnen sich durch eine große Zahl an Sensoren und Aktoren sowie einem hohen Datenaufkommen aus. Nach [Voge14] beginnen diese Systeme ab dem Einsetzen der Informationsflüsse durch Vorliegen binärer Daten. Für die Betrachtungen in dieser Arbeit wird jedem Datensatz neben seinem Wertcharakter der (Mess-)Größe auch immer sein zeitlicher Charakter, initial bestimmt durch den Zeitpunkt der Messdatenerfassung, zugerechnet. Beim Übergang von herkömmlichen, leittechnisch-geprägten Systemarchitekturen in serviceorientierte, cyber-physikalische Systeme sind nach [DRCC12] drei wesentliche Bereiche zu beachten: (1) Verwendung der gleichen Tools für die Integration von Benutzerschnittstellen und Steuerungsaufgaben, (2) Gruppierung von Geräten ähnlicher Aufgabenbereiche bei der Migration in eine SOA und (3) Wahrung der Echtzeit-Fähigkeit aller beteiligten Steuerungen. Die besonderen Anforderungen von CPS bezüglich *Cyber Security* sind in [Redd14] dargestellt, [Mcka12] zeigt einen Überblick der gängigen Bedrohungsszenarien.

Eine wichtige Triebfeder der Entwicklung in diesem Bereich ist die Miniaturisierung eingebetteter Systeme und die sich daraus ergebenden neuen Möglichkeiten der verteilten Datenerfassung und Datenverarbeitung. Nicht zuletzt durch die steigende Vernetzung der einzelnen Komponenten der Datenerfassung, prinzipiell auch über weite Strecken hinweg, rücken Aspekte der IT-Datensicherheit (engl. *security*) in den Mittelpunkt. Die Anforderungen werden in der IEC-Norm 624434 [Dini15] zusammengefasst; abgesehen von diesem Standardwerk sind weitere Empfehlungen verfügbar, beispielsweise ein Leitfaden des deutschen Bundesamts für Sicherheit in der Informationstechnik [Bund14] sowie die 2016 in Kraft getretene EU-Richtlinie [Euro16] über „Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen“ mit Auflagen für Betreiber und Anbieter sogenannter *essentieller Dienste*. Neben Energie, Transport, Gesundheit, Internet und Finanzwesen wird auch die Wasserversorgung als essentieller Dienst verstanden. Neue Herausforderungen in der industriellen Wassergütemesstechnik entstehen durch die Ablösung hierarchisch geprägter Leittechnik beispielweise durch erhöhte Herausforderungen im Bereich der IT-Sicherheit durch Fernzugriff des Bereitschaftspersonals auf kritische Infrastruktur. Cloud-Computing ist zum gut bekannten Schlagwort auch im Bereich der Wasserwirtschaft geworden. Anforderungen an die *Cloud Security* werden in [Tian12] dargestellt, vom Bundesverband der Energie- und Wasserwirtschaft ist außerdem ein Whitepaper zu den Anforderungen an sichere Steuerungs- und Telekommunikationssysteme [Bdew15] verfügbar. Maßnahmen der funktionalen Sicherheit (engl. *safety*) müssen durch die Anlagen vor Ort gegeben sein und sind in einer Reihe von Sicherheitsnormen und Gesetzen geregelt, eine erste Anlaufstelle bietet hier [Etg17].

In diesem Zusammenhang sei erwähnt: Es bestehen nach [Schn09] Anzeichen dafür, dass die vollständige und dauerhafte Speicherung sämtlicher anfallender Daten finanziell günstiger ist als die potentiell aufwändige und damit teure Löschung von Daten aus einem gegebenen Bestand. Bruce Schneier führt als Beweis dieser These die Datenschutzerklärung von Gmail, dem E-Mail-Dienst des Google-Konzerns an; wenn der Benutzer Daten entfernt, werden diese von den Sicherungssystemen unter Umständen nicht gelöscht, bleiben also weiter bestehen (siehe dazu [Goog17]).

Vom Gesichtspunkt der erfassten und weiterzuverarbeitenden Daten und ihrer entsprechend zunehmenden Heterogenität und Dichte, wird der Sammelbegriff *Big Data*, in etwa mit „Massendaten“ übersetzt, eingeführt (die Schwierigkeit einer Abgrenzung wird in [WaBa13] dargestellt). Eine erste Charakterisierung wurde durch das 3V-Modell, 2001 von [Lane01] vorgeschlagen. Die Daten charakterisierenden Begriffe dabei sind *Volume* (große Datenmengen), *Velocity* (hohe Datenraten und hohe Anforderungen in der Analyse) und *Variety* (Vielseitigkeit in Bezug auf Semantik und Struktur). Im Jahr 2013 wurde von [DWLL13] eine weitere Verfeinerung um *Value* (statistische Bewertungen) und *Veracity* (Aspekte der Datenauthentizität und Vertrauenswürdigkeit) vorgeschlagen (siehe dazu auch Abbildung 6).

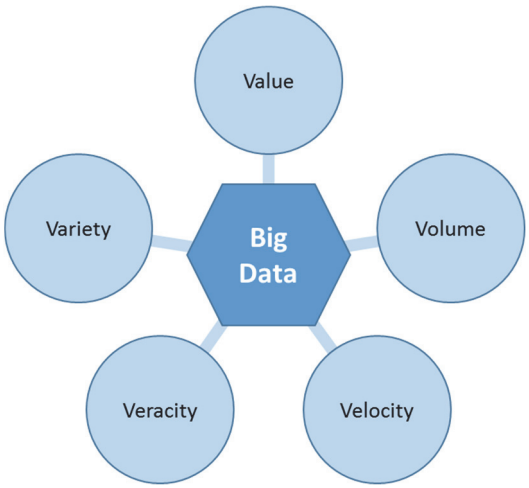

Abbildung 6: Definition von Big Data, eigene Darstellung nach [DWLL13]

Allen Entwicklungen im Bereich der Datenerfassung und -verarbeitung gemein ist die ausgeprägte Nutzung der seit den 1970er-Jahren exponentiell wachsenden, zur Verfügung stehenden Rechenleistung von Computersystemen. Gordon Moore hat dem nach ihm benannten Zusammenhang, wonach sich Komplexität und damit die Leistungsfähigkeit bei minimalen Kosten jedes Jahr um etwa den Faktor 2 erhöht, entdeckt; im Original schreibt [Moor65]: „The complexity for minimum component costs has increased at a rate of roughly a factor of two per year (…).“. Darauf basierend wurde eine exponentiell wachsende Rate der Anzahl von Komponenten je Schaltkreis prognostizierte, welche als Moorsches Gesetz bekannt wurde (siehe dazu Abbildung 7) und sich mittlerweile zu einer Art selbsterfüllende Prophezeiung für die Industrie entwickelt hat.

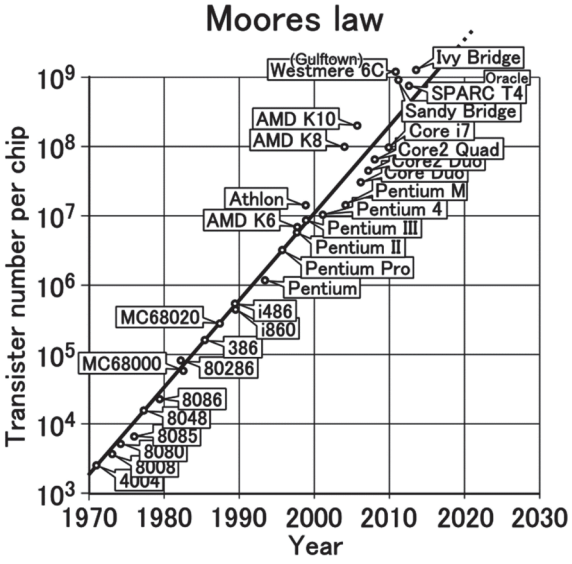

Abbildung 7: Moorsches Gesetz [Shig11][2]

In späteren Arbeiten wurde die erwartete Rate der exponentiellen Steigerung durch Gegenüberstellung mit den Entwicklungen der letzten Jahrzehnte etwas entschärft. Gordon Moore geht davon aus, dass sich die Zeiträume zwischen der tatsächlichen Verdoppelung der Integrationsdichte künftig verlängern werden [Moor03]. Er hat in dieser neuen Arbeit die Entwicklungen bis ins Jahr 2002 berücksichtigt.

Ein weiterer Versuch, die Vielzahl an technischen Entwicklungen und die Durchdringung mit und Vernetzung von leistungsfähiger Rechnertechnik in eine gemeinsame Begrifflichkeit überzuführen, wurde auf der Hannovermesse 2011 von [KaLW11] mit der Initiative „Industrie 4.0“ eingeführt. Der Begriff „4.0“ soll auf die derzeit laufende, vierte industrielle Revolution hinweisen. Eine industrielle Revolution wird nach [Dude16] allgemein als „den durch wissenschaftlichen Fortschritt und technische Entwicklung ausgelösten schnellen Wechsel der Produktionstechniken und die damit verbundenen Veränderungen in der Gesellschaft“ bezeichneten Vorgang charakterisiert.

[2] Autor: shigeru23, Lizenz: Creative Commons Attribution-Share Alike 3.0 Unported

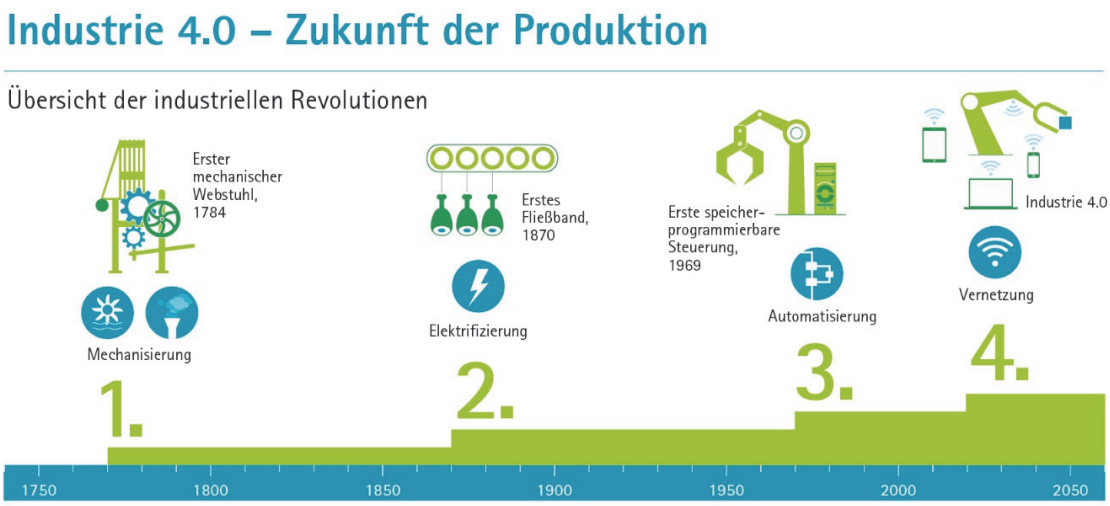

Abbildung 8: Industrie 4.0 – Zukunft der Produktion [Bmvi14]

Die Mechanisierung wird als erste industrielle Revolution bezeichnet. Sie ermöglichte mit der Entwicklung der Dampfmaschine eine verbesserte Versorgung der Bevölkerung und führte nachfolgend zu einem explosionshaften Anstieg der Bevölkerungszahl. Die Elektrifizierung im 19. Jahrhundert wurde als zweite industrielle Revolution geprägt durch die damit mögliche, dezentrale Versorgung der Maschinen mit Energie. Die dritte industrielle Revolution, auch digitale Revolution genannt wurde gekennzeichnet durch neue Entwicklungen im Bereich der Elektronik und Digitalisierung. Die ausgerufene vierte industrielle Revolution („Industrie 4.0“) schließlich wird bestimmt durch den breiten Einsatz von cyber-physikalischen Systemen und deren Vernetzung. [Baue14] ortet unter anderem große Kostenpotentiale und einen Paradigmenwechsel hin zu neuen, vernetzten Produktionssystemen und weitreichende Auswirkungen auf die Wertschöpfungskette. Abbildung 8 gibt einen Überblick des zeitlichen Verlaufs.

Als integrativer Bestandteil der vorliegenden Arbeit und für die Umsetzung ausgewählter Aspekte der Automatisierungstechnik wurde am Institut für Wassergüte, Ressourcenmanagement und Abfallwirtschaft der TU Wien die Messnetzplattform *i*^{TUW}*mon*, *intelligent information monitoring networks* ersonnen und vom Autor dieser Arbeit ab Mitte 2009 in der Programmiersprache *LabVIEW* (engl. für *Laboratory Virtual Instrumentation Engineering Workbench*) von National Instruments (Grundkonzepte siehe [EVZH07]) grundlegend (weiter-)entwickelt. Dieses Werkzeug wird für die in dieser Arbeit beschriebenen Datenerfassung eingesetzt und für neu zu integrierende Sensorik entsprechend erweitert. Der eigentliche Einsatzbereich von *i*^{TUW}*mon* umfasst unter anderem die automatisierte Bestimmung der Gewässergüte im Rahmen diverser Forschungsprojekte, Pilotanlagen und Versuchsaufbauten. Das Konzept wurde in den letzten Jahren stetig erweitert und an neue Herausforderungen angepasst. Die Motivation zur Entwicklung und ausgewählte Betriebserfahrungen mit diesem Werkzeug werden in den Abschnitten 2.1 und 2.2 dargestellt. Auf dieser Datenverarbeitung aufbauend wird in Abschnitt 4.1 der Forschungsschwerpunkt Environmental Monitoring, kurz als *EM* bezeichnet, an der Messstation am Fluss Raab vorgestellt und in Abschnitt 4.2 schließlich die Herausforderungen und Erfahrungen der praktischen Implementierung gezeigt. Im direkt folgenden Abschnitt werden die Grundlagen des maschinellen Lernens präsentiert.

1.3 Maschinelles Lernen: Abgrenzung und Einordnung

Der britische Mathematiker Alan Turing (* 23. Juni 1912, † 7. Juli 1954) gilt vielen als einer der Wegbereiter der Informationstechnologie und Computertechnik. Mit der nach ihm benannten *Turing Maschine* als Rechnermodell konnte nachgewiesen werden, dass (vereinfacht beschrieben) ein in einem Algorithmus umsetzbares Problem auf einer Maschine gelöst, also universell berechnet werden kann, es wird *berechenbar* [Turi36]. Sind Prozesse algorithmisierbar, können diese auf einer Maschine ausgeführt, berechnet und damit letztendlich gelöst werden. Nun stellt sich die Frage, ob Lernprozesse algorithmisierbar sind und was unter *Lernen* überhaupt verstanden werden kann. Für den Begriff Lernen gibt es eine große Zahl gängiger Definitionen. Eine Definition, auf dem Aspekt von Wiederholung einer Aufgabe beruhend, führt Herbert Simon 1983 [Simo83] an:

> „Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the same task or tasks drawn from the same population more efficiently and more effectively the next time."

Eine formale Definition, schon den Begriff *Computerprogramme* verwendend, schlägt Tom Mitchell 1997 [Mitc97] vor:

> „A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P*, if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."

Beide Definitionen beziehen sich auf die Wirkung einer gewissen Erfahrung auf das Ergebnis bei der Durchführung einer Aufgabe, wenn diese einem System mehr als einmal gestellt wird. *Maschinelles Lernen* (engl. *machine learning*) als Teilgebiet der *künstlichen Intelligenz* (*AI*, engl. *Artificial Intelligence*) ist ein Oberbegriff für eine Reihe mathematischer Verfahren zur Generierung von Information durch Erfahrung. Explizit nicht gemeint ist im Rahmen dieser Arbeit der (Forschungs-)Bereich sich selbst verändernder Programme beziehungsweise die Generierung von Technologien, die eine Maschine nach außen wie eine natürliche, menschliche Intelligenz wirken lassen sollen; ganz abgesehen davon, dass unter dem Begriff *Intelligenz* mangels allgemein anerkannter Definition eine große Bandbreite an Begriffen verstanden wird und eine Abgrenzung, inwieweit eine Maschine nun als intelligent zu bezeichnen wäre, entsprechend schwierig zu gestalten ist. Eine Einführung in den Bereich der künstlichen Intelligenz, über das Fachgebiet der theoretischen Informatik hinausgehend, bietet [RuNK12].

Mustererkennung, als eine Anwendung von maschinellem Lernen, ist in der Mitte der Gesellschaft angekommen, nicht zuletzt getrieben durch den weiterhin exponentiellen Anstieg der verfügbaren Rechenleistung [Moor03]. Gesichtserkennung, die automatische Verschlagwortung von Fotos und Übersetzung von in Bildern erkanntem Text in andere Sprachen sind auf breiter Basis und zum Teil auch lokal „errechnet" auf Mobilgeräten verfügbar (siehe Blog-Eintrag [Kamp16] und Meldung in einem Magazin [Brou16]). Digitale Assistenten, die über Spracheingabe bedient werden und die Ergebnisse per Sprachausgabe mitteilen sind mittlerweile Teil des Lieferumfangs von beinahe jedem Smartphone. Der Bereich der autonomen Fahrzeuge ist ein auch in der Praxis umgesetztes Anwendungsfeld von

maschinellem Lernen, neue Fragenstellungen im Bereich der Interaktion zwischen Menschen und einer großen Zahl autonomer Fahrzeuge sind dabei zu lösen [Brow17]. Beispiele für digitale Assistenten auf Basis von Spracherkennung sind Apples *Siri*, Microsofts *Cortana*, Amazons *Alexa* und Googles Assistent ohne Namen, der mit dem Schlagwort „OK, Google“ aktiviert werden kann; für weitere Informationen zur Anwendung sei auf die Seiten der jeweiligen Hersteller verwiesen. Für die aktuellste, im Einsatz befindliche Technologie im Bereich der Unterhaltungselektronik ist nur wenig wissenschaftliche Literatur allgemein verfügbar, einen möglichen Einstieg in diesen Bereich bietet [SPTS16]. Nur am Rande erwähnt seien hier die potentiell großen und aus heutiger Sicht nicht absehbaren Auswirkungen der automatisierten Algorithmik auf die Privatsphäre des Einzelnen und in weiterer Folge auf die gesamte Gesellschaftsstruktur [GeGP16]; *Daten* sind zum neuen Gold geworden.

Ein Grundbegriff des maschinellen Lernens ist der sogenannte *rationale Agent*, ein Computerprogramm, das durch sein Verhalten das bestmögliche Ergebnis erzielt. *Bestmöglich* deshalb, weil die perfekte Rationalität in komplexen Umgebungen nicht erreichbar ist [RuNK12]. Diese Computerprogramme bekommen über Sensoren ein bestimmtes, abstrahiertes Bild der Umgebung und versuchen nun, auf Basis von Feedback und vordefinierten Lernzielen, Verbesserungen zu erzielen und über das sogenannte Leistungselement mit Aktoren wiederum auf die Umgebung zu wirken. Der Vorgang des Lernens kann in diesem Modell als Interaktion zwischen Lernelement, welches den Lernvorgang steuert und bewertet und dem Leistungselement, welches durch den Vorgang des Lernens verbessert werden soll, gesehen werden. Die Verknüpfungen sind in Abbildung 9 dargestellt.

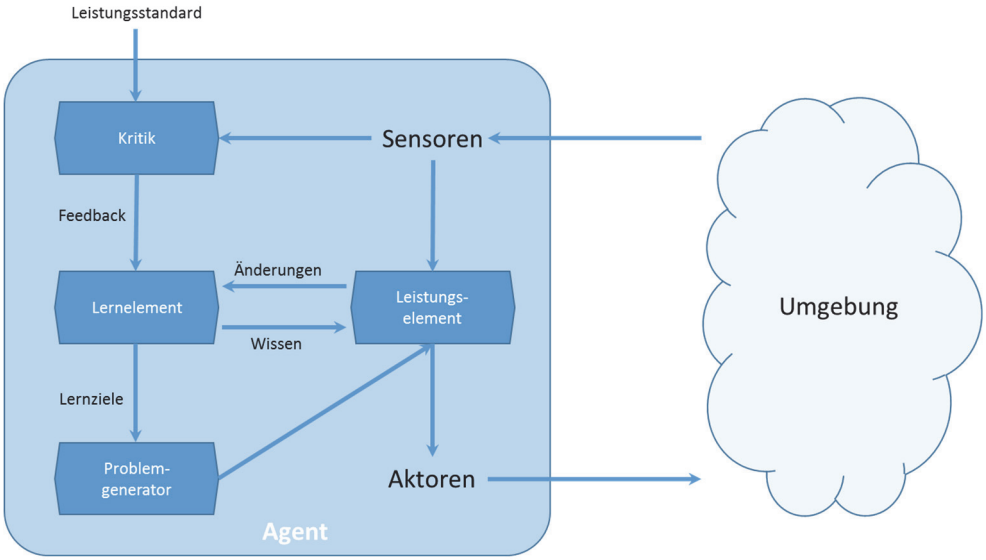

Abbildung 9: Lernender Agent, eigene Darstellung nach [RuNK12, S.83]

Maschinelles Lernen als Vorgang kann in drei grundsätzliche Konzepte eingeteilt werden. Beim *überwachten Lernen* (engl. *supervised learning*) stehen dem Agenten eine gewisse Zahl bekannter Eingabe-Ausgabe-Wertepaare zur Verfügung und das Ziel des Lernvorgangs ist das Finden eines funktionellen Zusammenhangs zwischen diesen Werten; der Agent hat in diesem Fall die Möglichkeit, im Nachhinein die tatsächliche Ausgabe mit der vom Leistungselement für spezifische Eingangswerte

vorhergesagten Ausgabe zu vergleichen. *Nicht überwachtes Lernen* (engl. *unsupervised learning*) bezeichnet jenen Vorgang, bei dem der Agent ein Muster selbstständig erlernt, ohne explizite Vorgabe und daher rein auf dem Datenstrom basierend. Beim *verstärkenden Lernen* (engl. *reinforcement learning*) erhält der Agent eine Bewertung der Aktion ohne Hinweis, wie diese beim nächsten Durchgang zu verbessern wäre. Im Weiteren wird unterschieden nach der Art der Ausgabe: Gehört die Ausgabe des Agenten einer endlichen Wertemenge an, wie zum Beispiel hell/dunkel, wahr/falsch, wird von einer *Klassifizierung* gesprochen; ist die Ausgabe eine Zahl aus einem beliebigen, kontinuierlichen Wertebereich, wird von *Regression* gesprochen [RuNK12].

Eine Auswahl von für die Modellierung der Gewässergüte relevanter Algorithmen des maschinellen Lernens wird in Kapitel 2.3 vorgestellt und deren Eignung für die praktische Umsetzung herausgearbeitet. In Kapitel 4.3 schließlich werden ausgewählte Algorithmen mit Langzeit-Messreihen beschickt und deren Eignung als Vorhersagemodell durch Vergleich mit herkömmlich erfassten Messdaten abgeschätzt.

2. State of the Art und verwandte Arbeiten

Das Ziel der folgenden Ausführungen ist die Darstellung von verwandten Arbeiten und angewandter Methodik auf dem Gebiet der Erfassung der Gewässergüte. Abschnitt 2.1 zeigt den typischen Aufbau einer Messstation zur Erfassung der Gewässergüte. Etablierte Gewässergütesensoren und deren Messprinzipien werden ebenfalls erläutert. Die Konzepte hinter den Wassergütemessstationen, wie sie zur Klärung von Fragestellungen aus der Forschung am Institut für Wassergüte, Ressourcenmanagement und Abfallwirtschaft im Einsatz sind, sowie die Entwicklung von $i^{TUW}mon$ sind in Abschnitt 2.2 vorgestellt. Die Ableitung von Messparametern anhand mathematischer Modellierungen, angewendet beispielsweise auf das Absorptionsspektrum einer Gewässerprobe, wird in Abschnitt 2.3 dargelegt. Der in dieser Arbeit verfolgte Ansatz der Generierung von Information durch Methoden des maschinellen Lernens liegt darin begründet.

2.1 Wassergütebestimmung: Sonden und Sensoren

Wassergütemessnetze, vor allem unter dem Gesichtspunkt des Einsatzes sämtlicher verfügbarer Sensorik unterschiedlicher Hersteller, sind im Vergleich zu Luftgütemessnetzen bisher wenig verbreitet; die gesamte Bandbreite relevanter Messparameter hinreichend präzise erfassende „Messstationen" sind industriell noch nicht verfügbar. Beginnend mit einfachen Datenloggern, die eine bestimmte Anzahl analoger Eingangssignale verarbeiten, sind Geräte bis hin zu sogenannten Multiparameter-Controllern etabliert, welche eine bestimmte Anzahl von Sonden eines Herstellers für die Datenerfassung kombinieren können und zum Teil auch Datenverarbeitung direkt am Gerät erlauben.

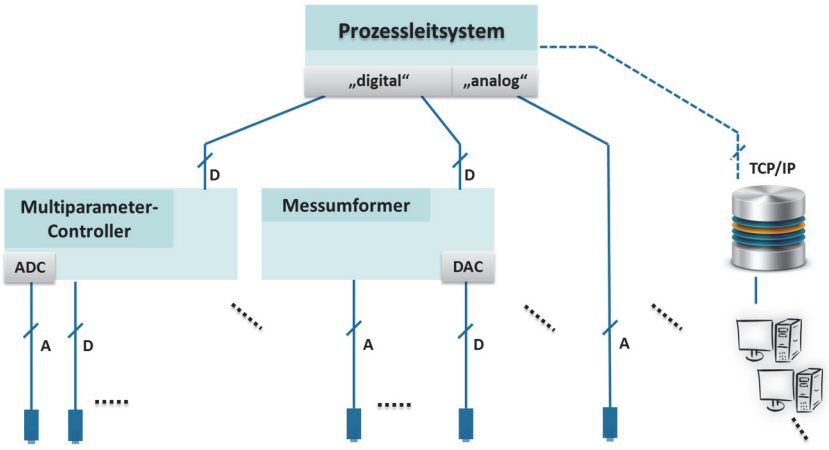

Abbildung 10: Aufbau Datenerfassungs- und Verarbeitungssystem [Wink17]

Der typische Aufbau eines Datenerfassungs- und Verarbeitungssystems in der Gewässergütemesstechnik mit deutlich erkennbarer Heterogenität durch Einsatz verschiedener Systeme ist in Abbildung 10 dargestellt. Eine Reihe von Sonden sind durch analoge oder digitale Schnittstellen an Umformer bzw. Multiparameter-Controller angebunden. In Kläranlagen übernimmt das Prozessleitsystem als übergeordnetes Gewerk die Datenaggregation und -weiterverarbeitung, sowie die Ausgabe der Stellgrößen der ebenfalls implementierten Regelungen.

Große Herausforderungen bei der Integration von Sensorik verschiedener Hersteller sind im Bereich der Standardisierung der Datenschnittstellen und der Datenformate zu finden [WFKW14]. Proprietäre Lösungen oder auch spezielle Anpassungen vorhandener Protokolle in Richtung eigener Dialekte erschweren die Interoperabilität von Gerätschaften verschiedener Hersteller. Oftmals ist die Anbindung der Sonden über klassische Spannungs- oder Stromschnittstellen das einzige zur Verfügung stehende Mittel (siehe dazu die Normen [Dini80, Dini85]). Einerseits ist dabei die einfache Einbindung vorteilhaft, andererseits wird neben dem Messwert selbst keine Information, zum Beispiel in Form von hilfreichen Metadaten über die aktuelle Kalibration und den Wartungsbedarf, übertragen. [FuWW13] gibt einen ersten Einblick zu den Anforderungen an die Datenintegration. Eine weitere Einschränkung einfach aufgebauter Datenlogger betrifft den zeitlichen Charakter der Datenerfassung: Zeitsynchronisation nach [Ietf10] ist mangels Vernetzung der Steuerungen oft nicht vorgesehen und an unterschiedlichen Stellen im Einzugsgebiet aufgenommene Daten damit nur bedingt vergleichbar. Außerdem ist beispielsweise im Falle eines Regenereignisses die gewünschte automatische Verdichtung des Messtaktes mangels Einbindung externer Datenquellen nicht möglich. Die von Gewässer-Trübungsmessungen abhängige Auslösung eines automatisierten Probenehmers ist ebenfalls selten realisiert.

Das Ziel einer Gewässergütemessstation ist die Maximierung der Verfügbarkeit plausibler Messdaten, um durch den Betrieb einer Datenerfassung am Ende Information über die Verhältnisse der Gewässer zu erhalten. Die Entwicklung führt von reinen Datensammlern mit zeitverzögerter, großteils manueller Datenprüfung ausgehend, in Richtung automatischer Messstationen mit integrierter Datenplausibilisierung. Damit ist Alarmierung und zeitnaher Reaktion auf technische Probleme im Fehlerfall möglich, reagieren statt agieren ist die Devise. Der wirtschaftliche Aspekt der Minimierung von Wartungseinsätzen beziehungsweise einer Maximierung der zeitlichen Abstände dazwischen ist ebenfalls Ziel einer Gewässergütemessstation. Durch den permanenten Kontakt der Sensorik mit dem nassen Messmedium und in der Folge auftretenden Ablagerungen beziehungsweise dem Verbrauch von Messreagenzien durch die Messung selbst ist die Datenerfassung in Gewässergütemessstationen prinzipiell betreuungsintensiv.

Neben der Erfassung des Gewässerzustandes mittels geeigneter Sensorik sind die Aspekte der Datenintegration von entscheidender Bedeutung. Die Messwerte werden um Informationen zur Kalibrierung, Laborwerten von einzelnen Stichproben und der in einem Wartungsprotokoll und einem Anlagenbuch festgehaltenen „Geschichte“ zum Betrieb der Station ergänzt. Die Verfügbarkeit von *SOPs* (engl. *Standard Operating Procedures*) zu Kalibration und Justierung sowie von Datenblättern zu den jeweils eingesetzten Gerätschaften auf der Station sind wichtige Aspekte zum Betrieb der Geräte entsprechend ihrer Vorgaben. Die Vorgaben zur Betriebsdatenhaltung und Berichtlegung für Abwasserreinigungsanlagen ist im Regelblatt 13 des „Österreichischen Wasser- und Abfallwirtschaftsverbands“

definiert [Öwav13], in dem auch Richtlinien zur Datenaggregation und Messhäufigkeit angegeben sind.

In Anlehnung an die Begriffe der DIN-Norm 1319-1 [Din95] ist in Abbildung 11 die Messkette, basierend auf dem Weg eines Messsignals, von der Erfassung bis zur Bereitstellung des Messwertes dargestellt.

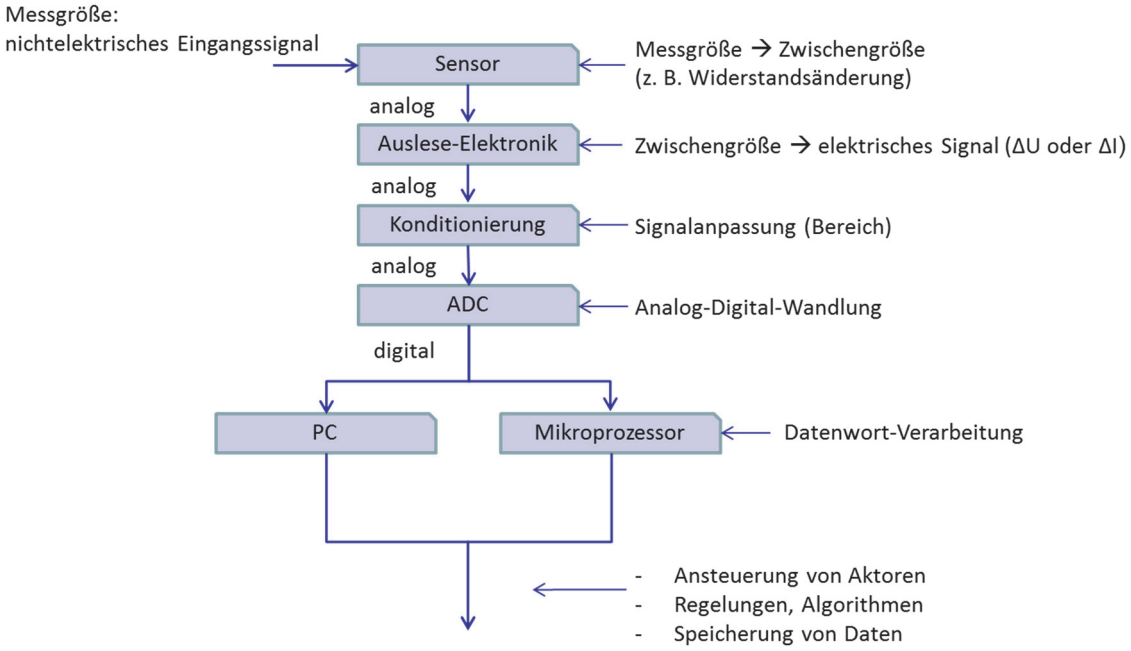

Abbildung 11: Messkette mit digitaler Datenverarbeitung [Wink17]

Die Messgröße wird als wert- und zeitkontinuierliches, nichtelektrisches Signal üblicherweise am Ausgangspunkt der Messkette, im Sensor, in eine elektrische Zwischengröße umgewandelt. In der Gewässergütemesstechnik ist dies beispielsweise eine Widerstandsänderung oder eine sich durch chemisch/physikalische Verhältnisse im Kontakt des Messmediums mit dem Sensor einstellendes, elektrisches Potential. Die Auslese-Elektronik erfasst diese elektrische Zwischengröße und führt diese einer Signalkonditionierung zu. Im Falle einer analog eingebundenen Sonde findet im nächsten Schritt die Umsetzung auf ein normiertes elektrisches Signal (Stromsignal bzw. Spannungssignal nach [Dini80, Dini85]) und anschließende Übertragung in die übergeordnete Datenerfassung statt. Bei digitaler Anbindung der Sonde wird das kontinuierliche Signal einer analog-digital-Wandlung mittels *ADC* (engl. *Analog Digital Converter*) unterzogen, die weitere Datenverarbeitung wird über digitale Datenübertragungssysteme, verkürzt als Datenbus bezeichnet, entsprechend einem definierten Datenübertragungsprotokoll bewerkstelligt. Die oft fälschlicherweise synonym gebrauchten Begriffe „Bus“ und „Protokoll“ bezeichnen verschiedene Aspekte einer Datenübertragung. Während als Datenbus der physikalisch gemeinsame Datenübertragungsweg bezeichnet wird, definieren die Regeln eines Kommunikationsprotokolls den Inhalt und die Bedeutung von Nachrichten.

Die Qualität der Wandlung des zeit- und wertkontinuierlichen Signals in digitale Datenworte ist maßgeblich für die Qualität der Datenerfassung; jedenfalls ist die „Gleichzeitigkeit" der aktuellen Verhältnisse im Messmedium mit der Anzeige der verarbeiteten Messdaten nicht mehr gegeben und muss bei der Interpretation der Daten berücksichtigt werden. Abgesehen davon sind bei der Datenerfassung der Gewässergüte sogenannte Analysatoren gebräuchlich, welche für die Erfassung eines aktuellen Messwertes, aufgrund beispielsweise chemischer Aufschlussverfahren oder Farbumschlags-Reaktionen, eine typisch längere Verzögerung im Bereich von 15 bis 30 Minuten bis zum Vorliegen eines gültigen Messwertes nach dem Aufnehmen einer Wasserprobe aufweisen (Zeitkonstante *T90*, siehe auch [Dine07]).

Nachdem der grundlegende Aufbau und die Messkette der Datenerfassung beschrieben wurde, sind im folgenden Abschnitt ausgewählte Messverfahren und Instrumente zur Erfassung der chemisch/physikalischen Parameter der Gewässergüte vorgestellt. Die Darstellung im Rahmen dieser Arbeit kann ausschließlich einführenden Charakter aufweisen und soll einen ersten Einblick in das große Instrumentarium der messtechnischen Erfassung der Gewässergüte bieten.

Gängige Messparameter und zugehörigen Messverfahren sind in Tabelle 1 angeführt. Je nach erforderlicher Anströmung mit Messmedium, Bauform der Sonde, Zugänglichkeit für Wartungszwecke und der Präzision der Datenerfassung, ergeben sich verschiedene Eignungen in Bezug auf den Messort der eingesetzten Instrumente; die Erfahrungswerte dazu sind ebenfalls in der Tabelle dargestellt. Eine Markierung mit „+" bedeutet „sehr gut geeignet", eine Markierung mit „~" steht für „eventuell problematisch". Ein leeres Feld bedeutet, das Messverfahren ist nicht geeignet beziehungsweise wird nicht am angeführten Messort eingesetzt [Wink11]. Der örtliche Aspekt definiert außerdem ein Unterscheidungsmerkmal zwischen der sogenannten online-Messtechnik, bei der die Instrumente vor Ort, im Gewässer, montiert sind mit quasi-kontinuierlicher Messdatenerfassung und der Messtechnik eines Labors, in dem meist eine zumindest manuelle Probenvorbereitung und eine zeitlich stichprobenartige Charakteristik der Messdaten vorliegt.

