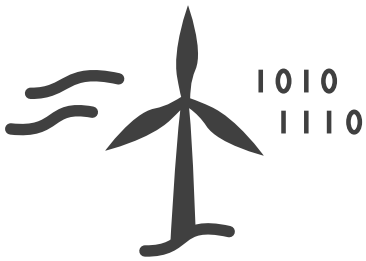


WiPMA

Projektgruppe „Wind Park Maintenance Assistant (WiPMA)“
- Zusammenfassung -

Projektzeitraum: 01.04.2021 bis 31.03.2022

Erstellt in Kooperation mit der Abteilung VLBA, der
Carl von Ossietzky Universität Oldenburg, unter der Leitung von Prof. Dr.-Ing. habil. Jorge Marx Gómez

Inhaltsverzeichnis

Abkürzungsverzeichnis

API	Application Programming Interface
CvOU	Carl von Ossietzky Universität
HTTP	Hypertext Transfer Protokoll
ML	Machine Learning
NBM	Normal Behavior Model
OCSVM	One-Class Support Vector Machine
PG	Projektgruppe
REST	Representational State Transfer
VLBA	Very Large Business Applications
WEA	Windenergieanlage
WiPMA	Wind Park Maintenance Assistent

1 Einleitung

Die vorliegende Arbeit ist die Zusammenfassung der Projektgruppe Wind Park Maintenance Assistent (WiPMA), welche an das Projekt WiSa Big Data der Abteilung Very Large Business Applications (VLBA) angegliedert ist. Die VLBA ist eine Abteilung des Fachbereiches Informatik der Carl von Ossietzky Universität. Das Ziel der Projektgruppe ist die Verbesserung der Planung von Wartungseinsätzen an (Offshore) Windenergieanlage.

Um dieses Ziel zu erreichen, werden Machine Learning-Algorithmen auf hochaufgelöste Daten von Windenergieanlages (WEAs) angewendet, die von Industriepartnern des Projekts WiSa Big Data zur Verfügung gestellt werden. Die Erkenntnisse werden aufbereitet und in einem Dashboard dargestellt. Die Analysen konzentrieren sich auf die Anwendung von Machine Learning (ML), um Auffälligkeiten zu finden und Fehler frühzeitig zu erkennen. Das Dashboard dient als Hilfe zur effizienten Planung von Wartungseinsätzen, da diese gezielt durchgeführt werden müssen.

1.1 Motivation

Bei dieser Projektarbeit wird der Fokus auf den Einsatz von Machine Learning in der Untersuchung der Daten auf Muster, welche eine Wartung von Windenergieanlage prognostizieren können, gelegt. Dies soll eine verbesserte und vorausschauende Planung von Wartungseinsätzen ermöglichen. Im Detail soll dadurch der Einsatz von Personal und Material effizienter gestaltet werden.

Das Themengebiet des Machine Learning bietet für die Projektgruppe die Möglichkeit der Auseinandersetzung mit einem Gebiet, welches fortwährend weiterentwickelt wird und neue Erkenntnisse liefert. Dies kann an der Menge an wissenschaftlichen Publikationen zu dem Thema Künstliche Intelligenz, die jedes Jahr veröffentlicht werden, abgeleitet werden [KRM+20, S. 13]. Somit ist die Auseinandersetzung mit dem Thema künstliche Intelligenz bzw. ML eine Motivation für die Gruppenmitglieder.

Die Motivation der Projektgruppe ergibt sich aus dem Problem, welches dem Projekt zugrunde liegt. Die Problemstellung wird in dem Abschnitt 1.2 dargestellt.

1.2 Problemstellung

Das Problem, mit dem sich die Projektgruppe auseinandersetzt, liegt in der Wartbarkeit von Offshore Windenergieanlage. Diese können nur an 80% aller Tage in einem Jahr erreicht und somit auch gewartet werden. Von diesen 80% lassen 75% der Tage nur eine Anreise mit dem Hubschrauber zu, während die restlichen 25% der Tage auch eine Anreise mit einem Schiff zulassen. [Kli21]

Dies bedeutet, dass größere Wartungen oder auch der Austausch von Komponenten einer solchen Offshore WEA nur an einem Fünftel aller Tage im Jahr zulässig sind. Daher müssen Wartungseinsätze möglichst effizient geplant werden, da die Zeiträume für solche Einsätze begrenzt sind. Zusätzlich werden Schiffe oder Hubschrauber für die Wartung benötigt. Somit sind die Einsätze mit hohen Kosten verbunden, da diese ebenfalls unterhalten oder gemietet werden müssen.

Außerdem entsteht bei einem Ausfall einer WEA ein Verlust, da eine stillstehende Anlage keine Energie produziert, welche vertrieben werden kann. Daher sollten die Anlagen so kontinuierlich wie möglich betrieben werden. Um dies zu gewährleisten müssen Komponenten rechtzeitig ausgetauscht und gewartet werden.

Aus den beiden aufgeführten Umständen ergibt sich das Problem der Planungseffizienz von Wartungseinsätzen. Dies ist auch das Problem, mit dem sich die Projektgruppe beschäftigt.

1.3 Vision

Sowohl die Projektziele als auch die daraus folgenden Anforderungen ergeben sich aus dem durch die Vision gezeichneten Zukunftsbild. Die Vision ist der Wegweiser und dient als Orientierung für alle Aktivitäten der Projektgruppe (PG). Das Visions-Statement lautet:

> Die Vision der Projektgruppe ist es passgenaue Wartungsarbeiten für Offshore-Windenergieanlage zu ermöglichen. Um den Paradigmenwechsel von Preventive zu Predictive Maintenance zu erreichen ist die Vorhersage von Bauteilausfällen notwendig.

1.4 Projektziel

Das Ziel der Projektgruppe WiPMA ist die Erstellung eines Wartungsassistenten für Offshore-Windenergieanlagen. Der Wartungsassistent soll über ein Dashboard für die Anwender zugreifbar sein. Auf dem Dashboard sind Rohdaten und Ergebnisse verschiedener Analysemodelle zu sehen. Das System soll skalierbar aufgebaut sein. Der Hauptfokus der Projektgruppe liegt auf der Erstellung der Analysemodelle. Die Modelle dienen dazu, die Anwender frühzeitig auf Zustandsveränderungen hinzuweisen bzw. diese zu prognostizieren.

