

Max Hentges, *2000
Leopold Aschenbrenner, *2001

Schule:
John F. Kennedy Schule Berlin

Eingang der Arbeit:
Juni 2016

Zur Veröffentlichung angenommen:
November 2016

Eine App meldet dicke Stadtluft

Ein Überwachungs-, Warn- und Vorhersagesystem für Feinstaub am Beispiel von Berlin

Wir haben ein Überwachungs-, Warn- und Vorhersagesystem für die Feinstaubbelastung am Beispiel von Berlin gebaut. Dieses wertet in Echtzeit die Daten eines Netzes von selbstgebaute Sensoren aus und lädt sie auf einen Server. Jeder Nutzer kann mit einer Smartphone-App die Daten sichten. Weiterhin erstellt das Überwachungssystem mithilfe von „Machine Learning“ eine regional differenzierte Prognose der Feinstaubbelastung für die nächsten Tage.

1 Einleitung

Im Herbst 2015 wurden in Paris, Peking und anderen Städten drastische Maßnahmen, wie z. B. Fahrverbote oder Fabrikschließungen angeordnet, um die Emission von Feinstaub zu reduzieren. Nahezu gleichzeitig schätzten Lelieveld et al in einem Artikel in der Fachzeitschrift „Nature“, [1] dass jährlich etwa 3,5 Millionen Menschen vorzeitig wegen Luftverschmutzung mit Feinstaubpartikeln sterben. Dabei entfallen etwa 34.000 Todesfälle auf Deutschland, etwa 55.000 auf die USA und etwa 1.357.000 auf China [1].

Auch in Berlin ist diese Belastung an mehr als 30 Tagen im Jahr über dem EU-Maximalwert. Wir haben uns gefragt,

wie wir uns vor dieser Gefahr schützen und aktuell informieren können. Die Daten sind oftmals nur mit Verzögerungen verfügbar und stellen nur wenige punktuelle Belastungen dar: Berlin hat zum Beispiel nur elf Messstationen für PM10-Konzentration [2].

Außerdem gibt es keine einfache Möglichkeit, die Daten in zugänglicher Form, etwa auf dem Smartphone, einzusehen und zu visualisieren. Deshalb haben wir ein Überwachungs-, Warn- und Prognosesystem für die Feinstaubbelastung am Beispiel von Berlin gebaut. Dieses System kann jedoch auch kostengünstig und schnell in anderen Städten der Welt realisiert werden. Damit können in Zukunft die Bürger von Städten

aus der ganzen Welt vor dem gefährlichen Feinstaub gewarnt werden und sich schützen.

2 Bedeutung von Feinstaub

2.1 Was ist Feinstaub?

Feinstaub gibt es in zwei Varianten: primär emittierter Feinstaub und sekundär gebildeter Feinstaub. Dabei entsteht primärer Feinstaub direkt an der Quelle, etwa bei Verbrennungsprozessen in Kraftwerken, Öfen und Heizungen und vor allem im Straßenverkehr. Gasförmige Vorläuferstoffe aus der Landwirtschaft wie z. B. Schwefel- und Stickoxide oder Ammoniak [3] sind eine Hauptursache der sekundären Feinstaubbildung. Natürliche Quellen wie z. B. Emissionen aus Vulkanen, Bodenerosion oder Wald-

und Buschfeuer tragen ebenfalls zur Emission von Feinstaub bei [4,5]. Bei Feinstaub wird – vereinfacht gesagt – zwischen PM10 (PM, particulate matter) mit einem maximalen Durchmesser von 10 Mikrometer (μm) und PM 2,5 mit einem maximalen Durchmesser von 2,5 Mikrometer (μm) unterschieden.

2.2 Welche Auswirkungen hat Feinstaub auf die Gesundheit?

PM10 kann beim Menschen bis in die Nasenhöhle und Luftröhre, PM2,5 sogar bis in die Bronchien gelangen [6]. Je nach Größe und Eindringtiefe der Teilchen sind die gesundheitlichen Auswirkungen unterschiedlich. Diese reichen von lokalen Entzündungen bis hin zu kardiovaskulären Effekten, Atemwegserkrankungen oder erhöhtem Krebsrisiko [7]. Es ist nachgewiesen, dass Feinstaub bei Personen mit Lungenerkrankungen wie Asthma oder COPD (chronic obstructive pulmonary disease) eine Symptomverschlechterung und eine Verminderung der Lungenfunktion verursacht [8].

3 Komponenten des Überwachungs-, Warn- und Vorhersagesystems

Unser Überwachungs-, Warn- und Vorhersagesystem für Feinstaub besteht aus den Komponenten Messstation sowie Datenverarbeitungs-System. Die Messstation misst die Feinstaubkonzentration in der Umwelt. Ihre Messwerte sendet sie mithilfe der Bluetooth-Technologie an die Smartphone-App, die sie mit der geographischen Position an den Server schickt, oder mithilfe der WiFi-Technologie direkt an den Server. Jeder Nutzer, der die Smartphone-App installiert hat, kann nun die aktuellen Messdaten vom Server anschauen und visualisieren.

Weiterhin erlaubt die Smartphone-App Zugriff auf eine Prognose der Feinstaubkonzentration des nächsten Tages. Diese Prognose wird vom Server mithilfe eines „Machine-Learning“ Systems an die Smartphone-App geschickt. Schließlich überwacht der Server auch die Daten. Wenn die Feinstaubkonzentration zu hoch ansteigt, oder ein kritischer Anstieg prognostiziert wird, werden die Nutzer durch eine Push-Benachrichtung auf ihr Handy gewarnt.

4 Messmethoden

Grundsätzlich gibt es verschiedene tech-

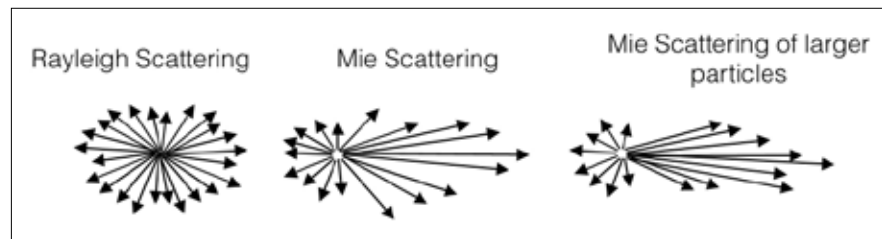


Abb. 1: Rayleigh und Mie Scattering [10].

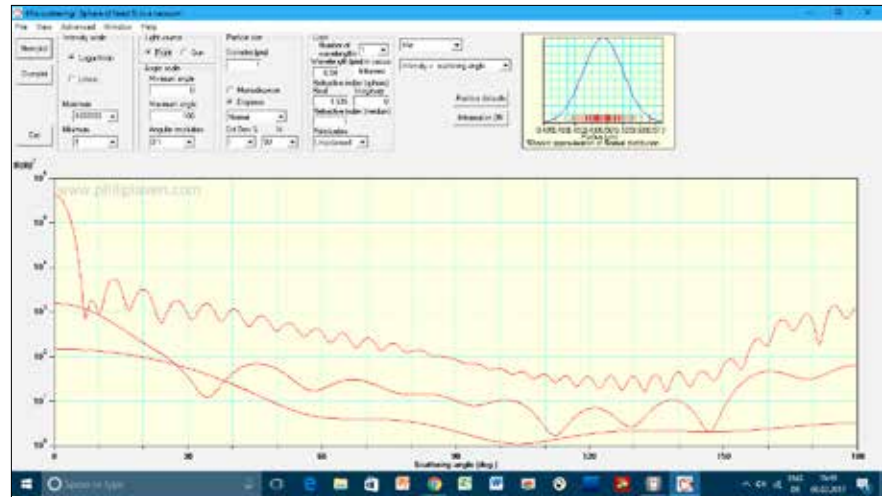


Abb. 2: Ergebnisse der Simulationsrechnung der Mie Streuung für unterschiedliche Partikelgrößen.

nische Möglichkeiten, um die Feinstaubkonzentration zu messen.

4.1 Gravimetrische Referenzverfahren

Das gravimetrische Referenzverfahren, das durch die 39. Bundes-Immissionsschutzverordnung (BImSchV) vorgeschrieben und durch Europäische Normen [9] beschrieben wird, ist eine Methode, um die Masse von Partikeln in der Luft zu messen. Hierzu wird die zu messende Luft mit den Partikeln durch einen speziellen Filter gesaugt. Die Partikel lagern sich auf dem Filter ab, dessen Gewichtszunahme anschließend gemessen wird. Die Massezunahme des Filters im Verhältnis zur angesaugten Luftmenge ist daher ein Maß für die Partikelkonzentration. Durch vorgeschaltete Abscheider kann die Messung auf z.B. Partikel $<10\mu\text{m}$ (PM10) oder $<2,5\mu\text{m}$ (PM2,5) beschränkt werden. Diese Messmethode liefert zwar genaue und gerichtlich anerkannte Ergebnisse, erfordert jedoch einen sehr hohen Aufwand sowohl bei der Gewichtsmessung, als auch bei der Messung des Luftstroms. Für die Ziele unseres Projekts dauert die Messung zu lange, die Messapparatur ist nicht leicht transportabel und auch viel zu teuer.

