Handbuch

1. Management

Herzlichen Glückwunsch zu Ihrer Entscheidung, moderne Analysemethoden in Ihrem Werk zum Einsatz kommen zu lassen!

Um aus der Analysennutzung das Beste zu machen, schlagen wir vor, dass Sie sich etwas Zeit nehmen, um sich mit dem verfahrenstechnischen Prozess vertraut zu machen und darüber nachzudenken, wie Sie diese Lösung am besten in Ihre bestehende Organisation integrieren. Ein neues Leistungspotenzial einzusetzen ist weitaus komplexer, als nur ein neues Softwarepaket zu installieren.

Bei Ihrer Planung sollten Sie u.a. Folgendes berücksichtigen:

1. Eine Software-Installation braucht IT-Experten, die das Programm zum Laufen bringen und die Berechtigungen einrichten.
2. Ingenieure werden die Daten auswählen, die modelliert werden sollen.
3. Prozessexperten werden das Modell überprüfen, werden es feinjustieren und über seine Eignung entscheiden.
4. Die Anwender der Software müssen trainiert und angeleitet werden, um sie optimal zu nutzen.
5. Das Management wird über die Anwendungspolitik zu entscheiden haben.
6. Aktionspläne müssen im Rahmen eines Change Managements entworfen und umgesetzt werden, damit die Ergebnisse der Analysen auch praktisch genutzt werden können.
7. Der gewonnene Nutzen und die gemachten Erfahrungen müssen dem Unternehmen als Feedback rückgekoppelt werden.
8. Mit algorithmica technologies wird man Kontakt aufnehmen, wenn es nötig ist, das Modell oder die Software den eigenen Erfordernissen kundenspezifisch anzupassen.

Wir möchten Sie deshalb dazu ermuntern, dies alles als eine Aufgabe des Projektmanagements anzusehen, bei dem es Stakeholders, ein Führungsteam, einen Projektmanager und diverse Teamplayer gibt. Wir behandeln in diesem Handbuch das Thema des Projektmanagements nicht im Detail und gehen vielmehr davon aus, dass Ihr Unternehmen diese Kompetenz bereits beherrscht.

Die Einführung eines Analyse-Programms wird jedenfalls für eine Reihe von Leuten Veränderungen ihrer Arbeitsumgebung notwendig machen. Es wird also eine Umstellung erfordern, diese Mitarbeiter mit neuen Arbeitsmethoden vertraut zu machen und die neue Technologie in den Arbeitsablauf zu integrieren. Ein Mindestmaß an Change Management wird somit erforderlich sein, um das Programm im Unternehmen fest zu verankern.

1.1. Projektplan

Das Projekt, die neue Software in Ihrem Unternehmen einzuführen, beinhaltet sechs Phasen:

1. Die Software muss installiert und konfiguriert werden. Das wird im Kapitel über

die Installation näher beschrieben.
2. Es muss festgelegt werden, auf welche Tags das Modell ausgerichtet werden soll, und für jedes dieser Tags müssen einige Informationen definiert werden. Welche Informationen das sind, wird im Installations-Kapitel kurz angesprochen und im Umsetzungs-Kapitel näher ausgeführt.
3. Historische Daten müssen vorbereitet und für die ausgewählten Tags hochgeladen werden. Das Modell wird erlernt auf der Basis dieser historischen Daten.
4. Das Modell wird verwendet und bewertet aufgrund seiner Qualität und seiner Tauglichkeit. Allerdings wird dann noch eine Feinjustierung der Modellparameter notwendig sein, damit das Modell die tatsächliche Situation präzise beschreibt und das anstehende Problem auch tatsächlich lösen kann.
5. Ist das Modell ausreichend angepasst, kann die Software in produktiver Weise eingesetzt werden, indem man sie an die Quellen der aktuellen Daten anschließt, so dass sie periodisch die Antworten auf die gestellten Fragen berechnen kann.
6. Jetzt ist zwar die Aufgabe der Anwendungs-Installation beendet, doch beginnt nun – mit Hilfe des Change-Managements – die Aufgabe, die Lösung in den Arbeitsprozess des Unternehmens zu integrieren.

1.2. Organisation

Es wird nachdrücklich empfohlen, ein Projektmanagement-Team ins Leben zu rufen, um die moderne analytische Software im Unternehmen einzuführen. Wie oben bereits kurz angedeutet, wird dieses Projekt es nötig machen, dass eine Reihe von technischen und innerbetrieblichen Funktionsträgern an diesem Prozess teilnehmen und einige Entscheidungen getroffen werden, die ein gemeinschaftliches Agieren dieser Funktionsträger erfordern.

Dem Projekt sollte ein Projektmanager vorstehen, der die Verantwortung dafür übernimmt, dass die nötigen Aufgaben erfüllt werden, und der auch die Autorität innehat, die am Projekt Beteiligten aufzufordern, ihre jeweiligen Inputs zu geben. Das Projekt kann mittels Leistungsindikatoren (engl. key performance indicators, oder KPI) gesteuert werden, die von der Software vorgeschlagen werden – wie etwa die Zahl der Optimierungsvorschläge, die Zahl der bereits umgesetzten Vorschläge, die Genauigkeit des Modells usw .

Die Phasen des Projektes wurden weiter oben bereits kurz eingeführt. Phasen 1 und 5 werden federführend von der IT übernommen und erfordern die Kooperation der IT-Abteilung. Phase 3 ist im Wesentlichen die Aufgabe der Person, die die historischen Daten verwaltet; aber auch hier könnte die Zusammenarbeit der IT-Abteilung notwendig sein.

Phasen 2 und 4 erfordern tiefergehende Kenntnisse und Erfahrungen im Bereich von Prozessveränderungen. Wir empfehlen dazu, Phase 2 als Workshop durchzuführen, der vom Prozessmanager sowie von Ingenieuren und Anlagenbetreibern besucht wird. Diese Personen können dort im Detail die notwendigen Informationen besprechen (wie sie im Kapitel über Know-How beschrieben werden). Phase 4 beinhaltet die Bewertung der Qualität und Passgenauigkeit des Modells bzw. auch seine Feinjustierung für den Fall, dass die Qualität noch verbessert werden muss. Wir

empfehlen, dass dieser Schritt von einem Prozess-Experten durchgeführt wird. Im Allgemeinen wird ein Prozess-Experte seine persönliche Philosophie haben, wie die Dinge funktionieren, und diese Philosophie wird ihn auch bei der Abstimmung des Modells leiten. Dieser Person sollte man ausreichend Zeit geben, um der Aufgabe gerecht zu werden und sich in die Details einzuarbeiten. Diese Phase stellt einen kritischen Schritt dar, um die Software korrekt zu konfigurieren. Zuvor unterlaufene Fehler und Versäumnisse können jetzt noch korrigiert werden. Fehler, die in dieser entscheidenden Phase noch gemacht werden, können später zu Problemen führen.

Phase 6 ist die Phase des Change Managements. Hier wird das Modell in die tagtäglichen Abläufe der Anlage eingeführt. Das könnte es erfordern, Richtlinien und Verfahrensabläufe zu verändern und entsprechende Management-Absicherungen einzuholen. In jedem Fall wird es nötig sein, die Anlagenfahrer davon zu überzeugen, die Modelle zu akzeptieren und ihnen auch zu vertrauen. Damit die Modelle wirklich von Nutzen sind, sollten die Software-Ergebnisse entsprechende menschliche Handlungen nach sich ziehen. Bei Vorhandensein einer Advanced Process Optimizer (APO) Software müssen die diesbezüglichen Vorschläge umgesetzt werden. Im Falle von IHM beispielsweise muss man den Alarmsignalen nachgehen, und es werden gegebenenfalls auch Wartungsarbeiten notwendig sein. Das Management und die Anlagenfahrer müssen sich über die zu verfolgende Vorgehensweise verständigen und darin übereinstimmen. Diese Phase ist der kritische Punkt, an dem es sich entscheiden wird, ob die Software wirklich nutzbringend zum Einsatz kommt. Wir empfehlen, dass der Prozess der Nutzer-Anwendung gemessen wird, und zwar mit objektiven, numerischen KPIs – wie etwa die Zahl der umgesetzten Vorschläge (APO) oder die Zahl der Alarmsignale, auf die reagiert wurde (IHM).

Der ganze Prozess kann vernünftigerweise innerhalb von drei Monaten umgesetzt werden. Wir empfehlen, dass das Projektteam einen ebensolchen Zeitrahmen anstrebt; denn die Erfahrung hat gezeigt, dass die Projektzufriedenheit und die Benutzerakzeptanz bei viel längeren Projektlaufzeiten sinkt.

2. Installation

Dieses Kapitel beschreibt, wie die Software so zu installieren ist, damit sie korrekt läuft. Nach der Installation muss die Software allerdings noch für den speziellen Anwendungsfall konfiguriert werden, wie dies im Kapitel über die individuellen Lösungen beschrieben wird.

2.1. Systemvoraussetzungen

Die Software-Anwendung läuft auf einem 64-Bit Microsoft Windows Betriebssystem. Wir empfehlen Windows Server 2008R2 oder eine spätere Version. Der Server sollte für die historischen und die zukünftigen Daten genügend Speicherplatz haben. Dazu empfehlen wir mindestens 1 TB freie Speicherkapazität. Prozessor und Speicher können handelsüblich sein, etwa ein i7-Prozessor und ein 16GB RAM-Speicher. Falls Rechnergeschwindigkeit ein Problem darstellen sein sollte, kann man dies durch eine schnelle high-end GPU Grafikkarte von Nvidia erreichen, weil wir deren CUDA/CULA-Support nützen können. Das ist viel effizienter, als wenn man eine high-end CPU kauft.

Wir empfehlen, dass der Server ein RAID-System beinhaltet, um Datenverlust wegen Speicherausfall zu verhindern. Wir empfehlen ferner, dass regelmäßig ein Daten-Backup der Datenbank vorgenommen wird, um einen Systemverlust auszuschließen.

Die Firewall zwischen dem OPC-Server und der Anwendung muss ebenso geöffnet sein wie die Firewall zwischen der Anwendung und dem Office Netzwerk des Nutzers.

algorithmica benötigt ein Nutzerkonto des Servers mit Administratorenrechten.

Am Tag der ersten Software-Installation benötigt der Server vollen Internetzugang, um die Ruby-on-Rails-Umgebung korrekt zu installieren. Nach dieser anfänglichen Installation kann der Internetzugang für alle Zeit gekappt werden.

Um fortlaufende Updates/Upgrades-Dienste zu leisten, Beratung zu geben und kundenspezifische Anpassungen vornehmen zu können, benötigt algorithmica Fernzugriff auf den Server.

Hier ist eine Checkliste der notwendigen Voraussetzungen:

- Hardware
  - Mindestens 1 TB freier Speicherplatz
  - Standardprozessor, z.B. i7 920 oder besser
  - Mindestens 16GB RAM
  - Optional eine NVIDIA GPU
  - Optional ein RAID-System
  - Optional ein regelmäßiger Daten-Backup
- Software
  - 64-Bit Microsoft Windows Betriebssystem
  - algorithmica Nutzerkonto mit Administratorenrechten
  - Offene Firewall zwischen OPC-Server und der Anwendung
  - Offene Firewall zwischen der Anwendung und dem Office Netzwerk
  - Internetzugang am Tag der Software-Installation
  - Fernzugriff für algorithmica

algorithmica ist eine Software-Firma. Wir stellen Hardware weder zur Verfügung noch verwalten wir diese. Die IT-Organisation des Nutzers muss die notwendige Hardware zur Verfügung stellen, sie konfigurieren und die Server-Umgebung instandhalten.

2.2. Installation

Die Software-Installation wird vollautomatisch vom Installer-Programm vorgenommen, das Ihnen von algorithmica technologies zur Verfügung gestellt wird. Dazu brauchen Sie nichts weiter zu tun, als Ihren Computer mit einer Internetverbindung auszustatten und den Installer auszuführen.

Der Installer wird die folgenden Programme installieren

1. PostgreSQL: Die Datenbank-Management-Software, die sämtliche Daten enthält.
2. pgAdmin: Die Administrationskonsole für PostgreSQL.
3. Ruby: Die Sprache hinter dem Nutzer-Interface.

4. Ruby on Rails: Das Framework, mit dem das Nutzer-Interface geschrieben wurde.
5. Bundler: Der Package-Manager für Ruby on Rails.
6. Git: Ein Versionierungswerkzeug, mit der aktuelle Versionen von Plug-ins geladen werden.
7. Node.js: Eine Javascript-Bibliothek, die für das Nutzer-Interface benötigt wird.
8. Sqlite: Eine Datenbank-Management-Software, die zwar nicht angewendet wird, aber einen integralen Teil von Rails darstellt.
9. SQL Server Support: Notwendiges Rails-Hilfsprogramm, um die Verbindung zur Datenbank herzustellen.
10. Rails DevKit: Notwendiges Rails-Hilfsprogramm, um einige Plug-ins zusammenzustellen, die in Codeform dargestellt werden.
11. Gems: Eine große Zahl von Erweiterungen für Ruby on Rails, bekannt als "gems".
12. AI: Die Software von algorithmica technologies für maschinelles Lernen, welche die Analyse hinter den Anwendungen vornimmt.
13. Softing OPC: Der OPC Client, der zum Datenlesen verwendet wird, nutzt das OPC Toolkit der Softing AG, für den algorithmica eine Entwicklungs-Lizenz hat.

Alle diese Programme sind lizensiert durch Lizenzen von MIT, BSD oder GNU LPGL. Diese können für kommerzielle Zwecke ohne Zusatzkosten genutzt werden. Alle diese Programme werden nur verwendet, um Daten zu speichern und ein Nutzer-Interface anzubieten. Alle mathematischen Analysen werden von der Software durchgeführt, die algorithmica technologies entworfen hat – und zwar ohne Verwendung von Komponenten Dritter.

Der Installer erstellt auf dem Computer ein neues Nutzer-Konto mit dem Namen "postgres" und dem Passwort "admin". Dieses Konto ist wichtig und wird von der Datenbank benötigt. Innerhalb der Datenbank wird der Installer einen neuen User erstellen mit Namen "viewer" und dem Passwort "viewer".

2.3. Netzwerk Einrichtung

Die Software ist nun lokal installiert. Wollen Sie jetzt die Software testen, brauchen Sie mit dem Einrichten des Netzwerkes nicht fortzufahren. Wollen Sie hingegen von anderen Computern aus die Oberfläche anschauen, ist das Einrichten des Netzwerkes notwendig.

Die Oberfläche überträgt seine Webseite auf Port 3000. Damit Sie von einem anderen Computer die Oberfläche anschauen können, benötigt der andere Computer Zugang zur IP-Adresse des Anwendungscomputers, und der Anwendungscomputer muss in der Lage sein, seine Inhalte zurückzusenden. Das könnte es erforderlich machen, die Firewalleinstellung zu verändern. Weil dies von der Netzwerkkonfiguration abhängt, sollten Sie diesbezüglich mit Ihrer IT-Abteilung sprechen. Am Anwendungscomputer muss Port 3000 für das HTML-Protokoll geöffnet sein, um seine Inhalte zu übertragen.

2.4. Testen

Bitte überprüfen Sie die erfolgreiche Installation, indem Sie Ihren Browser starten und eingeben: "http://localhost:3000". Jetzt sollten Sie eine Login-Seite sehen. Das

ist die Oberfläche. Loggen Sie sich ein als "admin" und mit dem Passwort "admin". Wenn dieses Login funktioniert, ist die Software korrekt lokal installiert.

Um zu überprüfen, ob die Netzwerkeinrichtung auch richtig funktioniert, laden Sie die Seite erneut, indem Sie die IP-Adresse des Computers in denselben Browser tippen, d.h. "http://[IP-Adresse]:3000". Wenn Sie jetzt bei derselben Login-Seite landen, funktioniert auch die Netzwerkeinrichtung, und Sie sollten jetzt in der Lage sein, von jedem anderen Computer Ihres Büronetzwerkes auf die Oberfläche zuzugreifen, wenn Sie die IP-Adresse in den Browser eingeben. Sollte das jedoch nicht funktionieren, obwohl der vorherige Schritt erfolgreich war, so müssen die Netzwerkeigenschaften geändert werden. Bitte setzen Sie sich mit Ihrer IT-Abteilung in Verbindung, um in dieser Sache unterstützt zu werden.

Der Installer wird für das Interface auch einen Windows-Service einrichten. Wird der Computer je neu gestartet, wird auch die Oberfläche automatisch zusammen mit dem Neustart des Computers gestartet. Somit kann auf die Oberfläche jederzeit zugegriffen werden, sobald der Computer eingeschaltet ist.

Damit ist die Software-Installation beendet.

2.5. Problembehandlung

In dieser Phase kann es verschiedene Probleme geben:

1. *Der Installer stürzt ab mit einer Fehlermeldung.*Bitte überprüfen Sie, ob Sie Administratorenrechte für den Computer besitzen, auf dem Sie den Installer gestartet haben. Bitte prüfen Sie auch, ob Ihre Internet-Verbindung funktioniert oder in irgendeiner Weise eingeschränkt ist. Überprüfen Sie zudem, ob Ihr Betriebssystem ein 64-Bit-Microsoft-Windows-System ist.
2. *Der Installer wird ohne Fehlermeldung beendet, aber die Seite "http://localhost:3000" zeigt nichts an.*Abhängig von Ihrem System kann es eine Zeitverschiebung von wenigen Minuten geben zwischen dem Ende der Installation und dem Start des Webservers. Warten Sie bitte bis zu fünf Minuten, nachdem der Installer sich abschaltet, und versuchen Sie es dann erneut. Wenn sich die Seite immer noch nicht laden lässt, ist der Webserver nicht eingeschaltet. Um ihn manuell zu starten, starten Sie bitte eine Eingabeaufforderung, navigieren Sie dann zum Installationsordner und führen Sie den Befehl "rails s -b [IP-Adresse]" aus, um den Webserver zu starten. Warten Sie zwei Minuten, um ihn zu starten und die Webpage dann neu zu prüfen. Die Webseite sollte nun ganz normal hochfahren – oder aber die Eingabeaufforderung wird eine Nachricht enthalten, die Ihnen die Ursache des Problems nennt.
3. *Die Seite "http://localhost:3000" wird gut angezeigt, die Seite "http://[IP-Adresse]:3000" jedoch nicht.*Die Software und der Oberflächenserver arbeiten einwandfrei. Das Problem besteht allerdings im Routing zur IP-Adresse. Setzen Sie sich bitte mit Ihrem IT-Netz-Administrator wegen der Öffnung des Port 3000 fürs HTML-Protokoll in Verbindung.
4. *Die Seite "http://[IP-address]:3000" wird am lokalen Computer gut angezeigt, aber nicht auf den anderen Computern.* Es gibt ein Problem mit der Netzverbindung und höchst wahrscheinlich mit der Firewall. Setzen Sie sich

bitte mit Ihrem IT-Netz-Administrator in Verbindung.

5. *Es besteht ein anderes Problem; oder eines der oben genannten Probleme bleibt bestehen.*Setzen Sie sich mit algorithmica technologies in Verbindung und bitten Sie um Hilfestellung. Geben Sie genaue und detaillierte Informationen über Ihr System, was Sie bisher getan haben und wie sich das Problem genau darstellt. Vielen Dank für Ihre Geduld !

2.6. Daten-Voraussetzungen

Jede Datenanalyse hängt von der Qualität der zur Verfügung gestellten Daten ab. Wir gehen in zunächst davon aus, dass alle wichtigen Aspekte des Prozesses oder der Maschinerie im Datensatz aufgenommen werden. Fehlen wichtige Daten, wird die Modellierung nicht so gut funktionieren wie wenn diese Daten bereitstehen. Auf der anderen Seite ist es auch nicht gut, das Modell mit einer Vielzahl von unwichtigen Messungen zu überfrachten. Die wichtigste Tätigkeit zur Modellierung ist eine vernünftige Auswahl der Tags, die für die Modellierung bereit stehen sollen. Wir können sagen, dass bei einer Industrieanlage mit einem komplexen Leitsystem weniger als 10% aller verfügbaren Messungen für die modellhafte Darstellung des Prozesses nötig und wichtig sind.

Falls Sie im Zweifel darüber sind, ob eine Messung berücksichtigt werden soll oder nicht, sollten Sie sich eher für (und nicht gegen) diese Messung entscheiden. Der Grund dafür ist derselbe, der auch für menschliches Lernen entscheidend ist: Wenn Sie irrelevante Informationen bereitstellen, mag das zeitaufwändig und irritierend sein, aber diese Informationen werden Sie nicht daran hindern, Zusammenhänge zu begreifen. Fehlen aber wichtige Informationen, so kann es sein, dass Sie kein ausreichendes Verständnis gewinnen. Im Zweifelsfall ist es also besser, Messdaten mit einzubeziehen als sie wegzulassen.

Die meisten Archivsysteme verfolgen die Regel, nach der sie den neuen Wert eines bestimmten Tags nur dann aufzeichnen, wenn es von dem letzten aufgezeichneten Wert um einen Mindestbetrag abweicht. Diesen Mindestbetrag bezeichnet man als den Kompressionsfaktor. Das bedeutet, dass manche Messungen häufig aufgezeichnet werden und andere nur selten. Für den Zweck des Aufzeichnens ist dies ein enorm platzsparender Mechanismus.

Für die Analyse und das maschinelle Lernen müssen wir die Daten allerdings zeitlich abgleichen. Für jeden Zeitstempel müssen wir den Wert jedes Tags zu dieser Zeit kennen. Die meisten Methoden des maschinellen Lernens erfordern es, dass der Zeitabstand zwischen den fortlaufenden Messungen immer derselbe ist.

Um von der üblichen historischen Datenaufzeichnung zu dieser Tabelle der abgeglichenen Daten zu kommen, verwenden wir die allgemeine Regel, dass der Wert eines Tags solange als derselbe angenommen wird, bis wir einen neuen Wert erhalten. Eine Zeitreihe kann dann wie eine Treppe aussehen. In den meisten Fällen führt diese Datenauflistung zu einem Wachstum des gesamten Datenvolumens, weil viele Werte oftmals erscheinen. Das wird man aber nicht verhindern können.

Aus diesem Grund wählen wir ein vernünftiges Datenintervall. Das heißt, dass die Zeitabstände zwischen den sukzessiven Ablesungen in dieser Tabelle nicht zu klein sein dürften, um nicht zu viele Daten anzusammeln, aber auch nicht zu groß, weil

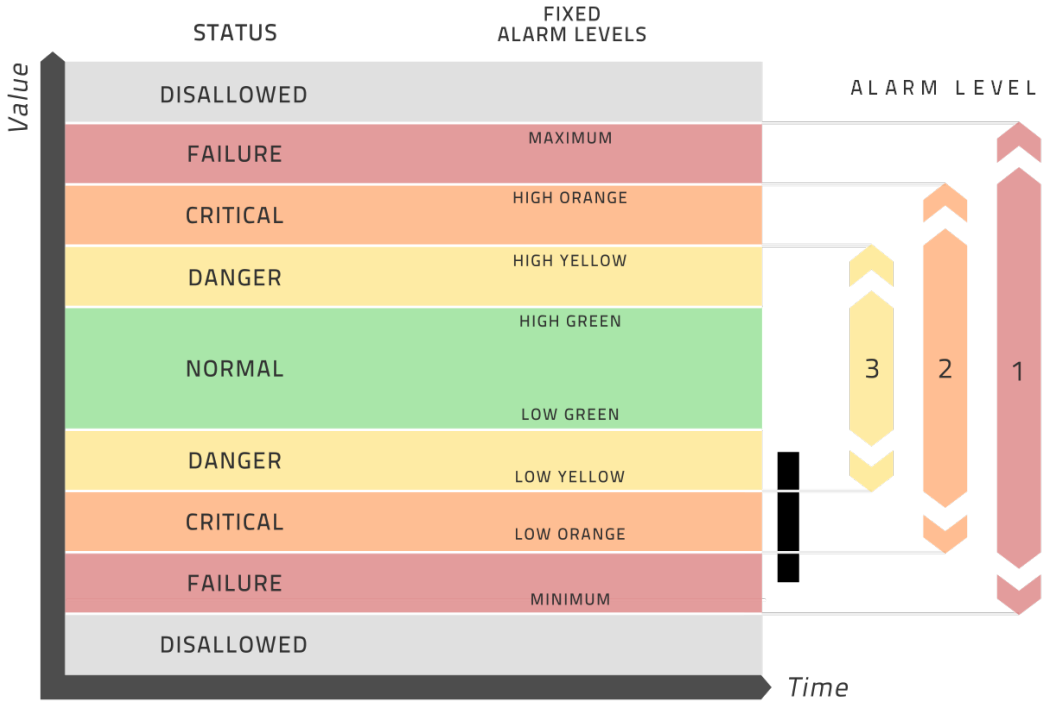
sonst der dynamische Prozess nicht sichtbar wird. Das Zeitintervall sollte deshalb mit Sorgfalt ausgewählt werden, möglichst basierend auf der Erfahrung der inhärenten Zeitskala des Prozesses, den man modellieren möchte.

Ein Datensatz muss einen Anfang und ein Ende haben. Um ein IHM-Modell zu trainieren, wählen wir sorgfältig eine Trainingsperiode aus, so dass wir für eine bestimmte Zeitspanne das reibungslose Verhalten der zu überprüfenden Maschine aufzeigen können. Sind wir davon überzeugt, dass externe Bedingungen, etwa die Jahreszeiten, für eine genaue Darstellung des Prozesses von entscheidender Bedeutung sind, so benötigen wir Daten für diese verschiedenen Bedingungen. Um den reibungslosen Zustand einer Maschine genau darzustellen, reicht in der Regel eine Zeitspanne von drei bis sechs Wochen aus. Diese Wochen müssen nicht unbedingt fortlaufend sein. Fallen saisonale Veränderungen ins Gewicht, wählen wir je eine Woche aus dem Frühling, dem Sommer, dem Herbst und dem Winter. Für ein Optimierungs-Modell empfehlen wir allerdings ein ganzes Jahr aufzuzeichnen, um die saisonalen Varianten ausreichend einzubeziehen.

Für die Zeitspannen, die wir für das Training des Modells auswählen, kann es aber vorübergehende Zustände geben, die wir nicht zugrunde legen sollten. Wenn eine Maschine beispielsweise jeden Abend abgeschaltet wird, wollen wir das Modell natürlich nicht auf diese Zeit ausrichten, während der die Maschine nicht funktionsbereit ist. Aus diesem Grund bietet die Software Ausschlussmöglichkeiten für bestimmte Messwerte an. Man könnte etwa bestimmen, dass alle Datenpunkte, bei denen die Rotationsrate einer Turbine weniger als 500 U/min beträgt, vom Modell ausgeklammert werden sollen. Die für solche Entscheidungen notwendigen Tags müssen dann in den Datensatz aufgenommen werden.

Nachdem diese Entscheidungen getroffen wurden, müssen die ausgewählten Tags dem Modell bekannt gemacht werden, indem gewisse Informationen bereitgestellt werden. Neben einigen verwaltungstechnischen Informationen wie Name und Werteinheit benötigen wir noch einige zusätzliche Fakten. Weil es zuweilen Messfehler gibt, müssen wir die erlaubte Bandbreite der Messungen kennen, damit sehr niedrige oder sehr hohe Messwerte als Ausreißer identifiziert und ausgeklammert werden können. Wir benötigen auch die Messunsicherheit, damit wir diese Unsicherheit in den Ergebnissen der Analyse (als Ergebnisunsicherheit) berücksichtigen können. Wenn Sie mehr darüber wissen möchten, schauen Sie im Abschnitt über den mathematischen Hintergrund nach. Für IHM müssen wir auch wissen, welche Messungen dynamische Begrenzungen erhalten sollen, die einen Alarm auslösen. In der Praxis werden die allermeisten Messungen des Modells keine Alarmsignale hervorrufen, sie liefern aber Informationen und den Kontext für diejenigen Messungen, die Alarme hervorrufen sollen. Für APO etwa müssen wir wissen, welche Tags direkt vom Anlagenfahrer kontrolliert werden können und welche nicht kontrolliert werden können.

Als Alternative könnten Sie für jede Messung bis zu drei statische Alarm-Spielräume spezifizieren. Das ist notwendig, damit IHM die traditionelle Zustandsüberwachung unterstützen kann – zusätzlich zum Ansatz der dynamischen Grenzwerte. Dieses System von Analyse und Alarm unterscheidet sich von der Modellanalyse und ist als völlig optional zu betrachten. Diese Bandbreiten werden am besten in der Grafik unten erklärt.