2 Architektur

Im Zuge der Anforderungen an die Architektur wurden die Architekturvorschläge untersucht und abgewogen. Dabei ist jeweils zwischen der Auswahl eines Webanwendungs-Stacks und der Auswahl der Datenstromverarbeitung zu unterscheiden.

Ein grober Architekturentwurf ist in Abbildung Abbildung 1 zu sehen. Die technischen Details zu den jeweiligen Komponenten, werden in Abschnitt 3 und 4 erläutert.

2.1 Auswahl des Webanwendung-Stacks

Bei der Architektur der Webanwendung wurde sich für einen Hybrid aus Monolith- und Micro-Services entschieden. Die einzelnen Anwendungsfälle des Projektes werden über Micro-Services umgesetzt, während die Datenhaltung über eine zentrale Datenbank betrieben wird.

Grund für die Umsetzung von Micro-Services ist die generische Ausgestaltung und die Erweiterbarkeit der Anwendung. Sollten also weitere Anwendungsfälle oder verfeinerte ML-Modelle hinzukommen, können diese über weitere Micro-Services ergänzt werden, ohne anderweitige Anpassungen in der Anwendung vornehmen zu müssen.

Jeder aus den Anforderungen hervorgehender Anwendungsfall ist ein unabhängiges Modul. Jedoch greifen alle Module auf den gleichen Datenbestand zu. Durch die zentrale Datenbank können keine Inkonsistenzen zwischen den einzelnen Micro-Services auftreten. Inkonsistenzen werden durch Relationen innerhalb der Datenbank verhindert. Darüber hinaus ist

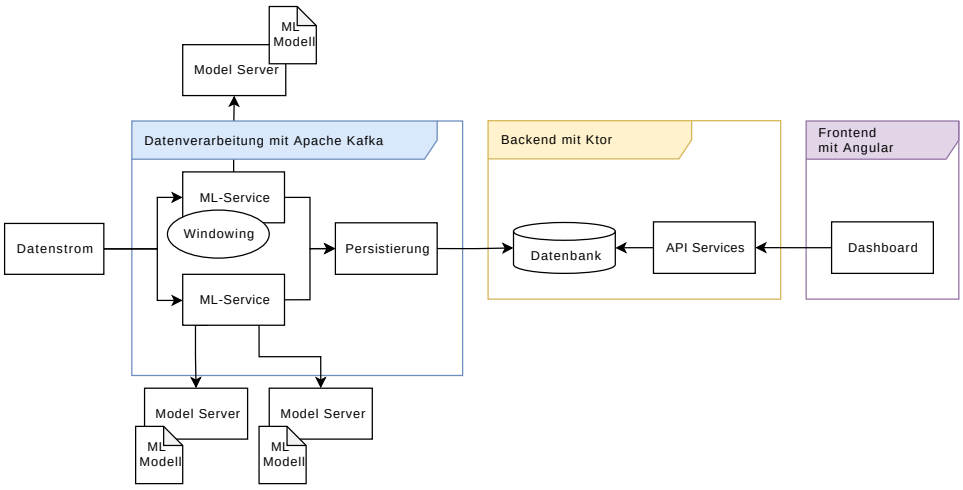


Abbildung 1: Vereinfachte Architektur

durch ein zentrales Datenbanksystem die Integration weiterer Analyse- oder Dashboardsysteme möglich.

2.2 Auswahl der Datenstrom-Architektur

Bei der Datenstromverarbeitung wurde sich ebenfalls für einen Hybrid aus Kappa und Lambda Architektur entschieden. Grund dafür sind die Anforderungen, welche eine unterschiedliche zeitliche Auflösungen des Datenstroms und verschiedene Aggregierungen von vergangenden Datenätzen voraussetzen.

Das Konzept des zentralen Datenstroms aus der Kappa Architektur spiegelt sich in dem zentralen Eingangsdatenstrom der Sensordaten wieder. Dieser vereinigt alle Sensordaten in sekündlicher Auflösung über alle zu betrachtenden WEAs. In Abhängigkeit vom Anwendungsfall zweigen sich von diesem Datenstrom unterschiedlich aufgelöste Datenströme ab, welches der Batch-Verarbeitung der Lambda Architektur ähnelt.

Da in jedem Anwendungsfall eine unterschiedliche Beschaffung der Eingangsdaten aus dem Datenstrom definiert ist, ist eine reine Umsetzung von Lambda oder Kappa nicht zu erwägen. Zum Einen ist die lambda-orientierte Aufteilung und Abstimmung von Echtzeit-Verarbeitung und Batch-Verarbeitung nicht notwendig, auf der anderen Seite ist durch die Abzweigung der Datenströme eine Verarbeitung in einem zentralen Datenstrom nicht sinnvoll.

Der konzipierte Datenstrom resultiert im WiPMA-System demnach in viele unterschiedliche Ergebnisse, welche in der zentralen Datenbank aus Unterabschnitt 2.1 gebündelt und gesammelt werden.

3 Frontend

Das Frontend ist für die Visualisierungen der ML-Ergebnisse zuständig. Es ermöglicht einen benutzerfreundliches Zugriff auf die Daten und Prozesse des jeweils zugrunde liegenden Backends in einer vereinfachten und abstrahierten Darstellung. Die Anwendung der Projektgruppe WiPMA hat zum Ziel die Planung von Wartungsarbeiten an WEAs zu verbessern. Dazu werden zwei Rollen, welche in dem Tätigkeitsfeld vorkommen mit unterschiedlichen Sichten auf die Daten ausgestattet. Diese sind die Rolle des Planers und des Supporters.

