NETZLASTMANAGEMENT VIA LADESÄULEN AUF BASIS VON DEEP REINFORCEMENT LEARNING UNTER BETRACHTUNG VERSCHIEDENER BEOBACHTUNGSRÄUME

Dennis Salvador Versen*, Yuzhuo Fu

*Helmut-Schmidt-Universität Hamburg, Holstenhofweg 85, versend@hsu-hh.de

Kurzfassung: Die zunehmende Elektrifizierung des Verkehrs- und Wärmesektors führt zu erhöhten Lastspitzen in Niederspannungsnetzen, insbesondere durch konzentriertes Laden von Elektrofahrzeugen in den Abendstunden. Diese Arbeit untersucht den Einsatz von Deep Reinforcement Learning (RL) zur präventiven Steuerung von Elektrofahrzeug-Ladevorgängen mit dem Ziel, Netzüberlastungen zu vermeiden und gleichzeitig die Ladekapazität möglichst vollständig zu nutzen. Es konnte anhand drei verschiedener Szenarien mit unterschiedlich abrufbaren Messdaten gezeigt werden, dass Deep RL eine vielversprechende Begleittechnologie zum kostenintensiven Netzausbau darstellt und in Gegenden, wo wenig Netzlast-relevante Daten erfasst werden können, zur Netzstabilität beitragen kann.

Keywords: Lastmanagement, Engpassmanagement, EV-Lademanagement, Deep Reinforcement Learning

1 Einleitung

Die steigende Last in elektrischen Energienetzen aufgrund der Elektrifizierung von immer mehr Verbrauchern kann zur Verletzung von Grenzwerten bei Netzbetriebsmitteln führen. Die Elektrifizierung eines Großteils des Verkehrssektors sowie des Wärmesektors führt zwangsläufig zu hohen Lastspitzen an Kabeln und Transformatoren. Diese zum Teil niedrig dimensionierten Betriebsmittel sind aktuell eher unzureichend für die volatilen elektrischen Lasten und den Grad der Technologiedurchdringung ausgelegt. Neben einer zeitweisen Unterversorgung können auch Schäden an besagten Betriebsmitteln eintreten und damit einhergehende längerfristige Stromausfälle.

Eine mögliche, aber kostenintensive Lösung wäre der flächendeckender Netzausbau. Mit Blick auf die Fristen des Pariser Klimaabkommens und des Verbrennerverbots [1] [2] bis 2035 ist dies ein ambitioniertes Unterfangen. Derzeit müssen steuerbare Verbraucher mit einer Leistung von über 4,2 kW beim Netzbetreiber gemeldet sein. Bei drohender Netzüberlastung kann der Netzbetreiber die Leistungsaufnahme solcher steuerbaren Verbraucher temporär drosseln. Darunter fallen folgende Verbraucher: Wallbox, Wärmepumpen, Batteriespeicher, Klimaanlagen [3]. Von den genannten Verbrauchern eignen sich diejenigen am meisten zur Regelung, die ihre Leistungsaufnahme zeitlich verlagern können. In diesem Sinne eignen sich insbesondere Batterien und E-Autos (EVs).

1.1 Problemstellung

Das gleichzeitige Laden von Elektrofahrzeugen in den Abendstunden stellt eine Herausforderung für die Stromnetze dar. Bereits eine frühe US-amerikanische Studie mit 2.704 Elektrofahrzeugen zeigte, dass ohne zeitvariable Stromtarife die Nachfrage ab etwa 16:00 Uhr stetig ansteigt und ihren Höhepunkt um 20:00 Uhr erreicht [4]. Die bislang größte Studie zum privaten Ladeverhalten in Deutschland mit 51.893 Wallbox-Besitzern ohne private PV-Anlagen zeigt, dass Kunden ihre Fahrzeuge konzentriert in den Abendstunden laden, 65 Prozent zwischen 18 und 21 Uhr sowie 49 Prozent zwischen 21 und 0 Uhr [5]. Diese zeitliche Konzentration der Ladevorgänge führt zu erheblichen Spitzenlasten im Stromnetz genau in jenen Stunden, in denen auch der allgemeine Haushaltsverbrauch seinen Höhepunkt erreicht.

Bei der netzorientierten (kurativen) Steuerung werden Verbrauchereinheiten bei auftretenden Überlasten oder drohenden Netzschäden in Echtzeit gedrosselt. Sobald Anzeichen einer Überlastung erkannt werden, erfolgt die Leistungsreduzierung auf minimal 4,2 kW, um Schäden am Netz zu vermeiden. Hierbei liegen die Reaktionszeiten bei 3 min [6], was insbesondere bei schnell auftretenden Lastspitzen (kumuliertes Laden in den Abendstunden) zum Problem werden kann.

In wissenschaftlichen Arbeiten ist mathematische Optimierung ein häufig erforschter Ansatz (s. Kap. 2) für das Last- und Resourcenmanagement. Eine Optimierung über das gesamte Netz würde voraussetzen, dass das System vollständig in einem Modell mit all seinen äußeren Einflüssen, Störgrößen und schwer vorhersagbaren Wechselwirkungen abgebildet werden kann. Die Entwicklung eines physikalischen Modells scheitert derzeit an mehreren Faktoren: Die geringe Digitalisierung der Niederspannungsnetze erschwert die Echtzeiterfassung von Netzzuständen erheblich. Zudem können viele relevante Informationen vom Verbraucher nicht unmittelbar bereitgestellt werden, sowohl aus technischen Gründen als auch aufgrund datenschutzrechtlicher Vorgaben. Zwar messen Smart Meter Einheiten alle 15 Minuten den Verbrauch, diese können allerdings nur einmal am Tag gebündelt an den Netzbetreiber gesendet werden [7]. Diese Informationslücken verhindern eine vollständige Systemoptimierung.