Die Messverfahren lassen sich grob in zwei Bereiche unterteilen: *Elektrische* Verfahren (amperometrisch, potentiometrisch) und optische Verfahren. Erstere werden für die Erfassung des pH-Wertes, der elektrischen Leitfähigkeit (induktiv und konduktiv erfasst), des gelösten Sauerstoffs (O_2) und des Redoxpotentials verwendet. Eine Reihe von Nährstoffparametern wie Ammonium-Stickstoff (NH_4-N), Nitrat-Stickstoff (NO_3-N) sowie Chlorid (Cl) können mit sogenannten ISE-Sonden, ionen-sensitive Sonden, erfasst werden (die Grundlagen sind in [HoHo91] beschrieben).

Tabelle 1: Parameter, Messverfahren und geeignete Messorte, eigene Darstellung nach [Wink11]

Parameter			Messort					
Name	Abkürzung	Messverfahren	Kanalnetz	Zulauf	Ablauf, Vorklärung	biologische Stufe	Ablauf	aufnehmd. Gewässer
pH-Wert	pH	potentiometrisch	+	+	+	+	+	+
Leitfähigkeit	EC	induktiv	+	+	+	+	+	
		konduktiv					+	+
Temperatur	T	elektr. Widerstand	+	+	+	+	+	+
gelöster Stauerstoff	O_2	Lum., amperom.				+	+	+
Feststoffe, Trockensubstanz	TS	Streulicht	+	+	+	+	+	+
Ammonium-Stickstoff	NH_4-N	photometrisch		~	~	+	+	+
		gassensitiv		~	~	+	+	+
Ammonium-Stickstoff, Nitrat-Stickstoff, Chlorid	NH_4-N, NO_3-N, Cl	Ionen-sensitiv, ISE	+	+	+	+/~	~	+/~
Nitrat-Stickstoff	NO_3-N	UV-Absorption	+			+	+	+
Phosphat-Phosphor	PO_4-P	photometrisch		~	~	+	+	+
Nitrit-Stickstoff	NO_2-P	photometrisch				+	+	+
chemische und biochemischer Sauerstoffbedarf	CSB, BSB_5	UV-Absorption	+	+	+		+	+

Prominente Gewässerparameter, welche mittels *optischer* Verfahren erfasst werden, sind zum Beispiel der im Messmedium gelöste Sauerstoff (O_2), durch Detektion der Photolumineszenz (Grundlagen dazu siehe [VMGW87]). Die auf diesem Verfahren beruhenden Sonden sind von mehreren Herstellern verfügbar und werden zum Beispiel von der Firma HACH als LDO-Sonden (engl. für *Luminescent Dissolved Oxygen*) bezeichnet. Neben der Erfassung der Trübung und in weiterer Folge der Trockensubstanz durch Messung des Streulichtes (zur Anwendung im Abwasser siehe [JRGB08]), zählt auch der große Bereich der Erfassung von Nährstoffparametern mittels photometrischen Verfahren bzw. der Messung der Absorption der Gewässerprobe in einem bestimmten Wellenlängenbereich zu den optischen Verfahren. Hier sind auch die sogenannten Analysatoren zu finden, welche auf Basis chemischer Analytik und Messung von Farbumschlägen nach Teils komplexer Probenaufbereitung wie Filtration, thermische Behandlung oder Beimengung von direkt vor Ort vorzuhaltenden Reagenzien arbeiten. Neben permanentem Chemikalienverbrauch weisen diese Geräte einen relativ großen zeitlichen Abstand zwischen Probenahme und dem Vorliegen eines Messergebnisses auf [Wink11], erfüllen aber höhere Anforderungen in Bezug auf Genauigkeit und Präzision der erfassten Parameter. Grundlegendes Kriterium für die Sicherung einer qualitativ hochwertigen Datenerfassung mittels automatisierter Messtechnik ist jedenfalls die Qualitätssicherung durch entsprechende Laboranalytik bzw. der regelmäßig wiederkehrenden Kalibration [Wink11].

Zur Orientierung sind in den folgenden Darstellungen beispielhafte Bauformen dreier ausgewählter Hersteller von Sondenmesstechnik angeführt. Abbildung 12 zeigt, von links oben nach rechts unten, jeweils eine Sonde für die konduktive Messung von Leitfähigkeit (*CLS82D*), eine ionensensitive Multiparametersonde für Erfassung von Ammonium und Nitrat (*CAS40D*), eine auf dem Photolumineszenz-Verfahren beruhende Sauerstoff-Sonde (*COS61D*), einen pH-Sensor (*CPS171D*), eine Sonde zur Erfassung des spektralen Absorptionskoeffizienten SAK (*CAS51D*) sowie einen Sensor für die Trübungsmessung (*CUS52D*) der Firma Endress+Hauser GmbH.

Abbildung 12: Sonden, © Endress+Hauser GmbH [Endr17][3]

In Abbildung 13 sind, wieder von links oben nach rechts unten gesehen, eine optische Sauerstoffsonde (*LDO*), ein pH-Sensor, eine Sonde für die Messung der Trübung (*Solitax sc*) sowie ein Sensor zur Erfassung organischer Belastungen (*UVAS plus sc*) der Firma Hach Lange GmbH abgebildet. Als Beispiele für die Analysatoren zeigt Abbildung 14 einen Ammonium-Analysator (*Amtax sc*) und ein Phosphat-Analysator (*Phosphax sc*), ebenfalls von Hach Lange GmbH.

Abbildung 13: Sonden, © Hach Lange GmbH [Hach17][4]

[3] Abbildungen mit freundlicher Genehmigung durch den Hersteller.

[4] Abbildungen mit freundlicher Genehmigung durch den Hersteller.

Abbildung 14: Analysatoren, © Hach Lange GmbH [Hach17][5]

Eine Auswahl verfügbarer Sonden von WTW, Xylem Analytics Germany Sales GmbH & Co. KG, von links nach rechts, zeigt Abbildung 15. Die ersten drei Sensoren dienen zur Bestimmung der Leitfähigkeit (*EC*), gefolgt von einer optischen Sonde zur Sauerstoffmessung (Lumineszenz) und schließlich noch einer Auswahl verschiedener Bauformen von Instrumenten zur Bestimmung des pH-Wertes.

Abbildung 15: Sonden, © WTW, Xylem Analytics Germany Sales GmbH & Co. KG [Xyle17][6]

Ein guter Ausgangspunkt für die Vertiefung mit dem Thema Wasserbeschaffenheit und online-Sensorik bietet die DIN-Norm 15839 [Dine07]; eine Vertiefung des hier nur am Rande erwähnten, jedoch fundamental wichtigen Bereichs der Sonden-Kalibration bietet die ISO-Norm 8466-1, siehe [Iso90].

[5] Abbildungen mit freundlicher Genehmigung durch den Hersteller.

[6] Abbildungen mit freundlicher Genehmigung durch den Hersteller.

Aufbauend auf der Vorstellung einer Reihe typischer Gewässergütesensoren wird im folgenden Abschnitt die Messnetzplattform *i^TUW mon* vorgestellt.

2.2 Die Messnetzplattform i^TUW mon

Einen ersten Einblick in die theoretischen Hintergründe der Designstrategie eines Wassergüte-Monitoringnetzwerks geben [StRo08] und das Standardwerk [SWLS83], die wichtigsten zu erfassenden Wassergüteparameter werden in [Worl08] auf Seite 1.2-29 vorgestellt. Der zunehmende Bedarf im Bereich des online-Monitorings, angetrieben durch sich ähnelnde Fragestellungen in einer Reihe von Projekten am Institut für Wassergüte, Ressourcenmanagement und Abfallwirtschaft der TU Wien, war der Grundstein für die Entwicklung einer wiederverwendbaren Plattform zur Erfassung ausgewählter Parameter der Wassergüte. Die Bezeichnung *i^TUW mon* (aus der ursprünglichen Abkürzung *i2mon* für *intelligent information monitoring networks*;) meint das am Institut konzipierte Werkzeug für das Monitoring. Ursprünglich wurde *i^TUW mon* zur Erfassung der Gewässergüte auf Basis autonom laufender, örtlich verteilter Messstationen verwendet. Diese Plattform dient darüber hinaus zur Unterstützung der Beantwortung aktueller Fragestellungen im Bereich Gewässergüte und zur Prototypenentwicklung für verschiedene Forschungsprojekte am Institut für Wassergüte, Ressourcenmanagement und Abfallwirtschaft. Einen ersten Einblick in den Stand der Entwicklung gibt [WiFW12]. Maßgebliche Triebfeder war die Aufgabe der kontinuierlichen Überwachung des durchschnittlich rund 20 m breiten Flusses Raab im Rahmen der vom BMLFUW, Bundesministerium für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft, finanzierten Projekte „NaWas – Raab, Nachhaltige Wassergütewirtschaft – Online-Monitoring“. Eine Weiterentwicklung der Funktionalität in Richtung Steuerung und Regelung der Abwasserreinigung hat im Rahmen des Projektes „KomOzAk“, der weitergehenden Reinigung kommunaler Abwässer mit Ozon sowie Aktivkohle für die Entfernung organischer Spurenstoffe stattgefunden. Der Endbericht des Projektes ist über die Webseite des BMLFUW zu beziehen [KHKS15]. Die Implementierung des Regelungskonzeptes für die Ozonung wurde auf der DWA MSR-Tagung (von der Deutschen Vereinigung für Wasserwirtschaft, Abwasser und Abfall veranstaltete Mess-, Steuer- und Regelungstechnik-Tagung) in Wiesbaden-Niedernhausen präsentiert [WiSK17]. Im institutseigenen Technikum sind eine Reihe von Versuchskläranlagen und diverse Aufbauten zur Klärung beziehungsweise Erprobung neuer Verfahren im Bereich der Abwasserbehandlung, der Membranfiltrationstechnologie, zur Untersuchung von Antibiotika-Resistenzen und der Entwicklung von Verfahren zur Etablierung granulärer Belebtschlämme aufgebaut. Hier wird *i^TUW mon* als Datenaggregator im Hintergrund, auf Basis virtueller Maschinen als Laufzeitumgebung für die Applikation, eingesetzt. Den Endanwenderinnen und Endanwendern stehen so jederzeit aktuelle, redundant abgelegte und auf Plausibilität untersuchte Messdaten zur Datensichtung und dem anschließenden Export zur Weiterverarbeitung in anderen Programmen zur Verfügung.

Abbildung 16: Schaltschrankaufbau (links) und Messwanne (rechts) (Foto: Autor)

Eine typische Messstation ist in Container-Bauweise ausgeführt. Das Messmedium wird in einer sogenannten Bypass-Anbindung über eine Pumpe und einen Saugkorb in eine Messwanne im Inneren des Containers gefördert, wo auch die Sensorik platziert ist (siehe Abbildung 16 und Abbildung 17). Die Probenförderung durch eine Messwanne wird auf konstanten Durchfluss hin überwacht; die definierte Anströmung mit Messmedium und der eingehauste Aufbau der Sensorik hat sich als sehr vorteilhaft für die Datenerfassung erwiesen (Abbildung 16, rechts). In Abbildung 16, links, ist der Schaltschrank der Raab-Messstation dargestellt. Das Messprogramm läuft auf einem Industrie-PC, diverse Sondencontroller sind ebenfalls erkennbar. Eine typische Messstation mit 35 Datenkanälen liefert durchschnittlich 6,2 Millionen auf Plausibilität geprüfte Messwerte pro Jahr. In [WFKW14] wird gezeigt, dass damit der erhöhte Steuerungsaufwand durch eine durchwegs höhere Datenqualität gerechtfertigt werden kann.

Abbildung 17: Messcontainer (Foto: Autor)

Das Entwicklungsziel von *i*^{TUW}*mon* war die möglichst vollständige Integration von Messsonden unterschiedlicher Hersteller mit dem Ziel der gemeinsamen, zeitgleichen Datenerfassung. Vergleichs-

weise einfach einzubindende, analoge Datenerfassung über Datenkommunikation mit den Sondencontrollern unter Anwendung gängiger digitaler Datenübertragungsprotokolle bis hin zur „Fernsteuerung“ von herstellerspezifischen Programmen wurden entsprechend adaptiert und eingebunden. Neben der reinen Datenerfassung waren auch Herausforderungen im Bereich der Zeitsteuerung von Messungen, regelmäßiger Sondenreinigung und Reinigung der Ansaugvorrichtung mittels Applikation von Druckluft, gegeben. Wesentliche Anforderung aus Anwendersicht war die Verfügbarkeit von Messdaten aller auf einer Station eingesetzten Sonden eines spezifischen Samplingzeitpunktes in einem gemeinsamen Datensatz vereint. Der dafür notwendige Aufwand des händischen Zusammenfügens der Daten unterschiedlicher sogenannter „Datenlogger“, welche nur Daten einzelner Sonden mit jeweils eigener, von den anderen Sensoren unabhängiger, Zeitspalten aufzeichnen konnten, musste entfallen. Die gesammelten Daten sollen schließlich in geeigneter Weise dargestellt und für eine zügige Datensichtung am jeweiligen Computerarbeitsplatz zur Verfügung stehen.

Die Softwarearchitektur von *i*^{TUW}*mon* wird in drei wesentliche Bereiche unterteilt (siehe Abbildung 18): Die *Datenerfassung* (engl. *data acquisiton*), befasst sich mit der Einbindung der unterschiedlichen Sensorik, der Zeitsteuerung und dem verlässlichen Stationsbetrieb. Der Bereich *Information aus Daten* (engl. *data to information*) ist für die Generierung von Information aus den aufgenommenen, noch nicht weiter interpretierten Rohdaten, zuständig; die Datenplausibilisierung ist dabei ein zentraler Aspekt.

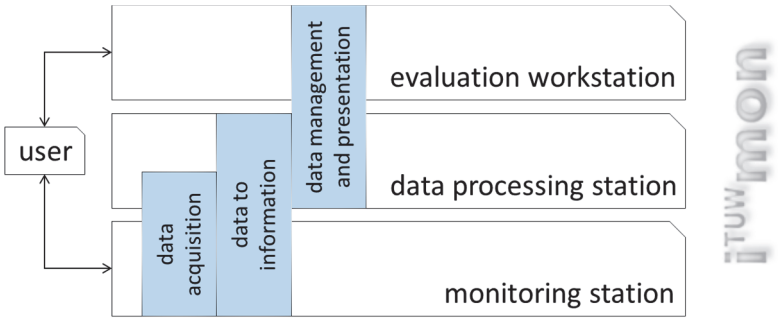

Abbildung 18: Architektur der Monitoringplattform [WFKW14]

Die Aufgaben der gemeinsamen, redundanten und gesicherten Datenhaltung und die Präsentation der Daten in geeignet gestalteten Benutzerschnittstellen (engl. *UIs, user interfaces*) findet schließlich im Bereich *Datenmanagement und Präsentation* (engl. *data management and presentation*), statt. Die örtliche Zuordnung wird in die Schichten *Messstation* (engl. *monitoring station*), *Datenverarbeitungsstation* (engl. *data processing station*) und *Arbeitsplatzcomputer* zur Datensichtung (engl. *evaluation workstation*), eingeteilt. Gemeinsame Basis der Kommunikation der verschiedenen Bereiche untereinander ist ein entsprechend den Erfordernissen „Zeit und Wert und Metadaten“ nach gestaltetes, abstraktes Datenformat (siehe Abbildung 19).

Abbildung 19: Datenstruktur der Messdaten in *i*^{TUW}*mon* [WFKW14]

Ein *Datenkanal* (engl. *data channel*) ist demnach die Basis jeglicher Datenmanipulation in *i*^{TUW}*mon* und jeder Bearbeitungsschritt beruht auf der Manipulation der zugehörigen Bereiche dieses Datenobjekts. Bei der *Datenerfassung* wird entsprechend der Konfiguration der Messstation für jeden Messkanal ein Datenobjekt angelegt; eine physikalisch vorhandene Sonde kann je nach Ausstattung, Daten für mehrere nachgereihte Messkanäle liefern. Direkt nach der Datenerfassung, typischerweise noch auf der Messstation selbst, werden die Messdaten einer Bewertung und Plausibilitätsüberprüfung unterzogen (*Information aus Daten*). Die Basis der Einordnung des aktuellen Messergebnisses bilden einfache statische Überprüfungen wie die Überprüfung von Wertebereich, Minima und Maxima und der Erkennung von „Strichfahren". Zusätzlich werden durch Langzeitmessreihen bekannte Zusammenhänge, auch verschiedener Datenkanäle untereinander, herangezogen (wie zum Beispiel Verdünnungskurven, abhängig beispielsweise vom Volumenstrom des betrachteten Gewässers). Im letzten Schritt der Datenverarbeitung schließlich werden eine Reihe von Datenkanälen zu einem gemeinsamen Datensatz kombiniert. Basis für die Datenkombination sind die jeweiligen Zeitpunkte der Datenerfassung, über sämtliche Messstationen des Einzugsgebietes gerechnet. Die gesammelten Messdaten stehen nun für die Datensichtung und Weiterverarbeitung am Arbeitsplatzcomputer zur Verfügung (*Datenmanagement und Präsentation*).

Ein Datenkanal wird aus einer Reihe von Parametern aufgebaut: *Timestamps* beinhaltet die Zeitstempel der Datenerfassung in verschiedenen Formaten; unter anderem der koordinierten Weltzeit *UTC* (engl. *Coordinated Universal Time*) und der Lokalzeit am Messort, die *MEZ* (*Mitteleuropäische Zeit*) beziehungsweise *MESZ* (*Mitteleuropäische Sommerzeit*). Das Datenfeld der Zeitstempel gilt als primärer Schlüssel zur weiteren Verarbeitung des Datensatzes. Der zum Datenerfassungszeitpunkt aktive Messmodus (engl. *mode*) bezeichnet, mit dem Wertebereich *General*, *Maintenance* oder *Trigger*, bildet den zweiten Eintrag der Datenstruktur. Im Normalbetrieb ist der Modus *General* aktiv; die Daten werden entsprechend den Benutzervorgaben zur Messhäufigkeit erfasst. Im Falle einer Stationswartung ist durch Eingriff in den Ablauf, beispielsweise durch Sondenkalibration, mit teils unerwartetem Verhalten der einzelnen Messkanäle zu rechnen (zum Beispiel Ausreißer oder Strichfahren durch Festhalten des letzten gültigen Messwertes im Rahmen der Kalibrierung). Zum Ausscheiden dieser Daten wird der Erfassungsmodus manuell auf *Maintenance* umgestellt. Besondere Gewässerzustände während beispielsweise einem Regenereignis bedingen den Wunsch nach Änderung der Dichte der Datenerfassungszyklen. Solche Ereignisse zeigen sich typischerweise in sprunghaften Anstiegen der Werte der Trübungsmessung beziehungsweise des Volumenstroms eines Fließgewässers und führen zu einer automatischen Umschaltung in den Messmodus *Trigger* mit einer zeitlich verdichteten Datenerfassung und der konfigurierbaren Ansteuerung von automatischen Probenehmern zur Einbringung von Stichproben als Basis einer weitergehenden Analyse der Gewässerparameter im Labor. Im Datenfeld *SignalSpec* (engl. *signal specification*) sind die den Datenkanal näher definierenden Informationen wie Parametername, Einheit, Messort und die physikalische Datenquelle abgelegt. Analoge Eingänge, diverse serielle Schnittstellen, Profibus [Prof16], ModbusTCP [Modb06a], Spektrometerschnittstellen und externe Signale (als sogenannte „virtuelle Kanäle" eingebundene, typischerweise in der zentralen Datenbank durch Rechenmodelle erstellte Parameter) fungieren als mögliche Datenquellen. *ValueRAW* und *ValueSCAL* bezeichnen jene Felder, in denen die Messwerte als solche abgelegt sind. Die Bezeichnung *RAW* steht für Rohwerte (beispielsweise das Signal in der Einheit mA) und *SCAL* für die umgerechnete, auf die Messbereiche skalierte Größe in der Zieleinheit. Beide Felder

sind als Array, also einem Feld an Datenwerten, implementiert und ermöglichen so die Ablage ganzer Absorptionsspektren in einer gemeinsamen Struktur, wo jedem Eintrag im Array ein entsprechender Absorptionswert des Messmediums an einer spezifischen Wellenlänge entspricht. Information zur Datenplausibilitätsüberprüfung vor Ort werden in den Feldern *PlausibilityDEF* (engl. *definition of plausibility*) zur Ablage der aktuell gültigen Kriterien und *PlausibilityASM* (engl. *assumption of plausibility*) zur Ablage der Ergebnisse der Plausibilitätsprüfung gespeichert. Einen Einstieg in das Thema der Datenplausibilisierung bietet [WKFW13]. Der letzte Eintrag der Datenstruktur, mit *UID* (engl. *universal identifier*) bezeichnet, fungiert als eine Art „Fingerabdruck" des jeweiligen Messkanals zum aktuellen Messzeitpunkt. Diverse Seriennummern, Daten zur Netzwerkumgebung, Information zum aktuellen Datenerfassungszyklus und die Anzahl der Tics des Systemzeitgebers seit einem bestimmten Referenzdatum werden zur Berechnung der *UID* herangezogen. Anhand dieser Information ist eine systemweite, eindeutige Zuordnung jedes Datensatzes gegeben.

Die Hauptaufgabe der als ***Datenerfassung*** bezeichneten Schicht ist die zeitliche Steuerung der Datenerfassung und die Integration verschiedener Sensorschnittstellen in dem beschriebenen, abstrakten Datenformat zur Weiterverarbeitung. Die analoge Einbindung der Sensorik ermöglicht prinzipiell wenig Metainformation über den reinen Messwert hinaus zu übertragen. Bei analogen Stromschleifensignalen (siehe [Dini85]) mit dem Wertebereich 4-20 mA beispielsweise sind neben dem eigentlichen Messwert nur Informationen zu einem Drahtbruch (0 mA) und einer Bereichsüberschreitung (21 mA) darstellbar. Bei digitaler Signalanbindung sind im Gegensatz dazu prinzipiell beliebig umfangreiche Metadaten in Form dem Messwert nachgestellter und vom Sondencontroller entsprechend einem vereinbarten Übertragungsprotokoll implementierter Datenfelder übertragbar. Für den Anwender interessante Metadaten sind zum Beispiel Informationen zu erfassten Rohwerten, etwaige Fehlermeldungen während der Messung und Hinweise auf Wartungsbedarf beziehungsweise dem Reagenzienverbrauch. Abgesehen davon ist digitale Datenübertragung weniger anfällig auf elektrische Störungen und Einflüsse durch elektromagnetische Einwirkung. Bei analoger Datenübertragung wirkt jede Beeinflussung auf die elektrischen Verbindungen wertändernd auf das zu erfassende Signal (zum Beispiel durch Rauschen), bei digitaler Datenübertragung findet ein Vergleich mit Schwellwerten statt. Einflüsse wirken nur dann störend auf die Datenübertragung, wenn durch sie das Signal über oder unter dem beabsichtigten Schwellwert verändert wird, was bei geschickter Definition der Schwellwerte eine relativ störungsfreie Übertragung ermöglichen kann. Aktuelle Messtechnik arbeitet intern meist auf Basis integrierter Mikrocontroller-Schaltungen und so findet die Wandlung der analogen Eingangsgröße sehr nahe am Messort selbst statt; eine digitale Signalanbindung ist in diesem Fall besonders empfohlen, da weitere, Rauschen in den Signalpfad einbringende, analog/digital-Wandlungen entfallen können. Auf Seiten der digitalen Einbindung von Instrumenten ergeben sich neue Herausforderungen in Bezug auf die eingesetzten Protokolle; nach Standards wie ModbusTCP zur Verfügung gestellte Datenverbindungen sind zum Teil mit gewissen herstellerspezifischen Anpassungen versehen, sodass von einer Art „Dialekt" gesprochen werden kann. Die Modbus-Organisation bietet auf ihrem Webauftritt umfangreiche Dokumentationen zu Modbus an; besonders hervorzuheben ist dabei der ModbusTCP-Standard [Modb06a] und das für das Verständnis einer standardkonformen Server-Client-Struktur sehr hilfreiche Dokument der Implementierungsrichtlinien [Modb06b]. Ein Vorteil des ModbusTCP-Protokolls ist die Verwendung von Ethernet beziehungsweise die strukturierte Verkabe-

lung der physikalischen Schicht der Datenkommunikation. Vom Gesichtspunkt des Software Engineerings aus sind während der Implementierung der verschiedenen Schnittstellen die Konzepte von *Code Reuse* und der *Modularisierung*, also der Kapselung von Funktionalität in kleine, gut handhabbare Einheiten, die an verschiedenen Stellen der Implementierung eingesetzt werden können, zur Anwendung gekommen.

Ein weiterer, fundamental wichtiger Aspekt der Datenerfassung ist eine korrekte Zeitsteuerung sämtlicher Aufgaben im Rahmen der Messdatenerfassung. Die Erfassung der Wassergüte bedingt den Kontakt elektrischer Sondenmesstechnik mit feuchtem Messmedium und so muss besonderes Augenmerk auf Sauberhaltung der Messmembranen beziehungsweise regelmäßiger Wartung, sprich der Entfernung etwaiger organischer beziehungsweise anorganischer Ablagerungen optischer Instrumente gelegt werden. Im laufenden Messbetrieb kann einer allzu starken Verschmutzung der Sondenköpfe mit einer regelmäßig ausgelösten, an den Messtakt der Datenerfassung angepassten Reinigung durch gesteuerte (stoßweise) Applikation von Druckluft, entgegengewirkt werden. Im Falle einer Bypass-Lösung, wo das Messmedium durch eine innerhalb des Messcontainers aufgebaute Messwanne gepumpt wird, muss ebenfalls Sorge getragen werden, dass der Saugkorb im Fließgewässer frei von Verklausungen und Verschmutzung gehalten wird. Auch in diesem Fall ist regelmäßige Applikation von Druckluft beziehungsweise eine Überwachung der Volumenströme der Anlage und entsprechende Alarmierung im Fehlerfall, angezeigt. Die Implementierung sieht für diese Aufgabe drei miteinander verzahnte Zeitschleifen vor, deren Terminierung ähnlich dem Algorithmus eines dynamischen Schedulings (vgl. [Kope11, S.251]) implementiert ist. Die *Messung* erhält eine hohe Priorität und wird typisch im Bereich von zehn vollständigen Messzyklen pro Stunde mit einer Dauer von jeweils rund drei Minuten ausgeführt. Analysatoren werden mangels Triggermöglichkeit meist freilaufend betrieben und bei Vorliegen eines neuen Messergebnisses wird dieses beim jeweils nächsten Messzeitpunkt übernommen. Für jeden Messkanal ist entsprechend seiner Ansprechzeit *T90* (siehe [Dine07]) eine passende Konfiguration hinterlegt und die Terminierung sorgt dafür, dass zum gewünschten Messzeitpunkt (mit Ausnahme nicht triggerbarer Analysatoren) alle beteiligten Geräte gleichzeitig einen aktuellen Messwert vorweisen können. Die *Sondenreinigung* findet im Vergleich zur Messdatenerfassung typischerweise etwa drei Mal pro Stunde statt und dauert durchschnittlich eine Minute. Die *Saugkorbreinigung* wiederum soll einmal täglich durchgeführt werden und benötigt für den kompletten Ablauf aus Pumpen-Stopp, Reinigung und dem erneuten Anfahren der Probenförderung unter vorhergehender Evakuierung der Ansaugleitung mittels Venturi-Ventils rund zwölf Minuten. Die gewünschten zeitlichen Abläufe sind vom Benutzer frei konfigurierbar und können sich, abhängig von den Messdaten, auch ändern, um zum Beispiel während eines Regenereignisses die Messzyklen zu verdichten, um umfangreichere Daten des dynamisch interessanten Ereignisses zu erhalten. Die Aufgabe der Terminierung ist nun, die Ressource *Zeit* fair auf alle erforderlichen Aufgaben aufzuteilen. Abbildung 20 zeigt die Verhältnisse bei einem Konflikt von Messung und Reinigung zur Minute Null eines einstündigen Messzyklus mit vier Messzeitpunkten und einer ebenfalls konfigurierten Sondenreinigung zur vollen Stunde.

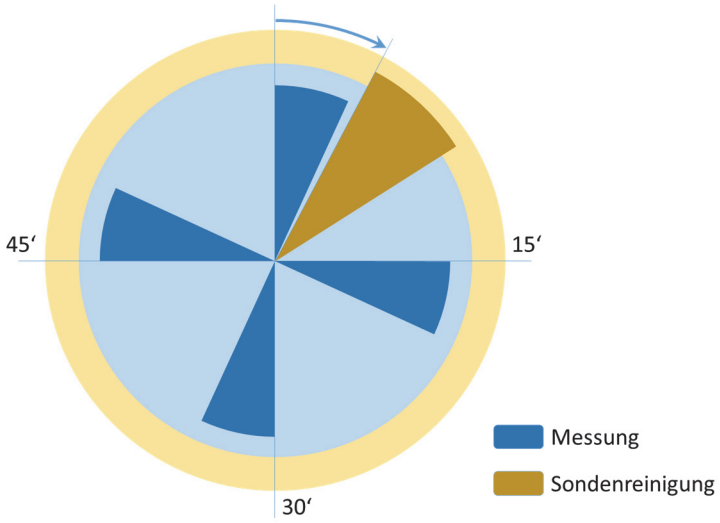

Abbildung 20: Zeitschleifen der Datenerfassung (eigene Darstellung)

Entsprechend der Priorisierung der Aufgaben geht die Messung in diesem Fall vor, die zeitlich unkritische Sondenreinigung wird an das Ende der durchgeführten Messung zur vollen Stunde verschoben. Die Reinigung des Saugkorbes ist eine im Vergleich dazu selten durchgeführte Aufgabe, benötigt aber entsprechenden Vorrang, da sie typischerweise mehr Zeit in Anspruch nimmt als zwischen zwei aufeinanderfolgenden Messzeitpunkten zur Verfügung steht und folglich niemals stattfinden würde. Die Saugkorbreinigung bekommt daher höchste Priorität im System, da auf lange Sicht die Reinigung der Ansaugsituation für die korrekte Datenerfassung fundamental wichtig ist.

Ein weiterer, im Hinblick auf den Einsatz mehrerer Messstationen in einem Einzugsgebiet wichtiger, Aspekt ist die Zeitsynchronisation aller beteiligten Messinstanzen im Messnetz. Die Vergleichbarkeit der erfassten Daten und die Suche nach den auf das ganze Gebiet wirkenden Einflussgrößen (Abflüsse im Regenwetterfall, Ausbreitung von Spitzenwerten der Volumenströme der Oberflächengewässer) ist erst nach der Sicherstellung einer gemeinsamen Zeitbasis möglich. Die Zeitbasen der beteiligten $i^{TUW}mon$-Instanzen werden kontinuierlich überwacht und längstens alle 15 Minuten wird ein automatischer Abgleich mit Zeitservern im Messnetz nach einer Implementierung des NTP-Protokolls (engl. *Network Time Protocol*) durchgeführt [Ietf10]. Diese Methode stellt für die Datenerfassung zur Wassergütebestimmung eine hinreichend präzise Zeitbasis zur Verfügung.

Aus Sicht des Software-Engineerings sei ein besonderer Algorithmus hervorgehoben: die dynamische Instanzierung der Messkanäle in $i^{TUW}mon.Measurement$. Das Konzept des abstrakten Messkanals betrifft nicht nur die Datenstruktur in Datenerfassung und -ablage, sondern auch das für den Messvorgang zuständige Softwarekonstrukt. Das für das Datensampling eines Messkanals zuständige Unterprogramm wird dabei über eine statische Referenz zur Laufzeit instanziiert, mit Konfigurationsparametern beschrieben und läuft, bei n konfigurierten Messkanälen, in n-facher Instanz im Speicher. Die Messergebnisse werden in ein Messdatenarray, abgelegt in einer globalen Variable, zurückgeschrieben. Semaphoren verhindern dabei Konflikte durch gleichzeitiges Beschreiben der gleichen Zieldatenstruktur (vgl. [Stal07, S.219]).

Nach der Datenerfassung und Einbringung sämtlicher relevanter Information der Messung in die Datenstruktur des Messkanals findet der erste Schritt der nachgereihten Datenverarbeitung statt.

Die größte Herausforderung beim Betrieb automatisierter Wassergütemessungen ist die Generierung von ***Information aus*** den vorliegenden ***Daten***. Die Unterscheidung relevanter von nicht plausiblen Messdaten wird anhand einer statistischen Bewertung des Signals im Vergleich zum Verlauf in der näheren Vergangenheit getroffen. Einfache Tests auf die Einhaltung des zulässigen Wertebereiches durch Setzen von Minimal- und Maximalwerten, die Überwachung des zeitlichen Verlaufs auf maximal erlaubte, beziehungsweise minimal notwendige, Wertänderung von Messsignalen zwischen zweier, aufeinanderfolgender Messzeitpunkte sind vergleichsweise einfach zu implementieren. Größere Herausforderungen treten bei Fragestellungen betreffend die wechselseitigen Abhängigkeiten zwischen verschiedenen Messkanälen und deren zeitlichem Verhalten auf. In [WKFW13] werden mögliche Zusammenhänge zwischen gelöstem Sauerstoff und der elektrischen Leitfähigkeit des Messmediums mit dem Ziel der Unterscheidung zwischen Systemdynamik und dem Vorliegen eines Gerätefehlers untersucht. Weitere interessante Kandidaten zur verschränkten Beobachtung sind die elektrische Leitfähigkeit und die Chlorid-Konzentration (siehe [FuWi15] bzw. Abschnitt 2.3). Allen Ansätzen gemeinsam ist das Bestreben, unplausible Daten zeitlich direkt nach der Datenerfassung auszumachen, um einen korrigierenden Eingriff im System durchführen zu können und nicht erst im Nachhinein zum Teil größere Datensammlungen als ungültig markieren zu müssen. Das Ergebnis der Plausibilitätsbewertung schließlich wird auf der Benutzerschnittstelle mittels farblich codierten Feldern (rot, gelb, grün) zur Darstellung gebracht; Details dazu sind im folgenden Abschnitt dargestellt.

Die Anwendung von Konzepten des maschinellen Lernens im Bereich der Wassergütebestimmung fußt auf diesen grundsätzlichen Überlegungen: Neben der Erfassung rein der Wassergüte zuzurechnender Messkanäle könnten auch weitere, operationale Parameter wie (Außen-)Temperaturen, der Luftdruck, die Sonneneinstrahlung oder Ähnliches, einen Einfluss auf die Qualität und Plausibilität der erfassten Zusammenhänge haben und diese Zusammenhänge wären durch maschinelles Lernen zu finden. Die Erkenntnisse des maschinellen Lernens könnten einerseits zur Einführung eines Surrogatparameters als Ersatz einer bestehenden Sonde führen und andererseits als eine verbesserte Absicherung mittels industrieller Messtechnik herkömmlich erfasster Messdaten dienen. Mit einer Reihe einfach und kostengünstig zu erfassender Parameter könnte auf den Sauerstoffgehalt eines Gewässers geschlossen werden, ohne, abgesehen von der für die Algorithmen notwendigen Datenerfassung zum Erlernen der Zusammenhänge, tatsächlich eine Sauerstoffsonde im Dauerbetrieb einsetzen zu müssen. Aller Erfahrung nach sind die ermittelten Zusammenhänge bezüglich der Dynamik und auch des Wertebereichs der Parameter ausschließlich in einem stationsspezifischen Kontext zu sehen, die direkte Übertragbarkeit der Modelle auf andere Gewässer ohne erneute Lernphasen wird typischerweise kaum gegeben sein (zur Übertragbarkeit von Modellen siehe Endbericht des Projektes „IMW3“, innovative Messtechnik in der Wassergütewirtschaft [CEFG13]).

Der Bereich ***Datenmanagement und Präsentation*** behandelt die Datenhaltung, die konsistente Kombination verschiedener Datenquellen im Messnetz, die Präsentation beziehungsweise den Export der auf Plausibilität hin geprüften Datensätze und schließlich Routinen für die automatische Datenpublikation. Basis der Datenverarbeitung ist wiederum die bereits vorgestellte, abstrakte Datenstruktur. Neben einer langfristig stabilen, redundanten Datenhaltung auf den Stationen beziehungsweise einem

zentralen Messdatenserver war auch die Entwicklung intuitiver Benutzerschnittstellen eine wesentliche Zielsetzung. Die Messdaten werden nach der Erfassung in die zentrale Datenbank eingebracht; für die Realisierung ist das Open Source Enterprise-Datenbankmanagementsystem PostgreSQL in Anwendung [Post17a]. Für dieses System sprechende Argumente sind, neben einer vom Start weg sehr guten Leistungsfähigkeit ohne die Notwendigkeit des Einsatzes komplexer Tuning-Mechanismen, eine sehr gute Unterstützung durch eine engagierte Online-Community. Für die Datenübertragung und den Fernzugriff auf die Stationen wurde ein virtuelles privates Netzwerk, ein *VPN* (engl. *Virtual Private Network*), auf Basis industrieller *3G*-Router der Firma Westermo (MRD-330, [West17]) und an der Gegenseite ein *VPN*-Konzentrator der Firma Fortinet (Fortigate 60c, [Fort17]) gewählt.