3.1 Unterscheidung der Rollen

Das Dashboard bzw. die Startseite des Frontends vereinigt alle Komponenten wie die Fehlerfrüherkennung, Anomalieerkennung und Zustandsüberwachung auf einen Blick. Um auf das Dashboard zu gelangen, muss vorher eine Authentifizierung erfolgen. Die drei Rollen Supporter, Planner und Administrator haben jeweils eine andere Sicht auf das Dashboard. Der Administrator vereinigt dazu die beiden Sichten des Planers und Supporters. Beim Planer liegt der Fokus auf der Fehlerfrüherkennung und beim Supporter auf der Anomalieerkennung. Die Abbildung 2 gibt ein Beispiel der Sicht des Administrator auf das Dashboardes wieder.

3.2 Darstellung der Anomalieerkennung

Zusätzlich werden im Dashboard für die Anomaliedarstellung im oberen Panel links (Abbildung 2) die Anlagen mit den häufigsten Anomalien angezeigt. Dazu werden Informationen über die Anomalieanzahl, die betroffene Anlage und den Healthscore gegeben.

Weitere Diagramme, die zur Anzeige der Anomalien verwendet werden, befinden sich auf der Seite Anomalieerkennung. Hier ist das erste Diagramm mit dem Namen Sensorübersicht zu finden. Die Sensorübersicht wird in

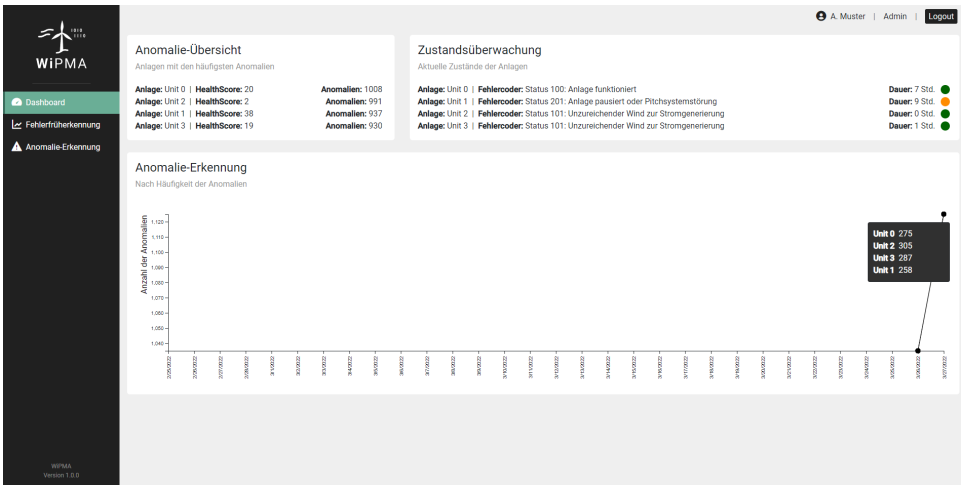


Abbildung 2: Admin-Sicht der Dashboardseite im Frontend

Form einer Heatmap dargestellt und gibt den täglichen Verlauf aller Features und dessen Häufungen und Einstufungen der Anomalien wieder.

Darunter wird der zeitliche Verlauf des Healthscore angezeigt. Diese Daten geben Aufschluss über den Zustand einer Anlage in den letzten 30 Tagen. Beispielsweise kann eine steigende Kurve auf eine intakte Anlage hinweisen. In der Abbildung 3 werden diesen beiden Diagramme nochmal veranschaulicht. Der Healthscore der Anlage 2 fällt dort täglich ab und kann darauf hindeuten, dass hier eine Wartung erfolgen sollte.

Das nächste Diagramm stellt dar, wie sich die Werte eines bestimmten Merkmals im Laufe der Zeit verändern (Abbildung 4). Um beim Diagramm mehrere Einstellungen vorzunehmen, muss dem Diagramm eine Reihe von Filtern (System-, Anlagen- und Featurefilter) hinzugefügt werden, um die angezeigten Daten zu verändern. Nach Auswahl aller Filter, wird die Grafik initialisiert und zeigt die Werte des Features im Zeitverlauf an. Regelmäßige Dateneinträge sind schwarz gefärbt, Anomalien werden rot dargestellt.

Die letzte Komponente auf der Seite der Anomalieerkennung ist eine Tabelle (Abbildung 5). Dort können Information über alle Features und Systeme eingeholt werden. Es kann über den Zeitraum, die Anlage oder das System gefiltert werden. In der Tabelle wird dann über das gewählte System die dazugehörigen Features nach Datum gelistet. Eine Einfärbung des Datensatzes in roter Farbe bedeutet, dass eine Anomalie in diesen Datensatz vorhanden ist.

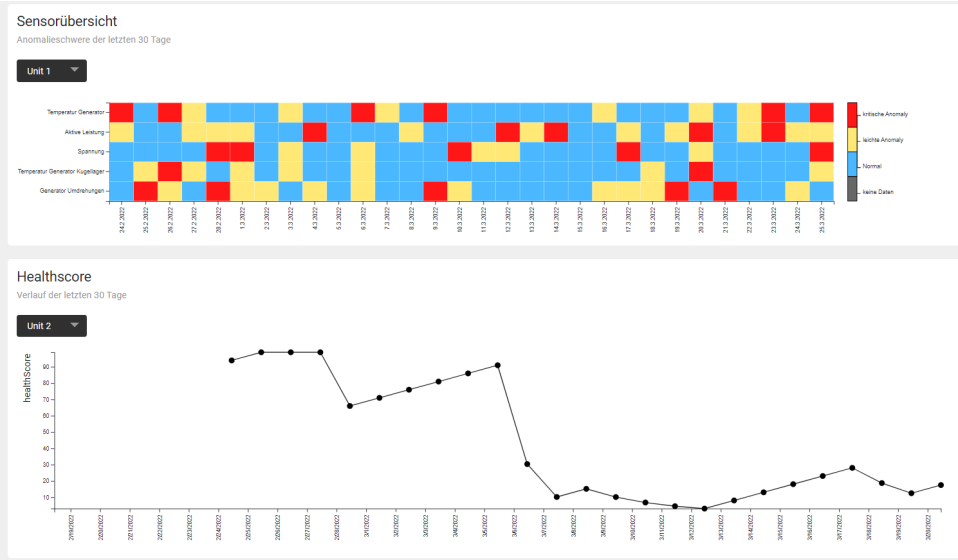


Abbildung 3: Darstellung der Sensorübersicht und des Healthscores im Frontend

3.3 Darstellung der Fehlerfrüherkennung

Auf der Seite der Fehlerfrüherkennung sind zwei Diagramme zu finden (Abbildung 6). Das erste Diagramm gibt einen minütlichen zeitlichen Verlauf von drei Transformatoren und eine dazugehörige Vorhersage wieder. Die Transformatorenwerte sind für die Unterscheidung des jeweiligen Typs farblich gekennzeichnet.