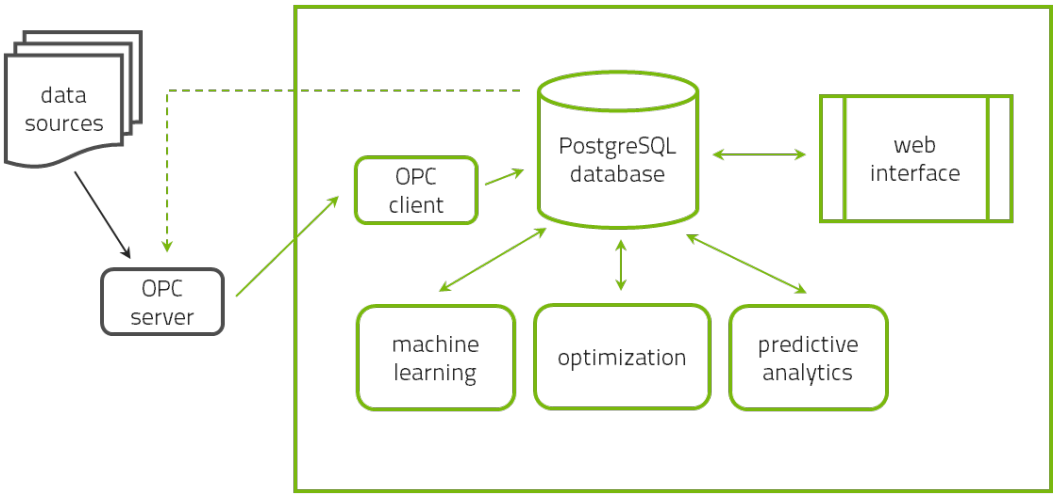


Summa Summarum: dies sind die Entscheidungen, die Sie zu Beginn im Hinblick auf Ihren Datensatz treffen müssen:

1. Welche Tags sollen aufgenommen werden?
2. Welcher Zeitabstand soll benutzt werden?
3. Über welchen Zeitraum soll der Datensatz laufen?
4. Über welchen Zeitraum soll das Training laufen?
5. Welche allgemeinen Zustände sollen ausgeklammert werden?
6. Welche Zustände sollen für den jeweiligen Tag ausgeklammert werden?
7. Für jeden Tag gilt es zu fragen:
    1. Welches soll die erlaubte Bandbreite der Messwerte sein?
    2. Wie unscharf ist die Messung?
    3. Soll dieser Tag modelliert werden und Alarme auslösen?
    4. Kann dieser Tag direkt vom Anlagenfahrer kontrolliert werden?
    5. Als Option: Welches sind die Bandbreiten für die statischen Alarmmeldungen bei gelben, orange und roten Alarmsignalen?

2.7. Software-Architektur

Die Software hat drei Hauptkomponenten. Die zentrale Datenbank, in der alle Daten und Modellinformationen gespeichert werden, ist eine PostgreSQL-Datenbank. Die Daten werden von einem Kommandozeilenprogramm analysiert, das Daten aus der Datenbank ausliest und die Ergebnisse der Analyse wiederum in die Datenbank speichert. Der Nutzer interagiert mit diesem System durch eine Browserbasierte graphische Oberfläche auf der Basis von Ruby-on-Rails.



Alle drei Komponenten werden typischerweise auf einem einzelnen Computer auf dem Gelände der Institution des Nutzers installiert. Alle Nutzer können mittels irgendeines Webbrowsers auf die Oberfläche zugreifen, und zwar von jedem Gerät aus, das mit dem Intranet der Institution verbunden ist, vorausgesetzt, die entsprechende Firewall erlaubt diesen Zugang. Weil die Software auf dem Gelände installiert wird, stellt dies keinen Cloud-Service dar, und die Daten bleiben vor Ort. Es gibt also kein Sicherheitsrisiko, sei es durch den Verlust firmeneigener Daten oder durch Hacker-Angriffe.

Die drei Komponenten können auch auf unterschiedlichen Servern installiert werden, falls der Nutzer den Datenbank-Server zwecks Speicherplatz und den zur Berechnung notwendigen Server zwecks Rechnergeschwindigkeit optimieren möchte, aber notwendig ist das nicht.

Anfangs wird davon ausgegangen, dass die historischen Daten in Gestalt einer Datei zur Verfügung gestellt werden. Der Grund dafür ist, dass ein Datenexport mit anschließendem Datenimport erfahrungsgemäß effizienter ist als eine direkte Datenverbindung etwa mittels OPC-HDA.

Das normale Lesen von Echtzeitdaten sollte aber mittels OPC-DA erfolgen. Zu diesem Zweck muss der OPC-Server spezifiziert werden (ein Beispiel ist opcda://Softing.OPCToolboxDemo_ServerDA.1/{2E565242-B238-11D3-842D-0008C779D775}), und eine zwischengeschaltete Firewall muss entsprechend geöffnet werden. Auch muss der volle Name jeder Tag auf dem OPC-Server spezifiziert werden, um die Daten lesen zu können.

3. How-To

Dieses Kapitel beschreibt verschiedene praktische Aufgaben, die jeweils unabhängig voneinander durchzuführen sind:

1. *Die Anlage definieren.*Um ein Modell zu erschaffen, muss es eine Anlage geben bzw. diese muss definiert werden.
2. *Vorbereitung der Daten.*Die wichtigste Aufgabe bei diesen modernen Analyse-Methoden ist es, die zu analysierenden Daten vorzubereiten und zu beschreiben.
3. *OPC-Verbindung.*Um Daten in Echtzeit zu analysieren, müssen diese durch OPC

mit einer Datenquelle verbunden sein.

4. *Online stellen.*Für Echtzeit-Analysen muss die Anwendung online gestellt werden.

3.1. Anlage definieren

Die Anlage ist die Einheit, die Ihren Datensatz definiert. Um ein Modell zu konstruieren, muss dafür als Erstes eine Anlage geschaffen werden.

Um diese Anlage benennen zu können, stehen Ihnen vier Textfelder zur Verfügung: Unternehmen, Einheit, Anlage und Equipment. Das sind lediglich Etiketten, mit deren Hilfe Sie mehrere Modelle in einer Oberfäche organisieren und Nutzerrechte verwalten können, um individuelle Modelle anzuschauen. Sie können auch ein Feld für Kommentare nutzen, um dort ausführlichere Erklärungen über Sinn und Zweck des jeweiligen Modells zu vermerken. Während einer Test- oder Tuningphase möchten Sie vielleicht mehrere Modelle konstruieren, um sie miteinander vergleichen zu können. Das Kommentarfeld ist auch nützlich, um im Auge zu behalten, welches Modell wie aufgebaut wurde.

Der kundenspezifische Code ist eine fünfstellige Zahl, die algorithmica technologies Ihnen zur Verfügung stellt, wenn Sie die Software-Lizenz für diese Anlage erworben haben; ihr Zweck ist es, jeden kundenspezifischen Schritt zu identifizieren, der für Sie geschrieben wurde und der in der Analyse-Software enthalten ist. Wenn Sie die Software testen, haben Sie diesen Code allerdings (noch) nicht, so dass Sie dieses Feld leer lassen können.

Die maximale Zeitverzögerung ist nur bei Nutzung des IHM (Intelligent Health Monitor) relevant. Die Auswahl unabhängiger Variablen, die der IHM verwendet, kann automatisch vorgenommen werden und er kann optional eine Zeitverzögerung beinhalten. Diese Option spezifiziert die maximale Zeitverzögerung in Form von so und so vielen Zeitstufen. Bitte beachten Sie die Beschreibung des IHM-Modells zwecks weiterer Details.

Die drei Felder "OPC-Name", "OPC-DA-Version" und "Subscriptions-Frequenz" sind für die OPC-Verbindung zu einer Datenquelle relevant. Relevant ist dies allerdings nur, wenn die Software in Echtzeit verwendet wird. Wenn Sie mehr Informationen benötigen, um das System online zu stellen, konsultieren Sie bitte den "How-to"-Abschnitt.

Die drei Zeitperioden, die Anfang und Ende der Referenz-Laufzeit und den Start der Betriebslaufzeit definieren, sind Konzepte, die vom APO benutzt werden, um den Erfolg der Vorschläge zu beurteilen. Die Referenz-Laufzeit ist der Zeitrahmen für die Daten, die gebraucht werden, um das Modell aufzubauen; und die Betriebslaufzeit beginnt, wenn die vom APO gemachten Vorschläge tatsächlich umgesetzt werden.

3.2. Vorbereitung Ihrer Daten

Beim maschinellen Lernen werden historische Daten analysiert, um gewisse Muster zu erkennen und um diese Muster in ein mathematisches Modell zu konvertieren. Damit das funktionieren kann, bedarf es nicht nur der historischen Daten, sondern auch einiger zusätzlicher Informationen über diese Daten. In diesem Abschnitt wird

beschrieben, wie man das alles vorbereiten soll, damit sich dieser Lerneffekt ergibt.

Die Situation, die wir modellieren wollen, wird im Allgemeinen viele Eigenschaften haben, die von Sensoren gemessen werden und die wir für unser Modell verwenden können. Der erste Schritt wäre, die Datenquellen auszuwählen, die für die Analyse relevant erscheinen. Der zweite Schritt wäre, jede dieser Datenquellen mit relevanten Eigenschaften zu versehen – etwa die erlaubte Bandbreite der Messwerte. Der dritte und letzte Schritt wäre es, eine Tabelle mit historischen Messwerten für jede der Datenquellen zu erstellen.

Sind alle diese Daten erst einmal aufgelistet, kann die Datenanalyse und das (maschinelle) Lernen beginnen.

3.2.1. Tags auswählen

Um Ihre Modell-basierte Lernerfahrung zu starten, wählen Sie die Tags aus, die für Ihr Modell berücksichtigt werden sollen. An Ihrer Anlage wurden zahlreiche Sensoren installiert, die eine Vielzahl von Eigenschaften messen. Viele davon werden für das Modellieren jedoch nicht relevant sein. Um Ihnen die Auswahl zu erleichtern, sollten Sie die folgenden Richtlinien beachten:

1. Tags, die die Anlagenfahrer regelmäßig modifizieren, sollten für das Modell berücksichtigt werden.
2. Tags, die wichtige Randbedingungen für den Prozess repräsentieren, sollten ebenfalls in das Modell einbezogen werden.
3. Tags, die für Sicherheit, Qualität, Effizienz und Geldwert relevant sind, sollten auch ins Modell einfließen.
4. Tags, die nur für das An- und Abfahren der Anlage oder für Notfälle verwendet werden, dürften für den normalen Betrieb irrelevant sein und müssen nicht berücksichtigt werden.
5. Tags, die für Condition Monitoring oder ähnliche Zwecke verwendet werden, dürften für Optimierungsmodelle auch nicht relevant sein.

Nach unserer Faustregel dürften bis zu 90% der Tags fürs Modellieren irrelevant sein. In den meisten Fällen erfordert ein Prozess-Modell nur ein paar hundert Tags, obwohl viele tausend Tags verfügbar sind.

Wenn Sie im Zweifel darüber sind, ob Sie ein Tag fürs Modellieren benötigen, berücksichtigen Sie es lieber. Zu viele Daten sind besser als zu wenige Daten. Allerdings: Je mehr Tags einbezogen werden, umso schwerfälliger und zeitaufwändiger wird das Modellieren und deshalb ist es in Ihrem Interesse, die Zahl der Tags relativ niedrig zu halten.

3.2.2. Metadaten vorbereiten

Wenn Sie die Tags ausgesucht haben, die Sie einbeziehen wollen, müssen Sie als nächstes einige Informationen über jeden Tag angeben. Die meisten davon sind relativ unkompliziert.

Es gibt zwei Arten, wie Sie diese Daten vorbereiten können:

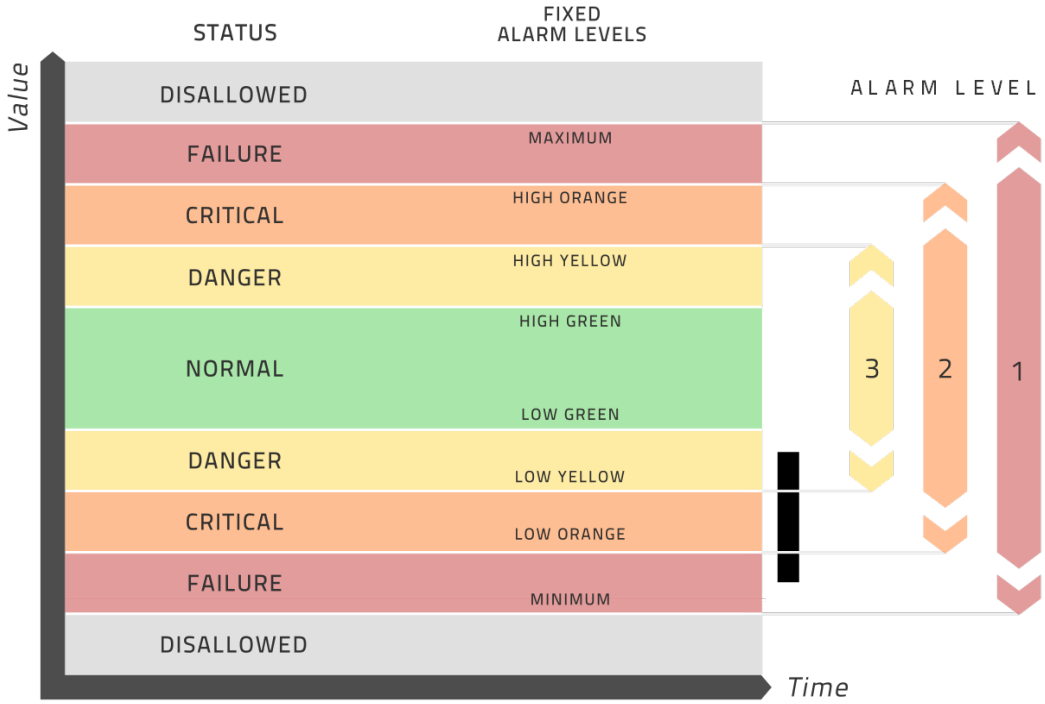
1. Sie können die Taginformation direkt mittels der Oberfläche angeben oder editieren, indem Sie auf `Anlage -> Tags editieren` klicken. Dieses Formular

erlaubt es Ihnen, neue Tags hinzuzufügen oder die Informationen jeden Tag zu editieren. Vergessen Sie nicht, Ihre Arbeit abzuspeichern! Das ist die bequemste Art, diese Informationen zu definieren. Beachten Sie bitte, dass die Tabelle die Funktion kopieren/einfügen unterstützt, aber seien Sie vorsichtig, wenn Sie kopieren/einfügen für große Informationsmengen verwenden, weil es dann Probleme mit der Ausrichtung geben kann.

2. Sie können eine Datei vorbereiten, in der die Informationen enthalten sind und die Sie dann in einem Zug in die Datenbank hochladen können. Diese Datei wird eine Textdatei sein, die 16 abgegrenzte TAB-Spalten enthält. Werden nur normale ASCII-Zeichen verwendet, kann man diese Datei als ASCII sichern, aber wenn non-ASCII-Zeichen nötig sind (etwa die deutschen Umlaute ä, ö, ü oder Symbole wie ° or µ), dann sollte man die Datei im UTF8 Format abspeichern. Diese Datei kann eine Kopfzeilen-Reihe mit Namen der jeweiligen Spalten enthalten.

Hier folgt nun eine kurze Erläuterung für jede Spalte und ob sie überhaupt nötig ist

table class="manualtable">SpaltennameErfordernisBeschreibungTagerforderlichDer eindeutige Identifikator für eine Zeitreihe. Das ist oft eine alphanumerische Zeichenkette, die vom Leitsystem oder Archivsystem verwendet wird, um den Tag zu kennzeichnen.PLS TagerforderlichDies ist oft dasselbe wie die Spalte "Tag" und stellt den eindeutigen Identifikator dar, der vom OPC-Server verwendet wird. algorithmica verwendet diese Spalte, um in der OPC-Datenquelle nach dem aktuellen Wert diesen Tag zu fragen. Deshalb muss es alle OPC-Informationen enthalten.NameempfohlenEine kurze Beschreibung, worum es bei diesem Tag geht.BeschreibungoptionalEine längere Beschreibung, worum es bei diesem Tag geht.EinheiterforderlichDie physikalischen Einheiten des Tags, z.B. "°C"MinimumerforderlichDer kleinste erlaubte Wert. Alle Werte, die niedriger als dieser sind, werden als physikalisch nicht möglich ignoriert. Für eine ausführlichere Diskussion zum Konzept von Minimum und Maximum siehe bitte den entsprechenden Abschnitt im Kapitel über Terminologie.MaximumerforderlichDer größte erlaubte Wert. Alle Werte größer als dieser werden als physikalisch nicht möglich ignoriert.KontrollierbarkeiterforderlichBestimmt, ob dieser Tag direkt vom Anlagenfahrer kontrolliert, überhaupt nicht kontrolliert oder nur indirekt kontrolliert werden kann.niedriges grünoptionalInnerhalb der Bandbreite von erlaubten Werten [minimum, maximum] können wir drei Gruppen von fest fixierten Alarmstufen (von innen nach außen) definieren: grün, gelb und orange. Bitte sehen Sie sich das Bild unten zum besseren Verständnis an.hohes grünoptionalniedriges gelboptionalhohes gelboptionalniedriges orangeoptionalhohes orangeoptionalDeltaerforderlichDie Messunsicherheit dieses Tags wird in denselben Einheiten dargestellt wie der Wert selbst. Wenn Sie dies beim ersten Mal einrichten, lesen Sie bitte die ausführlicheren Erläuterungen dieses Konzepts im Kapitel über Terminologie.GrenzwerterforderlichDiese Spalte ist entweder "TRUE" oder "FALSE" abhängig davon, ob dieser Tag durch IHM einen dynamischen Grenzwert erhält oder nicht.



Diese Abbildung zeigt die Bedeutung der höheren und niedrigeren Farbbegrenzungen sowie der minimalen und maximalen Werte. Jeder Messwert außerhalb der Bandbreite zwischen minimal und maximal wird als ein unmöglicher Wert betrachtet. Jeder Messwert innerhalb dieser Bandbreite hat eine Farbe gemäß dem angezeigten Schema. Ein normaler Betrieb sollte im grünen Bereich ablaufen.

Spalten, die nicht nötig sind, können leer bleiben. Vor allem die Optimierung durch APO berücksichtigt nicht die farbigen Begrenzungen. Ein Beispiel dafür befindet sich in den Installationsdateien.

3.2.3. Vorbereitung der Datenverarbeitung

Die Datendatei ist eine ASCII-Datei. Die erste Spalte ist ein Zeitstempel. Die weiteren Spalten enthalten Gleitkommazahlen, eine für jeden Tag des Modells. Die Datei ist Semikolon-begrenzt. Die Datei kann eine Kopfzeile mit den Spaltennamen enthalten. Es ist von essentieller Bedeutung, dass die Anordnung der Spalten in der Datendatei dieselbe ist wie die Reihenanordnung der Tagdatei.

Der Zeitstempel kann in zwei Formaten erscheinen:

1. Die übliche europäische Zeitdarstellung: dd.mm.yyyy hh:MM:ss.xxx
2. Das ISO 8601-Format: yyyy-mm-dd hh:MM:ss.xxx

In beiden Fällen ist der Gebrauch von Millisekunden ‚.xxx' optional.

Hier ist eine beispielhafte Datei, die Semikolon-begrenzt ist.

Zeit	CHA01CE011	LAE11CP010	LAE11CT010	LBA10CF001
01.01.2015 00:00:00	1.1	2.1	3.1	4.1
01.01.2015	1.2	2.2	3.2	4.2

01:00:00				
01.01.2015 02:00:00	1.3	2.3	3.3	4.3
01.01.2015 03:00:00	1.4	2.4	3.4	4.4

Eine beispielhafte Datei ist in den Installationsdateien enthalten.

3.3. Einrichten der OPC-Verbindung

Um Ihr Modell in Echtzeit zu aktualisieren, benötigt das Modell Zugang zu den laufenden Daten mittels einer OPC Verbindung. Wir gehen davon aus, dass Ihr Leitsystem, Ihr Archivsystem oder eine andere Datenquelle Ihnen den Zugang zu einem OPC-Server ermöglicht. Algorithmica bietet Ihnen einen OPC Client an, damit Sie diese Daten lesen können.

Die grundlegenden Daten einer Anlage (klicken Sie auf `Anlage -> editieren`) beinhalten drei Felder, die für die OPC-Verbindung relevant sind:

1. *OPC Name.* Das ist der volle Name des OPC-Servers in Ihrem Netz. Ein Beispielname ist opcda:///Softing.OPCToolboxDemo_ServerDA.1/{2E565242-B238-11D3-842D-0008C779D775}.
2. *OPC DA version.* OPC beinhaltet mehrere Protokolltypen. Algorithmicas Anwendungen benutzen nur den DA-Typ, der den Datenzugang ermöglicht. Er wird dazu verwendet, den aktuellsten Wert eines einzelnen Tags zu erfragen. Dieses Protokoll ist verfügbar in den Versionen 2 oder 3. Bitte spezifizieren Sie, welche Version Ihr OPC-Server unterstützt.
3. *Frequenz des Abonnement.* Dies ist eine Sekundenzahl, die anzeigt, um wie viele Sekunden wir warten sollten, bevor wir einen neuen Wert erfragen. Bitte wählen Sie diese Zahl sorgfältig aus. Denn die Anwendung wird jeden Tag in dieser von Ihnen ausgewählten Häufigkeit erfragen, die erfragten Werte in der Datenbank abspeichern und die Modellberechnungen entsprechend durchführen. Eine niedrige Zahl könnte nicht nur die Rechnerkapazitäten Ihres Netzes überfordern, sondern wahrscheinlich auch keinen praktischen Nutzen erzielen, weil die menschliche Reaktionszeit nicht ausreichend ist, damit umzugehen. Wenn Sie allerdings eine zu große Zahl wählen, kann dies einen erheblichen Zeitverzug nach sich ziehen zwischen dem modellierten physikalischen Effekt und einer möglichen Reaktion darauf. Wenn Sie im Zweifel darüber sind, welche Sekundenzahl Sie eingehen sollen, empfehlen wir eine Zeit zwischen 300 und 900 Sekunden, also 5 bis 15 Minuten. Idealerweise sollte dieser Wert derselbe sein wie die Datenkadenz, die für den historischen Datensatz ausgewählt wurde.

Wenn Sie es noch nicht gemacht haben, geben Sie bitte diese Werte in die Oberfläche ein und speichern Sie sie. Wir gehen davon aus, dass Sie jeden Tag mit dem Feld "PLS Tag" ausgestattet haben, das ist der vollständige OPC Objektname dieses Tags. Die Kombination der Information des OPC-Servers und des OPC-Objektes (OPC item) eines jeden Elementes (tag) ermöglicht es der Anwendung, die Werte auszulesen.

Beachten Sie bitte, dass der Anwendungscomputer in der Lage sein muss, über das

Netz Zugriff auf den OPC-Server zu erhalten. Das könnte es erfordern, an der Firewall Ihres Netzes Veränderungen vorzunehmen.

Um festzustellen, ob der OPC-Server erreicht werden kann und die Objekte sich korrekt lesen lassen, gehen Sie bitte zu `Anlage -> OPC Diagnostik`. Dieses Formular sollte Ihnen einige grundlegende Informationen über den OPC-Server und den aktuellen Wert eines jeden Tags anzeigen. Finden Sie ein Häkchen neben dem Serverstatus und neben jedem Tag, ist die Verbindung gut und die Objektnamen wurden alle gefunden. Befindet sich neben dem Serverstatus jedoch ein X, ist entweder der Servername falsch oder er kann über das Netz nicht erreicht werden. Konsultieren Sie in diesem Fall Ihren Netz-Administrator. Befindet sich ein Häkchen neben dem Serverstatus, aber ein X neben einigen Tags, so können die entsprechenden Objekte nicht auf dem OPC-Server gefunden werden. Bitte überprüfen Sie dann die Objektnamen und korrigieren Sie sie, indem Sie zu `plant -> edit tags`

wechseln. Auch wenn Sie alle diese Informationen erfolgreich abgeliefert bzw. deren Richtigkeit und Funktionsfähigkeit überprüft haben, wird die Echtzeit-Berechnung noch nicht sofort einsetzen. Denn: Sie können die Echtzeit-Berechnung jederzeit an- und ausschalten auch unabhängig von den für deren Verwendung notwendigen Informationen.

3.4. Die Anwendung online/offline setzen

Um festzustellen, ob Ihre Anwendung aktuell in Echtzeit läuft oder nicht, gehen Sie bitte auf `Anlage -> OPC an/aus`. Der dort angezeigte Status wird Sie darüber informieren, ob der Prozess gegenwärtig läuft, wann die letzten Daten abgelesen wurden und auf welchen Rhythmus die Ablesung eingestellt ist.

Aufgrund von Netzverzögerungen und zeitweiligen Kommunikationsunterbrechungen im Computernetz gehen wir von folgender Definition aus, um zu entscheiden, ob die OPC-Datenverbindung online ist oder nicht: Wenn seit der letzten Datenablesung nicht mehr Zeit verstrichen ist als zwei Ableseintervalle, betrachten wir die Verbindung als online. Ist die letzte Ablesung hingegen länger her als zwei Ableseintervalle, so betrachten wir die Verbindung als offline.

Je nachdem, welcher Status vorliegt, können Sie mit Hilfe eines Buttons den Echtzeit-Prozess an- oder ausschalten. Haben Sie diesen Button angeklickt, sollten Sie ein wenig warten, bis die Verbindung hergestellt oder getrennt ist. Das kann – abhängig von Ihrem Netz – ein paar Minuten dauern. Wenn Sie diese Seite aktualisieren, können Sie anschließend den Verbindungsstatus erkennen.

Diese Seite bietet Ihnen auch die Option, die OPC Logdatei zu konsultieren. Dieser zeichnet wichtige Vorgänge Ihrer OPC-Verbindung auf. Es gibt aber in der Regel keinen Grund, diese Datei zu konsultieren, es sei denn, es besteht ein Problem mit der Verbindung.

4. Advanced Process Optimizer (APO)

Eine Anlage zu betreiben ist eine komplexe Aufgabe für die Anlagenfahrer, die alle

paar Minuten detaillierte Entscheidungen zu treffen haben, wie sie die Sollwerte der Anlage einstellen müssen, um auf gewisse externe Faktoren zu reagieren. Die Anlage soll ja stets die Erwartungen erfüllen, welche die Kunden verlangen, und sie muss auf Wetterveränderungen ebenso reagieren wie auf die Qualität des Rohmaterials, um nur zwei Faktoren zu nennen.

Anlagen werden in unterschiedlichen Schichten gefahren. Ein häufig zu beobachtendes Phänomen ist, dass nach einem Schichtwechsel das neue Schichtteam es besser zu machen glaubt als das alte. Entsprechend werden einige Sollwerte neu eingestellt. Danach benötigt die Anlage – allein schon wegen ihrer schieren Größe – erhebliche Zeit, bis das Equilibrium wiederhergestellt ist. Nach acht Stunden, wenn die nächste Schicht übernimmt, wiederholt sich der Vorgang. Daraus folgt, dass die Anlage nur selten wirklich optimal läuft.

Das gilt nicht nur aufgrund der unterschiedlichen Überzeugungen der verschiedenen Schichtteams. Es hat auch zu tun mit dem Überangebot an Informationen, da eine Anlage zigtausende von Sensoren haben kann, die menschliche Anlagenfahrer unmöglich alle im Auge behalten können. Deshalb muss jeder Anlagenfahrer aufgrund von Ausbildung und Erfahrung darüber entscheiden, welche wenigen Sensoren er für seine Entscheidung heranzieht. Diese wenigen Sensoren bieten ihm zwar zahlreiche Informationen, aber keineswegs alle.

Automatisierungen und Leitsysteme beschränken sich in der Regel auf lokal eng begrenzte Bereiche und sind eher darauf angelegt, von unten nach oben zu funktionieren. Sie funktionieren am besten bei in sich abgeschlossenen Systemen wie bei einer Turbine, einem Hochofen, einem Boiler u.ä. Eine übergreifende Methodologie für solche Systeme einzurichten ist schwierig und wird selten in Betracht gezogen. Es ist aber gerade beim Zusammenspiel zwischen den verschiedenen Komponenten einer Anlage, dass das Optimierungspotenzial noch weitgehend ungenutzt bleibt.

