Aufgabenstellung

Das Projekt zielt auf die Entwicklung eines intelligenten Batterie-Management-Systems (iBMS) zur Optimierung der Lebensdauer von Hochvoltbatterien ab. Dieses System soll in der Lage sein, den Ladezustand (SOC), den Gesundheitszustand (SOH) und den Funktionszustand (SOF) der Batterie präzise zu bestimmen und zu optimieren.

Voraussetzungen

Die Durchführung des Projekts setzt folgende Voraussetzungen voraus:

- Verfügbarkeit von Hochvoltbatterien und Batteriemodulen.
- Zugang zu fortschrittlichen Sensordaten und Echtzeit-Datenverarbeitungskapazitäten.
- Expertise in den Bereichen Künstliche Intelligenz (KI), physikalische Modellierung und Batterietechnologie.
- Kollaborative Infrastruktur zwischen den beteiligten Partnern, wie ElringKlinger AG, Huber Automotive AG, Fraunhofer IPA und IPV der Universität Stuttgart.

Planung und Ablauf des Vorhabens

Das Projekt ist in mehrere Arbeitspakete (AP) unterteilt:

- AP 1: Definition der System- und Betriebsgrenzen. Elektrische und thermische Charakterisierung von Zellen und Modulen.
- AP 2: Entwicklung eines KI-basierten Modells zur Zustandsdiagnose. Fokus auf Temperatur-, Lade- und Alterungszustand.
- AP 3: Entwicklung eines digitalen Zwillings zur Vorhersage der Lebensdauer und Alterungsprozesse.
- AP 4: Integration und Implementierung der Algorithmen auf ein BMS-Steuergerät.
- AP 5: Aufbau eines Simulationstools für den elektrischen Antriebsstrang.
- AP 6: Test und Demonstration der entwickelten Systeme.
- AP 7: Erstellung eines Markteinführungskonzepts.

Wissenschaftlicher und technischer Stand zu Begin des Projekts

Die akkurate und verlässliche Bestimmung der aktuellen Zustände von Batteriezellen, -modulen und -systemen ist entscheidend für die Sicherheit und Langlebigkeit von Batterien. Wesentliche Betriebszustände umfassen den Ladezustand (SOC), die Temperatur, den Gesundheitszustand (SOH-R/SOH-C) und die verbleibende Restnutzungsdauer (RUL). Eine genaue Bestimmung dieser Zustände maximiert die Lebensdauer, Leistung und Kapazität der Batterie und reduziert gleichzeitig Kosten und Emissionen. Dies kann auch die Nutzerakzeptanz erhöhen und die Nutzung für Second-Life-Anwendungen erleichtern.

Der SOC wird gewöhnlich durch das Verhältnis der Restkapazität zur Nennkapazität der Zelle definiert. In vielen BMS wird der SOC mittels Stromzählermethode bestimmt, ergänzt durch eine

Korrektur anhand der Leerlaufspannung (OCV) nach der Relaxation. Diese Methode führt jedoch zu Fehlerdrift durch Messrauschen.

Der Alterungszustand wird entweder durch die abnehmende Kapazität (SOH-C) oder den Anstieg des Innenwiderstands (SOH-R) definiert. Die verbleibende Restnutzungsdauer (RUL) beschreibt die verbleibende Zeit, bis eines der Kriterien eine definierte Grenze unterschreitet. Gängige Methoden zur Bestimmung des Alterungszustands wie die differentielle Kapazitätsbestimmung, Strompulsmethode oder elektrochemische Impedanzspektroskopie erfordern meist hohen Messaufwand und spezifische Bedingungen.

Temperatur ist ein wichtiger Faktor für die Bestimmung der Zustandsparameter und Betriebsoptimierung hinsichtlich der Lebensdauer der Batterie. Ein effektives Thermomanagement hält die Zellen auf optimalen Betriebstemperaturen, beeinflusst jedoch die Leistungsabgabe und das Alterungsverhalten.

Zustände können experimentell oder durch Modelle geschätzt werden. Es gibt zwei Ansätze: physikalische und datenbasierte Modelle. Physikalische Modelle beschreiben Vorgänge durch mathematische Formeln und bieten Transparenz, während datenbasierte Modelle auf empirischen Daten basieren und weniger Rechenleistung erfordern. Maschinelles Lernen, insbesondere Künstliche Neuronale Netzwerke (KNN), eignet sich besonders für die Abbildung des Alterungsverhaltens.

KNNs sind in der Lage, Batteriezustände anhand charakteristischer Parameter oder direkt messbarer Größen wie Strom und Spannung vorherzusagen. Verschiedene Netzwerktopologien, darunter rekurrente neuronale Netzwerke (RNN) und Long Short-Term Memory (LSTM) Netzwerke, zeigen eine verbesserte Lernfähigkeit hinsichtlich der Zeitabhängigkeit des Alterungszustands.

Im Projekt werden die Vorteile beider Ansätze kombiniert: Echtzeit-Zustandsbestimmung erfolgt datenbasiert durch KNNs, während die Zustandsprognose auf physikalischen Modellen basiert. Aufgrund der hohen Rechenleistung physikalischer Modelle werden Betriebsdaten zwischengespeichert und bei Internetverbindung in die Cloud geladen, um zusätzliche Funktionen zu ermöglichen.

Dieser hybride Ansatz, der physikalische Modelle und Cloud-BMS kombiniert, wird als Digital Twin (DT) bezeichnet. Der DT nutzt historische Sensordaten zur Simulation und Vorhersage der Batterieleistung. Das Projekt zielt darauf ab, die Machbarkeit unter realen Bedingungen zu demonstrieren und industrielle Anwendungen vorzubereiten.

Zusammenarbeit mit anderen Stellen

- **ElringKlinger AG (EK):** Koordination und Bereitstellung der Batteriemodule.
- **Huber Automotive AG (HUB):** Entwicklung und Integration der BMS-Hardware und -Software.

- **Fraunhofer IPA:** Entwicklung des digitalen Zwillings und Validierung der Algorithmen.
- **IPV der Universität Stuttgart:** Entwicklung von KI-Algorithmen und Unterstützung bei der Zyklisierung der Zellmodule.

Verwendung der Zuwendung und erzieltes Ergebnis

Die Fördermittel werden verwendet für:

- Entwicklung und Validierung von KI-Algorithmen zur Zustandsdiagnose.
- Erstellung und Test eines digitalen Zwillings.
- Implementierung und Integration der Algorithmen auf einem BMS-Steuergerät.
- Durchführung von Tests und Demonstrationen unter realen Bedingungen. Die erwarteten Ergebnisse umfassen eine verbesserte Lebensdauer und Effizienz von Hochvoltbatterien sowie optimierte Batteriemanagementsysteme.

Im Falle des ipv's wurden die Fördermittel für die Serverinfrastruktur (zum trainieren der Algorithmen), Zellexperimente und Personalmittel verwendet.

Wichtigste Positionen des zahlenmäßigen Nachweises

Die wichtigsten finanziellen Positionen umfassen:

- Personalkosten für die Entwicklung und Forschung.
- Materialkosten für die Batteriemodule und Testeinrichtungen (EK und ipv).
- Kosten für die Hardware- und Softwareentwicklung.
- Kosten für Tests und Demonstrationen. Detaillierte finanzielle Aufschlüsselungen sind im PDF nicht explizit enthalten, können jedoch aus den Projektberichten entnommen werden.

Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die Arbeit ist notwendig, um die Herausforderungen der Batteriealterung und Effizienz in Elektrofahrzeugen zu bewältigen. Die geleistete Arbeit ist angemessen, da sie auf fortschrittlichen Technologien basiert und potenziell große Vorteile für die Industrie und Umwelt bietet.

Speziell für das ipv ist die Notwendigkeit der Serverinfrastruktur von großer Bedeutung um mit der zunehmenden Datenmenge nützliche wissenschaftliche Erneuerungen zu leisten.

Voraussichtlicher Nutzen

Der Nutzen des Projekts umfasst:

- Verbesserung der Lebensdauer von Batterien.
- Reduzierung der CO2-Emissionen durch effizientere Batterien.
- Senkung der Betriebskosten durch optimiertes Batteriemanagement.
- Erhöhung der Akzeptanz von Elektrofahrzeugen durch zuverlässigere Batterien.

- Potenzial für zukünftige industrielle Anwendungen und Erweiterungen.

Bekannter Fortschritt bei anderen Stellen

Der Fortschritt umfasst:

- Entwicklung und Integration von KI-Algorithmen und digitalen Zwillingen in BMS.
- Ähnliche Projekte und Studien, die zur Verbesserung von Batteriemanagementsystemen beitragen, wie IntelliBat, hyPowerRange, MiBZ und Everlasting.
- Vergleich und Weiterentwicklung bestehender Methoden zur Zustandsbestimmung von Batterien.

Erfolgte oder geplante Veröffentlichungen

Von Seitens ipv wurden zwei Veröffentlichungen herausgebracht und eine Dissertation erstellt die noch am Laufen ist.

Veröffentlichungen:

Artificial Feature Extraction for Estimating State-of-Temperature in Lithium-Ion-Cells Using Various Long Short-Term Memory Architectures

https://www.mdpi.com/2313-0105/8/4/36

A Novel Long Short-Term Memory Approach for Online State-of-Health Identification in Lithium-Ion Battery Cells

https://www.mdpi.com/2313-0105/10/3/77

Entwicklung neuer Zustandsschätzungsmodelle

In diesem Kapitel werden die neuen Algorithmen und Modelstrategien vorgestellt.

Trainingsstrategie

Dieses Kapitel erläutert den Mechanismus der benutzerdefinierten Trainingsschleife, die für alle in dieser Studie diskutierten Modelle entwickelt wurde. Angesichts der Vielfalt der Modelle ist die Trainingsstrategie nicht einheitlich; sie variiert, um den spezifischen Anforderungen jedes Modells und dem aktuellen Stand des Trainingsprozesses gerecht zu werden. Dieser adaptive Ansatz stellt sicher, dass die Trainingsmethodik optimal auf die Ziele und

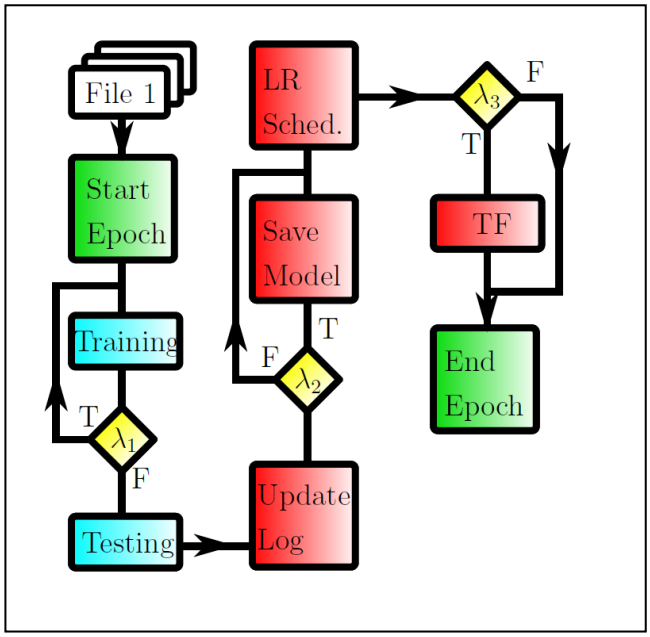
Eigenschaften jedes Modells abgestimmt ist, wodurch die Effektivität und Effizienz der Trainingsschleife verbessert wird.

Epoch-Flow

Die folgende Abbildung veranschaulicht die allgemeine Trainingsstrategie für diese Studie.

