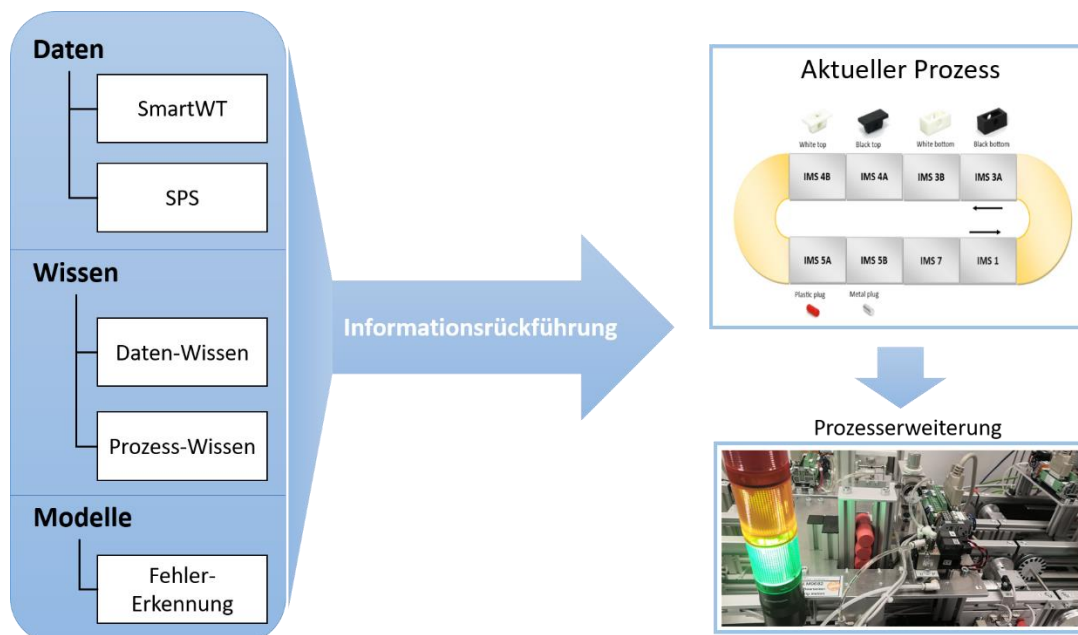


Data Science in der Produktion

Übung 3: Prozessanpassung

Autor: Bux, Tobias

Mail: tobias.bux@isw.uni-stuttgart.de



Übung 1

Daten sammeln

Übung 2

Daten modellieren

Übung 3

Prozessanpassung



Versionsverwaltung

Version	Autor	Datum	Status	Änderung
1.0	Tobias Bux	15.10.2019	akzeptiert	

Inhalt

Abbildungsverzeichnis.....	5
Abkürzungen und Formelzeichen.....	6
1. Organisatorisches	7
1.1. Allgemeine Regeln zur Durchführung des Praktikums	7
1.2. Zum Aufbau des SAL	7
1.3. Allgemeine Sicherheitsrichtlinien und praktikumsspezifische Sicherheitsrichtlinien	8
2. Einleitung.....	9
2.1. Lernziel der Übung	9
2.2. Vorausgesetzte Vorkenntnisse	9
2.3. Verwendete Hardware	9
2.4. Verwendete Software	9
3. Theoretische Grundlagen	10
3.1. Einführung zum Thema allgemein.....	10
3.1.1. OPC UA	10
3.1.2. UaExpert	10
3.1.3. MQTT	11
3.1.4. Python	12
3.1.5. Maschinelles Lernen.....	12
3.2. Anwendungsbezogene Grundlagen für die Aufgaben	13
3.2.1. Aufbau der Montagelinie	13
3.2.2. MQTT-Parametrisierung.....	13
3.2.3. OPC UA-Parametrisierung	14
3.2.4. Machine Learning mit mglearn	15
3.2.5. CommunicationManager.....	16
3.2.6. MachineLearningManager	17
4. Aufbau und Ablauf des Praktikums	18
4.1. Ausgangszustand	18
4.2. Ablauf	18
4.3. Ziel-/Endzustand.....	19
5. Vorbereitungsaufgaben.....	20
5.1. Aufgabe V1	20
5.2. Aufgabe V2	20
6. Versuchsdurchführung	21
6.1. Aufgabe 1 – Vorbereitung	21
6.1.1. Lernziel der Aufgabe.....	21



6.1.2.	Beschreibung und Ablauf der Aufgabe.....	21
6.1.3.	Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis	22
6.2.	Aufgabe 2 – Diskussionsrunde Prozesseingriff.....	23
6.2.1.	Lernziel der Aufgabe.....	23
6.2.2.	Beschreibung und Ablauf der Aufgabe.....	23
6.2.3.	Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis	23
6.3.	Aufgabe 3 – Datenerfassung	24
6.3.1.	Lernziel der Aufgabe.....	24
6.3.2.	Beschreibung und Ablauf der Aufgabe.....	24
6.3.3.	Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis	24
6.4.	Aufgabe 4 – Informationsrückführung	25
6.4.1.	Lernziel der Aufgabe.....	25
6.4.2.	Beschreibung und Ablauf der Aufgabe.....	25
6.4.3.	Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis	26
6.5.	Aufgabe 5 – Informationsrückführung (Bonusteil).....	27
6.5.1.	Lernziel der Aufgabe.....	27
6.5.2.	Beschreibung und Ablauf der Aufgabe.....	27
6.5.3.	Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis	27
7.	Hinweis auf nötiges Vorgehen zum Beenden der Versuchsdurchführung	28



Abbildungsverzeichnis

Abbildung 1: Ausschnitt aus dem Knotenmodell der SPS der Lukas Nülle Anlage	10
Abbildung 2: Reihenfolge in UaExpert zum Auslesen von Variablen aus einem OPC UA-Server	11
Abbildung 3: Montagelinie (Lukas Nülle Anlage)	13
Abbildung 4 Übersicht des Übungsvorhabens	18

Abkürzungen und Formelzeichen

- MQTT - Message Queuing Telemetry Transport
- OPC UA - Open Platform Communications Unified Architecture
- ISW - Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen
- CRISP - Cross-industry standard process
- SPS - Speicherprogrammierbare Steuerung
- LN – Lukas Nuelle
- SAL – Studentisches Applikationslabor

1. Organisatorisches

1.1. Allgemeine Regeln zur Durchführung des Praktikums

Studentische Praktika am ISW sind dafür gedacht theoretisches Wissen durch praktische Anwendung zu vertiefen, problemorientiertes, methodisches Vorgehen zu erlernen und Einblicke in ggf. neue praktische Tätigkeitsfelder zu erhalten. Damit alle Studierenden einen möglichst großen Nutzen aus den Praktika ziehen können, wird die Einhaltung folgender Regeln erwartet:

1. Die Umdrucke zu den Praktika sind selbstständig auf Papier oder elektronisch zu beschaffen und zur Übung mitzubringen. Es gilt der neueste, sich auf ILIAS befindende, Stand.
2. Die „Richtlinie zu Verhalten, Sicherheit und Arbeitsschutz am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)“, welche am Ende der Praktikumsskripte abgedruckt sind, sind vor Beginn der Praktika gewissenhaft zu lesen. Zu Beginn jedes Versuchs muss der Studierende dem Betreuer schriftlich bestätigen, dass die Richtlinien gelesen und verstanden wurden. Fragen zu den Richtlinien müssen unbedingt vor Beginn des Versuchs geklärt werden.
3. Jeder Versuch ist vorzubereiten. Darunter fällt zum einen das detaillierte Studium der Umdrucke und ggf. weiterer Informationen, auf die explizit im Umdruck hingewiesen wird. Zum anderen sind die Vorbereitungsaufgaben SCHRIFTLICH auszuführen und zum Praktikum mitzubringen. Sollten die Aufgaben für Sie im Vorfeld nicht lösbar sein, suchen Sie sich eigenständig und rechtzeitig Hilfe bei Ihren Kommilitonen, Ihrem Übungsbetreuer oder bei Herrn Seyfarth. Sollten Sie nicht alle Voraussetzungen, welche für die Durchführung des Versuchs notwendig sind (siehe Kapitel 2.2), erfüllen, müssen Sie diese eigen- und selbstständig aufholen. Die Vorbereitungszeit für diesen Praktikumsversuch kann die angesetzten 8 Stunden dann evtl. übersteigen.
4. Ihre Lösungen der schriftlichen Vorbereitungsaufgaben werden zu Beginn des Versuches von den Praktikumsbetreuenden gesichtet. Ohne weitgehend korrekte Bearbeitung der schriftlichen Vorbereitungsaufgaben ist die Teilnahme am Versuch nicht möglich. Der Versuch wird dann als nicht bestanden gewertet.
5. Weiterhin findet vor Beginn des Versuchs ein SCHRIFTLICHES Antestat statt. Das Antestat besteht aus 10 Multiple Choice Fragen zum Inhalt des Praktikumsskripts. Um das Testat zu bestehen müssen 60% der Aufgaben richtig beantwortet werden. Nur nach bestandenem Antestat nimmt der Studierende an dem jeweiligen Praktikumsversuch teil. Ist das Antestat nicht bestanden, wird der Studierende von diesem Versuch ausgeschlossen und dieser Versuch als nicht bestanden gewertet.
6. Es wird eine aktive Mitarbeit im Praktikum erwartet. Nicht aktive Mitarbeit (z.B. Arbeitsverweigerung, einfaches „Absitzen“ der Zeit) führt zum Nichtbestehen des Praktikumsversuchs.
7. Die Praktikumsversuche finden im studentischen Applikationslabor (SAL) am ISW statt.