Hier können Ansätze des sogenannten modellfreien selbstbestärkenden Lernens durch KI eine vielversprechende Lösung sein. Deep Reinforcement Learning Algorithmen dieser Art benötigen kein Modell oder eine umfangreiche Zustandsbeschreibung. Sie lernen rein datengetrieben.

1.2 Zielsetzung und Forschungsfrage

Diese Arbeit evaluiert unterschiedliche modellfreie Ansätze basierend auf KI und selbstlernenden Agenten (RL) zur Echtzeit-Regelung eines Niederspannungsnetzes (NSN). Dabei liegt der Schwerpunkt auf der Verwendung realer Verbraucherprofile und der Bewertung der Eignung dieser Daten für das modellfreie Reininforcement Learning. Obwohl vorhergehende Arbeiten [8] und [9] bereits vielversprechende Ergebnisse gezeigt haben, so basierten die Lastprofile aus zum Teil künstlich hergestellten repetitiven Daten. Bei der Verwendung synthetischer Daten besteht immer die Möglichkeit, dass die KI lernt, wie diese erzeugt wurden, anstatt die Dynamik und Wahrscheinlichkeiten der Umgebung zu lernen. Das Verwenden realer Daten erhöht die Validität und macht die Ergebnisse aussagekräftiger. Die Forschungsfrage, die hier beantwortet werden soll, lautet somit: Kann ein prädiktives Echtzeit-Lastmanagement mit modellfreien Reinforcement Learning in einer komplexen stochastisch dynamischen Umgebung mit realen Messdaten zur Netzstabilität beitragen, und welche Daten sind hierzu erforderlich?

1.3 Aufbau der Arbeit

Die Arbeit ist in folgende Abschnitte unterteilt: In Kapitel 2 werden aktuelle Forschungsarbeiten zum Thema Steuerung von Verbrauchern zur Reduktion von Netzengpässen vorgestellt. Kapitel 3 zeigt die hier verwendeten Methoden. Darunter fallen der Aufbau der Netzsimulation, die hier verwendeten Daten für Last- und Ladeprofile sowie Beschreibung des Netz-Regelansatzes über Deep Reinforcement Learning (DRL). Zudem werden die unterschiedlichen Regel-Szenarien vorgestellt. In Kapitel 4 folgt der Aufbau eines Demonstrator-Scenarios und die Darstellung der daraus folgenden Ergebnisse. Zum Schluss wird in Kapitel 5 noch ein Fazit sowie der Ausblick auf weitere Arbeiten gegeben.

2 Stand der Forschung

Die Steuerung von Verbrauchereinheiten zur Vermeidung von Netzengpässen in Niederspannungsnetzen ist aufgrund der zunehmenden Elektrifizierung ein aktives

Forschungsfeld. Hierbei sind Optimierungsverfahren, basierend auf detaillierten Kenntnissen der Umgebung und Vorhersagemodellen, ein viel diskutierter Ansatz in der Forschung. Das Optimierungsproblem eines Lademanagements über einen definierten Zeitraum ist allerdings ein nicht-konvexes Problem und stellt daher eine Herausforderung für die klassische Optimierung dar. Unterschiedliche Arbeiten basierend auf verschiedene Herangehensweisen, um mit diesem Problem umzugehen: Veröffentlichung [10] adressiert das Problem mit Second-Order Conic Relaxation, wodurch ein relaxiertes Problem mit einem globalen Optimum erhalten werden kann. In [11] wird ein genetischer Algorithmus mit Problemzerlegung und evolutionärer Heuristik vorgestellt. In [12] erfolgt eine Navigation durch den nicht-konvexen Lösungsraum mit Multi-Objective-Particle-Swarm-Optimization.

Obwohl Optimierungsansätze häufig zu sehr genauen Lösungen kommen, stoßen sie jedoch bei der hohen Komplexität, den vielfältigen Störgrößen und der Stochastik realer Netze an ihre Grenzen. Hier bieten modellfreie Reinforcement-Learning-Ansätze vielversprechende Alternativen, da sie keine genauen Informationen der Umgebung benötigen, sondern nur eine Interaktion mit ihr.

Eine zentrale Schwäche vieler bisheriger Ansätze liegt in der Verwendung künstlich generierter oder stark vereinfachter Lastprofile. Synthetische Daten weisen häufig repetitive Muster auf und bilden nicht die volle Komplexität des realen Verbrauchsverhaltens ab. Dies birgt die Gefahr, dass RL-Agenten lernen, wie die Daten erzeugt wurden, anstatt robuste Strategien für reale stochastische Umgebungen zu entwickeln. Die Übertragbarkeit solcher Ergebnisse auf die Praxis bleibt daher fraglich. Beispielsweise nutzen die Autoren von [13] synthetische Daten zur Erstellung von Lastprofilen und dynamischen Strompreisen und führen damit ein Reinforcement-Learning durch, wobei sie auf Ladeplanung von Elektrofahrzeugen sowie Spannungsstabilität im Verteilernetz zielen.

Arbeiten, in denen die gleichen realistische Datensätze wie in dieser Arbeit genutzt werden beschäftigen sich bisher mit der optimalen Ausnutzung des Eigenenergieverbrauchs von Photovoltaik-Nutzern in Verbindung mit einem Deep Q-Network wie in [14]. Mit historischen Smart-Meter-Daten aus der Pecan Street Datenbank steht hier die Erhöhung des durchschnittlichen Eigenverbrauchs von Solarenergie zum Laden von Elektrofahrzeugen im Vordergrund.

Ein weiterer in der Forschung oft vernachlässigter Aspekt ist die zeitliche Verfügbarkeit von Daten. In der Praxis werden einige Messdaten jedoch nur in 15-Minuten-Intervallen erhoben und unterliegen, insbesondere in Deutschland, strengen Datenschutzanforderungen [15]. Viele theoretische Ansätze setzen vollständigen Zugriff auf detaillierte Verbraucherdaten voraus. In [16] wird vorausgesetzt, dass die State of Charges (SOC) sowie die Anwesenheit eines EVs bekannt sind. Es wird jedoch nur von Zeitintervallen gesprochen aber nicht welche Länge Sie haben.