Diese Abbildung zeigt, was eine einzelne Epoche ausmacht. Der erste Schritt besteht darin, eine Reihe von Dateien für Trainingszwecke zu filtern. Nicht jeder Fahr- oder Ladezyklus wird sofort genutzt, aus zwei Gründen:

- Das Volumen der Dateien ist erheblich, daher werden nicht alle vor einem Validierungszyklus verarbeitet, um Handhabbarkeit und Effektivität zu gewährleisten.
- Dateien werden in den Arbeitsspeicher (RAM) geladen, eine begrenzte Ressource. Daher wird in jeder Epoche nur ein Teil der Dateien ausgewählt, um RAM-Kapazitätsprobleme zu vermeiden.



Nach der Auswahl der Dateien und der Bestimmung der Länge jeder zu verwendenden Datei beginnt die Epoche mit der Phase „Start Epoch“ und geht direkt in die „Training“-Phase über. Nach dem Training bewertet der Entscheidungspunkt λ_1, ob weitere Dateien für das Training vorhanden sind. Wenn mehr Dateien verfügbar sind, wird λ_1 auf 'T' (True) gesetzt und bleibt so, bis alle Dateien verarbeitet sind, woraufhin λ_1 auf 'F' (False) umschaltet. Bei Erreichen von $\lambda_1 = \mathrm{F}$ wird das Modell im „Testing“-Phase validiert. Die Validierung verwendet eine Datei aus einer Zelle, die nicht im Trainingsdatensatz enthalten ist, um die Generalisierungsfähigkeit des Modells zu bewerten. Alle Daten aus den Trainings- und Validierungsphasen werden in einer durch Komma getrennten Datei (CSV) protokolliert, um Datenverlust im Falle eines Systemausfalls zu verhindern.

Der nachfolgende Entscheidungspunkt λ_2 wird durch drei Bedingungen bestimmt:

- Wenn die Anzahl der Epochen $\mathrm{n_{epoch}}$ kleiner oder gleich 2 ist, dann $\lambda_2 = \mathrm{T}$.
- Wenn $\mathrm{n_{epoch}}$ 2 überschreitet und der Validierungsverlust *niedriger ist als frühere Verluste*, $\lambda_2 = \mathrm{T}$
- Wenn $\mathrm{n_{epcoh}}$ 2 überschreitet und es eine Änderung im Teacher Forcing-Parameter im Vergleich zur vorherigen Epoche gibt, $\lambda_2 = \mathrm{T}$.

Teacher Forcing ist eine Trainingsstrategie, die hauptsächlich in der Zeitreihenanalyse verwendet wird, insbesondere bei der Verarbeitung sequenzieller Daten. Diese Technik ist besonders vorteilhaft für Modelle, bei denen die Ausgabe eines Zeitschritts als Eingabe für den nachfolgenden Zeitschritt dient. In herkömmlichen Trainingsregimen ohne Teacher Forcing wird die Modell-Ausgabe aus einem vorherigen Zeitschritt als Eingabe für die nächste

Vorhersage recycelt. Obwohl diese Methode das Betriebsverhalten des Modells während der Inferenz widerspiegelt, kann sie die Fehler während des Trainings verschlimmern, wenn die anfänglichen Vorhersagen ungenau sind, was zu Fehlerakkumulation und -propagation durch die Sequenz führt.

Anstatt die Modell-Ausgabe wiederzuverwenden, verwendet Teacher Forcing die tatsächliche Ausgabe (oder einen Teil der tatsächlichen Ausgabe) aus dem Trainingsdatensatz im vorherigen Zeitschritt als nächste Eingabe. Diese Methode stellt sicher, dass das Modell während des Trainings genaue Eingaben erhält, wodurch die Wahrscheinlichkeit einer Fehlerausbreitung über die Sequenz verringert wird. Die Hauptvorteile dieses Ansatzes sind eine schnellere Konvergenz und eine verbesserte Stabilität während des Trainings. Diese Methode erhöht jedoch auch die Abhängigkeit von den Trainingsdaten, da das Modell durchgehend korrekte Eingaben erhält. Folglich kann das Modell Schwierigkeiten haben, sich während der Inferenz anzupassen, wenn genaue Eingaben nicht verfügbar sind.

Um diese Nachteile zu mindern, wird eine schrittweise Reduzierung des Teacher Forcing-Parameters ϑ_{TF} verwendet. Diese schrittweise Verringerung des Parameters hilft, die Lücke zwischen den Trainingsbedingungen und den Inferenzszenarien zu überbrücken, wodurch das Modell allmählich lernt, korrekte Ausgaben eigenständig zu erzeugen.

Der „LR Sched.“-Block, der für den Lernratenscheduler steht, passt dann die Lernrate ϑ_{LR} an. Entscheidungspunkt λ_3 wird erfüllt, wenn $\vartheta_{\mathrm{AR}} = \mathrm{T}$. Dies führt zum 'TF'-Block, der überprüft, ob der Validierungsverlust *weniger als 2% beträgt. Wenn dieses Kriterium erfüllt ist, wird der Teacher Forcing-Parameter* ϑ_{TF} mit folgender Gleichung aktualisiert:

$$\vartheta_{\mathrm{TF,new}} = \max(\vartheta_{\mathrm{TF}} - 0.1, 0)$$

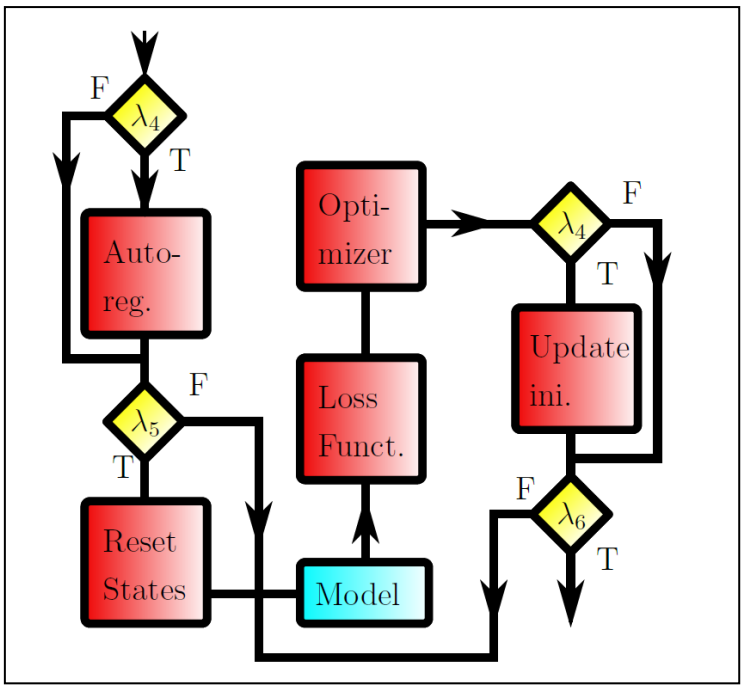
Diese Sequenz vervollständigt eine einzelne Epoche.

Trainingsalgorithmus

Dieses Kapitel bietet einen detaillierten Einblick in den „Training“-Block. Die folgende Abbildung zeigt alle Komponenten, die zum Verständnis des Trainingsalgorithmus notwendig sind.

Zu Beginn ist eine Entscheidung λ_4 erforderlich, basierend auf zwei Bedingungen:

1. $\vartheta_{AR} = T$
2. Der Teacher Forcing-Parameter für die aktuelle Epoche hat sich im Vergleich zur vorherigen Epoche geändert.

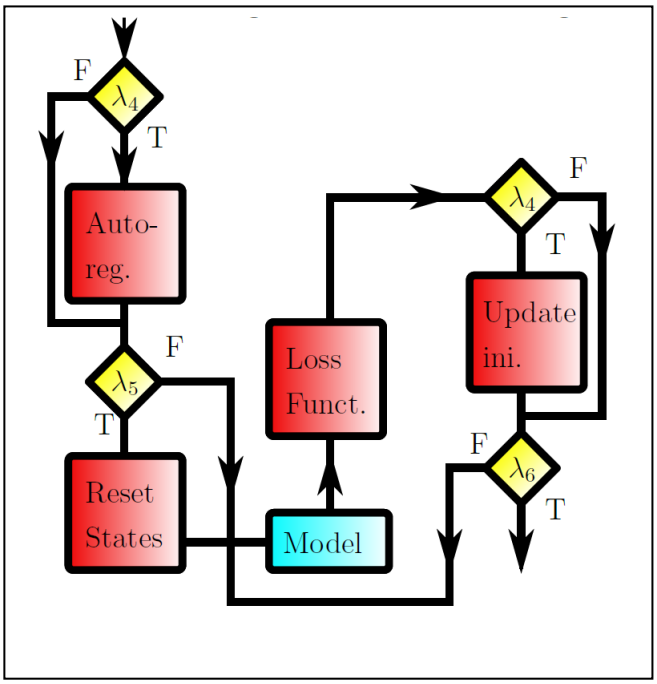


Anschließend bewertet der Algorithmus, ob das zu trainierende Modell zustandsbehaftet ist. Wenn ja, gibt $\lambda_5 = T$ an, dass die Zustände des Modells zurückgesetzt werden, wenn eine neue Datei zum Training eingeführt wird. Die Daten werden dann vom künstlichen neuronalen Netz (ANN) verarbeitet, um Vorhersagen $Y_{pred,i}$ für die entsprechenden Eingaben X_i im „Model"-Block zu erzeugen. Diese Vorhersagen werden im „Loss Function"-Block mit den tatsächlichen Werten Y_i verglichen, wobei der Trainingsverlust berechnet wird. Der berechneten Verluste werden dann an den „Optimizer"-Block weitergeleitet, wo die Modellgewichte gemäß der Backpropagationsmethode aktualisiert werden.

Eine weitere Entscheidung λ_4 bestimmt, ob der „Update Ini."-Block umgangen oder durchlaufen wird. Dieser Block passt den Initialisierungstensor X_{ini} an, indem der erste Eintrag entfernt und die neu generierte Vorhersage Y_{pred} angehängt wird. Wenn $\$\vartheta_{AR} \neq 1$, wird die Vorhersage vor der Aktualisierung des Initialisierungstensors mit folgender Gleichung modifiziert:

$$Y_{ini} = Y_{pred}(1 - \vartheta_{TF}) + Y\vartheta_{TF}$$

λ_6 überprüft, ob die Dateiverarbeitung abgeschlossen ist. Wenn $\lambda_6 = T$, endet der Algorithmus; wenn $\lambda_6 = F$, wird der nächste Teil der Datei in das Modell eingegeben.



Validierungsalgorithmus

Das Verfahren im „Testing"-Block wird in der folgenden Abbildung detailliert dargestellt.

Der Validierungsalgorithmus spiegelt den Trainingsprozess wider, mit zwei Ausnahmen:

- Der „Optimizer“-Block wird weggelassen, da während der Validierung keine Rückpropagation erfolgt.
- Der „Loss Function“-Block berechnet den Validierungsverlust *anstelle des Trainingsverlusts.*

Lernratenscheduler

Dieses Kapitel untersucht den „LR Sched.“-Block.