1.2. Zum Aufbau des SAL

An insgesamt 20 Laborarbeitsplätzen besteht die Möglichkeit durch modulare Gestaltung der Arbeitsplätze an Themen für Forschung und Lehre, Schulungen, Praxisübungen sowie Versuchen individuelle Versuchsaufbauten zu realisieren. An das SAL ist ein separater IT- und Serverraum mit Kühlung, Strom etc. für bis zu vier Serverracks angeschlossen. Dabei sind mehrere physische Netzwerke

zur Nachbildung realer Produktionsszenarien implementiert und ermöglichen durch die professionelle Ausstattung die Realisierung von Szenarien zur Elektro-, Elektronik- und Softwareentwicklung.

1.3. Allgemeine Sicherheitsrichtlinien und praktikumsspezifische Sicherheitsrichtlinien

Die Praktikumsversuche werden an Versuchsständen im Studentischen Applikationslabor des ISW (4. OG, Seidenstraße 36, 70174 Stuttgart) durchgeführt. In den Versuchsständen sind zumeist industrielle Steuerungstechnik und gebräuchliche elektro-mechanische Komponenten für Werkzeugmaschinen und Fertigungseinrichtungen verbaut. Daher können zum einen hohe Spannungen und Ströme an den Versuchsständen auftreten. Zum anderen können Komponenten der Versuchsstände sehr schnell beschleunigt werden, hohe Kräfte erzeugen oder an Bearbeitungsprozessen beteiligt sein. Um einen sicheren und reibungslosen Ablauf der Praktikumsversuche so weit wie möglich zu garantieren, müssen die Praktikums Teilnehmer die allgemeinen „Richtlinien zu Verhalten und Arbeitsschutz am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen“ unterschrieben und dem Praktikumsbetreuer zu Beginn der Übung aushändigen. Fragen zu den allgemeinen Richtlinien beantworten die Betreuenden gerne vor Übungsbeginn. Die Richtlinie befindet sich am Ende des Dokuments.



Ein Verstoß gegen die allgemeinen Richtlinien zu Verhalten, Sicherheit und Arbeitsschutz am ISW führt zum Ausschluss von den Praktika.

2. Einleitung

Data Science gewinnt in der Produktionstechnik eine immer größere Bedeutung, da der Zugang zu Daten durch eine gestiegene Konnektivität von u.a. Produktionsanlagen einfacher wird und die Performance der Computer in den letzten Jahren gestiegen ist. Für ein strukturiertes Vorgehen im Bereich des Data Science wurde in der Vorlesung das CRISP-Modell (Cross-industry standard process for data mining) vorgestellt. Dieses Modell soll anhand des in der Vorlesung vorgestellten Anwendungsfalls (Block C, CRISP Phase 1) in den Übungen praktisch eingeführt, verstanden und bearbeitet werden.

In den bisher durchgeführten Übungen 1 und 2 wurden Daten der Lukas Nuelle Anlage im SAL erfasst, aufbereitet und anschließend für das Training unterschiedlicher Modelle verwendet. Somit wurden alle Vorbereitungen getroffen, um in dieser Übung den tatsächlichen Prozess zu überwachen und mit Hilfe der trainierten Modelle, Aussagen über potentielle Fehler in der Produktion zu treffen.

2.1. Lernziel der Übung

Nach der Durchführung der Übung kennen Sie mehrere Möglichkeiten Livedaten einer Produktionsanlage zu erfassen und sofort zu verarbeiten. Weiter verstehen Sie, welche Herausforderungen die Verarbeitung von Livedaten mit sich bringt. Sie sind in der Lage ein trainiertes Modell mit Informationen zu versorgen und die Ergebnisse der Voraussage des Modells zu deuten.

Neben diesen technischen Details erfahren Sie welche Komplexität die Änderung eines existierenden Prozesses mit sich bringt. Sie verstehen, warum theoretische Problemlösungen nicht immer ohne Anpassungen in einen praktischen Prozess überführt werden können.

2.2. Vorausgesetzte Vorkenntnisse

Neben dem Durcharbeiten dieses Skripts sollten Sie sich mit der Programmiersprache Python vertraut machen. Für die Durchführung der Übung sind fortgeschrittene Kenntnisse in Python notwendig.

2.3. Verwendete Hardware

- Montagelinie (Lukas Nülle Anlage)
<https://www.lucas-nuelle.de/217/apg/10362/CSF-1-Industrie-40-mit-IMS.htm>
- Werkstückträger
- Windows-PC

2.4. Verwendete Software

- Python (Programmierungsumgebung)
<https://www.python.org/>
- OPC UA (Kommunikationsprotokoll und Datenformat)
- python-opcua (OPC UA Client Implementierung für Python)
<https://github.com/FreeOpcUa/python-opcua>
- MQTT (Kommunikationsprotokoll)
- Paho-mqtt 1.4.0 (MQTT Client Implementierung für Python)
<https://pypi.org/project/paho-mqtt/>
- UA Expert (Grafische OPC UA Client-Anwendung)

3. Theoretische Grundlagen

In diesem Kapitel wird zunächst in Abschnitt 3.1 die Theorie zu den grundlegenden Elementen im Praktikum erklärt, worauf hin in Abschnitt 3.2 auf die anwendungsbezogenen Grundlagen zur Durchführung der Aufgaben eingegangen wird. Die Einbettung der Grundlagen in das Szenario aus der Einleitung in Kapitel 2 erfolgt bei der Versuchsdurchführung in Kapitel 6.

Die Informationen in Kapitel 3.1.1, 3.1.3, 3.1.4 sowie 3.1.5 wurden bereits in den Übungen 1 und 2 vermittelt. Sollten Sie diese Übungen besucht haben und die Informationen noch präsent sein, können Sie die Abschnitte überspringen.

3.1. Einführung zum Thema allgemein

3.1.1. OPC UA

OPC UA (Open Platform Communications Unified Architecture) ist ein Standard, der ein Datenmodell sowie ein Kommunikationsprotokoll zum plattform- und herstellerunabhängigen Austausch von industriellen Daten beschreibt. Der Datenaustausch wird durch eine Server-Client Kommunikation realisiert. Das heißt der Server stellt die Daten bereit, auf den die Clients zugreifen können und die Daten lesen und schreiben können. Die Daten werden in einem hierarchischen, knotenorientierten Datenmodell im Server repräsentiert (Abbildung 1). OPC UA hat sich in der Produktionstechnik als Quasistandard etabliert. Alle neueren Geräte der Feldebene, wie z.B. eine SPS (Speicherprogrammierbare Steuerung), unterstützen OPC UA. [1]

Es gibt eine Vielzahl von Clientimplementierungen in verschiedenen Programmiersprachen, wie Python, C und Java. Dadurch ist die Verarbeitung von Daten mit OPC UA relativ einfach umzusetzen.

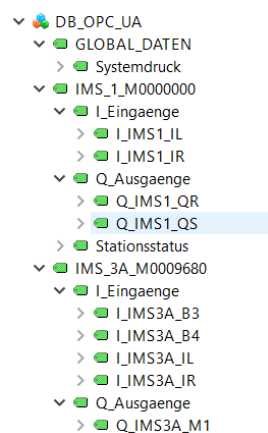


Abbildung 1: Ausschnitt aus dem Knotenmodell der SPS der Lukas Nülle Anlage

3.1.2. UaExpert

UaExpert ist ein OPC UA-Client der Firma Unified Automation, der nach einer Registrierung kostenlos heruntergeladen werden kann. Mittels UaExpert können Daten aus einem OPC UA-Server angezeigt und verändert werden. Die einzelnen Schritte zum Auslesen einer Variable sind in Abbildung 2 abgebildet. Zuerst muss der Server in UaExpert angelegt werden (1). Hierfür wird im Fenster die URL des Servers eingetragen, die sich aus dem Protokoll, dem Hostname/der Ip und dem Port zusammensetzt. Z.B. `opc.tcp://192.168.128.2:4840`, der OPC UA-Server ist in diesem Fall mit dem Protokoll `opc.tcp` erreichbar. Er befindet sich unter der IP 192.168.128.2 und ist unter dem Port 4840 erreichbar (2). Wurde der Server angelegt, muss er ausgewählt (3) und anschließend verbunden werden (4). Ist der Verbindungsaufbau erfolgreich, wird der Inhalt des Servers im Adressraum-Fenster unten links (5) angezeigt. Zum Auslesen einzelner Variablen, müssen diese aus dem Adressraum in den DataAccessView in der Mitte (6) gezogen werden.