In Verbindung dazu existiert ein weiteres Diagramm, in dem eine Wettervorhersage und die Wirkleistung im zeitlichen Verlauf dargestellt wird. Hier kann jeweils das Wetter oder die Wirkleistung als Feature ausgewählt und nach der dazugehörigen Anlage gefiltert werden.

3.4 Darstellung der Zustandsüberwachung

Beide Rollen haben Zugriff auf die Ergebnisse der Zustandsüberwachung. Die Zustände werden oben rechts im Panel ausgegeben (Abbildung 2). Die Darstellung der Zustandsüberwachung erfolgt über ein Ampelsystem. Dieser Ansatz bietet eine visuelle Darstellung des Zustands jeder WEA im gegenwärtigen Moment.

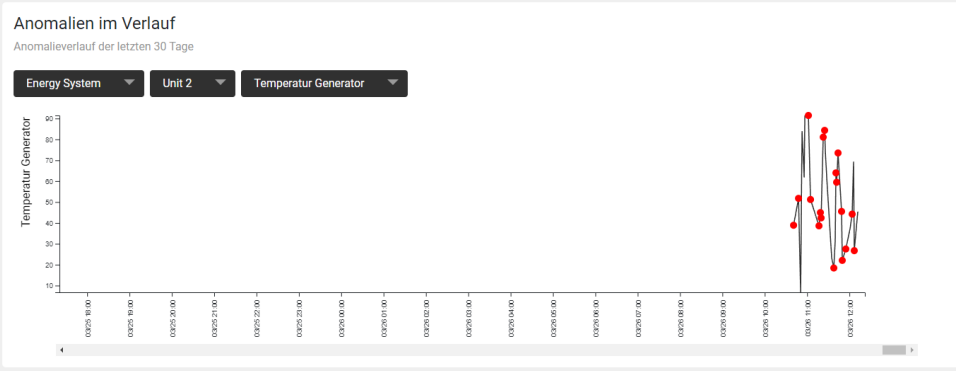


Abbildung 4: Darstellung der Verläufe an Anomalien im Frontend

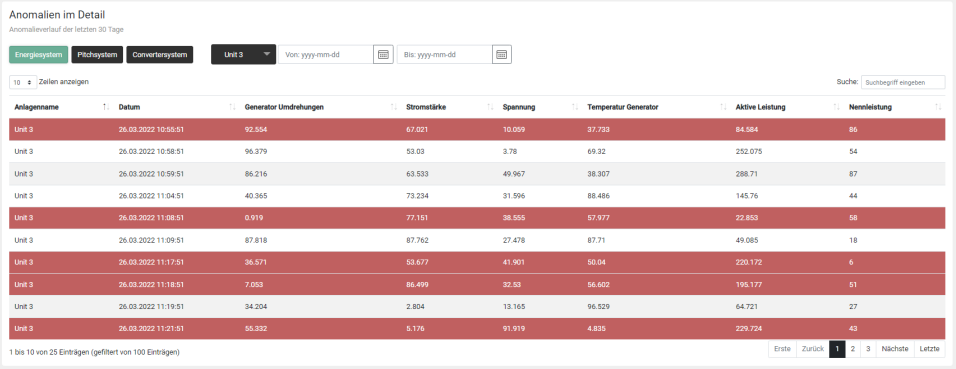


Abbildung 5: Darstellung der Anomalien als Tabelle im Frontend

4 Backend

Kernaufgabe des Backends ist die Datenstromverarbeitung, Mapping und Ausführung der ML-Modelle, sowie Persistierung und Application Programming Interface (API) Verwaltung für das Frontend.

4.1 Datenstromverarbeitung

Im Rahmen dieses Kapitels muss eine Verarbeitung eines Datenstroms stattfinden. Dazu muss in Echtzeit eine Ausführung und Persistierung der ML-Modelle mithilfe der Sensor-Daten des Datenstroms durchgeführt werden. Die Umsetzung dieses Konzepts erfolgt durch die Verkettung mehrerer Kafka-Topics mithilfe von sogenannten ML- und *Persistence*-Services. Beide Services sind Programme, die über eine Kafka-API auf bestimmte Kafka-Topics hören und darüber eingehende Daten in Echtzeit verarbeiten.

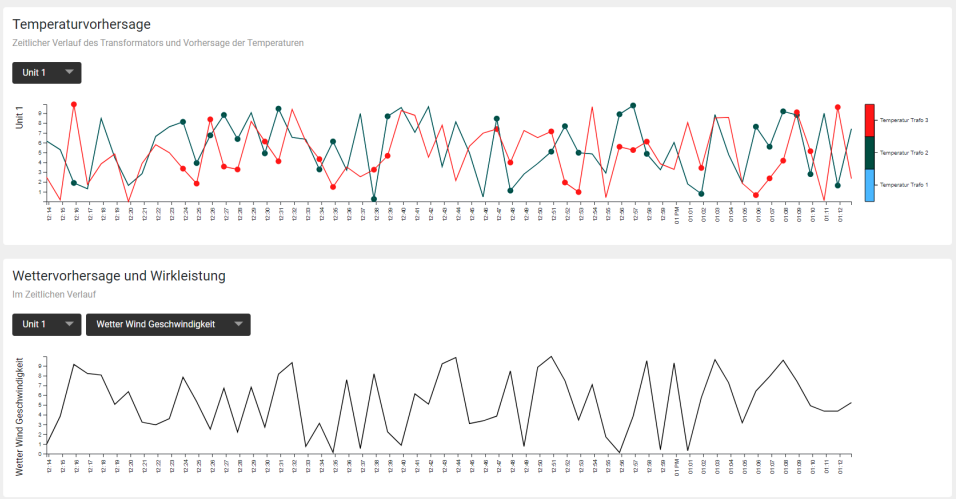


Abbildung 6: Darstellung der Fehlerfrüherkennungsseite im Frontend

Für jeden Anwendungsfall des Projektes ergibt sich eine unterschiedliche Verkettung und Verschaltung von Kafka-Topics, Persistence-Services und ML-Services.