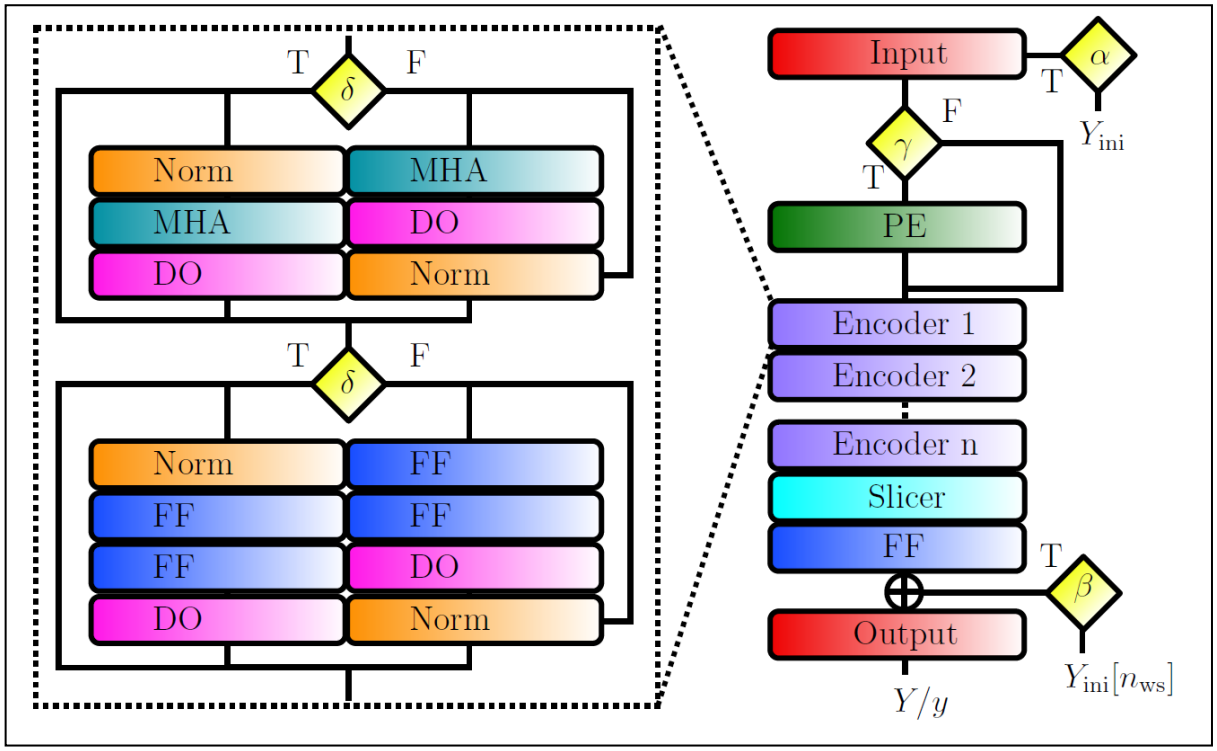
Hier werden die Verluste der letzten drei Epochen herangezogen. Sollte die Steigung der linearen Regression positive sein, wird die Lernrate verkleinert. Bei einer negativen Steigerung wird sie vergrößert um den Lernprozess zu beschleunigen.

Zeitreihenprognose

Dieses Kapitel behandelt die entwickelten Modelle zur Lösung von Zeitreihenprognoseproblemen, insbesondere die Echtzeitschätzung des Ladezustands (SoC) und der Zelltemperatur $\mathbf{T_{cell}}$. Wir untersuchen sowohl LSTM- als auch transformerbasierte neuronale Netze und betrachten verschiedene architektonische Modifikationen.

Encoder-Modelle

Dieser Abschnitt befasst sich mit der reinen Encoder-Architektur, die von transformerbasierten neuronalen Netzen abgeleitet ist. Die folgende Abbildung zeigt den schematischen Aufbau dieses Modells.



Diese Abbildung zeigt einen von oben nach unten gerichteten Datenfluss, der von der Eingangsschicht ausgeht. Diese Schicht akzeptiert Matrizen der form $\mathrm{X}[\mathrm{n_{bs}}, \mathrm{n_{ws}}, \mathrm{n_{feat}}]$. Mit $\mathrm{n_{bs}}$ repräsentative für die Batch Größe, $\mathrm{n_{ws}}$ ist die Fenstergröße und $\mathrm{n_{feat}}$ ist die Anzahl der Eingangsparameter.

Innerhalb des Encoders werden die Reihenfolge und Positionierung der Normalisierungsschichten, die als „Norm“ bezeichnet untersucht.

Mehrere Encoderschichten sind in Reihe geschaltet. Nach der letzten Encoderschicht wählt eine Slicing-Schicht den letzten Eintrag jeder Charge und jedes Merkmals des Fensters aus, wodurch die Tensorform auf $\mathrm{X}[\mathrm{n_{bs}}, 1, \mathrm{n_{feat}}]$ angepasst wird. Anschließend komprimiert ein einzelnes Neuron diese Information, um einen Output $\mathrm{Y}[\mathrm{n_{bs}}, 1{,}1]$ zu erzeugen. Für Fälle, in denen $\mathrm{n_{bs}} = 1$, vereinfacht sich die Tensorform zu $\mathrm{y}[1{,}1{,}1]$.

Darüber hinaus ermöglicht dieses Modell die Erforschung von zwei weiteren Konzepten:

- Artifical Reccurence: Dieser Ansatz untersucht die Nützlichkeit der Wiederverwendung der vorherigen Ausgabe des Modells als nächste Eingabe. In diesem Fall*, wird ein Initialvektor* $\mathrm{Y_{ini}}$ wie folgt festgelegt:

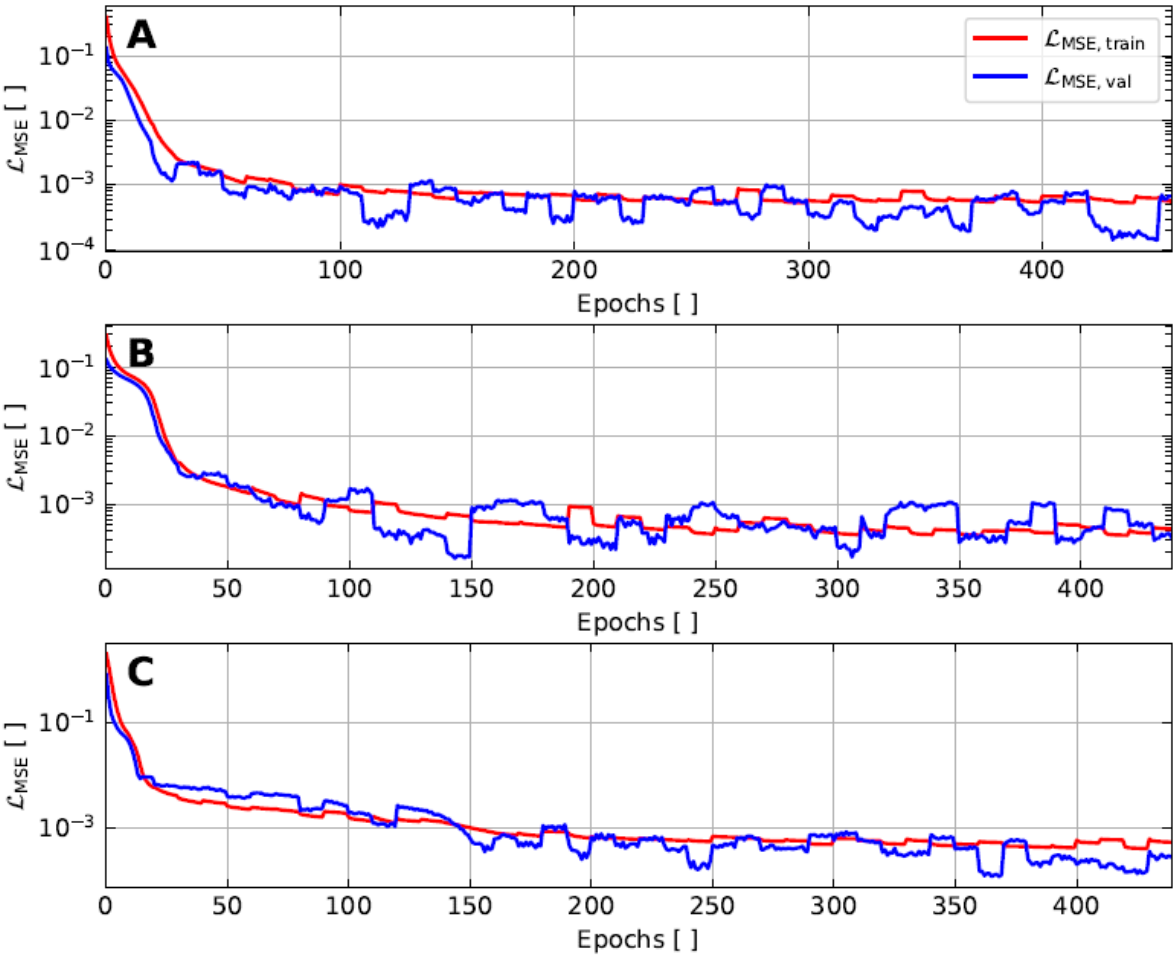
$$\mathrm{Y_{ini,i}} = \begin{bmatrix} \mathrm{y_{i-n}} \\ \vdots \\ \mathrm{y_{i-2}} \\ \mathrm{y_{i-1}} \end{bmatrix}$$

  Dieser Vektor enthält vergangene Ausgaben und wird mit der aktuellen Eingabematrix verkettet.

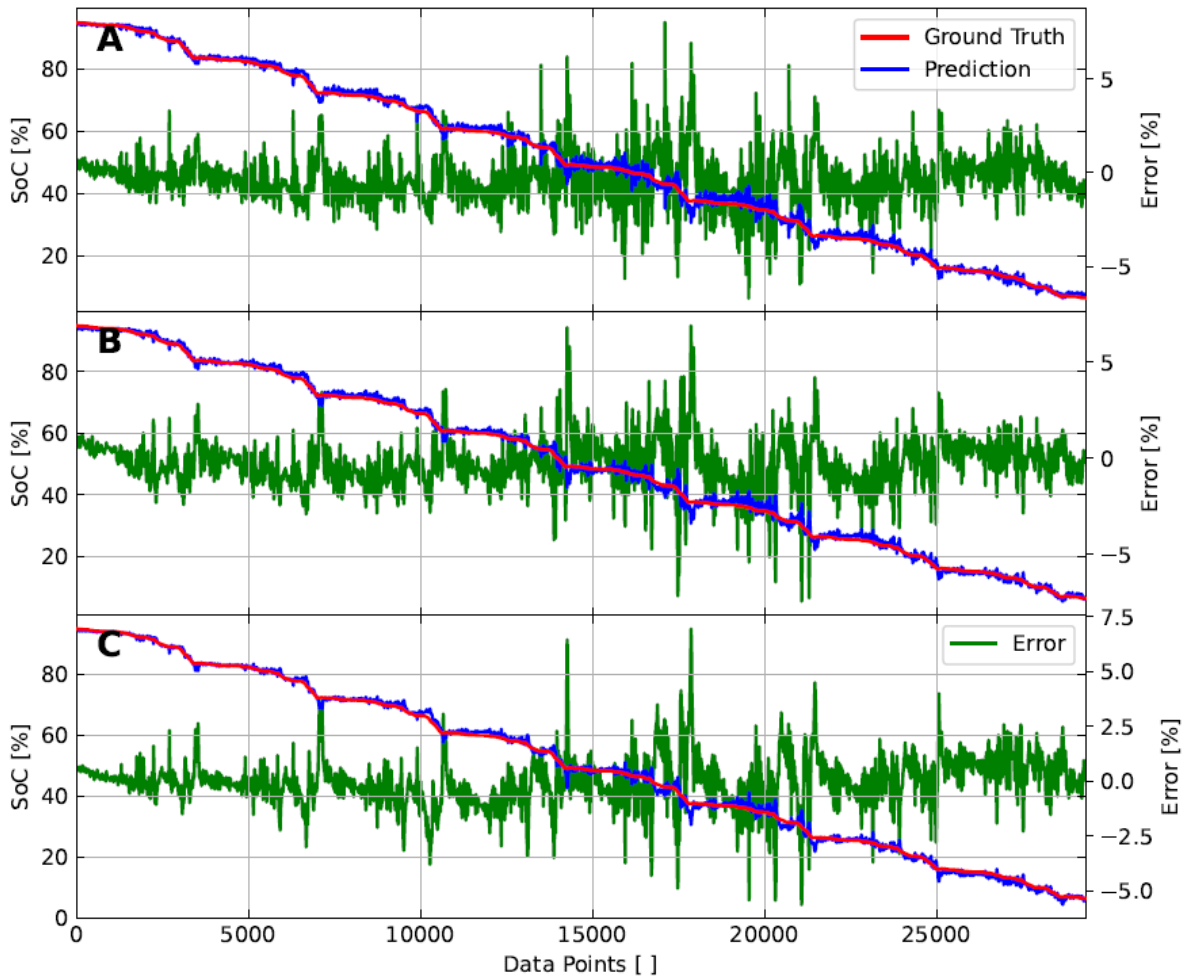
- Artifical Feature Extraction: Hier *wird das letzte Element des Initialisierungsvektors* mit der Ausgabe aus der letzten verborgenen Schicht in einer anderen dichten Schicht kombiniert, die durch ein Pluszeichen im ANN-Diagramm markiert ist. Dadurch wird das Modell so trainiert, dass es die inkrementelle Änderung der Ausgabe berechnet, anstatt ihren absoluten Wert, was seine Vorhersagegenauigkeit und Anpassungsfähigkeit erhöht.