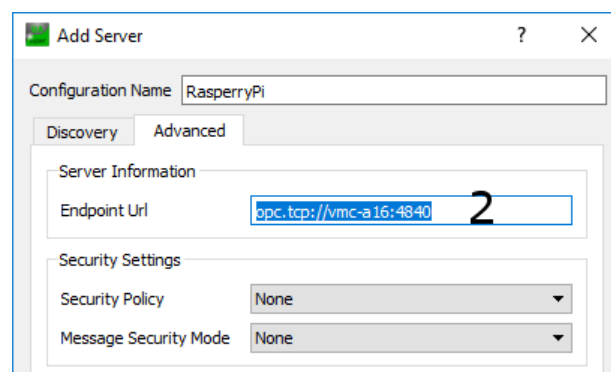
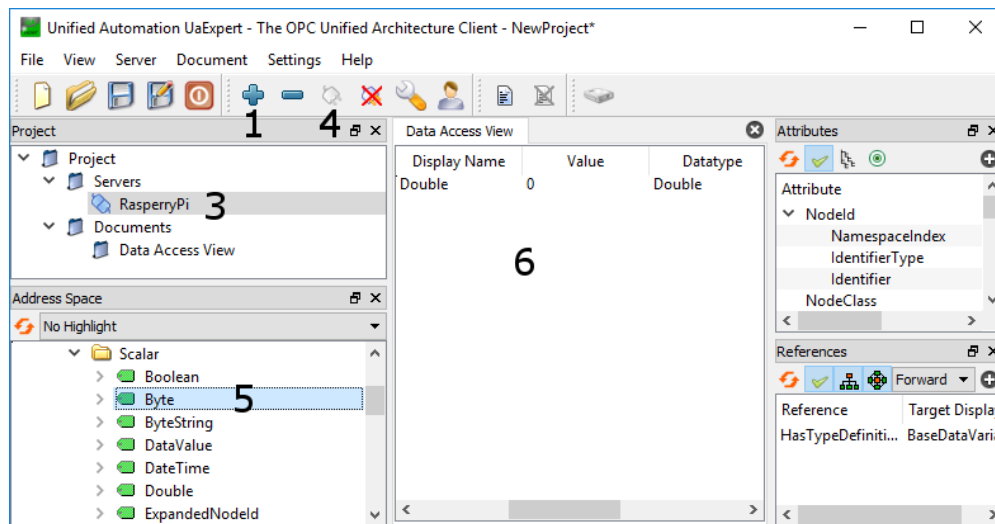


Abbildung 2: Reihenfolge in UaExpert zum Auslesen von Variablen aus einem OPC UA-Server

3.1.3. MQTT

Allgemein

MQTT steht für Message Queuing Telemetry Transport. Es handelt sich um ein einfaches und „leichtes“ Messaging-Protokoll, das für Geräte und Netzwerke mit geringer Bandbreite, hoher Latenzzeit oder unzuverlässiger Netzwerke entwickelt wurde. Die Prinzipien des Protokolls bestehen darin, den Bedarf an Netzwerkbandbreite und Geräteressourcen zu minimieren und gleichzeitig zu versuchen, die Zuverlässigkeit und Liefersicherheit zu gewährleisten. Diese Prinzipien erweisen sich als ideal für die entstehende "Machine-to-Machine" oder "Internet of Things" Welt sowie für mobile Anwendungen, bei denen Bandbreite und Akkuleistung im Vordergrund stehen. [2]

Kommunikation und Datenmodell

MQTT setzt auf eine brokerbasierte Publisher/Subscribe-Kommunikation. Die Publisher senden ihre Daten, z.B. die Sensorwerte der Werkstückträger, an einen zentralen Broker, indem sie die Daten in Nachrichten verpacken. Die Subscriber können die Daten am Broker abonnieren und erhalten vom Broker beim Ankommen neuer Nachrichten eine Kopie der Nachrichten. Die Organisation der Nachrichten erfolgt in sogenannten Topics, die hierarchisch strukturiert sind. Spezifiziert wird dies durch eine pfadähnliche Angabe (z. B. "ebene1/ebene2/ebene3"). Abonniert der Subscriber die "ebene1", so empfängt er auch Nachrichten, die an "ebene2" gesendet werden.

3.1.4. Python

Allgemein

Python ist eine universelle, üblicherweise interpretierte, höhere Programmiersprache. Sie hat den Anspruch einen gut lesbaren, knappen Programmierstil zu fördern. So werden beispielsweise Blöcke nicht durch geschweifte Klammern, sondern durch Einrückungen strukturiert. Python unterstützt mehrere Programmierparadigmen, z.B. die objektorientierte, die aspektorientierte und die funktionale Programmierung. Ferner bietet es eine dynamische Typisierung. Wie viele dynamische Sprachen wird Python oft als Skriptsprache genutzt.

Installation und Konfiguration

Python kann von <https://www.python.org/downloads/> heruntergeladen werden. Zur Installation muss der Installer ausgeführt werden.

Detailliertere Informationen finden sie unter:

https://www.tutorialspoint.com/python/python_environment.htm

Erstellen und Starten von Skripten

Ein Skript wird durch das Anlegen einer neuen Datei mit der Endung “.py” erzeugt. Zur Ausführung des Skriptes wird die Kommandozeile verwendet.

```
python meinSkript.py
```

Programmieren mit Python

Auf <https://www.learnpython.org/en/Welcome> gibt es unter dem Abschnitt “Learn the Basics” ein interaktives Tutorial mit dem sie sich die Grundlagen von Python aneignen können.

3.1.5. Maschinelles Lernen

Diese Übung baut auf den Ergebnissen aus Übung 2 auf. Dort wurden unterschiedliche Verfahren verwendet, um intelligente Modelle zu trainieren. Sollten Sie Übung zwei nicht besucht haben, erhalten Sie in diesem Abschnitt einen Überblick zu den Grundlagen des Maschinellen Lernens. Mehr Details zu den maschinellen Lernverfahren erhalten sie in der Vorlesung in Block D.

Maschinelle Lernverfahren lassen sich in vier Kategorien einteilen, die im Folgenden kurz vorgestellt werden. [4]

Überwachtes Lernen

Überwachtes lernen bedeutet, dass jemand anwesend ist, der beurteilt, ob die richtige Antwort erhalten wird. Das heißt konkret, dass ein vollständiger Satz von beschrifteten (Labels) Daten während des Trainings eines Algorithmus vorhanden sein muss, wobei die Beschriftung die korrekte Antwort beinhaltet.

Unüberwachtes Lernen

Saubere, perfekt beschriftete Datensätze sind nicht immer leicht zu bekommen. Für Fälle, bei denen die Daten diese Eigenschaften nicht aufweisen, kann unüberwachtes Lernen genutzt werden. Beim unüberwachten Lernen erhält ein Lernmodell einen Datensatz ohne ausdrückliche Anweisungen, was damit zu tun ist. Der Trainingsdatensatz ist eine Sammlung von Beispielen ohne ein bestimmtes gewünschtes Ergebnis oder eine korrekte Antwort. Der Algorithmus versucht dann, die Struktur in den Daten automatisch zu finden, indem es nützliche Merkmale extrahiert und ihre Struktur analysiert.

Semiüberwachtes Lernen

Semiüberwachtes Lernen ist eine Mischung aus überwachtem und unüberwachtem Lernen. Das heißt der Trainingsdatensatz beinhaltet sowohl beschriftete als auch unbeschriftete Daten. Diese Methode

ist besonders nützlich, wenn es schwierig ist, relevante Merkmale aus den Daten zu extrahieren, und Beschriftungsbeispiele sind eine zeitaufwändige Aufgabe für Experten.

Bestärkendes Lernen

In dieser Art des Lernens versucht ein Agent, den optimalen Weg zu finden, um ein bestimmtes Ziel zu erreichen oder die Leistung bei einer bestimmten Aufgabe zu verbessern. Wenn der Agent Maßnahmen ergreift, werden diese bewertet und er erhält eine Belohnung. Das Hauptziel ist die Vorhersage des besten nächsten Schrittes, um die größte Endbelohnung zu erhalten.

Um seine Entscheidungen zu treffen, stützt sich der Agent sowohl auf Erkenntnisse aus früheren Rückmeldungen, als auch auf die Erforschung neuer Taktiken, die einen größeren Nutzen bringen können. Dies beinhaltet eine langfristige Strategie - so wie der beste sofortige Zug in einem Schachspiel auf lange Sicht nicht helfen kann zu gewinnen, versucht der Agent, die kumulative Belohnung zu maximieren.

3.2. Anwendungsbezogene Grundlagen für die Aufgaben

3.2.1. Aufbau der Montagelinie

Die Montagelinie im SAL besteht aus 8 Bearbeitungsstationen (Abbildung 3), die durch eine SPS gesteuert werden. Auf dem Förderband sitzen Werkstückträger, auf denen das Produkt an den einzelnen Stationen zusammengebaut wird. An Station 3A/3B wird das Unterteil montiert, an Station 4A/4B das Oberteil und an Station 5A/5B werden die beiden Teile durch einen Zylinder verpresst. Die Farbe der einzelnen Teile kann bei der Bestellung eines neuen Produkts frei ausgewählt werden. In Station 7 werden die gefertigten Teile von einem pneumatischen Greifer erfasst und auf einen Ablagestapel transportiert. Der Werkstückträger fährt anschließend wieder zu den Stationen 3A/3B, um das nächste Produkt zu transportieren.