4.2. Light scattering Messprinzip

Eine andere Messmethode beruht auf dem physikalischen Prinzip der Streuung von Licht an kleinen Teilchen (siehe Abb. 1). Das eintreffende Licht kann das Medium, in dem die Partikel suspendiert sind, entweder ungestört durchdringen oder es wird an den Partikeln gestreut oder von diesen absorbiert. Für Teilchen, die deutlich kleiner sind als die Wellenlänge des einfallenden Lichts, hat Rayleigh dieses Phänomen beschrieben [10].

Für unsere Zwecke möchten wir jedoch Partikel mit einem Durchmesser im Bereich von 2,5 und $10\mu\text{m}$ messen. Sichtbares Licht hat eine Wellenlänge von 0,4 bis $0,7\mu\text{m}$ und selbst wenn wir infrarotes Licht mit einer Wellenlänge von z. B. $0,94\mu\text{m}$ verwenden, sind die Partikel noch um einen Faktor 10 größer als die Wellenlänge des verwendeten Lichts.

Auf Basis der Maxwell'schen Gleichungen hat Gustav Mie eine Formel entwickelt, um das Phänomen der Streuung von elektromagnetischen Wellen, wie z. B. Licht, an Partikeln zu beschreiben [11]. Dabei hängt das Maß der Streuung unter anderem von der Partikelgröße ab. Je kleiner die Partikel sind, desto inten-

siver ist die Streuung. Wenn die Partikel deutlich kleiner als die Wellenlänge sind ($d \ll \lambda$), entspricht das Ergebnis der Mie Streuung dem Rayleigh Phänomen (siehe Abb. 1). Um die Mie-Streuung zu berechnen, haben wir das Programm Mie Plot [12] benutzt. Als Wellenlänge des Lichtes haben wir die typische Wellenlänge einer Infrarot LED und für den Reflexionsgrad die typischen Werte von Aerosolen [13] in dem Medium Luft eingesetzt. Mit Hilfe des Mie-Plot Programms konnten wir dann die Intensität des gestreuten Lichts in Abhängigkeit vom Streuwinkel für verschiedene Partikelgrößen ermitteln (siehe Abb. 2). Hier kann man leicht erkennen dass, je kleiner die Partikel sind, desto geringer wird die Lichtintensität in Ausbreitungsrichtung (0°) und desto höher wird der Anteil des gestreuten Lichts.

5 Komponenten der Messstation

Die einzelnen Komponenten der Messstation und ihre Verbindungen sind in Abb. 3 dargestellt.

5.1 Partikelsensor

Von Elektronikherstellern sind eine Reihe von preiswerten industriellen Feinstaubsensoren erhältlich, die alle mit dem Lightscattering Messprinzip arbeiten. Wir haben hierzu zunächst mehrere Sensoren von Sharp [14] sowie einen Sensor von Telaire Amphenol in Erwägung gezogen. Während die Sharp Sensoren ein analoges Ausgangssignal liefern, das während eines genau definierten Zeitintervalls ausgelesen werden muss, liefert der Telaire Sensor ein viel einfacher durch einen Mikrocomputer zu verarbeitendes digitales Ausgangssignal. Der Telaire Sensor ähnelt auch sehr dem in der Literatur häufig beschriebene Shinyei Sensor [15]. Der Telaire Sensor besteht aus einer Fotodiode und einer LED. Zusätzlich ist ein Heizwiderstand eingebaut, um durch thermische Konvektion eine Luftströmung zu erzeugen. An der Fotodiode ist auch eine Auswertelektronik angeschlossen (siehe Abb. 4). Diese Elektronik erzeugt einen Low Impulse, wenn Partikel am Sensor vorbei strömen. Die Dauer und Anzahl der Low Impulse während einer Messperiode ("Low Pulse Occupancy Rate") stellt dabei ein Maß für die Partikelkonzentration dar. Weiterhin erlaubt der Sensor durch zwei unterschiedliche Ausgänge auch eine Diffe-

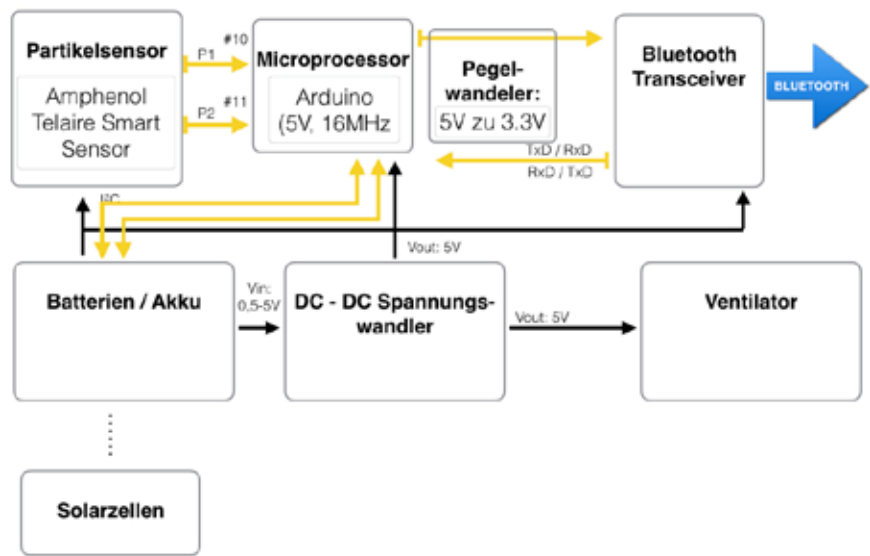


Abb. 3: Blockschaltbild der Messstation.

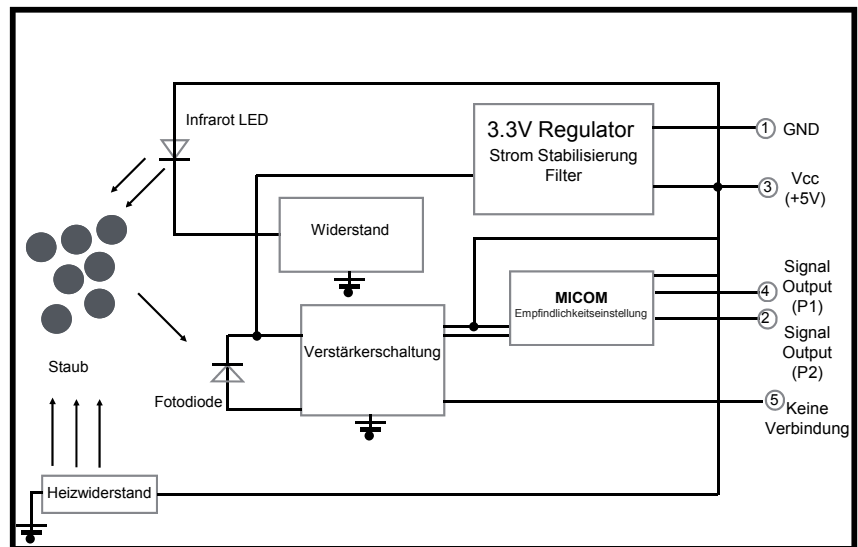


Abb. 4: Aufbau des Telaire Sensormoduls zur Messung von Feinstaub (Bild nach [16] erstellt).

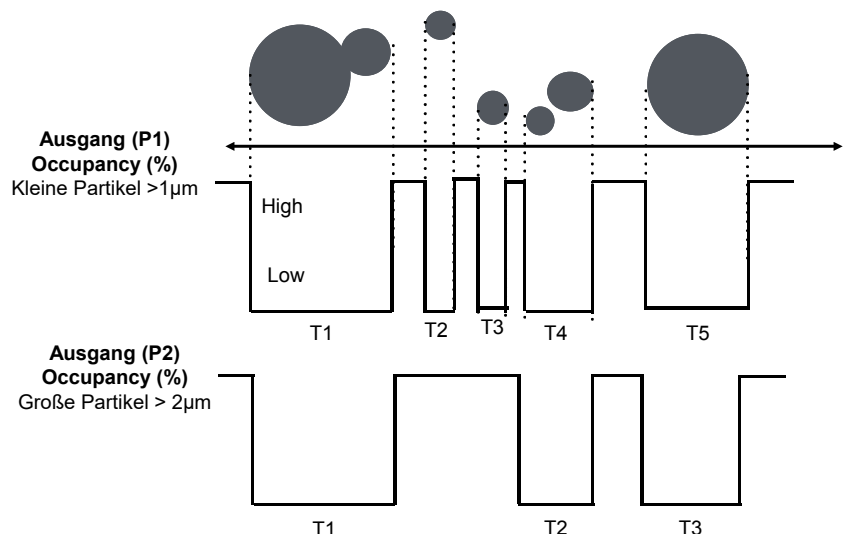


Abb. 5: Ausgangssignale des Telaire Sensors (Bild nach [17] erstellt).

renzierung zwischen kleinen Partikeln (1-2 μm Durchmesser) und größeren Partikeln mit einem Durchmesser von ca. 3 - 20 μm . (siehe Abb. 5).