Für beide Szenarien ist es wichtig, die sequentielle Reihenfolge während des Trainings der Eingabefenster beizubehalten. Für Modelle, bei denen $\alpha = \mathrm{F}$ und $\beta = \mathrm{F}$, ist jedoch vor dem Training das Shuffling der Fenster bevorzugt.

Weder artifical recurrence noch artifical feature extraction konnte das Base-Model verbessern. Ausserdem wurde herausgefunden, dass PE und die Skalierung von großer Bedeutung sind. Die folgende Abbildung zeigt den Lernprozess des Trainings- und Validierungsverlaufs.



In der nächsten Abbildung werden die Validierungsergebnisse präsentiert:

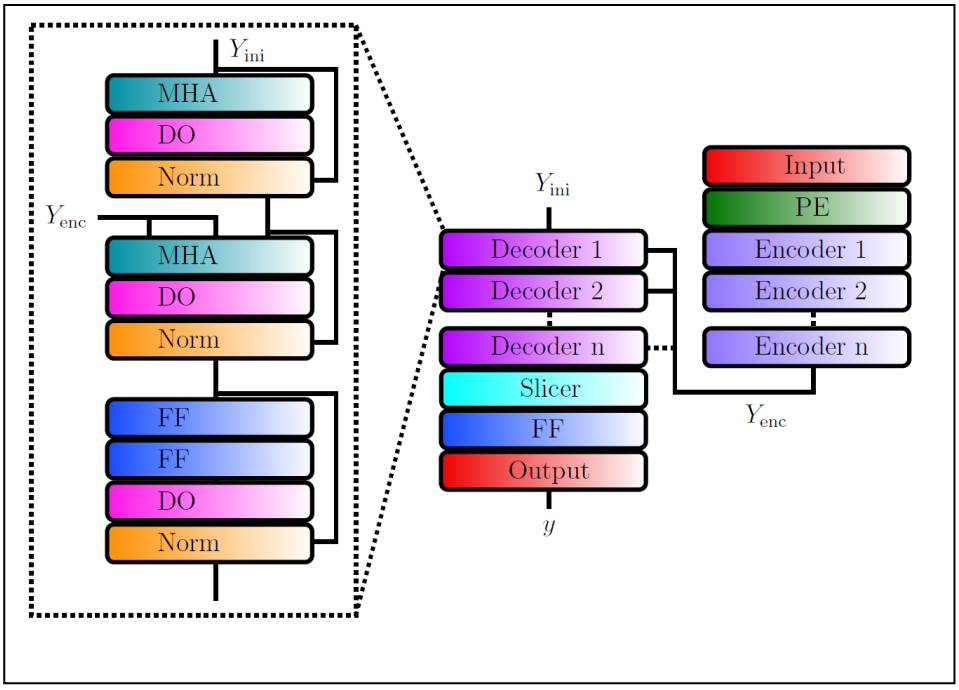


Das beste Ergebnis weicht mit nur 1,08 % von der Realität ab.

Transformermodelle

Dieser Abschnitt behandelt transformerinspirierte Modelle, die auf Encoder-Decoder-Strukturen basieren, wie in der folgenden Abbildung dargestellt.

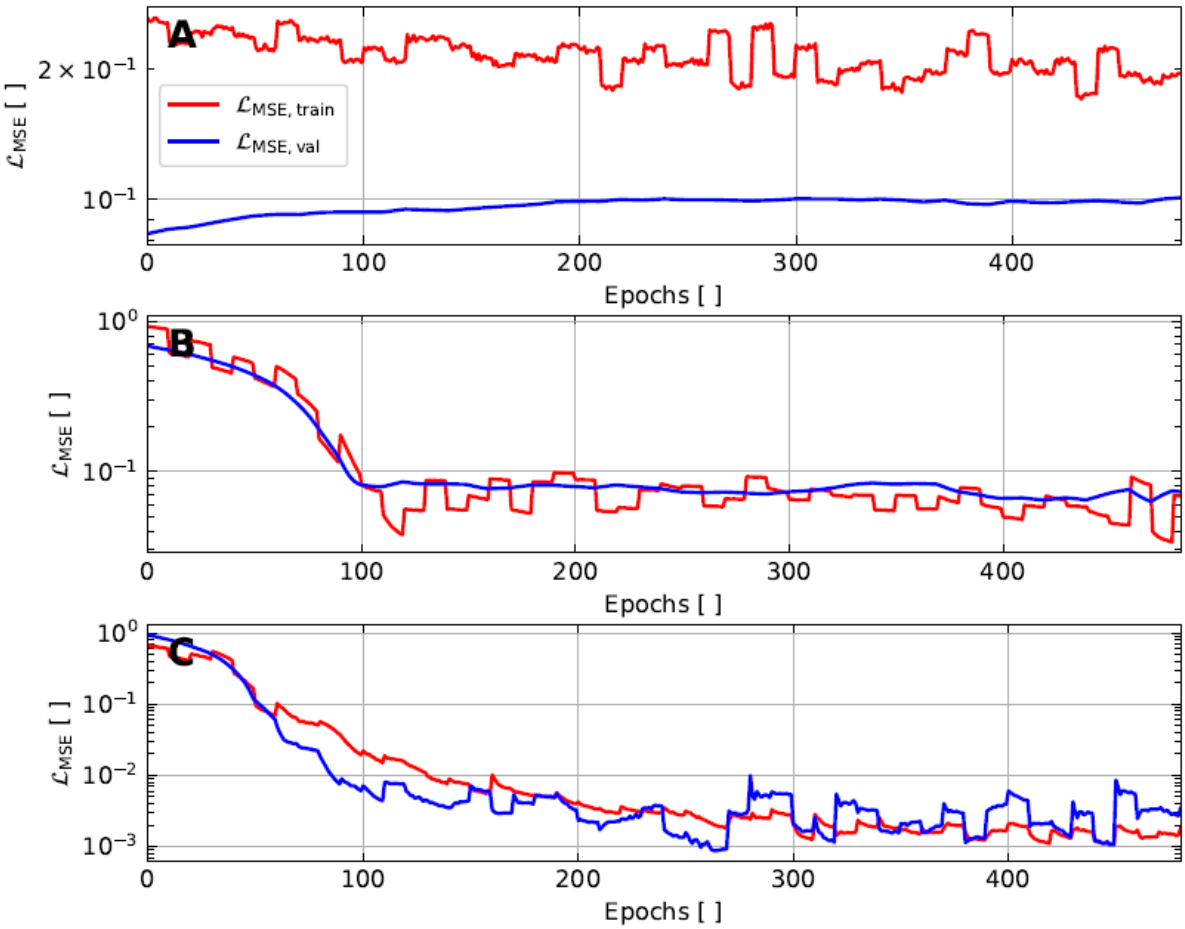
Jeder Decoder erhält zwei Eingaben: die Ausgabe aus der letzten Encoderschicht $\mathbf{Y_{enc}}$ und die Initialisierungs-ausgabe $\mathbf{Y_{ini}}$. Zunächst wird $\mathbf{Y_{ini}}$ durch einen Multi-Head-Attention(MHA)-Block verarbeitet, der eine MHA-Schicht, eine Dropout-Schicht (DO) und eine Normalisierungs-schicht (Norm) umfasst, die alle mit einer Restverbindung integriert sind.



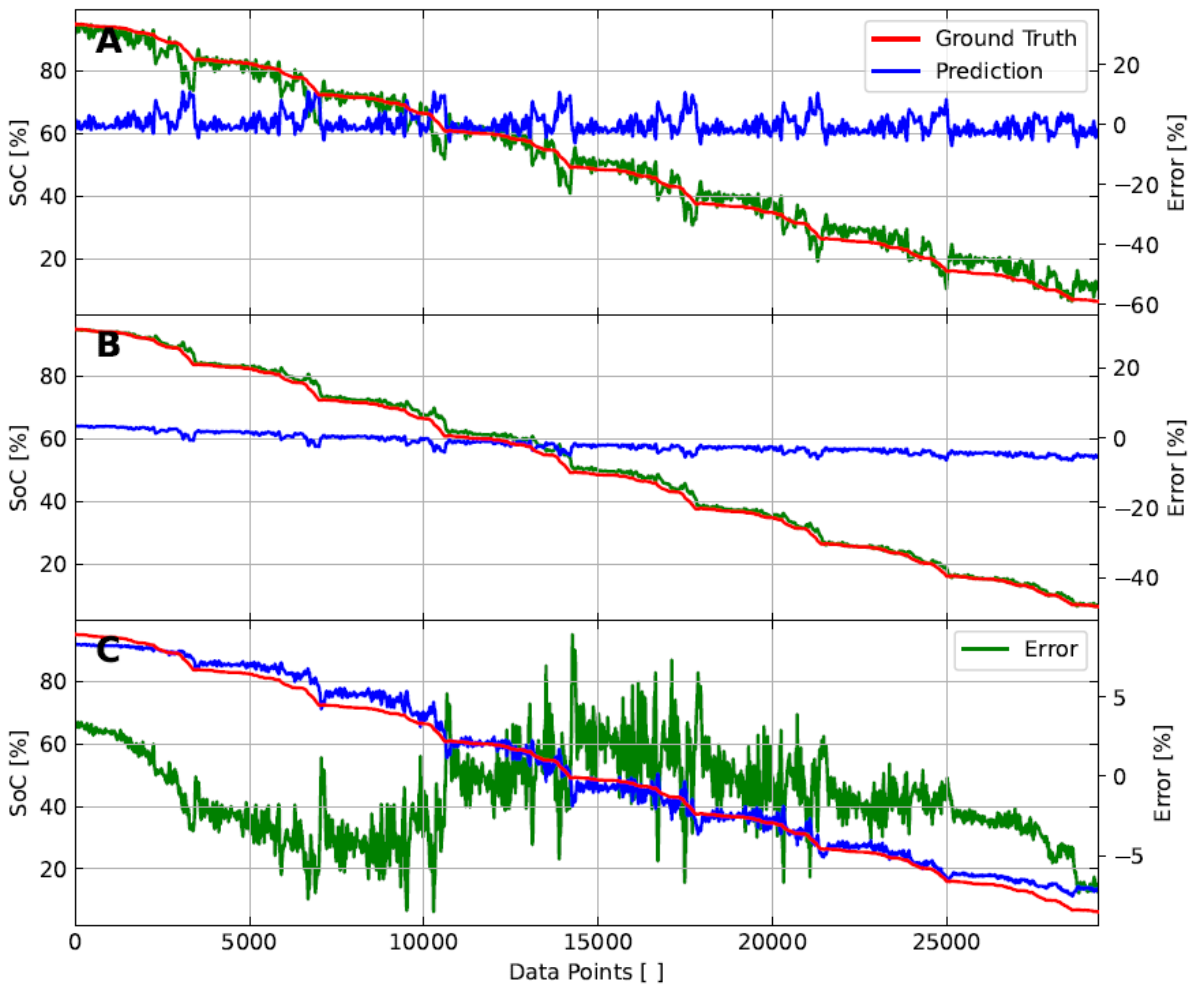
Die Ausgabe aus diesem MHA-Block wird durch eine feedforward (FF)-Schicht der Größe $\mathbf{n_{feat}}$ angepasst, um die Dimensionen von $\mathbf{Y_{enc}}$ zu entsprechen. Anschließend wird diese Ausgabe von einer weiteren MHA-Schicht verarbeitet, wobei $\mathbf{Y_{enc}}$ sowohl als Schlüssel- als auch als Wert-Eingaben dient und die transformierte $\mathbf{Y_{ini}}$-Ausgabe als Abfrage. Dieser zweite MHA-Block folgt derselben Konfiguration wie der erste, integriert jedoch eine Restverbindung nur von der Ausgabe des ersten MHA-Blocks.