4.2 Machine Learning-Services

Jeder ML-Service ist an einen bestimmen ML-Anwendungsfall gebunden. Dabei ist jeder ML-Service für die Kodierung der Eingangsdaten des Datenstroms in ein für den korrespondierenden Modell-Server gültiges Schema zuständig.

4.3 Persistence-Services

Der Persistence-Service ist zuständig für alle Ergebnisse, welche im Rahmen der ML-Services auf ein bestimmtes Topic schreiben. Der Persistence-Service agiert mit allen Topics, die eine Ausgabe für ein ML-Modell darstellen. Demnach stellt es das Ende einer Topic-Verkettung dar, da keine weiteren Topics verwendet werden.

4.4 Modell-Serving

Das sogenannte *Modell-Serving* ist ein zentrales aus der Architektur hervorgehendes Konzept für die Echtzeit-Ausführung von ML-Modellen. Die

Grundidee des *Modell-Servings* ist es, die Modelle über eine Representational State Transfer (REST)-API bereitzustellen, sodass pro Hypertext Transfer Protokoll (HTTP)-Anfrage die Modelle ausgeführt werden können. Im Projektkontext werden zwei verschiedene Frameworks für Modell-Serving verwendet: Tensorflow-Serving und Fast-API.

Tensorflow-Serving

Tensorflow-Serving ist eine für Produktionsumgebungen ausgerichtetes Framework für die Bereitstellung von Keras-Modellen [Ten21]. Die technische Umsetzung des Tensorflow-Serving geschieht über ein von Tensorflow bereitgestelltes parametrisierbares Docker-Image, mit welchem sich Images mit beliebigen Keras-Modellen bauen lassen. Ein gestarteter Container gibt dabei einen HTTP-Endpunkt frei, an welchem die `predict`-Schnittstelle des Modells angesprochen werden kann.

Eigene Implementierung mittels Fast-API

Um die technologische Barriere zwischen den Trainings und der Bereitstellung von Modellen aufzulösen, wurde sich für die Implementierung des Modell-Servers mithilfe des Web-Frameworks Fast-API entschieden. Grundvoraussetzung für das verwendete Web-Framework ist die Unterstützung von Python. Das ermöglicht maximale Flexibilität bei der produktiven Bereitstellung der Anwendung, da keine direkten Abhängigkeiten zu den verwendeten Bibliotheken oder dem Modell-Format (z.B. `.joblib`, `.pb`) existieren. Demnach können trainierte Modelle aus dem Jupyter-Notebook in eine dafür kompatible Python-Umgebung geladen werden. Die Integration in die Produktivumgebung erfolgt nach dem Schema des Tensorflow-Servings. Dazu wird das Modell innerhalb eines Docker-Containers hochgeladen, dessen `predict`-Schnittstelle über eine REST-API für die Modellausführung zur Verfügung gestellt wird.

5 Machine Learning

Das folgende Kapitel behandelt die Machine Learning Verfahren, die bei der Erreichung des Projektziels angewendet wurden. Dabei ist das Gebiet des Machine Learning in drei Teilbereich aufgeteilt worden. Diese sind die

Anomalieerkennung, die Fehlerfrüherkennung und die Zustandsüberwachung. Jeder dieser Teilbereiche verwendet verschiedene Methoden und Algorithmen, um die Anforderungen umzusetzen.

5.1 Anomalieerkennung

Anomalieerkennung hat zum Ziel Unregelmäßigkeiten und Ausreißer in großen Datenmengen zu finden. Es gibt eine Vielzahl an möglichen Algorithmen, die auf verschiedenen mathematischen Methoden basieren. Die Herausforderung im WiPMA Projekt ist, dass eine Unterteilung in normale und nicht-normale Daten im Vorfeld nur eingeschränkt möglich ist. Das bedeutet, dass die Ergebnisse der Modelle nicht oder nur grob überprüft werden können und supervised Lernalgorithmen somit ungeeignet sind. Daher werden Algorithmen verwendet, die kein Labeling der Daten voraussetzen, dasss wiederum eine Bewertung der Ergebnisse durch den Anwender und dessen Verständnis über die WEA notwendig macht.

5.1.1 Unsupervised Ansätze

Es werden für die drei Subsysteme Energie, Pitch und Umrichter jeweils zwei Algorithmen zur Erkennung von Punktanomalien bzw. Unregelmäßigkeiten in Zeitreihen verwendet. Diese sind ein Isolation Forest und eine One-Class Support Vector Machine. Als Framework wird PyOD eingesetzt [ZNL19]. Insgesamt entstehen so sechs individuell trainierte Modelle, die an das Backend gegeben und dort auf neue Daten angewendet werden. Es handelt sich hierbei um Methoden zur multivariaten Anomalieerkennung in Zeitreihen.

Die als CSV-Datei bereitgestellten Daten werden als Pandas Dataframe eingelesen und die entsprechenden Features nach Subsystem ausgewählt. Die sekündlichen Rohdaten werden pro Datenzeile (pro Zeitstempel) nach der L^2 Norm normalisiert und exklusiv des Zeitstempels in das Modell gegeben.