Die einfachste Form der Datenhaltung basiert auf der direkten Ablage in einem binären Datenformat auf einem geeigneten Datenträger, welcher zum Auslesen und der Weiterverarbeitung der Daten am Ende einer Messperiode vom Messort zum Arbeitsplatzcomputer gebracht wird. Die Dateninterpretation und Behebung von Störeinflüssen wird hierbei meist mit großer zeitlicher Verzögerung oder überhaupt erst am Ende einer Messkampagne durchgeführt. Für langfristige Datenablage und Verfügbarkeit nahe Echtzeit ist diese Methode nicht geeignet und daher die Ablage der Daten in Datenbanksysteme zielführender. Datenbanken können nach ihrer Zugehörigkeit zu zwei Modellen unterschieden werden, dem *relationalen* und dem *nicht-relationalen* Datenbankmodell. Kennzeichen eines relationalen Datenmodells ist die Frage nach dem Verhältnis, der Beziehung, von Daten zueinander. Die *Records* (Datensätze) mit ihren *Attributes* (Eigenschaften, in Spalten abgelegt) werden in *Relations* (Tabellen) abgelegt (nach [Codd70]). Die Daten werden über relationale Algebra verknüpft und unter Einsatz einer strukturierten Sprache, *SQL* (engl. *Structured Query Language*) verarbeitet beziehungsweise abgefragt (siehe [Isoi04]). Das Transaktionskonzept ist eine notwendige Bedingung für die Erfüllung der Forderung nach Datensicherheit und führt zu sogenannten *ACID*-Kriterien, die eine relationale Datenbank erfüllen muss (nach [HaRe83]): *Atomicity* (Unteilbarkeit) einer Transaktion („ganz oder gar nicht“), die *Consistency* (Konsistenz) der Datenbankintegrität wird durch eine Transaktion nicht verletzt, *Isolation* zwischen verschiedenen Transaktionen im Sinne einer nicht-Beeinflussung und *Durability* (Dauerhaftigkeit) einer einmal korrekt abgelaufenen Transaktion. Relationale Datenbanken eignen sich für große, strukturiert vorliegende Datenmengen. Extrem umfangreiche Datenmengen, auch mit strukturellen Änderungen behaftet, werden zunehmend in Datenbanken nach dem nicht-relationalen Modell abgelegt. Ein Beispiel dafür ist *NoSQL* (engl. *not only SQL*), beschrieben von [Poko13]. Aufgeweichte *ACID*-Kriterien, der Verzicht auf *SQL* und das typische Ablagekonzept *Key-Value* (Schlüssel-Wert) sind bei den Big-Data-Datenbankimplementierungen *MongoDB* beziehungsweise *CouchDB* zu finden [SrGK15] und werden hier nicht weiter verfolgt.

Im nun folgenden Teil dieses Abschnitts werden die Benutzerschnittstellen der wesentlichen Programmteile vorgestellt (Datenpräsentation). Wertvolle Anregungen und die Grundprinzipien zur Gestaltung von Benutzerschnittstellen bietet [Zühl12, S.184]. Abbildung 21 zeigt die am Messstationsrechner laufende Instanz des Messprogramms *i*^{TUW}*mon.Measurement*. Der mit „1“ bezeichnete Bereich weist unter anderem auf den aktuell laufenden Messmodus (Eintrag „*Mode*“), die letzte Aktualisierung der (quasi-kontinuierlichen) Werte der Messkanäle (Eintrag „*live*“) und den letzten Messzeitpunkt im aktuellen Messprogramm (Eintrag „*sampled*“) hin. Die Messkanäle, mit „2“ bezeichnet, werden in Listenform dargestellt. Die mit „3“ bezeichnete Spalte gibt den Messwert zum letzten

Sampling-Zeitpunkt wieder, im mit „4“ bezeichneten Bereich werden die laufend aktualisierten Livewerte dargestellt.

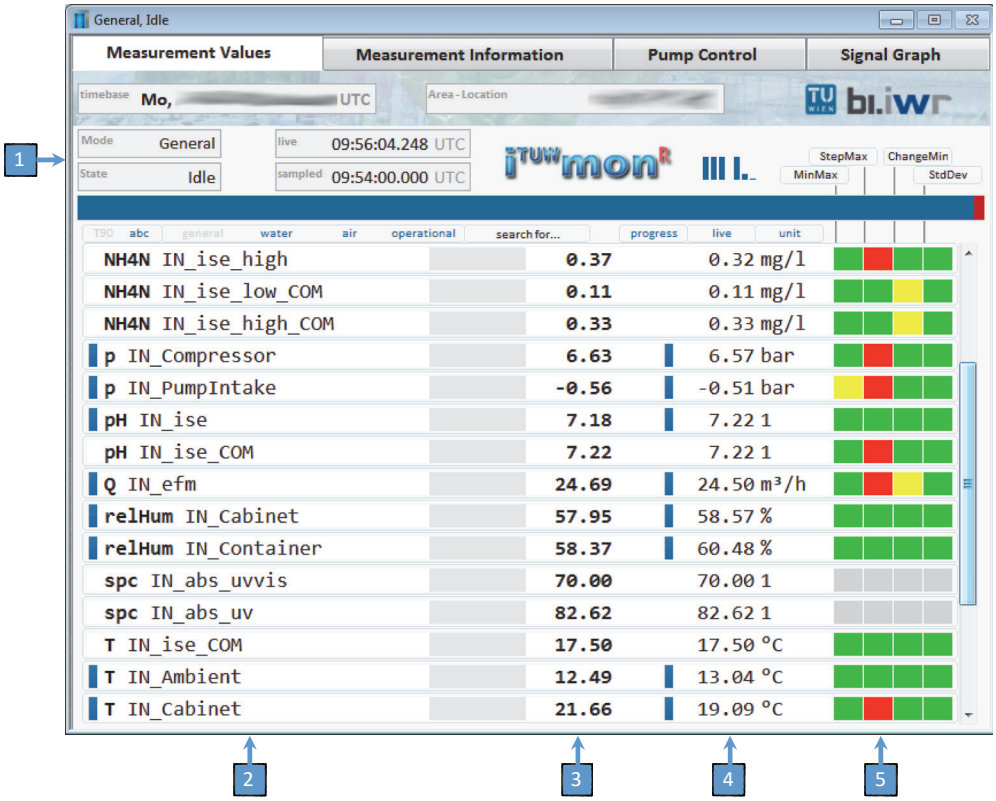

Abbildung 21: *i*^{TUW}*mon.Measurement* [Wink16] (adaptiert)

Bei einem Messvorgang werden, je nach spezifischer *T90*-Zeit, die Daten der Live-Spalte in die Sampled-Spalte übertragen. Dieser Vorgang wird durch eine Animation mit Pfeilen und wachsendem Balken entsprechend der seit Messstart abgelaufenen Zeit dargestellt (Screenshot in Abbildung 22). Ebenfalls in der Abbildung erkennbar sind sogenannte *virtuelle Kanäle*; Datenkanäle, die nicht auf der Station selbst erfasst werden, sondern über externe Schnittstellen beziehungsweise als äquivalent modellierte Parameter (siehe Abschnitt 2.3) in das Messnetz eingebracht werden. *Q_ps_hydro* bezeichnet den, durch einen Übertragungsfehler in der Darstellung als „outdated“, also abgelaufen, markierten, extern erfassten Volumenstrom des Flusses Raab im südlichen Burgenland. *SAC_5mm* und *SAC_15mm* (engl. *spectral absorption coefficient*) bezeichnen die, aus einem kompletten Absorptionsspektrum erfassten und in der Datenbank durch Modellierung berechneten, spektralen Absorptionskoeffizienten bei einer Wellenlänge 254 nm.

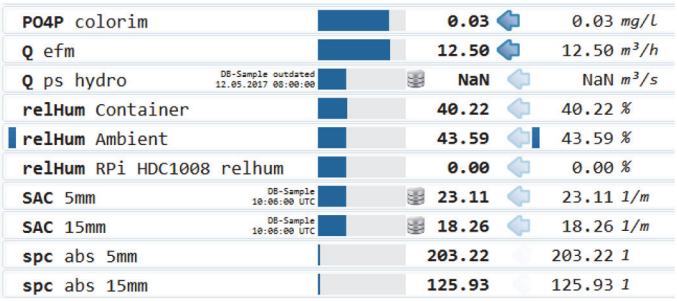

Abbildung 22: Darstellung eines Messvorgangs (eigene Darstellung)

Das Ergebnis der Datenplausibilitätsprüfung wird in Abbildung 21 im mit „5“ bezeichneten Bereich dargestellt. Die Farben sind entsprechend einer Ampel-Logik gestaltet und sollen dem Anwender einen ersten Eindruck zur Plausibilität der Daten zum letzten gemeinsam erfassten Messzeitpunkt geben. *Grün* bedeutet in diesem Fall „Signal plausibel“, *gelb* lässt auf beginnende Probleme der Datenerfassung schließen und *rot* markiert unplausibles Verhalten, jeweils bezogen auf das in der Kopfzeile ausgewiesene Testkriterium. Zum Beispiel weist das in Abbildung 21 angeführte Signal *Q_IN_efm*, der Durchfluss der automatischen Probeförderung in die Messwanne der Station, eine sowohl dem sonstigen Signalverlauf nach unübliche, minimale Signaländerung, durch ein gelbes Feld im Bereich *ChangeMin* und zusätzlich dazu ein stark sprunghaftes Verhalten, gekennzeichnet durch ein rotes Feld im Bereich *StepMax*, auf. Auslöser für so ein Signalverhalten könnte eine kurzfristige Verstopfung des Saugkorbes sein, welche sich durch den deswegen aufgebauten Unterdruck in der Förderleitung wieder gelöst haben dürfte.

Zur Datensichtung und dem Datenexport ist das Tool *i^TUW mon.Examine* in Verwendung. Mit diesem am Arbeitsplatzcomputer laufenden Programm können die Daten, wieder auf Basis des abstrakten Datenformats, dargestellt, interessante Ausschnitte selektiert und schließlich zur Weiterverarbeitung in ein Austauschdatenformat exportiert werden. Eine Reihe zweidimensionaler, konfigurierbarer Graphen, mit der Möglichkeit des Setzens von Cursor zur Ermittlung und Markierung interessanter Extremwerte, ist bei der Datensichtung behilflich. Abbildung 23 zeigt die Benutzerschnittstelle des Programmes. Im linken Teil des Bildes ist die Konfigurationsoberfläche dargestellt. Nach einem Login der Benutzerin oder des Benutzers und Festlegung des gewünschten Zeitbereiches werden die Daten aus der Datenbank geladen und in einem zweiten Fenster, in der Abbildung rechts zu sehen, dargestellt. Neben der Konfigurationsmöglichkeit zur Auswahl einzelner Messkanäle ist im Reiter *search* eine Volltextsuche über den gesamten Datenbestand möglich; nach Eingabe von beispielsweise „*T_*“ im Suchfeld werden sämtliche Temperaturkanäle der Station gemeinsam in einem Graphen zur Anzeige gebracht. Die Darstellung kann weitgehend frei konfiguriert werden (Zoom, Farben, Strichstärken und -typen).

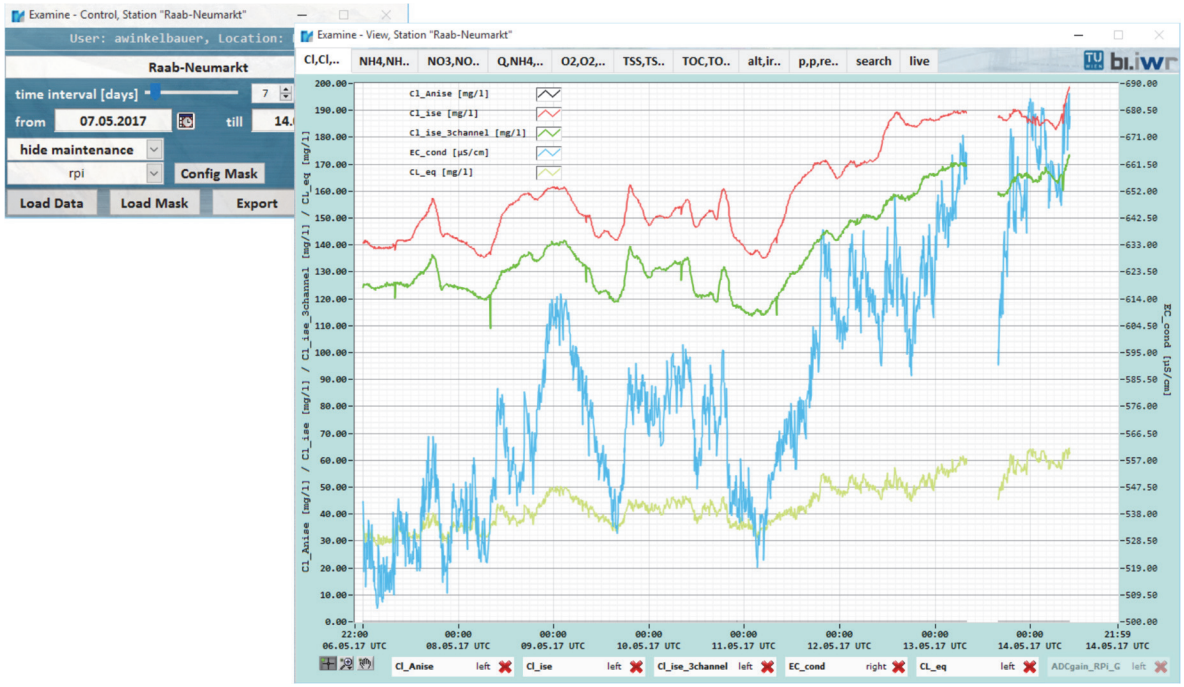

Abbildung 23: *i*^{TUW}*mon.Examine* (eigene Darstellung)

Die konsistente Kombination verschiedener Datenquellen wird durch einen Mechanismus namens *Merged-Export* zur Verfügung gestellt. Der Umfang erhobener Daten einer Station wird durch den zeitlich gestaffelten Einsatz verschiedener Sonden bestimmt. Jeder Messkanal ist durch nicht notwendigerweise kontinuierliche, äquidistante Messzeitpunkte bestimmt. Das komplette Bild des Gewässerzustands über der Zeit ergibt sich durch die Kombination aller Messkanäle zu einer gemeinsamen Zeitreihe mit nur mehr einer gemeinsamen Zeitspalte. Die ursprünglich dafür notwendige Rechenzeit beim Datenexport-Vorgang wurde durch geeignete Maßnahmen in der Datenbank auf die gesamte Zeit der Datenerfassung verteilt und die für den *Merged-Export* notwendige Datenstruktur wird kontinuierlich, nach Einlagen eines einzelnen, neuen Messwertes, befüllt. Im Endergebnis stehen sämtliche Messdaten aller Sonden, auf einer gemeinsamen Zeitspalte ausgerichtet, zu jeder Zeit vollständig zur Verfügung und die Dauer des Datenexportes ist auf die Dauer des Auslesens der Daten für den gewünschten Zeitraum aus der Datenbank reduziert.

Im Projekt „NaWas“ wurde auf automatisiertes Datenreporting gesetzt. Die Veröffentlichung der Messdaten findet im Dateiformat PDF statt; jeder aktuelle Interbrowser kann dieses Format ohne die Notwendigkeit von Plugins zum Rendern der Inhalte darstellen. Abbildung 24 zeigt einen typischen Werteverlauf des Berichtes, erstellt mit dem Tool *i*^{TUW}*mon.ADR* (Abkürzung für *automatisierter Datenreport*).

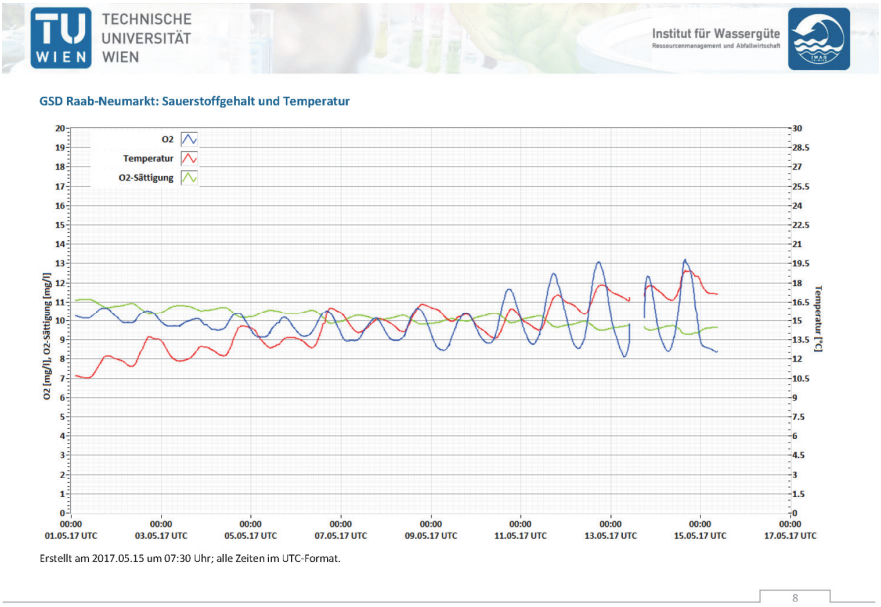

Abbildung 24: Automatisierter Datenreport, Ausschnitt [Iwag17]

Die darzustellenden Messkanäle und deren Darstellungsparameter wie zum Beispiel die Reihenfolge, die Strichstärken und -farben werden mit einem dafür gestalteten Werkzeug konfiguriert, Abbildung 25 zeigt einen Ausschnitt der Konfigurationsoberfläche. Der automatisch zu erstellende Bericht kann unter Einsatz diverser Gestaltungselemente (Drop-Down-Menüs, Farbauswahlfelder) konfiguriert werden. Die darzustellenden Signale werden aus einer Liste von Signalnamen (Datenfeld *SignalSpec* im Messkanal-Datenformat) ausgewählt. Einfache, auf einem gemeinsamen Konzept basierte Benutzerschnittstellen waren das Ziel der Softwareentwicklung. Eingriffe in die Konfiguration werden gemeinsam mit Informationen zur angemeldeten Benutzerin beziehungsweise dem angemeldeten Benutzer (Login beim Programmstart) und dem aktuellen Zeitstempel abgelegt und ermöglichen gemeinsam mit den Sicherungen aller veröffentlichten Reports die eindeutige Nachvollziehbarkeit der Datenveröffentlichung über einen langen Zeitraum.

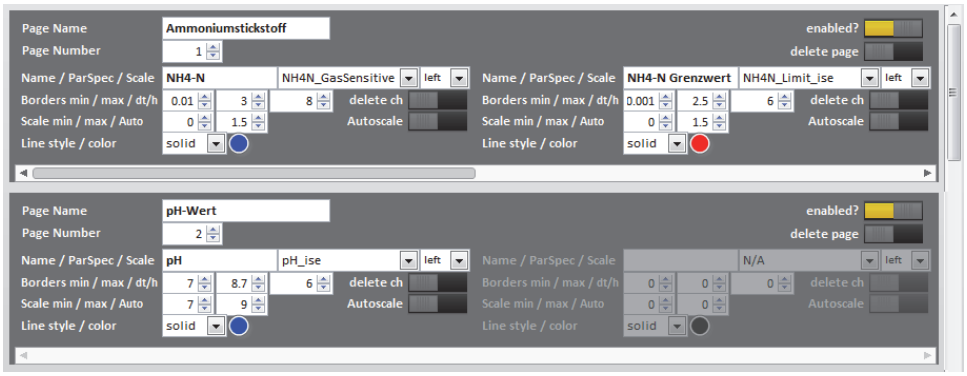

Abbildung 25: Konfigurationsoberfläche von *i*^{TUW}*mon.ADR* (eigene Darstellung)

Die Datenvisualisierung zur Betriebsüberwachung ist im Projekt KomOzAk [KHKS15] zur Anwendung gekommen. Ein auf der Leitwarte der Kläranlage aufgestelltes iPad der Firma Apple stellt dabei im Safari-Browser den aktuellen Anlagenzustand durch kontinuierlich aktualisierte, statische Bilder auf einem Webserver im virtuellen privaten Netzwerk dar (siehe Abbildung 26). Jede Änderung des Betriebszustandes und allfällige Alarmzustände werden laufend aktualisiert und je nach vorliegendem Fehlerfall wird textuell auf eine mögliche Fehlerbehebungsmethode hingewiesen beziehungsweise die Kontaktdaten der zu verständigenden verantwortlichen Person eingeblendet.

Abbildung 26: Darstellung des Stationszustandes in der Leitwarte [KHKS15]

Die Varianten einer webbasierten Datenvisualisierung unter Einsatz eines *Raspberry Pi* wurden in [WiKr15] untersucht; für grundlegende Informationen zur Computerplattform *Raspberry Pi* siehe [Rasp09]. Die untersuchten Visualisierungsdienstleister werden in Tabelle 2 in Hinblick auf den Ort der Datenverarbeitung und auf den Grad der Interaktivität darstellt. Der Begriff *cloud-basiert* meint in diesem Zusammenhang die Datenübertragung zum Dienstleister, welcher die notwendigen Berechnungen ausführt und die Ergebnisse auf ebendort gehosteten Webseiten zur Darstellung bringt.

Tabelle 2: Visualisierungsdienstleister aus [WiKr15] (aktualisiert)

	Lizenz	cloud-basiert	Interaktivität	Referenz
Google Charts	Google Terms Of Service	ja	dynamisch	https://developers.google.com/chart/
Highcharts	Creative Commons Attribution-NonCommercial 3.0 / paid license	nein	dynamisch	http://www.highcharts.com/
matplotlib	based on PSF license (Python Software Foundation)	nein	statisch	http://matplotlib.org/
plotly	plot.ly Terms Of Service	ja/nein	dynamisch	https://plot.ly/
Pygal	LGPL (GNU Lesser General Public License) v3	nein	semi-dynamisch	http://pygal.org/
RRDtool	GPL (GNU General Public License) v2	nein	statisch	http://oss.oetiker.ch/rrdtool/

Mit statischer Darstellung ist üblicherweise ein (laufend aktualisiertes) Bild ohne weiterer möglicher Benutzerinteraktion am Endgerät gemeint; die dynamische Darstellung ermöglicht hingegen weitergehenden Eingriff in die Visualisierung durch Anpassung der Skalen, Zoom-Funktionen und dem Ein- und Ausblenden interessanter Datenbereiche (siehe Abbildung 27). Bei der Nutzung *cloud-basierter*

Dienste zur Datenvisualisierung sind die Lizenzinformationen zu beachten und Fragen zur Datenübertragung in andere Länder und den dort geltenden Rechtssystemen zu klären. Darüber hinaus kann es alleine durch die Inanspruchnahme der Dienste zu einer (teilweisen) Übertragung von Rechten an den Daten kommen.

Abbildung 27: Statische und dynamische Datenvisualisierung [WiKr15]

In den letzten Jahren hat sich ein Trend zur Implementierung interaktiver Webservices unter Verzicht auf die Nachrüstung der Internetbrowser mit Plugins von Drittanbietern entwickelt. Der aktuelle HTML5-Standard [Wcwo14] bietet die dafür notwendigen Sprachelemente und wird von allen gängigen Browsern unterstützt.

Die Datenerfassung der meteorologischen Umgebungen bedient sich der Datenverarbeitungs- und Datenvisualisierungsfunktionen von $i^{TUW}mon$. Die Einbindung erfolgt durch das Entwickeln einer neuen Datenquelle, basierend auf der Datenübertragung über ModbusTCP. Die Daten werden nach den beschriebenen Grundlagen verarbeitet, zur Darstellung gebracht und stehen anschließend in einer zentralen Datenbank zur weiteren Verarbeitung mittels Algorithmen des maschinellen Lernens zur Verfügung.

2.3 Modellierung und Datenplausibilisierung

Die weitergehende Verarbeitung von Messdaten zur Informationsgenerierung durch Modellierung von Äquivalenzparametern beziehungsweise zum Absichern von Messergebnissen durch automatische Datenplausibilisierung ist ein wesentlicher Aspekt bei der Weiterentwicklung der beschriebenen Messnetzplattform.

Unter *Modellierung von Äquivalenzparametern* wird ganz allgemein die modellbasierte Berechnung von Zielparametern zur Bestimmung der Wassergüte auf Basis von beispielsweise der Messung der UV-Absorption mit Spektrometersonden als Ersatzgröße verstanden. Der Zielparameter wird nicht direkt erfasst, sondern aus anderen Zusammenhängen errechnet. Ein typisches Modell eines äquivalent modellierten Parameters kann folgende Form aufweisen:

$$AB_{eq} = k_1 \cdot abs_{230nm} + k_2 \cdot abs_{235nm} + k_3 \cdot abs^2_{310nm}. \quad \textbf{2.1}$$

Der Parameter AB_{eq} errechnet sich in diesem Beispiel aus den Absorptionskoeffizienten $abs_{\lambda nm}$ an drei bestimmten Wellenlängen, gewichtet mit den Faktoren $k_{1,2,3}$. Die mathematischen Grundlagen zum Finden von Zusammenhängen in großen Datenmengen mittels der *PLS-Regression, PLSR* (engl. *partial least square-regression*) werden in [Kavs02] und [Kräm07] dargelegt. Nach [Wink11] sind für die Generierung eines Modells rund 25 bis 30 sehr gute Referenzdatenpunkte notwendig, bestehend aus den Absorptionsspektren und zugehörigen Ergebnissen der Laboranalyse. Wurde die Tauglichkeit eines Modells anhand der Modellvalidierung an aktuellen Messdaten gezeigt, kann mittels des modellbasierten Ansatzes aus den Messdaten einer Sonde eine größere Zahl äquivalenter Messparameter errechnet werden. Die Grenzen der Modellierung liegen nach [Wink11] in der nur spärlich vorhandenen Übertragbarkeit der Modelle auf verschiedene Messorte und -applikationen. Die Modelle für einen Kläranlagenablauf sind außerdem stark unterschiedlich zu jenen eines beispielsweise offenen Fließgewässers.

Ein weiterer Problembereich ist die zum Teil nicht mögliche Kompensation von Störeinflüssen durch Überlagerung der Absorptionspeaks verschiedener Substanzen. In [WiSK17] wird auf diese Problematik in der weitergehenden Abwasserbehandlung eingegangen: Die spezifische Ozondosis, bezogen auf einen Äquivalentwert des gelösten, organischen Kohlenstoffs (engl. *DOC, Dissolved Organic Carbon*), berücksichtigt den teils hohen, stöchiometrischen Ozonverbrauch durch Nitrit nicht. Für die Kompensation dieses Einflusses müsste nun, entsprechend dem Konzept der Bestimmung der Regelungsparameter der Anlage aus UV-Absorptionsspektren, der Äquivalenzparameter für Nitrit aus eben diesen errechnet werden. Dies ist jedoch durch die Überlagerung des Nitrit-Absorptionspeaks mit jenen von Nitrat nicht möglich; die Kompensation würde also den Einsatz einer eigenen Nitrit-Sonde verlangen. Bei stark schwankendem Zielparameter kann es außerdem vorkommen, dass kein definierter Zusammenhang zwischen dem Absorptionsspektrum und dem Zielparameter ausgemacht werden kann [Wink11]. Einen grundlegenden Überblick zur Modellierung von Äquivalenzparametern auf Basis von *in situ*-Spektrometrie liefert [WSBT08]. Die erwarteten Unsicherheiten im Bereich der Modellierung im Bereich der städtischen Abwässer wird in [Daeb06] erarbeitet, auch in Bezug auf Änderungen der Sauerstoffsättigung durch nitrifizierende Bakterien. Diese Mikroorgansimen oxidieren Ammoniak zu Nitrit und Nitrit zu Nitrat und sind von grundlegender Bedeutung für die biologische Abwasserreinigung. Die Modellierung mittels *PLSR* wird in diesem Zusammenhang als taugliche Methode beschrieben. Neben den in Abschnitt 2.2 beschriebenen Vorteilen von *PostgreSQL* hat sich die Möglichkeit, die (Modell-)Berechnungen direkt in der Datenbank abwickeln zu können, als großer Vorteil für die Handhabung größerer Datenmengen erwiesen. Für komplexe Programmkonstrukte steht die prozedurale Sprache *PL/pqSQL* zur Verfügung [Post17b]. Neben der Modellierung äquivalenter Parameter werden auch die Tabellen des *Merged-Exports* direkt in der Datenbank errechnet.

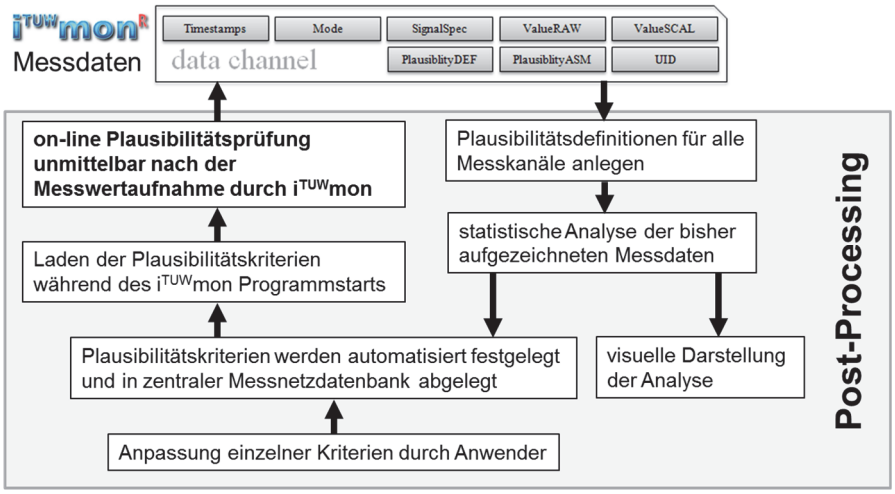

Abbildung 28: Post-Processing zur Datenplausibilisierung [FuWi15]

Die Absicherung von Messergebnissen durch die *automatisierte Datenplausibilitätsprüfung* ist ein wesentlicher und notwendiger Aspekt der weitergehenden Datennutzung erfasster Messparameter und ein weiteres Beispiel für das Finden von Zusammenhängen in den erfassten Messdaten. Neben der systembedingt jeder Messung anhaftenden Abweichungen der Datenerfassung (der Unmöglichkeit der Bestimmung des „wahren Wertes“) ist zum Teil umfangreiches Expertenwissen zur Interpretation von Zeitreihen notwendig. Ein Ansatz, einen Teil dieses Wissens in Algorithmen zur Datenplausibilisierung umzusetzen, wird in [FuWi15] untersucht. Basierend auf der Datenstruktur des Messkanals findet das Post-Processing in einem nachgereihten Verfahrensschritt statt (siehe Abbildung 28). Grundlage der Datenplausibilisierung ist die statistische Analyse und Modellierung der bis zu jedem Zeitpunkt erfassten Messdaten einer Messstation und die anschließende Festlegung der Kriterien durch den Anwender. Die ermittelten Plausibilitätskriterien werden zur Laufzeit von *i*^{TUW}*mon* übernommen, in das Datenfeld *PlausibilityDEF* eingetragen und dienen fortan als Basis der Plausibilitätsbewertung jeder weiteren Einzelmessung. Als besonders hilfreiche Datenquellen haben sich jene Daten erwiesen, die mehrfach, also redundant, mittels verschiedener Sonden aufgenommen wurden, eine Bilanzierung erlauben oder in besonders gut statistisch abgesicherter Korrelation mit dem betrachteten Zielparameter in Verbindung stehen [FuWi15].

Das eingesetzte Verfahren beruht im Wesentlichen auf einer Kerndichteschätzung nach Parzen [Parz62]. In die Verteilung der vorhandenen Messdaten wird dabei eine Standardnormalverteilung nach Gauß eingepasst und vorab festgelegte Perzentilgrenzen als Kriterium für die Plausibilitätsbewertung eines Messsignals verwendet. Als *nicht plausibles* Signal wird demnach die Über- beziehungsweise Unterschreitung dieser Grenzen gewertet. Im Randbereich wird die halbe Standardabweichung als Maß zur Ermittlung der Zwischenschritte von *nicht plausibel* (Plausibilität = 0%) zu *plausibel* (Plausibilität = 100%) herangezogen [FuWi15]. Für die Bestimmung der zulässigen Signaländerungen der Sauerstoffkonzentration im Oberflächengewässer wird das 98%-Perzentil zur Ermittlung der maximalen Signaländerung pro Zeiteinheit herangezogen; die minimal notwendige Signaländerung pro Zeiteinheit wird mit dem 1%-Perzentil definiert (Erkennung von „Strichfahren“). Die Bewertung der Plausibilität erfolgt direkt nach der Datenerfassung mit dem zum Zeitpunkt der Messung

geltenden Kriterien. Die Verhältnisse einer Messdatenreihe über zwei Jahre Stationsbetrieb sind in Abbildung 29 dargestellt.

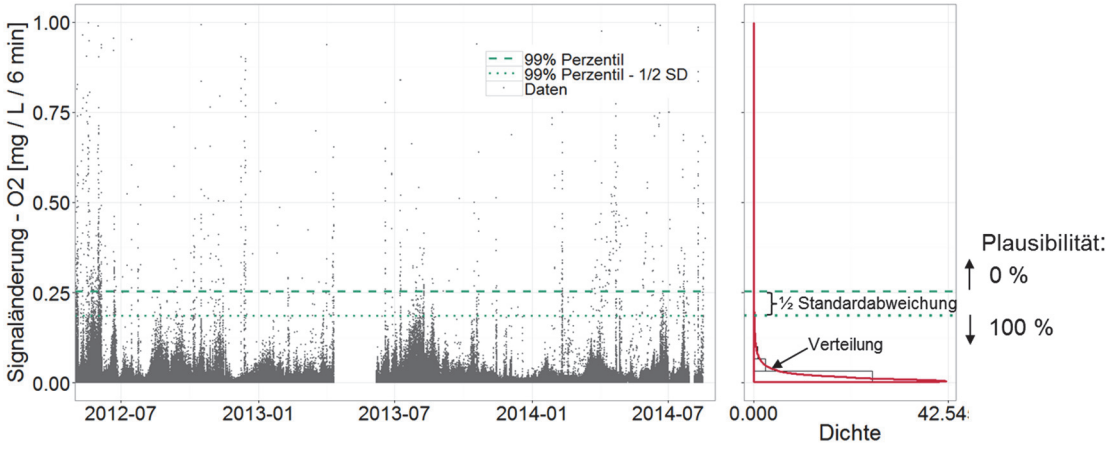

Abbildung 29: Plausibilitätskriterien der Sauerstoffkonzentration [FuWi15]

Die modellbasierte Plausibilitätskontrolle anhand gegenseitiger Abhängigkeiten von Messsignalen wird über eine Korrelation der Messdaten bewerkstelligt. Nach Ermittlung der Modellresiduen, in Abbildung 30 rechts dargestellt, wird wieder das Verfahren der Bestimmung der Perzentilgrenzen angewendet und ein Modell-Erwartungswert für die Chloridkonzentration in Abhängigkeit der gemessenen Leitfähigkeit festgelegt. Die Grenz- und Erwartungswerte als Ergebnis der Datenplausibilisierung werden kontinuierlich angepasst und der Verlauf zu Dokumentationszwecken in der Datenbank abgelegt.

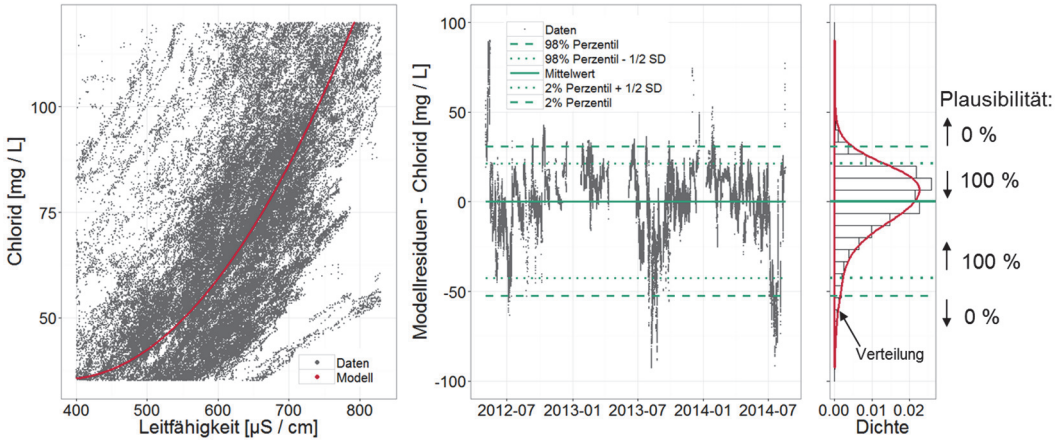

Abbildung 30: Korrelation Chloridkonzentration zu Leitfähigkeit [FuWi15]

Ein Ergebnis der Datenplausibilisierung einer Chloridkonzentration ist in Abbildung 31, nach Messbereichsunter- und Messbereichsüberschreitung, sowie in Abbildung 32, nach Korrelation mit der elektrischen Leitfähigkeit des Gewässers plausibilisiert, dargestellt. Die Plausibilisierung nach dem erlaubten Messbereich ist intuitiv noch nachvollziehbar und lässt sich mit „je näher am Rand des durch sämtliche bisherigen Messdaten definierten Bereiches, desto unplausibler“ beschreiben.