Auch in [17] wird keine Aussage zur Auflösung der diskreten Zeitintervallen gemacht. Hier wird ein Multi-Agent RL vorgestellt, bei dem jedes Elektrofahrzeug einen eigenen Agenten hat, der seine Ladestrategie individuell optimiert und sich mit anderen Agenten koordiniert. Der Fokus liegt auf kundenseitiger Kostenminimierung und Multi-Objective-Optimierung mit tabularem Q-Learning. Lokale Informationen werden hier nicht geteilt, was den Ansatz in dieser Hinsicht praxisnah macht.

Die Arbeit [18] stellt auch ein Lademanagement mit RL in einer datenschwachen Umgebung vor. In [18] werden zur Behandlung dieses Settings verschiedene RL-Algorithmen miteinander verglichen. Dabei handelt es sich um klassische tabulare diskrete RL-Algorithmen, die keine Neuronalen Netze verwenden. Zudem wird hier von nur einem Szenario ausgegangen, in dem der SOC zu jedem Zeitschritt abrufbar ist. Die Arbeit weist auf die beschränkte Skalierbarkeit von tabularen RL hin und empfiehlt den Einsatz von Deep Reinforcement Learning.

Im Gegensatz zu den vorgestellten Arbeiten, nutzt diese Arbeit Single-Agent Deep RL mit einem zentralen Agenten beim Netzbetreiber, der alle Ladevorgänge in einer informationsarmen Umgebung über einen einheitlichen diskriminierungsfreien Dimmfaktor

steuert. Dazu werden reale Messdaten aus Haushalten in Kombination mit Ladeverhalten verwendet, was in der Praxis einen deutlich realistischeren Ansatz darstellt.

3 Methodik

Für das präventive Lademanagement durch DRL werden verschiedene Methoden angewandt. Um die RL-Agenten zu trainieren, muss eine Trainingsumgebung aufgebaut werden. Diese sollte die Eigenschaft eines markovschen Entscheidungsproblems (MDP) haben, da RL sich besonders zur Lösung solcher Probleme eignet. Dazu wird die Simulation eines Netzmodelles mit hoher Durchdringung von EV-Ladepunkten sowie der Einsatz von Last- und Ladeprofilen zur Generierung der Zeitreihen benötigt. Um die Simulationsgeschwindigkeit optimal zu halten, werden Vorgänge vektorisiert. Das folgende Kapitel soll die angewendeten Methoden beschreiben. Zudem werden die Konzepte des Deep RL erklärt, wobei das MDP als Entscheidungsproblem für eine Niedersapnnungsnetzumgebung erläutert wird.

3.1 Systemmodellierung

Für die Modellierung des Systems wird die Datenbank Pandapower genutzt [19]. Pandapower ist ein Open Source Tool zur Modellierung und Berechnung von Stromnetzen. Das in dieser Arbeit betrachtete Niederspannungsnetz wird von einer 400 kVA-Ortsnetzstation versorgt. Das Netz ist ein Strahlennetz mit Sammelschienen, die jeweils drei Mehrfamilienhaushalte mit jeweils drei Ladesäulen versorgen. Zudem werden 10 öffentliche Ladepunkte am obersten Netzknoten hinzugefügt (Abb. 1).

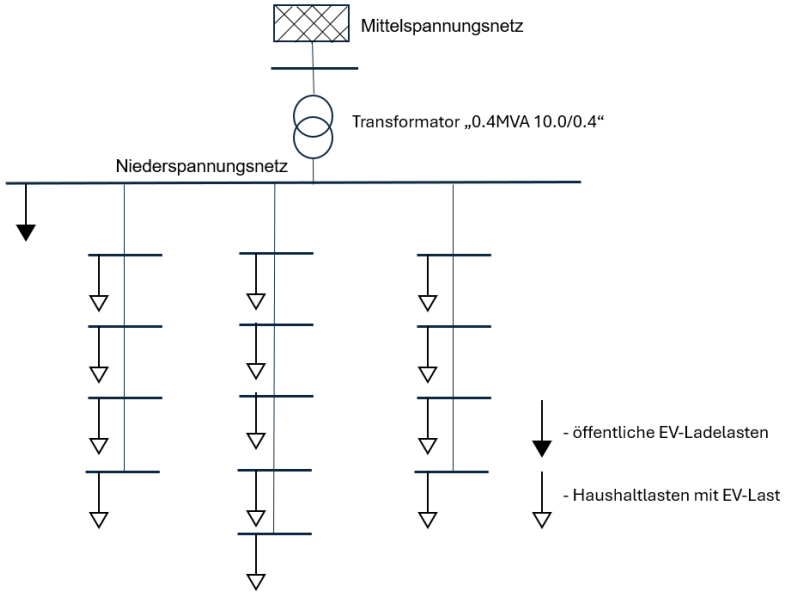

Abbildung 1:Darstellung der simulierten Netztopologie mit hoher EV-Durchdringung

Für die Bereitstellung der auf den empirischen Last-Daten basierenden Szenarien, wird eine simulationsgestützte Umgebung auf Basis von Pandapower umgesetzt. Der Prozess folgt einem iterativen Schema, bei dem die Haushalts- und Ladesäulenlasten pro Zeitschritt aus den vorhandenen Datenframes entnommen und als Lasten in das Netzmodell integriert werden. Die Ermittlung des Systemzustands erfolgt anschließend durch eine Newton-Raphson-Lastflussberechnung. Parallel dazu verwerten neuronale Netze die Zustandsinformationen, um Steuerungseingriffe zu generieren, welche die Lastwerte im Datenframe dynamisch anpassen. Um den Rechenaufwand über den Simulationszeitraum minimal zu halten, wird eine Vektorisierung der Eingangsdaten angewandt, wodurch die Verarbeitungsgeschwindigkeit der prädiktiven Modelle signifikant erhöht wird. Hierbei gibt