Für den Isolation Forest und die One-Class Support Vector Machine (OCSVM) wird das Backend von Scikit-Learn von PyOD verwendet [LL18]. Es gibt einige Modellparameter, die für das Training angepasst werden können. Die standardmäßigen Belegungen können der Dokumentation von [ZNL19] entnommen werden. Es wurde die Annahme getroffen, dass ca. 5% der Daten

Anomalien enthalten (auch, wenn keine Fehlercodes vorliegen). Dies wird den Modellen mit *contamination=0.05* mitgegeben.

Für die Anzeige im WiPMA Dashboard wird pro Subsystem der Schnitt der erkannten Anomalien beider Modelle bestimmt und nur diese Punkte als Anomalie markiert.

Die schlussendliche Interpretation der erkannten Anomalien fällt dem Anwender des Dashboards zu. Die Ergebnisse der Modelle lassen sich in diesem Fall nicht analytisch evaluieren, da nicht bekannt ist, ob und wo in den Daten (abgesehen von den Fehlercodes) Anomalien auftreten. Mit diesen Modellen lassen sich Hinweise auf Unregelmäßigkeiten erkennen. In Abhängigkeit von der *contamination* wird der entsprechende Anteil auffälliger Datenpunkte aufgezeigt und kann vom Anwender näher betrachtet werden.

5.1.2 Normal behavior model

Das Ziel mit diesem Ansatz ist die Klassifizierung eines Tages in die Kategorien normal, leicht oder stark anomal. Zu diesem Zweck werden Modelle benötigt, die Werte eines Attributes mit hoher Genauigkeit vorhersagen können [LBF20]. Dieser Ansatz ist die Basis für einen Teil der Anomalieerkennung und soll dem Planer bei einer vorausschauenden Planung helfen.

Im Normal Behavior Model (NBM)-Ansatz werden Regressionsmodelle zur Vorhersage bestimmter Attribute, im Folgenden auch *Target-Channel* genannt, trainiert. Somit sollen die Modelle für jeden Target-Channel aus ausgewählten Eingangs-Features den entsprechenden und als normal angenommenen Wert des Target-Channels vorhersagen. Diese Vorhersagen können dann mit den eigentlichen Wert des Target-Channels verglichen werden.

Da die Annahme getroffen wurde, dass die Modelle das normale Verhalten abbilden, kann von einem fehlerhaften bzw. anomalen Verhalten gesprochen werden, wenn die beiden Werte stark voneinander abweichen [LBF20, S. 1-2]. Die eigentliche Methodik ist in Abbildung 7 abgebildet.

Für die Evaluation des NBM Ansatzes wurden die Daten der Anlage A1 verwendet. Die Trainingsdaten umfassen die Daten aus dem Jahr 2020 und die Testdaten die Daten aus dem Jahr 2021. Die Trainingsdaten wurden wie in [LBF20, S. 3–4] beschrieben aufbereitet. Die Testdaten wurden in zehn-

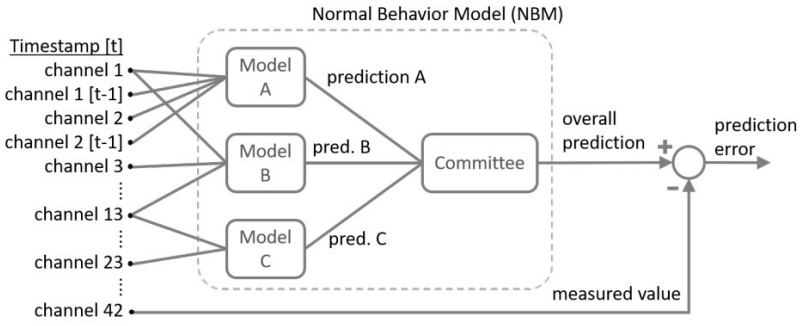


Abbildung 7: Methodik von NBM [LBF20, S. 2]

Minuten-Intervallen gemittelt, normalisiert und verschoben. Details zu den einzelnen Schritten kann [LBF20, S. 3–4] entnommen werden.

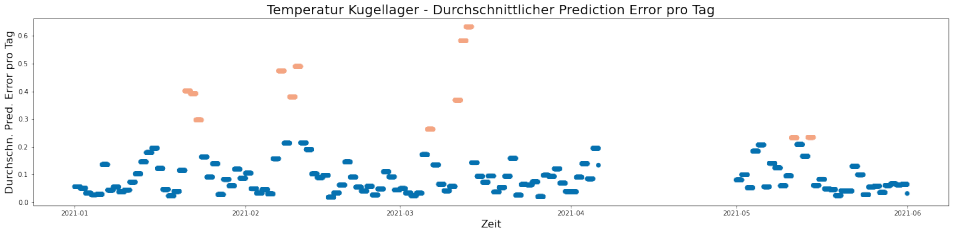


Abbildung 8: Prediction Error Tagesdurchschnitt für das Zielattribut *Temperatur Kugellager*, eingefärbt nach der Anomalieschwere

Abbildung 8 zeigt den durchschnittlichen Prediction Error pro Tag für das Zielattribut *Temperatur Kugellager*. Dort ist zu erkennen, dass es zwölf Tage gibt, die mit einer Anomalischwere von eins eingestuft wurden. Dies weist auf eine leichte Anomalie hin. Der durchschnittliche Predictions Error dieser Tage ist deutlich erhöht.

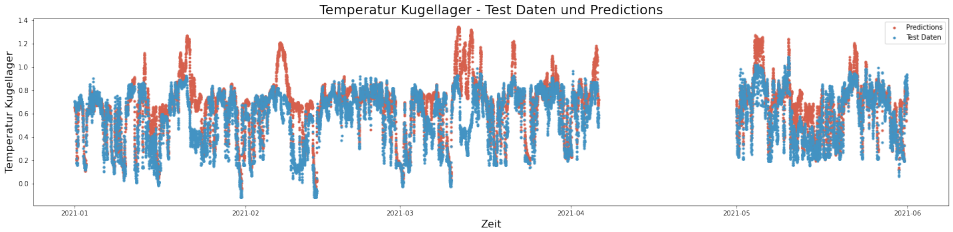


Abbildung 9: Testdaten und Predictions für das Zielattribut *Temperatur Kugellager*

Zusammenfassend kann also gesagt werden, dass der Ansatz Anomalien und vor allem ein von Anomalien betroffenes Attribut erfassen kann.