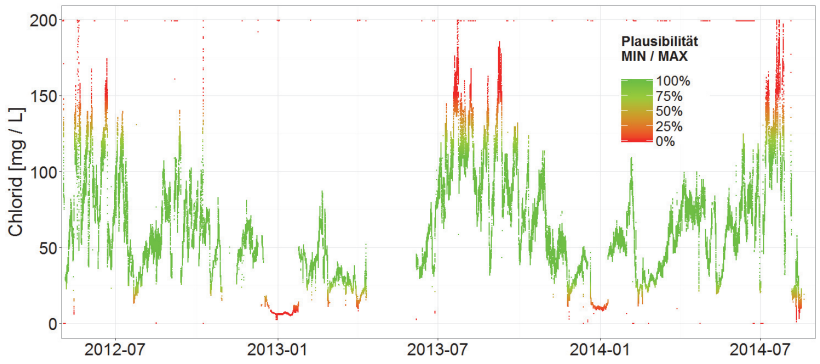

Abbildung 31: Chloridkonzentration der Raab, Plausibilisierung nach Messbereich [FuWi15]

Die Plausibilisierung nach Korrelation mit der Leitfähigkeit ergibt ein etwas weniger intuitives Bild und den Messdaten ist ohne Kenntnis weiterer Zusammenhänge nicht mehr eindeutig zuordenbar, ob diese als plausibel einzustufen sind oder nicht. Genau hier liegt der Gewinn der Plausibilisierung durch Korrelation für die praktische Anwendung. Auch wenn die endgültige Festlegung des Vertrauens in eine in Form einer Zeitreihe vorliegender Messdaten immer die Endanwenderin beziehungsweise der Endanwender treffen muss („educated guess"), gibt das Verfahren der automatisierten Datenplausibilisierung statistisch tragbare Einschätzungen bezüglich des Signalverhaltens, auch im Vergleich zu weiteren erfassten Messparametern am aktuellen Untersuchungsort. Zusätzliche Kriterien der Datenplausibilisierung, die Implementierung weiterer statistischer Zusammenhänge, hat sich dabei für die Einschätzung als sehr hilfreich erwiesen.

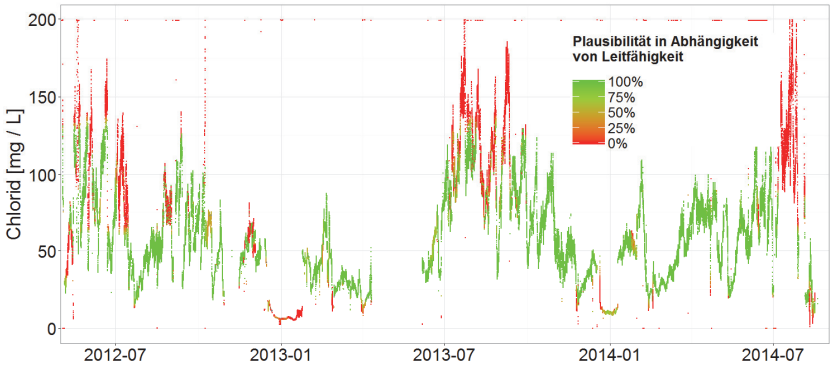

Abbildung 32: Chloridkonzentration der Raab, Plausibilisierung nach Korrelation [FuWi15]

Aufbauend auf den bisher vorgestellten verwandten Arbeiten zu Methoden und Techniken der Datenerfassung, Verarbeitung, Modellierung und Interpretation wird im folgenden Abschnitt das Messkonzept des Environmental Monitorings durch kostengünstige, hochintegrierte Messtechnik vorgestellt. Ein Ansatz zur Modellierung eines bestimmten Parameters, des im Wasser gelösten Sauerstoffs, basierend auf einer Zahl weiterer erfasster Datenquellen, wird im nächsten Schritt implementiert. Es wird erwartet, dass dadurch ein Surrogat-Parameter als Alternative zur direkten Sauerstoffmessung modelliert werden kann beziehungsweise eine Verbesserung der Datenplausibilisierung durch ein zusätzliches, statistisches Kriterium eintreten wird.

3. Modelle und Konzepte

Die Ideen des Environmental Monitorings zur Generierung eines Surrogat-Parameters beziehungsweise zur verbesserten Absicherung vorhandener Messdatenreihen durch Anwendung von Algorithmen aus dem Bereich des maschinellen Lernens wird in diesem Teil der Arbeit dargestellt. Im ersten Teil, Abschnitt 3.1, wird sozusagen „von unten“ beginnend die verfügbare Sensorhardware und die Anbindung an eine zentrale Datenerfassung untersucht. Die Konzepte zum Aufbau der Messhardware als Erweiterung der bestehenden Wassergütemessstation an der Raab werden ebenfalls im ersten Teil entwickelt. Die Methodik der Datenanbindung an die bestehende Infrastruktur der Messstation und die Einbindung der Datenstruktur in *i*^{TUW}*mon* über eine entwickelte Datenschnittstelle wird im zweiten Teil, Abschnitt 3.2, beschrieben. Der letzte Teil, Abschnitt 3.3, widmet sich schließlich der Modellierung mittels Algorithmen des maschinellen Lernens. Basierend auf den in der Einleitung vorgestellten Grundlagen, werden Algorithmen und deren Anwendbarkeit im Bereich der Modellierung von Zielparametern untersucht und für die nachfolgende Implementierung ausgewählt.

3.1 Sensorik am Raspberry Pi

Aus der *Do it yourself*-Bewegung der Heimwerker ist mit fortschreitender Miniaturisierung und steigender Verfügbarkeit leistungsfähiger Mikrocomputer die Subkultur der sogenannten *Maker-Szene* entstanden. Neben der Etablierung von 3D-Druckern zur programmatischen Umsetzung mechanischer Bau- und Ersatzteile ist eine große Zahl hochintegrierter, kostengünstiger Sensoren zur Erfassung der physikalischen Umgebungsparameter verfügbar. Viele in der Literatur beschriebene Systeme behandeln die Implementierung von drahtlosen Sensornetzwerken und deren Eignung zur Erfassung beispielsweise der Umgebungstemperatur mittels tragbaren Geräten. [MoFS17] vergleicht drei drahtlose IoT-Sensoren zur Messung von Temperatur und relativer Luftfeuchtigkeit und [TsXi16] stellt ein selbstgebautes Gerät zur Überwachung der PM2.5-Feinstaubkonzentration und UV-Bestrahlung vor. Die beschriebene drahtlose Vernetzung ist für die vorliegende Arbeit weniger relevant, die Zielsetzung liegt in der Nutzung der Infrastruktur vor Ort und einer möglichst wartungsarmen, langfristig stabilen und autonomen Datenerfassung.

Die im Hinblick auf die Messung der meteorologischen Umgebungsparameter in Frage kommende Sensoren sind in Tabelle 3 angeführt. Maßgeblich für die Auswahl ist die gute Verfügbarkeit der Hardware und ein gemeinsamer Datenkommunikationsbus zum Datenaustausch. Allen Parametern gemeinsam ist die Verfügbarkeit der Sensortechnik in Form von, den Prototypen-Aufbau der Station sehr vereinfachenden, Sensorplatinen. Der Vorteil liegt in den bereits vordefinierten Anschlüssen von Spannungsversorgung, Datenleitungen zur Kommunikation und etwaigen Pins zur Sensorinitialisierung.

Tabelle 3: Ausgewählte Parameter des Environmental Monitorings (eigene Darstellung)

Signalname	Parameter	Messprinzip	Messbereich		Einheit	Chip
T_ir	Temperatur	IR-Thermosäule	-40,0	125,0	°C	TMP006
T	Temperatur	ohmsch	-40,0	85,0	°C	BMP280
p	Luftdruck	mikro-elektromechanisch	300,0	1100,0	hPa	BMP280
relHum	relative Luftfeuchte	kapazitiv	0,0	100,0	%	HDC1008
UVindex	UV-Index	photoelektrisch, IR-kompensiert	0,0	11,0	1	SI1145
UVindex	UV-Index	photoelektrisch, Diode	0,0	15,0	1	GUVA-S12SD
LUX	Einstrahlung	photelektrisch, hohe Dynamik	0,1	40000,0	lux	TSL2561
RGBC	Farbe, Einstrahlung	photoelektrisch, IR-gefiltert	0,0	65535,0	1	TCS34725
RGB_cam	Kamerabild, Farbe	CMOS-Sensor	0,0	255,0	1	RPI-CAM

Die zur Verfügung stehenden Messprinzipien lassen sich in drei Bereiche einteilen. Die erste Kategorie bilden Sensoren mit elektrischer Zwischengröße, wie zum Beispiel die Temperaturerfassung, welche über temperaturabhängige Widerstände beziehungsweise einer Infrarot-Thermosäule (zur berührungslosen Temperaturmessung durch Erfassung der Infrarotstrahlung) stattfindet. Die Messung der relativen Luftfeuchtigkeit erfolgt über einen kapazitiven Sensor. Zur Erfassung des Luftdrucks kommt als zweite Kategorie der Messprinzipien ein sogenanntes mikro-elektromechanisches System zum Einsatz. In der dritten Kategorie finden sich optische Sensoren zur Erfassung von UV-Index, Sonnenbestrahlung und der Grundfarben.

Allen angeführten Sensoren zugrunde liegt ein grundsätzlich ähnlicher Aufbau wie in Abbildung 33 dargestellt. Die nichtelektrische Eingangsgröße wirkt auf das Sensorelement und wird in eine elektrische Zwischengröße umgesetzt. Für die Temperaturmessung kann dies eine Widerstandsänderung, für optische Messungen ein entsprechend kleiner, durch Anregung freier Ladungsträger an einem Halbleiterübergang angeregter, Stromfluss sein.

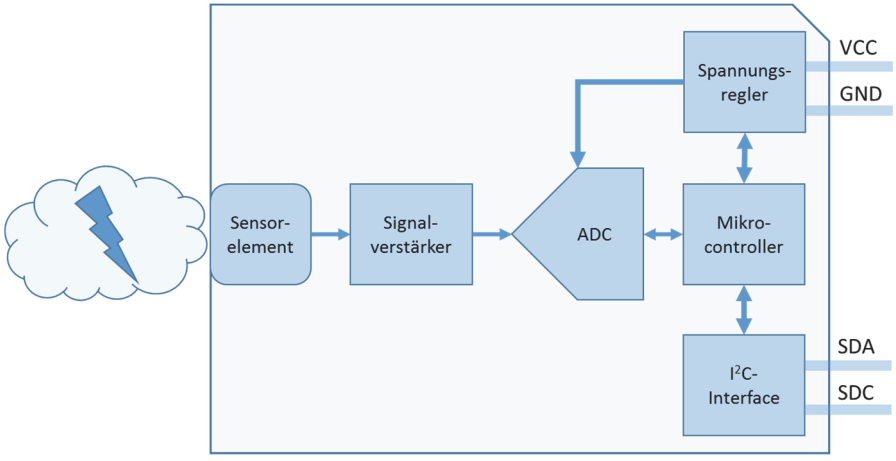

Abbildung 33: Blockdiagramm Sensor (eigene Darstellung)

Ein Signalverstärker setzt das Signal in einen besser zu verarbeitenden Bereich eines höheren Spannungssignals bei entsprechend kleiner Ausgangsimpedanz um (Wirkung als Spannungsquelle). Im nächsten Schritt der Datenerfassung wandelt der Analog-Digital-Konverter *ADC* das analoge Signal in ein digitales Datenwort zur Weiterverarbeitung im Mikrocontroller um. Die Daten werden vom

Mikrocontroller in regelmäßigen Abständen in Speicherregister des *I²C*-Bausteins (engl. *Inter-Integrated Circuit*) geschrieben und von dort bei Datenkommunikation mit dem Busmaster nach einem vereinbarten Protokoll an die zentrale Datenerfassung übertragen.

Die digitale Datenkommunikation findet über den Zweidrahtdatenbus *I²C* statt [Nxps14]. Die Kommunikation über nur zwei Leitungen, *SDA* für *Serial Data* und *SCL* für *Serial Clock*, ermöglicht die kupferschonende, weil ohne zusätzlich notwendigen Chip Select-Leitungen ausgeführte, Anbindung einer Vielzahl von Busgeräten, zwischen beispielsweise den Controller-*ICs* (engl. *Integrated Circuit*) und der Peripherie einer elektronischen Schaltung. *I²C* ist ein serieller, auf Master-Slave-Konzept basierender Datenbus: Die Datenkommunikation wird durch einen Busmaster initiiert, welcher die ihm bekannten Slaves durch deren Adressen am Bus anspricht und den Datenaustausch durchführt. Die Datenübertragung wird mit einem Start- und einem Stoppzeichen initiiert und terminiert, gekennzeichnet durch Pegelwechsel auf der Leitung *SDA* von HIGH auf LOW (während Pegel auf Leitung *SCL* auf HIGH) als Startzeichen und einem Pegelwechsel auf Leitung *SDA* von LOW auf HIGH (während Pegel auf Leitung *SCL* auf HIGH) als Stoppzeichen. Bei einer gültigen Datenübertragung zwischen Start und Stopp darf sich der Pegel auf *SDA* während *SCL* HIGH nicht ändern. Der Standard sieht Arbitrierungsmechanismen beim Einsatz von mehr als einem Busmaster vor; aufgrund der geplanten Anwendung eines einzigen Busmasters wird hier nicht näher auf Details dazu eingegangen. Der Master spricht die Slaves am Bus über eine 7-Bit lange Slave-Adresse an, welche direkt nach dem Start-Signal gesendet wird. Von den theoretisch 2^7 möglichen Slaves sind abzüglich reservierter Adressen praktisch bis zu 112 adressierbare Geräte an einem gemeinsamen Datenbus zu betreiben. Das achte Bit der Datenübertragung wird zur Definition der Datenrichtung verwendet: LOW bedeutet Datenfluss in Richtung Slave, HIGH in Richtung Master. Mit diesem Ansatz ist bidirektionale Datenübertragung implementierbar.

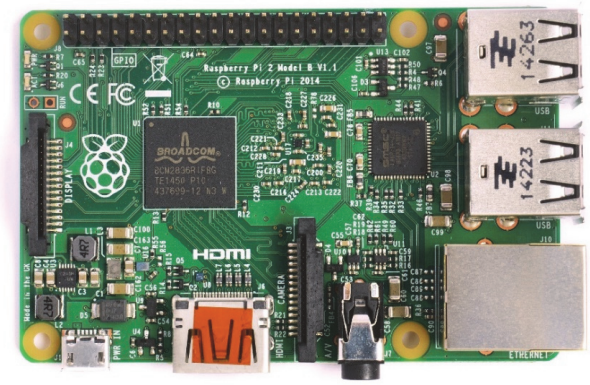

Abbildung 34: Raspberry Pi 2 Model B [Mult15][7]

Für die zentrale Datenverarbeitung bietet sich die Verwendung der kreditkartengroßen Mikrocomputerplattform *Raspberry Pi* in Version 2 Modell B, veröffentlich im Februar 2015, an [Rasp09], dargestellt in Abbildung 34. Kernstück ist das *SoC* (engl. *System on Chip*) BCM2836 der Firma Broadcom,

[7] Autor: Multicherry, Lizenz: Creative Commons Attribution-Share Alike 4.0 International

mit einer 32-Bit Quadcore-*CPU* (engl. *Central Processing Unit*) auf Basis der Cortex-A7-Architektur ausgestattet [Arm13]. Die Grafikausgabe wird mit einer VideoCore IV-*GPU* bewerkstelligt. Der Kerntakt beträgt 900 MHz, der Arbeitsspeicher ist 1 GByte groß. Die Platine ist mit einer Reihe von Anschlüssen für die Peripherie ausgestattet. 4 USB-Ports, ein 10/100-Mbit Ethernet Port und ein *HDMI*-Port sorgen für die Verbindung zur Außenwelt. Das Betriebssystem und die zugehörigen Daten werden auf einer, an der Unterseite der Platine montierten, microSD-Speicherkarte abgelegt. Zusätzlich zu den meteorologischen Sensoren wird eine Kamera mit *CMOS*-Bildsensor (engl. *complementary metal-oxide-semiconductor*) zur Himmelsbeobachtung über das *CSI* (engl. *Camera Serial Interface*) des *Raspberry Pi* eingebunden.

Für die Prototypenentwicklung steht eine 40-Pin Stiftleiste, der sogenannte *GPIO-Header* (engl. *general purpose input output*) zur Verfügung, auf welcher alle relevanten Ein-Ausgabeports des *SoC* herausgeführt sind. Dazu zählen Anschlüsse der I^2C-Schnittstelle und weitere Datenübertragungsbusse, serielle Schnittstellen, ein Ausgang für Pulsweitenmodulation und weitere, frei programmierbare Pins für digitale Signalverarbeitung. Passend zur Hardware wurde Linux als Betriebssystem gewählt. Die Idee zu diesem System wurde erstmals 1991 von Linus Torvalds mit „I'm doing a (free) operating system (…) [Linu11]“ in einer usenet-Gruppe erwähnt. Das Betriebssystem wird seit dieser Zeit stetig weiterentwickelt und es steht eine große Zahl sogenannter Distributionen, in Form vorselektierter Pakete unterschiedlichen Programm- und Funktionsumfangs, zur Verfügung. Eine der bekanntesten ist die Debian-Distribution [Spis16]. Dem Mikrocomputersystem *Raspberry Pi* wird aufgrund der nativen Unterstützung der Programmiersprache *Python* [Pyth17], der Verfügbarkeit des Linux-Betriebssystems und der direkten Anbindung des Embedded Systems über Ethernet der Vorzug gegenüber Mikrocontroller-Plattformen wie zum Beispiel *Arduino* gegeben.

Die Sensoren werden gemeinsam mit dem *Raspberry Pi* außerhalb der Messstation in einem geeigneten Gehäuse im Freien montiert. Dieses wind- und wetterfeste Gehäuse soll mit möglichst wenigen Durchführungen für Leitungen der Energieversorgung und der Datenkommunikation ausgestattet werden, weshalb sich eine Anbindung über IEEE 802.3 af, *PoE* (engl. *Power over Ethernet*), anbietet [Ieee03]. Bei diesem Verfahren wird neben den Daten auch die Spannungsversorgung der angeschlossenen Peripherie über die Twisted-Pair-Leitungen des Ethernetkabels übertragen. Auf Seiten der Messstation ist dafür ein sogenannter PoE-Injektor und auf Seiten des Messaufbaus ein PoE-Splitter vorzusehen. Der Injektor übernimmt eingangsseitig die Ethernet-Verbindung und die Gleichspannungsversorgung; der Splitter bietet auf seiner Ausgangsseite eine RJ45-Buchse für die Ethernet-Verbindung und eine Hohlbuchse mit 5V Gleichspannung zur Versorgung des *Raspberry Pi* über einen Micro-USB-Anschluss. Die erwartete, zu übertragene Leistung für den Mikrocomputer inklusive der angeschlossenen Sensoren liegt im Bereich von 5 Watt und ist somit problemlos mittels PoE-Anbindung realisierbar. Am Gehäuse des Versuchsaufbaus ist daher nur eine Kabeldurchführung (inklusive wasserdichter Verschraubung) vorzusehen.

Nach der Beschreibung der Konzepte der Hardwarerealisierung widmet sich der nächste Abschnitt der Datenerfassung und Einbindung des Versuchsaufbaus in $i^{TUW}mon$.

3.2 Datenaggregation und Datenverarbeitung

Das Konzept der Messdatenerfassung mit dem *Raspberry Pi* und der Einbindung in die *i*^{TUW}*mon*-Plattform wird in diesem Abschnitt beschrieben. Das Ziel der Datenverarbeitung ist die Sicherstellung der Verfügbarkeit aller herkömmlich erfassten Wassergütemessdaten der Station gemeinsam mit den Messkanälen des *Environmental Monitoring*-Prototypen in der zentralen Datenbank. Das grundlegende Konzept ist in Abbildung 35 dargestellt. Die Teile *I*^{2}*C-Schnittstellentreiber* und *ModbusTCP-Server* laufen lokal am Linux-System des *Raspberry Pi*. Der Schnittstellentreiber ist zuständig für den Datenabruf aller Messkanäle über vom Hersteller zur Verfügung gestellte *I*^{2}*C*-Treiberprogramme und setzt die Daten in ein geeignetes Format für übergeordnete Programmteile um. Der *ModbusTCP-Server* wird nach abgeschlossenem Bootvorgang gestartet, ruft zyklisch die Daten vom Schnittstellentreiber ab und schreibt diese in die entsprechenden Register. Der Zugriff auf die Kommandozeile des *Raspberry Pi* wird über das Netzwerkprotokoll *SSH* (engl. *secure shell*) auf TCP-Port 22, abgewickelt [Ietf06]; OpenSSH [Open17] wird als Implementierung des Protokolls am *Raspberry Pi* genutzt. Der Versuchsaufbau ist damit im sogenannten *headless*-Modus, ohne angeschlossene Peripherie wie Bildschirm und Maus beziehungsweise Tastatur, auch aus der Ferne bedienbar.

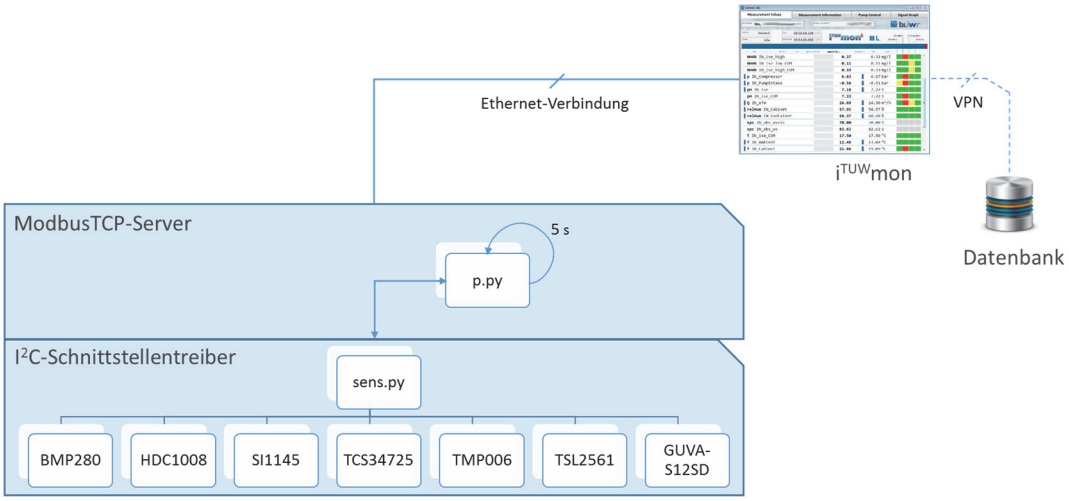

Abbildung 35: Softwarekonzept Environmental Monitoring (eigene Darstellung)

Die Software zur Datenerfassungsaufgabe am *Raspberry Pi* zeichnet sich durch eine relativ gering zu erwartende Programmkomplexität, abgesehen von den verwendeten Bibliotheken und Treibern, aus. Sie wird mittels der Programmiersprache *Python* [Pyth17] umgesetzt. Ein wesentlicher Vorteil von *Python* ist die Optimierung auf Einfachheit und Übersichtlichkeit des Quellcodes, unter anderem durch eine auf wesentliche Bausteine beschränkte Syntax und dem Stilmittel der Einrückung zur Programmstrukturierung im Vergleich zur umfangreich notwendigen Klammernsetzung beispielsweise in der Sprache *C*. Die einzelnen Programmteile werden als *Python*-Programme, sogenannte *Python*-Skripte entworfen und mit Linux-Bordmitteln als Dienst ausgeführt.

Für *i*^{TUW}*mon* wird ein Softwaremodul zum Ansprechen der neuen Datenschnittstelle vorgesehen, welches die Daten von einem zu implementierenden ModbusTCP-Client übernimmt und in das übliche

Datenformat des Messkanals übersetzt. Die Daten sind gemeinsam mit den weiteren Messdaten der Station, entsprechend den üblichen Zyklen der Datenerfassung, in der zentralen Datenbank verfügbar.

Nach Beschreibung der Konzepte für die Sensorik und der notwendigen Programmstrukturen zur Einbindung folgt im nächsten Abschnitt die Beschreibung der Weiterverarbeitung der Daten mittels Modellierung durch Einsatz ausgewählter Algorithmen des maschinellen Lernens.

3.3 Modellierungskonzepte des maschinellen Lernens

Im Hinblick auf die Modellierung und mögliche Vorhersagen künftiger Werteverläufe auf Basis erfasster Zeitreihen der Vergangenheit wird aus dem Gebiet des maschinellen Lernens auf das Konzept des überwachten Lernens zurückgegriffen. Der Zielparameter ist durch einen kontinuierlicheren Zeitverlauf gekennzeichnet und nicht durch binäre Aussagen wie wahr/falsch klassifiziert, die Methode der Regression (und nicht der Klassifikation) ist daher Mittel der Wahl. Im folgenden Abschnitt werden die mathematischen Grundlagen ausgewählter Algorithmen formuliert und vorgestellt. Die bewusst trivial gehaltenen Darstellungen dienen der Veranschaulichung der Zusammenhänge. Die Grundlagen der linearen Regression beispielsweise werden in einer großen Zahl von Implementierungen angewendet und münden in numerisch optimierte Algorithmen-Bibliotheken, denen die fundamentalen Zusammenhänge nur mehr schwer „anzusehen“ sind. Die folgenden Ausführungen beruhen auf den Lecture Notes von Andrew Ng aus Standford [Ng16] aufgrund der anschaulichen Darstellungen und Nomenklatur im Vergleich zu weiterer, grundlegender Literatur. Ein von ihm gestalteter Machine Learning-Kurs auf Coursera bringt ausgewählte Kapitel in gekürzter Form [Ng17]. Im dieser Arbeit werden die wesentlichen und für das Verständnis der Modellierung von neuronalen Netzten notwendigen Teile zusammengefasst beschrieben.

Die Informationsverarbeitung im menschlichen Gehirn basiert auf Nervenzellen, den Ganglienzellen (in Abbildung 36 dargestellt). Nach [Psch82, S.559] wird unter einem *Neuron* die Gesamtheit der Ganglienzellen und deren Fortsätze verstanden. Eingangsseitige Verbindungen werden über verästelte Fortsätze, den Dendriten, hergestellt, ausgangsseitige über das Axon. Die elektro-chemische Schnittstelle zwischen (Nerven-)Zellen wird Synapse genannt und dient, in verkürzter Darstellung, der Informationsübertragung durch Überspielung von Aktionspotentialen [Pfüt03, S.176,187].

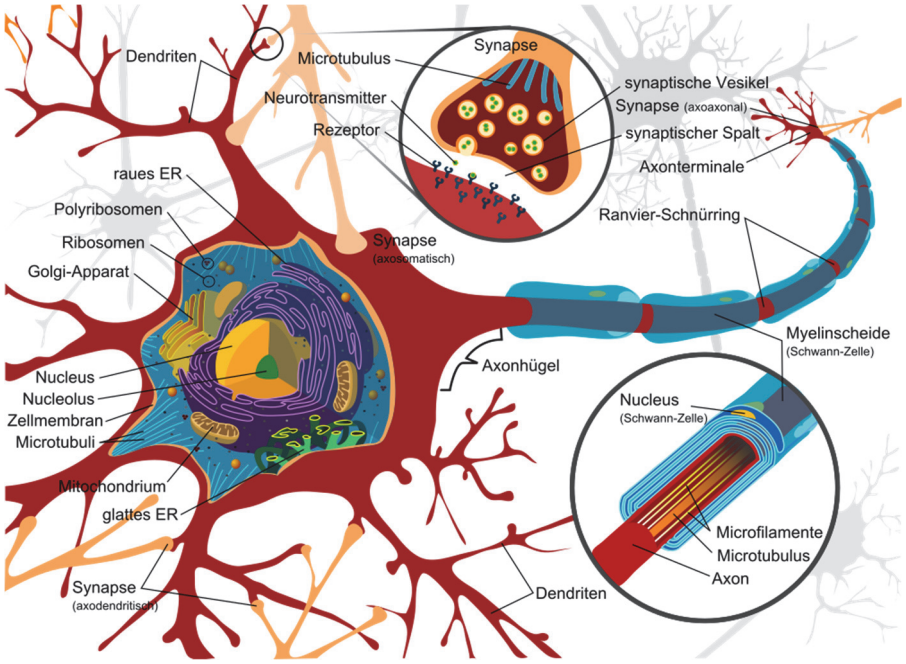

Abbildung 36: Nervenzelle [Mari07][8]

Das Grundprinzip der Informationsverarbeitung kann nach [Pfüt03, S.191] wie folgt verstanden werden: An den postsynaptischen Membranen entstehen durch Aktionsimpulse anderer, über Synapsen verbundener, Nervenzellen postsynaptische Potentialdifferenzänderungen, welche Ausgleichsströme bewirken, die am Axonhügel summarisch wirksam werden (analoge Signalverarbeitung). Wird durch die Summe der einwirkenden Impulse ein bestimmter Schwellwert überschritten, löst die betroffene Nervenzelle einen Aktionsimpuls aus (digitale Signalverarbeitung). Dieses Verhalten ist verantwortlich für die Motivation zur Entwicklung sogenannter *künstlicher neuronaler Netzwerke* (engl. *ANNs*, *Artificial Neural Networks*). Mittels neuronalen Netzen wird versucht, die Informationsverarbeitung nach dem Vorbild der Natur nachzubauen. Ein neuronales Netz wird nach [Hayk98, S.24] wie folgt definiert:

> „A neural network is a massively parallel, distributed processor made up of simple processing units, which has a natural propensity for storing experiental knowledge and making it available for use."

Die mathematischen Grundlagen der Modellierung von neuronalen Netzten bauen auf den nun folgenden Algorithmen auf. Der Vorgang der Modellierung ist dabei ganz allgemein als eine Optimierungsaufgabe zu verstehen.

Als erster Algorithmus des maschinellen Lernens wird das statistische Verfahren der *linearen Regression* vorgestellt. Basis des Verfahrens ist das Vorliegen einer Anzahl von m Tupeln von (Trainings-) Werten $\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \ldots, \left(x^{(m)}, y^{(m)}\right)$, wobei x als Eingangsgröße (engl. *feature*) und y als

[8] Autor: Zeichnung - Mariana Ruiz Villarreal (LadyOfHats), Übersetzung Deutsch - NEUROtiker, Lizenz: public domain

Ausgangsgröße beziehungsweise Zielwert (engl. *target*) definiert wird. Der Index bezeichnet in diesem Fall, würde die Darstellung in Tabellenform erfolgen, die entsprechende Zeilennummer. Nun wird mittels der Hypothese h_θ unter Verwendung der Parameter θ_0 und θ_1 versucht, folgenden linearen Zusammenhang zu modellieren (nach [Ng17], Lecture 2):

$$h_\theta\left(x^{(i)}\right) \coloneqq \theta_0 + \theta_1 \cdot x^{(i)} \tag{3.1}$$

Man möchte eine Funktion h_θ finden, welche bei gegebenen Eingangswerten x im Vergleich zur gemessenen Ausgangsgröße y einen möglichst ähnlichen, im Idealfall gleichen, Funktionswert liefert. Das Ziel der Modellierung ist also die Minimierung der *Abstände* zwischen h_θ und y durch fortlaufende Adaptierung der Hilfsparameter θ_0 und θ_1. Als ein Maß für die Güte der Modellierung wird die sogenannte Kostenfunktion $J(\theta_0, \theta_1)$ verwendet, welche nach der *Methode der kleinsten Quadrate* (engl. *least squares* nach Legendre und Gauß) wie in [Ng17], Lecture 2, ausgeführt wie folgt definiert werden kann:

$$J(\theta) = J(\theta_0, \theta_1) \coloneqq \frac{1}{2m}\sum_{i=1}^{m}(h_\theta\left(x^{(i)}\right) - y^{(i)})^2 \tag{3.2}$$

Das Optimierungsziel des Algorithmus ist im Finden des Minimums der Kostenfunktion über alle Parameter, im linearen Fall mit nur zwei Parametern nach [Ng17], Lecture 2 durch

$$\min J(\theta) = \min_{\theta_0,\theta_1} J(\theta_0, \theta_1) \tag{3.3}$$

gegeben. Mit der sogenannten *Norm im quadratischen Mittel* (siehe [Tasc15, S.270]) kann das Minimierungsproblem auch in folgender Weise dargestellt werden:

$$\min_\theta \|f(\theta) - y\|_2^2. \tag{3.4}$$

Diese Minimierungsaufgabe kann unter Voraussetzung einer reellwertigen, differenzierbaren Funktion $f\colon \mathbb{R}^n \to \mathbb{R}$ zum Beispiel mit dem *Gradientenverfahren* (engl. *gradient descent*) gelöst werden. M. Augustine Cauchy hat 1847 erste Überlegungen dazu angestellt, noch ohne explizit den Begriff „Gradient“ zu verwenden [Cauc47]. Die Anwendung des Gradienten in der Darstellung

$$\tau^{(k+1)} \coloneqq \tau^{(k)} - \alpha^{(k)} \cdot \vec{\nabla} f(\tau^{(k)}) \tag{3.5}$$

dient zur Berechnung der Parameter τ des $(k+1)$-ten Iterationsschrittes aus jenen des k-ten. Dabei beteiligte Faktoren sind $\vec{\nabla} f(\tau^{(k)})$, der Gradient der Funktion f an der Stelle $\tau^{(k)}$ und die sogenannte Lernrate oder auch Schrittweite α. Der vektorielle Differentialoperator Nabla mit seinem Symbol $\vec{\nabla}$ nach William Rowan Hamilton lässt sich algebraisch als gewöhnlicher Vektor behandeln. Seine Komponenten entsprechen partiellen Ableitungsoperatoren $\frac{\partial}{\partial \tau}$. In einem orthogonalen Koordinatensystem mit den Eins-Tangentenvektoren $\vec{e}_i$ lässt sich Nabla nach [Prec10] darstellen als

$$\vec{\nabla} = \sum_i \vec{e}_i \, \partial_i. \tag{3.6}$$

Formal erfolgt die Bildung des Gradienten durch ein einfaches Produkt des Nabla-Operators mit der Funktion f als

$$\text{grad} f \coloneqq \vec{\nabla} f. \tag{3.7}$$

Der Gradient, angewendet auf ein Skalarfeld, führt zu einer vektorwertigen Funktion, ein Gradientenfeld. Der Betrag des Gradienten entspricht dabei der größten Änderungsrate des Skalarfeldes. Die Richtung des Gradienten zeigt in Richtung des stärksten Anstiegs beziehungsweise, mit negativem Vorzeichen versehen, in Richtung des stärksten Abstiegs auf der in unserem Beispiel von den Parametern θ_0 und θ_1 aufgespannten Ebene im Raum. Genau dieser Zusammenhang ist für das Finden der minimalen Parameter der Kostenfunktion $J(\theta_0, \theta_1)$ hilfreich. Das Gradientenverfahren führt zu einer Bewegung auf dieser Ebene in Richtung des nächsten (lokalen) Minimums. Die vektorielle Basis wird bei der Berechnung vernachlässigt und der Wert des Gradienten geht mit der Schrittweite $\propto$ gewichtet direkt in den Parameter der nächsten Iteration ein.

Unter Verwendung von Gl. **3.3** ergibt sich mit Gl. **3.7** mit $j = 0$ und $j = 1$, wobei alle Parameter θ gleichzeitig berechnet werden und ausgehend vom Iterationsschritt k zu

$$\theta_j^{(k+1)} = \theta_j^{(k)} - \alpha^{(k)} \cdot \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)^{(k)} \tag{3.8}$$

oder verallgemeinert

$$\theta_j = \theta_j - \alpha \cdot \frac{\partial}{\partial \theta_j} J(\theta). \tag{3.9}$$

Beim Vorliegen eines einzigen Trainingsbeispiels ($m = 1$) ergibt sich mit Gl. **3.2** und nach den Ausführungen von [Ng16, S.4]

$$\begin{aligned} \frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \left[\frac{1}{2} \sum_{i=1}^{1} (h_\theta(x) - y)^2 \right] = \frac{\partial}{\partial \theta_j} \left[\frac{1}{2} (h_\theta(x) - y)^2 \right] = \\ = (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y) = (h_\theta(x) - y) \cdot x_j, \end{aligned} \tag{3.10}$$

und eingesetzt in Gl. **3.9** schließlich

$$\theta_j = \theta_j - \alpha \cdot (h_\theta(x) - y) \cdot x_j. \tag{3.11}$$

Dieser Zusammenhang ist als LMS-Regel (engl. von *least mean squares*) oder Widrow-Hoff-Regel bekannt [WiHo60]).