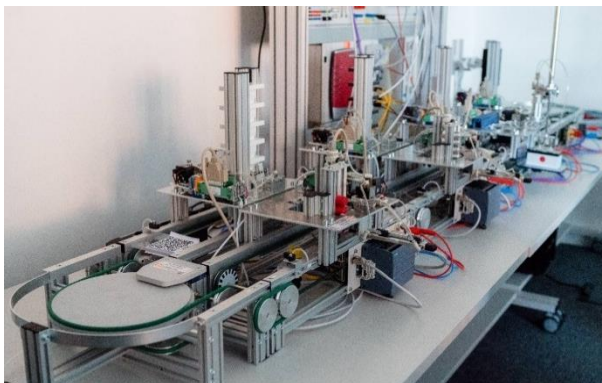


Abbildung 3: Montagelinie (Lukas Nuelle Anlage)



Die SPS hat die Kommunikationsschnittstelle OPC UA (Siehe 3.1.1), die es ermöglicht die Daten der Sensoren und Aktoren der Maschine auszulesen (Siehe 3.2.3). Zusätzlich ist jeder Werkstückträger mit Sensoren (Beschleunigung, Magnetfeld, Temperatur und Luftfeuchtigkeit) ausgestattet, die über MQTT (Siehe 3.2.2) auslesbar sind. Neben der Datenerfassung ist auch ein schreibender Eingriff auf die Anlage möglich. Der OPC UA Server der SPS bietet diese Möglichkeit. Rein theoretisch kann somit der Ablauf des oben beschriebenen Prozesses angepasst werden. Die Komplexität der Zustandsmaschine dieser Anlage verhindert jedoch die meisten direkt eingreifenden Prozessanpassungen.

3.2.2. MQTT-Parametrisierung

Um sich mit einem MQTT Broker zu verbinden, benötigt der Client zwei grundlegende Informationen: eine IP-Adresse sowie einen Port.

- Der Broker in diesem Praktikum ist unter dieser IP Adresse erreichbar: 141.58.103.26
- Für den Verbindungsaufbau wird folgender Port verwendet: 1883

Über den Broker werden zyklisch Sensordaten versendet. Hierfür werden vier unterschiedliche Topics verwendet:

Topic	Beschreibung
esp32acc	Werte des Beschleunigungssensors
esp32mag	Werte des Magnetfeldsensors
esp32status	Werte über den Zustand des Sensorboards
esp32clima	Klimatische Werte

Eine Nachricht des esp32acc Topics ist nachfolgend beispielhaft dargestellt:

SENSE,007,m,5072,2019-09-11T13:36:02.466714000Z,915.00,-282.00,-32768.00

Man erkennt, dass mehrere Datenwerte, durch ein Komma getrennt, in der Nachricht verschickt werden. In der nachfolgenden Übersicht sind die Bedeutung und der Datentyp der Datenwerte der einzelnen Nachrichten je Topic aufgelistet.

Topic	Feld 1	Feld 2	Feld 3	Feld 4	Feld 5	Feld 6	Feld 7	Feld 8
esp32acc	keyword (String)	swtid (int)	type (String)	msgno (int)	time (String)	xacc (float)	yacc (float)	Zacc (float)
esp32mag	keyword (String)	swtid (int)	type (String)	msgno (int)	time (String)	xmag (float)	ymag (float)	Zmag (float)
esp32status	keyword (String)	swtid (int)	type (String)	msgno (int)	time (String)	batvold (float)	-	-
esp32clima	keyword (String)	swtid (int)	type (String)	msgno (int)	time (String)	temp (float)	humi (float)	Pres (float)

Für die Übung sind die Felder swtid, xacc, yacc und zacc wichtig. Ersteres gibt an, welcher Werkzeugträger die Daten sendet. Die anderen drei Felder liefern die Sensorwerte des Magnetfeldsensors in x-, y- und z-Richtung.

3.2.3. OPC UA-Parametrisierung

Im OPC UA Server der SPS sind die Daten aller Programmbausteine der SPS abgebildet. Nachfolgend ist ein Teil der Knoten-IDs aufgelistet. Die fett gedruckten Knoten werden für einen Teil der Übung gebraucht.

- 'ns=3;s="DB_OPC_UA"."GLOBAL_DATEN"."Systemdruck"',
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."I_Eingaenge"."I_IMS5A_B5"'
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."I_Eingaenge"."I_IMS5A_B6"'
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."I_Eingaenge"."I_IMS5A_IL"'
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."I_Eingaenge"."I_IMS5A_B4"'
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."Q_Ausgaenge"."Q_IMS5A_M1"'
- 'ns=3;s="DB_OPC_UA"."IMS_5A_M0009682"."Q_Ausgaenge"."Q_IMS5A_M2"'
- 'ns=3;s="DB_OPC_UA"."IMS_7_M0000000"."I_Eingaenge"."I_IMS7_B3"'
- 'ns=3;s="DB_OPC_UA"."Qi_CHARGER"."Ladetemperatur_PT100"'

Für die unterschiedlichen Anwendungsfälle im Praktikum sind nicht alle Knoten relevant. Die Fettgedruckten Knoten geben die Zustände der Endlagensensoren von Modul 5A wieder. Anhand der

Benennung ist zu erkennen, welcher Knoten welchen Endlagensensor beschreibt. „IL“ beschreibt den Zustand des Endlagensensors beim Eingang zu Station 5a. „B4“ beschreibt den Endlagensensor des Zylinders bei der Einfuhr des Stifts in Station 5a.

Um die Syntax der Knoten-ID einer OPC UA Variable verstehen zu können, hilft folgender Hinweis:

Die ID einer Variablen ist als Baumstruktur abgebildet. Jeder Knoten hat maximal einen Elternknoten und beliebig viele Kinderknoten. Die Punkte innerhalb der ID sind Übergänge zwischen den einzelnen Knoten. Zusätzlich zur ID gibt es einen Namespace, der als Stammknoten einer ID-Baumstruktur gesehen werden kann.

Namespace der OPC UA Variable	ID der Variable
ns=3	s="DB_OPC-UA"."IMS_7_M0000000"."Q_Ausgaenge"."Q_IMS7_QS"

Die Bedeutung der Variablen-ID ist wie folgt aufgebaut:

- **DB_OPC-UA:** Toplevel-Knoten, unter dem alle über OPC UA verfügbaren Nodes im Namespace 3 angeordnet sind
- **IMS_7_M0000000:** Steht für eine Station der Lukas Nuelle Anlage. In diesem Fall Station 7
- **Q_Ausgaenge:** Alle Ausgänge, die in Station 7 vorliegen
- **Q_IMS7_QS:** Ein spezifischer Ausgang – in diesem Fall QS

3.2.4. Machine Learning mit mglearn

Als Machine Learning Paket in Python wird in dieser Übung, wie auch in Übung 2, mglearn genutzt. Dieses Paket fasst viele Algorithmen und Funktionalitäten zusammen, die in dieser Übung benötigt werden. Als Grundlage für diese Bibliothek dient scikit.

Das Erstellen, Trainieren und Evaluieren eines Modells wurde bereits in Übung 2 durchgeführt. Im folgenden Abschnitt steht eine Zusammenfassung der Befehle, die Sie in Übung 3 benötigen:

Die eigentliche Nutzung des Modells erfolgt über die `predict()` –Methode. Hierbei werden Features als Parameter in das Modell gegeben, welches daraufhin eine Deutung der Daten vornimmt:

```
clf = KNeighborsClassifier(n_neighbors=3)
clf.fit(X_train, y_train)
print("Test set predictions: {}".format(clf.predict(X_test)))
Test set predictions: [1 0 1 0 1 0 0]
```

In diesem Beispiel wird die k-NN Klassifikation durchgeführt. Bei der Instanziierung der Klasse, wird die Anzahl von drei Nachbarn festgelegt. Anschließend wird das Training mit vorher festgelegten Trainings- und Testsets durchgeführt. Die `predict()` –Methode erzeugt ein Array, bei dem jedes Feld eine Klassifizierung vornimmt. Die möglichen Klassifikationen in diesem Fall sind 1 und 0. Es können je nach Anwendungsfall auch weitere Klassifikationen möglich sein. Wie der Methodenname bereits deutlich macht, handelt es sich hierbei um Schätzungen. Wie sicher diese Schätzungen sind, kann mit `predict_proba()` ermittelt werden:

```
print("Test set prediction probabilities: ".format(clf.predict_proba(X_test)))
Test set prediction probabilities:
[[ 0.8, 0.2],
 [ 0.4, 0.6],
 [ 0.8, 0.2],
 [ 0.1, 0.9],
```

```
[ 0.7, 0.3],  
[ 0.1, 0.9],  
[ 0.4, 0.6]])
```

Für das Eintreten jeder möglichen Klassifikation wird eine Wahrscheinlichkeit angegeben. Mit Hilfe dieser Methode ist es entsprechend möglich, das stark abstrahierte „1“ oder „0“ der `predict()`-Methode etwas genauer zu hinterfragen. Für jede mögliche Klassifizierung ist der Wahrscheinlichkeitswert innerhalb eines Arrays angegeben. Die erste Zeile des Beispiels sagt aus, dass der Zustand 1 mit 80 prozentiger Wahrscheinlichkeit korrekt ist. Zustand 0 hat hingegen nur eine 20 prozentige Wahrscheinlichkeit.