Aus diesem Grund sollten wir die ganze Anlage als ein einzelnes komplexes System betrachten. Sie ist ja ein physikalischer Apparat, der den Naturgesetzen gehorcht. Aus diesem Grund kann eine Anlage mit Hilfe von Differentialgleichungen beschrieben werden. Diese sind zwar sehr kompliziert, aber existent. Das Gleichungssystem, mit dessen Hilfe die Anlage beschrieben wird, nennen wird das Modell der Anlage. Weil dieses Modell die Anlage in ihren wichtigsten Eigenschaften repräsentiert, nennt man es oft den digitalen Zwilling der Anlage.

Es gibt zwei grundlegende Arten, ein solches Modell zu erstellen. Bei der ersten Methode geht man Stück-für-Stück vor und erstellt einfache Modelle von Pumpen, Kompressoren und anderen Systemen und fügt diese Modelle zu einem größeren Modell zusammen. Diesen Ansatz nennen wir den First-Principles-Approach. Es erfordert allerdings viel Zeit und Anstrengung seitens der Fachingenieure, um dieses Modell anfänglich zu erstellen und dann über die ganze Laufzeit der Anlage aktuell zu halten.

Die zweite Möglichkeit basiert auf der Basis der vom Leitsystem erhobenen Prozessdaten, die genutzt werden, um aus ihnen ein empirisch-basiertes Modell zu erstellen. Diese Methode kann automatisch durch einen Computer erfolgen, und zwar ohne auf menschliche Expertise angewiesen zu sein. Dieses Modell kann

schnell erstellt werden und ist zudem in der Lage, sich selbst jederzeit auf den neuesten Stand zu bringen. Diesen Ansatz nennen wir maschinelles Lernen

4.1. Prozessoptimierung

Zweck des maschinellen Lernens ist es, von einer Anlage eine mathematische Darstellung zu entwickeln, bei der die Messdaten des Prozessleitsystems genutzt werden. Diese Darstellung muss notwendigerweise den extrem wichtigen Zeitfaktor berücksichtigen, denn die Anlage enthält komplexe Ursache-Wirkungs-Verhältnisse, die von einem Modell dargestellt werden müssen. Und diese vollziehen sich über unterschiedliche Zeitskalen. Das heißt, dass die Zeit zwischen Ursache und Wirkung manchmal Sekunden dauert, manchmal Stunden und manchmal Tage. Die Modellierungseffekte der verschiedenen Zeitskalen stellen eine komplizierte Eigenschaft der Daten dar.

Das Modell sollte so gestaltet sein, dass wir den Zustand der Anlage zu einem zukünftigen Zeitpunkt aufgrund des Zustandes der Anlage in der Vergangenheit und in der Gegenwart berechnen können. Diese Art von Modell kann dann zyklisch betrieben werden, so dass wir den Zustand für jeden Zeitpunkt in der Zukunft berechnen können.

Der Zustand der Anlage besteht aus den gesammelten Daten der Tags, die für den Betrieb der Anlage wichtig sind. Typischerweise gibt es mehrere tausend Tags in und an der Anlage, die wichtig sind. Diese Tags fallen in drei Kategorien. Da sind zum einen die Randbedingungen. Das sind die Tags, über welche die Anlagenfahrer keine Kontrolle haben. Beispiele dafür sind das Wetter oder die Qualität der Rohmaterialien. Zum andern haben wir die Sollwerte. Das sind die Tags, die von den Anlagenfahrer im Leitsystem direkt eingestellt werden können und die die Funktionsfähigkeit der Anlage bestimmen. Drittens haben wir die Istwerte. Das sind alle anderen Messungen, die von den Bedienern beeinflusst werden können (da sie keine Randbedingungen sind), wenngleich nicht direkt (es sind keine Sollwerte), da sie sich aufgrund der Interdependenz der Anlage selbst verändern, je nachdem, wie sich die äußeren Randbedingungen und Sollwerte mit der Zeit verändern. Ein typisches Beispiel hierfür ist eine Schwingungsmessung. – Alle diese Daten werden vom maschinellen Lernen dazu verwendet, vom gesamten Anlagenprozess ein Modell zu erstellen.

Ist das Modell erst einmal erstellt, wollen wir es freilich dazu benutzen, die Leistung der Anlage zu optimieren. Hierfür benötigen wir eine genaue Definition von "Leistung". Das könnte irgendein numerisches Konzept sein. Manchmal ist es eine physikalische Größe wie die Abgabe von Schadstoffen (z.B. NOX, SOX) oder eine Ingenieursquantität wie der Gesamtwirkungsgrad der ganzen Anlage, oder eine geschäftliche Quantität wie die Wirtschaftlichkeit. Wir können diese Leistungswerte auf der Grundlage des Zustands der Anlage für jeden möglichen Zeitpunkt berechnen.

Nun haben wir eine genau definierte Optimierungsaufgabe: Finde diejenigen Sollwerte heraus, die eine maximale Leistung ermöglichen, wobei zu berücksichtigen ist, dass die Randbedingungen so sind, wie sie sind. Zusätzlich zu den natürlichen Randbedingungen (wir können das Wetter nicht ändern) könnte es noch weitere Randbedingungen geben, die etwa mit Sicherheitsvorschriften oder anderen

Prozessbegrenzungen zu tun haben.

Das ist ein komplexes, in höchstem Maße nicht-lineares, multidimensionales und vielfältigen Bedingungen unterworfenes Optimierungsproblem, das wir mit Hilfe des Simulated Annealing lösen. Das Wesen dieser Aufgabe erfordert eine sogenannte heuristische Optimierungsmethode, da eine exakte Lösung der Aufgabe viel zu komplex wäre. Das sogenannte Simulated Annealing enthält einige einzigartige Eigenschaften. Es konvergiert zum globalen Optimum und kann auch in begrenzter Zeit eine sinnvolle Lösung anbieten.

Bei der realpraktischen Vorgehensweise messen wir in regelmäßigen Abständen (etwa einmal pro Minute) den Zustand der Anlage, indem wir die Messwerte vom OPC-Server abrufen, dann das Modell entsprechend aktualisieren, den optimalen Messwert herausfinden und schließlich einen Bericht darüber erstellen, welche Sollwerte neu eingestellt werden müssen. Das kann als offene Schleife (open-loop) angeboten werden, damit die Anlagenfahrer die empfohlene Aktion von Hand eingeben, oder besser noch: als geschlossene Schleife (closed-loop), wobei die Justierung direkt vom Leitsystem vorgenommen wird. Sobald eine Veränderung einsetzt, etwa das Wetter, werden die notwendigen Justierungen berechnet und entsprechend berichtigt. Auf diese Weise kann die Anlage jederzeit mit den für eine optimale Leistung notwendigen Sollwerten gefahren werden.

4.2. Korrektur- und Validierungsregeln

Wenn der Algorithmus des maschinellen Lernens das Modell erlernt, entnimmt er die historischen Daten Punkt für Punkt aus der Datenbank. Jeder "Punkt" wird zuerst korrigiert und anschließend validiert. Nur validierte Punkte werden vom maschinellen Lernen erfasst, während invalide Punkte ignoriert werden.

Nehmen Sie beispielsweise ein Ventil bzw. der Tag, welcher misst, wie weit das Ventil offen steht. Ein ganz geöffnetes Ventil verbucht eine 100-prozentige Öffnung, und ein ganz geschlossenes Ventil verbucht eine 0-prozentige Öffnung. Demzufolge betragen die Minimal- und Maximalwerte für dieses Tag 0 und 100. Allerdings gibt uns die Datenbank – aus verschiedenen Gründen – Ergebniswerte, die zuweilen etwas unter 0 oder etwas über 100 liegen. Das sind keineswegs Messfehler oder Fehlpunkte. Wir sollten diese Daten darum nicht aussortieren, sondern korrigieren. Das heißt: Ein Messwert unter 0 ist eigentlich 0, und ein Messwert über 100 ist tatsächlich nur 100. Das wissen wir, weil wir die Funktionsweise des Ventils kennen, weshalb wir solche Werte korrigieren. Korrekturregeln sind Regeln, die den Zahlenwert eines Tags zu einem festen Wert korrigieren, sofern der aufgezeichnete Wert unter oder über einem bestimmten Schwellenwert liegt. Standardmäßig sind keine Korrekturen an den Daten programmiert. Sind Korrekturen also nötig, müssen sie manuell vorgenommen werden.

Der Algorithmus des maschinellen Lernens sollte nur solche Betriebsbedingungen lernen, die vernünftig sind, und zwar in dem Sinn, dass sie dem Modell gestatten, die Einstellungen so zu programmieren, dass diese Bedingungen der Anlage auch zu einem zukünftigen Zeitpunkt erreicht werden. Jeder Ausnahmezustand, in der sich die Anlage zu gewissen Zeiten befunden haben mag und den wir nicht für wünschenswert halten, sollte darum vom maschinellen Lernen ausgespart bleiben.

Auch Zustände, bei denen die Anlage offline ist, nicht produktiv ist, gewartet wird oder bei denen andere abnormale Bedingungen vorliegen, sollten als nicht valide markiert werden. Eine Validierungsregel erfordert es, dass ein bestimmter Tag sich innerhalb zu spezifizierender Grenzwerte befinden muss. Standardmäßig muss jeder Tag einen Wert zwischen dem minimalen und maximalen Wert ergeben, um valide zu sein.

Zusätzlich zu den standardmäßigen Validierungsregeln können wir von Hand noch weitere hinzufügen. Diese zusätzlichen Regeln müssen nicht immer gelten, und wir können spezifizieren, dass eine Regel nur dann anzuwenden ist, wenn ein Tag sich oberhalb oder unterhalb eines bestimmten Wertes befindet. Diese Geltungsfunktion erlaubt es Ihnen, auch komplexe Regeln einzugeben, die jeweils unter bestimmten Bedingungen gelten. Zum Beispiel könnten Sie eine Vorgabe machen, nach der ein Prozess bei Volllast eine hohe Temperatur haben und bei Halblast nur eine niedrige Temperatur haben soll.

4.3. Zielfunktion

Zweck der Prozessoptimierung ist es, Sie darüber in Kenntnis zu setzen, welche Sollwerte Sie modifizieren müssen, damit Ihre Anlage den besten Betriebspunkt erreicht. Um das zu ermöglichen, müssen Sie definieren, was Sie mit besten meinen. Jede in Zahlen ausgedrückte Quantität kann dafür in Frage kommen, z.B. Effizienz, Ertrag, Gewinn usw.

APO wird dann die Zielquantität jeweils maximieren, weshalb es wichtig ist, wie Sie Ihr Ziel formulieren. Wenn Sie allerdings möchten, dass APO eine bestimmte Quantität minimieren soll, z.B. Kosten, brauchen Sie nur ein Minus-Zeichen vor die Zahl setzen. In dem Falle wandelt sich die Maximierung in eine Minimierung um.

Alle Informationen, die benötigt werden, um die Zielfunktion zu evaluieren, müssen auf der Datenbank abrufbar ein. Nehmen wir als Beispiel den Umsatz durch den Verkauf von Strom. Auf der einen Seite benötigen wir einen Tag, das die produzierte Strommenge bemisst. Das dürfte in der Regel ohnehin Teil des Prozess-Modells sein. Wir benötigen aber auch den finanziellen Wert einer verkauften Stromeinheit. Dieser Wert ist in der Regel kein Teil des Prozess-Modells. Um ihn aber als Teil der Zielfunktion einzubeziehen, müssen Sie diesen Tag auch in die Datenbank einbeziehen. Beachten Sie, dass finanzielle Werteinheiten auch Tags (und nicht Konstanten) sind, da Preise/Kosten sich im Laufe der Zeit verändern.

Die Zielfunktion von APO ist die Summe von so vielen Summanden wie Sie sie haben wollen, wobei jeder Summand drei Bestandteile enthält, die miteinander multipliziert werden.

1. Der Faktor ist eine Zahl, mit der Einheiten umgewandelt werden oder mit der ein statischer Multiplikator aufgenommen wird, der aus chemischen oder sonstigen Gründen nötig ist. Er bietet auch die Möglichkeit, ein Minuszeichen einzufügen, so dass dieser Wert von der Zielfunktion abgezogen statt hinzuaddiert wird.
2. Der erste Tag ist der Haupttag dieses Summanden.
3. Der zweite Tag ist optional. Er wird vor allem für eine Zielfunktion gebraucht, die den Gewinn berechnet, so dass wir bei jedem Summanden die Menge mit dem

Preis multiplizieren müssen.

4. Es ist sehr wichtig, dass jeder Summand in den gleichen physikalischen Einheiten evaluiert wird, so dass die Addition der Summanden zu einer Summe einen aussagefähigen Sinn ergibt. In der Oberfläche können Sie auch einen Kommentar für jeden Summand hinterlegen, um Ihre Eingabe zu dokumentieren.

   Das Formular erlaubt es Ihnen, eine Zeitspanne zu spezifizieren, über die hinweg die Summanden der Zielfunktion evaluiert und als Gesamtergebnis miteinander addiert werden sollen. Das dient nur dem Überprüfen und wird das Modell in keinster Weise beeinflussen. Bitte nutzen Sie dieses Möglichkeit, um die Plausibilität der Werte zu überprüfen. In der Praxis stellen wir oft fest, dass Tags in anderen Einheiten aufgezeichnet werden, als man eigentlich erwarten würde (beispielsweise Tonnen statt kg), und das führt dann zu großen Abweichungen in den numerischen Werten. Zusätzlich dazu wird zuweilen ein korrektiver Faktor für die molare Masse einer Substanz nötig sein.

   Jedes in der Zielfunktion verwendeter Tag sollte idealerweise semi-kontrollierbar sein. Ist ein Tag kontrollierbar, kann der Optimierer es einfach auf seinen maximalen Wert einstellen. Ist ein Tag gar nicht kontrollierbar, kann man hinsichtlich dieses Zielaspektes ohnehin nichts machen. Diese Empfehlung ist allerdings nur eine Richtlinie und keine strenge Vorschrift. Wenn es – zur korrekten Gewinnberechnung – beispielsweise notwendig ist, einige unkontrollierbare Tags einzubeziehen, dann müssen diese eben auch mit einbezogen werden.

   Die Zielfunktion ist das Kernstück von APO und wird Sie bei jeder Entscheidung leiten. Darum muss sie mit großer Sorgfalt spezifiziert werden. Entspricht diese Funktion Ihren eigentlichen Geschäftszielen, wird APO Sie genau dorthin bringen.

4.4. Das Modell trainieren

Das Modell zu trainieren ist ein automatischer Prozess. Sie können das Training des Modells auslösen entweder über den APO Wizard oder über das APO-Menü, indem Sie die Modell-Option selektieren.

Das Formular wird Sie dazu auffordern, diejenige Methode auszuwählen, nach der Sie vorgehen wollen. In beiden Fällen wird das aktuelle Modell gelöscht und ein neues auf der Basis der historischen Daten erstellt. Die Methode nur modellieren wird genau das tun, aber nicht mehr. Bei dieser Vorgehensweise bleiben alle bereits gemachten Vorschläge und alle per Hand eingegebenen Reaktionen der Anlagenfahrer erhalten. Diese Option ist dann für Sie die richtige, wenn APO bereits einige Zeit gelaufen ist und Sie das Modell lediglich aktualisieren wollen, wobei Sie die in der Vergangenheit vorgeschlagenen und umgesetzten Empfehlungsanweisungen beibehalten möchten. Die Methode Neuberechnung wird hingegen nicht nur das alte Modell, sondern auch alle bislang gemachten Empfehlungen sowie per Hand eingegebenen Veränderungen löschen. Es wird dann alles neu erzeugen, nachdem das neue Modell erstellt ist. Das ist die geeignete Option, wenn Sie APO zum ersten Mal einsetzen oder wenn Sie Ihr Modell feinjustieren wollen.

Die Zeitbegrenzung kann verwendet werden, um die Neuberechnung aller historischen Empfehlungen zu beschleunigen, indem Sie nur eine Zeitspanne wählen zwischen dem Anfang der historischen Daten bis hin zu einem einzugebenden Zeitpunkt. Wir empfehlen diese Option allerdings nicht, sondern schlagen vor, das Modell aufgrund aller akkumulierten Daten zu erstellen.

Bitte beachten Sie, dass das Training eine beträchtliche Zeit in Anspruch nimmt. Die Zeit ist direkt und linear proportional zu der Anzahl von Datenpunkten der Datenbank. Der Prozess wird dominiert von der Lese- bzw. Schreibgeschwindigkeit der Datenbankvorgänge und somit von der Geschwindigkeit der Festplatte. Seien Sie geduldig.

Am Ende des Trainings, wird das Modell auf der Datenbank abgespeichert. Haben Sie Neuberechnung ausgewählt, werden Sie nun in der Lage sein, sich die neu berechneten Empfehlungen anzusehen und die Qualität des Modells zu bewerten. Wir empfehlen Ihnen nachdrücklich, dass Sie das auch wirklich tun.

4.5. Begutachtung der Modellqualität

Es gibt drei Schritte, um die Qualität eines trainierten APO-Modells zu bewerten:

1. Plausibilität prüfen
2. Den Modellbericht lesen
3. Das Modell feinjustieren

Wir werden hier den ersten Schritt beschreiben und die beiden nächsten erst beim Bericht und bei der Feinjustierung besprechen.

Die Modell-Plausibilität schaut sich die historischen Daten im Verhältnis zu den Spezifikationen an, die für jeder Tag gemacht wurden, und versucht dann zu prüfen, ob diese Spezifikationen Sinn machen. Das Erste, was überprüft wird, ist dieMessspanne eines jeden Tags. Aufgrund der historischen Daten wird berechnet, wie viele Punkte innerhalb und wie viele außerhalb der Spanne zwischen Minimum und Maximum liegen. Diese Spanne hat den Zweck, alle normalen und angemessenen Werte zu markieren. Aus diesem Grund sollte nur eine kleine Minderheit von historischen Beobachtungen außerhalb dieser Spanne liegen. Allerdings stellen wir in der Praxis häufig fest, dass manche Tags zahlreiche Punkte aufweisen, die außerhalb dieser Spanne liegen.

Die minimalen und maximalen Werte sind deshalb wichtig, weil jeder Punkt außerhalb der erlaubten Spannbreite von der Modellierung ausgeschlossen bleibt. Im Allgemeinen müssen diese Werte angepasst werden, bevor genügend valide Punkte verfügbar sind, um ein sinnvolles Prozessmodell zu trainieren. Im Plausibilitätsformular werden Sie nicht nur die Prozentzahlen der historischen Punkte ablesen können, die außerhalb der Spannbreite liegen, sondern auch die jeweils kleinsten und größten Werte für den jeweiligen Tag. Diese Informationen sollen Ihnen helfen, die zweckdienlichen Korrekturen vorzunehmen.

Die zweite Überprüfung betrifft die Messunsicherheiten der unkontrollierbaren Tags. Das Modell gruppiert die vorhandenen historischen Daten auf der Basis der Messwerte und Unsicherheiten der unkontrollierbaren Tags in sogenannte Cluster. Für alle so gruppierten Cluster wird ein separates Optimierungsmodell erstellt. Das

ist notwendig, damit keine operationalen Bedingungen vorgeschlagen werden, die unrealistisch und unerreichbar sind, weil eines oder mehrere der unkontrollierbaren Tags verschieden sind. Weil sie unkontrollierbar sind, stellen sie für den Prozess Randbedingungen dar. Je mehr Tags als unkontrollierbar markiert werden und je kleiner deren Unsicherheit ist, umso größere Restriktionen bedeuten sie für das Modell. Aber ein restriktives Modell wird weniger optimieren können.

Bei dieser Prüfung sollten Sie die unkontrollierbaren Tags überprüfen. Einige von ihnen könnten aus dem Modell entfernt werden, sofern sie nicht wirkliche Randbedingungen darstellen. Schauen Sie sich bitte die Unsicherheiten dieser Tags an. Das Plausibilitätsformular wird Ihnen sagen, wie viele Cluster auf der Basis des jeweiligen Tags gebildet wurden. Wenn Sie darüber im Zweifel sind, beginnen Sie lieber mit höheren Unsicherheitswerten, um ein vernünftiges Modell zu gewinnen. Diese Werte können später immer noch gesenkt werden, um das Modell restriktiver zu gestalten.

Der dritte Plausibilitätscheck betrifft die Validität der historischen Daten. Nur valide Punkte werden für das Training verwendet. Ein Punkt ist valide, wenn alle kontrollierbaren Tags innerhalb ihrer (minimalen und maximalen) Bandbreite liegen und wenn alle Validitätsregeln beachtet wurden. Vielleicht haben Sie auch eine Anzahl von maßgeschneiderten Bedingungen, die erfüllt werden müssen. Eine übliche maßgeschneiderte Bedingung ist, der Anlage einen stabilen Zustand vorzuschreiben, um die Validität zu erreichen. Das würde alle transienten Zustände vom Lernprozess ausschließen. Haben wir zu wenige valide Punkte in den historischen Daten, kann es sein, dass es zu wenige Daten gibt, von denen das Modell lernen kann. Weil der Betrieb einer Anlage im Allgemeinen einen Normalzustand erfordert, folgt daraus, dass, wenn es nur wenige valide Punkte gibt, einige Einstellungen korrigiert werden müssen.

Wenn diese drei Prüfungen zur Zufriedenheit durchgeführt wurden, sollte das maschinelle Lernen in der Lage sein, ein vernünftiges Prozessmodell zu trainieren. Bitte trainieren Sie nun das Modell und schauen Sie sich dann den Modellbericht an.

4.6. Den Modellbericht verstehen

Nachdem Sie das Modell trainiert haben, können Sie APO den Modellbericht generieren lassen. Entweder stellt dies den letzten Schritt im APO-Wizard dar oder Sie können diesen Bericht erhalten, indem Sie im APO-Menü die Option Bericht auswählen. Der Berichtgenerator erfordert es, dass man einen Zeitraum für den Bericht wählt und auch die Granularität des Berichts. Standardmäßig wird die gesamte Zeitperiode ausgewählt, für die es in der Datenbank Daten gibt. Der Bericht wird für diese Zeitspanne erstellt; und die Statistiken werden erhoben entsprechend der als Granularität angegebenen Häufigkeitsabständen.

Der Bericht selbst enthält eine Erläuterung seiner Datentabellen und Grafiken. Er befasst sich vor allem mit der Analyse von Optimierungspotenzialen, wie sie ihm von APO bereitgestellt werden. Jede Empfehlung bietet eine Gelegenheit, die Leistung der Anlage zu verbessern. Wird diese Empfehlung umgesetzt, wird aus diesem Potential ein tatsächlicher Gewinn. Diese Potentiale und tatsächlichen Gewinne werden im Bericht zusammengefasst. Bitte überprüfen Sie aber auch, ob die Dimension der

Verbesserung tatsächlich realistisch ist. Ist die Verbesserung zu groß, könnte es sein, dass das Modell noch nicht alle Restriktionen oder Randbedingungen kennt, denen der Verarbeitungsprozess unterliegt. Sind die Verbesserung zu gering, kann es sein, dass die Restriktionen, denen das Modell unterliegt, strenger sind als in Wirklichkeit. Eine detailliertere Analyse der eigentlichen Empfehlungen folgt im Laufe der Feinjustierung, aber schon jetzt sollte eine Bewertung der groben Verbesserungswerte vorgenommen werden, die der Bericht bereitstellt.

Der Bericht wird auch eine Liste der gebräuchlichsten Empfehlungen enthalten. Eine einzelne Empfehlung kann mehrere Zeilen von Veränderungen enthalten. Diese Veränderungen werden hier als ermittelter Durchschnitt aufgeführt und danach sortiert, wie oft sie auftreten. Sie werden sehen, wie oft die Tags verändert werden und um welchen Durchschnittswert sie verändert wurden. Dieser Durchschnittswert wird mit einer Standardabweichung zur Verfügung gestellt, so dass Sie daraus ersehen können, wie viel Abweichung von diesem Durchschnitt vorkommt. Tags, die in fast jeder Empfehlung erscheinen, haben wahrscheinlich einen Unsicherheitsfaktor, der zu klein ist und deshalb nahezu jeden Wert sub-optimal macht. Tags, die in so gut wie keiner Empfehlung erscheinen, dürften keinen großen Einfluss auf den Betriebsprozess haben und sollten vielleicht sogar ignoriert werden. Auf der anderen Seite könnten sie eine Unsicherheit haben, die zu groß ist, so dass ihr wahrer Einfluss unerkannt bleibt. Sind die Veränderungen oftmals sehr groß, so gibt es vielleicht begrenzende Faktoren, die man dem Modell aufnötigen soll.

4.7. Die Feinjustierung des Modells

Wenn Sie Ihr Modell erstellt haben und es sich in einem guten Zustand befindet – nämlich auf der Basis der Einstellungen der Tags, der Plausibilitätsüberprüfungen und des Modellberichts – so ist es Zeit, die Feinjustierung durchzuführen. Das ist ein Prozess, der einige Zeit in Anspruch nehmen kann und bei dem es darum geht, sich viele einzelne Empfehlungen genauer anzuschauen; aber es ist dieser Prozess, der Sie in die Lage versetzt und Ihnen das Vertrauen geben wird, das Modell in der Praxis tatsächlich einzusetzen.

Wählen Sie bitte aus dem APO-Menü die Option für die Feinjustierung (fine tuning). Um die Feinjustierung für die Anlage zu beginnen, klicken Sie bitte auf den Knopf Initialisieren. Nun werden Ihnen 100 zufällig ausgewählte historische Empfehlungen präsentiert, die Sie überprüfen können. Sie sind im Überblick der Feinjustierung gelistet. Sie können diese Liste weiter reduzieren, indem Sie im Formular den Knopf Filteranklicken. Sie können die Empfehlungen filtern, indem Sie entweder einen Empfehlungs-Zeitraum auswählen oder indem Sie prüfen, ob eine Empfehlung als "ok" markiert ist oder als "nicht ok". Zu Anfang sind alle Empfehlungen als "nicht ok" markiert.

Nun sollten Sie sich jede der Empfehlungen sorgfältig anschauen. Beachten Sie, dass jede Empfehlung zu einer besonderen Zeit, die in der Vergangenheit liegt, gemacht wurde, und zwar in Reaktion auf einen ganz bestimmten Zustand der Anlage und seiner Umgebung. Wenn Sie diese Empfehlung nun überprüfen, sollten Sie sich diese besondere Situation in Erinnerung rufen. Die Fragen, die Sie sich hinsichtlich jeder Empfehlung stellen sollten, sind diese:

1. Hätten alle die hier aufgeführten Empfehlungen tatsächlich umgesetzt werden können?
2. Welche Einstellungen im Modell müssen modifiziert werden, um unrealistische Empfehlungen zu verhindern?

Ist die Empfehlung, so wie sie ist, realistisch, markieren Sie sie bitte als ok. Ist sie nicht okay, können Sie die Kommentarfunktion nutzen, um einen Kommentar zu hinterlassen, der Ihnen auch zu einem späteren Zeitpunkt noch sagt, was daran nicht okay war, warum und was zu tun sei.

Eine Empfehlung ist selbsterklärend. Beachten Sie, dass jede Empfehlung immer nur etwas empfiehlt, was die Anlage in einen Zustand versetzt, den sie in der uns bekannten Historie zumindest einmal durchlaufen hat. Diese frühere Erfahrung ereignete sich zu einer Zeit, die in der Tabelle als "Vorige Erscheinung" markiert ist. Wenn Sie auf diesen Zeitstempel klicken, erscheint ein Vergleich mit Einzelheiten zwischen dem derzeitigen Zeitpunkt und jenem historischen Zeitpunkt, so dass Sie beide Zustände miteinander vergleichen können. Das ist nützlich, um zu verstehen, welchen vergleichbaren Zustand das Modell für sich heranzieht, um vielleicht die eine oder andere Veränderung am Modell vorzunehmen, oder um einfach zu verstehen, auf welcher Grundlage die Empfehlung basiert.

Wenn Sie alle Empfehlungen durchgegangen sind und eine Reihe von Veränderungen gesammelt haben, die Sie am Modell vornehmen möchten, führen Sie diese Veränderungen durch und trainieren Sie das Modell dann noch einmal unter den nun geänderten Bedingungen. Das Modell wird nun alle Empfehlungen neu berechnen. Gehen Sie dann noch einmal zurück zum Überblick der Feinjustierung und klicken Sie auf den Knopf Aktualisieren. Hier können Sie nun die angepassten Versionen der ausgewählten Empfehlungen nachlesen, die Sie sich zuvor angeschaut hatten.