5.2 Signalverarbeitung

Zur Signalverarbeitung unseres Projekts haben wir einen Arduino pro mini benutzt, da dieser kostengünstig ist und nur einen sehr geringen Platzbedarf hat. Ein Arduino ist ein kleiner Microcontroller auf Basis der Atmel Chipreihe ATmega, der im Jahr 2006 in Italien entwickelt wurde. Für den Arduino gibt es eine eigene Programmierumgebung oder Integrated Development Environment (IDE), und er lässt sich leicht mit einer an die Programmiersprache C++ angelehnten Computersprache programmieren. In dem Programm wird die Zeit, während am Sensorausgang ein „low state“ anliegt, gemessen.

Diese Zeiten werden während eines Messintervalls von sechs Sekunden aufaddiert. Die „Low Pulse Occupancy Rate“ ergibt sich, indem man die aufaddierte Zeit durch die gesamte Zeit des Messintervalls dividiert. Dieser Wert (in Promille) wird ca. alle sechs Sekunden an das verbundene Smartphone oder mit dem WiFi-Modul an den ThingSpeak Server geschickt.

5.3 Datenübertragung

Die serielle Schnittstelle TxD und RxD des Arduinos ist an ein Bluetooth Transceiver Modul angeschlossen. Da die Arduino Ausgänge Pegel von 5 V liefern und das Bluetooth Modul nur Eingangsspannungen von max. 3,3 V toleriert, muss ein Pegelwandler dazwischengeschaltet werden. Wir haben diesen Pegelwandler als Widerstandnetzwerk realisiert. Das Bluetooth Transceiver Modul baut eine Bluetooth Verbindung zu einem in der Nähe befindlichen Smartphone auf.

Neben dem mit dem Bluetooth 2.0 Standard operierenden Bluetooth Modul haben wir auch eine weitere Version der Messstation mit einem Bluetooth 4.0 Low Energy Modul gebaut. Dieser neuere Standard ist insbesondere für die Kommunikation mit Apple iPhones besser geeignet als der einfachere 2.0 Standard. Als weiteren Vorteil akzeptiert das von uns verwendete Mo-



Abb. 6: Messstation mit Ventilator (schwarzer Kasten auf der roten Abdeckung).

dul Eingangspiegel von bis zu 5 V, daher kann in dieser Version der Messstation der Pegelwandler entfallen. Eine dritte Variante, um die Messdaten zur Weiterverarbeitung zu senden, nutzt ein WiFi Netzwerk. Hierzu wurde statt des Bluetooth Moduls ein WiFi Transceiver Modul basierend auf dem ESP 8266 Mikrochip [18] eingesetzt. Die Messdaten zusammen mit den Identifikationsdaten der Messstation werden dann alle 30 Sekunden automatisch auf einen im Internet verfügbaren Server (ThingSpeak) [19] hochgeladen und können von dort zur Weiterverarbeitung abgerufen werden. So ist kein manuelles Hochladen des Nutzers erforderlich.

Hierbei eignet sich die WiFi Version der Messstation besonders für Nutzer in industriellen Ländern und Städten, wo WiFi Internet-Zugang ubiquitär verfügbar ist. Die Bluetooth Version der Messstation eignet sich besonders für Entwicklungsländer, wo meist nur ein Mobiltelefonnetz flächendeckend verfügbar ist. Grundsätzlich wäre es auch denkbar, z. B. für entlegene Orte die Signale der Messstation durch ein GSM-Mobilfunkmodem oder ein Iridium Satelliten-Funkmodem zu übertragen, wobei allerdings die Kosten hierfür unser Budget übersteigen [20].

5.4 Stromversorgung

Wir möchten, dass die Messstation mobil und unabhängig vom Stromnetz ist. Deshalb haben wir einen DC/DC Spannungswandler eingesetzt, der Eingangsspannungen zwischen 0,5 und 5 V in eine geregelte Ausgangsspannung von genau 5 V umwandelt. Mit dieser Spannung werden Sensor, Arduino, Bluetooth-Transceiver und Ventilator versorgt. Der Spannungswandler erhält die Eingangsspannung z.B. von einem

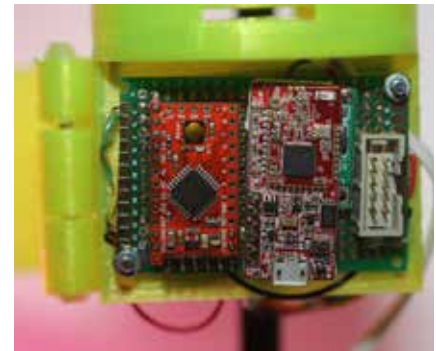


Abb. 7: Die elektrischen Komponenten der Messstation.

Akku. Für unsere ersten Messungen haben wir drei NiMH Akku-Zellen in Reihe geschaltet und so eine Spannung von ca. 3,6 V erhalten. Alternativ lässt sich auch eine Lithium-Ionen Zelle einsetzen, die eine Zellenspannung von ca. 3,6 V aufweist. Das geringere Leistungsgewicht einer solchen Zelle ist sicherlich von Vorteil, allerdings muss eine solche Zelle dann durch komplexere Schutzschaltungen vor Überladung geschützt werden. Der Akku kann durch Solarzellen aufgeladen werden. Ein vorhandenes Solarzellenmodul liefert je nach Sonnenintensität eine Spannung von ca. 3 bis 5 V, womit der Akku bei ausreichendem Sonnenlicht geladen werden kann. Zum Schutz der Solarzellen ist eine Diode in den Stromkreis eingebaut. Für einen netzunabhängigen Einsatz müssen Akkukapazität, Größe und Leistungsfähigkeit der Solarzellen sowie Häufigkeit der Messungen an den Standort angepasst werden.

5.5 Gehäuse mit Ventilator und Probeentnahme

Der gesamte Aufbau der Messstation ist in einem Gehäuse eingebaut, das mit einem 3D Drucker hergestellt wurde (siehe Abb. 6 und 7). Um einen kontinuierlichen Luftstrom zu erzeugen, haben wir in die Gehäuseabdeckung noch einen Ventilator eingebaut. Wir möchten nicht nur die Staubkonzentration messen, sondern auch dessen Herkunft bestimmen. Dazu kann in das Gehäuse zwischen Sensor und Ventilator ein Tesafilm-Streifen eingeklebt werden. Nach den Messungen kann man den Tesafilm entnehmen und unter einem Mikroskop untersuchen.

5.6 Kalibrieren des Sensors

Der Sensorhersteller stellt ein Kalibrierdiagramm zur Verfügung, mit des-

sen Hilfe sich aus der Low Pulse Occupancy Rate eine Partikelkonzentration ableiten lässt. Allerdings muss für unsere Zwecke das Kalibrierdiagramm angepasst werden. Durch den Einbau des Sensors in ein Gehäuse und den Einsatz eines Ventilators haben sich die Strömungsverhältnisse geändert. Weiterhin bestehen zwischen einzelnen Sensoren erhebliche produktionsbedingte Toleranzen und schließlich möchten wir zum Vergleich mit den Daten des Berliner Luftgütemessnetz (BLUME) ein Ergebnis in $\mu\text{g}/\text{m}^3$ erhalten. Hierzu haben wir in einem ersten Schritt unsere Messstation zusammen mit einem höherwertigen, kommerziellen Partikel-Zähler in eine aus einem Glas-Aquarium selbst gebaute Messkammer eingesetzt (siehe Abb. 8).

In dieser Messkammer werden unterschiedliche Umgebungen mit niedriger, mittlerer und höherer Feinstaubbelastung simuliert. Indem die Luft für die Messkammer durch Staubsaugerfilter und einen nachgeschalteten Feinstaubfilter angesaugt wird, lässt sich in der Messkammer nahezu ein Reinraum simulieren. Werden diese Filter jedoch vollständig entfernt und z. B. Zigarrenrauch in die Messkammer geleitet, so kann man eine Umgebung mit sehr hoher Feinstaubbelastung simulieren. Mit Hilfe einer Regressionsgerade lässt sich nun ein Umrechnungsfaktor zwischen der Low Pulse Occupancy Rate unserer Messstation und der von dem höherwertigen Messgerät angezeigten Partikelkonzentration ermitteln (siehe Abb. 9). Auch dieses Messgerät funktioniert nach der light scattering Methode und gibt daher eine Partikelkonzentration in Partikel pro Volumeneinheit an.