Der nachfolgende FF-Block ist strukturell identisch mit der Encoder Modelle *und wird ähnlich von anderen Schichten gemäß der Standard-Transformer-Modellarchitektur* gefolgt. Dieses Modell integriert von Natur aus artifical recurence und stellt somit die Kontinuität in der Sequenz von Eingaben und Ausgaben sicher. Folglich werden nur Konfigurationen mit *berücksichtigt, und artifical feature extraction wird in dieser Modellkonfiguration nicht untersucht*.

Der trainingsverlauf der Transformer Modelle wird in der nächsten Abbildung dargestellt:



In Model A kann man feststellen, dass der trainingsverlauf wesentlich unstabiler ist. Die nächste Abbildung zeigt die Ergebnisse der Validierungsdaten.



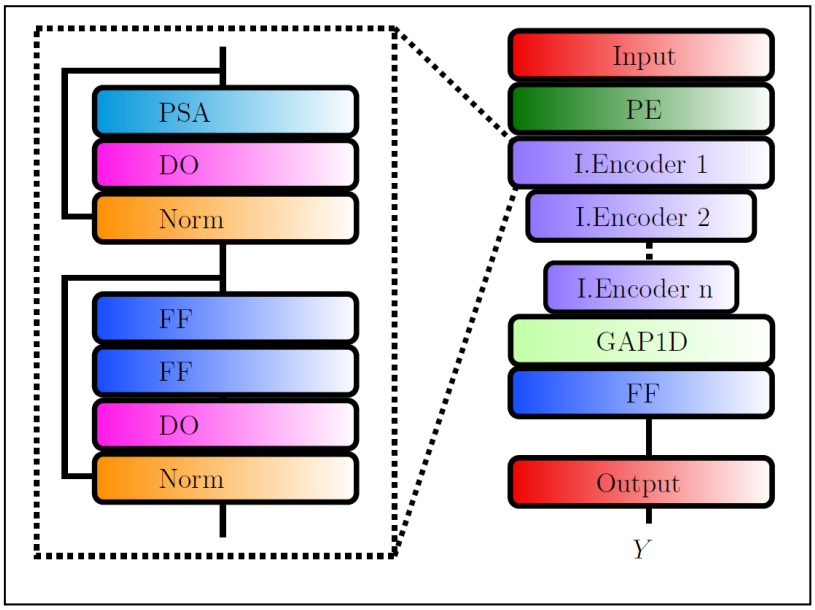
Das beste Ergebnis zeigt einen Abweichung von 2,9% zur Realität. 25,9% und 28,1% sind die Abweichungen für Model A und Model B.

Informer-Modelle

Dieses Kapitel befasst sich mit der Architektur der Informer-Modelle, diese basieren auf die der reinen Encoder-Strukturen. Diese Modelle nutzen fortschrittliche Self-Attention-Mechanismen, um die Zeitreihenprognosefähigkeiten zu verbessern. Das Schema der Modellstruktur sowie deren Modifikationen sind in der folgenden Abbildung dargestellt.

Zunächst wird die Eingabematrix durch eine Positional Encoding(PE)-Schicht transformiert, um sie für die Verarbeitung durch die Informer-Encoderschichten vorzubereiten. Diese integrieren eine ProbSparse-Self-Attention-Schicht (PSA), die Sparsamkeit in den MHA-Mechanismus einführt. Die PSA-Schicht, optimiert die Aufmerksamkeitsverarbeitung, indem sie sich auf die einflussreichsten Abfragen konzentriert:

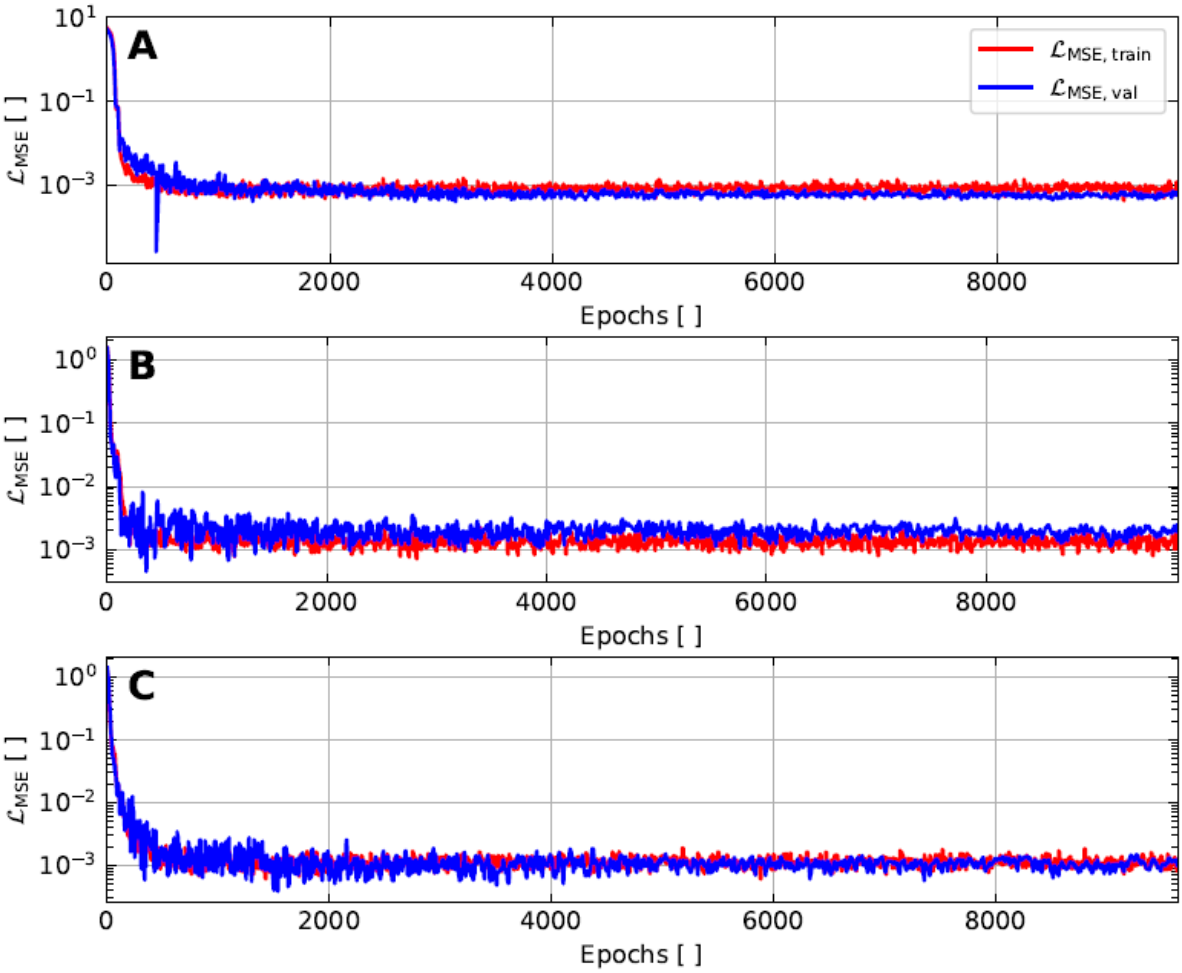
- Kopfteilung: Der Eingangsdaten werden in mehrere Köpfe unterteilt, um parallele Verarbeitung zu ermöglichen und dem Modell zu ermöglichen, verschiedene Merkmale aus den Eingabedaten zu erfassen.
- Sparsamer



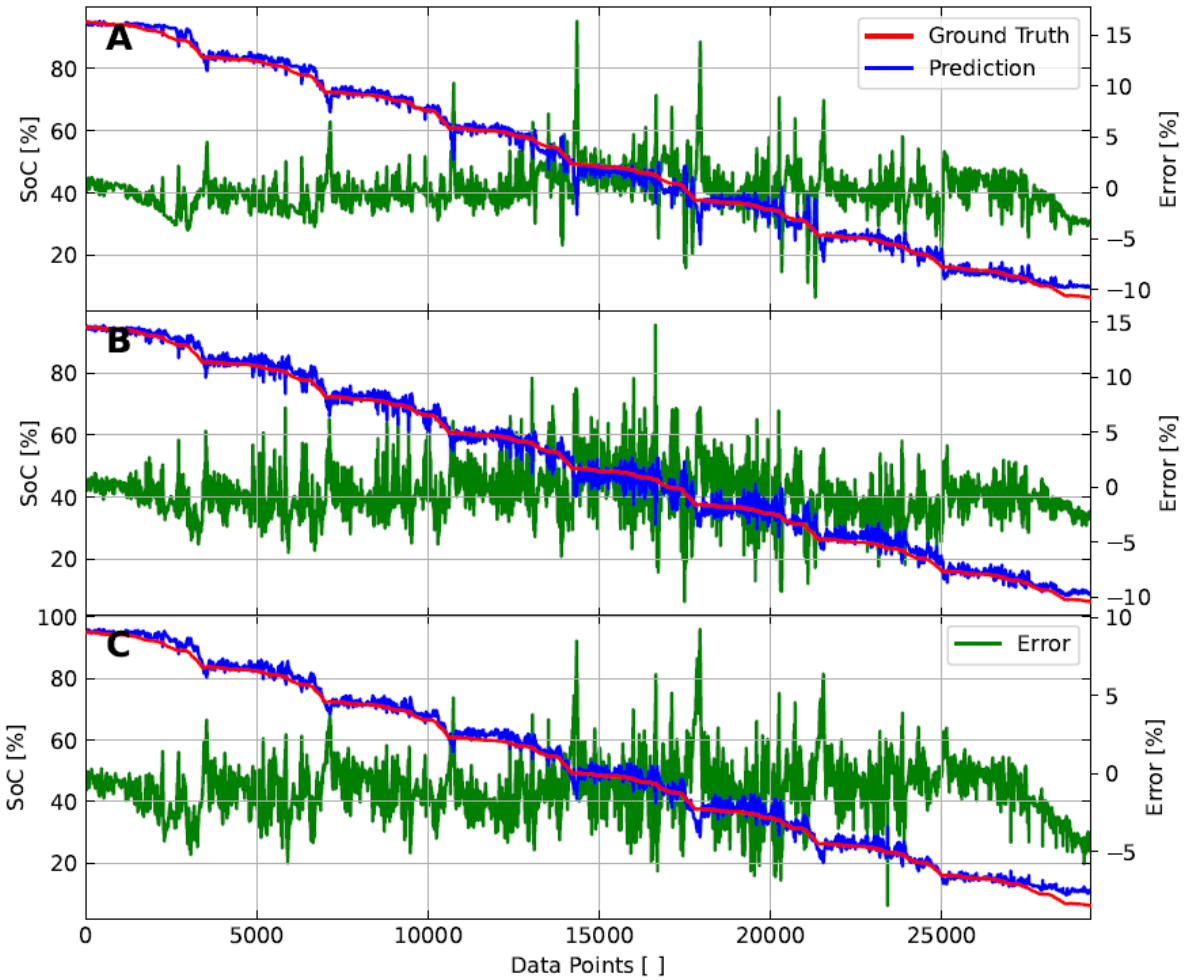
Aufmerksamkeitsmechanismus: Dieser Mechanismus verbessert die Recheneffizienz, indem er sich auf die wichtigsten Abfragen konzentriert, wodurch signifikante Informationen priorisiert und die Gesamtkomplexität reduziert wird.
- Skalierte Punktprodukt-Attention: Aufmerksamkeitswerte werden durch die Quadratwurzel der Dimension der Schlüssel angepasst, um die Stabilität der Softmax-Funktion während des Modelltrainings zu erhöhen.