Nach Ausführung der partiellen Differentiation unter Berücksichtigung der Kettenregel der Differentialrechnung für differenzierbare Funktionen u und v, $(u \cdot v)' = u'(v(\tau_0)) \cdot v'(\tau_0)$, und Einsetzen von $j = 0$ und $j = 1$ folgt aus Gl. **3.8** und adaptiert nach [Ng17], Lecture 2:

$$\begin{aligned} \theta_0^{(k+1)} = \theta_0^{(k)} - \alpha^{(k)} \cdot \frac{\partial}{\partial \theta_0} J^{(k)} = \theta_0^{(k)} - \alpha^{(k)} \cdot \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot 1 \\ \theta_1^{(k+1)} = \theta_1^{(k)} - \alpha^{(k)} \cdot \frac{\partial}{\partial \theta_1} J^{(k)} = \theta_1^{(k)} - \alpha^{(k)} \cdot \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}. \end{aligned} \tag{3.12}$$

Das Gradientenverfahren ist geeignet zum Finden von lokalen Optima der Parameter θ, die Schrittweite α muss nach [Ng17], Lecture 2 im Laufe der Optimierung nicht angepasst werden.

Die Erweiterung der einfachen linearen Regression zu einer sogenannten multivariablen Regression oder auch *allgemeines lineares Modell* genannt (engl. *general linear model*), beruht auf der Verfügbarkeit zusätzlicher Eingangsparameter. Aus den m Tupeln vom vorigen Ansatz mit nur einer bestimmenden Eingangsgröße x wird eine Liste aus einer Anzahl m von Messwerten i, einer Anzahl n Spalten von Eingangsparameter j, $x_j^{(i)}$ mit $i = 1, 2, \dots m, j = 1, 2, \dots n$ und einer Spalte der Ausgangsgröße $y^{(i)}$, wiederum aus m Messwerten bestehend (siehe Tabelle 4).

Tabelle 4: Datenbasis des allgemeinen linearen Modells (eigene Darstellung)

Eingangsparameter				Ausgangsparameter
$x_1^{(1)}$	$x_2^{(1)}$	…	$x_n^{(1)}$	$y^{(1)}$
$x_1^{(2)}$	$x_2^{(2)}$	…	$x_n^{(2)}$	$y^{(2)}$
…	…	…	…	…
$x_1^{(m)}$	$x_2^{(m)}$	…	$x_n^{(m)}$	$y^{(m)}$

Die Hypothese h_θ wird nun so erweitert, dass sämtliche Eingangsparameter mit einem Parameter θ_j, dem sogenannte Gewicht des Eingangsparameters j, versehen werden (siehe [Ng17], Lecture 4),

$$h_\theta(x) \coloneqq \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \theta_2 \cdot x_2 + \cdots + \theta_n \cdot x_n \text{ mit } x_0 \coloneqq 1$$
$$\text{bzw. } h_\theta(x) = \boldsymbol{\theta}^T \cdot \boldsymbol{x}, \qquad \textbf{3.13}$$

bei Darstellung der Summenbildung als Vektorprodukt. Die zugehörigen Parameter $\boldsymbol{\theta}$ und $\boldsymbol{x}$ in vektorieller Form sind definiert als

$$\boldsymbol{\theta}^T = [\theta_0, \theta_1, \dots, \theta_n]^T \text{ und } \boldsymbol{x} = [x_0, x_1, \dots, x_n]. \qquad \textbf{3.14}$$

Die Kostenfunktion der allgemeinen linearen Regression errechnet sich unter Beibehaltung der Nomenklatur wieder zu

$$J(\theta) \coloneqq \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2. \qquad \textbf{3.15}$$

Mit Gl. **3.12** und [Ng17], Lecture 4, ergeben sich die Parameter für die Berechnung des Gradienten der $(k+1)$-ten Iteration wie vorhin beschrieben zu

$$\theta_j^{(k+1)} = \theta_j^{(k)} - \alpha^{(k)} \cdot \frac{\partial}{\partial \theta_j} J(\theta)^{(k)} =$$
$$= \theta_j^{(k)} - \alpha^{(k)} \cdot \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)}, \qquad \textbf{3.16}$$

mit $j = 1,\ 2, \ldots n$, wobei wieder alle Parameter θ gleichzeitig berechnet werden müssen und der Algorithmus bis zur Konvergenz der Parameter durchgeführt wird. Konvergenz bedeutet in diesem Zusammenhang, dass durch weitere Iterationen sich der gefundene, minimale Wert der Parameter, nicht mehr wesentlich ändert. Durch Summation über alle $x_j^{(i)}$ in Gl. **3.16** wird dieses Verfahren auch *stapelverarbeitendes* Gradientenverfahren (engl. *batch gradient descent*), genannt; bei jedem Iterationsschritt wird über sämtliche Eingangsgrößen aufsummiert. Eine Verallgemeinerung der Regression mit einem allgemeinen linearen Modell ist die sogenannte *polynomiale Regression*. Die Hypothese h_θ wird verallgemeinert, indem nicht nur lineare Abhängigkeiten von den Eingangsgrößen, sondern auch Polynome der Ordnung k mit in die Modellierung einbezogen werden. Eine Hypothese aus einem Polynom zweiter Ordnung mit $k = 2$ und $x_0 \coloneqq 1$ aufgebaut kann zum Beispiel wie folgt aussehen:

$$h_\theta(x) \coloneqq \theta_0 \cdot x_0 + \theta_1 \cdot x_1 + \theta_1 \cdot x_1^2 + \theta_2 \cdot x_2 + \theta_2 \cdot x_2^2 + \cdots. \qquad \textbf{3.17}$$

Die Optimierungsaufgabe zum Finden einer optimalen Kostenfunktion ist dementsprechend komplexer aufgebaut.

Ein wichtiger Faktor für das Gelingen der Konvergenz der Modellierung ist die Notwendigkeit eines in etwa gleich großen Wertebereiches der Eingangsgrößen (siehe [Ng17], Lecture 4). Dieses Verfahren wird engl. *feature scaling* genannt. Ein einfaches Verfahren, das Einpassen aller Werte in einen fixen Wertebereich, wird *Normalisierung* oder engl. *rescaling* genannt. Die Berechnung wird mit

$$\boldsymbol{d}_{\boldsymbol{fs\_norm}} = \frac{\boldsymbol{d} - min(\boldsymbol{d})}{max(\boldsymbol{d}) - \min(\boldsymbol{d})} \qquad \textbf{3.18}$$

ausgeführt. Bei der Normierung auf den Mittelwert spricht man von *Standardisierung* (siehe dazu auch [Ng17], Lecture 4), ausgeführt als

$$\boldsymbol{d}_{\boldsymbol{fs\_std}} = \frac{\boldsymbol{d} - mean(\boldsymbol{d})}{stddev(\boldsymbol{d})} \qquad \textbf{3.19}$$

Die Implementierung von Gl. **3.19** wird vektorisiert ausgeführt, die beteiligten Größen sind demnach Vektoren. Zur Iteration über die einzelnen Messwerte muss daher keine Schleifenstruktur in der Implementierung vorgesehen werden. Der Mittelwert der aktuellen Spalte $mean(\boldsymbol{d})$ wird vom jeweiligen Messwert $\boldsymbol{d}$ subtrahiert und schließlich durch die Standardabweichung $stddev(\boldsymbol{d})$ der aktuellen Spalte dividiert; sämtliche Werte des Vektors $\boldsymbol{d}_{\boldsymbol{fs}}$ am Ende der Verarbeitung sind damit in einem ähnlichen Wertebereich eingepasst.

Wie bereits eingangs formuliert, kann die reale Implementierung vom dargestellten Algorithmus in Form einer Verallgemeinerung abweichen. Die technischen Grenzen eines Algorithmus können in der Anwendung relativ leicht erreicht werden, zum Beispiel durch zu kleinem oder zu großem Eingangsdatenumfang. In Bezug auf das *batch gradient descent*-Verfahren stellt beispielsweise [WiMa03] dessen Effizienzprobleme dar. Für die Realisierung bietet sich allgemein die Verwendung von bereits auf Effizienz hin optimierter, erprobter Bibliotheken an. Für die beabsichtigte Umsetzung in der Sprache *R* [Rdev11] steht für die lineare Regression das Modul *glm* (Dokumentation siehe [Rdoc17]) zur Verfügung. Die Abkürzung *glm* steht für *generalisiertes lineares Modell* (engl. *generalized linear model*), das beschriebene *allgemeine lineare Modell* (engl. *general linear model*) ist als Spezialfall des *glm* aufzufassen.

Bevor an die Modellierung eines neuronalen Netzes herangegangen werden kann, ist noch die Erarbeitung weiterer Grundbegriffe notwendig: das Problem der Überanpassung, des sogenannten *Overfittings* und ein dafür möglicher Lösungsansatz, der Einsatz von *Regularisierung* und schließlich die Methode der Klassifikation mittels *logistischer Regression*.

Mit dem Begriff *Overfitting*, etwa mit *Überanpassung* übersetzt, wird nach [Ng17], Lecture 7, die Problematik der fehlenden Generalisierung des Lernalgorithmus auf neue Daten zusammengefasst. Das Modell liefert dabei für vorhandene Daten eine sehr gute Hypothese, scheitert aber bei der Vorhersage neuer Datensätze. Ein Ansatz zur Lösung ist die sogenannte *Regularisierung* (engl. *regularization*). Die Kostenfunktion der linearen Regression wird, unter Beibehaltung der Indizes des allgemeinen linearen Modells nach [Ng17], Lecture 7, wie folgt adaptiert:

$$J(\boldsymbol{\theta}) \coloneqq \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2\right]. \qquad \textbf{3.20}$$

Dieser Ansatz bewirkt, dass nun auch die Quadrate der Parameter θ_j, mit einem allgemeinen Faktor λ multipliziert, mit in die Minimierungsaufgabe aufgenommen werden. Unter Anwendung des Gradientenverfahrens ergibt sich die Rechenvorschrift für die Parameter θ_j bei *regularisierter linearer Regression* zu (adaptiert von [Ng17], Lecture 7)

$$\theta_j^{(k+1)} = \theta_j^{(k)} - \alpha^{(k)} \cdot \frac{1}{m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x_j^{(i)} + \lambda \cdot \theta_j\right], \qquad \textbf{3.21}$$

im Vergleich zu Gl. **3.16** geht der zu optimierende Parameter mit λ gewichtet in die Summe ein, wird also ebenfalls minimiert.

Die Einführung der *logistischen Regression* (engl. *logistic regression*) ist aus der Problemstellung der Datenklassifikation erwachsen ([Ng17], Lecture 6). Die Ausgangsgröße eines Modelles bekommt einen diskreten Wertcharakter, 0/1 oder falsch/wahr. Eine mögliche Hypothesefunktion mit einem definierten Wertebereich von $0 \leq g(z) \leq 1$ bietet die Sigmoidfunktion $g(z)$, unter anderem in [Ng17], Lecture 6, definiert als

$$g(z) = \frac{1}{1 + e^{-z}}. \qquad \textbf{3.22}$$

Die Kostenfunktion wird bei logistischer Regression nicht mehr durch die Methode der kleinsten Quadrate, sondern ist aufgrund des Verhaltens der Ableitung der Sigmoidfunktion $g(z)$ durch die Summation über Logarithmenfunktionen definiert, Details dazu siehe [Ng16, S.12,17].

Der Wertverlauf der Sigmoidfunktion ist in Abbildung 37 dargestellt. Für große, positive Werte von z wird $g(z) = 1$, für große, negative Werte von z wird $g(z) = 0$. Der Parameter z kann als Zahlenwert das Ergebnis eines Summationsprozesses über eine Reihe von Eingängen sein, der Funktionswert der Sigmoidfunktion $g(z)$ als Ausgang, der bei Überschreiten eines gewissen Schwellwertes dieser Summation sozusagen „feuert“, gesehen werden. Bei der logistischen Regression wird für die Hypothesefunktion $h_\theta(x)$ nun die Sigmoidfunktion $g(z)$ verwendet und auch die Kostenfunktion muss entsprechend adaptiert werden.

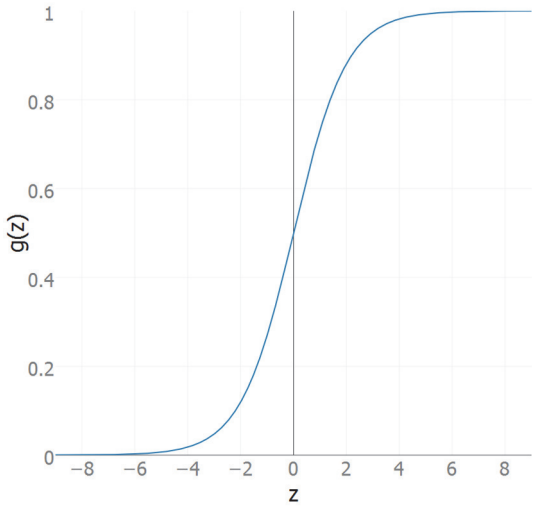

Abbildung 37: Sigmoidfunktion (eigene Darstellung)

In einem letzten Schritt wird schließlich die *regularisierte logistische Regression* definiert; kurz gesprochen wird die logistische Regression unter Verwendung der Sigmoidfunktion und einer komplexeren Kostenfunktion einer ähnlich in Gl. **3.22** beschriebenen Erweiterung mit einem Faktor λ zugeführt (Details dazu siehe [Ng17], Lecture 6). Mit diesen Ausführungen ist der „Werkzeugkasten" für die Einführung der neuronalen Modellierung gefüllt, die im folgenden Teil dieses Abschnitts vorgestellt wird.

Die Grundstruktur eines künstlichen Neurons ist übernommen von [Hayk98, S.33], die Notationen entsprechen der vorhergehenden Ausführungen zu linearer Regression angepasst und in Abbildung 38 dargestellt. Die Eingangsgrößen werden in einem ersten Schritt mit den Parametern $\theta_j, j = 1,\ 2, \ldots n$ entsprechend der Zahl der zur Verfügung stehenden Spalten der Eingangsgrößen, gewichtet und aufsummiert,

$$u = \theta_0 + \sum_{i=1}^{n} \theta_j \cdot x_j. \qquad \textbf{3.23}$$

Der Parameter θ_0 wird allgemein als „Bias-Knoten" bezeichnet und liefert dem neuronalen Netz im Rahmen des Lernvorgangs einen eventuell benötigten konstanten Faktor (ähnlich dem Faktor k in einer Geradengleichung $y = kx + d$). Die Bildung der Summe entspricht dem Vorgang der Erregung der Nervenzelle über von anderen Nervenzellen aufsetzenden Synapsen auf den Dendriten.

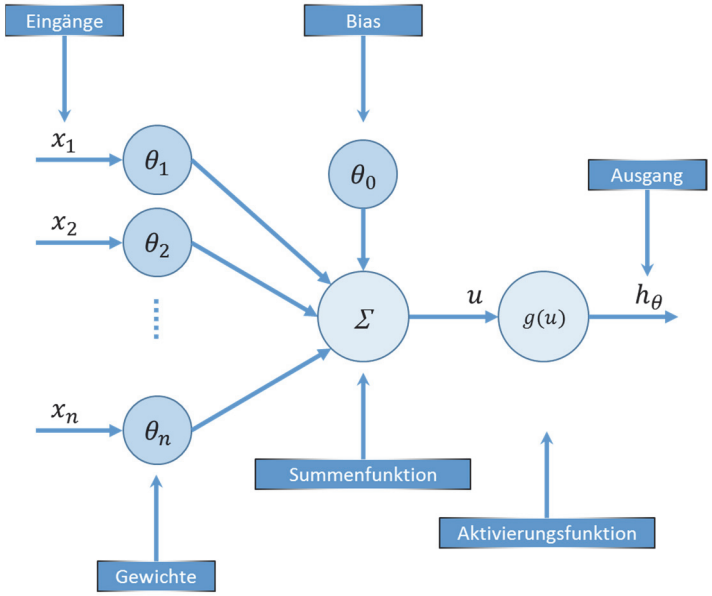

Abbildung 38: Nichtlineares Modell eines Neurons, eigene Darstellung nach [Hayk98, S.33], adaptiert

Im nächsten Schritt muss diese Summe einer Aktivierungsfunktion zugeführt werden. Damit wird das beschriebene Verhalten einer Nervenzelle, bei der Überschreitung eines gewissen Schwellwerts der Erregung, eine Aktion auszulösen, nachempfunden. Wird dieser Schwellwert überschritten (analoge Signalverarbeitung), soll das Neuron „feuern", den Ausgang also auf HIGH setzen (digitale Signalverarbeitung). Andrew Ng verwendet in [Ng17], Lecture 8, durchgängig auf die Sigmoidfunktion; Simon Haykin [Hayk98, S.35] stellt zwei weitere Aktivierungsfunktionen, den rechteckigen beziehungsweise stückweise linearen Verlauf vor. Im Rahmen dieser Arbeit wird die Sigmoidfunktion als Aktivierungsfunktion ausgewählt. Der Ausgang h_θ des Neurons ergibt sich damit zu

$$h_\theta = g(u) = \frac{1}{1+e^{-u}} = \left[1 + exp\left[-(\theta_0 + \sum_{i=1}^{n} \theta_j \cdot x_j)\right]\right]^{-1} . \qquad \textbf{3.24}$$

Der Ausgang des Neurons wird in einen beschränkten Wertebereich umgesetzt, $0 \leq g(u) \leq 1$, und dient so in weiterer Folge als Eingang nachfolgender Neuronen. Unter Zusammenfassung der Gewichte, der Summenfunktion und Aktivierungsfunktion in einem gemeinsamen Knoten und nicht dargestellten Bias-Eingängen kann ein künstliches neuronales Netz wie in Abbildung 39 gezeigt, aufgebaut werden (Nomenklatur wieder nach [Ng17], Lecture 7). Jeder Knoten rechts im Bild weist den internen Aufbau nach Abbildung 38 auf, in der Abbildung links verkleinert dargestellt. Jedem Knoten $a_i^{(j)}$ wird eine Matrix $\Theta^{(j)}$ entsprechend den Gewichten des Übergangs zwischen den Schichten, von Layer j auf Layer $j+1$, zugeordnet ([Ng17], Lecture 8). Das Netz im Beispiel weist vier Schichten (engl. *layer*) auf. Die erste Schicht, von links kommend, übernimmt die Eingangsgrößen, $a_j^{(0)} = x_j$. Die Schichten 2 und 3 werden als *hidden layer* bezeichnet und deren Überragungsfunktion ist in Gl. **3.24** dargestellt. Die Schicht 4 schließlich wird durch den Ausgangsknoten gebildet. Im Falle einer Klassifikationsaufgabe ist die Aktivierungsfunktion von $a_1^{(3)}$ wieder die Sigmoidfunktion, im Falle

einer Regressionsaufgabe kann der Ausgang fast direkt als skalierender Faktor eines Zielwertebereiches der Datenvorhersage verwendet werden.

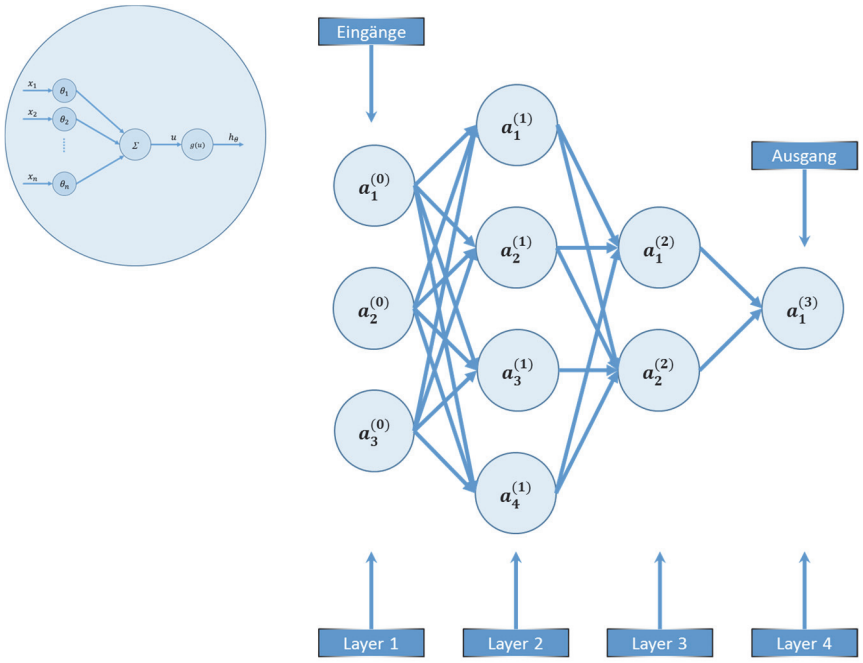

Abbildung 39: Beispiel eines künstlichen neuronalen Netzes (eigene Darstellung)

Der Signalfluss von links nach rechts, also die beschriebene Umsetzung der Eingangssignale in ein Ausgangssignal, wird auch *forward propagation* genannt. In der Nomenklatur von [Ng17], Lecture 8, spricht man von *Aktivierung* $\boldsymbol{a}^{(j)}$ bei insgesamt n Knoten im Layer j mit Bias $a_0^{(j)}$ in vektorieller Form dargestellt als $(\boldsymbol{a}^{(j)})^T = \left[a_0^{(j)}, a_1^{(j)}, \dots, a_n^{(j)}\right]^T$. Die Rechenvorschrift lautet

$$\boldsymbol{a}^{(j)} = h_\theta(u) = g\left(\boldsymbol{\Theta}^{(j-1)} \cdot (\boldsymbol{a}^{(j-1)})^T\right), \qquad \textbf{3.25}$$

mit der Sigmoidfunktion g(z), der Matrix der Gewichte $\boldsymbol{\Theta}$ und den Aktivierungen der Knoten der vorhergehenden Schicht, $\boldsymbol{a}^{(j-1)}$, transponiert eingezogen. Die in Abbildung 38 dargestellte Summenfunktion ist implizit durch die Multiplikation der Matrix der Gewichte mit den Aktivierungen des vorherigen Layers gegeben. Beim Modellieren der Gewichte, dem eigentlichen Lernvorgang, spricht man von Fehlerrückführung oder *backpropagation*. Bei der Fehlerrückführung wird ganz allgemein jedem Neuron k wie in [Hayk98, S.73] ein Fehlersignal $e_k(n)$ zugeordnet. In der vorletzten Schicht ergibt sich mit der Ausgangsgröße y_k als Zielwert

$$e_k(n) = d_k(n) - y_k(n). \qquad \textbf{3.26}$$

Der Term $d_k(n)$ bezeichnet das erwartete Ergebnis der Berechnung, der Term $y_k(n)$ gibt das tatsächliche Ergebnis des Trainingsdatensatzes wieder. Umgelegt auf die bekannte Nomenklatur, mit den Bezeichnungen aus Abbildung 39 und den Ausführungen in [Ng17], Lecture 9, ergibt sich für den Fehler $\boldsymbol{\delta}^{(j)}$ im letzten Layer j vor dem Ausgangslayer des neuronalen Netzes (alle Terme in vektorieller Darstellung)

$$\boldsymbol{\delta}^{(j)} = \boldsymbol{a}^{(j)} - \boldsymbol{y} \text{ mit } \boldsymbol{a}^{(j)} = h_\theta(\boldsymbol{a}^{(j-1)}), \quad \textbf{3.27}$$

hier kommen also die wahren Werte $\boldsymbol{y}$ des Trainingsdatensatzes zur Anwendung. Für die vektorwertige Formulierung für den Operator der elementweisen Multiplikation der Matrizen wird das sogenannten Hadamard-Produkt wie folgt definiert: $a = b \circ c: a_i = b_i \cdot c_i$. Weiters wird mit $g'(u)$ die Ableitung der Sigmoidfunktion bezeichnet. Damit ergeben sich die Fehlerterme der weiteren Schichten, im Beispiel für Layer $j-1$, adaptiert aus [Ng17], Lecture 9, zu

$$\boldsymbol{\delta}^{(j-1)} = \left[(\boldsymbol{\Theta}^{(j-1)})^T \cdot \boldsymbol{\delta}^{(j)}\right] \circ \left[g'(\boldsymbol{\Theta}^{(j-2)} \cdot \boldsymbol{a}^{(j-2)}.)\right]. \quad \textbf{3.28}$$

Die Berechnung des Fehlers $\boldsymbol{\delta}$ eines gewählten Layers setzt sich demnach zusammen aus Gewichten der aktuellen Schicht, verknüpft mit dem Fehler der nächsten Schicht, elementweise multipliziert mit der Ableitung der Sigmoidfunktion mit eingesetzten Gewichten und Aktivierungen aus der Schicht davor. Die Terme des nächsten Layers j wirken sich dabei auf die Terme des aktuellen Layers $j-1$ aus, man spricht in diesem Zusammenhang von *backpropagation*. Die Fehlerterme selbst werden im nächsten Schritt wieder einer *forward propagation* zugeführt, so kann beispielsweise der Fehlerterm $\delta_4^{(1)}$ des Knoten $a_4^{(1)}$ mit den Gewichten $\Theta_{k,l}$ wie folgt berechnet werden:

$$\delta_4^{(1)} = \Theta_{14}^{(1)} \cdot \delta_1^{(2)} + \Theta_{24}^{(1)} \cdot \delta_2^{(2)}. \quad \textbf{3.29}$$

Der Lernvorgang lässt sich wie folgt zusammenfassen: Die Eingangsparameter des Trainingsdatensatzes werden an das Netz angelegt und entsprechend der Sigmoidfunktion mit den jeweiligen Gewichten an den Ausgang übertragen. Der Ausgang wird mit den gewünschten Ausgangswerten des Trainingsdatensatzes verglichen, der Fehler berechnet und dieser Fehler wieder an den Eingang zurück übertragen. Das Ziel des Algorithmus ist die Minimierung der Fehlerterme δ durch Korrektur der eingesetzten und zu Beginn beispielsweise mit Zufallszahlen befüllten Gewichte. Die dabei von [Ng17], Lecture 7, eingesetzte Kostenfunktion $J(\Theta)$ mit den Gewichten Θ ist formal ähnlich jener bei der bereits um Regularisierung erweiterten *logistischen Regression*. Die Ableitung der Kostenfunktion errechnet sich nach [Ng17], Lecture 7, und zur Vereinfachung noch ohne Regularisierung zu

$$\frac{\partial}{\partial \boldsymbol{\Theta}^{(l)}} J(\boldsymbol{\Theta}) = \boldsymbol{a}^{(j)} \boldsymbol{\delta}^{(l+1)}. \quad \textbf{3.30}$$

Es wird damit also die Verknüpfung der Gewichte, den Fehlertermen und der Kostenfunktion eingeführt. Für die Minimierungsaufgabe, die Berechnung der minimalen Kostenfunktion, kann wieder das Gradientenverfahren in Form des LMS-Verfahrens (siehe Gl. **3.11**) herangezogen werden; in der Literatur wird von der *Delta-Regel* ([Hayk98, S.185,Gl. 4.12] gesprochen.

Ähnlich der Vorgangsweise beim *allgemeinen linearen Modell*, bietet sich auch im Fall der Modellierung eines neuronalen Netzes die Verwendung vorgefertigter Bibliotheken an. [GüFr10] haben das Paket *neuralnet* [FrGü16] präsentiert, in dem eine Reihe gängiger *ANN*-Algorithmen implementiert und für die direkte Verwendung in eigenen Anwendungen in der Sprache *R* zur Verfügung stehen. Die Modellierung wird über den Funktionsaufruf gestartet und bei Konvergenz des Modells kann das Ergebnis für die Vorhersage von Zeitreihen verwendet werden. Die Anwendung der Modellierungsalgorithmen auf erfasste Messdaten wird im nächsten Teil der Arbeit als konkrete Implementierung der beschriebenen Konzepte vorgestellt.

4. Implementierung

Die Erweiterung des bestehenden Parameterumfangs durch Entwicklung und Implementierung einer hochintegrierten Sensortechnik-Einheit zur Erfassung meteorologischer Umgebungsbedingungen war ein Forschungsschwerpunkt im Projekt „NaWas“ im Berichtsjahr 2016 und soll in dieser Arbeit im Detail beschrieben werden. Der Aufbau des Prototyps und dessen Einbindung konnten erfolgreich abgeschlossen werden, das Modul Ende Juni 2016 seinen Betrieb aufnehmen und mit der Datenerfassung begonnen werden. Basierend auf den bereits vorgestellten Konzepten für den Hardwareaufbau, wird im Rahmen dieser Arbeit im Abschnitt 4.1 die Realisierung des Environmental Monitorings auf der Messstation an der Raab vorgestellt. Die für die Datenerfassung notwendige Software, sowohl am *Raspberry Pi* als auch auf Seite von *i*^{TUW}*mon* ist Inhalt des Abschnitts 4.2. Die Umsetzung des maschinellen Lernens mittels der Programmiersprache R wird in Abschnitt 4.3 beschrieben. Sämtliche Programmlistings dienen der Veranschaulichung ausgewählter Kernaufgaben der beteiligten Programme und geben nur einen Teil der für die Umsetzung dieser Arbeit notwendigen Software wieder.

Abbildung 40: Implementierungsschritte des Environmental Monitorings ([Mult15], eigene Darstellung)

4.1 Realisierung der Messhardware und Versuchsaufbau

Der in Abbildung 40 dargestellten Schritte der Implementierung basieren auf einer Reihe von Sensoren, die bereits in Tabelle 3 auf Seite 42 vorgestellt wurden. Als Bezugsquelle wurde die Firma *Adafruit* aufgrund der guten Verfügbarkeit der Sensoren und der durchgängig verfügbaren Anbindung über I^2C-Schnittstellen ausgewählt [Adaf17]. Besonders hervorzuheben sind die Charakteristika des optischen Farbsensors *TCS34725*, der *SI1145* für UV-Index, sichtbares Licht und IR-Strahlung und der SiC(Siliziumcarbid)-Photodiode *GUVA-S12SD*. Wie in den Grundlagen zur Gewässergüte in Abschnitt 1.1 dargelegt, beruht die Photosynthese und damit letztlich die Sauerstoffkonzentration der Gewässer maßgeblich auf der Sonneneinstrahlung, welche daher durch den Prototypenaufbau erfasst werden soll. Auf die Erfassung der globalen Sonneneinstrahlung durch ein Pyranometer wurde, unter Maßgabe des Entwicklungsziels der Verwendung ausschließlich hochintegrierter, vergleichsweise kostengünstiger Sensoren, verzichtet und dafür, soweit möglich, die Erfassung mittels mehrfach redundant aufgebauter Bausteine unterschiedlicher Technologien vorgesehen, die im Folgenden beschrieben werden.

Das Blockdiagramm des *TCS34725* ist in [Ag16, S.3] dargestellt. Der einstrahlende infrarote Anteil des Lichtes wird mit einem Filter geblockt und die verbleibende Bestrahlung wird durch einen roten, grünen und blauen Farbfilter geleitet. Die spektrale Empfindlichkeit, in [Ag16, S.11] dargestellt, zeigt, dass der Sensor im interessanten Wellenbereich von rund 400 bis 650 nm ein gutes Ansprechvermögen aufweist. Das eingestrahlte Licht führt zu einem Stromfluss in den vier Photodioden (der Diode für „clear“ ist kein Filter vorgeschaltet) und dieser wird jeweils von einem *ADC* unter konfigurierbarer Integrationszeit erfasst. Die Messwerte der Beleuchtungsstärke, nach Farbe aufgetrennt, stehen als 16-Bit-Datenwort äquivalent einem Stärkemaß im Wertebereich 0 (dunkel) bis 65535 (hell) zur Verfügung. Durch Anpassung der Integrationszeit ist ein großer Dynamikbereich des Sensors gegeben. Der ähnlich aufgebaute Baustein *SI1145* beruht auf zwei Photodioden, jeweils eine für sichtbares und eine für infrarotes Licht. Besonders interessant für die Anwendung ist die Ausgabe des kalibrierten *UV*-Index als Zahlenwert im Bereich 1 bis 11. Nach [LuPr06, S.6] wird der *UV*-Index definiert als nach Schädigungspotential gewichtete, durchschnittliche UV-Einstrahlung in Wm^{-2} und ist ein von der Weltgesundheitsorganisation WHO definiertes Maß für die Belastung durch hautschädigende *UV*-Strahlung (siehe auch [Isoi99]). Ein weiterer optischer Sensor auf Basis der für ultraviolette Strahlung besonders empfindlichen Keramik Siliziumcarbid, ist der Baustein *GUVA-S12SD*. Für die Einbindung ist die Verwendung eines externen *ADC* notwendig. Das Sensorboard verfügt über einen Vorverstärker, der das Stromsignal in ein Spannungssignal in der Größenordnung von fünf Volt umsetzt. Durch die im Datenblatt [Roit15] angegebene, annähernd lineare Charakteristik kann von der Spannung am *ADC* auf den strahlungsinduzierten elektrischen Strom und in weiterer Folge auf den *UV*-Index und die Bestrahlung mit *UV*-A-Strahlung in $mWcm^{-2}$ zurückgerechnet wurde (*UV-A* bezeichnet hier „nahes *UV*-Licht“ im Wellenbereich 380 bis 315 nm). Als Vorgriff zur softwaremäßigen Einbindung sei auf Abbildung 41 verwiesen.

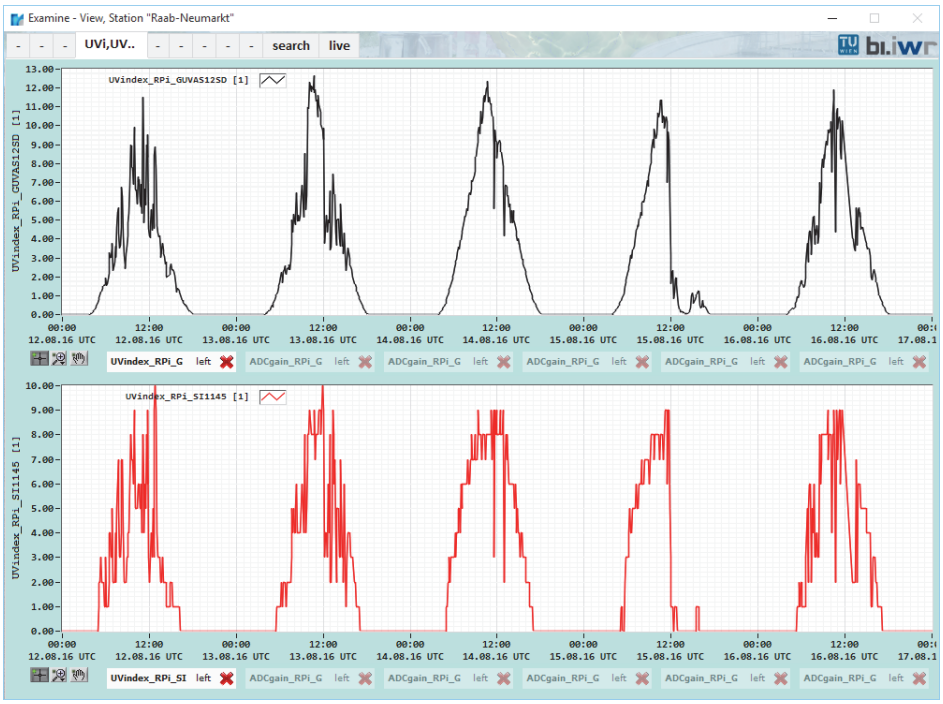

Abbildung 41: Vergleich der gemessen UV-Indizes (eigene Darstellung)

Die Messwerte des UV-Index des *GUVA-S12SD* weisen kontinuierlichen Wertcharakter auf während der mit dem *SI1145* aufgenommene Index aus scheinbar wertdiskreten, auf volle Einerstelle gerundeten Zahlen besteht. Abgesehen davon weisen beide Wertverläufe sehr gleichförmigen Charakter auf, auch bezüglich der kurzfristigen Einbrüche der Sonnenbestrahlung, zum Beispiel in vorüberziehenden Wolken begründet.