$(x < y)$ einen booleschen Wert zurück, der eine falsche von einer wahren Aussage trennt, indem er eine 0 oder eine1 zuordnet.

$$\boldsymbol{P_{CP,t} = CP_{occ,t} \cdot (\Delta E_t < 0) \cdot \mathrm{P_{CP,max}} \cdot a_t} \quad (1)$$

$$\boldsymbol{\Delta E_{t+1} = \min(0, P_{CP,t} \cdot \Delta t + \Delta E_t + SOC_{new,t})} \quad (2)$$

Um die aktuellen Leistungswerte der Ladesäulen $\boldsymbol{P_{CP,t}}$ zu errechnen (1), werden bei der Initialisierung $(t = 0)$ eine boolesche Occupancy Matrix $\mathbf{CP_{occ}}$ sowie eine Ladezustandsmatrix $\mathbf{SOC_{new}}$, die den Ladestand eines EVs bei Ankunft am Ladepunkt beinhaltet, aus den Ladeprofilen erstellt. Die Matrizen beinhalten pro Zeitschritt t einen Ladezustandsvektor bzw. einen Occupancy-Vektor. Die Ladeprofile beinhalten die Ladesessions für jeden Ladepunkt und beinhalten die folgenden Daten: Ankunftszeit, Abfahrzeit, Ladezustand bei Ankunft und maximale Ladeleistung. Nach jedem Zeitschritt werden die Daten, dann aus den Matrizen gelesen und mit dem Dimmfaktor a_t und der maximalen Ladeleistung $\mathbf{P_{CP,max}}$ verrechnet. Dabei wird geprüft welche Autos noch Defizit $(\boldsymbol{\Delta E_t} < 0)$ an Ladung haben. Am Ende des Zeitschritts werden dann die neuen Ladestände $\boldsymbol{\Delta E_{t+1}}$ aktualisiert (2). Da $\boldsymbol{\Delta E}$ den Bedarf anzeigt, kann es nur 0 oder negativ sein.

3.2 Datengrundlage

Diese Arbeit nutzt für die Profile der Haushalte sowie für die EV-Daten die Forschungsdaten aus der Pecan Street Datenbank [20]. Diese sind zusammengetragene Daten aus Haushalten in Kombination mit privatem EV-Gebrauch. Die Pecan Street Datenbank bietet mit über 500 veröffentlichten Studien einen der umfangreichsten Datensätze für Smart-Grid-Forschung. Ein Problem realer Daten ist, dass die Messdaten nicht immer vollständig sind und zeitweise Messreihen fehlen. Da die Simulation Kontinuität in den Daten benötigt, werden die fehlenden Messabschnitte mit Daten aus anderen Jahren für dieselben Monate und Wochentage aufgefüllt. Die Abbildungen 2 und 3 zeigen die Transformatorauslastung sowie die Kabelauslastung für den Zeitraum 15.04.-15.06.2015 ungeregelt.