Schauen Sie sich alle diese Empfehlungen also ein zweites Mal an. Diejenigen, die zuvor schon mit ok gekennzeichnet waren, werden auch jetzt noch ok sein und müssen nicht überprüft werden. Diejenigen, die zuvor nicht ok waren, sollten jetzt verbessert sein. Sind Sie Ihrer Meinung nach jetzt okay, markieren Sie sie auch als ok. Diese Vorgehensweise sollten Sie solange wiederholen, bis alle Empfehlungen als ok markiert sind. An diesem Punkt ist das Modell ausreichend angepasst da eine signifikante und repräsentative Menge an Empfehlungen als gut und praktikabel befunden wurden.

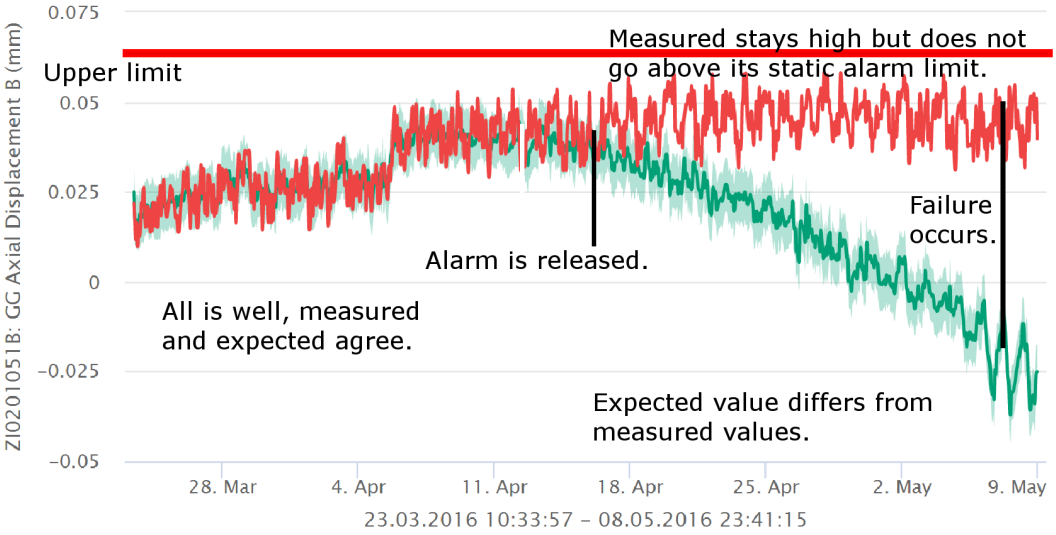
Sie können Ihre zu Ende gebrachte Arbeit jederzeit überprüfen, indem Sie auf Überblick klicken, damit Sie dort schnell nachzählen können, wie viele der Empfehlungen nach jedem Durchlauf als ok gekennzeichnet sind. Sie können auch auf den Bericht klicken, um einen ausführlichen und längeren Bericht über die Entwicklung einer jeden Empfehlung und all Ihrer Kommentare zu erhalten.

Am Ende der Feinjustierung kann das Modell für den realen Anlagenbetrieb verwendet werden. Wir empfehlen, dass Sie dieses Modell zunächst ein paar Tage lang als eine Art Probelauf betreiben, für den Fall, dass irgend ein Wesensmerkmal sich erst im laufenden Betrieb zeigt.

5. Intelligent Health Monitor (IHM)

Der Intelligent Health Monitor (IHM) macht die Zustandsüberwachung smart, indem er diese ins Zeitalter des maschinellen Lernens versetzt. Eine normale Zustandsüberwachung (condition monitoring) leidet darunter, dass immer wieder ein falscher Alarm ausgelöst wird, dass trotz schlechten Zustands gar kein Alarm ausgelöst wird und dass ein solches Monitoring erhebliche menschliche Expertise erfordert, um eingerichtet und aufrecht erhalten zu werden. Der IHM löst alle drei Probleme, indem er effektive, effiziente und präzise Methoden einsetzt, um den Gesundheitszustand der Geräte zu überprüfen. Das tut er, indem er die Definition des Gesundheitszustandes verändert: nämlich von Grenzwerten, die von Menschen für jeden Tag gesondert spezifiziert werden, hin zu einer ganzheitlichen Definition, die auf der in der Vergangenheit liegenden Leistung der Maschinerie basiert.

Ungesunde Zustände werden aufgespürt, weil beim ganzheitlichen Modellansatz die verschiedensten Messwerte einbezogen werden. Beispielsweise bieten Temperatur, Druck und Rotationsrate wertvolle Hinweise darüber, ob eine Vibration akzeptabel ist oder nicht. Durch diesen Ansatz werden falsche Alarm-Auslösungen verhindert. Weil diese Methode auf historischen Daten basiert, müssen Wartungsingenieure Alarmgrenzen nicht länger spezifizieren, warten und dokumentieren.



Als Folge der zunehmenden Verlässlichkeit hinsichtlich der Erkennung des Gesundheitszustands steigert sich die Verfügbarkeit der Maschinen und der ganzen Anlage. Mit der gesteigerten Verfügbarkeit verbessert sich auch die tatsächliche Produktionskapazität der Anlage. Weil es nun möglich wird, die maschinelle Ausrüstung proaktiv zu warten, kann die Wartung zunehmend vorausgeplant und darum auch kostengünstiger werden. Weil die maschinelle Ausrüstung seltener ausfällt, sondern proaktiv gewartet und repariert werden kann, werden Kollateralschäden vermieden und Wartungskosten erheblich gesenkt.

5.1. Condition Monitoring

Wenn wir in einer Industrieanlage technische Maschinen betreiben, sind wir ständig darum besorgt, ob diese Maschinen sich jederzeit in gutem Betriebszustand befinden. Tätigkeiten, die damit zu tun haben, diesen gesunden Betriebszustand zu ermitteln, fassen wir unter dem Begriff Condition Monitoring zusammen. Das wird im Allgemeinen dadurch erzielt, dass wir an den Maschinen Sensoren anbringen, die

Quantitäten wie Vibration, Temperatur, Druck, Durchfluss usw. messen. Diese Messwerte werden an das Leitsystem übertragen und gewöhnlich in einem Archivsystem gespeichert.

Im Zusammenspiel zwischen dem Leitsystem und dem Archivsystem werden die laufend eingehenden Daten analysiert, um den Zustand der Maschinerie zu bestimmen. Zu diesem Zweck definiert man in der Regel eine Ober- und Untergrenze für jeden Tag zur Auslösung eines Alarms. Bewegen sich die Messwerte jenseits der Ober- bzw. Untergrenzen, wird ein Alarm ausgelöst.

Wird ein Alarm ausgelöst, schaut sich ein Wartungsingenieur die Daten an und entscheidet dann, ob und was in diesem Fall zu tun ist. Entscheidet sich der Ingenieur dafür, nichts zu tun, handelt es sich gewöhnlich um einen falschen Alarm. Befindet sich die Maschinerie allerdings in keinem guten Zustand, ohne dass ein Alarm ausgelöst wird, sprechen wir von einem fehlenden Alarm. Beide Fälle sind problematisch. Der falsche Alarm ist unerwünscht, weil er Zeit und Ressourcen verschwendet. Der fehlende Alarm ist gefährlich, weil er wahrscheinlich zu einem ungeplanten Ausfall führt, der Kollateralschäden und Produktionsverlust nach sich zieht.

Der Grund für beide, den falschen Alarm und den fehlenden Alarm, liegt gewöhnlich in der zu simplen Art der Analyse: Die statische Beschaffenheit der Ober- und Untergrenzen ist nicht in der Lage, die Komplexität der verschiedenen Betriebsbedingungen industrieller Maschinen zu erfassen. Hinzu kommt, dass wenn jeder Messwert individuell analysiert wird, dies einen großen Nachteil bedeutet. Alle Messungen, die an einer einzelnen Maschine vorgenommen werden, sind naturgemäß irgendwie miteinander verbunden. Wenn diese natürliche Verbindung ignoriert wird, verzichtet die Analyse auf eine wichtige Informationsquelle.

Diesen Mangel kann man teilweise abstellen durch das, was man Event Framing nennt. Dabei werden bestimmte Betriebsbedingungen einer Maschine als zu einer Gruppe gehörig definiert, und man bestimmt dann die Ober- und Untergrenze für diese Gruppe. Beispielsweise kann man die Daten einer Turbine entsprechend in Volllast, Teillast und Leerlauf aufteilen, indem man eine Zustandsbedingung der Rotationsrate definiert. Obwohl das schon eine gewisse Verbesserung bedeutet, wird dadurch die handwerkliche Arbeit vermehrt, die nötig ist, um diese Art von Analyse einzurichten und zu erhalten. Bei Werksanlagen mit zehntausenden von Messwerten wird das allerdings schnell unermesslich und fehleranfällig.

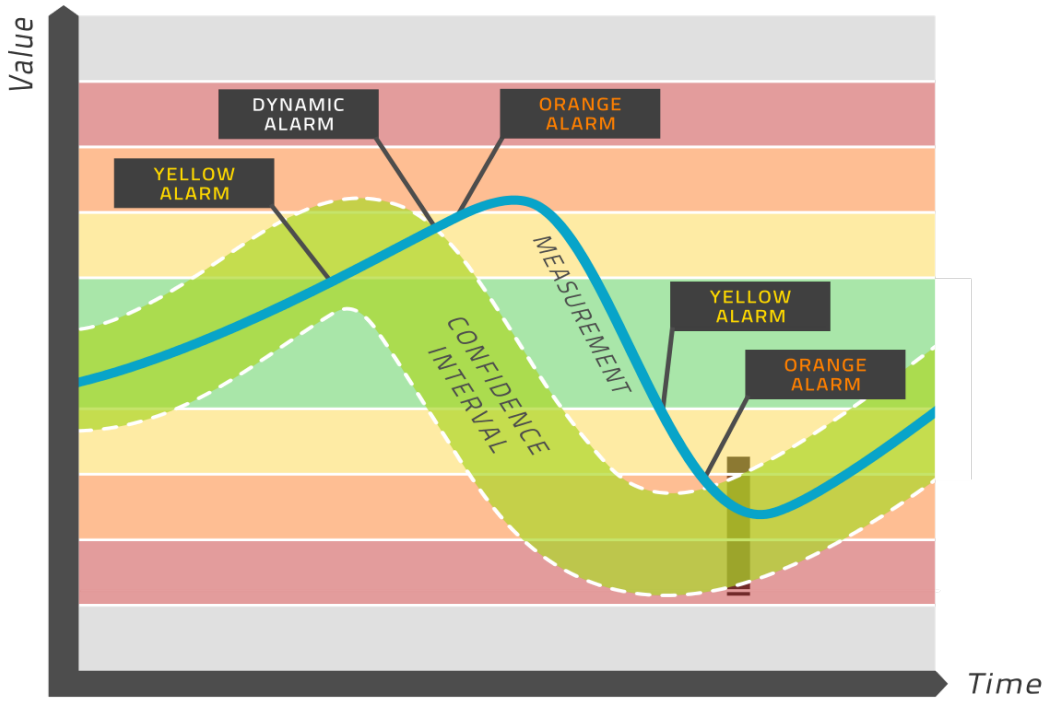
Wir können resümieren, dass normales Condition Monitoring den Gesundheitszustand eines Maschinenteils auf der Basis von Messungen definiert, für die – nach dem menschlichem Ermessen von Ingenieursfachleuten – gewisse Grenzwerte gesetzt werden. Doch diese Definition für den Gesundheitszustand ist in seiner Wirksamkeit begrenzt und erfordert erheblichen menschlichen Aufwand, und zwar nicht nur anfänglich, sondern fortlaufend.

5.2. Das Konzept der dynamischen Grenzwerte

Der Intelligent Health Monitor (IHM) konstruiert ein Modell des maschinellen Lernens für eine Zeitreihe als Funktion von mehreren solchen Zeitreihen einer ganzen Anlage oder einer Maschine. Wenn die zugrunde liegende Physik und Chemie des

industriellen Prozesses über längere Zeiträume dieselbe bleibt, wird der Modellprozess ein genaues, präzises und zuverlässiges Modell für die betreffende Messung abliefern können.

Dieses Modell wird auf der Basis eines Zeitraums konstruiert, innerhalb dem die Maschinerie bekanntermaßen gesund war. Somit wird dieses Modell als Definition des gesunden Zustands für den jeweiligen Tag zugrunde gelegt, das ja modelliert werden soll – statt dass man als gesunden Zustand die statischen Grenzwerte heranzieht wie beim normalen Condition Monitoring.



Das Modell wird in Echtzeit evaluiert und erbringt einen Erwartungswert für jeden relevanten Tag. Ein Konfidenzintervall wird um diesen erwarteten Wert herum berechnet, woraus sich eine Bandbreite von Werten ergibt (hellgrünes Band in der obigen Grafik), die als gesund gelten. Liegt der gemessene Wert innerhalb dieser Bandbreite, ist die Maschinerie gesund. Liegt der gemessene Wert außerhalb dieser Bandbreite, wird ein Alarm ausgelöst. Das steht in Kontrast zum üblichen Condition Monitoring, bei dem Alarme auf der Basis von Grenzwerten ausgelöst werden, die sich im Laufe der Zeit nicht verändern, wie normale gelb, orange oder rote Regionen, wie man sie in vielen Condition Monitoring Programmen findet. Das ist der Grund, weshalb wir diesen Ansatz als dynamische Grenzwerte bezeichnen. Schauen Sie sich bitte das Bild oben für eine visuelle Darstellung dieses Prozesses an.

Sollten Sie daran interessiert sein zu erfahren, wie das Modell selbst zustande kommt, können Sie sich das in der Sektion zum mathematischen Hintergrund ansehen.

5.3. IHM-Konzepte

Die Methoden des maschinellen Lernens nutzen empirische Daten, die in der Vergangenheit an einer Maschine gemessen wurden, als diese Maschine sich erwiesenermaßen in einem gesundem Zustand befand. Aus diesen Daten konstruieren die Methoden des maschinellen Lernens automatisch und ohne

menschliches Zutun eine mathematische Repräsentation der Beziehungen aller Parameter rund um diese Maschine.

Es kann mathematisch bewiesen werden, dass ein neuronales Netz in der Lage ist, einen komplexen Datensatz mit großer Genauigkeit darzustellen, solange das Netz groß genug ist und die Daten stets denselben Gesetzen gehorchen. Weil die Maschine den Naturgesetzen unterliegt, ist diese Voraussetzung natürlich gegeben. Aus diesem Grund verwenden wir ein neuronales Netz als Rahmenwerk, bei dem jede einzelne Messung an der Maschine im Zusammenhang mit den anderen Messungen der Maschine modelliert wird. Der Algorithmus des maschinellen Lernens findet die richtigen Werte für die Modell-Parameter, so dass das neuronale Netz die Daten genau repräsentiert.

Die Auswahl der Messungen, die für die Modellierung einer ganz bestimmten Messung in Betracht gezogen werden, kann automatisch erfolgen. Dazu benutzen wir eine Kombination zwischen Korrelationsmodellierung und Hauptkomponentenanalyse.

Abb. Die Verschiebung der Zentralachse eines Kompressors wird gemessen (rot) und modelliert (grün) anhand des Konfidenzintervalls des Modells (hellgrün). Es zeigt sich, dass das Modell das Verhalten der Maschine exakt modelliert, sogar während der Transienten, d.h. auch beim An- und Abfahren. Wir beobachten allerdings oft eine Abweichung zwischen dem Modell und den Messungen zu jenem Zeitpunkt, da Volllast erreicht wird. Das zeigt ein Maschinenproblem an, das einen Alarm auslöst.

Das Ergebnis ist, dass jeder Tag der Maschine eine Formel erhält, mit deren Hilfe der erwarteten Wert dieser Tag berechnet werden kann. Weil diese Formel auf Daten basiert, die sich erwiesenermaßen als gesund herausstellten, gilt diese Formel als die Definition des Gesundheitszustandes der Maschine. Ungesunde Zustände werden dann als Abweichungen von diesem Gesundheitszustand betrachtet.

Es ist wichtig, den optimalen Gesundheitszustand zu modellieren und nach Abweichungen von diesem Ausschau zu halten, da diese Gesundheit ja als der normale Zustand gilt, und für das normale, gesunde Verhalten liegen ja auch sehr viele Daten vor. Wenige Daten gibt es indes für Zustände, die ungesund sind, und überdies sind diese Daten auch sehr verschiedenartig – aufgrund einer Unzahl von unterschiedlichen Fehlfunktionen. Solche Fehlfunktionen unterscheiden sich je nach Hersteller und Modell einer Maschine, was die umfassende Darstellung von möglichen Fehlerquellen sehr komplex macht. Fehlerhafte Zustände zu beschreiben ist also weniger ein Problem der Datenanalyse als vielmehr ein Problem der Datenverfügbarkeit. Darum ist dieses Problem von fundamentaler Bedeutung, das in

der Praxis aber kaum umfassend angegangen werden kann.

Wir können jederzeit den erwarteten gesunden Wert mit dem gemessenen Sensorwert vergleichen. Weil sich der erwartete Wert aus dem Modell ergibt, kennen wir die Wahrscheinlichkeitsverteilung der Abweichungen, d.h. wir können vorhersagen, wie wahrscheinlich es ist, dass eine Messung von dem erwarteten (gesunden) Wert um einen bestimmten Betrag abweichen wird. Von dieser Wahrscheinlichkeitsverteilung können wir somit die Wahrscheinlichkeit eines gesunden Zustands berechnen oder, umgekehrt, das Konfidenzintervall dazu nutzen, um einzuschätzen, ob ein Sensorwert zu weit vom gesunden Zustand abweicht. In einem solchen Fall wird ein Alarm ausgelöst.

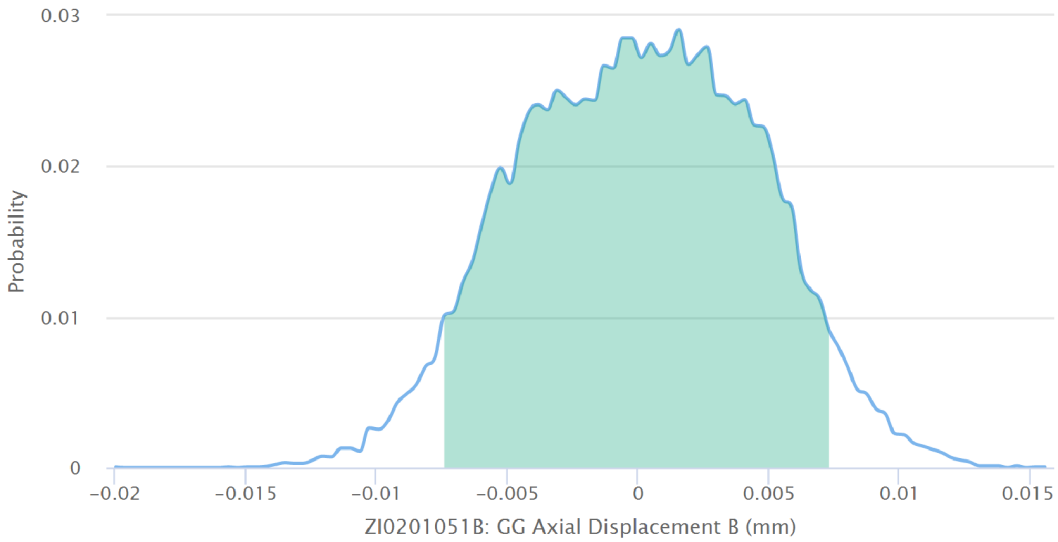


Abb. Die Abweichung zwischen Modell und Sensorwert (horizontale Achse) steht hier der Wahrscheinlichkeit gegenüber, mit der eine Abweichung auftritt (vertikale Achse). Wir erwarten eine glockenförmige Kurve für ein gutes Modell, d.h. viele Punkte mit kleinen Abweichungen und wenige Punkte mit großer Abweichung, bei einer allgemeinen Symmetrie zwischen den Abweichungen oberhalb und unterhalb des Modells. Von dieser Verteilung können wir leicht ablesen, wie wahrscheinlich eine zu beobachtende Abweichung sein wird und wie gesund der jeweils gemessene Zustand ist. Die Grafik übersetzt eine Sensormessung unmittelbar in einen Gesundheitsindex.

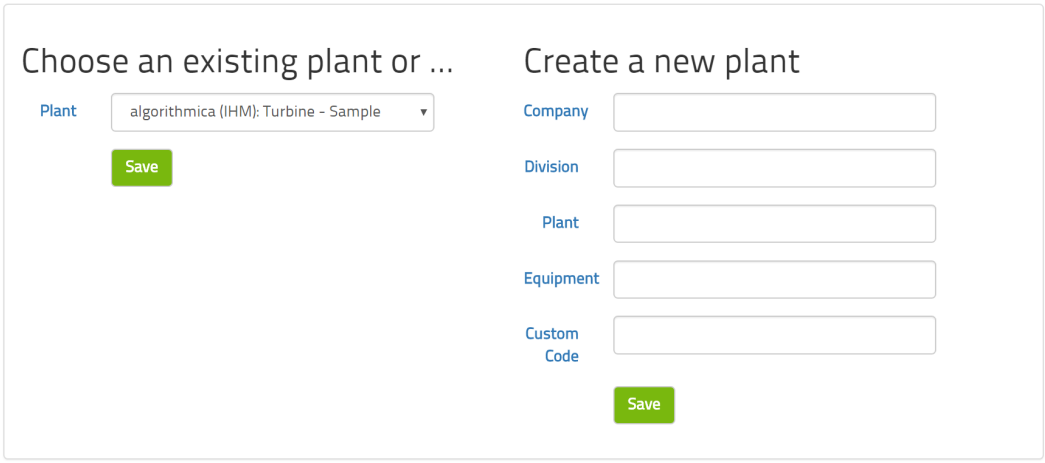
Der Alarm kann noch ergänzt werden durch die Information darüber, wie ungesund der Zustand ist, indem die Wahrscheinlichkeit eines schlechten Gesundheitszustands angegeben wird. Da der erwartete Zustand auf der Basis einer (gewöhnlich kleinen) Anzahl anderer Maschinen-Parameter berechnet wurde, ist es im Allgemeinen möglich, die Ursache einem anderen Messwert zuzuschreiben. Damit wird dem menschlichen Ingenieur eine Hilfestellung gegeben, der sich bei Auslösen eines Alarms um die Diagnose des Problems bemühen wird und eine Gegenmaßnahme ergreifen will.

Bei einem üblichen Condition Monitoring zeigt sich in der Praxis oft, dass, wenn eine Maschine von einem stabilen Zustand in einen anderen stabilen Zustand wechselt, dann zahlreiche (falsche) Alarme ausgelöst werden, da ein einfacher Analyseansatz

mit den sich schnell verändernden Bedingungen nicht fertig wird. Ein neuronales Netz hingegen kann ohne Weiteres extrem nicht-lineare Beziehungen darstellen; sogar ein Hochfahren der Maschine oder ein Lastwechsel können präzise modelliert werden, ohne dass ein Alarm ausgelöst wird, solange sonst alles so ist, wie es sein soll.

5.4. Inbetriebnahme

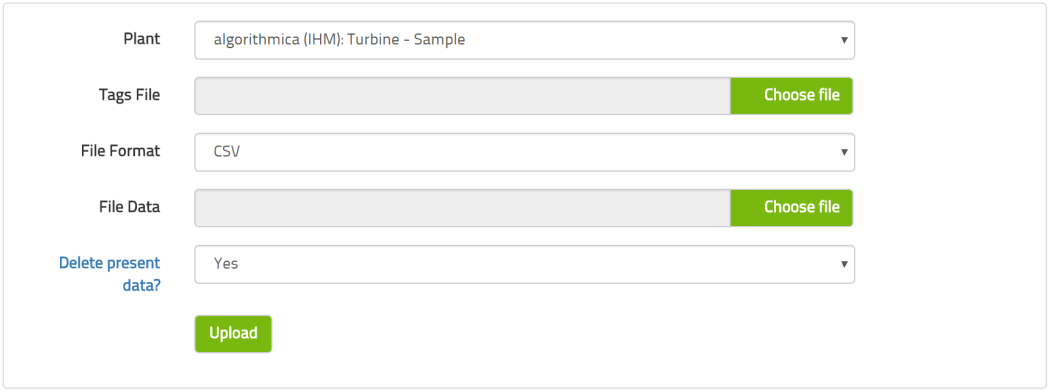
Nachdem Sie IHM auf Ihrem System installiert haben, werden Sie in der Oberfläche ein Menü vorfinden. Einer der Punkte ist der Assistent, darunter der IHM Setup Assistent. Wenn Sie darauf klicken, werden Sie durch einen Workflow geleitet, um einen neuen Datensatz mit Modell und dynamischen Grenzwerte zu generieren.



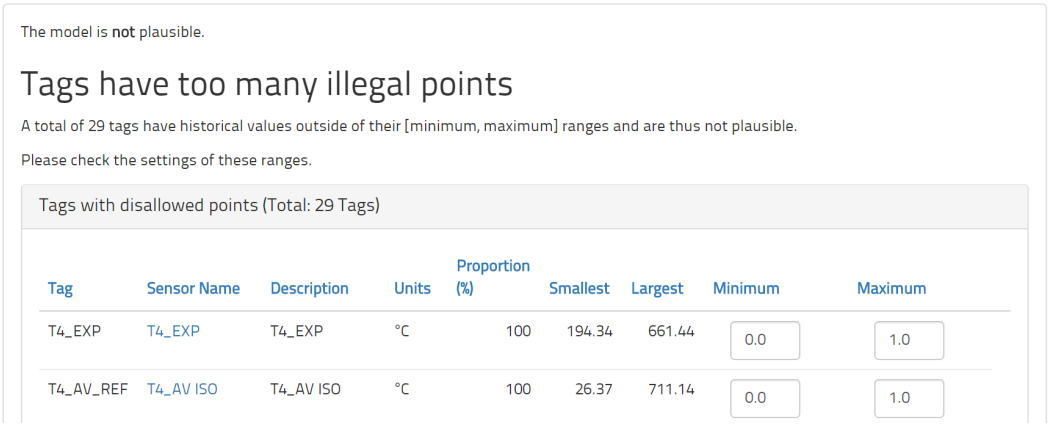
Als Erstes generieren Sie eine neue Anlage. Obwohl sie Anlage genannt wird, können Sie auch für jede neue Maschine eine solche eigene Anlage erstellen. Die Entscheidung darüber, welche Datenpunkte zu einem singulären Analyserahmen – hier als "Anlage" bezeichnet – gehören sollen, bleibt Ihnen überlassen und sollte davon abhängig gemacht werden, welche Teile zusammenhängen. Die IHM-Software kann mehrere Anlagen gleichzeitig handhaben.