Zusätzlich zur optischen Erfassung der Bestrahlungsstärke kommt, wie in Tabelle 3 angeführt, auch eine Infrarot-Thermosäule zur Messung der Wolkentemperatur zum Einsatz. Nach Untersuchungen von [CWBD98] und den dort angeführten Vorarbeiten ist die Messung der Absorption im infraroten Bereich bei Wellenlängen zwischen etwa 5 bis 14 µm hinreichend geeignet zur Erkennung und dem Monitoring der Bewölkung. Der eingesetzte Sensor *TMP006* (Datenblatt siehe [Texa15]) weist seine höchste Empfindlichkeit im Bereich 6 bis 13 µm auf und ist daher für das Bewölkungs-Monitoring durch berührungslose Temperaturmessung geeignet.

Die Sensoren wurden für eine erste Inbetriebnahme auf einem Prototypen-Steckbrett aufgebaut. Nach Konfiguration und Adaptierung der Treiberschnittstellen in *Python* wurde der Hardwareaufbau auf eine Lochrasterplatine übertragen und mittels Flachbankkabel elektrisch angebunden. Neben der Datenübertragung mit I^2C wird auch die Versorgungsspannung der Sensorboards und des externen *ADC* über die 3,3 V-Schiene des *Raspberry Pi* bewerkstelligt. Sämtliche nicht-optischen Sensoren wurden mit weißem Papier zum Schutz vor zusätzlicher Erhitzung durch direkte Sonneneinstrahlung abgedeckt, wie in Abbildung 42 dargestellt.

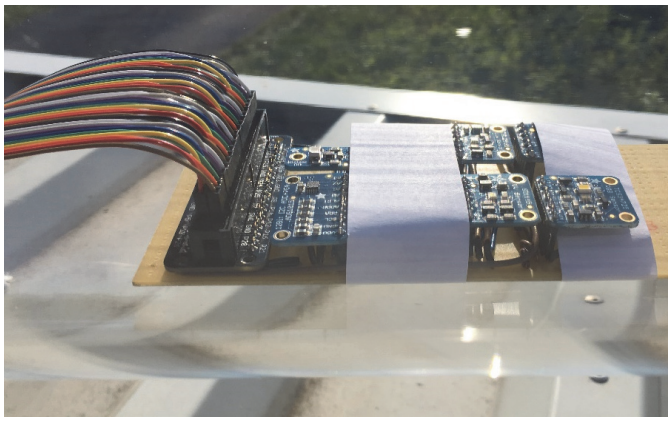

Abbildung 42: Messaufbau Sensorik (Foto: Autor)

Für den wetterfesten Aufbau am Dach der bestehenden Messstation ist ein für UV- und Infrarotlicht weitestgehend transparentes Gehäuse notwendig, ein entsprechend großes Luftvolumen soll als Temperaturpuffer dienen. Die Firma Bilek+Schüll GesmbH wurde mit der Fertigung eines Acrylglas-Rohr XT mit einem Außendurchmesser von 200 mm, 1 m Länge und der Fräsung für einen Dichtring des abnehmbaren Deckels beauftragt, welches in der Werkstätte des Instituts für Wassergüte, Ressourcenmanagement und Abfallwirtschaft mit einer wasserdichten Kabeldurchführung für die Anbindung mit strukturierter Verkabelung versehen wurde. Die Röhre wurde mit gummierten, aus Edelstahl ausgeführten Rohrschellen und zwei Verschraubungen am Dach der Messstation befestigt. Die ins Stationsinnere führenden Gewindestangen wurden mit Bitumenmasse gegen Wettereinflüsse abgedichtet.

Abbildung 43: Messaufbau am Dach der Station (Foto: Autor)

Abbildung 43 zeigt den fertiggestellten Messaufbau direkt vor dem Beginn der Datenerfassung; im rechten Teil der Abbildung sind die eingelegten Päckchen, gefüllt mit Silica-Gel zur Verhinderung der Kondensation von Wasserdampf, auf der Innenseite zu erkennen. Das röhrenförmige Gehäuse sorgt außerdem dafür, dass bei Regenwetter oder Kondensation auf der Außenseite die Wassertropfen ungehindert abfließen können und damit nicht direkt oberhalb der optischen Sensoren verweilen und so die Messung nicht beeinflussen.

4.2 Datenerfassung und Einbindung

Die Datenerfassungssoftware wird durch *Python*-Programme am *Raspberry Pi* und der Gestaltung einer zusätzlichen Datenquelle in $i^{TUW}mon$ realisiert. Aufbauend auf den Konzepten von Abschnitt 3.2 wurden zwei wesentliche Programmteile am *Raspberry Pi* implementiert (siehe Abbildung 44): Datenerfassung und Zugriff auf die I^2C-Treiberschnittstelle mit *sens.py* und Betrieb des ModbusTCP-Servers für die Datenabfrage aus $i^{TUW}mon$ mit *p.py*. Die Einbindung der Kamera und die zeitgesteuerte Aufnahme und Ablage der Bilddaten erfolgt mittels des Tools *PiCamera* von [Jone16]; das zugehörige Python-Programm *cam.py* wird nach dem Systemstart als Dienst aufgerufen, läuft in einer Endlosschleife und legt ein Bild pro Minute auf der Speicherkarte des *Raspberry Pi* ab.

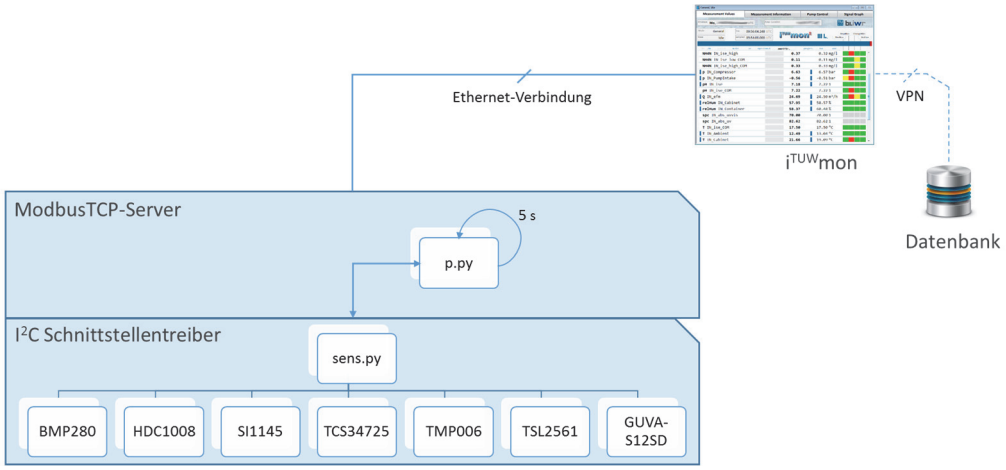

Abbildung 44: Umsetzung des Softwarekonzepts zur Sensoreinbindung (eigene Darstellung)

Das Programm *sens.py* kommuniziert mit den jeweiligen, vom Hersteller der Sensorboards gelieferten *I²C*-Treibern und abstrahiert die Daten in ein geeignetes Übergabeformat. Der Kern der Funktionalität ist im folgenden *Python*-Code Listing 1 am Beispiel der Einbindung des *HDC1008*-Sensors zur Erfassung von Temperatur und relativer Luftfeuchtigkeit dargestellt.

```
01          import Adafruit_HDC1008 as HDC1008_conn
02          def HDC1008(par):
03             hHDC1008 = HDC1008_conn.HDC1008() # open connector
04             if par == "t": # return air temperature
05               return hHDC1008.readTemperature()
06             elif par == "hum": # return air humididty
07               return hHDC1008.readHumidity()
08             elif par == "all": # return all parameters as cs-list
09               return [hHDC1008.readTemperature(),hHDC1008.readHumidity()]
10             else:
11               return 9999 # all other cases
```

Listing 1: Python-Code zur Einbindung des HDC1008-Bausteins zur Erfassung der Lufttemperatur und relativen Luftfeuchtigkeit

Die Verbindung zum Schnittstellentreiber von Adafruit [Adaf17] wird über einen sogenannten Connector realisiert und die Treibereinbindung, mit dem gewünschten Parameter in Klammern, angegeben und aufgerufen. Je nach Parameter wird die entsprechende Routine im Treiber aufgerufen und der ermittelte Messwert mit **`return`** dem aufrufenden Programm übergeben. Die Implementierung des ModbusTCP-Servers beruht auf der Bibliothek *PyModbus* und der Adaption des Programmbeispiels *Updating Server Example* (siehe [Coll09]). Für die Implementierung wurde das Konzept des *Updating Writers* gewählt. Die aktuellen Messdaten werden durch einen im Hintergrund laufenden Thread abgefragt und zyklisch alle fünf Sekunden in den Kontext des ModbusTCP-Servers eingebracht. Die wesentliche Adaptierung des Beispiels wird in Listing 2 gezeigt.

Der *ModbusTCP*-Server stellt die Daten über eine Reihe von Registern zur Verfügung, welche in den Zeilen 12-21 parametriert werden. In Zeile 22 wird die Länge des insgesamt zu erwartenden Datensatzes in ein Register eingetragen. Die Datenabfrage durch *i^TUWmon* liest dieses Register zuerst und in

einem zweiten Schritt werden die Messdaten, deren Anzahl durch den Inhalt dieses Registers parametriert wird, eingelesen. Das erste Messdatenfeld wird mit der aktuellen *UTC*-Zeit des *Raspberry Pi* befüllt (Zeile 24), gefolgt von operationalen Daten wie der Laufzeit seit Systemstart, der *CPU*-Auslastung und der *CPU*-Temperatur (Zeilen 25 bis 27). Nach diesen Feldern werden, im Listing ab Zeile 28, sämtliche Messergebnisse der Reihe nach in die Struktur eingebracht. Die Zahlen sind dabei nach IEEE 754 in 32-Bit Float-Darstellung codiert [Ieee08].

```
01          import sens as s # import sens.py for interfacing
02
03          # ...
04
05          # program based on the Updating Server Example, BSD-License
06          # all credits to Galen Collins
07          # http://pymodbus.readthedocs.io/en/latest/examples/updating-server.html
08          # https://github.com/riptideio/pymodbus
09          #
10
11          def updating_writer(a):
12              log.debug("updating the context")
13              context  = a[0]
14              register = 3
15              slave_id = 0x00
16              address  = 0x10
17              channelnum = 23
18          # define number of registers to read from iTUWmon
19              fulllen = 19 + channelnum*2
20              values   = context[slave_id].getValues(register, address, count=50)
21              builder.reset()
22              builder.add_32bit_float(fulllen)
23          # operational Infos
24              builder.add_string(s_timestamp())
25              builder.add_32bit_float(f_uptime())
26              builder.add_32bit_float(f_cpu_p())
27              builder.add_32bit_float(f_cpu_t())
28          # EM-channels starting here
29              BMP280 = s.BMP280("all") # t,p,alt,sealevel
30              builder.add_32bit_float(BMP280[0])
31              builder.add_32bit_float(BMP280[1])
32              builder.add_32bit_float(BMP280[2])
33              builder.add_32bit_float(BMP280[3])
34              HDC1008 = s.HDC1008("all") # t, hum
35              builder.add_32bit_float(HDC1008[0])
36              builder.add_32bit_float(HDC1008[1])
37
38          # ...
```

Listing 2: Python-Code des ModbusTCP-Servers und Einbindung der Messdaten

Die Softwaremodule auf Seiten von *i*^{TUW}*mon* sind in Abbildung 45, die Benutzeroberfläche zum Debugging und ein Ausschnitt des Quellcodes des Moduls sind in Abbildung 46 dargestellt. Ein *LabVIEW*-Programm wird auch *VI* (engl. *Virtual Instrument*) genannt und das sogenannte *Front Panel* ist die Benutzerschnittstelle eines *VI*, die Bedienoberfläche des in Software nachempfundenen Messinstruments. Das mit „1“ bezeichnete Feld in Abbildung 45 zeigt die Rohdaten der Halteregister des ModbusTCP-Servers am *Raspberry Pi*. Das Feld mit „2“ bezeichnet zeigt die entsprechend nach Codierung in IEEE 754 errechneten Daten, in *LabVIEW* als Datentyp *DBL* (engl. *double* für Gleitkommazahlen doppelter Genauigkeit) geführt.

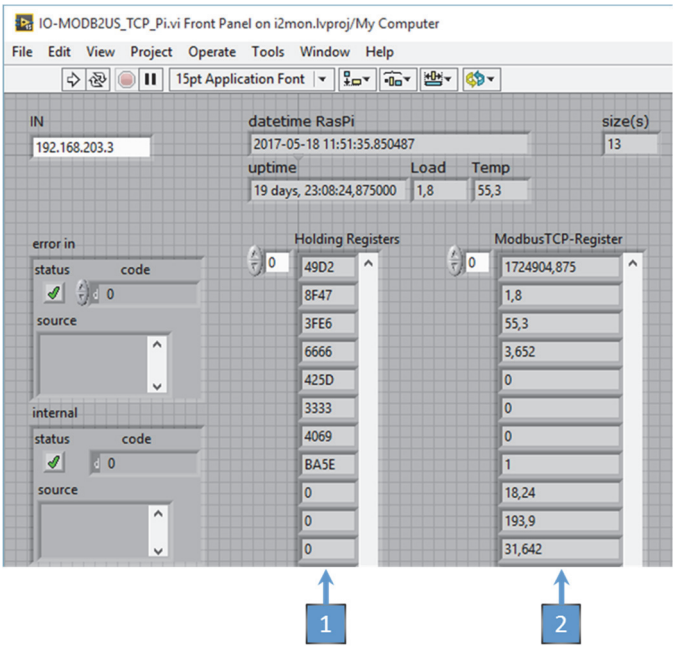

Abbildung 45: ModbusTCP-Read in *i^TUW mon*, FP (engl. *Front Panel*) (eigene Darstellung)

Der zugehörige Quellcode zum *Front Panel* ist in Abbildung 46 dargestellt. Der mit „1“ bezeichnete Bereich ist zuständig für das Öffnen des TCP-Sockets auf Port 5020. Zuerst wird die Anzahl der einzulesenden Datenkanäle als 2-Byte Zahl vom ModbusTCP-Halteregister 16 gelesen. Im nächsten Schritt wird diese Anzahl an Parametern, beginnend mit Register 18, eingelesen.

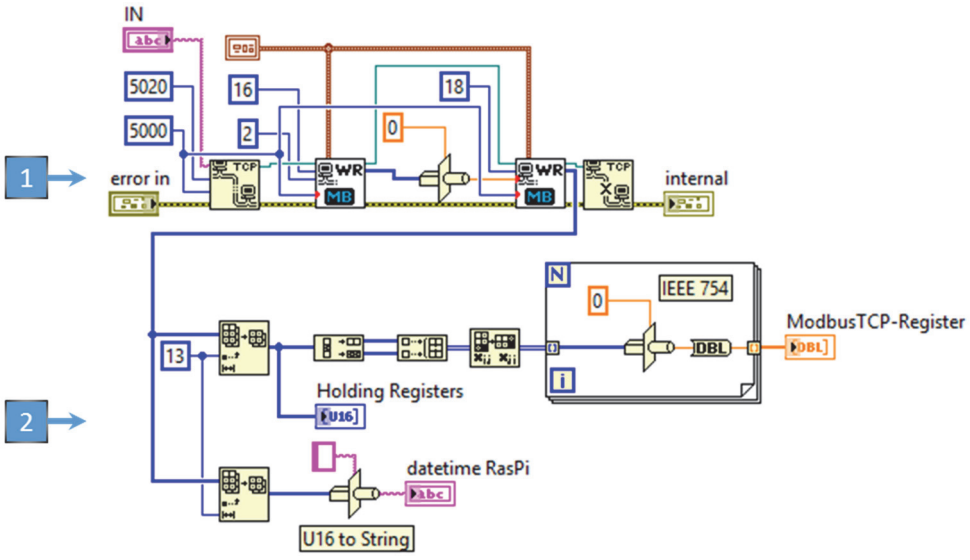

Abbildung 46: ModbusTCP-Read in *i^TUW mon*, BD (engl. *Block Diagram*) (eigene Darstellung)

Der mit „2“ bezeichnete Bereich setzt die empfangenen Daten in das gewünschte Zielformat um. Dafür werden jeweils zwei Byte nicht-vorzeichenbehafteter Ganzzahlen (engl. *UINT*, *unsigned integer*)

in den Halteregistern in ein 4-Byte-Feld eingeordnet und in eine Gleitkomma-Zahlendarstellung umgewandelt. Die Messdaten stehen damit, gemeinsam mit allen anderen Datenquellen der Station, zur Weiterverarbeitung in übergeordneten Modulen zur Verfügung. Die Datenerfassung mit *i*^{TUW}*mon* wird an der Raabstation typischerweise mit zehn Messzyklen pro Stunde Laufzeit ausgeführt. Zusätzlich dazu werden die Sensorwerte bei jedem Sampling über die *I*^{2}*C*-Schnittstelle in einer *SQLite*-Datenbank [Hipp17] direkt am Speicher des *Raspberry Pi* für künftige Verwendung im, für die Datenerfassung der Gewässergüte vergleichsweise hochauflösenden, Raster von einer Messung alle fünf Sekunden abgelegt.

4.3 Umsetzung ausgewählter Algorithmen des maschinellen Lernens

Als Programmiersprache zur Umsetzung der Algorithmen wurde R [Rdev11] gemeinsam mit der Entwicklungsumgebung (*IDE,* engl. *Integrated Development Environment*) *RStudio* von [Rstu17] ausgewählt. Zur Lösung der Aufgabestellung sind fünf Skripte zuständig: *read.R* dient dem Laden und Übersetzen der Daten aus der zentralen Messdatenbank in das R-Zieldatenformat und *preprocessing.R* ist zuständig für die Entfernung von Ausreißern und nicht erwünschten Zeitreihen. Die eigentliche Modellierung des generalisierten linearen Modells und des künstlichen neuronalen Netzes findet im Modul *model.R* statt. Im Skript *fun.R* sind alle die für die Datenverarbeitung notwendigen (Hilfs-)Funktionen definiert. Im letzten Schritt wird mit dem Skript *predictor.R* die Datenvorhersage und die Eignung der Modellierung untersucht. Für die Modellierung wird als relevanter Bereich der Quelldaten der Zeitraum zwischen 1. Juli 2016 und 20. April 2017 festgelegt. In diesem annähernd 300 Tage umfassenden Zeitraum ist für die erste Evaluation mit einer bereits größeren Variabilität der Messdaten zu rechnen, um für die Modellierung möglichst repräsentative Werteverläufe der beteiligten Messkanäle, auch durch Wirkung verschiedener Jahreszeiten, zur Verfügung zu haben.

Nach dem Laden aller Kanäle des Environmental Monitoring-Aufbaus unter Berücksichtigung der Plausibilitätsbewertung (siehe Abschnitt 2.3) werden in einem ersten Bearbeitungsschritt sämtliche Einträge mit undefinierten Werten *NaN* (engl. *Not a Number*) entfernt. Diese Einträge kommen durch fehlende Messdaten zum jeweiligen Messzeitpunkt zustande, beispielsweise, wenn während einer Wartung oder durch einen Sensorausfall das Gerät nicht ansprechbar ist. Es wird hierbei deutlich, dass dieser Verarbeitungsschritt sehr große Auswirkungen auf den Umfang des Datenbestandes hat: Schon ein einziger *NaN*-Wert eines Messkanals zum Messzeitpunkte kann zum Verlust der gesamten Messdatenzeile, über alle Datenquellen hinweg, führen. Der Ansatz, die *NaN*-Werte durch den Median der Messdaten des betreffenden Kanals zu ersetzen, ist für den betroffenen Messkanal nur bedingt hilfreich, kann dadurch aber zumindest die Daten der anderen Quellen erhalten. Statistisch gesehen wird der Messdatenreihe damit keine weitere, neue Information hinzugefügt, wie es ein echter Messwert bewerkstelligen würde. Die Mediane machen die Daten für die Modellierung damit nicht „interessanter“, die in den Messdatenreihen versteckte Information, der Verlauf der Signale basierend auf bisher unbekannten Zusammenhängen, ist allerdings maßgeblich für eine gute, mit kleinen Abweichungen behaftete Modellierung. Aus diesem Grund wird im ersten Ansatz in dieser Arbeit explizit auf das Ersetzen der *NaN*-Werte durch den Median der Reihe verzichtet. Der Datenumfang fällt entsprechend geringer aus, gleichzeitig ist aber gewährleistet, dass in jeder Zeile, also zu jedem Messzeitpunkt, von allen beteiligten Kanälen auch valide Messdaten vorliegen.

Die nächste Aufgabe der Daten-Vorverarbeitung ist das Erkennen und das Entfernen von Ausreißern unter Anwendung schärferer Kriterien als sie bei der bisher verwendeten Plausibilitätsprüfung angewendet werden. Für die Ausreißererkennung wurde das Paket *outliers* von Lukasz Komsta verwendet [Koms06, S.11]. Die Selektion wird durch eine polynomiale Regression dritter Ordnung auf Basis ausgewählter Punkte statistischer Tests durchgeführt. Diese Tests beruhen im Wesentlichen auf der Annahme einer den Daten zugrundeliegenden Verteilungsfunktion, der Suche nach Extremwerten im Datensatz und einem anschließenden Vergleich der Werte untereinander, um schließlich eine auf Quantilgrenzen basierte Entscheidung über das Vorliegen eines Ausreißers zu treffen. Ein Vorteil des gewählten Pakets ist die Möglichkeit, Ausreißer durch den Median der aktuellen Spalte der Messdaten zu ersetzen, um so die Grundgesamtheit bei eventuell nur wenigen vorliegenden Datensätzen nicht weiter einschränken zu müssen. Die letzte Aufgabe der Vorverarbeitung ist das sogenannte *feature scaling*, im konkreten Beispiel als Normalisierung ausgeführt (Normierung auf Maxima und Minima, siehe Gl. **3.18** in Abschnitt 3.3).

Nach dem Einlesen und Vorbereiten der Daten wird im Modul *model.R* die tatsächliche Modellierung ausgeführt. Die folgenden Ausführungen beruhen auf der Ausarbeitung von [Alic15] und sind für die gegenständliche Aufgabenstellung adaptiert worden.

Die Daten des überwachten Lernens werden aus den gemessenen Daten gewonnen, die vorhandenen Zeitreihen werden in einem typischen Verhältnis 70:30 in einen Trainings-Datensatz und einen Test-Datensatz für die Modellierung aufgeteilt. Die Aufteilung beruht in der Literatur meist auf Erfahrungswerten, [Reit10] bietet einen guten Einstieg in diese Problematik. Die Zeilenindizes der Datensätze werden dabei mit der R-Funktion `sample(...)` zufällig ausgewählt. Auf die Modellvalidierung, der Test der Eignung des Modells an einem bestimmten Teil der zur Verfügung stehenden Daten, wird im ersten Ansatz, aufgrund der bereits eingeschränkten Zahl an Messdaten, verzichtet.

Im ersten Schritt wird das generalisierte lineare Modell [Rdoc17] erstellt. Das Listing 3 zeigt den Quellcode in *R*.

```
1           f <- as.formula(paste((modch), "~.", sep = ""))
2           lm <- glm(f, data=train, family=gaussian)
```

Listing 3: R-Code der generalisierten linearen Modellierung

Die mit `modch` bezeichnete Variable enthält den Namen der zu modellierenden Größe, in diesem Fall die Konzentration des gelösten Sauerstoffs. Das Ergebnis der linearen Modellierung kann mit Eingabe von `summary(lm)` aufgerufen werden, eine beispielhafte Ausgabe des Befehls ist im Listing 4 dargestellt.

```
> summary(lm)

Call:
glm(formula = f, family = gaussian, data = train)

Deviance Residuals:
       Min          1Q      Median          3Q         Max
-3.6882770  -0.7878476   0.0140963   0.8035450   2.9534580

Coefficients:
                              Estimate     Std. Error  t value               Pr(>|t|)
(Intercept)                7.08210949111  0.52007593625 13.61745 < 0.000000000000000222 ***
p_air_rpi_bmp280_baro      0.00183382172  0.00054330168  3.37533             0.00074397 ***
relhum_rpi_hdc1008_relhum  0.05854470495  0.00226302365 25.87013 < 0.000000000000000222 ***
t_rpi_bmp280_air           0.00448783377  0.01424862416  0.31497             0.75280303
t_rpi_tmp006_die          -0.11959008167  0.01466566633 -8.15443 0.0000000000000045879 ***
t_rpi_tmp006_obj           0.00287587013  0.00166732100  1.72484             0.08462900 .
vis_rpi_si1145             0.00356194059  0.00009226239 38.60664 < 0.000000000000000222 ***
days                      -0.00938003173  0.00170216375 -5.51065 0.0000000378925310239 ***
hours                     -0.00287258015  0.00304675880 -0.94283             0.34582133
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 1.137876258)

    Null deviance: 15992.6632  on 4208  degrees of freedom
Residual deviance:  4779.0803  on 4200  degrees of freedom
AIC: 12499.264

Number of Fisher Scoring iterations: 2
```

Listing 4: Ergebnis der linearen Modellierung in R

Die Spalte `Estimate` bezeichnet die Koeffizienten, mit denen die Werte der Spalten, im linken Teil dargestellt, in das lineare Modell eingehen. Die Kennzeichnung der Zeilen, ganz rechts im Listing, gibt Auskunft über die Signifikanz des Parameters. Im gezeigten Beispiel ist die Eingangsgröße `t_rpi_bmp280_air`, der Luftdruck mit dem *BMP280*-Sensor erfasst, für das Modell nicht signifikant. Die Größe `rpi_tmp006_die`, also die Chiptemperatur des *TMP006*-Sensors, geht dafür signifikant in das Modell ein (und zwar mit dem Koeffizienten von rund `-0.119`). Hilfreiche Hintergründe zu den Interna der Modellierung und den mathematischen Zusammenhängen gibt [Geye03]. Die Vorhersage des Zielparameters unter Verwendung des generalisierten linearen Modells wird mit der Funktion `predict(lm, data)` bewerkstelligt, wobei die Struktur `data` die Daten in gleicher Struktur wie bei der Modellierung enthält.

Für die Modellierung eines neuronalen Netzes sind wichtige Parameter im Vorfeld zu klären. Zuerst stellt sich die Frage nach der Anzahl der verdeckten Schichten (engl. *hidden layer*). Nach [Heat08, S.158] sind zwei verdeckte Schichten ausreichend, um jeden beliebige Werteverlauf darstellen zu können. Die Anzahl der Neuronen pro verstecktem Layer wird nach [Mast33, S.176] für ein vierschichtiges Netzwerk (Eingangsschicht, zwei versteckte Schichten, Ausgangsschicht) nach der sogenannten *geometric pyramid rule* wie folgt errechnet:

$$r = \sqrt[3]{\frac{n}{m}}, \qquad \textbf{4.1}$$

$$NHID_1 = m \cdot r^2 \text{ und } NHID_2 = m \cdot r$$

Die Anzahl der Eingangsneuronen wird mit n bezeichnet, die Anzahl der Ausgangsneuronen mit m. Der Faktor $NHID_1$ gibt die Anzahl der Neuronen im ersten versteckten Layer an, $NHID_2$ jene im zweiten. Aus dem Gesamtdatenbestand werden im nächsten Schritt 30 Kanäle, gereiht nach geringster Anzahl an NaN, selektiert und einer weiteren Filterung zugeführt. Nach rigoroser Streichung ganzer Zeilen bei Vorliegen von *NaN*-Werten in einzelnen Kanälen bleiben sechs Kanäle des *Raspberry Pi*-Aufbaus mit jeweils etwa 6.000 Messzeitpunkten für die Modellierung übrig. Zusätzlich werden Stunden und Tage als Zahlenwerte aus den Messzeitpunkten extrahiert und ebenfalls als Eingangsgröße der Modellierung dem Datensatz hinzugefügt. Zur Abdeckung der jahreszeitlichen Schwankungen kann bei Vorliegen von Datenreihen über ein ganzes Jahr auch die Verwendung der Jahreszeit als Datenquelle angedacht werden. Für in Summe acht Eingangskanäle werden nach den Zusammenhängen in Gl. **4.1** somit vier Neuronen im ersten versteckten Layer und zwei Neuronen im zweiten angesetzt. Die tatsächliche Modellierung wird mit dem Befehl `neuralnet(...)` gestartet (Beschreibung aller Parameter siehe [FrGü16, S.7]); der Quellcode ist in Listing 5 dargestellt.

```
1           library(neuralnet)
2           n <- names(train_)
3           f <- as.formula(paste(modch, " ~", paste(n[!n %in% paste(modch)],
                                collapse = " + ")))
4           nn <- neuralnet(f, data=train_,hidden=c(5,2), threshold=0.1)
```

Listing 5: R-Code der neuronalen Modellierung

Der Parameter `threshold` definiert das Abbruchkriterium des Lernprozesses und bezeichnet den zu unterschreitenden Schwellenwert der partiellen Ableitungen der Fehlerfunktion δ (siehe Gl. **3.29**). Der Lernprozess in diesem Beispiel konvergiert, wenn ein `threshold` kleiner `0.1` erreicht wird. Für die Modellierung wird mit dem Algorithmus *RPROP* (engl. *resilient backpropagation*) nach [Ried94] gearbeitet. Mit dem Befehl `nn$result.matrix` können Informationen zum generierten neuronalen Netz `nn` abgerufen werden, unter anderem die Zahl der benötigten Iterationsschritte und die Gewichte der einzelnen Neuronen.

Das Ergebnis der Modellierung lässt sich auch graphisch darstellen. Abbildung 47 zeigt das modellierte neuronale Netz. Auf der linken Seite der Abbildung ist die Reihe der Eingangsgrößen aufgeführt, auf der rechten Seite die Zielgröße der Sauerstoffkonzentration. Die Gewichtungsfaktoren sind aus Gründen der Übersichtlichkeit ausgeblendet, die Biasfaktoren sind in blau dargestellt. Die Vorhersage des Zielparameters unter Verwendung des modellierten, künstlichen neuronalen Netzes wird mit der Funktion `compute(nn, data)` bewerkstelligt, wobei die Struktur `data` die Daten in gleicher Struktur wie bei der Modellierung enthält.

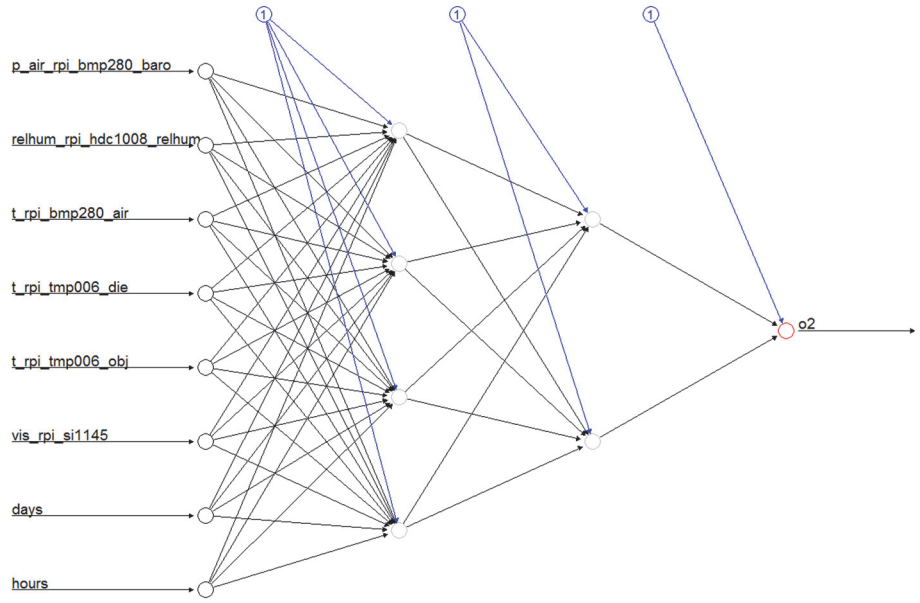

Abbildung 47: Neuronales Netzwerk (eigene Darstellung, Export aus R)

Für die Vergleichbarkeit der erstellten Modelle wird im letzten Schritt noch die mittlere quadratische Abweichung *MSE* (engl. *mean square error*) über alle Testdaten berechnet. Der Zusammenhang ergibt sich zu

$$MSE = \frac{1}{m}\sum_{m}(O_{2,pred} - O_{2,meas})^2 \qquad \textbf{4.2}$$

wobei $O_{2,pred}$ die vorhergesagten Werte, $O_{2,meas}$ die Messwerte und m die Anzahl der Zeilen der Eingangswerte bezeichnet.

Die Ergebnisse der Modellierung werden in Abschnitt 5 dargelegt. Der *MSE* des generalisierten linearen Modells wird dabei mit *MSElm* und jener des neuronalen Netzes mit *MSEnn* bezeichnet. Zur besseren qualitativen Einschätzung der Vorhersageergebnisse durch Vergleich der Testdaten mit den tatsächlichen Messwerten wird im letzten Schritt der Modellierung die Funktion `rollmedian` eingesetzt, um eine Signalglättung in geringem Ausmaß zu bewerkstelligen [Zeil17].

5. Ergebnisse und Diskussion

Die Ergebnisse der praktischen Umsetzung werden im finalen Abschnitt 5 präsentiert. Die Datenerfassung basiert auf dem Hardwareaufbau des *Environmental Monitoring* und der Einbindung in die bestehende Messnetzplattform. Anschließend wurden die Grundkonzepte des maschinellen Lernens erarbeitet und eine Implementierung durch Verwendung bekannter und numerisch optimierter Bibliotheken erstellt. Im letzten Schritt wurden die Implementierungen mit erfassten Daten eines längeren Messzeitraums beschickt und die Eignung der vorgestellten Konzepte und Implementierungen evaluiert. Im finalen Abschnitt wird die wichtigste Frage dieser Arbeit beantwortet: Sind die vorgestellten Konzepte für die Anwendung in der Datenerfassung tauglich und kann ein ausgewähltes Signal hinreichend gut modelliert werden? In Abschnitt 5.1 die Evaluierung der Methoden dargestellt. Abschnitt 5.2 widmet sich der Diskussion der Vorhersagen durch Vergleich mit dem konventionell erfassten Messparameter des gelösten Sauerstoffs im Gewässer und in 5.3 sind die Erkenntnisse zusammengefasst und ein Ausblick auf die mögliche Verfeinerung der Methodik wird gegeben.

5.1 Evaluierung und Schlussfolgerungen

Die erste Evaluierung des aufgebauten Prototyps wurde anhand einer Zeitreihendarstellung ausgeführt. In Abbildung 48 sind einige typische Tagesgänge, der Verlauf der Werte über einen Tag, beginnend mit Ende September 2016 dargestellt. Deutlich erkennbar ist der Zusammenhang von Sonnenbestrahlung im sichtbaren bzw. infraroten Bereich mit der Umgebungstemperatur bzw. der Wassertemperatur. Die Temperaturkurve schwenkt kurz nach dem Maximum, relativ rasch zu Beginn der fallenden Flanke der Beleuchtungsstärke, nach unten. In den Nachtstunden kühlt die Umgebung ab und am nächsten Morgen startet der Zyklus in entgegengesetzter Richtung erneut. Die gemessene Konzentration des gelösten Sauerstoffs zeigt das in Abschnitt 1.1 vorgestellte, erwartete Verhalten. Die kurzfristigen Einbrüche der Bestrahlung zwischen 22. und 23. September 2016, gut sichtbar an den Messdaten des *SI1145* im Infrarotbereich, lassen auf Bewölkung schließen, was durch stichprobenartige Auswertung der aufgezeichneten Bilder bestätigt wird (siehe Abbildung 49).