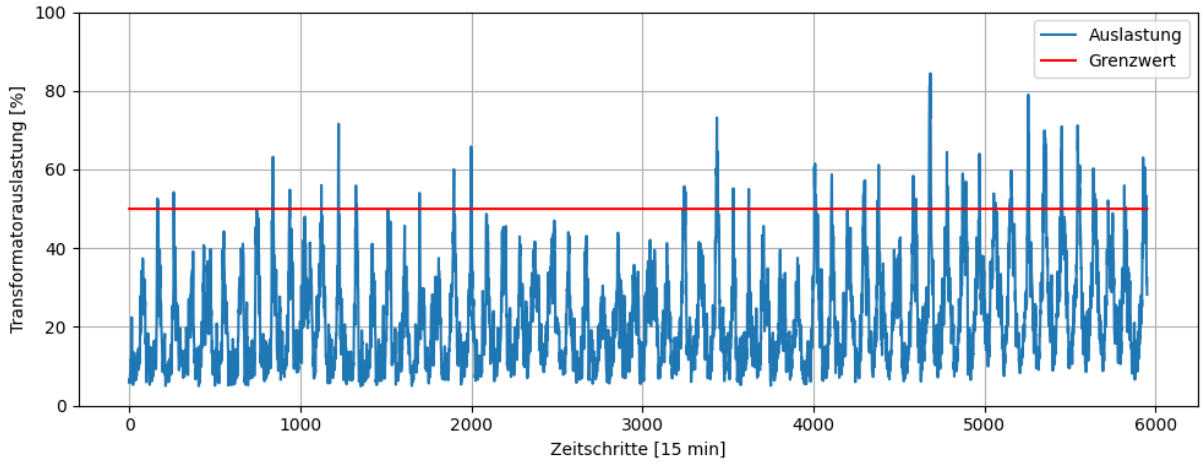

Abbildung 2: Transformatorauslastung für den Zeitraum 15.04.-15.06.2015

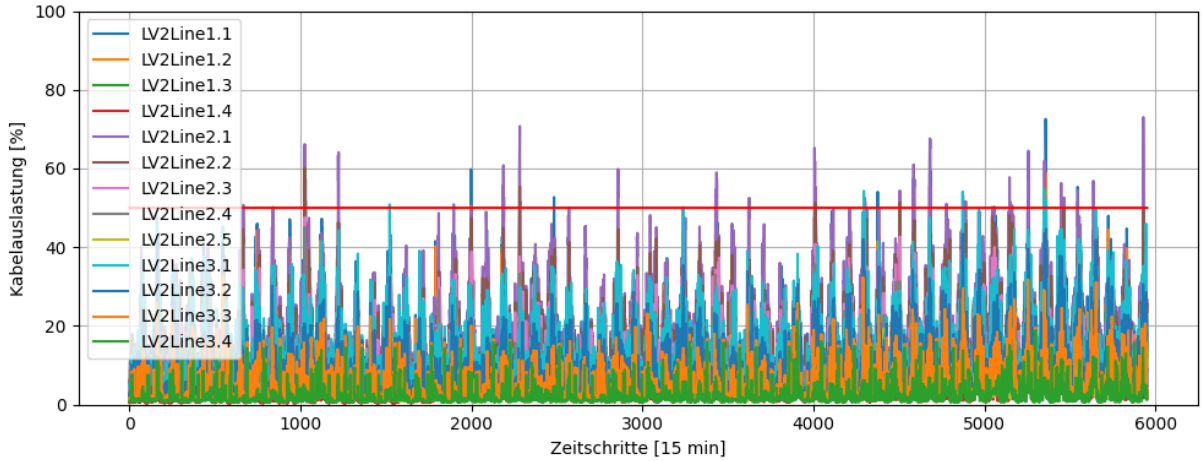

Abbildung 3: Kabelauslastung für den Zeitraum 15.04.-15.06.2015

Da es sich hier um ältere Daten handelt, können hier nicht die Lastspitzen erzeugt werden, wie es vermutlich bei den heutigen leistungsstärkeren Ladepunkten der Fall ist. Für die Regelung wird daher ein Grenzwert von 50% der Betriebsmittelauslastung angenommen. Für das Training und die Evaluation werden die Daten in Trainings- und Validierungsdaten aufgeteilt.

3.3 MDP-Formulierung

Für die Umsetzung des Lastmanagements über die Ladesäulen werden verschiedene model-freie DRL-Algorithmen herangezogen. DRL eignet sich insbesondere zur Lösung von stochastischen unsicheren Umgebungen auch markovsche Entscheidungsprobleme (MDP) genannt [21]. Daher wird das hochgradig volatile Last- und Lademanagement als MDP formuliert. MDPs können durch das markovsche Tupel

$$\{S, A, P, R\}$$

beschrieben werden, bestehend aus Zustandsmenge S, Aktionsmenge A, Wahrscheinlichkeit der Zustandstransition $P(s'|s,a)$; $s, s' \in S$; $a \in A$ und eine Belohnungsfunktion $r = R(s,a)$ die den Zustandsübergang bewertet. Für die betrachtete NSN-Umgebung kann ein Zustand aus den abrufbaren Messdaten bestehen, wie z. B. Transformatorauslastung, Kabelauslastung, Haushaltslast, Ladesäulenlast, EV-Daten etc. Die Aktionen a werden hier durch einen globalen diskriminierungsfreien Dimmfaktor beschrieben, der pro Zeitschritt (15 min) die Last an den Ladesäulen (LS) reguliert. Die Wahrscheinlichkeiten der Zustandsübergänge P ist durch das ungewisse zukünftige Ladeverhalten der EV-Nutzer gegeben. Wie eingangs beschrieben, ist in der Realität der vollständige Zustand, aufgrund von fehlender Digitalisierung und Datenschutz nicht immer abrufbar. Daher spricht man in diesem Fall auch von einem partiell observierbaren MDP. Die Belohnungsfunktion ist abhängig von den zu erreichenden Zielen und von den Messdaten. Für die NSN-Umgebung ist die Belohnungsfunktion wie folgt definiert:

$$\boldsymbol{r_t = \mathrm{W} \cdot \sum_{i}^{\mathrm{N}} E_{i,t} - \mathrm{T} \cdot \min\left(0, \theta_t - \theta_{ref}\right) - \mathrm{U} \cdot \sum_{l}^{\mathrm{L}} \min\left(0, \kappa_{l,t} - \kappa_{ref}\right) - \mathrm{V} \cdot \Delta a_t} \qquad (3)$$

Dabei wir die gesamte geladene Energie E_i der N EVs als Belohnung definiert. Die Transformatorlast θ sowie die Kabellasten κ werden mit den Referenzwerten θ_{ref} und κ_{ref} verglichen. Verstöße werden als negative Belohnung integriert. Um ein weiches Regeln zu ermöglichen und zusätzlichen Stress auf das Netz zu vermeiden, werden große Dimmfaktoränderungen Δa bestraft. $\mathrm{W}, \mathrm{T}, \mathrm{U}$ und V stellen Gewichte dar mit Intervallgrenzen [0,1]. Je nach Szenario kann die Belohnungsfunktion anders definiert sein. Zum Bespiel könnte

in einem Szenario die Information der geladenen Energie nicht übermittelt werden. Dann müsste dieser Term ersetzt werden.

3.4 Deep RL-Algorithmen

RL ist ein lernbasiertes Paradigma, bei dem ein Agent durch Interaktion mit der Umgebung eine optimale Strategie erlernt, indem er die langfristig kumulierten Belohnungen maximiert. Dieser Lernprozess beinhaltet in der Regel „Trail and Error“ und erfordert oft umfangreiche Interaktionen mit der realen oder einer simulierten Umgebung (Abb. 5). RL erfordert kein explizites Modell und legt den Schwerpunkt auf datengesteuertes, erfahrungsbasiertes Lernen und eignet sich besonders in unsicheren MDPs.

Zur Lösung des MDP beobachtet der Agent den Zustand s aus der Umgebung ① und erhält durch Interaktion *a* ② eine Belohnung *r*.