Allerdings wurde für dieses Messgerät in umfangreichen Versuchsreihen eine recht gute Korrelation ($R^2 = 0,81$) mit einem u.a. von der US Umweltschutzbehörde als Referenz-Messinstrument eingesetzten Geräts nachgewiesen [21]. Damit lässt sich nun die Low Pulse Occupancy Rate unserer Messstation in eine PM 2,5 Feinstaubkonzentration in $\mu\text{g}/\text{m}^3$ umrechnen. Eine perfekte Korrelation zwischen einer Methode, die die Masse der Partikel misst und einer Methode, die sich am Volumen orientiert, ist auch nicht zu erwarten. Unter-



Abb. 8: Selbstgebaute Feinstaub-Messkammer.

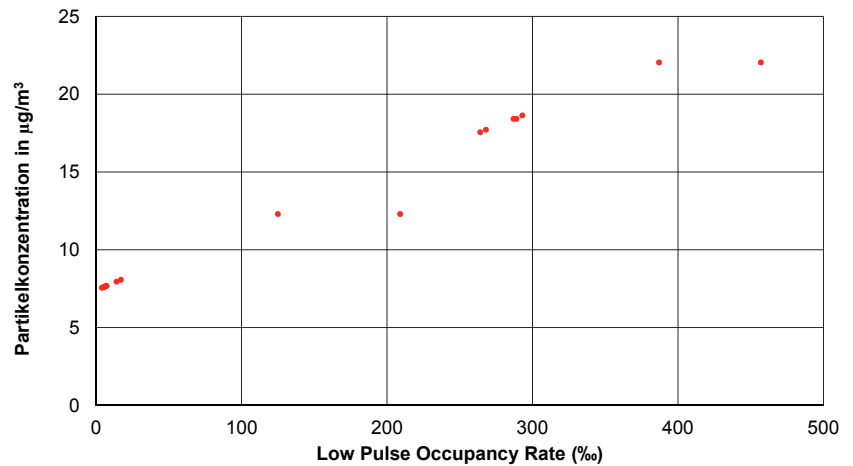


Abb. 9: Kalibrierkurve für unsere Messstation.

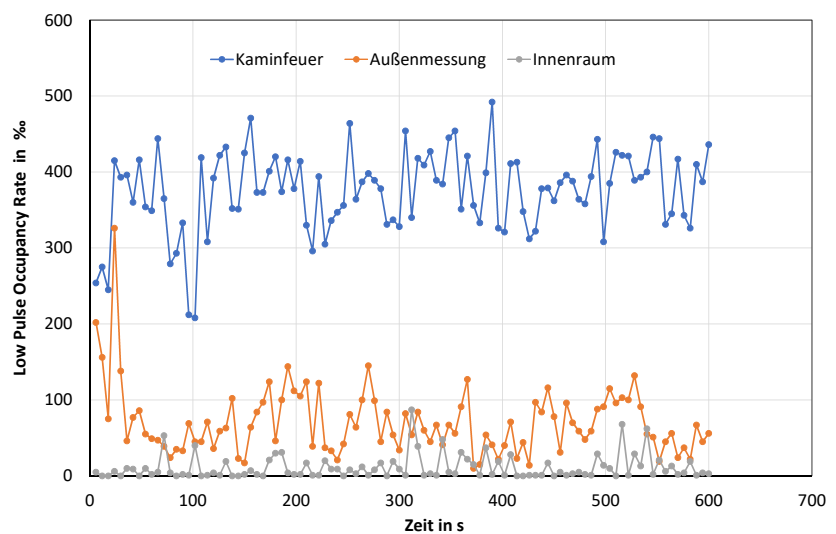


Abb. 10: Messung der Low Pulse Occupancy Rate (und damit der Partikelgröße) nahe eines Kaminfeuers, im Innenbereich und im Außenbereich

schiedliche Zusammensetzungen und damit Dichten des Feinstaubes führen daher auch zu unterschiedlichen Mes-

sergebnissen. Ein wichtiger Einflussfaktor ist dabei die Luftfeuchtigkeit: bei hoher Luftfeuchtigkeit haben die Fein-

staubpartikel mehr Masse, aber nicht viel mehr Volumen. Deshalb planen wir, zusätzlich auch noch Luftfeuchtigkeit und Temperatur durch weitere Sensoren zu erfassen, um so ggf. die Sensor-Messwerte zu korrigieren. Eine weitere Einschränkung ist, dass wir für die Vergleichsmessungen keine Reinraum-Messkammer hatten und daher insbesondere für geringe Partikelkonzentrationen diese Umrechnungstabelle noch verbessert werden sollte. Auch für sehr hohe Partikelkonzentrationen sind, u.a. wegen einer Sättigung des Sensors wohl Korrekturen an der Umrechnungstabelle zu erwarten.

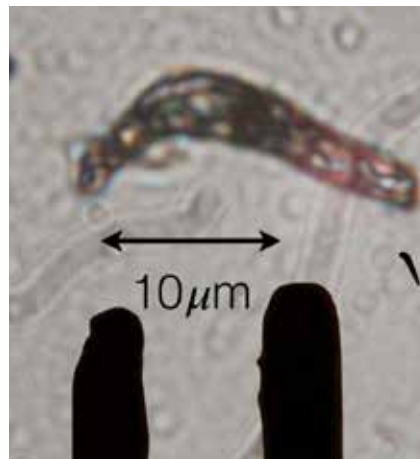


Abb. 11: längere, fadenartige Feinstaubpartikel.

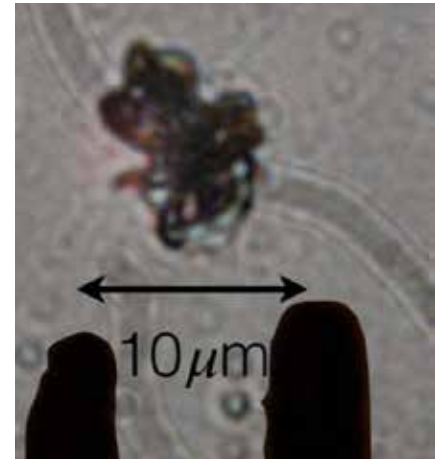


Abb. 12: eher kugelförmige Feinstaubpartikel.

5.7 Beispielmessungen

Die Auswertung der Messergebnisse unseres Messgeräts über 10 Minuten zeigt deutlich die unterschiedlichen Partikelbelastungen an. Während in der Nähe eines Kaminfeuers die Low Pulse Occupancy Rate Werte von ca. 375 Promille erreicht, betragen diese Werte bei einer Außenmessung im Durchschnitt ca. 71 Promille und bei einer Innenraummessung nur ca. 11 Promille (siehe Abb. 10). Dies stimmt mit den erwarteten Ergebnissen recht gut überein. Der Messwert bei der Außenmessung entspricht einer Partikelkonzentration von $14,1 \times 10^6$ Partikel/ m^3 und der Innenraummesswert einer Konzentration von $2,6 \times 10^6$ Partikel/ m^3 [22]. Zum Vergleich: für Raumluft geht man nach ISO von ca. 8×10^6 Partikel/ m^3 aus, wobei nur Feinstaub $\geq 1 \mu m$ berücksichtigt wird [23].

Weiterhin ermöglicht die Messstation auch, Partikel für eine mikroskopische Analyse zu sammeln. Bei einer ersten qualitativen Auswertung fallen zwei typische Bestandteile des aufgesammelten Feinstaubes auf, zum einen längere, fadenartige Gebilde, möglicherweise Fasern/ Abrieb und zum anderen kugelförmige, möglicherweise mineralische Gebilde (siehe Abb. 11, 12). Mit Hilfe eines Objekt-Mikrometers mit 1/100 mm Teilung lässt sich auch die Größe der einzelnen unter dem Mikroskop untersuchten Partikel ermitteln, indem mit gleicher Mikroskop-, Kamera- und Bildvergrößerung sowohl die zu untersuchende Feinstaub-Stichprobe als auch das Objekt-Mikrometer dargestellt werden.

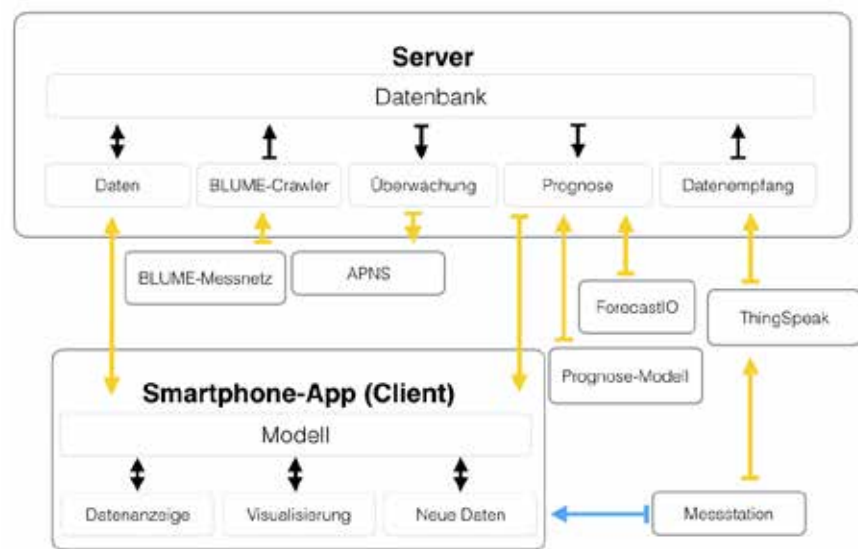


Abb. 13: Schema des Datenverarbeitungssystems.