	Tag	PLS Tag	Sensor Name	Description	Units	Minimum	Maximum	Delta	Limited	Low Green	High Green	Low Yellow	High Yellow	Low Orange	High Orange
1	SCH0201001	SCH0201001	GG Rotation Rate max Setpoint	GG Rotation Rate max Setpoint	rpm	0.00	15000.00	7500.00	No						
2	SCL0201001	SCL0201001	GG Rotation Rate min Setpoint	GG Rotation Rate min Setpoint	rpm	0.00	15000.00	7500.00	No						
3	SI0201001	SI0201001	GG Rotation Rate	GG Rotation Rate	rpm	0.00	15000.00	7500.00	No						14203.00
4	SI0201001A	SI0201001A	GG Rotation Rate A	GG Rotation Rate A	rpm	0.00	15000.00	7500.00	No						14203.00
5	SI0201001B	SI0201001B	GG Rotation Rate B	GG Rotation Rate B	rpm	0.00	15000.00	7500.00	No						14203.00
6	SI0201001C	SI0201001C	GG Rotation Rate C	GG Rotation Rate C	rpm	0.00	15000.00	7500.00	No						14203.00
7	ZI0201051A	ZI0201051A	GG Axial Displacement A	GG Axial Displacement A	mm	-1.00	1.00	1.00	Yes	-0.25	0.60	-0.35			0.70
8	ZI0201051B	ZI0201051B	GG Axial Displacement B	GG Axial Displacement B	mm	-1.00	1.00	1.00	Yes	-0.25	0.60	-0.35			0.70
9	VI0201061X	VI0201061X	GG Axial Vibration Forward Bearing X	GG Axial Vibration Forward Bearing X	µm	0.00	150.00	75.00	Yes		70.00				100.00
10	VI0201061Y	VI0201061Y	GG Axial Vibration Forward Bearing Y	GG Axial Vibration Forward Bearing Y	µm	0.00	150.00	75.00	Yes		70.00				100.00
11	VI0201071X	VI0201071X	GG Axial Vibration Backward Bearing X	GG Axial Vibration Backward Bearing X	µm	0.00	150.00	75.00	Yes		70.00				100.00
12	VI0201071Y	VI0201071Y	GG Axial Vibration Backward Bearing Y	GG Axial Vibration Backward Bearing Y	µm	0.00	150.00	75.00	Yes		70.00				100.00
13	KI0201081	KI0201081	GG Keyphasor	GG Keyphasor	-	0.00	1.00	0.50	No						
14	TI0201111A	TI0201111A	GG Forward Radial Bearing	GG Forward Radial Bearing	°C	-30.00	200.00	115.00	Yes		90.00				110.00
15	TI0201121A	TI0201121A	GG Backward Radial Bearing	GG Backward Radial Bearing	°C	-30.00	200.00	115.00	Yes		90.00				110.00
16	TI0201131A	TI0201131A	GG Thrust Bearing inactive	GG Thrust Bearing inactive	°C	-30.00	200.00	115.00	Yes		90.00				110.00
17	TI0201141A	TI0201141A	GG Thrust Bearing active	GG Thrust Bearing active	°C	-30.00	200.00	115.00	Yes		90.00				110.00

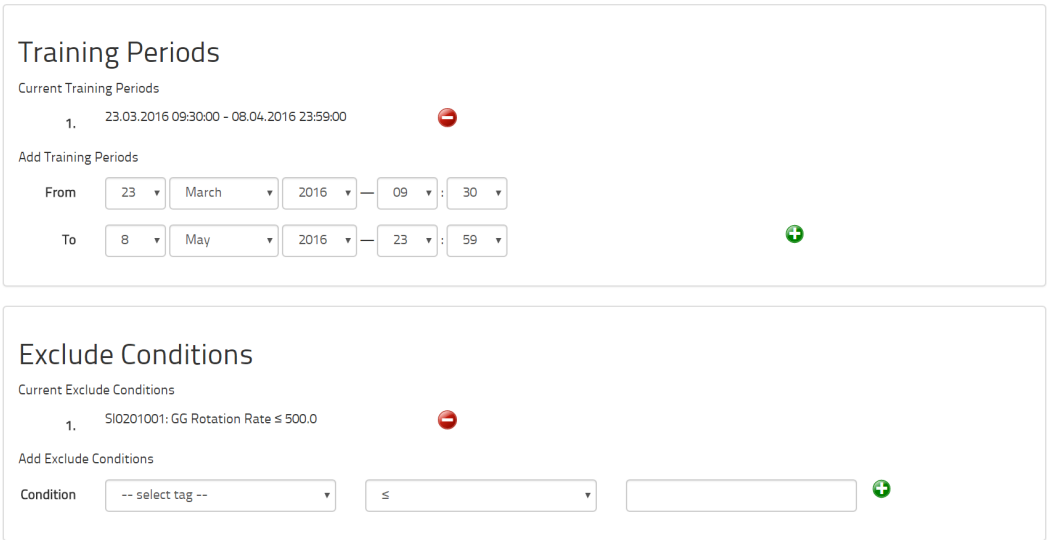
Als Zweites sollten Sie die Messungen und deren Metadaten definieren. Das wird in dem entsprechenden Wizard im Detail erläutert. Sie können diese Information direkt in das Formular auf dieser Seite eingeben; oder aber Sie bereiten diese Informationen extern vor, um sie im nächsten Schritt als Datei hochzuladen.



Drittens sollten Sie die historischen Daten hochladen, die Sie aus Ihrer Daten-Historie exportiert haben. Hier sollten Sie die Metadaten hochladen, sofern Sie sie zuvor noch nicht per Hand in das Online-Formular eingetippt haben.



Viertens wird es für diese Metadaten einen Plausibilitätscheck geben. Gibt es Unplausibilitäten, wird die Oberfläche erklären, welche diese sind und wie man sie korrigieren kann. Hier werden im Wesentlichen Tippfehler in den Messspannen und anderen Einträgen entdeckt, womit sichergestellt werden soll, dass alles numerischen Sinn ergibt.



Fünftens können Sie die Zeitspannen eintragen, in denen die Maschinerie sich in einem gesunden Zustand befand sowie etwa notwendige Ausschlussbedingungen. Diese beiden Wizard-Einträge sind auf alle Modelle dieses Datensatzes anwendbar.

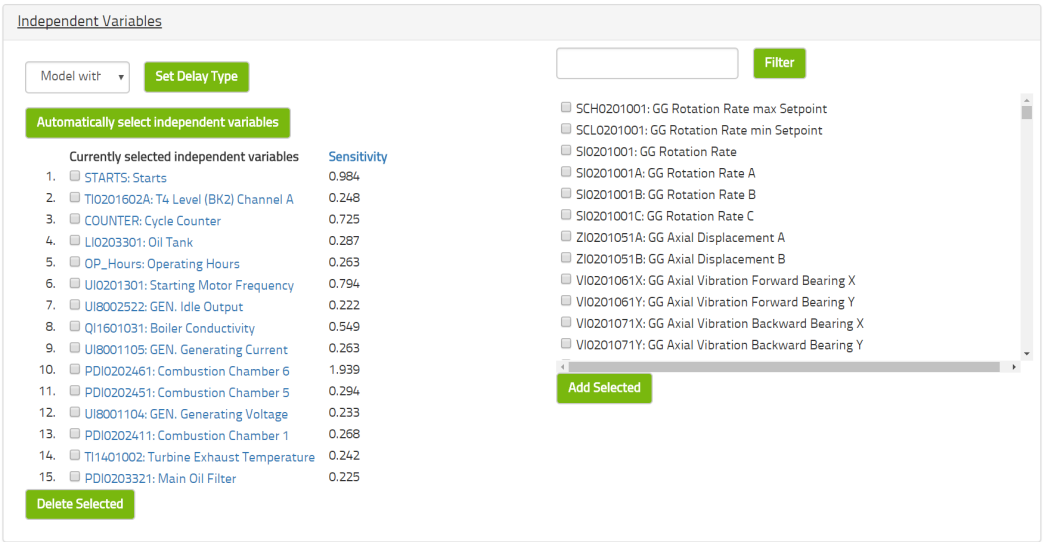
Sie können aber, wenn Sie möchten, diese beiden Informationen für jedes Modell individuell anpassen. Es ist wirkungsvoller, dies hier global zu tun. Es macht Sinn, die Zeiten mit gesundem Zustand und auch die Ausschlussbedingungen auf die ganze Maschine anzuwenden. Deshalb empfehlen wir, dass Sie die individuellen Anpassungen für das jeweilige Modell nur in Ausnahmefällen vornehmen.

Auf der letzten Seite befindet sich ein Knopf, um die Modellierung in Gang zu setzen. Wenn Sie auf diesen Knopf klicken, wird der Computer automatisch die unabhängigen Variablen selektieren und alle Modelle für die historischen Daten trainieren und anwenden, und zwar für jedes einzelne Modell, welches in den Metadaten angefordert wird. Abhängig von den angeforderten Modellen und der Anzahl der Trainingsdaten, kann dies eine erhebliche Zeit in Anspruch nehmen. Wir empfehlen deshalb, den Knopf am Ende eines Arbeitstages oder an einem Freitagnachmittag zu betätigen, damit der Computer viele Stunden lang arbeiten kann, ohne Sie bei Ihrer sonstigen Arbeit zu behindern.

5.5. Das Erstellen und Maßschneidern des Modells

Wir empfehlen, dem IHM Setup Wizard zu folgen, um zu Anfang mit einem neuen Datensatz zu beginnen. Es wird auch empfohlen, sich jedes Modell anzuschauen, um zu überprüfen, ob es gut ist oder nicht. Instruktionen darüber, wie man die Passgenauigkeit des Modells überprüft, befinden sich im Abschnitt zum mathematischen Hintergrund.

Wenn Sie auf die Bearbeitungsseite eines speziellen Modells gehen, können Sie dort die Trainingszeiten verändern und Ausschlusskriterien hinzufügen. Wir empfehlen Ihnen allerdings, das an dieser Stelle nicht zu tun, sondern solche Informationen vielmehr global für den gesamten Datensatz zu bearbeiten, indem Sie den IHM-Wizard benutzen. Solche Informationen für ein individuelles Modell zu verändern, sollte nur in besonderen Ausnahmesituationen gemacht werden.



Sie können die Auswahl von unabhängigen Variablen verändern, indem Sie Variablen von der Liste entweder entfernen oder ebensolche dort hinzufügen. Sie können weitere Einstellungen verändern wie etwa des gewünschte Konfidenzniveau oder die Anzahl der Neuronen im neuronalen Netz. Das Konfidenzniveau ist mit 0,9 vorgegeben, und die Zahl der Neuronen ist mit der doppelten Anzahl von

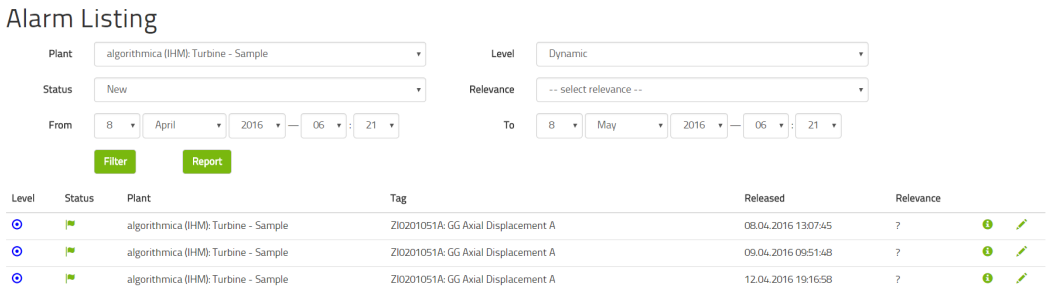
unabhängigen Variablen vorgegeben. Konsultieren Sie den Abschnitt zum mathematischen Hintergrund, wenn Sie mehr über diese Konzepte erfahren wollen.

Wenn Sie die gewünschten Veränderungen vorgenommen haben, finden Sie drei Knöpfe am Ende der Seite. Sie sind fürs Modellieren und Anwenden da oder um Beides zu tun (modellieren und anwenden). Modellieren heißt hier, dass das Modell unter den Bedingungen der neuen Einstellungen trainiert und in der Datenbank entsprechend abgespeichert wird. Anwenden bedeutet, dass das Modell auf der Basis der historischen Daten ausgeführt wird, wodurch die historischen Ergebnisse und Alarmauslösungen neu berechnet werden. Im Allgemeinen werden Sie sowohl modellieren als auch anwenden wollen.

Wenn Sie nur modellieren, bleiben die historischen Ergebnisse und Alarme erhalten. Das ist aber nur dann interessant, wenn Sie bereits mit dem System online gearbeitet haben und Sie die historischen Alarmauslösungen nur zum Zweck der Berichterstattung beibehalten wollen. Wenn Sie nur die Anwendung ausführen, wird das Modell durchgeführt, wie es ist. Hat sich das Modell nicht verändert, werden lediglich dieselben Ergebnisse reproduziert, wobei die Alarmhistorie sowie alle von Ihnen vorgenommenen manuell eingegebenen Alarmbearbeitungen gelöscht werden.

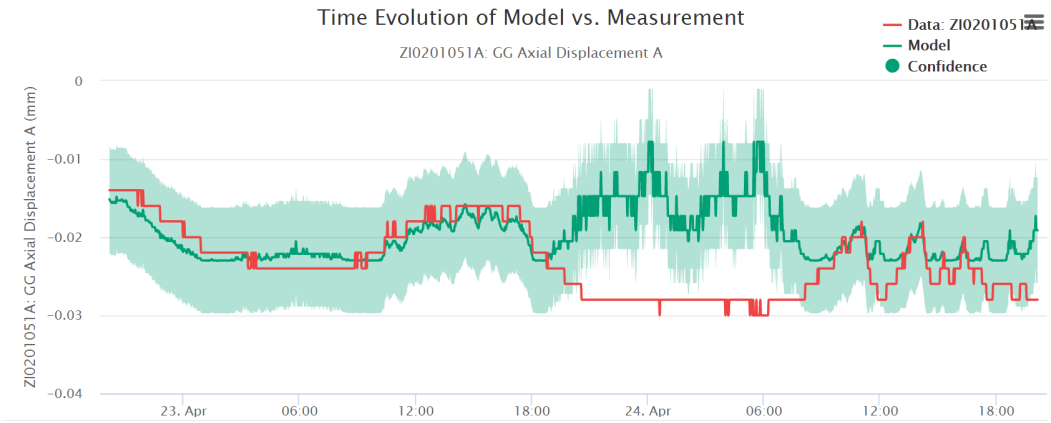
5.6. Alarme einrichten und verwalten

Ist das Modell erst einmal erstellt und angewandt worden, wird die Anwendung die historischen Alarmauslösungen berechnet haben. Diese kann man sich dann anschauen, indem man im Menü nach IHM und Alarm schaut. Um den Alarm auszuwählen, der Sie interessiert, können Sie mehrere Filter benutzen, um dann eine Liste von Alarmauslösungen zu erhalten.



Wenn Sie auf irgend einen Alarm klicken, werden Sie einen kurzen Bericht über diesen Alarm erhalten, einschließlich einer relevanten Grafik mit Informationen über die 24 Stunden vor und nach dem Alarm.

Alarm



Level	
Status	
Plant	algorithmica (IHM): Turbine - Sample
Tag	ZI0201051A: GG Axial Displacement A
Released	23.04.2016 20:09:30
Description	Dynamic alarm: Measurement -0.025986 outside dynamic range of [-0.0258932, -0.0123534].
Relevance	dealt with
Messages	1. **27.05.2017 21:28:57** (*admin*): Spare parts have been replaced

Wenn Sie auf Bearbeiten klicken, können Sie die Kategorie des Alarms verändern und Botschaften hinzufügen. Diese Informationen können Sie wieder abrufen, indem Sie diesen Alarm aufrufen oder den Alarmbericht erzeugen, der für periodische Dokumentationen nützlich sein kann.

6. Intelligent Soft Sensor (ISS)

Manchmal ist es schwierig oder teuer, die für uns interessante Quantität mit Hilfe eines Sensors zu messen, der an der Anlage angebracht ist. In diesem Fall könnten wir uns dafür entscheiden, diesen Wert mittels eines mobilen, handgehaltenen Sensors zu messen oder indem wir eine Probe entnehmen und diese im Labor untersuchen lassen. Doch kann dieser manuelle Prozess für sich und auf lange Sicht recht langwierig und teuer werden. Hinzu kommt, dass der gesuchte Wert immer erst dann zur Verfügung steht, wenn wir diesen händischen Prozess in Gang setzen und dann auch nur mit einiger Verzögerung.

Ein Soft Sensor ist eine kleine Software, die dem Zweck dient, den manuellen Messvorgang durch eine Computerberechnung zu ersetzen. So kann der Wert in Echtzeit und ohne manuelle Anstrengungen zur Verfügung gestellt werden. Diese Software kann genutzt werden, um Effizienz, Gas-Chromatographie, Schadstoffe oder andere schwer zu messende Größen zu ermitteln.

Der Soft-Sensor ist im Wesentlichen eine Formel, die eine uns interessierende Größe aufgrund von anderen Größen berechnet, die ganz normal gemessen werden können. Die Formel selbst wird durch maschinelles Lernen bestimmt. Zuerst bestimmt der Computer, welche Größen nötig sind, um den Wert zu berechnen, an dem wir interessiert sind. Dann werden die entsprechenden Daten gesammelt und das Modell trainiert. Schließlich kann bei Bedarf die Modell-Qualität überprüft und das

Modell entsprechend verbessert werden. Ist das Modell gut, kann es in Echtzeit eingesetzt werden, wobei es sich dann genau so verhält wie ein echter Sensor.

6.1. Einen Soft-Sensor erstellen

Im ISS-Menü können Sie die Liste aller Soft-Sensoren in Ihrer Anlage abrufen. Der Knopf am Ende der Seite führt Sie zu einer Seite, auf der Sie einen neuen Soft-Sensor erstellen können. Der Tag ist derjenige Tag, welcher die Daten für jene Größe enthält, die Sie interessiert und die Sie modellieren möchten. Der neue Tag ist die alphanumerische Nomenklatur für den neuen Tag, das sich als Ergebnis der Formel ergibt, die erstellt werden soll. Wir empfehlen, dass die Nomenklatur dem ursprünglichen Tag ähnelt, jedoch muss sie für diese Anlage eindeutig sein. Wenn Sie auf Erstellen klicken, wird nicht nur ein neues Soft-Sensor-Objekt generiert, sondern es wird der Tabelle mit den historischen Daten Ihrer Anlage auch eine neue Spalte hinzugefügt. Diese neue Spalte dient der Erfassung der berechneten Daten für den neuen Soft-Sensor.

Sie werden auf die Bearbeitungs-Seite des Soft-Sensors geleitet. Als Erstes müssen Sie eine oder mehrere Trainingszeiten definieren, welche die historischen Daten bestimmen, für die der Soft-Sensor trainiert werden soll. Von diesen Zeitspannen werden solche Punkte ausgeschlossen, die mit den Ausschlussbedingungen übereinstimmen, die Sie hier ebenfalls definieren können. Beide zusammen werden eine Liste von Datenpunkten ergeben, die fürs Training verwendet werden.

Dann müssen die unabhängigen Variablen ausgewählt werden. Das sind die Variablen, die in die Formel einfließen, die am Ende den Wert des Soft-Sensors berechnen soll. Wenn Sie wissen, welche Tags die gewünschte Größe beeinflussen, könnten Sie diese hier per Hand auswählen und dem Modell hinzufügen. Empfehlen tun wir das allerdings nicht. Es gibt dafür nämlich eigens einen Knopf, mit dessen Hilfe diese Auswahl automatisch getroffen wird, und zwar aufgrund einer Datenanalyse. Es ist besser, diese Auswahl erst dann per Hand nachzubearbeiten, wenn die automatische Auswahl zuvor getroffen wurde.

Wurden die Variablen, die in das Modell einfließen sollen, erst einmal ausgewählt, können nun noch weitere Modell-Parameter bearbeitet werden. Dem Soft-Sensor kann man einen Namen geben, und Sie können, wenn gewünscht, auch einen Kommentar hinzufügen. Wichtiger noch: Sie können auch die Anzahl der versteckten Neuronen bearbeiten. Die Formel für den Soft-Sensor wird ein neuronales Netz sein. Versteckte Neuronen sind im Wesentlichen die in dieser Formel enthaltenen freien Parameter. Je mehr versteckte Neuronen es gibt, umso mehr freie Parameter gibt es. Eine größere Anzahl von Parametern ermöglicht dem Modell in der Regel, sich besser auf die Trainingsdaten einzustellen, aber das gilt nur bis zu einem gewissen Punkt. Hier greift das Gesetz des abnehmenden Ertrags, und es kommt irgendwann auch der Punkt, an dem die Zahl der Parameter so groß wird, dass die Trainingsdaten sozusagen abgespeichert, aber nicht mehr erlernt werden. An diesem Punkt könnte das Modell vergangene Daten eins-zu-eins wiedergeben, aber es kann daraus nur schlecht eine verallgemeinernde Anwendung für eine neue Situation erstellen. Wir empfehlen deshalb, dass die Zahl der versteckten Neuronen nicht höher ist als etwa die doppelte Anzahl der unabhängigen Variablen.

Nachdem das alles eingerichtet ist, können Sie nun das Modell erstellen. Der Knopf Modellieren trainiert das Modell, der Knopf Anwenden nutzt das Modell, berechnet den Wert des Soft-Sensors für die gesamte zur Verfügung stehende Historie der Datenbank und speichert diese Werte auch dort; der Knopf Modellieren und Anwenden tut beides in einem Vorgang. Wir empfehlen, das Modellieren und Anwenden in einem Vorgang zu vollziehen. Danach können Sie sich das Ergebnis anzeigen lassen, um zu überprüfen, wie gut der Soft-Sensor ist.

6.2. Die Modell-Qualität bewerten

Navigieren Sie nun auf die Seite, auf der der Soft-Sensor angezeigt wird. Dorthin werden Sie automatisch gelenkt, nachdem Sie den Knopf "Modellieren und Anwenden" auf der Bearbeitungsseite betätigt haben; Sie können diese Seite aber auch mittels eines Links aus der Liste der Soft-Sensoren anklicken, die Ihnen im ISS-Menü angezeigt wird.

Oben auf der Seite können Sie die Zeitspanne auswählen, für die Sie die Zeitreihen angezeigt bekommen möchten. Damit werden Daten des ursprünglichen Tags und des Soft-Sensor Tags für eben diese Zeitspanne ausgewählt und dann auf zwei Weisen dargestellt: Die obere Grafik ist eine einfache Zeitreihen-Darstellung, und die zweite Grafik vergleicht den gemessenen Wert mit dem berechneten Wert. Idealerweise werden die beiden Tags gleiche Werte anzeigen. Realistischerweise werden sie sich jedoch unterscheiden aufgrund einiger inhärenter und zufallsbedingter Abweichungen, die mit dem Messvorgang zusammenhängen. Auf der Grafik, auf der beide Tags gegenübergestellt sind, finden Sie eine gerade schwarze Linie. Das ist die Ideallinie, die Sie erhalten würden, wenn beide Werte immer vollkommen gleich wären. Anhand dieser Linie können Sie beurteilen, bis zu welchem Grad die beiden übereinstimmen.

Die dritte Grafikkurve zeigt die Wahrscheinlichkeitsverteilung bei den Unterschieden zwischen den modellierten und den gemessenen Werten. Idealerweise ist dies eine Glockenkurve wobei die Mitte eine Abweichung von 0 bedeutet. Die Kurve sollte sich symmetrisch um diese Mitte herum gruppieren und sich nach oben scharf zuspitzen. Schauen Sie sich auch an, wo die Glockenkurve sich nach beiden Seiten abflacht. Dort gibt es je einen Punkt, an dem die von der Mitte scharf abfallende Kurve übergeht in eine sanfte Abflachung hin zur Achse; das ist der "Ellbogen" der Verteilkurve. Befindet sich dieser Ellbogen auf beiden Seiten bei einem Differenzwert vergleichbar mit der Messunsicherheit des ursprünglichen Tags, dann ist alles in bester Ordnung. Sollte die Kurve mehr als eine deutlich erkennbare Spitze haben, dürfte das Modell ein systematisches Problem enthalten, das mit hoher Wahrscheinlichkeit auf eine suboptimale Auswahl der unabhängigen Variablen zurückzuführen ist, d.h., dass dem Modell eine wichtige Informationsquelle fehlt.

Weiter unten auf der Seite sehen Sie einige statistische Größen, die die Bewertungsmethode noch genauer machen. Wenn Sie von der Qualität des Modells nicht überzeugt sind, sollten Sie einige Änderungen vornehmen: bei den Zeitspannen fürs Training, bei den Ausschlussregeln oder, noch wichtiger, bei der Auswahl der unabhängigen Variablen. Als letzten Ausweg könnten Sie dann noch die Zahl der freien Parameter anheben, indem Sie die Zahl der versteckten Neuronen erhöhen. Beachten Sie, dass eine unabhängige Variable durchaus auch eine Quelle der

Desinformation sein kann, so dass die Genauigkeit des Modells dadurch untergraben wird. Stellen Sie sich vor, dass Sie einen Soft-Sensor für die NOX-Emissionen eines Kraftwerks erstellen wollen; Sie können auch Ihre Herzfrequenz als unabhängige Variable einsetzen. Günstigstenfalls wird das keine Auswirkungen auf das Modell haben, aber es ist eher wahrscheinlich, dass dadurch das Modell geschwächt werden könnte, weil Ihre Herzfrequenz den Daten eine zusätzliche Struktur verleiht, ohne dass diese etwas mit der Zielgröße zu tun hat.

In dem Ausnahmefall, dass Sie zwar sorgfältig die Auswahl der unabhängigen Variablen und aller anderen Einstellungen überprüft haben und dennoch ein eher schlechtes Modell erhalten, sollten Sie schlussfolgern, dass es bei Ihrem Prozess noch eine wesentliche Information gibt, die Sie derzeit noch nicht messen oder die Sie dem Modell noch vorenthalten.

.

7. Hintergrund und Konzepte

Ein mathematisches Modell ist die mathematische Beschreibung eines Systems, das die Form einer traditionellen Gleichung hat, z.B. $y = mx + b$.

Das Modellieren ist ein Prozess, an dessen Ziel und Ende das mathematische Modell steht. Dieser Prozess beginnt meist mit Messdaten, die empirisch erhoben wurden. In der Industrie haben wir an den Anlagen Sensoren, die einen ständigen Datenstrom produzieren, mit dessen Hilfe wir ein Modell der Anlage erstellen können. Beachten Sie, dass bei dem Modellierungsvorgang Daten in ein Modell verwandelt werden, und zwar in ein Modell, das der Situation entspricht, wie sie von den Daten beschrieben werden. Das ist eigentlich alles.

In der Praxis ist es zwar schön, ein Modell zu haben, aber dieses allein löst ja noch kein Problem. Um ein spezielles praktisches Problem zu lösen, müssen wir das Modell praktisch anwenden. Das heißt, das Modell ist nicht schon das Ende der industriellen Problemlösung; vielmehr müssen der Modellierung noch weitere Schritte folgen; zumindest eine Art Unternehmensentscheidung und -analyse ist nötig. Das Modell ist auch nicht der Anfang der Problemlösung. Der Anfang der Problemlösung wird vielmehr dadurch gemacht, dass man das Problem erst einmal formuliert und dass genauer definiert wird, welche Daten nötig sind, die dann gesammelt und aufbereitet werden, um das Modell zu erstellen. Oft sind es gerade diese vorbereitenden Schritte, die der eigentlichen Modellierung vorausgehen, die besonders viel menschliche Zeit und Anstrengung erfordern, um die Lösung des Problems anzugehen.

Aus mathematischer Sicht ist das Modellieren eigentlich der komplizierteste Schritt auf dem Weg zu einer Problemlösung. Deshalb müssen wir gerade hier sorgfältig vorgehen – sowohl in Bezug auf die Methodologie als auch im Hinblick auf die Daten, denn vieles kann passieren, was einer schwarzen Kiste gleicht.

Allgemein gesprochen, beinhaltet das Modellieren zwei Schritte: (1) Die von Hand vorgenommene Auswahl einer funktionalen Form und eines Lernalgorithmus, und (2) die automatische Ausführung des Lernalgorithmus zur Bestimmung der Parameter

der ausgewählten funktionalen Form. Es ist wichtig zwischen beiden Schritten zu unterscheiden, denn der erste Schritt sollte von einem erfahrenen Modellierer vorgenommen werden, nachdem er sich mit dem Problem ausreichend vertraut gemacht hat, während der zweite Schritt eine Frage der zur Verfügung stehenden Rechnerkapazitäten ist (vorausgesetzt, die notwenige Software steht zur Verfügung). In der Praxis stellen diese beiden Schritte jedoch meist eine Schleife dar. Das heißt: Das Ergebnis der Berechnung macht es notwendig, die von Hand durchgeführte Auswahl zu überprüfen usw. Nach ein paar Schleifen kann schließlich ein Lernprozess beim menschlichen Modellierer einsetzen, der erkennt, welcher Ansatz am besten funktioniert. Das Modellieren ist somit ein Erkenntnisprozess mit einem ungewissen Zeitrahmen. Es handelt sich nicht um eine mechanische Anwendung von Regeln oder Methoden.

7.1. Konzepte

7.1.1. Konzept eines Modells

Das Modell einer Messung ist eine Formel, die diese Messung berechnet, und zwar aufgrund von anderen Daten wie etwa anderen Messungen. Ein einfaches Beispiel dafür ist die bekannte thermische Zustandsgleichung idealer Gase aus der Thermodynamik ***PV = nRT***. *Für ideales Gas setzt die Gleichung den Druck **P** und das Volumen **V** in Beziehung zur Temperatur **T** und der Menge des Gases **n**. Die Beziehung hängt von einem Parameter **R***

Als eine praktische Anwendung könnten wir ohne Weiteres das Volumen, den Druck und die Temperatur unseres Gases messen und so bestimmen, wie viel Gas sich in einem Behälter befindet, und zwar dadurch, dass wir die Gleichung einsetzen, ohne dass wir eine Messung über eine komplexe Durchflussmessung vornehmen. Ein solches Modell ist somit in der Praxis sehr nützlich, vorausgesetzt uns liegen die Daten der anderen unbekannten Größen vor.