3.2.5. CommunicationManager

Für die Durchführung der Übung sind viele Funktionalitäten der bisher beschriebenen Hilfsmittel in bereits bestehenden Python-Skripten gekapselt. Das Skript `CommunicationManager` ist Teil des Python Codes, den Sie für Ihre Übung vom ISW erhalten. Es enthält Methoden, die eine einfache Nutzung der OPC UA und MQTT Clients ermöglichen. Die Funktionalität des Skripts ist in public und private Methoden aufgeteilt. Public-Methoden werden von Ihnen in den Aufgaben genutzt. Private-Methoden sind für die interne Logik des Skripts wichtig. Private-Methoden müssen nur ggf. in den Bonus-Aufgaben angepasst werden. Sollte es nicht explizit in einer Übung erwähnt werden, dürfen Sie keine Methoden im `CommunicationManager` anpassen.

Public Methoden:

- **MQTT_CreateClient()**
Erstellt einen neuen MQTT Client
- **MQTT_subscribeToTopic()**
Erstellt eine MQTT Subscription. Topic werden als Parameter übergeben
- **MQTT_startClient()**
Startet den MQTT Client (Public)
- **MQTT_stopClient()**
Stoppt den MQTT Client (Public)
- **OPCUA_createClient()**
Erstellt einen neuen OPC UA Client
- **OPCUA_subscribe()**
Erstellt eine OPC UA Subscription. Variablen werden als Parameter übergeben (Public)
- **OPCUA_write()**
Schreibt einen Wert in eine OPC UA Variable (Public)
- **OPCUA_stopClient()**
Stoppt den OPC UA Client (Public)
- **getDataSnapshot()**
Gibt den aktuellen Snapshot des Datenmodells der Nülle Anlage zurück
- **startCollectingData()**
Startet die Datenerfassung über OPC UA und MQTT
- **SendSnapshotToMLModel()**
Sendet den aktuellen Snapshot des Datenmodells an das trainierte Machine Learning Modell
- **stopCollectingData()**
Stoppt die Datenerfassung über OPC UA und MQTT

Private Methoden

- **__MQTT_stopClient()**
Stoppt den MQTT Client (Private)
- **__OPCUA_subscribe()**
Erstellt eine OPC UA Subscription. Variablen werden als Parameter übergeben (Private)

- **__OPCUA_write()**
Schreibt einen Wert in eine OPC UA Variable (Private)
- **__OPCUA_stopClient()**
Stoppt den OPC UA Client (Private)
- **__addToNuelleData()**
Wählt aus, welche Daten an welcher Position des gemeinsamen Datenmodells der Lukas Nülle Anlage gespeichert werden sollen (Private)
- **__insertVariables()**
Nimmt einen erhaltenen Datenwert und speichert ihn in dem gemeinsamen Datenmodell der Nülleanlage (Private)
- **__on_message()**
Callback, der aufgerufen wird sobald der Client eine MQTT Message erhält
- **__MQTT_startClient()**
Startet den MQTT Client (Private)

3.2.6. MachineLearningManager

Das zweite Skript, das Ihnen zur Lösung der Aufgaben dieser Übung hilft, ist der MachineLearningManager. Analog zum CommunicationManager werden in diesem Skript Methoden für die einfache Nutzung der Machine Learning Bibliotheken zur Verfügung gestellt. Auch dieses Skript sollten Sie nicht anpassen, außer es ist ausdrücklich in einer der Übungsaufgaben gefordert.

- **predict()**
Übergibt einen Teil der Snapshotdaten als Funktionsparameter und bekommt als Antwort die Vorhersage über einen potentiellen Fehlerfall an Station 5a
- **predictProbability()**
Übergibt einen Teil der Snapshotdaten als Funktionsparameter und bekommt als Antwort die Vorhersage über einen potentiellen Fehlerfall an Station 5a inklusive Wahrscheinlichkeit der Korrektheit. Die Struktur der Antwort ist [hier](#) dargestellt.
- **prepareModels()**
Bereitet alle Modelle für Fehlerfallanalyse an Station 5a vor. Die Modelle wurden bereits in Übung 2 trainiert und mit Hilfe dieser Methode lediglich importiert.

Für die Nutzung der trainierten Modelle zur Fehlererkennung ist die Übergabe von Features in Form von Funktionsparametern notwendig. Die übergebenen Features wurden bereits beim Training der Modelle in Übung 2 vorgegeben und müssen beim „predict()“-Aufruf in gleicher Reihenfolge und Repräsentation übergeben werden, wie es auch im Modelltraining der Fall war. In dieser Übung sind folgende Features für die Modelle notwendig:

Features für das Modell zur Fehlererkennung an Station 5a

Variable	Xmag	Ymag	Zmag
Beschreibung	X-Wert Magnetfeldsensor	Y-Wert Magnetfeldsensor	Z-Wert Magnetfeldsensor

Beispielhafter Aufruf:

```
Result = machineLearningManagerObject.predict(Xmag, Ymag, Zmag)
```

4. Aufbau und Ablauf des Praktikums

4.1. Ausgangszustand

Der Ausgangszustand von Übung 3 ist das Ergebnis der ersten beiden Übungen. Eine solide Datenbasis der Lukas Nülle Anlage ist erhoben und wurde genutzt, um Modelle zur Erkennung von Anomalien im Prozess der Anlage zu erstellen und zu trainieren. Zum jetzigen Zeitpunkt liegen die trainierten Modelle vor. Eine Anwendung im tatsächlichen Prozess finden sie jedoch noch nicht.

4.2. Ablauf

Während der Übung greifen Sie auf Informationen aus den Übungen 1 und 2 zurück. Die Datenerfassung aus Übung 1 wird in dieser Übung angepasst. Sie verbinden sich mit dem intelligenten Werkzeugträger sowie der Steuerung der Lukas Nülle Anlage. In Übung 1 haben Sie Daten persistent in eine Datenbank gespeichert, um die Daten zu einem späteren Zeitpunkt nutzen zu können. Dies ändert sich in dieser Übung. Die erfassten Daten werden live verarbeitet. Sie nutzen hierfür einen Datenstream und ein sich ständig aktualisierendes Datenmodell der Lukas Nülle Anlage.

Um einen sinnvollen Prozesseingriff zu realisieren, wird Ihr angeeignetes Daten- und Prozessverständnis benötigt. Sie wissen aus Übung 1 und 2, welcher Prozess auf der Anlage durchgeführt wird. Zusätzlich wissen Sie, welche Daten die einzelnen Module und Teilprozesse der Anlage beschreiben. Dieses Wissen wurde in Übung 2 zur Erstellung und dem Training von Modellen genutzt, die Aussagen über Fehlverhalten im Prozess der Lukas Nülle Anlage liefern können. Bisher wurden die Modelle lediglich mit gespeicherten Datensätzen getestet und validiert. In Übung 3 werden Sie Livedaten des Prozesses in die Modelle geben und deren Antworten auswerten, um anschließend den Prozess zu beeinflussen (siehe Abbildung 4 Strukturelle Übersicht des Übungsvorhabens).

Hierfür wurde die Lukas Nülle Anlage mit Visualisierungskomponenten erweitert, die auf Fehler im Prozess hinweisen sollen. Ihr Ziel in Übung 3 wird es sein, die Vorhersagen der Modelle zu deuten und die Visualisierungskomponenten semantisch korrekt anzusteuern.