6 Datenverarbeitung

6.1 Aufbau

In Abb. 13 ist der vereinfachte Aufbau des Datenverarbeitungs-Systems unseres Projekts dargestellt. Dabei werden sowohl die einzelnen Module des Servers und der Smartphone-App gezeigt als auch ihre jeweiligen Verbindungen (schwarz). In blau wird die Bluetooth-Verbindung zu der Messstation gekennzeichnet. In orange werden Netzwerkverbindungen zwischen dem Client und dem Server, als auch den APIs (Application Programming Interface) von ForecastIO, APNS (Apple Push Notification Service) und dem Prognose-Modell dargestellt. Die einzelnen Verbindungen, die durch Pfeile angezeigt werden, stellen auch die jeweilige Richtung des Datenflusses dar. Wir werden im Folgenden die Funktion und Interaktion der einzelnen Komponenten erklären.

6.2 Client (Smartphone-App)

Die Smartphone-App ist für das Betriebssystem iOS (iPhone Operating System) gebaut. Dazu haben wir die objektorientierte Sprache Swift benutzt, die 2013 von Apple Inc. vorgestellt wurde und auf Objective-C basiert.

Client: Neue Daten: In dieser Ansicht und dem zuständigen Modul wird zuerst eine Bluetooth-Verbindung zu einer Messstation hergestellt. Dazu benutzen wir das „CoreBluetooth“ Modul, das vom iOS-SDK (Software Development Kit) bereitgestellt wird. Die Messstation schickt ca. alle 6 Sekunden den aktuellen Wert der „Low Pulse Occupancy Rate“. Dieser Wert wird nun in eine PM10-Konzentration umgerechnet. Das „Neue Daten“ Modul erfragt nun den Standort des Nutzers. Der Standort und ein 30-Sekunden Mittel der PM10 Konzentration werden anschließend an

das Modell überreicht. Außerdem zeigt diese Ansicht die Ausgabe des Sensors als PM10-Konzentration an.

Client: Datenanzeige: Dies ist die Hauptansicht der Smartphone-App und wird dem Nutzer direkt beim Öffnen der App gezeigt (siehe Abb. 14). Mithilfe des „CoreLocation“ Moduls des iOS-SDK wird der Standort des Benutzers abgefragt und durch einen „reverse geocode“ der Name des Standorts ermittelt und dargestellt. Weiterhin kommuniziert dieses Modul mit dem Modell, um die Daten der Feinstaubkonzentration in der Nähe abzufragen. Die aktuelle PM10-Belastung wird in $\mu\text{g}/\text{m}^3$ angezeigt. Weiterhin erscheint ein Zeitstrahl, der horizontal-scrollend die Tagesmittelwerte der Feinstaubbelastung in der Vergangenheit und die Prognose angibt. Durch eine Farbkodierung kann der Nutzer die Belastung einschätzen.

Client: Visualisierung: In dieser Ansicht kann der Nutzer der Smartphone-App die aktuelle Feinstaubbelastung visualisieren. Dieses Modul benutzt die Komponenten „MapKit“ und „CoreLocation“ des iOS-SDK, um eine Karte anzuzeigen. Diese zeigt in blau den aktuellen Standort des Nutzers. Mithilfe des Modells zeigt die Karte weiterhin eine Auswahl relevanter Messstationen mit ihrer Feinstaubbelastung an (siehe Abb. 15). Beim Tippen auf eine Messstation erscheinen weitere Informationen über die Belastung an diesem Ort, wie etwa ein Zeitstrahl und ein Graph der Belastungshistorie. Die Nutzer der Smartphone-App können außerdem mithilfe eines Sliders die Progression der Feinstaubbelastung einfach visuell nachvollziehen.

Modell: Das Modell bildet das Rückgrat des Clients, indem es die Daten der App verwaltet und verarbeitet. Dieses Modul verbindet sich mithilfe einer Netzwerkverbindung mit dem Server, um die diversen Daten vom Server herunterzuladen, Prognosen abzufragen und neue Daten von einer Messstation auf den Server hochzuladen. Nach dem Herunterladen der Daten bereitet das Modell die Daten auf, damit die Controller, also die diversen anderen Module, die Daten leicht darstellen können. Die Netzwerkverbindung ist asynchron, läuft also nicht auf dem „Main Thread“, um eine flüssige graphische Oberfläche zu ermöglichen. Eine Kommunikation mit den Controllern



Abb. 14: Datenanzeige der Smartphone-App.

funktioniert mithilfe von „Delegation“: verschiedene Protokolle dienen der standardisierten, intramodularen Kommunikation.

6.3 Server

Der Server läuft mithilfe des Services Parse [24]. Wichtigste Komponenten sind eine Datenbank, in der Standort und Feinstaubmesswerte gespeichert sind, sowie ein Cloud Code, in dem diese Daten, zusammen mit weiteren externen Daten, zu Überwachungs- und Warnhinweisen sowie zu Prognosen verarbeitet werden. Die Datenbank des Servers speichert zentral die Daten aller Messstationen. Dazu benutzt sie zwei Tabellen, die relationell verknüpft sind. Jedes Objekt in der Datenbank hat jeweils die Eigenschaften „createdAt“ (Datum und Zeit der Erstellung des Objekts), „updatedAt“ (Datum und Zeit der Aktualisierung des Objekts) und „uniqueID“ (eine einzigartige Kennung).

Die erste Tabelle enthält die jeweiligen Messstationen, die mit den Eigenschaften „latitude“ (Breitengrad) und „longitude“ (Längengrad) gespeichert werden. Die zweite Tabelle speichert alle Messungen mit der Eigenschaft „PM10“ (die PM10-Konzentration) und „date“ (Datum und Zeit der Messung) ab. In Zukunft sollen hier auch noch Temperatur- und Luftfeuchtigkeitsmessdaten unserer Messstationen gespeichert wer-



Abb. 15: Anzeige aller Messstationen mit ihrer qualitativen Feinstaubbelastung.

den. Schlussendlich hat jeder Standort in der Tabelle „Standorte“ eine Eigenschaft „particles“, die eine Relation zur Tabelle „Partikel“ darstellt. So wird jede Partikelmessung mit einem bestimmten Standort/Messstation verknüpft. Auf dem Parse Server kann auch Programmcode in der Programmiersprache Javascript exekutiert werden. Wir benutzen diverse Elemente dieses „CloudCodes“, um den Client zu entlasten, Datenbankoperationen durchzuführen, das BLUME-Messnetz zu crawlen und zur Erstellung der Prognose.

Dieses Datenmodul des CloudCodes erleichtert die Kommunikation zwischen dem Client-Modell und der Datenbank. Um ein flüssige Nutzererfahrung zu gewährleisten, muss unser Programm schnell exekutieren. Doch die Datenverbindungen von Smartphones können nur eine beschränkte Menge von Daten pro Zeit herunterladen, besonders unterwegs. Deshalb agiert das Datenmodul als Mittelsmann, um diverse Zugriffe auf den Server vorher zu filtern und zu optimieren. So gibt es u.a eine Funktion, die vom Client-Modell aufgerufen wird, die die Feinstaubdaten bei einem bestimmten Standort aufruft. Sie errechnet einen geeigneten normierten Wert aus nur den nächsten Stationen und sendet dem Modell nur genau die Daten der Tage, die angefragt wurden. Dieses Modul übernimmt außerdem die Speicherung von

neuen Daten, indem es dem passenden Objekt in der „Standort“-Tabelle den neuen PM10-Wert zuordnet. So kann Bandbreite in der Übermittlung zum Smartphone gespart werden und der Nutzer der App eine flüssigere und schnellere Anzeige erhalten. Das Modul Datenempfang des CloudCodes verbindet sich alle 30 Sekunden mit dem ThingSpeak API, um die aktuellsten Messwerte unserer Messstationen, die mit WiFi Messdaten hochladen, abzufragen. Diese Daten leitet es dann an das Datenmodul weiter, das die Messwerte der passenden Messstation in der Datenbank zuordnet. Das Berliner-BLUME Messnetz misst mittels 11 Stationen in Berlin die PM10-Konzentration. Diese Werte sind nur in einer Auflösung von Tagesmittelwerten vorhanden und können Berlin nur unzureichend abdecken, um die genauen, punktuellen Feinstaubbelastungen darzustellen. Jedoch können sie unsere Messdaten im Rahmen von Open Data/Data Fusion ergänzen. Das BLUME-Crawler Modul des CloudCodes lädt jeden Tag die Messdaten des Vortages von der BLUME-Webseite [25] herunter. Der Crawler durchforstet dann die HTML-Tabellen, um nur die relevanten Messdaten in die Datenbank einzufügen.