5.2 Fehlerfrüherkennung

Vorhersage Überhitzung des Transformators

Bei der explorativen Datenanalyse der einzelnen Fehlercodes ist bei dem Fehler, welcher für eine Überhitzung des Transformators steht, ein Muster kurz vor dem Auftreten dieses Fehlers aufgefallen.

Um eine bevorstehende Überhitzung des Transformators zu erkennen, werden zunächst für die Anomalieerkennung die Algorithmen Isolation Forest, OCSVM und Autoencoder verwendet. Die Modelle erkennen den Punkt, an dem die Temperaturen beginnen voneinander abzuweichen.

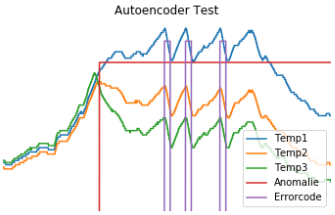
Läuft die Anlage ab diesem Punkt unter Volllast weiter, tritt eine Überhitzung sowie eine Abschaltung auf. Um dies frühzeitig zu erkennen, wird ein Vorhersagemodell für die Last der Anlage und somit die Temperaturen des Transformators entwickelt. Dieses Modell nutzt eine Windvorhersage über eine externe Wetter-API, da die Wirkleistung direkt vom Wind abhängt. Als Algorithmus wird ein Random Forest verwendet.

Für die Anomalieerkennung werden die Daten der drei Temperatur-Features pro Phase nach der L^2 Norm normalisiert. Ansonsten werden die sekündlichen Daten unverändert als Zeitreihe in die Modelle gegeben. Die Ergebnisse sind in Abbildung 10 gezeigt.

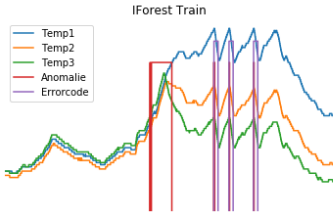
Im Backend des Systems wird die Vorhersage der Wirkleistung dann ausgeführt, wenn alle drei Modelle für mindestens eine Minute jeden Datenpunkt als Anomalie klassifizieren.

Für die Vorhersage der Überhitzung lässt sich der zeitliche Abstand der Last auf die Temperaturen des Transformators berechnet und ergibt -42 Minuten. Da die Temperaturwerte eine 42 minütige Verzögerung besitzen, wird für das Modell nicht nur die aktuelle Windgeschwindigkeit als Eingabe genutzt, sondern auch der vorherige Verlauf der letzten Minuten. Der Random Forest Regressor mit höchsten R2-Score von 0,69 nutzt die vergangenen 55 Minuten als Eingabe.

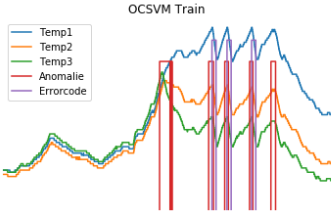
In Abbildung 11a ist der historische Verlauf der Temperaturen inklusive der Überhitzung und der Abschaltung um 9:51 Uhr, 11:08 Uhr und um 13:16 Uhr zu sehen. Die Prognose der Temperaturen, basierend auf den



(a) Ergebnis des Autoencoders.

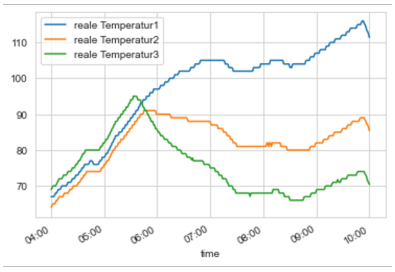


(b) Ergebnis des Isolation Forests.

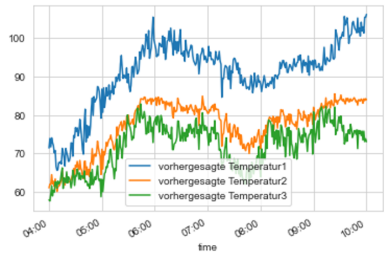


(c) Ergebnis der OCSVM.

Abbildung 10: Ergebnisse der Anomalieerkennung auf den Temperaturdaten des Transformators.



(a) Historische reale Temperaturen des Transformators



(b) Vorhergesagte Temperaturen des Transformators

Abbildung 11: Historische und vorhergesagte Temperaturen des Transformators mit einem Random Forest Regressor

historischen Windgeschwindigkeiten, ist in Abbildung 11b dargestellt. Der R2-Score für diesen Zeitraum ist 0,71. Die prognostizierten Temperaturen geben bis zur ersten Abschaltung die realen Temperaturen annähernd gut wieder.

Mit der Kombination aus Anomalieerkennung und Vorhersage ist es möglich, eine Überhitzung des Transformators bereits einige Stunden vor Auftreten zu erkennen. Dadurch kann auf dem Dashboard eine Warnung angezeigt werden und der Anwender kann über eventuelle frühzeitige Gegenmaßnahmen entscheiden.

5.3 Zustandsüberwachung

Die grundlegende Aufgabenstellung der Zustandsüberwachung ist eine Klassifizierung des aktuellen Zustands. Für eine Klassifizierung sind fest definierte Klassen erforderlich, weshalb Klassifizierzungsalgorithmen in den Bereich des supervised Machine Learning einzuordnen sind. Die Projektgruppe hat sich bei der Klassifizierung für einen Entscheidugsbaum entschieden. Entscheidungsbäume sind gerichtete Bäume, in den mithilfe von Entscheidungsregeln die Klassifikation eines Datenobjektes stattfindet. Die zu Beginn geplante Nutzung eines Random- Forest-Decision-Tree Algorithmus war aufgrund fehlender Labels in den Daten jedoch nicht möglich. Daraus hat sich ergeben, eigene Klassen erstellen zu müssen. Mit Clustering Methoden sind Muster in den Daten ermittelt worden, um daraus Klassen abzuleiten. Clustering Methoden sind unsupervised und können deshalb auf nicht gelabelten Daten verwendet werden. Die Ergebnisse waren die Grundlage, um Normalzustände in unterschiedlichen Bedingungen zu definieren. Diese wurden als Labels für einen Entscheidungsbaum verwendet.