Nach der Aufmerksamkeitsverarbeitung werden die Daten sequentiell durch das Netzwerk transformiert, wobei die Konsistenz mit den Eingabedimensionen beibehalten und ein nahtloser Informationsfluss gewährleistet wird. Jede Encoderschicht verfeinert die Ausgaben schrittweise und verbessert die Fähigkeit des Modells, komplexe Muster in den Daten zu erkennen. Die Dimensionalität der Sequenz wird durch Halbierung der Fenstergröße n_{ws} in jeder aufeinanderfolgenden Schicht reduziert. Die Ausgabe aus der letzten Encoderschicht durchläuft eine eindimensionale globale Mittelungsschicht (GAP1D). Anschließend konsolidiert eine FF Schicht die extrahierten Merkmale in einen einzigen Ausgabewert y pro Durchlauf, wodurch die Ausgabematrix Y entsteht. Diese Struktur stellt sicher, dass das Modell die zeitlichen Dynamiken in den Eingabedaten effizient erfasst und synthetisiert.

Die folgende Abbildung zeigt den Trainings- und Validationsverlauf der Informer Modelle.

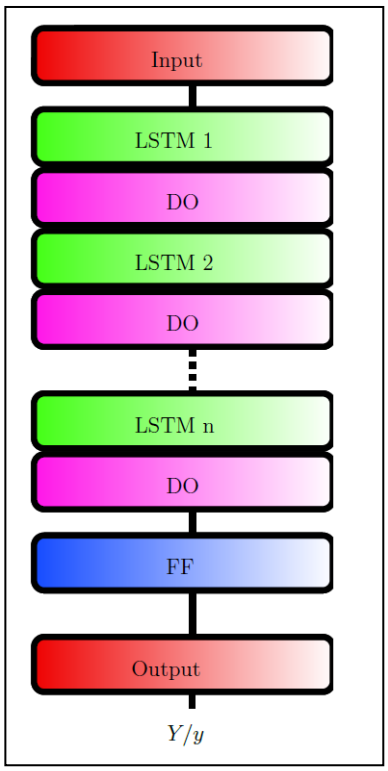


Das Training ist vergleichbar mit den Encoder Modellen was die Stabilität angeht. Die folgende Abbildung zeigt die Ergebnisse:



Das beste Modell konnte eine Abweichung von 1,96% erreichen.

Long Short-Term Memory-Modelle



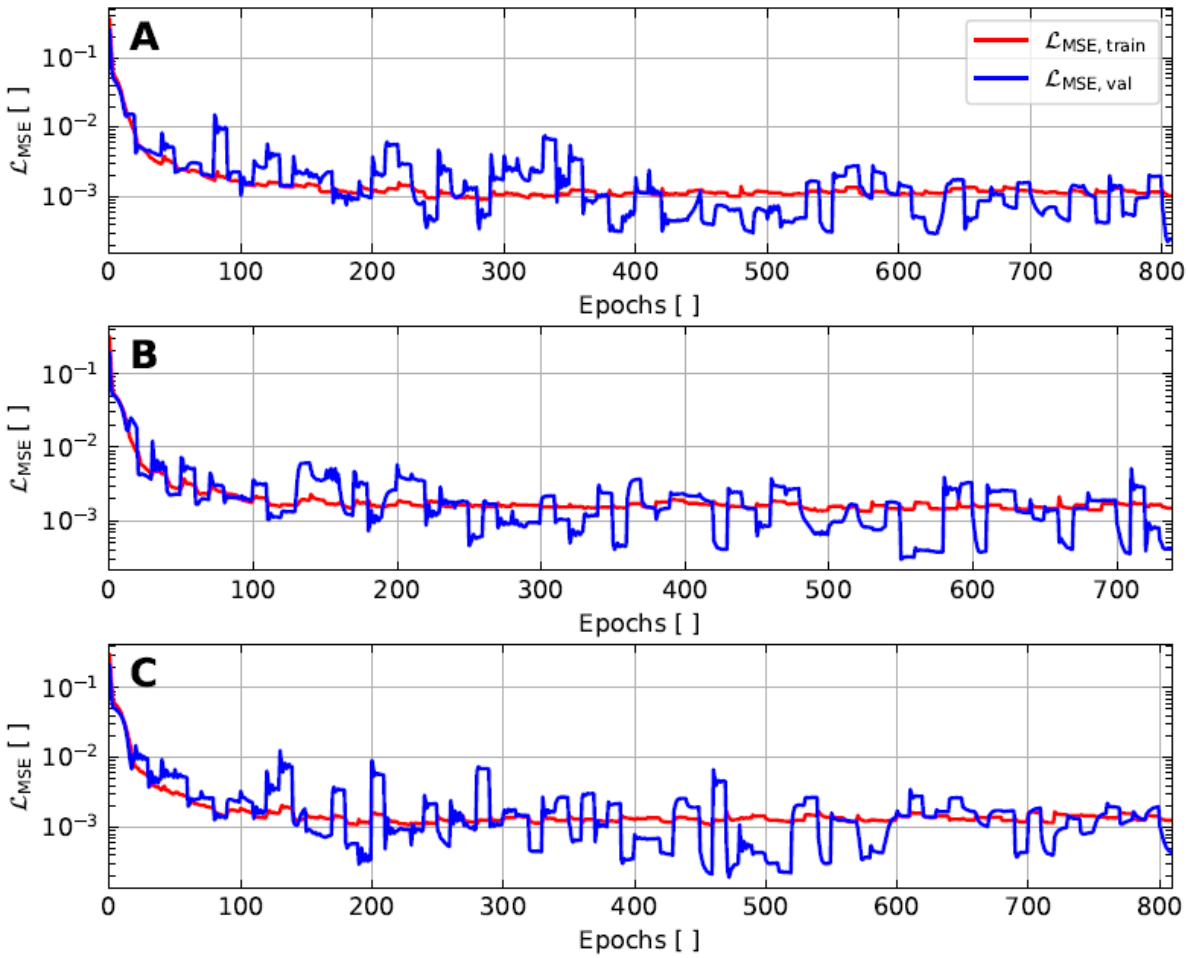
Dieser Abschnitt konzentriert sich auf Long Short-Term memory-Modelle (LSTM) zur Zeitreihenprognose, die zentral für die Bewältigung von Zeitreihenvorhersageaufgaben sind. LSTM-Modelle basieren auf LSTM-Einheiten, die darauf ausgelegt sind, langfristige Abhängigkeiten in Sequenzdaten zu erfassen.

In diesen Modellen durchläuft die Eingabematrix mehrere Schichten, die jeweils aus einer LSTM-Einheit gefolgt von einer Dropout-Schicht bestehen. Diese Konfiguration ist darauf ausgelegt, das Risiko von Überanpassung zu verringern, indem während des Trainings zufällig eine Teilmenge der Merkmale weggelassen wird. Die Ausgabe aus der letzten LSTM-Dropout-Schicht wird in eine FF-Schicht mit einem einzigen dichten Neuron eingespeist. Dieses Neuron aggregiert die verarbeiteten Informationen zu einem einzigen Ausgabewert für jeden Durchgang und bildet letztendlich die Ausgabematrix.

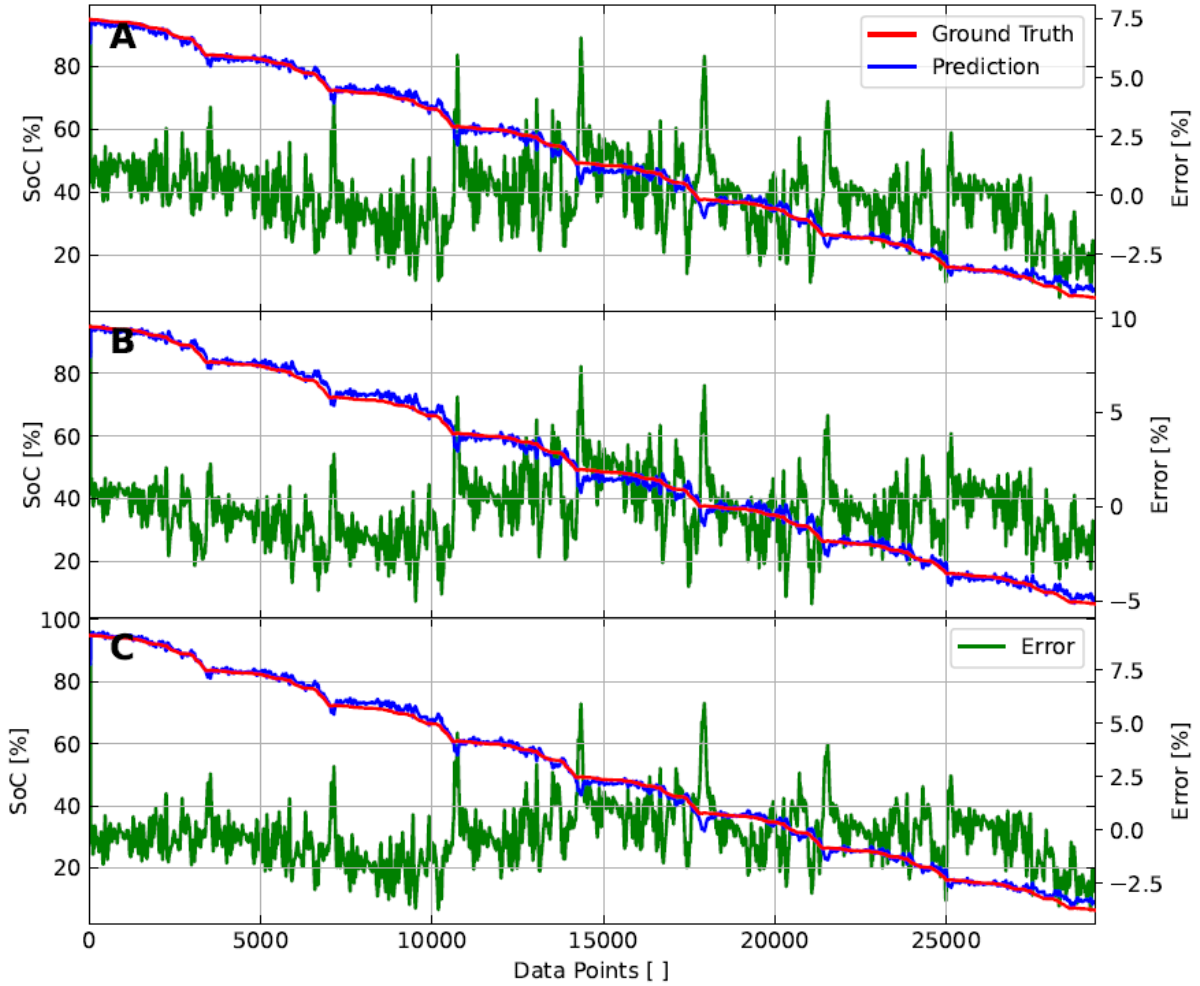
Für zustandsbehaftete LSTM-Architekturen ist es unerlässlich, die chronologische Sequenz der Eingabedaten über die Chargen hinweg beizubehalten, um die zeitliche Kohärenz während des Trainings zu gewährleisten. Diese Konfiguration ermöglicht es dem Modell, effektiv aus dem vollständigen historischen Kontext der Sequenz zu lernen. Im Gegensatz dazu behalten zustandslose Modelle keine Zustandsinformationen zwischen den Chargen bei, wodurch das Shuffling der Fenster vor dem Training ermöglicht wird, was die Robustheit des Modells verbessern kann, indem Abhängigkeiten von der Reihenfolge der Sequenzen verhindert werden.