In Zukunft sollen auch noch andere öffentliche Quellen in unsere Datenbank integriert werden. Das Prognose-Modul des CloudCodes interagiert mit dem JSON (JavaScript Object Notation)-API von ForecastIO. ForecastIO ist ein Service, der die Wetterdaten von verschiedenen Wetterdiensten kombiniert und durch ein JSON-API mithilfe einer Netzwerkverbindung verfügbar macht. Dadurch extrahiert dieses Modul meteorologische Daten wie Windgeschwindigkeit, Windrichtung, Temperatur und Luftfeuchtigkeit. Diese Daten übermittelt dieses Modul an das Microsoft Azure API, welches die vorhergesagte Feinstaubbelastung zur Weiterleitung an das Smartphone freigibt. Das Modul Überwachung des Servers fragt kontinuierlich die aktuellen Feinstaubwerte ab. Dieses Modul schickt dann mithilfe einer Netzwerkverbindung zum APNS (Apple Push Notification Services) eine Push-Benachrichtung an die jeweiligen Clients, ähnlich wie etwa beim Empfang einer SMS. Der Nutzer kann beim Client einstellen, ob er regelmäßig, etwa jeden

Tag um 10 Uhr bei Überschreiten von bestimmten Grenzwerten, etwa $50 \mu\text{g}/\text{m}^3$, oder einer Prognose von besonders hohen Feinstaubbelastungen benachrichtigt werden möchte.

6.4 Prognose

6.4.1 Faktoren

Für unsere Prognose haben wir eine Vielzahl von Parametern untersucht. Die Feinstaubbelastung vom Vortag wird benutzt, um eine generelle Orientierung der aktuellen Feinstaubbelastung zu erhalten. Die Differenz der Feinstaubbelastung (in $\mu\text{g}/\text{m}^3$) des Vor-Vortages mit der Feinstaubbelastung des Vortages hilft, eine Tendenz der Veränderung der Feinstaubbelastung zu erhalten. Weiterhin wird die Windgeschwindigkeit (in m/s) mit einbezogen, um austauscharme Wetterverhältnisse zu erkennen. Außerdem spielt die Windrichtung (in $^\circ$) eine kritische Rolle, denn Wind aus dem Osten und Südosten trägt viel Feinstaub aus Polen und der Tschechischen Republik, wo durch ältere Heizmethoden die Feinstaubbelastung besonders hoch sein kann, nach Berlin.

Wir haben auch den Effekt der relativen Feuchtigkeit (in %) analysiert. Weiterhin kann ein Kesseffekt entstehen, wenn die Lufttemperatur (in $^\circ\text{C}$) in der Höhe größer als in Bodennähe ist. So steigt keine warme Luft nach oben und es entstehen austauscharme Luftverhältnisse. Leider konnten wir keine Daten für die Lufttemperatur in verschiedenen Höhen erhalten, weshalb wir nur die Lufttemperatur in Bodennähe mit in unser Modell einbeziehen konnten. Schließlich spielt auch noch das Datum eine Rolle. Die Feinstaubbelastung variiert besonders in Straßennähe je nach Wochentag. Die Feinstaubbelastung ist im Winter erhöht, da Heizungen Feinstaub emittieren können. Zunehmend steigt die Feinstaubbelastung allerdings auch durch Freiluftfeuer im Sommer [26]. Weiterhin haben wir auch die Wolkenbedeckung analysiert und zwischen urbanen, suburbanen und ruralen Regionen unterschieden.

6.4.2 Machine Learning

Um unsere Prognose zu erstellen, benutzen wir Machine Learning. Machine Learning ist ein hochmoderner Unterbereich der KI (Künstlichen Intelligenz). Beim Machine Learning, oder Data Mining, untersuchen Computer große Daten-

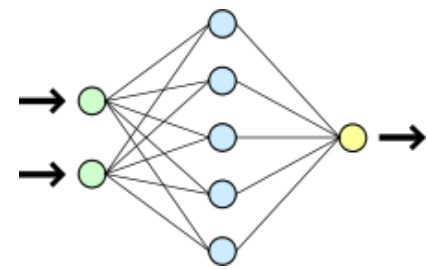


Abb. 16: Stark vereinfachte Darstellung eines neuronalen Netzes mit Eingabeebene (grün), Neuronen (hellblau) und Ausgabebene (gelb): [33]

mengen, um aus neuen Daten Schlüsse zu ziehen. Solche Technologien sind mittlerweile ein essentieller Bestandteil unseres täglichen Lebens, etwa beim Spam-Filter für Emails und Amazon-Empfehlungen. Dabei gibt es verschiedene Technologien, zum Beispiel „neural networks“, welche die Interaktion zwischen Neuronen simulieren, und „decision trees“, wo verschiedene Bedingungen Abzweigungen erzeugen und zu einer Entscheidung führen. Diese können komplexe, nicht-lineare Zusammenhänge modellieren.

Für unser Machine Learning Modell haben wir die oben aufgelisteten Parameter verwendet. Vom BLUME-Messnetz haben wir die Daten der Feinstaubbelastung in den Jahren 2005-2015 von urbanen, suburbanen und ruralen Messstationen erhalten. Die jeweiligen meteorologischen Daten haben wir von ForecastIO [27], das Wetterdaten der NOAA (Nation Oceanic and Atmospheric Administration) [28], des Norwegian Meteorological Institute [29], des Canadian Meteorological Center [30] und mehr [31] kombiniert, bezogen. Wir haben zuerst unseren Datensatz verarbeitet. Da es in unseren Daten mehrere Instrumentenwechsel gab, wurden die Daten vor und nach dem Instrumentenwechsel normiert. Daten für Tage, wo Messwerte durch Versagen eines Instruments gefehlt haben, wurden gelöscht. Danach haben wir unseren Datensatz so aufgeteilt, dass der erste Teil, der Trainingsdatensatz, 80 (zufällig ausgewählte) Prozent der Daten enthält und der zweite Teil, der Testdatensatz, die restlichen 20 Prozent der Daten enthält. Wir haben zwei verschiedene Algorithmen, die von Microsoft im Rahmen der Azure Cloud Platform [32] entwickelt wurden und mit dem Prinzip der Artificial Neural Networks bzw. Boosted Decision Trees funktionieren, mithilfe unseres Trainings-

datensatz trainiert und gegenüber unseres Testdatensatzes analysiert. Artificial Neural Networks (ANNs) bestehen aus einem Netz künstlicher Neuronen und zugehörigen Synapsen, die ihr biologisches Vorbild approximieren. Abb. 16 zeigt eine vereinfachte Darstellung eines neuronalen Netzes mit einer Eingabeebene (grün), die aus den Werten der Einflussfaktoren besteht, einer versteckten Ebene, die aus „Neuronen“ besteht (hellblau), und einer Ausgabebene (gelb), die eine Prognose ausgibt. Jedes „Neuron“ wird mit „Synapsen“ mit allen anderen „Neuronen“, die nicht in der gleichen Ebene sind, vernetzt. Dabei leiten die Eingabeneuronen ihren Wert über die „Synapsen“ an die „Neuronen“ in der versteckten Ebene weiter. Diese kombinieren die erhaltenen Werte und wenden darauf eine Funktion an. Dann leiten sie den Ausgabewert an das Ausgabeneuron weiter. Jedes der „Synapsen“ hat jedoch ein Gewicht: ein Faktor, mit der weitergeleitete Wert multipliziert wird. Diese Gewichte werden von Methoden der multidimensionalen Optimierung der Differentialrechnung anhand des Trainingsdatensatzes generiert [34]. Unser Artificial Neural Network besteht aus fünf versteckten Ebenen mit jeweils 100 „Neuronen“. Es wurde mithilfe unseres Trainingsdatensatzes in 1000 Iterationen trainiert und dann anhand unseres Testdatensatzes analysiert. Es ergab sich ein durchschnittlicher absoluter Fehler von $5,47 \mu\text{g}/\text{m}^3$ und ein R^2 , bzw. Bestimmungsmaß, von 0,72.