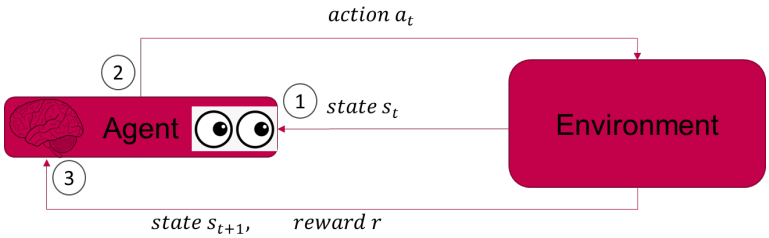

Abbildung 4: Schema des RL-Algorithmus

Die Informationen werden mit dem Folgezustand s_{t+1} in einem Replayspeicher ③ abgespeichert und dienen in Trainingsintervallen dem Agenten zum Erlernen einer Strategie. Hierzu eignen sich verschiedene RL-Algorithmen. Diese Arbeit evaluiert DRL-Algorithmen mit Policy Gradienten und stetigen Aktionsräumen. Hier tun sich insbesondere drei Kandidaten hervor:

- Deep Deterministic Policy Gradient (DDPG)
- Soft Actor Critic (SAC)
- Proximal Policy Optimization (PPO)

Das Prinzip besteht grundsätzlich in der Dualität zweier Neuronaler Netze, dem Actor Netzwerk $\mu(s)$ und dem Critic Netzwerk $Q(s,a)$. Der Actor nimmt den Zustand der Umgebung und wandelt ihn in eine Aktion. Der Critic hilft dabei dem Actor, die richtige Strategie zu erlernen, indem er die gewählte Actor-Aktion und den Zustand mit einem Q-Wert bewertet. Der Q-Wert ist ein Maß für die Qualität der gewählten Aktion in besagtem Zustand. Über die Bellmann-Gleichung (4) und eine Fehlerfunktion (5) wird das Critic-Netzwerk trainiert:

$$\boldsymbol{y = r_t + \gamma \cdot Q\left(s_{t+1}, \mu(a_{t+1}|s_{t+1})\right)}, \qquad \boldsymbol{\gamma \in [0,1]} \tag{4}$$

$$\boldsymbol{E(\phi) = \left(Q_\phi(s_t, a_t) - y\right)^2} \tag{5}$$

Die Bellmann-Gleichung ermöglicht es, den langfristigen Nutzen einer Handlung einzuschätzen, anstatt nur auf sofortige Belohnungen zu schauen [21]. Sie errechnet die Zielwerte y für die Fehlerfunktion E über die ausgeschüttete Belohnung r_t und einen über γ diskontierten Gewinn in der Zukunft $Q(\boldsymbol{s_{t+1}}, \mu(a_{t+1}|\boldsymbol{s_{t+1}}))$. Mittels Gradientenabstiegsverfahren werden die Gewichte $\boldsymbol{\phi}$ iterativ, mit dem Ziel der Minimierung des mittleren quadratischen Fehlers (MSE), angepasst. Diese Anpassung wird auch als Backpropagation bezeichnet. Um das Actor-Netzwerk zu trainieren, dient das Critic-Netzwerk

als Zielfunktion. Das Ziel des Actors ist es, eine Policy $\mu(a|\,\boldsymbol{s})$ zu erlernen, die $Q(\boldsymbol{s},a)$ maximiert. Über den Gradienten der Werte-Funktion (6) lassen sich die Gewichte $\boldsymbol{\omega}$ des Actor-Netzwerks so verändern, dass $Q(\boldsymbol{s},a)$ maximiert wird.

$$\arg\max_{\boldsymbol{\omega}} \boldsymbol{Q}\left(\boldsymbol{s}, \boldsymbol{\mu}_{\boldsymbol{\omega}}(\boldsymbol{a}|\boldsymbol{s})\right) \tag{6}$$

Das beschriebene grundlegende Prinzip, kann bei den einzelnen genannten Algorithmen abweichen, insbesondere bei PPO. DDPG hat deterministische Actor-Ausgabewerte während SAC die Ausgabewerte, über das Samplen eine Normalverteilung erhält. Während DDPG und SAC nach jedem Schritt ihren Actor anpassen können, braucht PPO längere Sequenzen mit gleicher Policy, bevor er seinen Actor aktualisieren kann, DDPG und SAC werden daher auch als Off-Policy Verfahren und PPO als On-Policy bezeichnet. Für weitere Ausführungen soll hier auf die Referenzen [22], [23], [24] verwiesen werden.

3.5 Szenarien

Neben dem Vergleich unterschiedlicher DRL-Algorithmen werden unterschiedliche Szenarien formuliert. Diese unterscheiden sich hauptsächlich in den beobachteten Zuständen. In einem perfekten Szenario würde der Zustand vollständig abrufbar sein. Für diese Arbeit werden nur Teile dieses Zustandsraumes verwendet. Von folgendem Zustandsraum wird dabei ausgegangen:

- t - Uhrzeit
- θ - Transformatorlast
- $P_{CP,i}$ - Ladeleistung für EV *i*
- $P_{HH,i}$ - Haushaltslast für HH *i*
- SOC_i - Ladestand für EV *i*
- N - Anzahl der ladenden EVs
- κ_l - Auslastung an Kabel *l*

Für die Szenarien werden drei unterschiedliche Beobachtungsräume definiert:

Szenario 1: $S1 \coloneqq \{t, \theta, \boldsymbol{\kappa}\}$,

Szenario 2: $S2 \coloneqq \{t, \theta, \boldsymbol{\kappa}, N\}$,

Szenario 3: $S3 \coloneqq \left\{t, \theta, \boldsymbol{\kappa}, \sum P_{CP,i}, \sum P_{HH,i}\right\}$

Das erste Szenario geht davon aus, dass dem Verteilnetzbetreiber die Informationen zu Ladesäulen und EVs nicht vorliegen. Nur die Leistungsflüsse an Transformator und Kabel können gemessen werden. Für Szenario 1 muss die Belohnungsfunktion angepasst werden, da hier keine Informationen über die ladenden EVs übermittelt werden können. Daher dient hier als positive Belohnung der Dimmfaktor selbst statt der geladenen Energie (3). Szenario 2 enthält als minimale Information die Anzahl der gerade ladenden Autos im LV-Gebiet. Szenario 3 geht von einer umfangreichen Smart Meter Messung aus, wobei die Leistungen von Haushalten und Ladesäulen alle 15 min gemessen und kumuliert übertragen werden. Szenario 2 und 3 geht von der Übermittlung kundenseitiger Informationen aus, die jedoch immer noch einen Verbraucherschutz bieten, da anhand der Informationen keine direkte Aussage über die aktuelle Lokalisation der Verbraucher gemacht werden kann.