Diese Struktur ermöglicht es dem LSTM-Modell, verschiedene Zeitreihenvorhersageaufgaben durch effektive Nutzung sowohl kurz- als auch langfristiger zeitlicher Abhängigkeiten in den Daten zu bewältigen.

Der Trainings- und Validierungsprozess ist in der nächsten Abbildung dargestellt:



Wie zu sehen ist, das trainieren der LSTM-Modelle sehr stabil. Die folgende Abbildung zeigt die Ergebnisse.



Das beste Ergebnis wurde mit 0,744% erreicht.

Klassifizierungsmodelle

Im Rahmen unserer Forschung verwenden wir die Restkapazität als Indikator für den Gesundheitszustand von Lithium-Ionen-Batterien. Traditionell erfolgt die Gesundheitsschätzung durch vordefinierte Lade-/Entladezyklen unter Laborbedingungen, wobei der Gesundheitszustand in Zwischenuntersuchungen analysiert und die Daten für die Modellierung verwendet werden. Diese herkömmliche Methode hat jedoch mehrere Nachteile:

- Unrealistische Bedingungen: Vordefinierte Lade-/Entladezyklen unter Laborbedingungen spiegeln nicht die tatsächlichen Einsatzbedingungen wider.

- Beschränkte Anwendbarkeit: Ein Vorhersagemodell sollte auch gealterte Zellen untersuchen können.

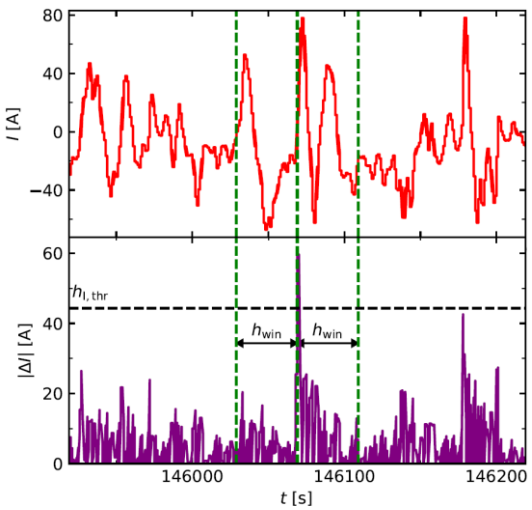


Abbildung 3: Pulsfilter

- Komplexitätsprobleme: Es ist schwer zu gewährleisten, dass das Modell nicht lediglich einen komplexeren Coulomb-Counter darstellt.

Aufgrund dieser Probleme haben wir uns von der traditionellen Methode abgewandt und eine alternative Entwicklungsstrategie gestartet. Unsere Methode verwendet die Daten der Untersuchung ausschließlich, um die Fahrzyklen korrekt mit dem Gesundheitszustand zu beschriften. Anschließend analysieren wir die Strompulse in den Fahrzyklen, wie in Abbildung 3 dargestellt.

Abbildung 3: Besteht aus zwei Diagrammen. Das obere Diagramm zeigt einen Ausschnitt aus dem Stromprofil eines Fahrzyklus, während das untere Diagramm den absoluten Wert der Stromänderung (ΔI) darstellt, berechnet nach der Formel:

$$\mathbf{\Delta I = I_i - I_{i-1}}$$

Wobei $\mathrm{I_i}$ der Stromwert am Datenpunkt i und $\mathrm{I_{i-1}}$ der Stromwert am Datenpunkt $\mathrm{i-1}$. er Stromwert am vorherigen Datenpunkt ist. Wenn eine festgelegte Schwelle $\mathrm{h_{I,thr}}$ überschritten wird, wird dieser Datenpunkt zur weiteren Analyse herausgefiltert. Ein weiterer Hyperparameter $\mathrm{h_{win}}$ bestimmt, wie viele Datenpunkte vor und nach dem Puls für die Zeitreihe verwendet werden.

Auf dieser Basis haben wir ein künstliches neuronales Netz (KNN) entwickelt, das die Spannungsantwort auf den Strompuls analysiert, um den Gesundheitszustand der Batterie zu bewerten. Die Ergebnisse sind in Abbildung 4 dargestellt.

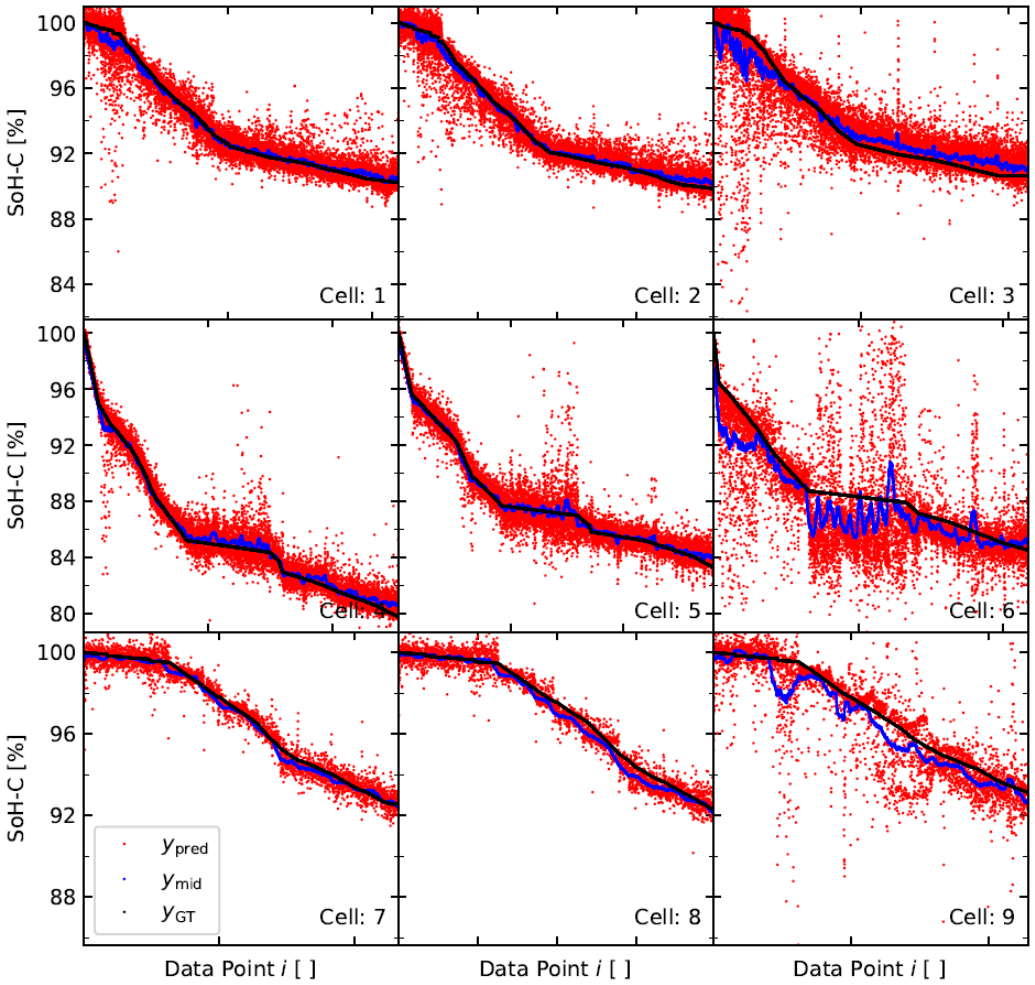


Abbildung 4: Ergebnisse der Gesundheitsschätzung

Abbildung 4: Zeigt neun Diagramme,die jeweils eine Zelle repräsentieren.Zellen {1,2,4,5,7,8} wurden für den Trainingsprozess verwendet,während die Zellen {3,6,9} zur Validierung dienten.Die Zellen 1-3 wurden bei 25 °C,die Zellen 4-6 bei 45 °C und die Zellen 7-9 bei 5 °C in einer Klimakammer zyklisiert.Jedes Diagramm enthält drei Graphen:

y_{GT} sind die tatsächlichen Gesundheitswerte, und y_{pred} sind die vom neuronalen Netz vorhergesagten Werte. Die Vorhersagen y_{pred} weisen ein hohes Rauschen auf. Um dies zu kompensieren, haben wir ein Hybridsystem implementiert, das die Daten des Modells weiterverarbeitet und glättet. Die Datenpunkte werden bis zu einer bestimmten Anzahl n_{mid} mit folgender Gleichung geglättet:

$$y_{mid,i} = \frac{1}{n_{mid}} \sum_{m=i}^{i+n_{mid}} y_{pred,m}$$

wobei $y_{mid,i}$ der geglättete Wert des Datenpunktes i ist. Um den optimalen Wert für n_{mid} zu finden, haben wir eine zusätzliche Untersuchung durchgeführt. Die

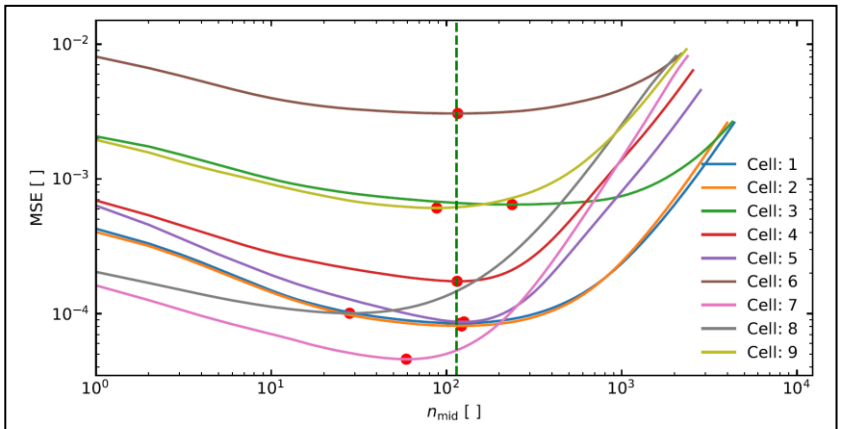


Abbildung 5: Anzahl der zu mittelnden Datenpunkte

Ergebnisse sind in Abbildung 5 dargestellt.