Als Vergleich: ein R^2 von 1 zeigt an, dass das Modell die abhängige Variable vollständig erklären kann, während ein R^2 von 0 anzeigt, dass das Modell keinen Zusammenhang mit der abhängigen Variabel hat. Ein Decision Tree, oder Entscheidungsbaum, besteht aus diversen Knoten bzw. Abzweigungen, die eine binäre Bedingung darstellen. Diese führen schlussendlich zu einem „Blatt“ bzw. Ausgabewert/Prognose. Abb. 17 zeigt ein stark vereinfachtes Beispiel eines Decision Trees. Ein Boosted Decision Tree besteht aus einem Ensemble, oder Sammlung, von vielen einfachen Decision Trees, deren gewichteter Mittelwert die Prognose ergibt. Dabei versucht der Computeralgorithmus, dass die jeweiligen Decision Trees im Ensemble ihre Fehlerquellen ausgleichen [35]. Unser Boosted Decision Tree besteht aus

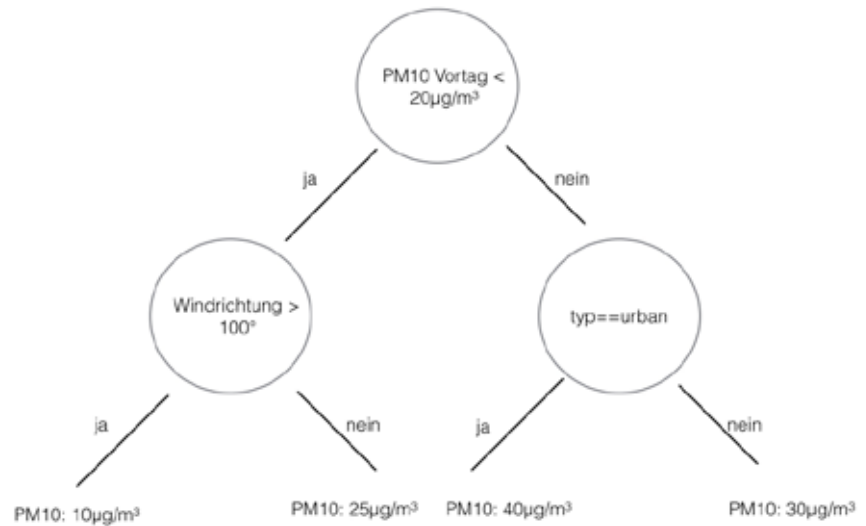


Abb. 17: Stark vereinfachte Darstellung eines Decision Tree.

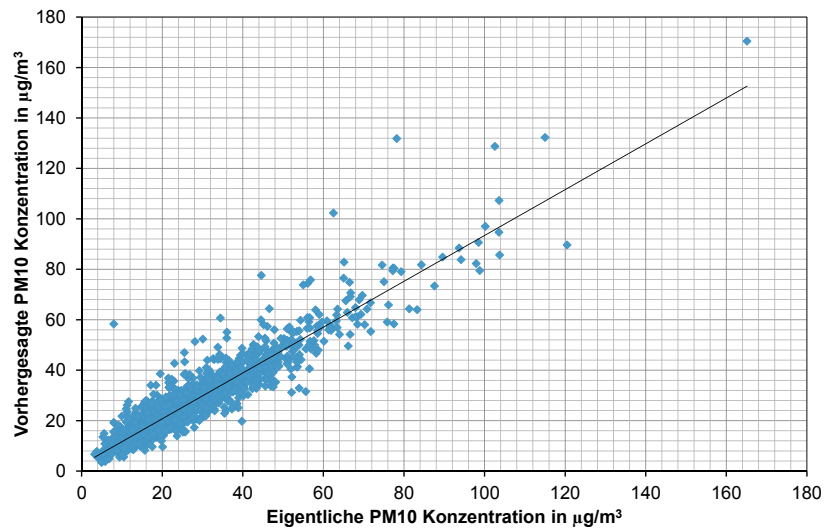


Abb. 18: Auswertung des Prognosesystems.

2000 verschiedenen Entscheidungsbäumen mit jeweils maximal 30 „Blättern“. Er wurde mithilfe unseres Trainingsdatensatzes trainiert und dann anhand unseres Testdatensatzes analysiert. Es ergab sich ein durchschnittlicher absoluter Fehler von $3,59 \mu\text{g}/\text{m}^3$ und ein R^2 , bzw. Bestimmungsmaß, von 0,87. Diesen Boosted Decision Tree haben wir dann als unser Prognosemodell verwendet und mithilfe eines JSON-APIs bereitgestellt, damit unser Server darauf zugreifen kann.

6.4.3 Analyse

Eine Permutational Analysis unseres Machine Learning Modells zeigt uns, welche Faktoren für die Prognose ausschlaggebend sind. Dabei sehen wir, dass 34,9 Prozent der Prognose durch den PM10-Wert des Vortags erklärt

wird. Der PM10-Wert des Vortags bezüglich des PM10-Werts hat den Pearson Korrelationskoeffizienten von 0,71. Daraus können wir schlussfolgern, dass der Vortagswert dem Algorithmus eine grobe Orientierung der Feinstaubbelastung gibt. Weiterhin hängt 17,8 Prozent der Prognose von der Windrichtung ab, wobei das Ausgleichspolynomial zum 4. Grad der Windrichtung bezüglich des PM10-Werts ein R^2 von 0,13 hat. Das bestätigt, dass die Windrichtung die Belastung durch Feinstaub aus dem Osten aufzeigt. Weitere 11,1 Prozent der Vorhersage hängen von der Temperatur ab, wobei das Ausgleichspolynomial zum 2. Grad der Temperatur bezüglich des PM10-Werts ein R^2 von 0,13 hat. Das deutet darauf hin, dass möglicherweise eine Inversionswetterlage bei kalten Temperaturen die Feinstaubbelastung erhöht.

Darüber hinaus hängen 10,6 Prozent der Prognose von der Windgeschwindigkeit ab, wobei die Windgeschwindigkeit gegenüber des PM10-Werts einen Korrelationskoeffizienten von $-0,2411$ hat. Das bestätigt, dass niedrigere Windgeschwindigkeiten austauscharme Luftverhältnisse erzeugen und so die Feinstaubbelastung steigt. 9,6 Prozent der Prognose hängen vom Datum ab und 9 Prozent der Prognose von der Wolkenbedeckung, wobei die Wolkenbedeckung gegenüber dem PM10-Wert den Korrelationskoeffizienten $-0,14$ hat. Das deutet darauf hin, dass eine niedrige Wolkenbedeckung die Freigabe von sekundärem Feinstaub aus Vorläuferstoffen durch erhöhte Sonnenstrahlung fördert. 7,1 Prozent der Feinstaubbelastung hängt von der Luftfeuchtigkeit ab. 7 Prozent der Prognose geht aus dem Typ (rural, suburban oder urban) der Region hervor. Dabei ergibt sich gegenüber dem PM10-Wert ein Korrelationskoeffizient von $0,28$, was die Vermutung der höheren Feinstaubbelastung in Ballungsgebieten bestätigt.

Wir konnten also durch statische Analyse unserer Prognose und unseres Datensatzes die Einflussfaktoren für Feinstaub aus unserer Diskussion mit Experten der Senatsverwaltung für Stadtentwicklung und Umwelt bestätigen und neue Einflussfaktoren erklären. Außerdem wird deutlich, welche komplexen geowissenschaftliche Systeme zur Feinstaubbelastung beitragen und durch unser Machine-Learning System modelliert werden können.

6.4.4 Vergleich mit anderen Systemen

Bei der Auswertung des Prognosemodells anhand des Testdatensatzes ergibt sich ein durchschnittlicher absoluter Fehler von $3,59 \mu\text{g}/\text{m}^3$ und ein R^2 , bzw. Bestimmungsmaß, von $0,87$ (siehe Abb. 18). Das Prognosesystem kann also die PM10-Konzentration des nächsten Tages bereits recht genau vorhersagen. Ein komplexes statisches Modell der TU

Graz zur Prognose der PM10-Konzentration in Graz-Mitte erreichte ein R^2 von $0,7$ [35]. Ein Artificial Neural Network von Forschern der University of Bari zur Prognose der PM10-Konzentration in Barcelona erreichte ein R^2 von $0,73$ [36]. Das Europäische Ausbreitungs- und Depositionsmodell, EURAD [37], der Universität Köln prognostiziert mittels einer Computersimulation unter anderem die PM10-Konzentration in Europa. In der Literatur wird dieses Modell oft als der neueste Stand der Technik gehandelt. Je nach Auswertzeitraum erreicht dieses Modell jedoch ein Root Mean Squared Error von $10-18 \mu\text{g}/\text{m}^3$ [38] – unser Prognosesystem hat ein Root Mean Squared Error von $5,37 \mu\text{g}/\text{m}^3$. Weiterhin kann dieses System nur grobe, gering auflösende Aussagen über etwa einen Mittelwert für eine ganze Stadt geben. Unser Prognosesystem, kombiniert mit unseren dezentralen Messstationen, kann potentiell eine regional deutlich differenzierte Prognose abgeben. Somit können Punktbelastungen von Feinstaub dargestellt und von Nutzern umgangen werden.