4 Experimente und Ergebnisse

Für Training und Evaluation werden die verfügbaren Datensätze in zwei Hälften aufgeteilt. Dabei dient eine Hälfte des Datensatzes als Trainingssatz, während die andere Hälfte zur Validierung herangezogen wird. Für das Training wird kein explizites Abbruchkriterium

definiert. In Reinforcement-Learning-Anwendungen ist es zwar üblich, Episoden bei bestimmten Regel- oder Grenzwertverletzungen vorzeitig abzubrechen, insbesondere bei sogenannten harten Restriktionen. Ohne dieses Vorgehen würde die Gefahr bestehen, dass der Agent Strategien erlernt, die Verstöße dieser Art gezielt in Kauf nehmen, sofern sie nicht unmittelbar zum Abbruch der Episode führen. Da in realen Netzen kleine Grenzwertüberschreitungen nicht zwangsläufig zu einem Zurücksetzen des Systemzustands führen und die Agenten perspektivisch auch im realen Betrieb Strategien lernen müssen sich davon zu erholen, wird auf ein solches Abbruchkriterium verzichtet. Eine Episode umfasst daher 3051 Zeitschritte und wird unabhängig von auftretenden Grenzwertverletzungen vollständig durchlaufen. Erst nach Abschluss der Episode wird der Umgebungszustand zurückgesetzt.

4.1 Training

Die Agenten werden nacheinander in der Simulationsumgebung trainiert. Zu Beginn des Trainings werden die Aktionen initial zufällig generiert, um den Zustands- und Aktionsraum explorativ zu erschließen, bevor das eigentliche Lernen einsetzt. Als Leistungsmaß für den Trainingsfortschritt dient der Episodenscore, der die kumulierte Belohnung über eine gesamte Episode beschreibt. Eine Konvergenz des Lernprozesses liegt vor, wenn sich der Episodenscore über aufeinanderfolgende Episoden nicht mehr signifikant erhöht. Abbildung 6 zeigt die Lernkurven der drei untersuchten Algorithmen für die Szenarien S1, S2 und S3.

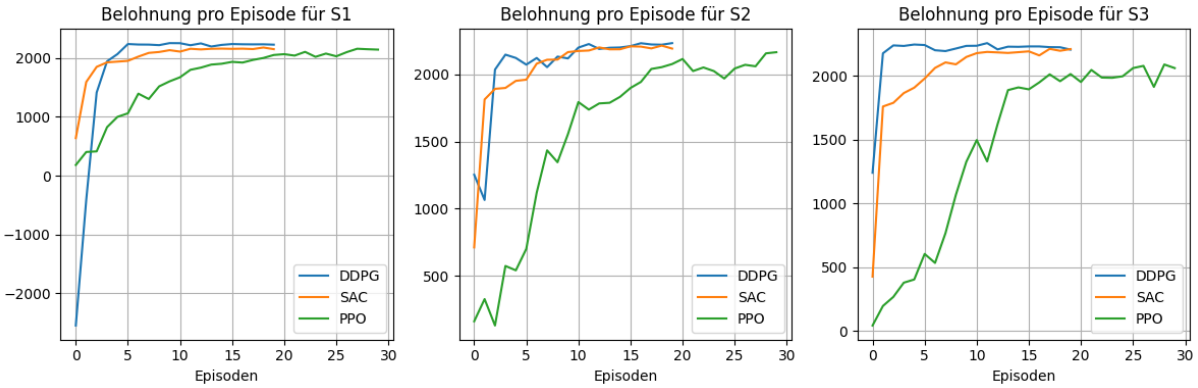

Abbildung 5:Lernkurven für Szenario S1, S2 und S3

Die Ergebnisse zeigen für alle Szenarien und Algorithmen eine steigende Lernkurve, was darauf hinweist, dass alle Agenten Strategien erlernen, die zu einer Erhöhung der kumulierten Belohnung führen. DDPG und SAC konvergieren dabei bereits nach etwa 10 Episoden, während PPO erst nach ungefähr 30 Episoden eine stabile Konvergenz erreicht. Dieses Verhalten ist charakteristisch für den PPO-Algorithmus, da es sich um einen On-Policy-Algorithmus handelt, der längere zusammenhängende Sequenzen benötigt, um stabile Updates der Policy vorzunehmen. DDPG und SAC sind hingegen Off-Policy-Algorithmen und können bereits nach jedem Zeitschritt lernen, sobald ihr Replay-Speicher ausreichend gefüllt ist [19]. Im Vergleich zu DDPG und SAC erreicht PPO insbesondere im Szenario S3 deutlich geringere Episodenscores. Die Trainingsergebnisse lassen darauf schließen, dass für die betrachteten Trainingsszenarien insbesondere DDPG und SAC besser geeignet sind. Um diese Annahme zu überprüfen, werden die trainierten Agenten im nächsten Abschnitt mit den zuvor nicht gesehenen Testdaten evaluiert.

4.2 Evaluation

Zur Evaluation werden die trainierten Agenten mit dem Testdatensatz in der Simulationsumgebung untersucht. Ziel der Evaluation ist es zu überprüfen, ob die Agenten generalisieren. Es wird analysiert, ob die Agenten die Trainingsdaten memorisiert haben und folglich nicht auf unbekannte Daten generalisieren können (Overfitting) oder ob sie die

Dynamiken der Umgebung gelernt haben und ihre erlernte Strategie auch auf unbekannte Daten gewinnmaximierend anwenden können.