An dieser Stelle müssen wir ein paar Termini präzisieren. Wir sprechen von einer Variable, wenn wir diese durch Messung oder Berechnung bestimmen, denn sie verändert sich im Laufe der Zeit, wie z.B. der Druck oder die Temperatur. Andere Dinge nennen wir Parameter, weil diese konstant bleiben und nur einmal (vor)bestimmt werden müssen, z.B. die ideale Gaskonstante. Eine Variable gilt als abhängig, wenn es sich um eine Variable handelt, die wir aufgrund von anderen Variablen berechnen wollen; diese anderen Variablen nennen wir dann unabhängig. Es sind die unabhängigen Variablen, die gemessen werden müssen, wohingegen die abhängigen Variablen berechnet werden können.

*Wenn wir, um bei obigem Beispiel zu bleiben, die Gasmenge berechnen wollen, dann ist **n** die abhängige Variable, die man berechnen kann, indem man **n = PV/RT** setzt, wobei **P**, **V** und **T** die unabhängigen Variablen sind und **R** ein Parameter ist.*

7.1.2. Modellierungsprozess

Um vom Datensatz zu einem fertigen Modell zu gelangen, bedarf es mehrerer Schritte. Einige dieser Schritte erfolgen automatisch durch die Software und bleiben für den Nutzer unsichtbar, aber wir wollen sie zum besseren Verständnis hier beschreiben.

Als Erstes müssen wir eine Formel bzw. Gleichung wählen, die am besten in der Lage ist, die uns zur Verfügung stehenden Daten zu repräsentieren. Haben wir es beispielsweise mit einem idealen Gas zu tun, dann wird die Zustandsgleichung idealer Gase eine gute Wahl sein. Ist das Gas nicht ideal, wird diese Gleichung nur in Annäherung funktionieren. Wir müssen uns der der Gleichung innewohnenden Begrenzungen bewusst sein, wenn wir anfangs die Gleichung auswählen. Es ist an dieser Stelle gut, sich auch bewusst zu machen, dass neuronale Netze erwiesenermaßen in der Lage sind, jeden Datensatz mit großer Genauigkeit darzustellen. Aus diesem Grund verwenden wir neuronale Netze als die Gleichung unserer Wahl.

Als Zweites wählen wir die Daten aus, mit denen das Modell trainiert werden soll. Damit das überhaupt Sinn macht, müssen wir verstehen, dass die Parameter eines mathematischen Modells zu diesem Zeitpunkt noch nicht bekannt sind. Um noch einmal auf die Gleichung für ideale Gase zurückzugreifen: Das hieße, dass wir den Wert der idealen Gaskonstanten *R* nicht kennen würden. Um den Wert des Parameters zu bestimmen, benötigen wir die Daten für alle Variablen dieser Gleichung. Wollen wir den besten Wert des Parameters (oder der Parameter) der Gleichung aufgrund der historischen Daten für diese Variablen bestimmen, so sprechen wir davon, das Modell zu trainieren.

Als Drittes überprüfen wir unsere Arbeit. Es ist üblich, den Datensatz in zwei Teile zu unterteilen. Einer wird verwendet, um das Modell zu trainieren. Der andere wird verwendet, um zu überprüfen, ob das Modell auch für solche Daten sinnvolle Ergebnisse erzielt, die nicht fürs Training verwendet wurden. Hier sprechen wir von Validierungsdaten. Ergeben die zuvor bestimmten Parameterwerte genaue Ergebnisse für die Validierungsdaten, ist das Modell gut und wir können den Modellierungsvorgang als beendet betrachten.

7.1.3. Die unabhängigen Variablen auswählen

Wenn wir noch einmal auf die Zustandsgleichung für ideale Gase zurückblicken, so können wir sehen, dass wir drei unabhängige Variablen haben, nämlich den Druck *P*, das Volumen *V*, und die Temperatur *T*, um daraus die Gasmenge *n* zu berechnen. Das erfordert das Modell. Aber nehmen wir einmal für den Augenblick an, wir wüssten das nicht.

Der Datensatz wird viele Variablen enthalten, die uns zur Verfügung stehen. Da gibt es wahrscheinlich verschiedene Drücke und Temperaturen. Es mag auch Vibrationen, Flüsse, Ebenen, Energieverbrauch usw. geben. Aus dieser Vielfalt von Angeboten müssen wir die drei unabhängigen Variablen auswählen, die uns die gewünschte Informationen liefern, d.h. jene Informationen, die wir brauchen, um die für uns interessante abhängige Variable zu berechnen.

Es gibt zwei Arten, diese Auswahl zu treffen. Als Erstes könnten wir unsere menschliche Intuition und unseren Sachverstand einsetzen. Es waren solche Einsichten, die beim idealen Gas zur Geltung gebracht wurden. Dieser Ansatz erfordert Sachverstand und auch etwas Zeit, ihn umzusetzen. Weil unser Sachverstand aber nicht in allen Fällen endgültig und eindeutig ist, sondern vielmehr hypothetischen Charakter hat, bedarf es mehrerer Versuchsschleifen von Versuch

und Irrtum, bis wir schließlich ein gutes Modell gefunden haben. Das ist freilich ein aufwändiger Prozess.

Als Alternative könnten wir den Computer einsetzen, um für uns eine Datenanalyse durchzuführen. Es gehört zu den wichtigen Vorzügen dieser Software, dass die Auswahl von unabhängigen Variablen automatisch erfolgt und dass mit hoher Wahrscheinlichkeit ein Satz von Variablen selektiert wird, die ein gutes Modell ergeben.

Auf der Bearbeitungsseite für die dynamische Grenzwerte oder für Soft-Sensoren finden Sie unter der Überschrift "Unabhängige Variablen auswählen" einen Knopf zur automatischen Auswahl unabhängiger Variabler. Diese Funktion geht davon aus, dass Sie bereits die Trainingszeiten und die Ausschlussbedingungen definiert haben. Sie wird sich dann die Daten holen, die mit diesen Vorbedingungen übereinstimmen und dann eine Korrelations-Analyse zwischen der abhängigen Variable und allen anderen Variablen wie auch eine Analyse aller Variablen untereinander durchführen. Die Analyse wird dann solche Variablen auswählen, die eine hohe Korrelation mit der abhängigen Variable, aber wenig Korrelation miteinander haben. Dadurch wird sichergestellt, dass wir ausreichende, aber keine redundanten Informationen im Datensatz haben, weil wir ja die Zahl der unabhängigen Variablen so niedrig wie möglich halten wollen, ohne dadurch die Genauigkeit des Modells zu gefährden. Nach der automatischen Auswahl können Sie die Liste der selektierten Variablen, so gewünscht, gerne noch nachbearbeiten.

Nachdem nun die unabhängigen Variablen ausgewählt wurden, kann das Modell trainiert werden.

7.1.4. Konfidenzintervall

Das Modell wird zu einem gegebenen Zeitpunkt ganz bestimmte Ergebnisse erhalten. Es handelt sich um den Wert, den wir als Messergebnis erwarten – freilich unter der Voraussetzung, dass sich die Maschinerie in gutem Zustand befindet. Befindet sich die Maschinerie in gutem Zustand, sollten der erwartete Wert und der gemessene Wert nahe beieinander sein. Ist die Maschinerie nicht in gutem Zustand, sollten die beiden Werte weit auseinander liegen.

Das Konzept nahe beieinander und weit auseinander muss allerdings präzisiert werden, um zu bestimmen, wann genau ein Alarm ausgelöst werden soll. Dafür verwenden wir das Konzept des Konfidenzintervalls, wofür wir im Allgemeinen einen Bereich (ein Intervall) von Werten auswählen, von denen eine bestimmte Anzahl innerhalb dieses Bereiches liegt. In unserem Fall wird das Konfidenzintervall rund um den Modell-Wert gezogen, so dass sich ein bestimmter Anteil der Trainingswerte innerhalb des Intervalls befindet. Diesen Anteil nennen wir das Konfidenzniveau und es ist in der Regel gleich 0,9. Das bedeutet, dass das voreingestellte Konfidenzintervall 90% der Trainingsdaten umfasst und die anderen 10% ausschließt.

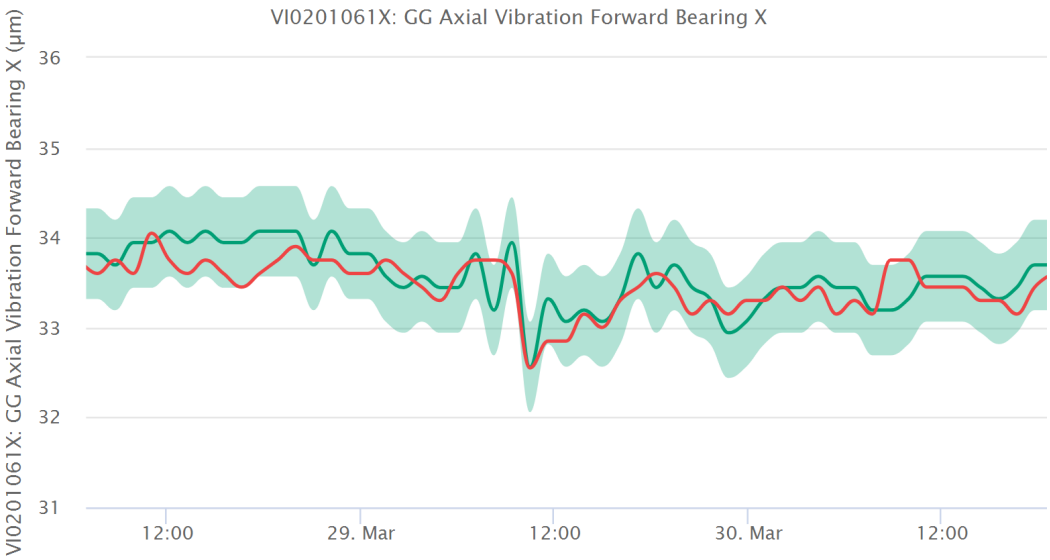
Das Konfidenzniveau kann vor dem Training angepasst werden – abhängig davon, wie stringent Sie Ihre Definition von "gesundem Zustand" definieren wollen. Einem wichtigen Maschinenteil wird man ein niedrigeres Konfidenzniveau zuordnen, so dass ein Alarm schon bei kleineren Abweichungen vom optimalen Gesundheitszustand ausgelöst wird. Weniger wichtigen Maschinenteilen kann man

ein höheres Konfidenzniveau zuordnen, damit Abweichungen hier nicht so oft einen Alarm auslösen. Das Konfidenzniveau ist umgekehrt proportional zu der Anzahl von Alarmen, die ausgelöst werden, d.h., je höher das Vertrauen (die Konfidenz), umso weniger Alarme werden ausgelöst.

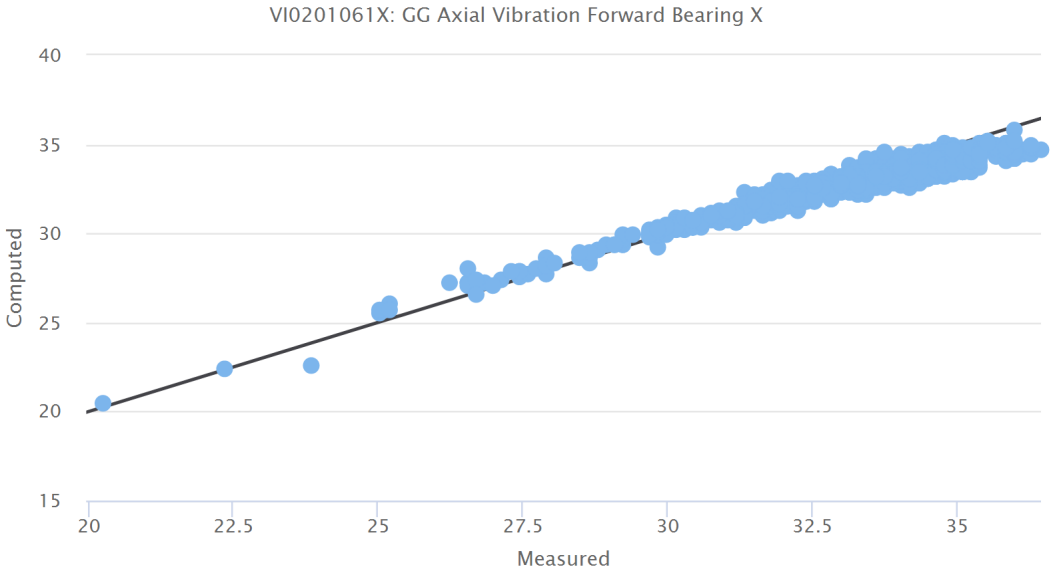
Das Konfidenzniveau ist ja nur eine zweckmäßige Zahl, der dem Modellierungs-Algorithmus die Möglichkeit gibt, das Konfidenzniveau auf der Basis der Daten zu berechnen. Es ist dieses berechnete Konfidenzintervall, das dazu dient, den Alarm auszulösen. Wenn Sie möchten, können Sie das Konfidenzintervall nach dem Training von Hand bearbeiten, um das Alarmaufkommen zu erhöhen oder zu minimieren.

7.1.5. Die Modellqualität beurteilen

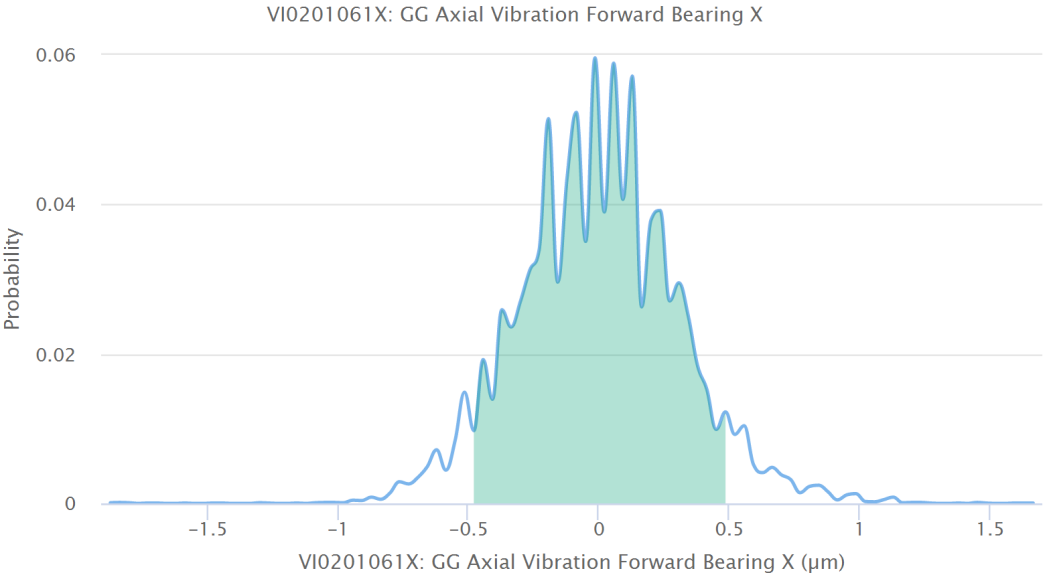
Das Modell sollte die Trainingsdaten sehr gut reproduzieren. Der echte Test für die Eignung des Modells liegt allerdings bei jenen Daten, mit denen das Modell nicht trainiert wurde. Aus diesem Grund wird der von Ihnen für die Zeitspanne des Trainings spezifizierte Datensatz automatisch unterteilt in einen Datensatz fürs Training und einen Datensatz zur Validierung.



Nach dem Training wird das Modell automatisch auf seine Eignung taxiert, und das kann in der Oberfläche auf verschiedene Weise angeschaut werden. Der direkteste Weg ist die historische Grafikdarstellung von Modell und Messung, bei der Sie die zeitliche Entwicklung sowohl des Modells als auch der Messungen nachvollziehen können. Wenn beide während des optimalen Gesundheitszustandes nahe beieinander und bei suboptimalem Gesundheitszustand weit auseinander befinden, dann funktioniert das Modell gut.



Im Allgemeinen erwarten wir, dass es für die Abweichungen zwischen Modell und Messung kaum Unterschiede gibt, wenn wir den Datensatz fürs Training mit dem Datensatz für die Validierung vergleichen, weil beide von dem optimalen Gesundheitszustand der Maschinerie ausgehen. Vergleichen wir in der grafischen Darstellung Modell mit Messung, so erwarten wir eine gerade Linie bei 45 Grad für beide Achsen welche im Diagramm durch eine schwarze Linie dargestellt wird.



Zusätzlich dazu erwarten wir auch Abweichungen zwischen Modell und Messung dergestalt, dass die meisten davon nur sehr klein sind. Stellen wir diese Abweichungen auf der horizontalen Achse dar und den Anteil der Punkte, die abweichen, auf der vertikalen Achse, so ergibt sich eine Wahrscheinlichkeitsverteilung der Abweichungen. Auch das wird in der Oberfläche dargestellt. Wir erwarten hierfür eine glockenförmige Kurve, ideal für eine normale oder Gaußsche Verteilung. Ist das der Fall, so ist dies ein weiterer Indikator für eine gute Modellqualität.

Wir messen auch die statistischen Größen wie Durchschnitt, Standardabweichung, Schiefe, und Wölbung für diese Wahrscheinlichkeitsverteilung, um weitere Details anzubieten. In der Praxis reicht eine visuelle Betrachtung dieser drei Grafiken aus, um

zu beurteilen, ob das Modell gut ist oder nicht.

Ist das Modell nicht gut, gibt es dafür zwei mögliche Gründe. Der erste könnte sein, dass die von Ihnen ausgewählten unabhängigen Variablen nicht die richtigen sind. Entweder müssen wir dann einige weitere hinzufügen oder einige störende entfernen. In der Regel bedarf es der Hinzufügung von weiteren Variablen. Zweitens könnte es sein, dass die ausgewählten Trainingszeiten nicht lang genug waren oder auch ungesunde Gesundheitszustände enthielten. Bitte überprüfen Sie das sorgfältig, weil die Integrität der Daten, die verwendet werden, um einen gesunden Zustand zu definieren, den entscheidenden Faktor für die gesamte Analyse ausmachen.

7.1.6. Messunschärfe

Es ist wichtig, sich zu vergegenwärtigen, dass alle empirischen Messungen unsicher sind. Wenn Sie sich morgens auf Ihre Körperwaage im Badezimmer stellen und feststellen, dass Sie 80 kg wiegen, fragen Sie sich bitte, was das bedeutet.

Zunächst: Die Waage hat eine ihr innewohnende Unsicherheit aufgrund der verwendeten Federn, mit deren Hilfe Ihr Gewicht gemessen wird und mit der die Anzeige bewegt wird. Diese Messunsicherheit wird gewöhnlich vom Hersteller im Handbuch angegeben.

Zweitens gibt es eine Unsicherheit des Ablesens der Messung. Weil Sie nach unten auf die nahe am Boden befindliche Anzeige schauen, ist ein Unterschied zwischen 80 kg und 81 kg für Sie kaum erkennbar.

Drittens gibt es Befindlichkeitsveränderungen. Weil Sie in den letzten Stunden Nahrung und Getränke konsumiert haben, hat sich Ihr Körpergewicht verändert, auch wenn Sie praktisch kein Fett hinzugewonnen oder abgenommen haben.

Diese drei Fehlerquellen addieren sich auf und machen die Messung unsicher um vielleicht 2 kg. Die gemessenen 80 kg sind somit in Wirklichkeit 80 kg +/– 2 kg, d.h.: Ihr Körpergewicht befindet sich in der Spanne zwischen 78 und 82 kg.

Alle drei Arten von Fehlerquellen lassen sich auch für industrielle Datensätze geltend machen. Schon der Sensor hat eine inhärente Unsicherheit; auch die Messkette vom Sensor zum Archivsystem beinhaltet eine Unsicherheit; und schließlich gibt es aufgrund der unterschiedlichen örtlichen Bedingungen – nämlich abhängig von der genauen Anbringung des Sensors – eine erhöhte Variabilität, die nicht unbedingt charakteristisch ist für die durchschnittliche Situation, um die es gewöhnlich geht.

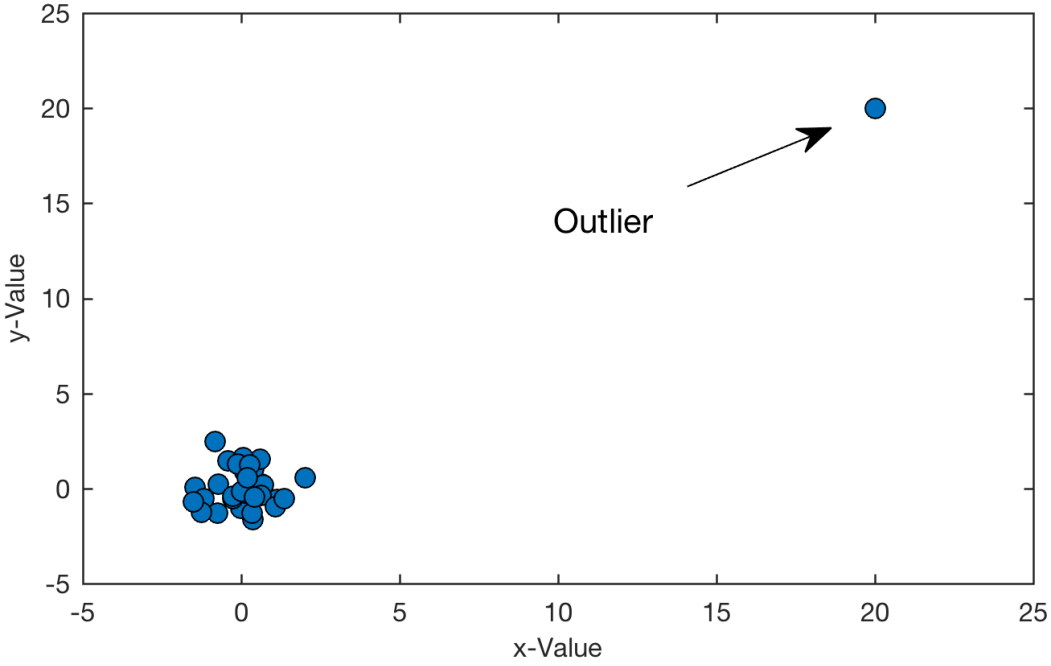
Die Bedeutung dieser Erkenntnis liegt in dem Umstand, dass, wenn wir für unsere Berechnung unsichere Daten verwenden, das berechnete Ergebnis bereits Unsicherheiten von den jeweiligen Datenquellen übernimmt. Wir können die Unsicherheit des berechneten Wertes bestimmen, wenn wir die Messunsicherheit der ursprünglichen Daten kennen. Bei mathematischen Modellen kommt es nicht nur darauf an, wie gut geeignet ein Modell ist, sondern auch, wie genau es ist.

7.1.7. Vorverarbeitung

Bevor Rohmessdaten im Zusammenhang mit maschinellem Lernen eingesetzt werden, durchlaufen diese gewöhnlich mehrere Bearbeitungsstufen. Diese

Methoden, die zusammen als Vorverarbeitung (engl. pre-processing) bezeichnet werden, sind von essentieller Wichtigkeit, um gute Ergebnisse zu erzielen.

Unplausible Werte werden vom Datensatz entfernt, und zwar entsprechend der erlaubten Schwankungsbereiche, wie sie in den Metadaten spezifiziert sind. Ausreißer werden als nächstes entfernt. Das sind Datenpunkte, die höchst irreguläre oder gar unmögliche Eigenschaften aufweisen wie etwa sehr große Gradienten. Aus diesen Gründen entfernte Werte werden ersetzt durch möglichst plausible Messdaten.



Werte werden in ein Koordinatensystem normalisiert, damit jede Variable einen Durchschnitt von Null und eine Standardabweichung von 1 aufweist. Dadurch werden natürliche Skalierungen ebenso entfernt wie auch Inkompatibilitäten von Daten, die es im Zusammenhang mit großem Datenumfang immer mal wieder gibt. Beispielsweise könnten Sie ein Gewicht in Kilogramm oder in Tonnen berechnen. Für unser menschliches Verständnis macht es keinen Unterschied, welche Maßeinheit Sie benutzen, aber für einen Computer sind Angaben in Kilogramm sehr viel größer als in Tonnen, so dass sie im Vergleich zu anderen Größen – etwa Druck – überproportionalen Einfluss nehmen können. Die Normalisierung entfernt solche Besonderheiten und fokussiert sich nur auf die relativen Veränderungen.

7.2. Maschinelles Lernen

Maschinelles Lernen ist – in seiner grundlegenden Definition – ein Weg, empirische Daten so effizient darzustellen, dass sie reproduziert, verallgemeinert, und wiedererkannt werden können oder dass darin ein Muster erkennbar wird. Das Ziel des maschinellen Lernens ist es also, Daten in Form von Gleichungen zusammenzufassen.

Ein sehr einfaches Beispiel für maschinelles Lernen wäre es, eine Reihe von Punkten auf einer geraden Linie darzustellen. Die Gerade ist ein Modell für Daten, die wir gemessen haben. Wir benutzen eine einfache Gleichung, um diese Gerade darzustellen: y, die senkrechten Achse, ist gleich m, die Neigung der Geraden,

multipliziert mit ***x***, der horizontalen Achse, plus der Konstanten ***b***, *dem Schnittpunkt der Geraden mit der senkrechten Achse. Somit stellt* ***y = mx + b*** *eine Zusammenfassung der Daten dar und beinhaltet zwei Parameter, nämlich* ***m*** *und* ***b****. Weil die Messungen für beide,* ***x*** *und* ***y*** *ungenau sind, haben auch die Modell-Parameter* ***m*** *und* ***b*** *ihre Unsicherheiten. Das erlaubt es dem Modell, mit dem ungenauen Charakter der dem wirklichen Leben entnommenen Datensätze umzugehen.*

Die Werte von ***m*** *und* ***b*** *aus einem speziellen Satz empirischer Daten herauszufinden ist, wie gesagt, ein einfaches Beispiel für maschinelles Lernen. Tausende von Datenpunkten werden auf zwei Parameter einer Gleichung reduziert.*

Die Methode, aus den Daten die Parameter eines Modells zu errechnen, nennen wir Algorithmus. Zusätzlich zum Algorithmus, der nötig ist, um aus den Daten die Parameter eines Modells zu berechnen, benötigt das maschinelle Lernen noch einen zweiten Algorithmus, um die Parameter zu aktualisieren, sobald neue Daten verfügbar gemacht werden. Das ist zwar – je nach Modell – nicht immer möglich, stellt aber einen großen Vorzug dar, weil das anfängliche Lernen in der Regel sehr zeitaufwändig ist.

Die Art und Weise, wie eine Maschine lernt, ist nicht dieselbe, wie Menschen lernen, darum ist es im Falle von neuronalen Netzen nützlich, sich einer Analogie zu bedienen.

Das Gehirn ist im Wesentlichen ein Netz von Neuronen, die durch Synapsen verbunden sind. Jedes Neuron und jede Synapse sind – für sich genommen – recht einfache Objekte, aber innerhalb eines Netzes können sie einige erstaunlich komplexe Aktionen durchführen. Nehmen Sie als Beispiel die Fähigkeit, dem Gesicht einer Person einen Namen zuzuordnen und dies mit Erinnerungen an diese Person zu verknüpfen. All das passiert unbewusst in Sekundenschnelle.

Weil der Mensch bei seiner Geburt nicht von vornherein alle Leute kennt, denen er im Laufe seines Lebens begegnen wird, wird das Gehirn darauf trainiert, Erfahrungen zu sammeln und daraus zu lernen, um dann dieses Wissen jederzeit schnell abrufen zu können.

So ähnlich funktioniert also auch ein künstliches neuronales Netz. Man fängt mit einem prototypischen neuronalen Netz an und setzt es dann gewissen Erfahrungen aus. Je mehr das Netz solchen Erfahrungen ausgesetzt wird, umso besser wird es in der Lage sein, diese Erfahrungen korrekt darzustellen und sie auch in Zukunft abzurufen.

Das Erlernen einer geraden Linie, wie wir es oben besprochen haben, ist nur ein einfaches Beispiel für maschinelles Lernen. Neuronale Netze sind jedoch in der Lage, auch extrem komplizierte Datensätze darzustellen.

7.2.1. Neuronales Netz

Ein neuronales Netz besteht aus Knotenpunkten, die miteinander in Verbindung stehen (nicht unähnlich den durch Synapsen verbundenen Neuronen). Die Eingabedaten werden an speziellen Knotenpunkten ins Netz eingespeist und können dann im Netz weitergegeben werden. Jedesmal, wenn eine Dateninformation über

eine Verbindung fährt oder einen Knotenpunkt erreicht, wird sie nach mathematischen Regeln verändert. Zu einem späteren Zeitpunkt verlässt diese Dateninformation das Netz wieder an anderen speziellen Knotenpunkten. Jede Verbindung und jeder Knotenpunkt enthält gewisse Parameter, die genau festlegen, was an dieser Stelle zu tun ist. Auf diese Weise stellt das Netz die Beziehung zwischen den Eingabedaten und den Ausgabedaten dar; es stellt also eine mathematische Funktion dar – ähnlich wie bei der Gleichung einer geraden Linie. Ein neuronales Netz ist eigentlich nicht mehr als eine Gleichung mit Parametern, die durch die Daten bestimmt werden, die das Netz darstellen soll.