7 Zusammenfassung und Ausblick

Unser vorgestelltes System aus Messstationen, Smartphone-App und komplexer Datenaufbereitungssoftware ermöglicht es, Überwachungs-, Warn- und Vorhersagesystem für die Feinstaubbelastung bereitzustellen. Dieses Projekt ist ein Beispiel, das zeigt, wie man mit Hilfe von günstigen Bauteilen – die Hardware-Kosten für eine Messstation liegen unter EUR 100 – in Verbindung mit den fast ubiquitär verfügbaren Smartphones, Cloud-Computing und Machine-Learning ein äußerst komplexes Mess- und Warnsystem aufbauen kann.

Für Städte wie Berlin, die bereits über ein Messnetzwerk verfügen, können mit unserem System viel engaschigere Messwerte, z. B. in der direkten Wohnumgebung ermittelt werden. Weiterhin stellt unser System Daten nahezu in Echtzeit

zur Verfügung und kann mit Hilfe von Wetterinformationen und historischen Daten eine regional differenzierte Prognose der Feinstaubbelastung erstellen. Auch in Städten und Regionen, in denen bisher noch keine Messstationen vorhanden sind, wie z. B. in vielen Entwicklungs- und Schwellenländern, kann so kostengünstig, schnell und dezentral ein Überwachungs-, Warn- und Vorhersagesystem aufgebaut werden. Unseren Prototypen wollen wir an mehreren Punkten zu verbessern. Bei der Messstation gehört hierzu die Verbesserung der Messgenauigkeit durch weitere Vergleichsmessungen. Um das Datenverarbeitungssystem zu verbessern, wollen wir unser Programm weiterentwickeln und stabiler machen. Das Benutzerinterface sollte durch weitere Visualisierungsmöglichkeiten, wie etwa Diagramme und Karten mit farbkodierten Flächen und Wetterdaten ergänzt werden. Außerdem sollen noch differenziertere Gesundheitsempfehlungen dargestellt werden. Als Verbesserung des Prognosesystems wollen wir weitere Parameter, wie etwa den Temperaturverlauf nach Höhe und die aktuelle, regionale Verkehrslage untersuchen. Schlussendlich sollte sich das Machine Learning Modell kontinuierlich anhand der neuen Daten verbessern.

Danksagung

Wir bedanken uns bei allen, die uns bei unserem Jugend Forscht Projekt unterstützt und geholfen haben. Speziell danken wir Professor Albrecht von Stülpnagel und Michaela Preuss von der Senatsverwaltung für Stadtentwicklung und Umwelt, Berlin für die Datensätze Feinstaub Berlin 2005 bis 2015 sowie die Diskussion von Einflussfaktoren für Feinstaubkonzentration, den Juroren des Regional- und Landeswettbewerbs – hier insbesondere Julian Petrasch – für ihre konstruktiven Anregungen und dem Sponsorenpool Jugend Forscht Berlin für die finanzielle Unterstützung unseres Wettbewerbsprojekts.

Quellenverzeichnis

- [1] Lelieveld, J., J. S. Evans, M. Fnais, D. Giannadaki & A. Pozzer: 1 The contribution of outdoor air pollution sources to premature mortality on a global scale, Nature 525, 367–371 (abgefragt 17 September 2015)
- [2] Von insgesamt 16 Messstationen, Senatsverwaltung für Stadtentwicklung und Umwelt
- [3] Umwelt Bundesamt. Aus welchen Quellen stammt Feinstaub?. <http://www.umweltbundesamt.de/service/ubafragen/aus-welchen-quellen-stammt-feinstaub> (abgefragt 25 Januar 2016)

- [4] United States Environmental Protection Agency. PM designations. <https://www3.epa.gov/pmdesignations/> (abgefragt 25 Februar 2015) (abgefragt 25 Januar 2016).
- [5] Umwelt Bundesamt. "Aus welchen Quellen stammt Feinstaub?". <http://www.umweltbundesamt.de/service/ubafragen/aus-welchen-quellen-stammt-feinstaub> " (abgefragt 25 Januar 2016)
- [6] Landesamt für Natur, Umwelt und Verbraucherschutz Nordrhein-Westfalen. Gesundheitliche Wirkungen von Feinstaub und Stickstoffdioxid im Zusammenhang mit der Luftreinhalteplanung. http://www.lanuv.nrw.de/fileadmin/lanuv/gesundheit/schadstoffe/gesundheitliche_wirkungen.pdf
- [7] Landesamt für Natur, Umwelt und Verbraucherschutz Nordrhein-Westfalen. Gesundheitliche Wirkungen von Feinstaub und Stickstoffdioxid im Zusammenhang mit der Luftreinhalteplanung. http://www.lanuv.nrw.de/fileadmin/lanuv/gesundheit/schadstoffe/gesundheitliche_wirkungen.pdf
- [8] Speizer, F.: Environmental Lung Diseases in Fauci et. al. : Harrison's Principles of Internal Medicine, 14th edition, New York, 1998
- [9] Beuth. DIN EN 12341
- [10] Nave, R. Rayleigh scattering. Hyperphysics <http://hyperphysics.phy-astr.gsu.edu/hbase/atmos/blusky.html>. (abgefragt 25 Januar 2016).
- [11] Mie, Gustav. Beiträge zur Optik trüber Medien, speziell kolloidaler Metallösungen. Annalen der Physik 4, no. 25 (1908)
- [12] Laven, P.: <http://www.philiplaven.com/mieplot.html>
- [13] Levoni, Chiara, Marco Cervino, Rodolfo Guzzi, and Francesca Torricella. Atmospheric aerosol optical properties: a database of radiative characteristics for different components and classes. Journal of Applied Optics 36, no. 30 (October 20, 1997)
- [14] SHARP Corporation Sensor GP2Y1 bzw. GP2UO6
- [15] Austin E, Novoselov I, Seto E, Yost MG (2015) Laboratory Evaluation of the Shinyei PPD42NS Low-Cost Particulate Matter Sensor. PLoS ONE 10(9): e0137789. doi: 10.1371/journal.pone.0137789
- [16] Amphenol Telaire Smart Sensor Application Note, St. Marys, Pennsylvania, August 2014
- [17] Amphenol Telaire Smart Sensor Application Note, St. Marys, Pennsylvania, August 2014
- [18] WiFi Modul ESP8266 von Espressif Systems
- [19] www.thingspeak.com
- [20] z.B. Iridium Satellitenmodem und Datenservice angeboten von Rock Seven Mobile Services Ltd
- [21] South Coast AQMD: Field Evaluation of the Dylos DC 1100 Pro. o.O., o.J. Web. 28 Feb. 2016, S. 4
- [22] Technische Universität Graz Erzherzog-Johann-Universität. Modellierung und Prognose der Feinstaubkonzentration in Graz-Mitte für die Zeitperiode Herbst/Winter. <http://www.stat.tugraz.at/stadl/papers/host04.pdf>
- [23] International Organization for Standardization. Cleanroom classification. www.iso.org 14644-1.
- [24] www.parse.com
- [25] Senatsverwaltung für Stadtentwicklung und Umwelt. Tageswerte der letzten 7 Tage. <http://www.stadtentwicklung.berlin.de/umwelt/luftqualitaet/de/messnetz/tageswerte/>
- [26] Gesprächsnotiz Interview mit Prof. von Stülpnagel und M. Preuss, Senatsverwaltung für Stadtentwicklung und Umwelt (14 Januar 2016)
- [27] www.forecast.io
- [28] National centers for Environmental Information. Data access model. <https://www.ncdc.noaa.gov/data-access>
- [29] API Norway. Weather 27 API. <http://api.met.no/#english>
- [30] National Weather Services. CMC. http://nomads.ncep.noaa.gov/txt_descriptions/CMCENS_doc.shtml
- [31] Forecast io. <http://forecast.io/raw/>
- [32] Microsoft Azure. Machine Learning. <https://azure.microsoft.com/en-us/services/machine-learning/>
- [33] https://upload.wikimedia.org/wikipedia/commons/thumb/3/3d/Neural_network.svg/2000px-Neural_network.svg.png
- [34] UT Toronto. What are Artificial Neural Networks? <http://www.psych.utoronto.ca/users/reingold/courses/ai/cache/neural2.html>" (abgefragt 24 März 2016).
- [35] Technische Universität Graz Erzherzog-Johann-Universität. Modellierung und Prognose der Feinstaubkonzentration in Graz-Mitte für die Zeitperiode Herbst/Winter. <http://www.stat.tugraz.at/stadl/papers/host04.pdf>
- [36] Universität Köln. Aktuelle Luftqualitätsvorhersagen der EURAD-Gruppe. <http://www.eurad.uni-koeln.de>
- [37] Universität Köln. Aktuelle Luftqualitätsvorhersagen der EURAD-Gruppe. <http://www.eurad.uni-koeln.de>
- [38] Copernicus. Quarterly report on the daily analyses and forecasts activities, and verification of the EURAD-IM performances. https://atmosphere.copernicus.eu/sites/default/files/FileRepository/Resources/Documentation/Regional_System/CAMS50_2015SC1_D50.5.1.EURAD-IM_201601.pdf