Abbildung 5: Visualisiert die Anzahl der zu mittelnden Datenpunkte $\mathrm{n_{mid}}$ über dem mittleren quadratischen Fehler (MSE) aller neun Zellen. Das globale Minimum variiert zwar, jedoch sind die Werte nah beieinander. Der ermittelte Mittelwert beträgt $\mathrm{n_{mid}} = 114$.

Hyperparameter-Tuning

Dieses Kapitel befasst sich mit der Hyperparameter-Optimierung für die untersuchten ANNs, ein kritischer, aber rechenintensiver Prozess, da jede Konfiguration trainiert werden muss, um ihre Leistung zu bewerten. Typischerweise kann das Hyperparameter-Tuning durch drei Hauptmethoden angegangen werden:

- Grid Search: Diese Methode umfasst eine systematische Erkundung eines bestimmten Bereichs von Hyperparameterwerten. Ein vordefiniertes Raster von Hyperparameterwerten wird erstellt, und der Algorithmus bewertet jede Kombination. Mit zunehmender Anzahl von Hyperparametern expandiert der Suchraum jedoch exponentiell, was Grid Search für große Datensätze oder Modelle mit zahlreichen Hyperparametern rechnerisch untragbar macht.
- Random Search: Diese Methode mildert einige Einschränkungen der Grid Search, indem sie zufällig Kombinationen von Hyperparametern zur Bewertung auswählt. Dieser Ansatz ist effizienter, insbesondere wenn einige Hyperparameter nur einen minimalen Einfluss auf die Modellleistung haben. Trotz seiner Effizienz bleibt er rechnerisch anspruchsvoll und garantiert nicht die Entdeckung des optimalen Hyperparametersatzes, insbesondere in umfangreichen Suchräumen mit einem begrenzten Evaluierungsbudget.
- Bayesian Optimization: Eine raffiniertere Technik, die eine probabilistische Modellierung verwendet, um die Leistung von Hyperparameterkombinationen basierend auf historischen Bewertungen vorherzusagen. Diese Methode wird weiter unten näher erläutert.

Bayesian Optimization umfasst zwei Hauptkomponenten:

- Surrogatmodell: Typischerweise werden Gaussian Processes verwendet, da sie flexibel sind und Unsicherheit schätzen können. Dieses Modell approximiert die Zielfunktion und liefert Einblicke in das erwartete Verhalten der Funktion an unerforschten Punkten und quantifiziert die Unsicherheit dieser Vorhersagen.
- Akquisitionsfunktion: Abgeleitet vom Surrogatmodell bestimmt diese Funktion den nächsten zu probierenden Punkt, indem sie Exploration (Sampling dort, wo Unsicherheit hoch ist) und Exploitation (Sampling dort, wo eine Verbesserung erwartet wird) ausbalanciert.

Zunächst werden einige Hyperparameterkombinationen zufällig ausgewählt, um eine Basis für die Akquisitionsfunktion zu schaffen. Diese Proben werden dann bewertet, um eine Kombination zu identifizieren, die wahrscheinlich eine Verbesserung bringt. Dieser Ansatz integriert frühe Stoppmechanismen und Datenreduktionsstrategien und nutzt nur etwa 1% des ursprünglichen Datensatzes, um den Evaluierungsprozess zu beschleunigen.

In dieser Studie wurde die Hyperparameter-Optimierung für mehrere Modelle durchgeführt, darunter befinden sich die Encoder basierten, die LSTM basierten und die Transformer basierten für die Wahl der Decoder Schichten. Der Hauptzweck dieser Optimierungsbemühungen bestand nicht nur darin, die besten Hyperparameter für jede Modellkonfiguration zu bestimmen, sondern auch, den Einfluss verschiedener Hyperparameter auf die Lern- und Generalisierungsfähigkeiten in verschiedenen Kontexten zu verstehen. Diese Untersuchung war entscheidend für die Gestaltung einer effizienten Architekturstrategie und die Sicherstellung, dass die Modelle robust gegen Realweltdaten validiert werden konnten.

Es ist wichtig zu beachten, dass die anfänglichen Hyperparmetereinstellungen, die durch diese Optimierungsphase identifiziert wurden, häufig von denen abwichen, die letztendlich in den finalen Modellen verwendet wurden. Diese Diskrepanz ergibt sich, weil die Trainingsdaten zur Zeitersparnis begrenzt sind. Diese Untersuchung hilft, den Suchraum effektiv zu verkleinern und leitet den Feinabstimmungsprozess, bei dem die Hyperparameter präziser basierend auf dem vollständigen oder erweiterten Datensatz angepasst werden.

Infolgedessen werden nur die endgültigen Ergebnisse nach umfangreicher Feinabstimmung dokumentiert und detailliert diskutiert, wobei die nach der anfänglichen Optimierung vorgenommenen Anpassungen reflektiert werden, um jedes Modell an die spezifischen Eigenschaften des Datensatzes und die in den Forschungszielen definierten Leistungsmetriken anzupassen.

Anhang

Beitrag des Ergebnisses zu den förderpolitischen Zielen

Das Projekt unterstützt die Ziele des Bundesministeriums für Wirtschaft und Energie im Rahmen des Fachprogramms „Neue Fahrzeug- und Systemtechnologien“ durch die Entwicklung eines intelligenten Batterie-Management-Systems (iBMS). Dieses System optimiert die Lebensdauer und Effizienz von Hochvoltbatterien, wodurch die CO2-Emissionen reduziert und die Akzeptanz von Elektrofahrzeugen erhöht werden.

Das ipv war zuständig für die Entwicklung der KI basierten Algorithmen.

Wissenschaftlich-technisches Ergebnis des Vorhabens

Die wesentlichen Ergebnisse umfassen die Entwicklung von KI-basierten Algorithmen zur Echtzeitdiagnose und Vorhersage der Batterieparameter.

Im Rahmen der Forschung wurden verschiedene Modellarchitekturen untersucht. Dabei zeigte sich, dass LSTM-basierte Algorithmen nach wie vor die besten Ergebnisse bei der Zeitreihenanalyse liefern. Dennoch haben auch Transformer-basierte Algorithmen ihren Platz, da sie in der Lage sind, mit den LSTM-Architekturen zu konkurrieren und in bestimmten Szenarien ebenfalls hervorragende Leistungen zu erbringen.

Fortschreibung des Verwertungsplans

Erfindungen/Schutzrechtsanmeldungen: Die Erkenntnisse Seitens ipv wurden veröffentlicht oder werden für die Dissertation verwendet.

Wirtschaftliche Erfolgsaussichten: Die Integration in ein Batteriemanagementsystem (BMS) steht noch aus. Mit zunehmender Rechenleistung und wachsender Datenmenge wird eine Weiterentwicklung und Optimierung der Algorithmen unvermeidlich sein.

Wissenschaftliche und technische Erfolgsaussichten: Die Ergebnisse können in öffentlichen Datenbanken, Netzwerken und für zukünftige Forschungsprojekte genutzt werden.

Wissenschaftliche und wirtschaftliche Anschlussfähigkeit: Die wissenschaftlichen Ergebnisse der Netzarchitekturen können für zukünftige Weiterentwicklungen genutzt werden.

Arbeiten, die zu keiner Lösung geführt haben

Die Integration mit einem echten BMS steht noch aus und das Fine-Tuning mit den Moduldaten.

Präsentationsmöglichkeiten für mögliche Nutzer

Die Ergebnisse des Projekts wurden in Fachzeitschriften und einer Dissertation präsentiert. Geplante Veröffentlichungen und Präsentationen dienen dazu, die entwickelten Technologien einem breiten Publikum vorzustellen und potenzielle Nutzer zu informieren.

Einhaltung der Ausgaben- und Zeitplanung

Das Projekt wurde innerhalb des geplanten Budgets und Zeitrahmens (mit Verlängerung) abgeschlossen.

Entwicklung neuer Zustandsschätzungsmodelle

Trainingsstrategie

Die Trainingsstrategie umfasst eine benutzerdefinierte Trainingsschleife, die je nach Modell und Trainingsprozess angepasst wird, um optimale Ergebnisse zu erzielen. Diese adaptive Methode verbessert die Effektivität und Effizienz der Trainingsschleife.

Epoch-Flow

Die Trainingsstrategie wird durch den „Epoch-Flow“ strukturiert, bei dem Dateien für Trainingszwecke gefiltert und in den Arbeitsspeicher geladen werden. Dieser Prozess wird

wiederholt, bis alle Dateien verarbeitet sind. Danach erfolgt die Validierung des Modells mit einer Datei aus einer nicht im Trainingsdatensatz enthaltenen Zelle, um die Generalisierungsfähigkeit des Modells zu bewerten.

Teacher Forcing

Teacher Forcing wird in der Zeitreihenanalyse verwendet, um Fehlerakkumulation zu vermeiden, indem tatsächliche Ausgaben aus dem Trainingsdatensatz als Eingaben verwendet werden. Eine schrittweise Reduzierung dieses Parameters hilft dem Modell, selbstständig korrekte Ausgaben zu erzeugen.

Validierungsalgorithmus

Der Validierungsprozess ähnelt dem Training, jedoch ohne Rückpropagation. Der Validierungsverlust wird berechnet, um die Leistung des Modells zu bewerten.

Lernratenscheduler

Die Lernrate wird basierend auf der Steigung der linearen Regression der Verluste der letzten drei Epochen angepasst. Eine positive Steigung führt zu einer Verkleinerung der Lernrate, eine negative zu einer Vergrößerung.

Zeitreihenprognosemodelle

LSTM-Modelle

LSTM-Modelle sind besonders geeignet für die Vorhersage von Zeitreihen, da sie in der Lage sind, langfristige Abhängigkeiten zu erfassen. Diese Modelle bestehen aus mehreren LSTM-Einheiten und Dropout-Schichten, um Überanpassung zu vermeiden. Die besten Ergebnisse wurden mit einer Abweichung von 0,744% erzielt.

Transformermodelle

Transformer-basierte Modelle verwenden Encoder-Decoder-Strukturen mit Multi-Head-Attention (MHA) und Feedforward-Schichten. Diese Modelle sind in der Lage, konkurrierende Ergebnisse zu liefern, wobei die beste Abweichung bei 2,9% liegt.

Informer-Modelle

Informer-Modelle verbessern die Zeitreihenprognose durch fortschrittliche Self-Attention-Mechanismen, die sich auf die wichtigsten Abfragen konzentrieren. Das beste Modell erreichte eine Abweichung von 1,96%.

Hyperparameter-Tuning

Die Hyperparameter-Optimierung wurde durch Grid Search, Random Search und Bayesian Optimization durchgeführt. Letztere Methode nutzt probabilistische Modellierung, um die besten Hyperparameterkombinationen zu identifizieren und den Evaluierungsprozess zu

beschleunigen. Die Optimierungsphase hilft, den Suchraum zu verkleinern und die Modelle präzise anzupassen.

Klassifizierungsmodelle

Traditionelle Methoden zur Gesundheitsschätzung von Lithium-Ionen-Batterien wurden durch eine alternative Methode ersetzt. Diese neue Methode analysiert Strompulse in Fahrzyklen und nutzt ein künstliches neuronales Netz, um den Gesundheitszustand der Batterie zu bewerten. Durch die Glättung der Datenpunkte konnte die Modellgenauigkeit verbessert werden.

Zusammenfassung

Die Forschung zeigt, dass LSTM-basierte Algorithmen weiterhin die besten Ergebnisse für Zeitreihen liefern, während Transformer- und Informer-Modelle ebenfalls vielversprechend sind. Die Integration dieser Modelle in ein Batteriemanagementsystem und die fortlaufende Verbesserung der Algorithmen sind entscheidend für die Optimierung der Batterieleistung.