Das ganze Aufheben um neuronale Netze basiert eigentlich auf der Tatsache, dass es einen mathematischen Beweis gibt (auf den wir hier nicht näher eingehen wollen), gemäß dem ein neuronales Netz in der Lage ist, einen Datensatz präzise und genau darzustellen, sofern es nur genügend Knotenpunkte enthält und solange die Daten intern konsistent sind. Das gilt selbst dann, wenn die Beziehungen innerhalb des Datensatzes höchst nichtlinear oder zeitabhängig sind.

Was meinen wir mit "intern konsistent"? Es bedeutet, dass die Datenquelle stets denselben Gesetzen gehorcht. Im Falle einer industriellen Produktion werden die Daten durch die Naturgesetze bestimmt. Weil diese sich nicht verändern, ist der Datensatz intern konsistent. Die treibenden Kräfte eines Aktienmarktes bleiben hingegen, um ein gegenteiliges Beispiel zu nennen, nicht konstant, weshalb neuronale Netze solche Datensätze nicht gut wiedergeben können.

Um es zusammenzufassen: Ein neuronales Netz ist eine komplexe mathematische Formel, die in der Lage ist, jeden intern konsistenten Datensatz präzise darzustellen. Der allgemeine Ansatz, dies zu bewerkstelligen, besteht darin, eine Formel mit Parametern zu erstellen und die Werte dieser Parameter mit Hilfe einer Rechenmethode zu bestimmen. Das nennen wir dann maschinelles Lernen.

Es kann nützlich sein, ein neuronales Netz als eine Zusammenfassung von Daten zu verstehen: Eine große Tabelle von Zahlen wird in eine Funktion umgewandelt – so als ob wir von einer wissenschaftlichen Abhandlung einen Abstrakt erstellen, also eine kurze Zusammenfassung. Beachten Sie, dass eine Zusammenfassung nicht mehr Informationen enthalten kann als der ursprüngliche Datensatz; im Gegenteil: Sie enthält weniger Informationen. Allerdings: aufgrund ihrer Kürze kann die Zusammenfassung sehr nützlich sein. Man könnte sagen, dass neuronale Netze auf diesem Weg Informationen in Wissen umwandeln, auch wenn es sich dabei um ein Wissen handelt, dass zum Zweck eines praktischen Nutzens noch gedeutet werden muss.

Die Zusammenfassung von Daten ist zwar schön, aber für konkrete Anwendungen noch nicht ausreichend. Um praktisch nutzbar zu sein, benötigen wir die interpolierenden und extrapolierenden Qualitäten eines Modells. Nehmen wir einmal an, ein Datensatz enthält Messungen für die unabhängige Variable x, deren Wert $x=1$ und $x=2$ sein können; dann hat das Modell eine interpolative Qualität, sofern der Ausgangswert dieses Modells, der zwischen diesen beiden Werten liegt, vernünftig ist; d.h., dass er etwa dem entspricht, was als Messung herausgekommen wäre, wenn wir den Wert tatsächlich gemessen hätten, z.B. bei $x=1,5$. Das ist freilich nur möglich, wenn die ursprünglichen Daten mit genügend hoher Auflösung erhoben

wurden, um alle diese Effekte abzudecken. Das Modell hat eine extrapolative Qualität, wenn der Ausgangswert dieses Modells für die unabhängige Variable außerhalb der beobachteten Spanne liegt und dennoch vernünftig ist, z.B. in diesem Fall bei *x=2,5*. Verhält sich ein Modell vernünftig sowohl in Bezug auf Interpolation wie auch auf Extrapolation, so sprechen wir davon, dass das Modell gut verallgemeinern kann.

Ein neuronales Netz wird im Allgemeinen wie ein Black-Box-Modell eingesetzt. Das heißt, man versteht es nicht als eine Funktion zum Verständnis für menschliche Ingenieure, sondern eher als ein Berechnungswerkzeug, mit dessen Hilfe man sich viele Experimente ersparen oder auf die Bestimmung von Werten durch direkte Beobachtung verzichten kann. Es ist diese Anwendung, welche die beiden oben genannten Aspekte erfordert: Wir benötigen eine Funktion zur Berechnung, und diese ist nur dann nützlich, wenn sie vernünftige Ausgangswerte für die unabhängigen Variablen produziert, die nicht gemessen werden sollen. Es wird das Ergebnis eines virtuellen Experimentes berechnet, das in Wirklichkeit nicht praktisch durchgeführt wird, bei dem wir aber sicher sein können, dass das berechnete Ergebnis mit dem übereinstimmt, was wir beobachten würden, hätten wir das Experiment tatsächlich durchgeführt.

7.2.2. Feed-Forward-Netz

Die meisten neuronalen Netze gehen im Allgemeinen davon aus, dass die Datenpunkte, die zum Training verwendet werden, unabhängige Messungen sind. Das ist ein Schlüsselpunkt beim Modellieren und verdient einige Erläuterungen.

Nehmen wir einmal an, wir haben eine Sammlung digitaler Bilder, die wir in zwei Gruppen unterteilen: Solche, auf denen menschliche Gesichter zu sehen sind, und solche, auf denen irgendetwas anderes zu sehen ist. Neuronale Netze können diese Klassifizierung erlernen, sofern die Sammlung groß genug ist. Die Bilder haben wenig miteinander zu tun; es gibt auch keine Ursache-Wirkungs-Beziehung zwischen irgendwelchen Bildern – jedenfalls keine, die bei der Unterscheidung zwischen menschlichen Gesichtern und anderen Bildern relevant wären.

Aber nehmen wir einmal an, wir wollten Landschaftsbilder in Winterbilder und Sommerbilder klassifizieren und diese Bilder zeigten dieselbe Landschaft, und zwar in kurzer Folge. Nun sind die Bilder nicht mehr unabhängig voneinander, sondern unterliegen einer Ursache-Wirkungs-Beziehung. Das impliziert, dass die Funktionsgleichung ***f(...)***, die wir suchen, ganz anders aufgebaut sein muss. Ihre Genauigkeit wäre viel besser, wenn sie nicht jedes Bild für sich beurteilen würde, sondern wenn sie ein Gedächtnis hätte von den letzten paar Bildern der Serie; denn auf die ersten paar Sommerbilder würden wahrscheinlich weitere Sommerbilder folgen. Bei dieser Version stellen wir fest, dass die Bilder abhängig von einem historischen Verlauf sind, so dass wir vom System ein zeit-orientiertes Gedächtnis erwarten würden, und zwar über eine gewisse Zeitspanne hinweg, die wir irgendwie festlegen müssten.

Wir haben folglich Netz-Modelle, die gut geeignet sind für Datensätze mit unabhängigen Datenpunkten und andere Modelle, die gut geeignet sind für Datensätze, bei denen die Datenpunkte zeitabhängig sind. Netze, die mit unabhängigen Punkten zu tun haben, nennt man Feed-Forward-Netze, und sie

bilden nicht nur den historischen Anfang des Feldes der neuronalen Netze ab, sondern auch ihre vorrangigsten Methoden. Die Netze, die sich mit zeitabhängigen Punkten befassen, werden Rekurrente Netze genannt, die moderner, aber schwieriger anzuwenden sind.

Das populärste neuronale Netz nennt man das Mehrschicht-Perceptron, das die folgende Form annimmt: $y = a_N(W_N \cdot a_{N-1}(W_{N-1} \cdot \ldots a_1(W_1 \cdot x + b_1) \ldots + b_{N-1}) + b_N)$ wobei N für die Anzahl der Schichten steht und W_i für die Gewichts-Matrizen; b_i sind Befangenheits-Vektoren, $tanh(...)$ sind Aktivierungsfunktionen und x steht wie gehabt für die Eingabegrößen.

Die Gewichts-Matrizen und die Befangenheits-Vektoren sind Platzhalter für die Parameter des Modells. Die sogenannte Topologie des Netzes bezieht sich auf die Freiheit, die wir bei der Auswahl der Matrizengrößen und Vektorengrößen haben, und auch auf die Anzahl der Schichten N. Die einzige Einschränkung, die wir haben, ist die Problem-inhärente Größe der Eingabe- und Ausgabe-Vektoren. Haben wir uns einmal auf die Topologie festgelegt, hat das Modell eine spezifische Anzahl von Parametern, die in diesen Matrizen und Vektoren residieren.

Um ein solches Netz zu trainieren, müssen wir erst die Topologie des Netzes sowie die Art der Aktivierungsfunktionen auswählen. Danach haben wir den Wert der Parameter innerhalb der Gewichts-Matrizen und der Befangenheits-Vektoren zu bestimmen. Der erste Schritt ist eine Frage der menschlichen Vorentscheidung und setzt erhebliche Erfahrungen voraus. Selbst nach Jahrzehnten der Forschung auf diesem Gebiet stellen die topologischen Entscheidungen eines neuronalen Netzes immer noch eine nur dunkel verstandene Kunst dar. Es gibt viele Ergebnisse, von denen sich die Praktiker dieser Kunst in ihren Ritualen leiten lassen können, aber das näher zu erläutern würde den Umfang dieses Handbuches sprengen. Der zweite Schritt kann durch standardisierte Trainingsalgorithmen erreicht werden, die wir zuvor erwähnt haben und die wir hier auch nicht weiter behandeln wollen.

Ein Einzelschicht-Perceptron ist somit $y = tanh(Wx + b)$, und war eines der ersten neuronalen Netze, die man untersucht hat. Das Einzelschicht-Perceptron kann ausschließlich linear trennbare Muster darstellen. Es kann bewiesen werden, dass ein Zwei-Schichten-Perceptron praktisch jede gewünschte Funktion mit nahezu beliebiger Genauigkeit darstellen kann, wenn nur die Gewichts-Matrizen und die Befangenheits-Vektoren jeder Schicht groß genug gewählt wurden.

7.2.3. Rekurrente Netze

Die grundlegende Idee eines rekurrenten neuronalen Netzes ist es, die Zukunft abhängig zu machen von der Vergangenheit, und zwar durch eine Form der Funktion, die dem Perceptron-Modell ähnelt. Ein sehr einfaches rekurrentes Modell könnte beispielsweise sein: $x(t+1) = a(W \cdot x(t) + b)$. Beachten Sie bitte sorgsam drei wichtige Eigenschaften dieser Gleichung, die sie von dem zuvor besprochenen Perceptron unterscheiden: (1) Sowohl Eingaben als auch Ausgaben sind jetzt keine Wert-Vektoren mehr, sondern zeitabhängige Funktions-Vektoren; (2) es gibt keine getrennten Eingaben und Ausgaben; vielmehr sind Eingaben und Ausgaben identisch, aber zu unterschiedlichen Zeiten; und (3) weil es sich um zeitabhängige rekurrente (d.h. wiederholbare) Beziehungen handelt, benötigen wir zur Beurteilung gewisse

Anfangsbedingungen, den Zustand $x(0)$.

Das Netz, dass wir einsetzen wollen, verwendet das Konzept eines Reservoirs, was im Wesentlichen nur aus einem Satz Knotenpunkte besteht, die irgendwie miteinander verbunden sind. Die Verbindung zwischen diesen Knotenpunkten wird durch eine Gewichts-Matrix W zum Ausdruck gebracht, die auf bestimmte Weise initialisiert wird. Im Allgemeinen wird sie völlig zufallsbedingt initialisiert, aber von großem Vorteil ist es, wenn sie mit Hilfe einer gewissen Struktur initialisiert wird. Gegenwärtig bedarf es noch einer gewissen Zauberei, die richtige Struktur dafür zu bestimmen, in jedem Fall hängt sie von dem zu lösenden Problem ab. Der augenblickliche Zustand jedes Knotenpunktes im Reservoir wird in einem Vektor $x(t)$ abgespeichert, der ebenfalls zeitabhängig ist. Es ist die Zeitreihe, die wir tatsächliche messen können und die wir voraussagen wollen. Der Eingabeprozess wird gesteuert durch eine Eingabe-Gewichts-Matrix W^{in}, welche die Eingabe-Vektor-Werte jedem gewünschten Neuron im Reservoir zur Verfügung stellt. Die Ausgabe des Reservoirs wird gegeben als Vektor $y(t)$. Der Systemzustand entwickelt sich im Laufe der Zeit gemäß $x(t+1) = f(W \cdot x(t) + W^{in} \cdot u(t+1) + W^{fb} \cdot y(t))$, wobei W^{fb} eine Feedback-Matrix ist, die optional ist für solche Fälle, bei denen wir das Ausgabe-Feedback wieder zurück ins System einspeisen wollen, und wobei $f(...)$ eine Sigmoidfunktion ist, gewöhnlich $tanh(...)$.

Die Ausgabe $y(t)$ wird vom erweiterten Systemzustand $z(t) = [x(t); u(t)]$ berechnet, unter Verwendung von $y(t) = g(W^{out} \cdot z(t))$ wobei W^{out} eine Ausgabe-Gewichts-Matrix darstellt und $g(...)$ eine sigmoidale Aktivierungsfunktion ist, z.B. $tanh(...)$.

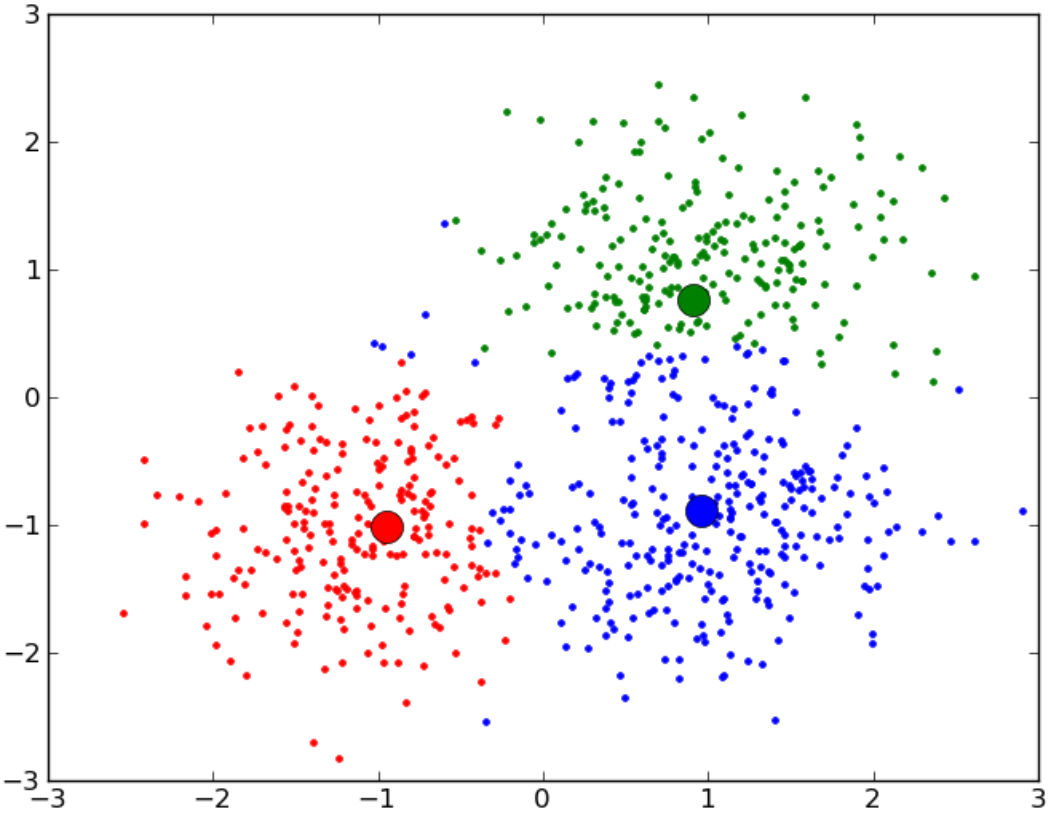
Die Eingabe und Feedback-Matrizen sind Teil der Problemspezifikation und müssen deshalb dem Nutzer zur Verfügung gestellt werden. Die interne Gewichtsmatrix wird irgendwie initialisiert und bleibt danach unberührt. Erfüllt die Matrix W bestimmte komplizierte Bedingungen, so hat das Netz sogenannte Echo-State-Eigenschaft. Das heißt, dass die Vorhersagen irgendwann unabhängig von den Anfangsbedingungen sind. Das ist von entscheidender Bedeutung, weil es dann keine Rolle mehr spielt, zu welcher Zeit die Modellierung beginnt. Solche Netze sind theoretisch in der Lage, nahezu jede Funktion (mit bestimmten technischen Voraussetzungen) darzustellen, wenn sie korrekt angeordnet sind.

7.2.4. Clustering

Quantitative Daten können oft in qualitative Gruppen (Cluster) unterteilt werden. Wenn wir beispielsweise eine Gasturbine betreiben, können wir sie im Leerlauf betreiben, mit Halblast oder mit Volllast. Diese Konzepte sind für einen menschlichen Ingenieur sinnvoll, der weiß, was darunter zu verstehen ist. Auch für einen Computer lässt sich dies definieren, indem strikte Regeln angewendet werden, die auf Messgrößen beruhen, wie etwa die Rotationsrate. Allerdings gehen mit der Erstellung solcher Regeln auch einige Probleme einher: (1) Die Regeln müssen notwendigerweise strikt und einfach sein, (2) ein menschlicher Ingenieur muss sich die Zeit nehmen, diese Regeln zu definieren, (3) die Regeln müssen mit Hilfe irgendeines Systems aufrecht erhalten und regelmäßig auf ihre Genauigkeit überprüft werden, weil sich die Maschinerie oder die Anlage im Laufe ihrer Lebenszeit verändert, (4) die Dateneingabe ist fehleranfällig, wenn wir den Umfang solcher Definitionen berücksichtigen, die für eine industrielle Anlage mit hunderten von

Maschinen und jeweils hunderten oder gar tausenden von Messungen nötig sind.

Alle diese Fragen können aber gelöst werden, wenn wir die vielen Punkte der Betriebsanlage auf der Grundlage einer Datenanalyse zusammen gruppieren, statt dass wir Regeln nach menschlichen Gesichtspunkten entwerfen. Diese Vorgehensweise nennen wir Clustering (Gruppierung). Dabei versuchen wir die Zahl der Cluster und deren Mitglieder mittels einer historischen Sammlung von Beobachtungen zu bestimmen, abhängig von zwei Kriterien: (1) Zwischen den Mitgliedern eines Clusters sollte es nur wenige Unterschiede geben, wohingegen es (2) zwischen Mitgliedern unterschiedlicher Cluster große Unterschiede geben sollte.



Wenn wir es mit quantitativen Daten zu tun haben, können wir einen Cluster im Wesentlichen als eine Kugel verstehen. Mit anderen Worten: Ein Cluster ist ein Punkt im Raum mit einem bestimmten Radius. Jede historische Beobachtung, die innerhalb dieses Radius zu liegen kommt, rechnen wir dem Cluster zu, und alle Punkte außerhalb dieser Kugel rechnen wir ihm nicht zu. Dabei ist klar, dass dieser Ansatz mit überlappenden Kugeln rechnen muss sowie mit Messpunkten, die in keine Kugel passen. Gleichwohl ist dieser Clustering-Ansatz der populärste.

In der Praxis stellen wir die Frage, wie viele Kugeln wir benötigen, wo sich diese befinden und wie groß sie sein müssen, damit sie den beiden Kriterien genügen. Dafür gibt es Methoden, mit deren Hilfe wir praktisch jeden Datensatz in sinnvolle Cluster aufteilen können. Unglücklicherweise bedürfen diese Methoden eines hohen Aufwands an erforderlichen Rechnerzeiten, aber immerhin können sie gut auf die Daten von Industrieanlagen angewandt werden.

Der Vorteil dieser Methoden ist es, dass man mit ihrer Hilfe einen automatisierten Ereignisrahmen (event framework) definieren kann. Wenn dann die Maschinen ihre Aufgabe erledigen, können wir leicht sagen, in welchem Zustand sich die Maschinen

befinden bzw. welche Ereignisse gerade stattfinden. Jedesmal, wenn die Maschine ihren Zustand oder ihr Ereignis verändert, kann das aufgezeichnet werden. Beide Eigenschaften zusammengenommen können dazu verwendet werden, für die Maschine eine Berechnung für äquivalente Betriebsstunden zu betreiben, da die Übergänge von einem Zustand zum nächsten zwar zeitlich rasch erfolgen, aber durchaus einige Lebenszeit der Maschinerie in Anspruch nehmen.

7.2.5. Variablenreduktion

Viele der Daten, die in einer typischen Datenhistorie einer Industrieanlage abgespeichert sind, werden nicht benötigt, um zuverlässige Aussagen über den Zustand der Maschinerie oder über die Optimierung des Verarbeitungsprozesses zu machen. Gewöhnlich sind es weniger als 10% aller Messungen, die tatsächlich gebraucht werden. Und unter diesen Daten gibt es im Allgemeinen noch erhebliche Doppelungen an Informationen.

Jedesmal, wenn wir ein System modellieren wollen, ist es unsere erste Aufgabe, die notwendigen Daten auf intelligente Weise auszuwählen. Dabei besteht allerdings die Gefahr, dass man der Analyse keine ausreichenden Daten zur Verfügung stellt, weshalb wir eher vorsichtig sein müssen und auch solche Messungen einbeziehen sollten, von denen wir nicht sicher sind, ob sie wirklich einen informativen Beitrag leisten können oder nicht. Um zu vermeiden, die Analyse mit unnötigen Daten zu überlasten, müssen wir festlegen, welche Messungen wirklich wichtig sind und welche nicht.

Hinzu kommt, dass manche Messungen mit anderen zusammenhängen, so dass sie zwar Informationen liefern, aber doch mit vergleichsweise wenig Mehrwert. Nehmen Sie beispielsweise Ihr Körpergewicht, Ihre Körpergröße, Ihr Geschlecht und das Ausmaß Ihrer Körperbetätigungen. Allein auf der Basis Ihrer Körpergröße, Ihres Geschlechts und Ihrer Körperbetätigung kann man Ihr Gewicht zwar nicht sicher wissen, aber dazu doch eine vernünftige Vorhersage machen. Wenn wir dann doch noch Ihr Gewicht tatsächlich messen, trägt das zwar zur Gesamtinformation bei, aber doch keineswegs so viel wie die ersten drei Werte, da Ihr Gewicht ja eng mit diesen zusammenhängt. Je nachdem, was man über eine Person herauszufinden wünscht, kann man das mit Hilfe der drei ursprünglichen Werte tun, ohne auf die zusätzliche Information (in diesem Fall: das gemessene Gewicht) unbedingt angewiesen zu sein.

Auf der Grundlage dieser Erkenntnisse können wir zwei unterschiedliche Ideen verfolgen, um die Zahl der Variablen eines Modells zu minimieren. Wir könnten solche Variablen identifizieren, die überhaupt nicht benötigt werden und sie völlig vom Modell ausschließen. Oder wir verwandeln den Datensatz in ein anderes Koordinatensystem mit weniger Dimensionen, bei dem jede neue Dimension allerdings als Kombination der ursprünglichen Dimensionen gedacht wird. Diese zweite Idee basiert auf der soeben beispielhaft erläuterten Erkenntnis, dass manche Messungen zwar zusätzliche Informationen liefern und deshalb nicht gänzlich ausgeschlossen werden sollen, sie aber keine zusätzliche Dimension rechtfertigen.

Diese Idee führt zu der Hauptkomponentenanalyse, die man am besten visuell durch die nachfolgende Grafik erklärt:

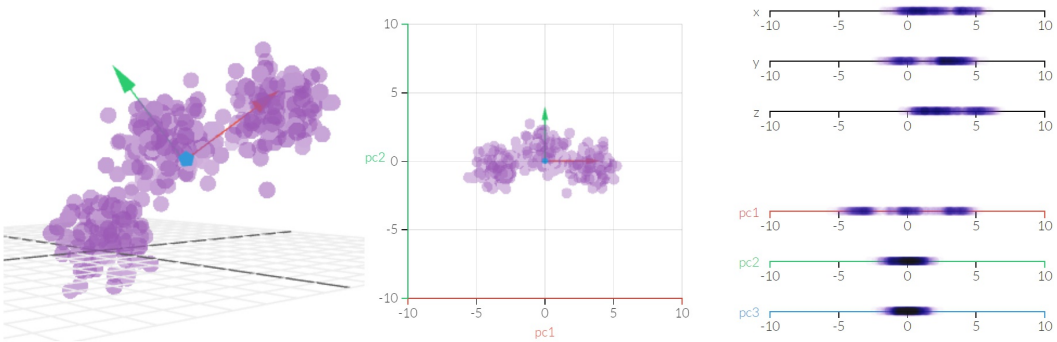


Abb. Links sehen wir den ursprünglichen dreidimensionalen Datensatz, dargestellt im ursprünglichen Koordinatensystem. Es war festzustellen, dass die größten Datenvariationen sich entlang des roten Pfeils im linken Schaubild zeigten. Die zweit- und dritthäufigsten Variationen befinden sich entlang des grünen und des (kaum sichtbaren) blauen Pfeils. Beim mittleren Bild wurden die Daten in einen zweidimensionalen Raum umgewandelt, bei dem der rote und der grüne Pfeil als die neuen Koordinaten angenommen wurden. Wir können leicht erkennen, dass die blaue Richtung nicht genug Informationen beiträgt, um die drei sichtbaren Datengruppen zu trennen, so dass wir diese Dimension ruhig ignorieren können. Das Bild rechts zeigt die Daten entlang der jeweils ursprünglichen, aber nun gedrehten drei Dimensionen. Wir können deutlich sehen, dass die erste Hauptkomponente (der rote Pfeil) ausreicht, um die drei vorhandenen Datengruppen zu trennen. Wir dürfen schlussfolgern, dass die ursprünglich dreidimensionalen Daten durch diese Umwandlungen in einen eindimensionalen Datensatz verschlichtet werden können.

Die Entscheidung darüber, wie viele Hauptkomponenten nötig sind, um die Informationen eines Datensatzes zu erklären, kann mit Hilfe von statistischen Informationen genau getroffen werden. In diesem Fall wird es für den menschlichen Betrachter schnell klar, dass eine Dimension ausreicht, um die drei Klassen von Datenpunkten zu trennen. Wenn es aber darum geht, 10.000 Dimensionen auf 500 zu reduzieren, wird das für den menschlichen Nutzer zwar nicht mehr sichtbar sein, kann aber dennoch präzise ausgeführt werden.

Wir verwenden diese Methode, um die Dimensionen eines Datensatzes zu verringern und dennoch den darin enthaltenen Informationsgehalt aufrecht zu erhalten. Der Vorteil des Ganzen ist, dass wir den Datensatz mit vergleichsweise wenigen Parametern modellieren können, ohne die Genauigkeit zu opfern. Das spart nicht nur Zeit, sondern verbessert auch die Allgemeingültigkeit des Modells; denn es gilt beim maschinellen Lernen als generell wahr, dass ein Modell mit wenigen Parametern ein größeres Potenzial zur allgemeinen Anwendung bietet.

7.3. Optimierung

Es gibt viele Optimierungsansätze. Die meisten davon sind exakte Algorithmen, die definitiv das wahre globale Optimum dessen berechnen, was als Input eingegeben

wurde. Unglücklicherweise haben diese Methoden meist den Zweck, nur ein ganz bestimmtes Problem zu lösen. Es gibt nur eine allgemeine Strategie, die stets das globale Optimum findet. Bei dieser Methode werden alle Optionen aufgelistet und die beste davon ausgewählt. Diese Methode ist allerdings keine realistische, weil bei den meisten Problemen eine Auflistung der Optionen so umfangreich wäre, dass sie praktisch nicht aufgelistet werden können.

Alle anderen Optimierungsmethoden leiten sich von dieser Methode ab und versuchen entweder, die schlechten Optionen auszuklammern, ohne sie überprüft zu haben, oder sie überprüfen nur diejenigen Optionen, die gutes Potenzial haben – auf der Basis von bestimmten Kriterien. Zwei solcher Methoden, die genetischen Algorithmen und das simulierte Annealing, haben sich bei einer Reihe von Anwendungen als erfolgreich erwiesen. Die erst später eingeführten genetischen Algorithmen haben dem simulierten Annealing zeitweise den Rang abgelaufen und sich ins Rampenlicht gedrängt. Allerdings hat ein direkter Vergleich zwischen diesen beiden Ansätzen gezeigt, dass das simulierte Annealing nahezu immer die Oberhand gewinnt, und zwar in allen drei wichtigen Kategorien wie: Umsetzungszeit, Verwendung von Rechnerkapazitäten (Speicher und Zeit) sowie Lösungsqualität. Und deshalb wollen wir uns auf das simulierte Annealing beschränken.