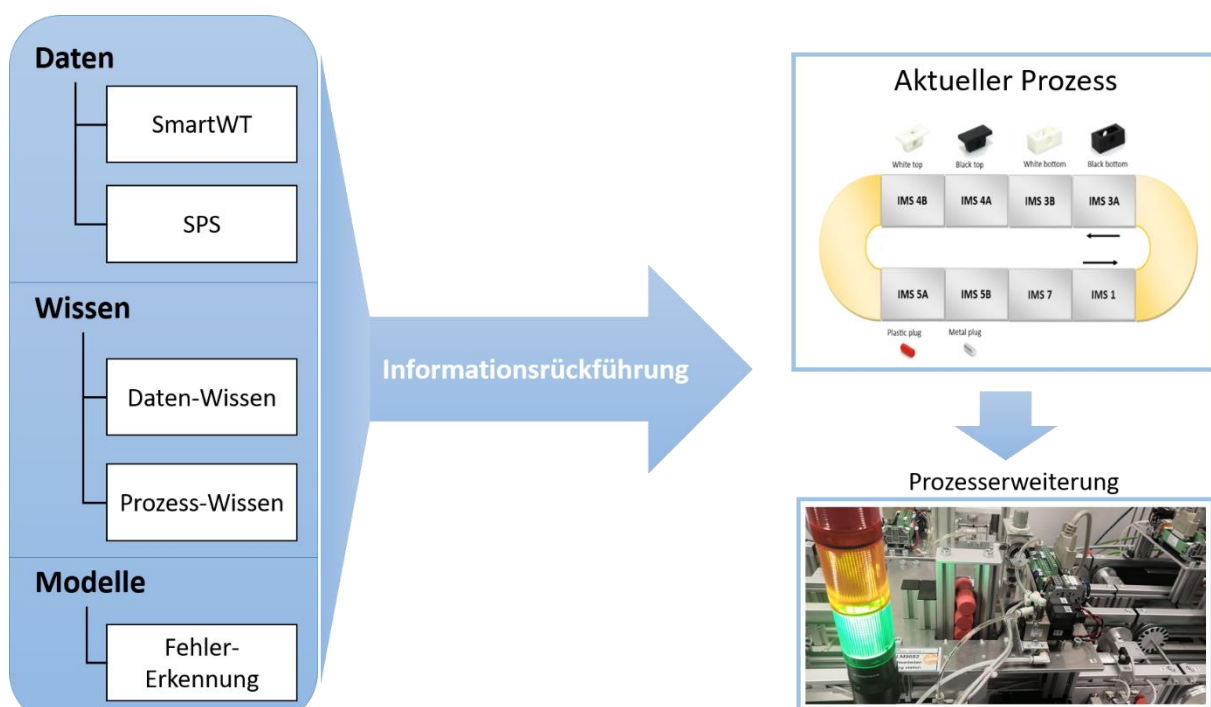


Abbildung 4 Strukturelle Übersicht des Übungsvorhabens

4.3. Ziel-/Endzustand

Am Ende der Übung haben Sie die Vorarbeiten aus Übung 1 und 2 genutzt und angepasst, um einen Prozesseingriff auf Basis der erhobenen Daten zu realisieren. Hierfür haben Sie Live-Daten der Lukas Nülle Anlage über OPC UA und MQTT erfasst und in einem gemeinsamen Datenmodell gespeichert. Von diesem sich ständig änderndem Datenmodell haben Sie zyklische Snapshots erstellt und mit Hilfe unterschiedlicher Modelle aus Übung 2 evaluiert. Basierend auf den Voraussagen der Modelle, haben Sie in Python eine Logik implementiert, die visuelle Warnsysteme der Lukas Nülle Anlage über OPC UA steuert. Auftretende Fehler können auf diese Weise zuerst visualisiert und nach Behebung wieder quittiert werden.

5. Vorbereitungsaufgaben

- Die Vorbereitungsaufgaben dienen der vertieften Vorbereitung; die Studierenden müssen diese SCHRIFTLICH bearbeiten und die Lösungen mitbringen.

5.1. Aufgabe V1

- Melden Sie sich bei Unified Automation an und installieren Sie anschließend den UaExpert. (<https://www.unified-automation.com/de/downloads/opc-ua-clients.html>)
- Verbinden Sie sich mit folgendem öffentlichen OPC UA Server:
opc.tcp://opcuaserver.com:48484
- Navigieren Sie zu den Variablen, welche die aktuellen Temperaturwerte für Sulzbach in Baden-Württemberg beinhalten. Ziehen Sie die aktuelle Temperatur (Temperature) und die gefühlte Temperatur (Apparent Temperature) in den Data Access View des UaExpert.
- Bringen Sie einen ausgedruckten Screenshot des UaExpert mit den beiden Variablen im Data Access View in die Übung mit.
- Schreiben Sie den NamespaceIndex, den IdentifierType sowie den Identifier beider Variablen auf die Rückseite des Ausdrucks.

5.2. Aufgabe V2

In Übung 1 haben Sie alle möglichen Daten der Lukas Nülle Anlage abgegriffen und gespeichert. In Übung 2 haben Sie mit Hilfe dieser Daten unterschiedliche Modelle erstellt und trainiert, die Fehler im Prozess der Lukas Nülle Anlage erkennen können. In Übung 3 wird basierend auf diesen Ergebnissen ein Prozesseingriff in die Anlage vorgenommen. Der aktuelle Erstellungsablauf eines Produkts auf der Lukas Nülle Anlage soll sich auf Basis des neu erworbenen Wissens aus Übung 2 ändern. Hierfür müssen jedoch zuerst Überlegungen angestellt werden, inwiefern der Prozess sinnvoll geändert werden kann. Dies ist das Ziel dieser Aufgabe!

- Bereiten Sie schriftlich mindestens eine Idee / einen Ansatz vor, wie der aktuelle Prozess der Lukas Nülle Anlage angepasst/erweitert werden könnte, um auf erkannte Fehler reagieren zu können. Überlegen Sie hierbei welche Eingriffe in die Anlage nötig sind und welche Vorteile Ihre Lösung im Vergleich zum Status Quo bringen wird.
Vereinfacht ausgedrückt: Wie können die erarbeiteten Modelle zur Fehlererkennung sinnvoll genutzt werden?

6. Versuchsdurchführung

6.1. Aufgabe 1 – Vorbereitung

Bevor Sie mit der eigentlichen Übung beginnen können, müssen Vorbereitungen getroffen werden. Hierfür richten Sie ihre IDE ein und legen ein Programm mit den zur Verfügung gestellten Dateien an. Diese Aufgabe wird gemeinsam durchgeführt. Bitte starten Sie diese Aufgabe nicht selbstständig.

6.1.1. Lernziel der Aufgabe

Sie können mit GIT ein Repository klonen und die im Repository enthaltenen Skripte mit PyCharm in einem neuen Python-Projekt öffnen.

6.1.2. Beschreibung und Ablauf der Aufgabe

Bevor Sie mit den eigentlichen Aufgaben der Übung starten können, müssen Sie Ihr System vorbereiten. Hierfür werden die vorbereiteten Python Skripte über git aus dem entsprechenden Repository geklont. Anschließend setzen Sie PyCharm, die Entwicklungsumgebung für diese Übung, auf und erstellen ein Projekt.

1. Legen sie einen neuen Ordner „D:\DataScienceUe3\“ an.
2. Rufen Sie in dem Ordner ein Commandline Tool auf und klonen Sie das Repository mit dem Befehl: `git clone https://github.com/iswunistuttgart/DataScienceUeb3.git`
3. Starten Sie das Programm PyCharm
4. Wählen Sie über File->Open den Ordner „D:\DataScienceUe3\“ aus und importieren Sie die heruntergeladenen Skripte in ein neues PyCharm Projekt.
5. Klicken Sie in PyCharm auf EditConfigurations
 - a. Wählen Sie im sich öffnenden Fenster links Python als Projekttyp aus
 - b. Auf der rechten Seite wählen Sie als Script die Datei Ueb3_Main_? aus

Wichtig:

Beginnend mit dieser Aufgabe haben Sie die Möglichkeit einen von drei Schwierigkeitsgraden der Übung zu wählen. Ihnen stehen drei Dateien zur Verfügung, von denen Sie eine im Laufe der Aufgaben 2 und 3 bearbeiten werden:

Ueb3_Main_Standard:

Diese Datei spiegelt den normalen Schwierigkeitsgrad der Übung wieder. Diese Option sollte von den Studierenden gewählt werden, die keine oder moderate Vorkenntnisse in Python oder im Umgang mit OPC UA und MQTT haben. Sind Sie sich unsicher, wählen Sie diese Datei!

Ueb3_Main_Advanced:

Diese Datei spiegelt einen erhöhten Schwierigkeitsgrad der Übung wieder. Diese Option sollte nur gewählt werden, wenn Vorkenntnisse in Python und im Umgang mit OPC UA und MQTT vorhanden sind. Die Übung wird schwieriger als vergleichbare Übungen des ISW.

Ueb3_Main_Difficult:

Diese Datei ist für Studierende gedacht, die bereits mehrere Python Projekte absolviert haben und sehr gut programmieren können. Diese Datei sollten Sie lediglich wählen, wenn Sie denken, dass sie die Aufgaben 3 und 4 ohne jegliche Anleitung lösen können.

Hinweis:

Während der Übungsdurchführung gibt es keine Hilfestellung seitens des Übungsleiters für die Übungen *Ueb3_Main_Advanced* und *Ueb3_Main_Difficult*. Wenn Sie diese Übungen gewählt haben und feststecken, orientieren Sie sich an den Hinweisen der *Datei Ueb3_Main*.

- c. Als Python Interpreter wird Python 3.7.0 gewählt
6. Installieren Sie, wenn nötig, fehlende Bibliotheken über die Kommandozeile (cmd)
 - a. Der Befehl hierfür ist: `pip install „Packagename“ –user`

Die Paketnamen, die installiert sein müssen sind:

- Paho-mqtt
- Opcua
- Scikit-learn
- numpy

6.1.3. Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis

Mit Abschluss der Aufgabe haben Sie alle nötigen Grundlagen geschaffen, um die folgenden Übungen praktisch durchzuführen.

6.2. Aufgabe 2 – Diskussionsrunde Prozesseingriff

Zu diesem Zeitpunkt der Übung haben Sie alle Vorbereitungen getroffen. Bevor Sie jedoch Livedaten erfassen und auswerten, sollte festgelegt werden, welche Aktionen ausgeführt werden sollen. Hierfür werden in einer Diskussionsrunde mögliche Prozesseingriffe in die Lukas Nülle Anlage diskutiert.

6.2.1. Lernziel der Aufgabe

Sie verstehen, wie die Lukas Nülle Anlage funktioniert und sind in der Lage zu erkennen, welche Prozesseingriffe möglich oder nicht möglich sind. Sie verstehen, dass die Komplexität von Fertigungssystemen des Öfteren Lösungsansätze verhindert, die auf den ersten Blick plausibel und einfach umzusetzen scheinen.

6.2.2. Beschreibung und Ablauf der Aufgabe

In einem gemeinsamen Brainstorming werden die von Ihnen und Ihren Kommilitonen vorbereiteten Lösungsansätze für die Lukas Nülle Anlage zusammengetragen und diskutiert. Nachdem alle Ideen gesammelt wurden, erklärt der Übungsleiter die Funktionsweise der Lukas Nülle Anlage. Mit diesem Wissen wird die Einsatzbarkeit der gesammelten Lösungsansätze bewertet.