Im Entscheidungsbaum wurde mithilfe einer Performance- und einer Temperaturüberwachung die Klassifizierung des aktuellen Zustands vorgenommen. Bei der Performanceüberwachung werden die Performance Feature darauf geprüft, ob sie im Erwartungsbereich liegen. Die Grenzwerte sind die pro Cluster beobachteten Maximal bzw. Minimalwert der Feature. Bei der Temperaturüberwachung werden die Temperatursensoren einzelner Bauteile betrachtet. Die Überhitzung eines Bauteils ist meist örtlich begrenzt, woraufhin die Korrelationen zwischen den Temperatur-

sensoren sinken. Unterschreiten die Korrelationen einen Grenzwert wird die Anlage als auffällig klassifiziert. Sind sowohl in der Performance- als auch Temperaturüberwachung keine Auffälligkeiten aufgetreten wird der Normalzustand für die Anlage ausgeben.

6 Fazit

Im Verlauf den Projektes haben die Gruppenmitglieder ein System entwickelt, welches automatisiert Daten mittels eines Datenstroms einspeist und diese dann durch trainierte Machine Learning-Modelle analysiert. Diese Modelle sind im Wesentlichen auf die Erkennung von Anomalien trainiert. Die Erkennung wird für die Ermittlung von Fehlern in eingehenden Daten verwendet. Diese werden im Prototypen als simulierter Datenstrom verarbeitet. Das Modell ist konzipiert echte eingehende Daten, in Form eines Datenstroms, zu verarbeiten. Die Modelle werden auch bei der Erkennung von Anomalien in prognostizierten Daten verwendet. Des Weiteren werden Zustände von Windenergieanlage erkannt und dargestellt. Dazu wird eine Zustandsüberwachung verwendet, welche auf einem Entscheidungsbaum aufbaut. Mittels dieses Baumes werden verschiedene Zustände unterschieden.

Die gewonnen Erkenntnisse werden, ebenfalls ohne manuelle Steuerung, direkt auf einem Dashboard in einer verständlichen Weise dargestellt. Das System ist in andere Systeme integrierbar, da durch die zur Verfügung gestellten Schnittstellen auf Daten, Machine Learning-Ergebnisse oder aufbereitete Erkenntnisse zugegriffen werden kann. Außerdem ist es möglich das System zu erweitern. Die FastAPI Technologie ermöglicht eine Erweiterung und einen Austausch der bereits im Systems implementierten Modelle.

Das System, welches die PG entwickelt hat, kann das Ziel, dass im Abschnitt 1.4 definiert ist, erfüllen. Allerdings ist das System aktuell noch im Zustand eines Prototypen, der nicht unter realen Bedingungen getestet wurde. Dies sollte vor einem Einsatz unbedingt erfolgen.

Die Anforderungen wurden fast alle erfüllt. Zu den Ausnahmen gehören ausschließlich Anforderungen, die als optionale Anforderungen definiert wurden. Die Anforderung, welche nicht erfüllt wurde, ist die Implementie-

rung einer GeoMap. Daher kann das Projekt als erfolgreich abgeschlossen angesehen werden.

Für das Projekt sind der PG Daten zur Verfügung gestellt worden. Diese wurden als Datengrundlage der Machine Learning-Modelle verwendet. Aus den Daten konnten Erkenntnisse mittels der Modelle gewonnen werden.

Auch sind aus den Daten verschiedene Anwendungsfälle und unterschiedliche Attribute identifiziert worden. Die Ausarbeitung der Attribute ist dabei erfolgt, ohne dass es eine detaillierte Erklärung des Datenkontextes gegeben hätte. Somit hat das Verstehen der Daten einen große Rolle im Projekt eingenommen. Der Aufwand für das Data Understanding im Projekt hätte durch eine detaillierte Beschreibung der Daten verringert werden können.

Literatur

[Kli21] KLIMASCHUTZ, Bundesministerium für Wirtschaft u.: *Wartung*. https://www.erneuerbare-energien.de/EE/Navigation/DE/Technologien/Windenergie-auf-See/Technik/Wartung/wartung.html. Version: 2021. – letzter Zugriff: 19.09.2021

[KRM+20] KRATOCHWILL, Lisa ; RICHARD, Philipp ; MAMEL, Sara ; BREY, Michael ; SCHÄTZ, Konstantin: *Globale Trends der künstlichen Intelligenz und deren Implikationen für die Energiewirtschaft*. https://www.dena.de/fileadmin/dena/Publikationen/PDFs/2020/dena-ANALYSE_Globale_Trends_der_kuenstlichen_Intelligenz_und_deren_Implikationen_fuer_die_Energiewirtschaft.pdf. Version: 2020. – letzter Zugriff: 22.03.2022

[LBF20] LIMA, L A M. ; BLATT, A ; FUJISE, J: Wind Turbine Failure Prediction Using SCADA Data. In: *Journal of Physics: Conference Series* 1618 (2020), sep, Nr. 2, 022017. http://dx.doi.org/10.1088/1742-6596/1618/2/022017. – DOI 10.1088/1742–6596/1618/2/022017

[LL18] LUBER, Dipl.-Ing. (FH) S. ; LITZEL, Nico: *Definition - Was ist Scikit-learn?* https://www.bigdata-insider.de/was-ist-scikit-learn-a-756150. Version: 2018. – letzter Zugriff: 05.01.2022

[Ten21] TENSORFLOW: *Serving Models*. https://www.tensorflow.org/tfx/guide/serving. Version: 2021. – letzter Zugriff: 02.12.2021

[ZNL19] ZHAO, Yue ; NASRULLAH, Zain ; LI, Zheng: PyOD: A Python Toolbox for Scalable Outlier Detection. In: *Journal of Machine Learning Research* 20 (2019), Nr. 96, 1-7. http://jmlr.org/papers/v20/19-011.html