7.3.1. Simuliertes Annealing

Wenn eine Legierung von verschiedenen Metallen angefertigt wird, werden diese Metalle alle über den Schmelzpunkt erhitzt, durchmischt und dann wieder abgekühlt – alles nach einem sorgfältig getakteten Zeitplan. Erlaubt man dem System, sich allzu schnell abzukühlen oder waren die Metalle anfangs nicht ausreichend zusammengeschmolzen, so kann es sein, dass sich vereinzelte Schwachstellen im System herausbilden und die Legierung unbrauchbar wird, weil sie dann zu brüchig ist oder andere Mängel aufweist. Simuliertes Annealing ist ein Optimierungsalgorithmus, der auf demselben Prinzip basiert. Der Vorgang beginnt bei einem willkürlichen Punkt. Dann wird dieser verändert, und es ergibt sich ein benachbarter Punkt. Der neue Punkt wird mit dem alten Punkt verglichen. Stellt er sich als besser heraus, wird er beibehalten. Ist er schlechter, behalten wir ihn trotzdem bei, gemäß einer gewissen Wahrscheinlichkeit, die von einem "Temperatur"-Parameter abhängig ist. Bei sinkender Temperatur sinkt die Wahrscheinlichkeit, eine Veränderung zum Schlechteren zu akzeptieren, so dass schwierige, ansteigende Veränderungen immer seltener akzeptiert werden. Am Ende ist die Lösung so gut, dass weitere Verbesserung kaum noch möglich sind und auch Veränderungen zum Schlechten nur noch selten vorkommen bzw. akzeptiert werden, weil die Temperatur sehr niedrig ist, mit der Folge, dass die Methode konvergiert und so zum Abschluss kommt. Das genaue Ausbuchstabieren der Temperaturwerte und anderer Parameter nennen wir den Abkühlungsplan (cooling schedule). In der Literatur werden viele Abkühlungspläne vorgeschlagen, aber diese wirken sich nur auf Einzelheiten aus, nicht jedoch auf die Gesamtphilosophie der Methode.

Wir können dabei sogleich einige verlockende Eigenschaften erkennen: (1) Es müssen jeweils nur zwei Lösungen abgespeichert werden, (2) wir müssen jeweils nur diese beiden miteinander vergleichen, (3) wir können zeitweilige Verschlechterungen der

Lösung in Kauf nehmen, (4) je mehr Zeit wir dem Algorithmus geben, umso besser wird die Lösung werden, und schließlich: (5) Diese Methode ist sehr allgemein und kann auf praktisch jedes Problem angewandt werden, weil sie vom eigentlichen Problem gar nichts verstehen muss. Der einzige Punkt, an dem das (zu lösende) Problem diese Methode betrifft, ist dort, wo der Punkt verschoben wird und beide Punkte miteinander verglichen werden.

8. Terminologie

In diesem Kapitel bieten wir Definitionen von Ausdrücken an, die in der Oberfläche häufig vorkommen. Diese Termini wurden an anderer Stelle dieses Handbuchs schon kurz beschrieben, werden aber an dieser Stelle ausführlicher behandelt.

8.1. Anlage

Unter einer Anlage verstehen wir die gesamte Einheit, die modelliert werden soll. Sie muss nicht notwendigerweise identisch sein mit einer kompletten Betriebs- oder Werksanlage. Beispielweise kann ein Kraftwerk sich dazu entschließen, nur eine einzelne Turbine zu modellieren. Umgekehrt könnte sich eine Chemieanlage dazu entschließen, Daten des Vorprozesses und des Folgeprozesses in ihr Modell einzubeziehen und damit über die Werksanlage hinaus zu erweitern.

Um eine Anlage von einer anderen Anlage zu unterscheiden, bieten wir vier Datenbank-Felder an, mit deren Hilfe man die Anlage benennen kann: Unternehmen, Bereich, Anlage und Equipment. Ein Beispiel dafür könnte sein: XY Unternehmen, Chemiebereich, Musterstadt, Turbine.

8.2. Tag

Ein Sensor ist ein technisches Instrument, das eine physikalische Gegebenheit in einen numerischen Wert umwandelt, der einem allgemein vereinbarten Maßstab entspricht. Ein Temperatursensor beispielsweise wird die Messung der Temperatur des ihn umgebenen Mediums als einen numerischen Wert auf der Celsius- oder Fahrenheitskala wiedergeben.

Es ist wichtig zu beachten, das Sensoren mehreren Fehlerquellen unterliegen können, die zu berücksichtigen sind, wenn wir den in Frage kommenden numerischen Wert interpretieren.

Zunächst hat jeder Sensor eine Messungenauigkeit, die – vom Hersteller spezifiziert – bedeutet, dass dieselbe physikalische Situation zu Ergebnissen führt, die innerhalb eines bestimmten numerischen Spektrums liegen kann.

Zweitens leidet jeder Sensor im Laufe der Zeit an einem Drift. Mit zunehmendem Alter nehmen Schmutz und andere den Sensor beeinflussende Faktoren zu, und das numerische Messergebnis hat dann die Tendenz zu steigen oder zu fallen. Das macht es nötig, Sensoren von Zeit zu Zeit zu kalibrieren oder auszuwechseln.

Drittens kann dem Sensor aufgrund der Position, an der er angebracht wurde, ein systematischer Fehler anhaften. Beispielsweise wird ein Sensor, der außen an einem großen Behälter angebracht ist, die Temperatur an der Außenhaut des Behälters messen, die aber nicht identisch sein muss mit der Durchschnittstemperatur

innerhalb des Behälters. Zum Zweck des maschinellen Lernens wird dieser Effekt zwar nicht die Qualität des Modells beeinflussen, weil die Fehlerabweichung ja immer gleich bleibt. Aber wenn eine physikalische Interpretation nötig wird, kann diese Abweichung durchaus von Bedeutung sein. Wenn wir beispielsweise wissen, dass die beste Temperatur eigentlich 100 Grad beträgt, wir aber 98 Grad messen, könnten wir versucht sein, noch etwas Wärme hinzuzugeben, was aber nicht nötig ist, wenn die vom Sensor am Behälter gemessene Außentemperatur regelmäßig zwei Grad weniger anzeigt als die Durchschnittstemperatur innerhalb des Behälters eigentlich beträgt.

Eine Messung ist das individuelle numerische Ergebnis eines ganz spezifischen Sensors. Eine Messung beinhaltet stets drei Einzel-Informationen: die Zeit, zu der die Messung vorgenommen wurde, der Tag, an dem (und wofür) es vorgenommen wurde, und der Wert, der dort zum angegebenen Zeitpunkt gemessen wurde.

Ein Tag beinhaltet also die Messungen, die ein Sensor vornimmt, und alle damit verbundenen Daten. Muss der Sensor kalibriert oder ausgewechselt werden, bleibt der Tag derselbe. Alle Daten, die je dazu verzeichnet wurden, werden unter der Bezeichnung dieses speziellen Tags abgespeichert. Die Tags sind also die Währung, auf deren Grundlage alle Prozess-Modelle aufgebaut werden.

Ein Zwilling ist ein anderer Tag, das dieselbe Größe in derselben Weise an etwa der gleichen Örtlichkeit misst. Wir betrachten Zwillinge als wünschenswert, wenn wichtige Messungen vorgenommen werden, für die wir uns keine Fehlfunktionen oder Ausfälle leisten können. Es ist der normale industrielle Standard, drei Sensoren zu installieren und dann den Durchschnittswert von zwei der drei Messungen, die einander am nächsten sind, als das offizielle Ergebnis zu verwenden. Weichen Siblinge erheblich voneinander ab, haben wir zwei potenzielle Probleme: Entweder funktioniert ein Sensor nicht richtig, oder die Maschine, an der die Messung vorgenommen wurde, zeigt an dieser Stelle tatsächlich eine echte Abweichung im Vergleich zu den Stellen, an denen andere Messungen vorgenommen wurden; im letzteren Fall befindet sich die Maschine in einem suboptimalen Gesundheitszustand.

8.3. Minimum und Maximum

Die normale Bandbreite der Messwerte wird definiert, indem für jeden Tag ein Minimalwert und ein Maximalwert festgelegt wird. Diese beiden Werte definieren also den Spielraum, innerhalb dessen sich der Tag vernünftigerweise bewegen dürfte. Sollte der Wert eines Tags irgendwann in der Vergangenheit nicht innerhalb dieser Bandbreite gelegen haben, wird man diesen Betriebszeitraum der Anlage von den Trainingsdaten ausschließen.

Nehmen wir ein Beispiel: Ein chemischer Prozess erfordert eine Temperatur von 250°C. In Wirklichkeit schwankt die Temperatur jedoch zwischen 220 und 270°C. Eine Temperatur oberhalb von 270°C wird ein automatisches Abschalten der Anlage hervorrufen. Wenn die Anlage nicht betriebsbereit ist, wird sich die Temperatur allmählich der Temperatur der Umgebung anpassen, die bis -20°C betragen könnte. Die Bandbreite sollte deshalb zwischen -20°C und 270°C liegen, denn auch eine vorübergehend stillgelegte Anlage wird als ein valider Zustand betrachtet.

Nehmen wir ein zweites Beispiel: Ein Ventil kann entweder offen oder geschlossen sein. Sein Zustand wird in Prozent gemessen. Streng genommen, ginge die Bandbreite von 0 bis 100%, also von völlig geschlossen bis völlig offen. Allerdings kann es sein, dass der Sensor, der den Ventilzustand misst, manchmal Werte von unter 0 oder von etwas über 100 ausgeben. Mit diesem Problem können wir auf zwei unterschiedliche Weisen verfahren: Entweder definieren wir die Bandbreite zwischen -5% und +105%. Oder wir definieren die Bandbreite exakt zwischen 0 und 100% mit einer zusätzlichen Korrekturregel, nach der Werte unterhalb von 0 als 0 gewertet werden und Werte oberhalb von 100 als 100 gewertet werden.

Nachdem Sie die Metadaten für die Anlage eingerichtet haben, wird die Analyse einen Plausibilitätscheck durchführen. Diese Überprüfung wird u.a. berechnen, wie viele der historischen Betriebspunkte der Anlage auf der Grundlage der Bandbreiten aller Tags ignoriert werden. Diese Informationen können Sie dazu verwenden, Ihre Eingaben zu korrigieren. Zusätzlich dazu hat jeder Tag eine Seite mit Eigenschaften, auf dem sich ein Histogramm befindet, auf dem die Verteilung der Werte aufgeführt ist. Das können Sie dazu verwenden, Erkenntnisse darüber zu gewinnen, welche Werte ein Tag meistens hat.

Häufig stellen wir fest, dass einige Bandbreiten so definiert wurden, dass sämtliche (oder nahezu alle) Betriebspunkte von der Modellierung ausgeschlossen wurden. Dieser Fall tritt gewöhnlich dann ein, wenn eine Bandbreite eine theoretisch erwünschte ist, die in der Wirklichkeit aber nur selten erreicht wird. Es kann auch sein, dass diverse Fehlerquellen dazu beitragen, dass die Bandbreite nicht eingehalten wird – etwa wenn der Tag in einer anderen Einheit gemessen wird als man eigentlich erwarten würde. Das kann passieren, wenn der Tag in Tonnen berechnet wird, die Bandbreite aber in Kilogramms definiert wurde.

8.4. Statische Alarmbegrenzungen

Zusätzlich zu den dynamischen Grenzwerte, die IHM berechnet, werden von IHM auch normale statische Grenzwerte unterstützt. Der grüne Bereich sollte normale Betriebswerte umfassen, der gelbe Bereich die gefährlichen Werte, der orangefarbene Bereich die kritischen und der rote Bereich steht für Schadensfälle. Sie können jeden bzw. alle der farbigen Bereiche verwenden, ganz wie Sie wollen. Durch IHM werden Sie alarmiert, wenn ein Messwert die Grenze von einer Farbe zur nächstgefährlicheren überschreitet. Diese farbigen Bereiche können für jeden Tag definiert werden, indem ihre jeweiligen Grenzwerte spezifiziert werden. Bitte schauen Sie sich dazu die Grafiken an.

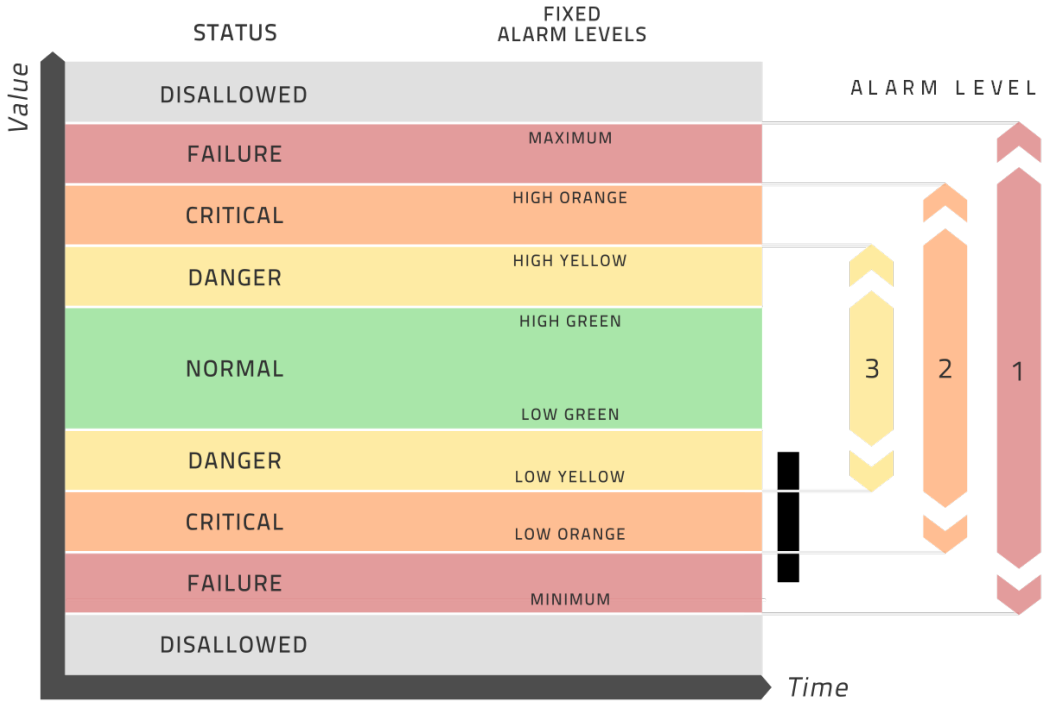
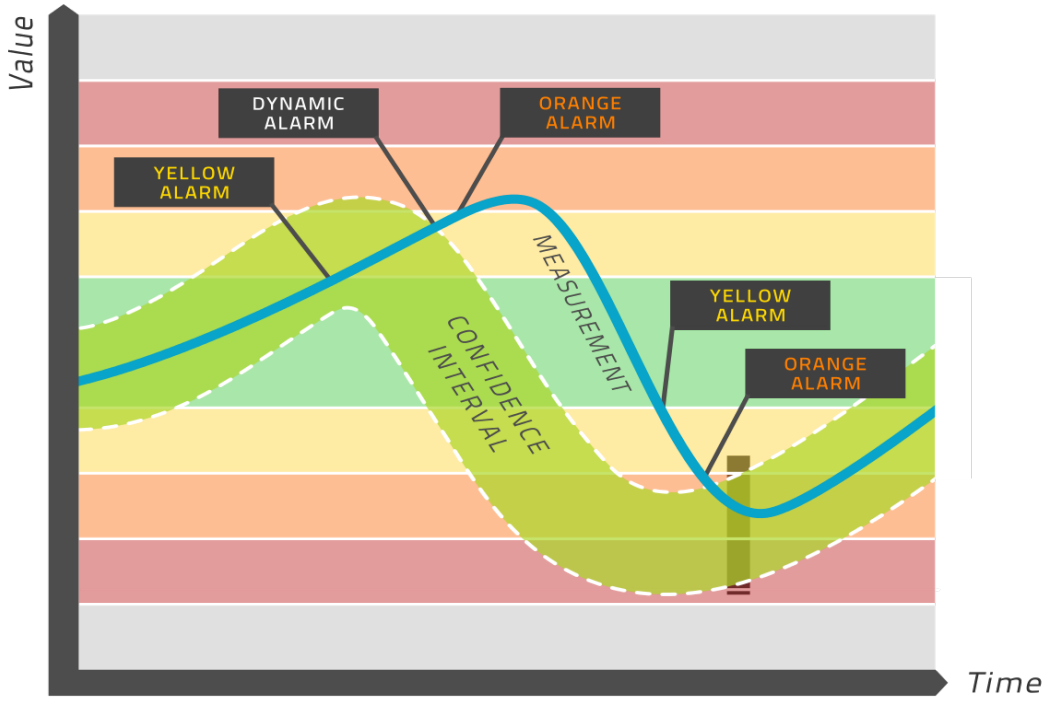


Abb. Überblick über alle Begrenzungen für ein jeweiligen Tag. Der Wert sollte sich innerhalb des grünen Bereiches bewegen. Ein Wert ist gefährlich im gelben Bereich, kritisch im orangefarbenen, schadhaft im roten und unzulässig im grauen Bereich.



8.5. Kontrollierbarkeit

Die Kontrollierbarkeit eines Tags enthält drei Möglichkeiten. Ein Tag ist entweder kontrollierbar, unkontrollierbar oder halb-kontrollierbar. Die Wahl, welcher dieser drei Kategorien ein Tag zugeordnet wird, ist wahrscheinlich die wichtigste Entscheidung, die Sie im Hinblick auf das mathematische Modell Ihrer Anlage zu treffen haben

zumindest für die Anwendung von APO.

Ein Tag ist kontrollierbar, wenn es vom Anlagenfahrer von Hand verstellt werden kann. Das geschieht gewöhnlich, indem der Sollwert eingestellt wird. Bei einer Anlage wird es meist eine ganze Reihe von eingeschalteten PID-Reglern geben, die verschiedene Werte automatisch auf der Basis von anderen Werten steuern. Obwohl ein Betreiber die theoretische Möglichkeit besitzt, einen PID-Regler abzuschalten und diesen kontrollierten Wert von Hand zu betreiben, dürfte das in der Regel nicht erwünscht sein und auch nicht praktiziert werden. Somit ist dieser Wert nur theoretisch kontrollierbar, nicht jedoch tatsächlich. Wir sollten nur solche Werte als kontrollierbar kennzeichnen, die tatsächlich vom Anlagenfahrer modifiziert werden. Es könnte sich dabei als nützlich erweisen, sich einmal mit dem Fahrer im Kontrollraum zu unterhalten, um dann eine Liste der Tags zu erstellen, die von Hand eingestellt werden.

Nur solche Tags sollten als kontrollierbar gekennzeichnet werden, die sich potenziell auf den Leistungsaspekt der Anlage auswirken, die Sie zu optimieren wünschen. Beispielsweise wird es einige wenige Tags geben, die für Notmaßnahmen relevant sind oder zum Hoch- und Herunterfahren benötigt werden und deshalb von Hand bedient werden. Aber zum Zweck der Optimierung des normalen Betriebs müssen wir solche Tags nicht unbedingt modifizieren.

Jeder kontrollierbare Tag kann in einer Empfehlung für den Bediener auftauchen, die den Zustand der Anlage verändern soll. Es wird im Allgemeinen nicht nötig sein, zu einer bestimmten Zeit die Werte aller kontrollierbaren Tags zu verändern, aber irgendwann kann jeder kontrollierbare Tag in irgendeiner Empfehlung erscheinen. Wenn Sie wissen möchten, wie oft ein Tag in einer Empfehlung auftaucht, können Sie das in dem Bericht nachlesen, der nach der Modellierung verfügbar gemacht wird.

Ein Tag ist unkontrollierbar, wenn es vom Anlagenfahrer in keinster Weise verändert werden kann. Es handelt sich um Tags, die den Status der Außenwelt messen. Aspekte des Wetters wie Temperatur, Luftfeuchtigkeit oder Luftdruck sind Beispiele dafür. Auch die Qualität der Rohmaterialien ist gewöhnlich ein wichtiger (unkontrollierbarer) Aspekt. Auch ist die Menge des Produktes, welche die Anlage herstellen soll, gewöhnlich ein Begrenzungsfaktor, der von den Kunden der Anlage vorgeschrieben wird.

Ein Anlagenbediener kann offensichtlich viele Dinge nicht kontrollieren. Wir sollten allerdings nur solche Aspekte der Außenwelt berücksichtigen, die tatsächlich einen Einfluss auf die Leistung der Anlage haben. Wenn Sie der Überzeugung sind, dass die Außentemperatur den Prozess beeinflusst (was gewöhnlich der Fall ist), dann berücksichtigen sie diese. Es gibt einige Prozesse, die empfindlich auf Luftfeuchtigkeit reagieren, aber in den meisten Fällen ist das nicht der Fall. Obwohl Sie die Luftfeuchtigkeit in Ihrer Wetterstation messen dürften, sollten Sie sie nur dann für das Modell als unkontrollierbar berücksichtigen, wenn Sie glauben, dass dies den Prozess beeinflusst.

Ein unkontrollierbarer Tag ist eine sogenannte Randbedingung der Anlage. Das mathematische Modell, das konstruiert werden soll, wird nach einem betrieblichen Messpunkt Ausschau halten, der besser als der aktuelle Punkt ist, aber von denselben Werten aller unkontrollierbarer Tags ausgeht. Jeder unkontrollierbare Tag

schränkt die Freiheit des Optimierers ein und reduziert mögliche Verbesserungen, die der Optimierer zu erzielen wünscht. Somit hat die Auswahl der unkontrollierbaren Tags erheblichen Einfluss auf das Modell.

Jeder Tag, das weder kontrollierbar noch unkontrollierbar ist, wird als semi-kontrollierbar bezeichnet. Das sind all jene Tags, die nicht direkt, sondern nur indirekt verändert werden können, und zwar durch die kontrollierbaren Tags. Die große Mehrheit der verfügbaren Tags fallen in diese Kategorie welche die Voreinstellung für alle Tags ist. Wenn Sie Ihre Tagliste erstellen, raten wir nachdrücklich dazu, zunächst alle Tags als semi-kontrollierbar zu betrachten und dann nur solche Tags als kontrollierbar bzw. unkontrollierbar zu kennzeichnen, die Sie als solche eindeutig identifizieren können. Auf diese Weise werden Sie am Ende mit einer Liste von relativ wenigen und sorgfältig ausgewählten Tags aufwarten, die modifiziert werden können, und solchen, die Randbedingungen für die Anlage darstellen.

8.6. Korrelation

Wenn zwei unterschiedliche Tags in Beziehung zueinander stehen, nennen wir das Korrelation. Wenn wir von Korrelation sprechen, meinen wir damit typischerweise, dass die Beziehung linear ist. Das heißt, dass wenn eines dieser Tags variiert, sich auch der andere Tag proportional zum ersten verändert; mit anderen Worten: Die Beziehung wird durch die Konstante der Proportionalität zum Ausdruck gebracht.

Ein Beispiel für zwei korrelierte Tags ist die Temperatur und der Druck einer Substanz in einem Behälter. Solange der Behälter sein Volumen beibehält und keine Substanz dem Behälter zugeführt wird oder von ihm abfließt, stehen Druck und Temperatur in direkter Beziehung zueinander: Steigt das eine, steigt auch das andere. Weil diese beiden Tags somit streng verbunden sind, gelten sie in hohem Maße als korreliert.

Ein anderes Beispiel von zwei korrelierten Tags ist die Rotationsrate und die Schwingungsamplitude. Allerdings hängt diese Beziehung noch von einer Reihe anderer Tags ab. Während diese zwei Tags gemeinsam anwachsen oder abnehmen, kann es aber auch sein, dass sich ein Tag ohne das andere verändert, und zwar aufgrund von anderen Bedingungen. Diese Korrelation ist somit schwächer.

Die Stärke der Korrelation wird gemessen durch den Korrelationskoeffizienten. Das ist eine Zahl zwischen -1 und 1. Ist der Koeffizient 1, werden sich die zwei Elemente exakt gleich verändern. Ist der Koeffizient -1, gibt es eine umgekehrte Korrelation, so dass, wenn ein Tag ansteigt, das andere sinkt. Ist der Koeffizient irgendwo dazwischen, ist die Beziehung zwischen beiden schwächer. Unter Praxisbedingungen werden zwei Tags allerdings nie perfekt korreliert sein, weil es bei jeder Messung stets eine natürliche willkürliche Variabilität gibt, zumindest eine Messunsicherheit.

Im Allgemeinen ist es nicht möglich, einen einzelnen Korrelationskoeffizienten in präziser Weise zu interpretieren, weil eine Korrelation stets vom Kontext abhängt. Was bedeutet es beispielsweise, wenn zwei Tags mit einem Koeffizienten von 0,758 korrelieren? Für sich genommen, heißt das noch nicht viel. Erst wenn wir die Korrelationskoeffizienten miteinander vergleichen, gewinnen sie an Bedeutung. Wenn zwei Tags den Koeffizienten von 0,758 haben und zwei andere den Koeffizienten von 0,123, so können wir sagen, dass das erste Paar in einer sehr viel engeren Beziehung zueinander steht als das zweite. Welcher Korrelationskoeffizient

hoch oder niedrig genannt wird, ist eine subjektive Entscheidung des Beobachters.

Bei industriellen Datensätzen beobachten wir manchmal Korrelationskoeffizienten von über 0,95 zwischen verbundenen Tags, und häufig erhalten wir Koeffizienten von über 0,8.

Wir können eine Korrelationsmatrix für eine ganze Kollektion von Tags berechnen, wobei wir die Korrelationskoeffizienten zwischen jedem Tagpaar berechnen. Diese Matrix kann dann analysiert werden, etwa um zu die Frage zu stellen: Welche Tags stehen zu einem bestimmten Tag in besonders enger Verbindung? Wir könnten solche Tags auswählen, die am engsten mit diesem einen uns besonders interessierenden Tag korreliert sind. Wir könnten uns zudem die Korrelationen zwischen den Tags anschauen und dann genau diejenigen eliminieren, bei denen wir eine hohe Korrelation erkennen, weil diese Tags nahezu dieselben Informationen enthalten.

Die Korrelationsmatrix kann somit die Grundlage dafür sein, Tags in Cluster zu gruppieren. Im Allgemeinen sind eng korrelierte Tags physisch miteinander verbunden und gehören deshalb zum selben Cluster.

8.7. Sensitivität

Das Konzept der Sensitivität steht in einer engen Beziehung zur Korrelation, weil wir wissen, wie eng zwei Tags miteinander zusammenhängen. Die Korrelation wird gewöhnlich im Zusammenhang mit einer linearen Beziehung verwendet, während die Sensitivität eher in einem allgemeinen Kontext, sei er linear oder nicht-linear, verwendet wird. Insbesondere wird die Sensitivität im Zusammenhang mit einem mathematischen Modell verwendet.

Wenn wir etwa eine Formel haben wie $y = f(a, b)$, so könnten wir fragen: Um wie viel wird sich y verändern, wenn a oder b sich ändern? Wenn es sich zeigt, dass y sich stark verändert, wenn a sich verändert, aber kaum verändert, wenn b sich ändert, so dürfte es naheliegen, dass wir die Gleichung vereinfachen können, wenn wir b aus der Formel entfernen.

Die Sensitivität spielt also eine wichtige Rolle bei dem Versuch, Modelle zu vereinfachen oder die Zahl der Variablen in einem Modell zu reduzieren. Sie spielt auch eine wichtige Rolle, wenn wir abschätzen wollen, wie viel Einfluss wir auf y ausüben können. Wenn etwa alle Tags, die eine hohe Sensitivität gegenüber y haben, nicht kontrolliert werden können, wird man auf y kaum einen Einfluss ausüben können. In diesem Fall wird y von seiner Umgebung bestimmt. Wenn aber, andererseits, ein oder mehrere Tags, die eine hohe Sensitivität haben, kontrolliert werden können, so kann das Modell für eine erweiterte Prozesskontrolle herangezogen werden.

Die Modelle, die von IHM und ISS erstellt werden, zeigen die Sensitivität der unabhängigen Variablen an und bieten dem Nutzer somit Erkenntnisse darüber an, wie er das Modell vereinfachen kann, oder darüber, wie nützlich das Modell überhaupt ist.