6.2.3. Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis

Nach Durchführung des Brainstormings und der Diskussionsrunde wissen Sie, welche Eingriffsmöglichkeiten in die Lukas Nülle Anlage realisierbar sind. Anschließend werden Sie eine dieser Eingriffsmöglichkeiten in Aufgabe 3, 4 und 5 umsetzen.

6.3. Aufgabe 3 – Datenerfassung

Im ersten praktischen Schritt der Übung wird eine Verbindung zur Lukas Nülle Anlage hergestellt. Hierfür wird OPC UA und MQTT als Übertragungsprotokoll verwendet. Für beide Technologien muss ein Client erstellt und mit der Lukas Nülle Anlage verbunden werden. Anschließend werden die benötigten Variablen abonniert und verarbeitet.

6.3.1. Lernziel der Aufgabe

Sie können OPC UA und MQTT Subscriptions in Python anlegen und Änderungen von abonnierten Variablen in einem gemeinsamen Datenmodell speichern.

6.3.2. Beschreibung und Ablauf der Aufgabe

Fügen Sie den fehlenden Code in der Datei *Ueb3_Main_Standard.py* (bzw. *Ueb3_Main_Advanced.py* oder *Ueb3_Main_Difficult.py*) ein. Die Datei enthält Kommentare, markiert mit [Aufgabe3], die als Anleitung für diese Aufgabe dienen. Sie haben Aufgabe 3 erfolgreich beendet, sobald Ihr Projekt folgende **Anforderungen** erfüllt:

- Sie haben einen MQTT- sowie einen OPC UA Client mit Hilfe des Communication Managers erstellt
- Beide Clients haben alle notwendigen Variablen abonniert und empfangen diese. Welche Variablen „notwendig“ sind, erfahren Sie in diesem Skript und im Python-Code.
- Sie haben die Auswahloptionen „0“ und „1“ der Konsolenanwendung implementiert.
- Ihr Programm kann fehlerfrei gestartet werden. Anschließend kann die Datensammlung mit Eingabe der „1“ gestartet und mit „0“ beendet werden.

6.3.3. Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis

Mit Abschluss der Aufgabe haben Sie einen OPC UA- sowie einen MQTT-Client parametrisiert. Sie haben mehrere Subscriptions angelegt und erhaltene Informationen werden in einem gemeinsamen Datenmodell gespeichert. Aus diesem Datenmodell können im weiteren Verlauf der Übung Snapshots generiert werden, die für den weiteren Übungsverlauf benötigt werden.

6.4. Aufgabe 4 – Informationsrückführung

Zu diesem Zeitpunkt ist bekannt, wie der Prozesseingriff in die Lukas Nülle Anlage realisiert werden kann. Sie nutzen dieses Wissen sowie die vorbereitete Live-Datenerfassung aus Aufgabe 3, um eine Informationsrückführung an die Anlage zu implementieren. Durch OPC UA Write Befehle steuern Sie die Warnlampen der Lukas Nülle Anlage an. Der zu setzende Lampenstatus ist von der Voraussage des genutzten Modells abhängig.

6.4.1. Lernziel der Aufgabe

Sie lernen ein trainiertes ML-Modell zu nutzen und die Rückgabewerte des Modells zu interpretieren. Anschließend werden Sie mit der Nutzung des UaExpert vertraut gemacht und verstehen hiermit den Aufbau eines OPC UA Servers. Des Weiteren sind Sie in der Lage über OPC UA einen schreibenden Aufruf vorzunehmen.

6.4.2. Beschreibung und Ablauf der Aufgabe

Füllen Sie die restlichen Lücken in *Ueb3_Main_Standard.py* (bzw. *Ueb3_Main_Advanced.py* oder *Ueb3_Main_Difficult.py*) aus. Die Kommentare, markiert mit [Aufgabe4] beschreiben Ihre Aufgaben näher.

Wichtig:

Bevor Sie die Warnlampen der Lukas Nülle Anlage direkt ansprechen, muss Ihr Programm mit dem UaExpert und einem persönlichen Variablensatz getestet werden. Der PC an dem Sie arbeiten hat eine Nummer. Die Variablen, die Sie für Ihre Tests verwenden, haben die Nummer Ihres PCs in der ID. Variablen für den PC mit der Nummer **13** sind beispielsweise unter folgenden Adressen zu erreichen:

- `'ns=3;s="DB_OPC_Meldesignale_13"."IMS_5A_Meldungen"."LampeGruen"'`
- `'ns=3;s="DB_OPC_Meldesignale_13"."IMS_5A_Meldungen"."LampeRot"'`
- `'ns=3;s="DB_OPC_Meldesignale_13"."IMS_5A_Meldungen"."LampeGelb"'`

Verhält sich das Programm Ihrer Meinung nach korrekt, zeigen Sie den erfolgreichen Durchlauf dem Übungsbetreuer. Nach dessen Freigabe können Sie Ihre Funktionalität direkt an der Lukas Nülle Anlage testen. Die richtigen Variablen-IDs für die Ansteuerung der Lampen erhalten Sie ebenfalls vom Übungsleiter.

Sie haben Aufgabe 4 erfolgreich bearbeitet, wenn das Programm folgende **Anforderungen** erfüllt:

- Sie rufen alle 500 ms zyklisch ein trainiertes Modell mit einem Snapshot der gesammelten Daten auf und werten dessen Antwort aus.
- Abhängig von der Vorhersage des Modells, schreibt ihr Programm OPC UA Variablen, um die richtige Warnlampe der Lukas Nülle Anlage zu bedienen. Tritt ein Fehler auf, soll die entsprechende Lampe rot leuchten. Ist kein Fehler vorhanden, wird die Lampe auf grün gesetzt. Bei Unklarheiten oder wenn keine Aussage möglich ist, soll die Lampe gelb leuchten.
- Das Programm kann mit dem Tastendruck „2“, die Warnlampe der Lukas Nülle Anlage ausschalten (Kein Licht)
- Das Programm kann mit dem Tastendruck „3“, einen Fehler an Station 5a quittieren (Grünes Licht)
- Das Programm beendet zusätzlich alle in dieser Aufgabe neu eingefügten Funktionalitäten beim Tastendruck „0“.
- Ihr Programm kann fehlerfrei gestartet werden. Anschließend können alle Optionen der Kommandozeileingabe ausgewählt werden. Ihr Programm kann jederzeit durch Eingabe der „0“ beendet werden.

6.4.3. Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis

Basierend auf den Vorhersagen wird das Warnlampen-System der Lukas Nülle Anlage angesprochen und korrekt genutzt. Eine Visualisierung potentieller Fehler ist somit möglich. Bei einem Testdurchlauf Ihres Programms, stellen Sie fest, dass die Vorhersagen des Modells nicht stabil sind.

6.5. Aufgabe 5 – Informationsrückführung (Bonusteil)

Achtung – Bonusaufgabe:

Mit Abschluss von Aufgabe 4 haben Sie das Praktikum erfolgreich abgeschlossen. Sie haben einen kompletten Kreislauf aus Handlungsplanung, Datenerfassung und Bereinigung, Modellerstellung und -training sowie Prozessintegration absolviert. Im richtigen Prozess würde an dieser Stelle iteriert werden, um die Modelle mit anderen Features zu trainieren oder die Ziele anzupassen. Für die Übung dieser Vorlesung gibt es allerdings ein alternatives Ende:

Sie haben alle bisherigen Aufgaben richtig bearbeitet, allerdings stellen Sie fest, dass die Predict()-Methode des genutzten Modells nicht immer das richtige Ergebnis liefert. In dieser Aufgabe finden Sie heraus, weshalb dieses Verhalten eintritt und passen Ihre Lösung entsprechend an.

6.5.1. Lernziel der Aufgabe

Sie nutzen Ihr Wissen über das Training von Machine Learning Modellen aus Übung 2, um eine Erklärung für die fehlerhaften Voraussagen des Modells zu treffen. Anschließend lernen Sie, dass eine Eingrenzung des Problemraums zu besseren Ergebnissen führen kann.

6.5.2. Beschreibung und Ablauf der Aufgabe

Passen Sie Ihr Programm so an, dass die Vorhersage des Modells lediglich genutzt wird, sobald sich ein Werkzeugträger unter Station 5a befindet. Nutzen Sie hierfür die Endlagensensoren, die Sie über OPC UA subscribed haben.

Sie haben Aufgabe 5 erfolgreich bearbeitet, wenn das Programm folgende **Anforderungen** erfüllt:

- Ihr Programm enthält alle Funktionen, die in Aufgabe 4 implementiert wurden
- Zusätzlich startet Ihre Analyse erst, sobald sich ein Werkzeugträger unter Station 5a befindet und endet, sobald der Werkzeugträger Station 5a verlässt
- Das Ergebnis der Analyse wird nach Verlassen der Station weiterhin statisch auf der Warnlampe angezeigt. Ein eventueller Fehler muss manuell über Ihr Programm quittiert werden.

6.5.3. Zielzustand der Aufgabe und Übertragung der Aufgabe in die Praxis

Sie haben die Probleme der trainierten Modelle verstanden und eine potentielle Lösung implementiert. Der Prozesseingriff in die Lukas Nülle Anlage ist hiermit abgeschlossen.