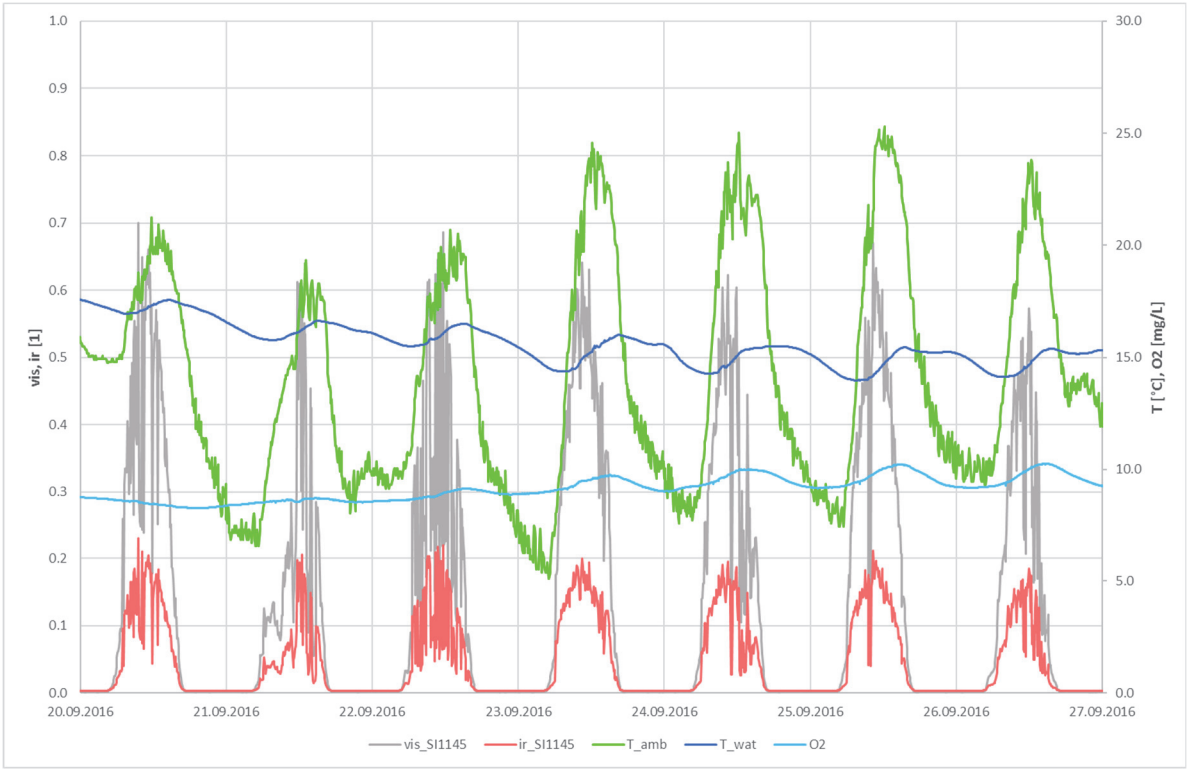

Abbildung 48: Messdaten – Tagesgang (eigene Darstellung)

04:49:00 UTC 12:13:00 UTC 15:55:00 UTC

Abbildung 49: Beispielbilder zur Bewölkung am 22. September 2016 (eigene Darstellung)

Die längerfristige Entwicklung der Verhältnisse ist in Abbildung 50 dargestellt. Ein interessanter Datenverlauf ist im Zeitbereich ab dem 19. November 2016 für die Dauer von rund zehn Tagen zu sehen. Die Sonnenbestrahlung in diesem Bereich bleibt im Wesentlichen gleich, der allgemeine Trend der Wassertemperatur weist in Richtung tieferer Temperaturen. Die ist bedingt durch den schrägeren Winkel der Sonneneinstrahlung im Winter. Trotzdem kommt es für einige Tage zu einem vergleichsweise sprunghaften Anstieg der Wassertemperatur und auch die Sauerstoffkonzentration fällt geringfügig ab. Dieser Zeitbereich ist ein interessanter Testfall für die Modellierung, da die Verhältnisse entgegen der üblichen Trends verlaufen.

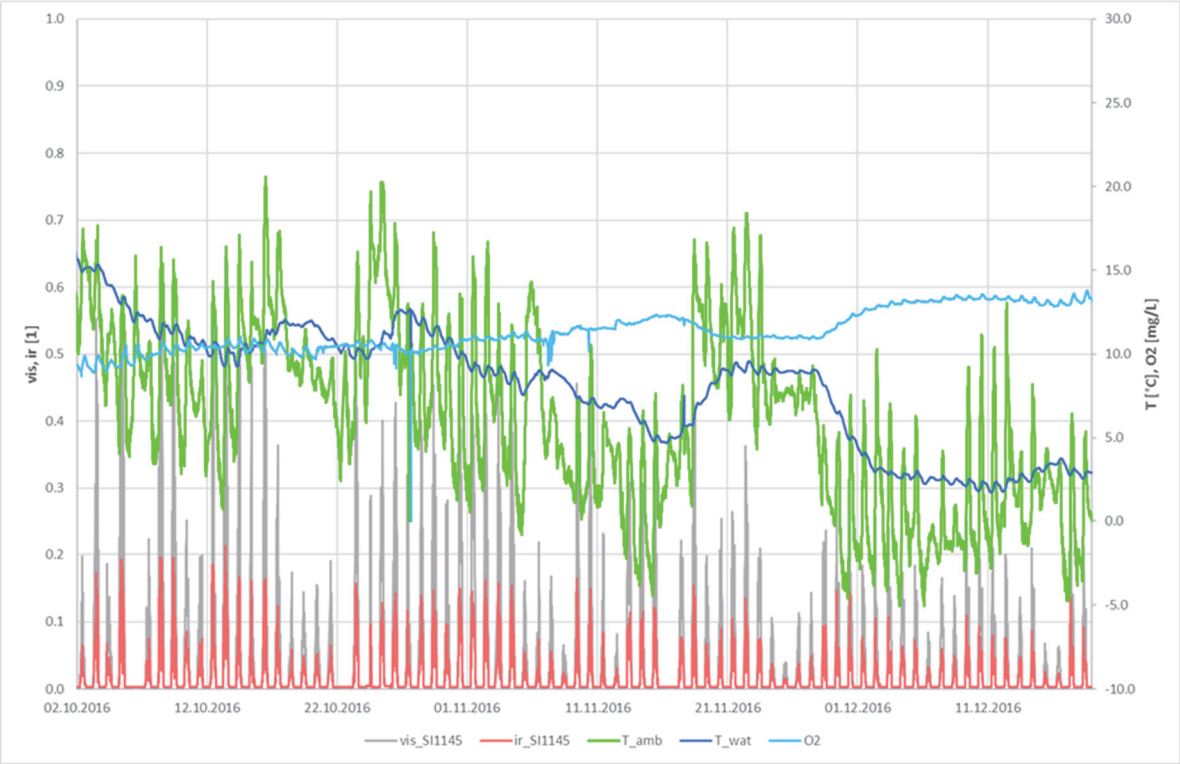

Abbildung 50: Datenentwicklung über zwei Wintermonate (eigene Darstellung)

Die Modellierung in *R* startet mit dem Laden der Rohdaten in Datenstrukturen der Modellierungsumgebung und einer Übersicht der Rohdaten, dargestellt in Abbildung 51. Die zum Teil starken Spitzen im Werteverlauf sind auf Tagesgänge des gelösten Sauerstoffs im Wasser zurückzuführen und haben direkt mit der Sonneneinstrahlung im Gewässer zu tun. Auffällig ist die ab Anfang März 2017 stark zunehmende Dynamik (Frühlingsbeginn).

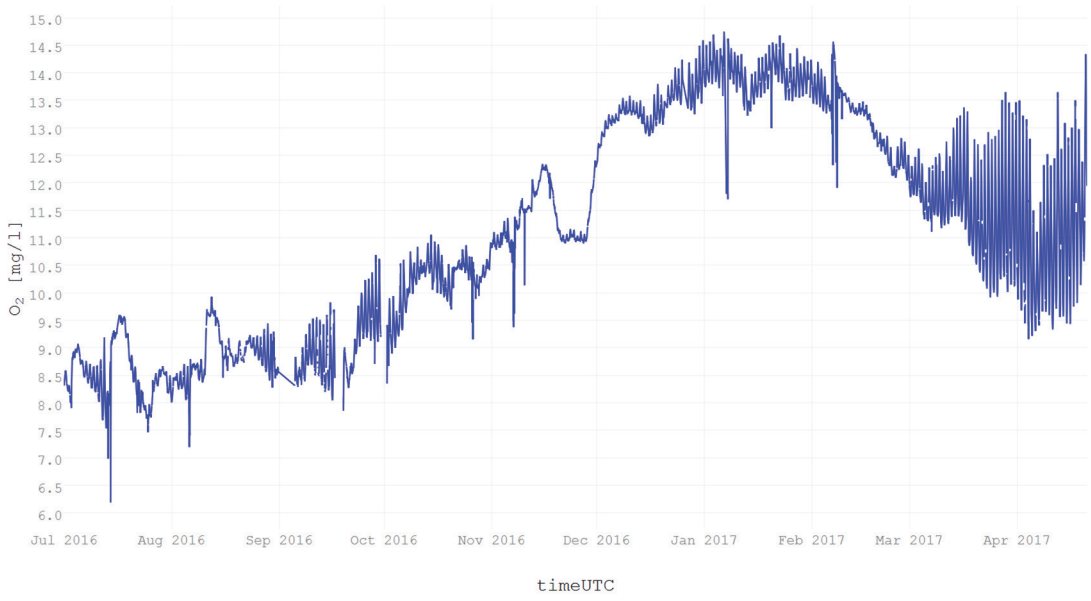

Abbildung 51: Werteverlauf der Sauerstoffmessung (eigene Darstellung)

Die Aufgabenstellung der Modellierung wird, um die Modelle mit Testdaten unterschiedlicher Dynamik zu beschicken, auf zwei separate Durchläufe aufgeteilt. Im ersten Schritt werden die Daten von 1. Juli 2016 bis 3. März 2017, ohne die hochdynamischen Daten, selektiert und mit *Datensatz A* bezeichnet. Danach wird der komplette Datensatz von 1. Juli 2016 bis 20. April 2017 als Basis verwendet und mit *Datensatz B* bezeichnet. Stark schwankende Daten weisen eine hohe Standardabweichung auf. Durch die Aufteilung in zwei separate Datensätze wird versucht, neben dem Einfluss des größeren Datenumfangs, einen eventuellen Zusammenhang der Dynamik der Daten mit der Modellierung sichtbar zu machen. Die beide Datensätze werden getrennt voneinander mit den im Abschnitt 4.3 beschriebenen Modellierungen behandelt. Für jeden Datensatz wird sowohl eine lineare Modellierung mit *glm* und eine neuronale Modellierung mit *neuralnet* erstellt. Der Grunddatensatz umfasst rund 58.400 Messzeitpunkte, die Anwendung der rigorosen *NaN*-Streichung sämtlicher Daten eines Zeitstempels beim Vorliegen eines einziges fehlerhaften beziehungsweise undefinierten Wertes führt auf rund 6.000 für die Modellierung zur Verfügung stehende Messzeitpunkte.

5.2 Diskussion der Vorhersagen

Eine Gegenüberstellung der gemessenen Werte und den Vorhersagen der Modellierungen für ***Datensatz A*** liefert Abbildung 52. Die Punkte „LM“ bezeichnen Ergebnisse der linearen Modellierung, die mit „NN“ bezeichnen jene der neuronalen Modellierung. Dabei gilt: Je näher die eingetragenen Punkte an der 45°-geneigten Gerade liegen, desto besser hat das Modell den (Test-)Datensatz vorhersagen können. Bei einer perfekten Vorhersage würden sämtliche Datenpunkte entlang der Geraden dargestellt werden. Zusätzlich sind die normierten Daten (*feature scaling*) wieder auf den Wertebereich der Eingangsdaten skaliert, um die Interpretation der Größenordnungen zu erleichtern. Die Auswertung

der MSE-Berechnung ergibt einen mittleren quadratischen Fehler von 0,89 für die lineare Modellierung und 0,50 für das neuronale Netz. Das bedeutet, die neuronale Modellierung ist für diese Aufgabenstellung die besser geeignete Methode.

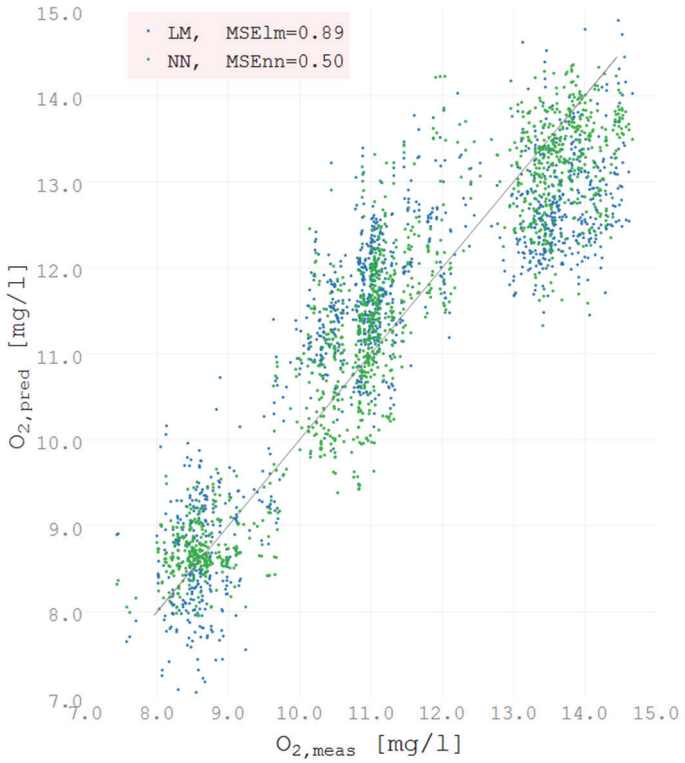

Abbildung 52: Datensatz A – Vorhersageplot (eigene Darstellung)

Abgesehen davon ist eine deutliche Konzentration der Punktwolken um im Wesentlichen drei Bereiche erkennbar: Eine Häufung zwischen 8-9 mg/l, eine um 11 mg/l und eine weitere im Bereich von 13.5 mg/l gelösten Sauerstoff im Gewässer. Die tatsächlichen Verhältnisse des gelösten Sauerstoffs sind stark gewässerspezifisch und unter anderem von der Beschattung und der Bodenzusammensetzung abhängig. Die dargestellten Werte im betrachteten Messzeitraum liegen, im Vergleich zu den langjährig erfassten Datenreihen, jedenfalls im üblichen Bereich.

Im nächsten Schritt werden die Modelle durch Vorhersage der Werte und Vergleich mit dem Trainings-Datensatz in Beziehung gestellt. Die Zeitreihe der Testdaten und der Vorhersagewerte des generalisierten linearen Modells zeigt Abbildung 53. „MEAS“ (engl. *measurement*) steht für Messdaten und „LM pred“ (engl. *linear modeling prediction*) für die Vorhersage des Messergebnisses auf Basis der linearen Modellierung. Bei der Interpretation ist zu beachten, dass die Indizes im Mittel nicht direkt aufeinander folgende Zeitstempel der Messung bezeichnen. Die Selektion der Trainingsdaten beruht auf zufälliger Auswahl von Zeitreihen aus dem Gesamtdatensatz; die Indizes der Testdaten sind gereiht nach aufsteigender Zeitachse, aber nicht durch die letzten 30% der Messwerte der Eingangsdaten im Zeitverlauf bestimmt, sondern durch „Daten ungleich Trainingsdaten“ zu beliebigen Zeitpunkten definiert. Deutlich erkennbar ist eine gewisse Abweichung zwischen Modellvorhersage und Testdaten, vor allem im Bereich der hohen Indizes der Messwerte. Die Modellierung mittels eines

generalisierten Modells kann also nur einen Ansatzpunkt liefern, den erwarteten Bereich des Signals vorherzusagen.

Abbildung 53: Datensatz A –Testdaten und glm-Modellierung (eigene Darstellung)

Die neuronale Modellierung, in Abbildung 54 dargestellt, liefert, bis auf einige Ausreißer, ein insgesamt zufriedenstellendes Bild. „MEAS“ bezeichnet wieder Messdaten, „NN pred“ (engl. *neural network modeling prediction*) die mittels neuronaler Modellierung vorhergesagter Daten. Besonders gelungen scheint die Modellierung bei den extremen Testdaten im Index-Bereich 800 bis 950, die Werte der Modellierung und der Messung passen im Mittel gut zusammen.

Abbildung 54: Datensatz A –Testdaten und neuralnet-Modellierung (eigene Darstellung)

Die Ergebnisse der Modellierung von ***Datensatz B*** zeigen ein ähnliches Bild. Die qualitative Verteilung der Punktwolken kann wieder in drei Bereiche mit ähnlich dichter Häufung an Messpunkten zugeordnet werden (siehe Abbildung 55). Der mittlere quadratische Fehler der neuronalen Modellierung ist abermals niedriger als jener der linearen Modellierung.

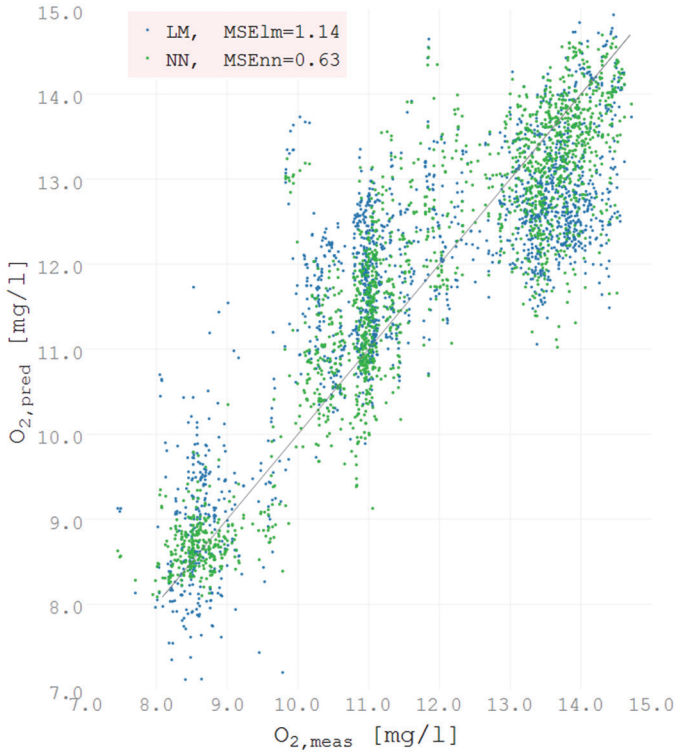

Abbildung 55: Datensatz B – Vorhersageplot (eigene Darstellung)

Die Gegenüberstellung der Testdaten von *Datensatz B* mit den Daten des generalisierten linearen Modells zeigt Abbildung 56, der etwas höhere *MSE* erschließt sich rein aus der Betrachtung der Trainingsdaten im Graphen nicht.

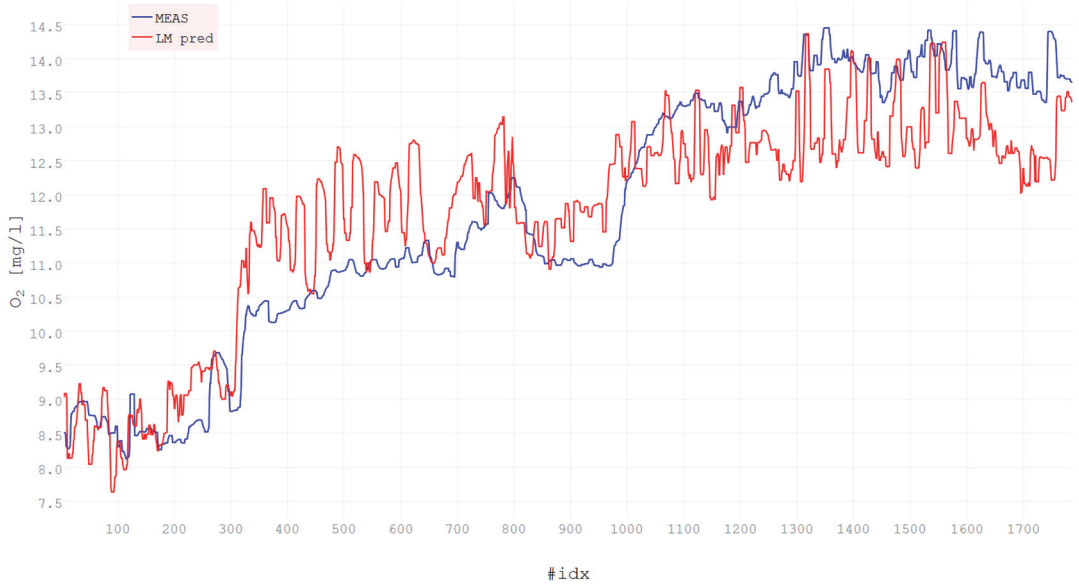

Abbildung 56: Datensatz B –Testdaten und glm-Modellierung (eigene Darstellung)

Der Vergleich zu den Daten der neuronalen Modellierung ist in Abbildung 57 dargestellt. Ähnlich wie beim *Datensatz A* wird auch hier der besonders sprunghafte Verlauf im Bereich des Index 800 bis 950 gut vorhergesagt.

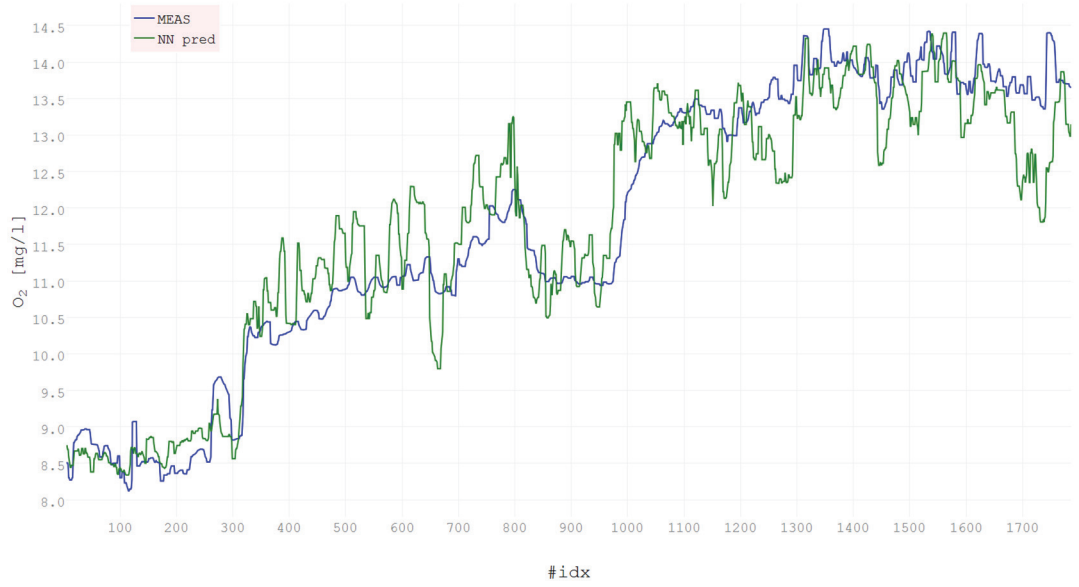

Abbildung 57: Datensatz B –Testdaten und neuralnet-Modellierung (eigene Darstellung)

Im direkten Vergleich der neuronalen Modellierungen der beiden Datensätze lässt sich, allein in Bezug auf die vorliegenden Trainingsdaten, folgendes Ergebnis formulieren: Beide Modellierungen liefern,

bei einer Aufteilung der Gesamtdaten im Verhältnis von 70:30 für Trainings- und Testdaten, zufriedenstellende Ergebnisse und lassen, ausschließlich die Testdaten betrachtend, auf eine gute Eignung sowohl als Surrogat-Parameter als auch zur Datenplausibilisierung schließen.

In einem letzten Schritt werden die tatsächlichen Messwerte mit jenen einer Datenvorhersage über den jeweiligen Messzeitraum der beiden Datensätze in Beziehung gestellt, jedoch auf den jeweils kompletten Datensatz angewendet. Einschränkend muss erwähnt werden, dass hier die Vorhersage auch auf Daten basiert, welche für die Modellierung verwendet worden sind. Die Generalisierung findet daher zum Teil auf Basis der Trainingsdaten der Modelle statt. Im Hinblick darauf, dass nur etwa jeder zehnte Datensatz für die Modellierung auch tatsächlich verwendet wird (*NaN*-Streichung), wird diese Einschränkung für eine erste Einschätzung hingenommen. Außerdem werden in den Eingangsdaten vorliegende *NaN*-Werte durch den Median der Spalte der Messdaten ersetzt; damit erklärt sich der vergleichsweise große Eingangsdatenumfang der Vorhersage. Die Ergebnisse vorwegnehmend sei erwähnt: Hier zeigt sich ein wesentlich differenzierteres Bild der Tauglichkeit.

Nachdem die neuronale Modellierung nach der Fehlerberechnung mittels *MSE* als geeignetere Methode gewählt wurde, sind in den folgenden Abbildungen nur mehr die neuronalen Modelle mit den Messdaten gegenübergestellt; die lineare Modellierung zeigt ähnliches Verhalten, jedoch mit noch größeren Abweichungen behaftet.

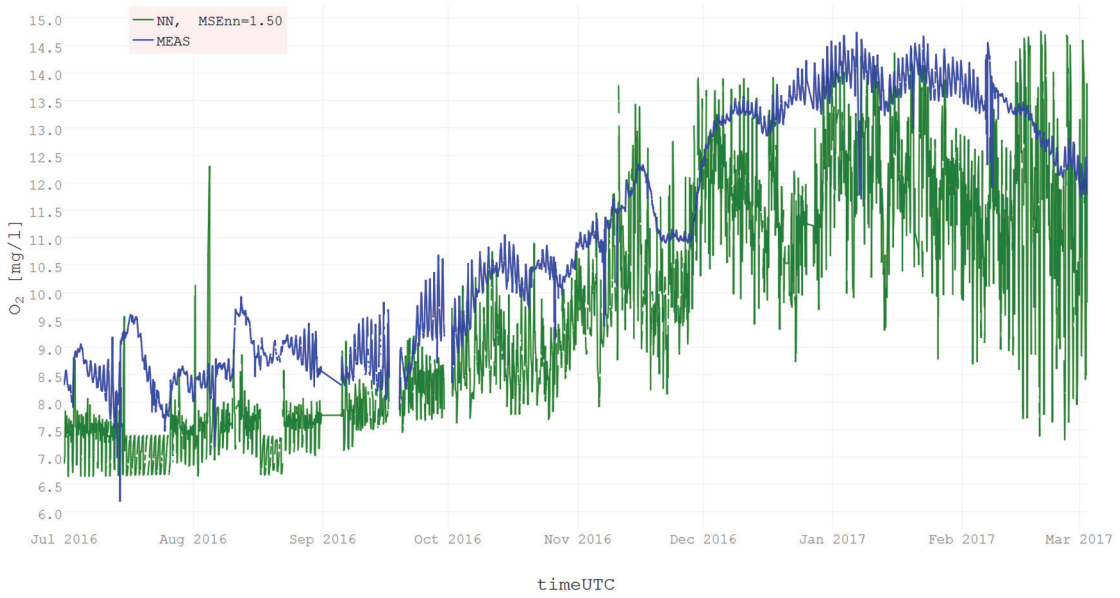

Abbildung 58: Datensatz A – Langzeituntersuchung der neuronalen Modellierung (eigene Darstellung)

Abbildung 58 zeigt den Vergleich ohne die stark schwankenden Daten im März 2017 (Datensatz A), die Vorhersagewerte wurden bereits einer gleitenden Medianbildung unterzogen. Die Abweichungen zwischen Messung und Modellierung im Datensatz A weisen ein *MSE* von 1,50 auf. Die vorhergesagten Werte des neuronalen Netzes lassen sind als durchwegs zu niedrig im Vergleich zu den tatsächlich gemessenen Werten des gelösten Sauerstoffs im Gewässer beschreiben.

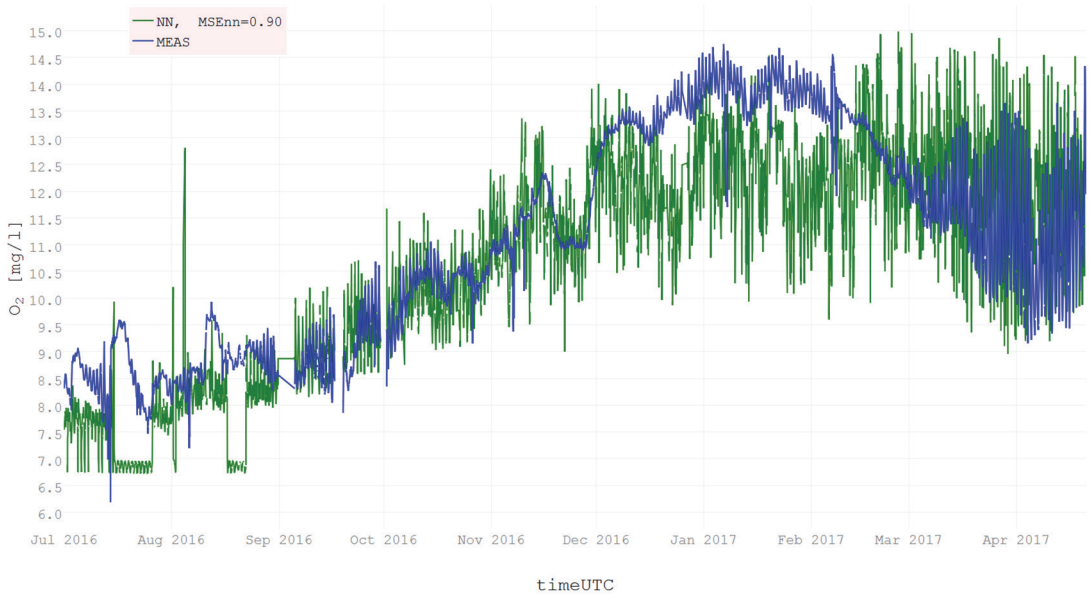

Abbildung 59: Datensatz B – Langzeituntersuchung der neuronalen Modellierung (eigene Darstellung)

Die vergleichende Darstellung in Abbildung 59 basiert auf *Datensatz B*, also inklusive der dynamischen Daten vom März 2017. Der *MSE* über alle Daten gerechnet liegt hier bei 0,90. Die Ergebnisse sind näher an den tatsächlich gemessenen Werten, der qualitative Werteverlauf im Bereich Dezember 2016 bis Anfang März 2017 weist jedoch nicht zufriedenstellende Abweichungen, besonders im Hinblick auf die bereits durchgeführte Signalglättung durch Medianbildung, auf. Der Vergleich der Modellierung ohne beziehungsweise mit den Daten vom März zeigt ebenfalls eine vergleichsweise große Sensitivität bezüglich der Dynamik der Trainingsdaten, wobei die Modellierung von *Datensatz B* (mit Daten vom März 2017) einen besseren MSE aufweist als jene vom *Datensatz A* (ohne Daten vom März 2017). Die Modellierung hat sich durch den erweiterten Umfang der Messdaten verbessert. Mehr und dynamischere Trainingsdaten bedeuten in diesem Fall geringere Abweichungen des Modells im Vergleich zu den Messwerten. Große Abweichungen treten vor allem im Winter auf. Das Modell unterschätzt die tatsächliche Konzentration des gelösten Sauerstoffs im Gewässer im Durchschnitt um 20%. Die Eignung der Modellierung als Surrogat-Parameter, also der komplette Ersatz der Sauerstoffsonde vor Ort ist beim vorliegenden Datensatz nicht gegeben. Die Eignung als Erweiterung der Datenplausibilisierung wird im letzten Schritt der Auswertung untersucht. Die Vorhersage des neuronalen Modells wird einer Signalglättung mit einem 50 Werte großen Fenster, entsprechend 5 Stunden Messbetrieb, unterzogen und um ein *erlaubtes Band* im Bereich Wert +/- 10% erweitert. Die Schranken sind in Abbildung 60 durch den ausgefüllten, grün eingefärbten Bereich dargestellt. Alle Messwerte außerhalb dieses Bereichs werden als nicht plausibel gewertet.

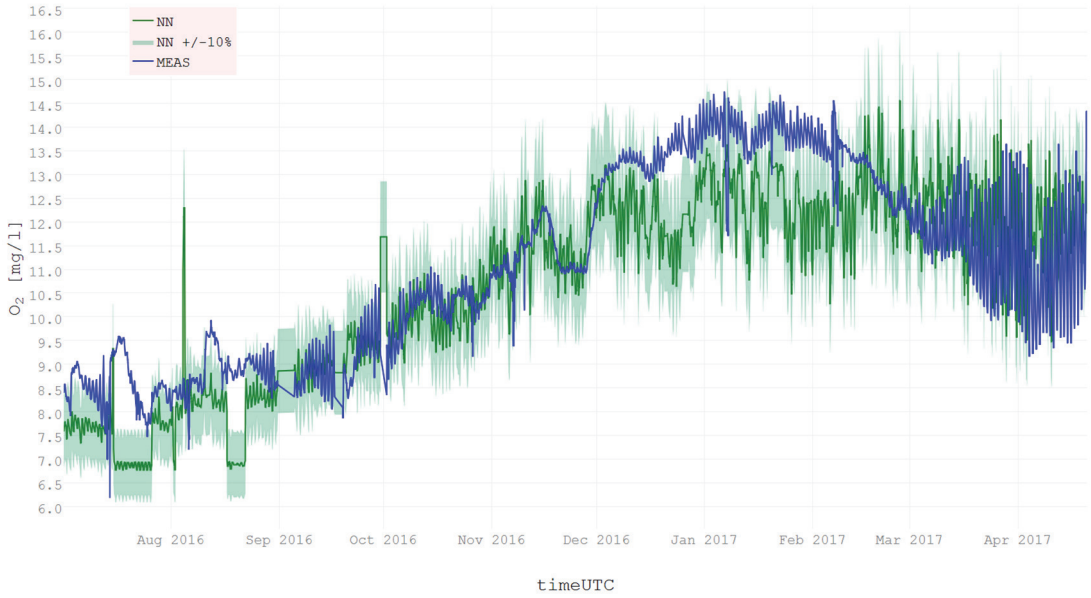

Abbildung 60: Datensatz B – Datenplausibilisierung (eigene Darstellung)

In Abbildung 61 ist der interessante Datenbereich vom November im Detail dargestellt. Der erste Vergleich der Vorhersagedaten mit den Messdaten zeigt, dass der Messwert (in blau dargestellt) durchwegs innerhalb der von den modellierten Werten bestimmten Schranken liegt.

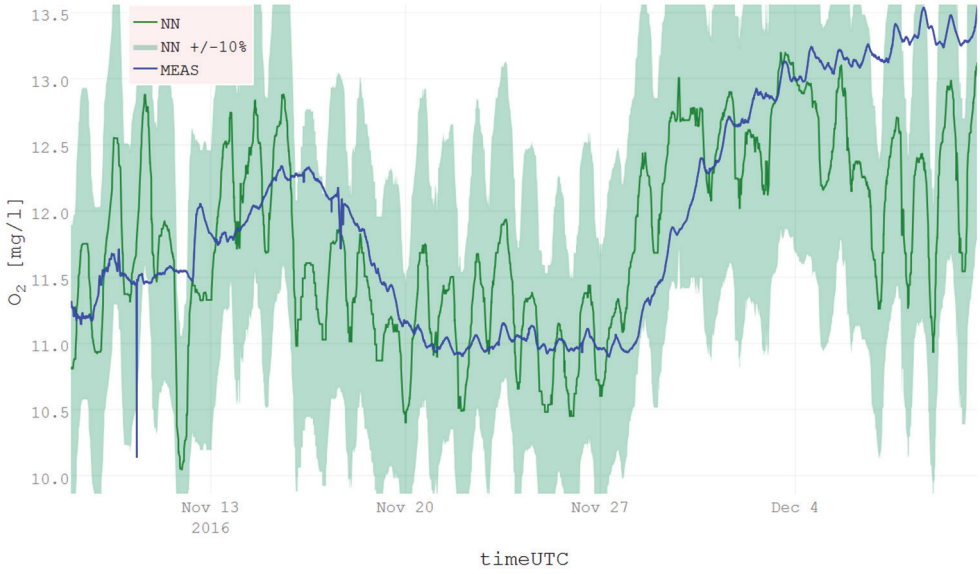

Abbildung 61: Datensatz B – Datenplausibilisierung Detail (eigene Darstellung)

Für die Interpretation dieses Ergebnisses muss berücksichtigt werden, dass plausibilisierte Messdaten, nach den in Abschnitt 2.3 beschriebenen Kriterien, Basis der Modellierung waren. Die gemessenen Daten, in allen bisherigen Abbildungen in blau dargestellt, wären demnach als *plausibel* einzustufen und die Werte eines *idealen* Modells sollten dazu deckungsgleich zur Darstellung kommen. Das Modell funktioniert, abgesehen vom Messzeitraum im Winter, für die praktische Anwendung hinreichend gut genug, um allein aus meteorologischen Umgebungsparametern Zeitreihen vorhersagen zu können, die in einem +/- 10% breiten Bereich um den tatsächlich gemessenen, plausiblen Wert liegen. Die Kombination mehrerer Kriterien kann die Datenplausibilisierung weiter verfeinern. Es wird daher geschlossen, dass die neuronale Modellierung zur erweiterten Datenplausibilisierung eine geeignete Methode ist und zur Anwendung als zusätzliches Kriterium zur Bewertung der Plausibilität der Sauerstoffmessung kommen kann.

Die weitere Verfeinerung der Modellierung kann zu einer Verbesserung der Vorhersagen führen; die dazu angedachten Maßnahmen sind im folgenden, finalen Abschnitt dieser Arbeit, neben der Zusammenfassung der Ergebnisse, dargestellt.