Bonus-Frage: Ist die erarbeitete Lösung korrekt? Teilen Sie Ihre Einschätzung inklusive Begründung dem Übungsleiter mit.



7. Hinweis auf nötiges Vorgehen zum Beenden der Versuchsdurchführung

- Schließen Sie alle Programme
- Löschen Sie den von Ihnen angelegten Ordner „D:\DataScienceUeb3“
- Melden Sie sich von Ihrem Rechner ab

Richtlinie zu Verhalten, Sicherheit und Arbeitsschutz am Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW)

Studierende und Wissenschaftliche Hilfskräfte

Allgemeine Regelungen der Universität

Grundsätzlich gilt für das Institutsgebäude des ISW die *Hausordnung der Universität Stuttgart* nach dem Beschluss des Rektors vom 26. Mai 2014, diese ist einzusehen unter:

http://www.uni-stuttgart.de/zv/bekanntmachungen/bekanntm_45_2014.pdf

Weiterhin wird ausdrücklich auf die *Richtlinie zur Verantwortung im Arbeits- und Umweltschutz an der Universität Stuttgart* verwiesen, welche vom Rektorat der Universität Stuttgart am 19.06.2007 beschlossen wurde. Diese ist einzusehen unter:

<http://www.verwaltung.uni-stuttgart.de/rundschreiben/2007/ra2007-079.pdf>

Besonders relevant sind dabei folgende Punkte:

Rechte und Pflichten der Beschäftigten und Studierenden (Auszug)

- Die Beschäftigten und Studierenden sind verpflichtet, nach ihren Möglichkeiten sowie gemäß der Unterweisung und Weisung durch den Verantwortlichen für ihre Sicherheit und Gesundheit Sorge zu tragen. Dabei haben sie auch für die Sicherheit und Gesundheit jener Personen zu sorgen, die von ihren Handlungen oder Unterlassungen bei der Arbeit betroffen sind.
- Die Beschäftigten und Studierenden haben insbesondere Maschinen, Geräte, Werkzeuge, Arbeitsstoffe, Transportmittel, sonstige Arbeitsmittel, Schutzvorrichtungen und die ihnen zur Verfügung gestellte persönliche Schutzausrüstung bestimmungsgemäß zu verwenden.
- Die Beschäftigten und Studierenden haben dem Verantwortlichen jede von ihnen festgestellte, unmittelbare Gefahr für die Sicherheit und Gesundheit sowie jeden an den Schutzsystemen festgestellten Defekt unverzüglich zu melden. Gegebenenfalls können die Mängel auch den zuständigen Stellen der Zentralen Verwaltung (...) gemeldet werden.
- In Lehrveranstaltungen und bei der Durchführung von Studien-, Diplom- und anderen Examensarbeiten sowie von Dissertationen hat der Verantwortliche den Studierenden bzw. Doktoranden/-innen die für die Sicherheit und den Gesundheitsschutz notwendigen Maßnahmen vorzugeben. Die Studierenden bzw. Doktoranden/-innen haben sich an diese Vorgaben zu halten.

Zusätzlich zu den allgemeinen Regelungen der Universität Stuttgart gelten folgende Regeln beim Umgang mit elektrischen Einrichtungen:

Geltungsbereich:

Laborräume gelten im Sinne der VDE-Vorschriften als Elektrische Betriebsräume (DIN VDE 0100). Tätigkeiten im Labor dürfen daher nur nach entsprechender Einweisung durchgeführt werden.

Unterrichtsräume bzw. -bereiche, in denen Übungen nur an Versuchsständen bzw. Geräten durchgeführt werden, von denen keine Gefährdungen im Sinne der VDE-Vorschriften ausgehen, gelten nicht als Elektrische Betriebsräume. In diesem Fall kann die Unterweisung der an der Übung teilnehmenden Studierenden auf den allgemeinen sowie den Versuchsstand betreffenden Teil reduziert werden.

1. Jeder Hochschulangehörige bzw. Studierende hat sich vor dem Einschalten einer elektrischen Anordnung mit deren Aufbau und Arbeitsweise sowie mit der Bedienung der verwendeten Geräte vertraut zu machen und sich vom augenscheinlich einwandfreien Zustand zu überzeugen. Es ist seine besondere Pflicht, sich über die Möglichkeiten des schnellen Abschaltens der gesamten Anordnung genau zu informieren. An nicht unterwiesenen Geräten darf nicht gearbeitet werden.
2. Bei Störungen von Geräten oder Anlagen ist deren Spannungsversorgung sofort abzuschalten.
3. Jeder Hochschulangehörige bzw. Studierende hat folgende Sicherheitsregeln beim Umgang mit elektrischen Betriebsmitteln und Anlagen zu beachten:
 - A) freischalten,
 - B) gegen Wiedereinschalten sichern,
 - C) Spannungsfreiheit feststellen,
 - D) erden und kurzschließen,
 - E) benachbarte, unter Spannung stehende Teile abdecken oder abschränken.
4. Grundsätzlich dürfen keine nassen elektrischen Geräte benutzt und keine nassen elektrischen Anlagen bedient werden. Elektrische Heizgeräte oder Tauchsieder dürfen nicht benutzt werden.
5. Niemand darf Reparaturen an elektrischen Geräten oder Anlagen durchführen, wenn er keine ausreichenden Kenntnisse über die damit verbundenen Gefahren und die sichere Arbeitsweise nachweisen kann.
6. Jeder Hochschulangehörige bzw. Studierende muss sich vor der Benutzung von ortsfesten und ortsveränderlichen elektrischen Betriebsmitteln über die besonderen Sicherheitsmaßnahmen informieren. Diese Sicherheitsmaßnahmen sind strikt einzuhalten.
7. Wird festgestellt, dass Einrichtungen oder Hilfsmittel sicherheitstechnisch nicht einwandfrei sind, so ist dieser Mangel unverzüglich dem für das Gerät oder die Anlage Zuständigen zu melden. Die Geräte oder Anlagen sind nicht weiterzuverwenden und der Benutzung durch andere Personen zu entziehen.
8. Private Geräte sowie solche ohne gültige Prüfplaketten dürfen erst nach Zustimmung und deren Prüfung eingesetzt werden.
9. Änderungen am Aufbau elektrischer Schaltungen und Systeme müssen im spannungslosen Zustand vorgenommen werden. Unter Spannung stehende Schaltungen müssen beaufsichtigt bleiben. Falls dies nicht möglich ist, müssen ein Warnschild und eine Zugangsbeschränkung angebracht werden. Für ausreichenden Berührungsschutz ist zu sorgen.
10. Arbeiten an Gleichspannungen über 60 V bzw. Wechselspannungen über 25 V, offenen Geräten, Schaltschränken oder Versuchsaufbauten mit freiliegenden Netzversorgungsanschlüssen sowie Geräten, an denen vorübergehend Schutzmaßnahmen aufgehoben sind, dürfen nur bei Anwesenheit mindestens einer zweiten, nachweislich unterwiesenen Person im Laborbereich durchgeführt werden. Diese Person muss eine elektrotechnische Fachkraft sein. Derartige Arbeiten kommen für die Ausführung durch Studierende oder Wissenschaftliche Hilfskräfte grundsätzlich nicht in Betracht. Bei Schaltungen mit Betriebsspannungen über 60 VDC bzw. 25 VAC

sind in angemessener Weise die gemäß der VDE vorgeschriebenen Schutzmaßnahmen anzuwenden.

11. Manipulationen an unter Spannungen stehenden Schaltungen sind stets mit nur einer Hand auszuführen! Die andere Hand legt man sich dabei auf den Rücken oder steckt sie in die Tasche.
12. Versuchsaufbauten sind übersichtlich, Versuchsschaltungen sind übersichtlich und soweit möglich berührungssicher aufzubauen. Not-Aus-Taster und Fluchtwege müssen jederzeit freigehalten werden. Versuchsaufbauten müssen vor Verlassen des Arbeitsplatzes vom Netz getrennt werden.
13. Besondere Vorsicht ist beim Betrieb von Robotern bzw. bewegten Maschinen, sich erheizenden oder möglicherweise explodierenden Bauteilen einzuhalten. Es ist darauf zu achten, dass der Gefahrenbereich abgesperrt ist. Der Aufenthalt im Gefahrenbereich ist verboten. Notwendige Schutzmaßnahmen (z. B. Schutzbrillen) sind zu benutzen, lose Kleidung oder lange Haare müssen entsprechend befestigt werden.
14. Bei Lötarbeiten ist auf eine ausreichende Belüftung zu achten, entsprechende Arbeiten dürfen nicht in der Nähe von Lebensmitteln durchgeführt werden.
15. Sämtliche Arbeiten dürfen nur bei guter körperlicher und psychischer Verfassung ausgeführt werden. Verletzungen und Unfälle jeglicher Art sind zu melden und zu dokumentieren. Bei gesundheitlichen Einschränkungen, wie z. B. dem Tragen eines Herzschrittmachers oder dem Vorliegen einer Schwangerschaft, ist der Verantwortliche zu informieren um Gefährdungen auszuschließen.

Mit meiner Unterschrift bestätige ich, die Richtlinien gelesen und verstanden zu haben.

Name

Matrikelnummer

Ort, Datum

Unterschrift Unterwiesener

Ort, Datum

Unterschrift Praktikumsbetreuer