5.3 Zusammenfassung und Ausblick

Aufgrund der gesammelten Erfahrungen mit Umsetzung, Betrieb und Datenmodellierung des *Environmental Monitorings* sind zusammenfassend folgende wesentliche Aussagen zu treffen:

- Für das Erreichen einer brauchbaren Modellierung sind gute, hinreichend präzise und vor allem möglichst kontinuierliche Messdatenreihen fundamental wichtig. Die Qualität der Modellierung ist maßgeblich von der Behandlung der nicht vorhandenen beziehungsweise fehlerhaften Messdaten abhängig. Der Datenbestand kann, wenn auf die Ersetzung fehlender Daten durch den Median des jeweiligen Parameters verzichtet wird, stark an Umfang verlieren. In diesem Fall stehen für die Modellierung wesentlich weniger Trainingsdaten zur Verfügung, wobei aber von einer insgesamt gesteigerten Datenqualität ausgegangen werden kann. Weitere Ansätze, wie zum Beispiel die Ersetzung problematischer Messwerte zum Beispiel durch den Mittelwert zeitlich naher Datensätze, sind in Vorbereitung.
- Die neuronale Modellierung weist Anzeichen einer Überanpassung (*overfitting*) auf. Die Generalisierung auf neue Daten führt auf größere Abweichungen im Vergleich zu den umfangreicheren Messdaten als die Anwendung des Modells auf die Trainingsdaten. Die Modellierung ist außerdem empfindlich auf die Dynamik der Messreihen im Frühling. Eine Möglichkeit, die Überanpassung einer Modellierung zu verhindern, ist *Regularisierung*. Für größere neuronale Netzwerke beschreibt [SHKS14] das Verfahren *Dropout* als vielversprechenden Ansatz, bei dem einzelne Neuronen im Rahmen des Trainings entfernt werden, das neuronale Netzwerk ausgedünnt und eine Überanpassung verhindert wird. Nach [SHKS14] neigen neuronale Netze mit kleinen Trainingsdatensätzen besonders zu Überanpassung. Der Ansatz der *Regularisierung* ist mit den gewählten Bibliotheken nicht kompatibel und bedingt daher weitere und komplexe Adaptierungen.

- Ein Ersatz vorhandener Sauerstoffsonden durch ein neuronales Modell als *Surrogat-Parameter* ist mit der vorliegenden Datenbasis und gewählten Realisierung aufgrund der Abweichungen im Langzeit-Datenverlauf nicht sinnvoll.
- Der modellierte Verlauf des im Wasser gelösten Sauerstoffs ist, wenn +/- 10%-Schranken auf Basis der modellierten Werte als Gültigkeitsgrenzen von plausiblen Daten eingeführt werden, für die Datenplausibilisierung in *i*^{TUW}*mon* verwendbar. Das neuronale Netz muss dafür auf dem *Raspberry Pi* direkt umgesetzt und die Wertgrenzen für die Datenplausibilisierung in die zentrale Messdatenbank übernommen werden. Zusätzlich sind Tests der Modellierung über längere Messzeiträume notwendig. Der bisher noch nicht verfolgte Ansatz zur Validierung des Modells mit einem Teil der vorhandenen Daten ist ebenfalls in Vorbereitung.
- Der Vergleich der modellierten mit den gemessenen Werten wurde im Rahmen dieser Arbeit über den gesamten vorliegenden Datenbestand mit einem Durchgang der Modellierung durchgeführt. Ein Ansatz zur Verbesserung und bei Vorliegen der Implementierung am *Raspberry Pi* könnte die kurzfristige Modellierung, beispielsweise im Wochenabstand, unter kontinuierlichem, erneutem Training des neuronalen Netzes, sein. Bei Vorliegen von Datenreihen über ein komplettes Jahr kann auch die Jahreszeit als zusätzlicher Parameter in die neuronale Modellierung einbezogen werden.
- Zur Modellierung zusätzlicher Zielparameter aus dem Bereich der Nährstoffe, wie gelöster Phosphor oder Ammonium-Stickstoff wird die Datenerfassung um weitere, kostengünstige Sensorik zur Bestimmung der Wassergüte ergänzt. Die Datenbasis wird umfangreicher und die Modellierung kann auf Basis der neuen Messdaten verfeinert werden. Die automatische Erkennung einer Signaldrift wäre ebenfalls wünschenswert.

Die Erweiterung des bestehenden Prototyps um die Erfassung ausgewählter Wassergüteparameter zum sogenannten EWM, *Environmental Water Monitoring*, auf Basis eines angepassten Leiterplattenlayouts soll im Anschluss an diese Arbeit in Betrieb genommen werden.

Literaturverzeichnis

[Adaf17] Adafruit, LLC:URL https://www.adafruit.com/. [abgerufen am 2017-05-31]

[Ag16] ams AG: *TCS3472 Datasheet*. URL http://ams.com/eng/content/download/319364/1117183/287875. [abgerufen am 2017-05-31]

[Alic15] Alice, Michy: *Fitting a Neural Network in R; neuralnet package*. URL https://datascienceplus.com/fitting-neural-network-in-r/. [abgerufen am 2017-05-31]. — datascience+

[Arm13] ARM: *Cortex™-A7 MPCore™ Technical Reference Manual*. URL http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ddi0464f/index.html. [abgerufen am 2017-05-31]

[AtIM10] Atzori, Luigi ; Iera, Antonio ; Morabito, Giacomo: The Internet of Things: A survey. In: *Computer Networks* Bd. 54 (2010), Nr. 15, S. 2787–2805

[Baue14] Bauernhansl, Thomas: Die Vierte Industrielle Revolution – Der Weg in ein wertschaffendes Produktionsparadigma. In: *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Wiesbaden : Springer Fachmedien Wiesbaden, 2014 — ISBN 978-3-658-04681-1, S. 5–35

[Bdew15] BDEW Bundesverband der Energie- und Wasserwirtschaft e.V.: *Whitepaper: Anforderungen an sichere Steuerungs und Telekommunikationssysteme*, 2015

[BeTV09] Berling, Bernhard ; Thrun, Werner ; Vogt, Wolfgang ; Heinrich, B. (Hrsg.): *Kaspers/Küfner Messen — Steuern — Regeln*. 8. Aufl. Wiesbaden : Vieweg+Teubner Verlag, 2009. — 344 S. — ISBN 978-3-8348-0006-0

[Bick93] Bick, Harmut: *Ökologie: Grundlagen, terrestrische und aquatische Ökosysteme, angewandte Aspekte* : Gustav Fischer Verlag, 1993 — ISBN 3-437-20432-7

[Bmvi14] bmvit, Bundesministerium für Verkehr, Innovation und Technologie: *Die Fabrik der Zukunft*. URL https://www.bmvit.gv.at/innovation/produktion/fabrik_der_zukunft.html. [abgerufen am 2017-05-31]

[Brou16] Broussard, Mitchel: *New Photos App Detects 4,432 Total Searchable Objects and 7 Facial Expressions [Updated]*. URL https://www.macrumors.com/2016/06/20/photos-app-detects-432-objects/. [abgerufen am 2017-05-31]

[Brow17] Brown, B.: The Social Life of Autonomous Cars. In: *Computer* Bd. 50 (2017), Nr. 2, S. 92–96

[Bund14] Leitfaden Cyber-Sicherheits-Check. In: Bundesamt für Sicherheit in der Informationstechnik (Hrsg.) (2014)

[Cauc47] Cauchy, Augustin: Méthode générale pour la résolution des systemes d'équations simultanées. In: *Comp. Rend. Sci. Paris* Bd. 25 (1847), Nr. 1847, S. 536–538

[CEFG13] Camhy, David ; Ertl, Thomas ; Fuiko, Roland ; Gamerith, Valentin ; Gruber, Günter ; Hofer, Thomas ; Höller, Martin ; Kinzel, Carolina ; u. a.: *Integrierte Betrachtung eines Gewässerabschnitts auf Basis kontinuierlicher und validierter Langzeitmessreihen* (Endbericht). Wien : Bundesministerium für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft, Sektion VII - Wasser, 2013

[CoBK14] Colombo, A. W. ; Bangemann, T. ; Karnouskos, S.: IMC-AESOP outcomes: Paving the way to collaborative manufacturing systems. In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, S. 255–260

[Codd70] Codd, Edgar F.: A relational model of data for large shared data banks. In: *Communications of the ACM* Bd. 13 (1970), Nr. 6, S. 377–387

[Coll09] Collins, Galen: *Pymodbus 1.0 documentation*. URL http://pymodbus.readthedocs.io/en/latest/index.html. [abgerufen am 2017-05-31]

[CWBD98] Clay, R. W. ; Wild, N. R. ; Bird, D. J. ; Dawson, B. R. ; Johnston, M. ; Patrick, R. ; Sewell, A.: A Cloud Monitoring System for Remote Sites. In: *Publications of the Astronomical Society of Australia* Bd. 15 (1998), Nr. 03, S. 332–335

[Daeb06] Daebel, Helge: *Parameter uncertainties in modeling urban wastewater systems*, University of Karlsruhe, 2006

[Degr06] Degreen: *Sonne Strahlungsintensität*, 2006

[Din95] DIN 1319-1:1995-01: *Grundlagen der Meßtechnik - Teil 1: Grundbegriffe*, 1995

[Dine07] DIN EN ISO 15839:2007-02: *Wasserbeschaffenheit - Online-Sensoren/Analysengeräte für Wasser - Spezifikationen und Leistungsprüfungen (ISO 15839:2003); Deutsche Fassung EN ISO 15839:2006*, 2007

[Dine08] DIN EN 60027-6:2008-04: *Formelzeichen für die Elektrotechnik - Teil 6: Steuerungs- und Regelungstechnik (IEC 60027-6:2006); Deutsche Fassung EN 60027-6:2007*, 2008

[Dine14] DIN EN 62264-1:2014-07: *Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie (IEC 62264-1:2013); Deutsche Fassung EN 62264-1:2013*, 2014

[Dini14] DIN IEC 60050-351:2014-09: *Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik (IEC 60050-351:2013)*, 2014

[Dini15] DIN IEC 62443-3-3:2015-06: *Industrielle Kommunikationsnetze - IT-Sicherheit für Netze und Systeme*, 2015

[Dini80] DIN IEC 60381-2:1980-06: *Analoge Signale für Regel- und Steueranlagen; Analoge Gleichspannungssignale*, 1980

[Dini85] DIN IEC 60381-1:1985-11: *Analoge Signale für Regel- und Steueranlagen; Analoge Gleichstromsignale; Identisch mit IEC 60381-1, Ausgabe 1982*, 1985

[DoHK01] Dokulil, Martin ; Hamm, Alfred ; Kohl, Johannes: *Ökologie und Schutz von Seen*. 1. Aufl. Wien : UTB, Stuttgart, 2001 — ISBN 978-3-8252-2110-2

[DRCC12] Delsing, J. ; Rosenqvist, F. ; Carlsson, O. ; Colombo, A. W. ; Bangemann, T.: Migration of industrial process control systems into service oriented architecture. In: *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, 2012, S. 5786–5792

[Dude16] Duden: industrielle Revolution. *Duden Wirtschaft von A bis Z: Grundlagenwissen für Schule und Studium, Beruf und Alltag*.

[DWLL13] Demchenko, Yuri ; Worring, Marcel ; Los, Wouter ; de Laat, Cees: *Towards Defining Big Data Architecture Framework*. URL http://www.delaat.net/posters/pdf/2013-09-16-rda-bdaf.pdf. [abgerufen am 2017-05-31]

[Endr17] *Endress+Hauser Management AG*. URL https://www.de.endress.com/de. [abgerufen am 2017-05-31]

[Etg17] ETG1992: Bundesgesetz über Sicherheitsmaßnahmen, Normalisierung und Typisierung auf dem Gebiete der Elektrotechnik (Elektrotechnikgesetz 1992 – ETG 1992), 2017

[Euro14] Europäische Union (EU): RICHTLINIE 2000/60/EG DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 23. Oktober 2000 zur Schaffung eines Ordnungsrahmens für Maßnahmen der Gemeinschaft im Bereich der Wasserpolitik, 2014

[Euro16] Europäische Union (EU): RICHTLINIE (EU) 2016/1148 DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 6. Juli 2016 über Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen in der Union, 2016

[EVZH07] Elliott, Chance ; Vijayakumar, Vipin ; Zink, Wesley ; Hansen, Richard: National Instruments LabVIEW: A Programming Environment for Laboratory Automation and Measurement. In: *JALA: Journal of the Association for Laboratory Automation* Bd. 12 (2007), Nr. 1, S. 17–24

[FMJB15] Feljan, A. V. ; Mohalik, S. K. ; Jayaraman, M. B. ; Badrinath, R.: SOA-PE: A service-oriented architecture for Planning and Execution in cyber-physical systems. In: *2015 International Conference on Smart Sensors and Systems (IC-SSS)*, 2015, S. 1–6

[Fort17] Fortinet: *Fortigate 60c*. URL https://www.fortinet.com/products/firewalls/firewall/fortigate-entry-level.html. [abgerufen am 2017-05-31]

[FrGü16] Fritsch, Stefan ; Günther, Frauke: *Package neuralnet*. URL ftp://64.50.236.52/.1/cran/web/packages/neuralnet/neuralnet.pdf. [abgerufen am 2017-05-31]

[FuWi15] Fuiko, Roland ; Winkelbauer, Andreas: Automatisierte Plausibilitätsprüfung von online Messdaten. In: *10. Fachtagung Mess- und Regelungstechnik in abwassertechnischen Anlagen*. Kassel : DWA, 2015

[FuWW13] Fuiko, Roland ; Winkelbauer, Andreas ; Winkler, Stefan: Data integration - A substantial prerequisite for correct data interpretation in online monitoring. In: *ICA 2013, 11th IWA Conference on Instrumentation Control and Automation*, *Proceedings*. Narbonne, 2013

[GeGP16] Gebhart, Gennie ; Grant, Starchy ; Portnoy, Erica: *Facial Recognition, Differential Privacy, and Trade-Offs in Apple's Latest OS Releases*. URL https://www.eff.org/deeplinks/2016/09/facial-recognition-differential-privacy-and-trade-offs-apples-latest-os-releases. [abgerufen am 2017-05-31]. — Electronic Frontier Foundation

[Geye03] Geyer, Charles J.: Generalized linear models in R. In: *R Reference Document* (2003), S. 1–23

[Goog17] Google: *Datenschutzerklärung & Nutzungsbedingungen*. URL https://www.google.com/policies/privacy/. [abgerufen am 2017-05-31]

[GüFr10] Günther, Frauke ; Fritsch, Stefan: neuralnet: Training of neural networks. In: , *The R Journal*. Bd. 1, 2010, S. 30–38

[Hach17] *Hach Lange GmbH*. URL https://at.hach.com/. [abgerufen am 2017-05-31]

[HaRe83] Haerder, Theo ; Reuter, Andreas: Principles of transaction-oriented database recovery. In: *ACM Computing Surveys (CSUR)* Bd. 15 (1983), Nr. 4, S. 287–317

[Hayk98] Haykin, Simon: *Neural Networks: A Comprehensive Foundation*. 2. Aufl. Upper Saddle River, N.J : Prentice Hall, 1998 — ISBN 978-0-13-273350-2

[Heat08] Heaton, Jeff: *Introduction to Neural Networks with Java, 2nd Edition*. 2. Aufl. St. Louis, Mo : Heaton Research, Incorporated, 2008 — ISBN 978-1-60439-008-7

[HeHe98] Hergt, Manfred ; Heinrich, Dieter: *dtv - Atlas Ökologie.* 5. Aufl. München : Deutscher Taschenbuch Verlag, 1998 — ISBN 978-3-423-03228-5

[Hipp17] Hipp, Wyrick & Company, Inc.: *SQLite*. URL https://sqlite.org/about.html. [abgerufen am 2017-05-31]

[HoHo91] Honold, Frank ; Honold, Brigitte: *Ionenselektive Elektroden: Grundlagen und Anwendungen in Biologie und Medizin*. Basel : Birkhäuser, 1991 — ISBN 978-3-7643-2560-2

[Ieee03] IEEE Computer Society: *IEEE Std 802.3af-2003: IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirements*, 2003

[Ieee08] IEEE Computer Society: *IEEE Std 754-2008: IEEE Standard for Floating-Point Arithmetic*, 2008

[Ietf06] IETF RFC4251: *The Secure Shell (SSH) Protocol Architecture*, 2006

[Ietf10] IETF RFC5905: *Network Time Protocol Version 4: Protocol and Algorithms Specification*, 2010

[Iso90] ISO 8466-1:1990-03: *Wasserbeschaffenheit - Kalibrierung und Auswertung analytischer Verfahren und Bewertung von Verfahrenskenngrößen - Teil 1: Statistische Auswertung der linearen Kalibrierfunktion*, 1990

[Isoi04] ISO/IEC 9075-1: 2004 (E): *ISO/IEC 9075-1: 2004 (E) Information technology—Database languages—SQL—Part 1: Framework (SQL/Framework)*, 2004

[Isoi99] ISO/IEC 9075-1: 2004 (E): *ISO 17166:1999-12; CIE S 007:1999-12: Erythema reference action spectrum and standard erythema dose*, 1999

[Iwag17] IWAG, TU Wien: *AutoDataReport Raab*. URL http://iwr.tuwien.ac.at/wasser/raab.html. [abgerufen am 2017-05-31]

[Jone16] Jones, Dave: *picamera — Picamera 1.13 Documentation*. URL https://picamera.readthedocs.io/en/release-1.13/. [abgerufen am 2017-05-31]

[JRGB08] Joannis, C. ; Ruban, G. ; Gromaire, M.-C. ; Bertrand-Krajewski, J.-L. ; Chebbo, G.: Reproducibility and uncertainty of wastewater turbidity measurements. In: *Water Science & Technology* Bd. 57 (2008), Nr. 10, S. 1667–1673

[KaLW11] Kagermann, Henning ; Lukas, Wolf-Dieter ; Wahlster, Wolfgang: Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution. In: *VDI nachrichten* (2011), Nr. 13

[Kamp16] Kamps, Haje Jan: *Apple introduces facial and object recognition for mobile photographers*. URL http://social.techcrunch.com/2016/06/13/apple-image-and-facial-recognition/. [abgerufen am 2017-05-31]. — TechCrunch

[Kavs02] Kavsek, Barbara: *Partial Least Squares (PLS) Regression and its Robustification*, Technische Universität Wien, 2002

[KCJD10] Karnouskos, S. ; Colombo, A. W. ; Jammes, F. ; Delsing, J. ; Bangemann, T.: Towards an architecture for service-oriented process monitoring and control. In: *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*, 2010, S. 1385–1391

[KHKS15] Kreuzinger, Norbert ; Haslinger, Julia ; Kornfeind, Lukas ; Schaar, Heidmarie ; Saracevic, Ernis ; Winkelbauer, Andreas ; Hell, Florian ; Walder, Christian ; u. a.: *KomOzAk Endbericht: Weitergehende Reinigung kommunaler Abwässer mit Ozon sowie Aktivkohle für die Entfernung organischer Spurenstoffe* (Endbericht) : Ministerium für ein lebenswertes Österreich, bmlfuw.gv.at, 2015

[Kolk50] Kolkwitz, Richard: *Oekologie der Saprobien: Über die Beziehung der Wasserorganismen zur Umwelt*, *Schriftenreihe für Wasser-, Boden und Lufthygiene* : Piscator, 1950

[Koms06] Komsta, Lukasz: Processing data for outliers. In: , *R News*. Bd. 2, 2006, S. 10–13

[Kope11] Kopetz, Hermann: *Real-Time Systems: Design Principles for Distributed Embedded Applications*. 2. Aufl. New York Dordrecht Heidelberg : Springer, 2011 — ISBN 978-1-4419-8236-0

[Kräm07] Krämer, Nicole: *Analysis of High-Dimensional Data with Partial Least Squares and Boosting*, Technische Universität Berlin, 2007

[Kroi04] Kroiss, Helmut: Vom Wert des Wassers in Zeiten der Globalisierung. In: *Wasser. Schatz der Zukunft. Impulse für eine nachhaltige Wasserkultur*. München : oekom verlag, Gesellschaft für ökologische Kommunikation mbH, 2004 — ISBN 3-936581-51-7, S. 30–36

[KuSt92] Kummert, Robert ; Stumm, Werner: *Gewässer als Ökosysteme: Grundlagen des Gewässerschutzes*. 3. Aufl. Stuttgart : vdf Verlage der Fachvereine an den schweizerischen Hochschulen und Techniken Zürich und B.G. Teubner, 1992 — ISBN 3-7281-1886-9

[Lane01] Laney, Doug: 3D Data Management: Controlling Data Volume, Velocity, and Variety. In: META Group (Hrsg.) , META Group (2001)

[Linu11] Linus Torvalds: *What would you like to see most in minix?–Google Groups*. URL https://groups.google.com/forum/#!msg/comp.os.minix/dlNtH7RRrGA/SwRavCzVE7gJ. [abgerufen am 2017-05-31]

[Litz13] Litz, Lothar: *Grundlagen der Automatisierungstechnik: Regelungssysteme - Steuerungssysteme - Hybride Systeme*. 2. Aufl. München : De Gruyter Oldenbourg, 2013 — ISBN 978-3-486-70888-2

[LuPr06] Lucas, Robyn ; Pruss-Ustun, Annette: *Solar ultraviolet radiation global burden of disease from solar ultraviolet radiation*. Geneva : World Health Organization, 2006 — ISBN 978-92-4-159440-0

[Mari07] Mariana Ruiz Villarreal: *Diagramm einer ganzen Nervenzelle*, 2007

[Mast33] Masters, Timothy: *Practical Neural Network Recipies in C++*. Boston : Academic Press Inc., 1933 — ISBN 978-0-12-479040-7

[Mcka12] McKay, Murray: Best practices in automation security. In: *2012 IEEE-IAS/PCA 54th Cement Industry Technical Conference*, 2012, S. 1–15

[MeGr11] Mell, Peter ; Grance, Timothy: The NIST definition of cloud computing (2011)

[Mitc97] Mitchell, Tom M.: *Machine Learning*. 1. Aufl. New York : McGraw-Hill Education, 1997 — ISBN 978-0-07-042807-2

[Modb06a] Modbus Organization: *MODBUS APPLICATION PROTOCOL SPECIFICATION V1.1b* : Modbus Organization, 2006

[Modb06b] Modbus Organization: *MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b* : Modbus Organization, 2006

[MoFS17] Mois, George ; Folea, Silviu ; Sanislav, Teodora: Analysis of Three IoT-Based Wireless Sensors for Environmental Monitoring. In: *IEEE Transactions on Instrumentation and Measurement* (2017), S. 1–9

[Moor03] Moore, Gordon E.: No Exponential is Forever: But „Forever" Can be Delayed! In: *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International* : IEEE, 2003, S. 20–23

[Moor65] Moore, Gordon: Cramming more components onto integrated circuits. In: *Electronics* Bd. 38 (1965), Nr. 8

[Mult15] Multicherry: *Top half of Raspberry Pi 2 Model B v1.1 viewed directly from above.*, 2015

[Ng16] Ng, Andrew: CS229 Machine Learning, Lecture Notes. In: *CS229 Lecture notes* (2016)

[Ng17] Ng, Andrew: *Coursera: Machine Learning*. URL https://www.coursera.org/learn/machine-learning. [abgerufen am 2017-05-31]

[Nxps14] NXP Semiconductors: UM10204: I2C-bus specification and user manual (2014)

[Open17] OpenBSD: *OpenSSH manual page*. URL http://man.openbsd.org/ssh. [abgerufen am 2017-05-31]

[Öwav13] ÖWAV: *Regelblatt 13: Betriebsdaten von Abwasserreinigungsanlagen – Erfassung, Protokollierung und Auswertung* : ÖWAV, 2013

[Parz62] Parzen, Emanuel: On Estimation of a Probability Density Function and Mode. In: *The Annals of Mathematical Statistics* Bd. 33 (1962), Nr. 3, S. 1065–1076

[Pfüt03] Pfützner, Helmut: *Angewandte Biophysik*. Wien : Springer, 2003 — ISBN 978-3-211-00876-8

[Poko13] Pokorny, Jaroslav: NoSQL databases: a step to database scalability in web environment. In: *International Journal of Web Information Systems* Bd. 9 (2013), Nr. 1, S. 69–82

[Post17a] The PostgreSQL Global Development Group: *PostgreSQL 9.6.3 Documentation*. URL https://www.postgresql.org/files/documentation/pdf/9.6/postgresql-9.6-A4.pdf. [abgerufen am 2017-05-31]

[Post17b] The PostgreSQL Global Development Group: *PL/pgSQL - SQL Procedural Language*. URL https://www.postgresql.org/docs/current/static/plpgsql-structure.html. [abgerufen am 2017-05-31]

[Prec10] Prechtl, Adalbert: Vorlesungen über Elektrodynamik (Skriptum). Wien, 2010

[Prof16] *PROFIBUS Systembeschreibung Technologie und Anwendung* : PI Support Center, 2016

[Psch82] Pschyrembel, Willibald / Zink, Christoph: *Pschyrembel Klinisches Wörterbuch mit klinischen Syndromen und Nomina Anatomica. 254. Auflage*. Berlin ; New York, 1982 — ISBN 978-3-11-007187-0

[Pyth17] Python Software Foundation: *Python 2.7.13 documentation*. URL https://docs.python.org/2/. [abgerufen am 2017-05-31]

[Rasp09] Raspberry Pi Foundation: *Raspberry Pi Documentation*. URL https://www.raspberrypi.org/documentation/. [abgerufen am 2017-05-31]

[Rdev11] R Development Core Team: *R: A Language and Environment for Statistical Computing*. Wien : R Foundation for Statistical Computing, 2011 — ISBN 3-900051-07-0

[Rdoc17] R Documentation: *R: Fitting Generalized Linear Models*. URL https://stat.ethz.ch/R-manual/R-devel/library/stats/html/glm.html. [abgerufen am 2017-05-31]. — Dokumentation des Pakets „stats“ Version 3.5.0

[Redd14] Reddy, Y. B.: Cloud-Based Cyber Physical Systems: Design Challenges and Security Needs. In: *2014 10th International Conference on Mobile Ad-hoc and Sensor Networks*, 2014, S. 315–322

[Reit10] Reitermanova, Zuzana: Data Splitting. In: *Proceedings of Contributed Papers*. Bd. 1. Prague, 2010 — ISBN ISBN 978-80-7378-139-2, S. 31–36

[Ried94] Riedmiller, Martin: Advanced supervised learning in multi-layer perceptrons—from backpropagation to adaptive learning algorithms. In: *Computer Standards & Interfaces* Bd. 16 (1994), Nr. 3, S. 265–278

[Roit15] Roithner Lasertechnik GmbH: *GUVA-S12SD Datasheet*. URL http://ams.com/eng/content/download/319364/1117183/287875. [abgerufen am 2017-05-31]

[Rstu17] RStudio, Inc.: *RStudio IDE*. URL https://www.rstudio.com/products/rstudio/. [abgerufen am 2017-05-31]

[RuNK12] Russell, Stuart ; Norvig, Peter ; Kirchner, Frank: *Künstliche Intelligenz: ein moderner Ansatz*, *it, Informatik*. 3. Aufl. München : Pearson, Higher Education, 2012 — ISBN 978-3-86894-098-5

[Scha12] Schaefer, Matthias: *Wörterbuch der Ökologie*. 5. neu bearbeitete und erweiterte Auflage. Aufl. Heidelberg : Spektrum Akademischer Verlag, 2012 — ISBN 978-3-8274-2561-4

[Schn09] Schneier, Bruce: The battle is on against Facebook and co to regain control of our files. In: *The Guardian* (2009)

[Shig11] shigeru23: *Moores law (1970-2011)*, 2011

[SHKS14] Srivastava, Nitish ; Hinton, Geoffrey E. ; Krizhevsky, Alex ; Sutskever, Ilya ; Salakhutdinov, Ruslan: Dropout: a simple way to prevent neural networks from overfitting. In: *Journal of Machine Learning Research* Bd. 15 (2014), Nr. 1, S. 1929–1958

[Simo83] Simon, Herbert A.: Why should machines learn? In: Michalski, R. S. ; Carbonell, J. G. ; Mitchell, T. M. (Hrsg.): *Machine Learning: An Artificial Intelligence Approach*. 2013-10-04. Aufl. Berlin : Springer, 1983 — ISBN 978-3-662-12407-9

[SLCZ15] Saldivar, Alfredo Alan Flores ; Li, Yun ; Chen, Wei-neng ; Zhan, Zhi-hui ; Zhang, Jun ; Chen, Leo Yi: Industry 4.0 with cyber-physical integration: A design and manufacture perspective. In: *2015 21st International Conference on Automation and Computing (ICAC)*, 2015, S. 1–6

[Spis16] SPI - Software in the Public Interest, Inc.: *Debian -- Das universelle Betriebssystem*. URL https://www.debian.org/index.de.html. [abgerufen am 2017-05-31]

[SPTS16] Shrivastava, Ashish ; Pfister, Tomas ; Tuzel, Oncel ; Susskind, Josh ; Wang, Wenda ; Webb, Russ: Learning from Simulated and Unsupervised Images through Adversarial Training. In: *arXiv:1612.07828 [cs]* (2016). — arXiv: 1612.07828

[SrGK15] Srivastava, Pragati Prakash ; Goyal, Saumya ; Kumar, Anil: Analysis of various NoSql database. In: *Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on* : IEEE, 2015, S. 539–544

[Stal07] Stallings, William: *Operating Systems: Internals and Design Principles*. 6. Aufl. Upper Saddle River, NJ : Prentice Hall, 2007 — ISBN 978-0-13-603337-0

[StRo08] Strobl, Robert O. ; Robillard, Paul D.: Network design for water quality monitoring of surface freshwaters: A review. In: *Journal of Environmental Management*, *Microbial and Nutrient Contaminants of Fresh and Coastal Waters*. Bd. 87 (2008), Nr. 4, S. 639–648

[SWLS83] Sanders, Thomas G. ; Ward, Robert C. ; Loftis, Jim C. ; Steele, Timothy D. ; Adrian, Donald D. ; Yevjevich, Vujica: *Design of Networks for Monitoring Water Quality*. Littleton, Colorado, U.S.A : Water Resources Pubns, 1983 — ISBN 978-0-918334-51-0

[Tasc15] Taschner, Rudolf: *Geometrie und Räume von Funktionen: mit zahlreichen Beispielen und 196 Aufgaben*, *Anwendungsorientierte Mathematik für ingenieurwissenschaftliche Fachrichtungen*. München : Fachbuchverl. Leipzig im Carl-Hanser-Verl, 2015 — ISBN 978-3-446-44245-0

[Texa15] Texas Instruments Incorporated: *TMP006/B Datasheet*. URL http://www.ti.com/lit/ds/symlink/tmp006.pdf. [abgerufen am 2017-05-31]

[Thom05] Thompson, C. W.: Smart devices and soft controllers. In: *IEEE Internet Computing* Bd. 9 (2005), Nr. 1, S. 82–85

[Tian12] Tianfield, Huaglory: Security issues in cloud computing. In: *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on* : IEEE, 2012, S. 1082–1089

[TsXi16] Tse, R. T. ; Xiao, Yubin: A portable Wireless Sensor Network system for real-time environmental monitoring. In: *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2016, S. 1–6

[Turi36] Turing, Alan Mathison: On computable numbers, with an application to the Entscheidungsproblem. In: *Proceedings of the London mathematical society* Bd. 2 (1936), Nr. 1, S. 230–265

[UhHo01] Uhlmann, Dietrich ; Horn, Wolfgang: *Hydrobiologie der Binnengewässer*. Stuttgart (Hohenheim) : UTB, Stuttgart, 2001 — ISBN 978-3-8252-2206-2

[Umwe17] Umweltbundesamt: *Wasserrahmenrichtlinie*. URL http://www.umweltbundesamt.at/umweltschutz/wasser/wrrl/. [abgerufen am 2017-05-31]

[VMGW87] Vanderkooi, J. M. ; Maniara, G. ; Green, T. J. ; Wilson, D. F.: An optical method for measurement of dioxygen concentration based upon quenching of phosphorescence. In: *The Journal of Biological Chemistry* Bd. 262 (1987), Nr. 12, S. 5476–5482

[Voge14] Vogel-Heuser, Birgit: Herausforderungen und Anforderungen aus Sicht der IT und der Automatisierungstechnik. In: *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Wiesbaden : Springer Fachmedien Wiesbaden, 2014 — ISBN 978-3-658-04681-1

[WaBa13] Ward, Jonathan Stuart ; Barker, Adam: Undefined By Data: A Survey of Big Data Definitions. In: *arXiv:1309.5821 [cs]* (2013), S. 572. — arXiv: 1309.5821

[Wcwo14] W3C, World Wide Web Consortium: *HTML5 A vocabulary and associated APIs for HTML and XHTML*. URL https://www.w3.org/TR/html5/. [abgerufen am 2017-05-31]

[West17] Westermo: *MRD-330*. URL http://westermo.com/web/web_en_idc_com.nsf/AllDocuments/DB9D1BFA332CCCFDC125789300340B8B. [abgerufen am 2017-05-31]

[WFKW14] Winkelbauer, Andreas ; Fuiko, Roland ; Krampe, Jörg ; Winkler, Stefan: Crucial elements and technical implementation of intelligent monitoring networks. In: *Water Science & Technology* Bd. 70 (2014), Nr. 12, S. 1926–1933

[WiFW12] Winkler, Stefan ; Fuiko, Roland ; Winkelbauer, Andreas: iTUWmon - A monitoring network platform for automated data plausibility assessment and data integration. In: *New Developments in IT & Water*, *Proceedings*. Amsterdam, 2012

[WiHo60] Widrow, Bernard ; Hoff, Marcian E.: *ADAPTIVE SWITCHING CIRCUITS* : STANFORD UNIV CA STANFORD ELECTRONICS LABS, 1960

[WiKr15] Winkelbauer, Andreas ; Krampe, Jörg: Water Quality Monitoring with a Raspberry Pi used as Data Visualization Appliance. In: *2nd IWA New Developments in IT & Water Conference*. Rotterdam, 2015

[WiMa03] Wilson, D. Randall ; Martinez, Tony R.: The general inefficiency of batch training for gradient descent learning. In: *Neural Networks* Bd. 16 (2003), Nr. 10, S. 1429–1451

[Wink11] Winkler, Stefan: Messtechnik für abwassertechnische Systeme – Stand der Technik, Anwendung und Nutzung. In: Kroiss, H. (Hrsg.): *Monitoring auf Kläranlagen: Daten*

erfassen, auswerten und anwenden, *Wiener Mitteilungen: Wasser, Abwasser, Gewässer*. Bd. 224 : Technische Universität Wien, Institut für Wassergüte, Ressourcenwirtschaft und Abfallwirtschaft, 2011 — ISBN 978-3-85234-134-7, S. 171–192

[Wink16] Winkelbauer, Andreas: Datenmanagement: Vom Sensor zum Report. In: Krampe, J. ; Svardal, K. (Hrsg.): *Neues aus der Mess-, Steuer- und Regelungstechnik*, *Wiener Mitteilungen: Wasser, Abwasser, Gewässer*. Bd. 239 : Technische Universität Wien, Institut für Wassergüte, Ressourcenwirtschaft und Abfallwirtschaft, 2016 — ISBN 978-3-85234-134-7, S. 53–78

[Wink17] Winkelbauer, Andreas: Vortragsskriptum zum ÖWAV MESSTECHNIKKURS: Signalübertragung und Skalierung (2017)

[WiSK17] Winkelbauer, Andreas ; Schaar, Heidemarie ; Kreuzinger, Norbert: Weitergehende Abwasserreinigung mit Ozon - praktische Implementierung eines Regelungskonzeptes. In: *11. Fachtagung MSR*. Wiesbaden-Niedernhausen : DWA, 2017

[WKFW13] Winkler, Stefan ; Kornfeind, Lukas ; Fuiko, Roland ; Winkelbauer, Andreas: Separating real system dynamics from measurement failures and artifacts by means of automated data quality assessment methods. In: *ICA 2013, 11th IWA Conference on Instrumentation Control and Automation*, *Proceedings*. Narbonne, 2013

[Worl08] World Meteorological Organization (Hrsg.): *Guide to hydrological practices*, *WMO*. 6. Aufl. Geneva, Switzerland : WMO, 2008 — ISBN 978-92-63-10168-6

[WSBT08] Winkler, Stefan ; Saracevic, Ernis ; Bertrand-Krajewski, Jean-Luc ; Torres, Andrés: Benefits, limitations and uncertainty of in situ spectrometry. In: *Water Science & Technology* Bd. 57 (2008), Nr. 10, S. 1651

[Xyle17] *Xylem Analytics Germany Sales GmbH & Co. KG*. URL https://www.wtw.com/de/home.html. [abgerufen am 2017-05-31]

[Zeil17] Zeileis, Achim: *rollmean function*. URL https://www.rdocumentation.org/packages/zoo/versions/1.8-0/topics/rollmean. [abgerufen am 2017-05-31]

[Zühl12] Zühlke, Detlef: *Nutzergerechte Entwicklung von Mensch-Maschine-Systemen*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012 — ISBN 978-3-642-22073-9

Erklärung

Hiermit erkläre ich, dass die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt wurde. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet.

Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder in ähnlicher Form in anderen Prüfungsverfahren vorgelegt.

Wien, am 13.06.2017

Andreas Winkelbauer, BSc.