Als Bewertungsgrößen werden die zeitlichen Auslastungsverläufe von Transformator und Kabel sowie der Verlauf des Dimmfaktors herangezogen. In der ungeregelten Referenzlast (Abb. 2) werden in dem betrachteten Zeitraum 21,66 MWh geladen. Ein optimales Ergebnis liegt vor, wenn diese Lademenge bei gleichzeitiger Vermeidung von Lastspitzen oberhalb von 50% der maximal zulässigen Auslastung erreicht wird.

4.2.1 Szenario 1

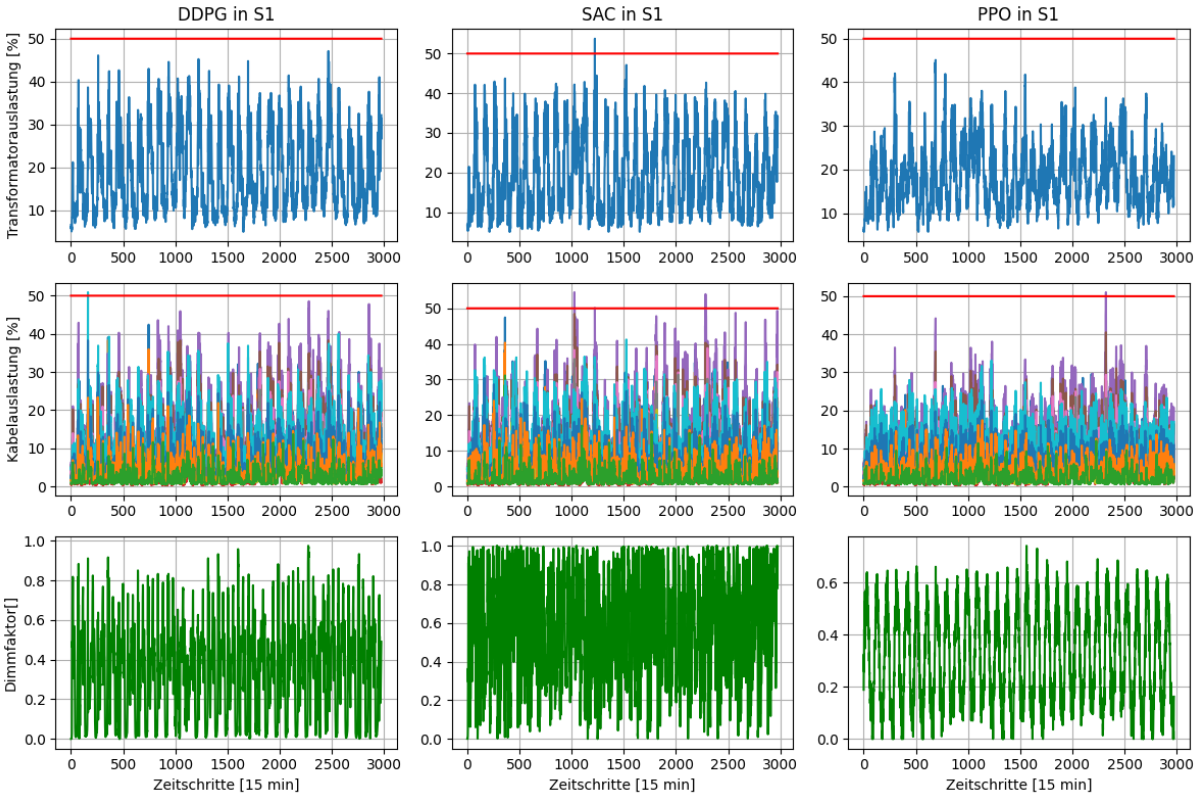

Abbildung 6: Darstellung der Transformatorauslastung, der Kabelauslastung und des Dimmfaktors im Test-Szenario 1

Tabelle 1: Auswertung der Grenzwertverletzungen und des Ladestroms in S1

Algorithmus	Transformator-Verletzungen	Kabel-Verletzungen	Max. Transformatorlast [%]	Max. Kabellast [%]	Ladeenergie [%]
DDPG	0	1 (≈2,8%)	47,1 (34,2%)	50,9 (28,1%)	**92,98**
SAC	1 (≈3,5%) *	5 (≈13,9%)	53,8 (24,9%)	54,5 (23,0%)	**96,29**
PPO	0	1 (≈2,8%)	45,1 (37%)	51,0 (28,0%)	91,62

**() - Klammern: Verhältnis zum ungeregelten Zustand*

Die Auslastungsverläufe von Transformator und Kabel im Testdatensatz (Abb. 7) zeigen, dass alle drei Agenten eine Strategie gelernt haben, die die relevanten Netzkomponenten unterhalb der definierten Grenzwerte betreibt. Dies deutet auf eine erfolgreiche Generalisierung der erlernten Strategien hin. Der Anteil der geladenen Energie relativ zur angeforderten EV-Ladung ist in Tabelle 1 dargestellt. Der SAC-Algorithmus erreicht mit 96,29 % den höchsten Erfüllungsgrad, während der PPO-Algorithmus mit 91,62 % den geringsten Anteil aufweist. Die

Unterschiede spiegeln sich im Verlauf der Dimmfaktorwerte wider: Während DDPG und SAC überwiegend nahe an der maximal zulässigen Ladeleistung operieren, weist PPO über weite Teile des Zeitraums eine reduzierte Ladeleistung auf. Gleichzeitig bleiben die Auslastungen aller Agenten in den meisten Zeitintervallen deutlich unterhalb der Grenzwerte, was auf eine konservative Ausnutzung der verfügbaren Netzkapazitäten hindeutet.

4.2.2 Szenario 2

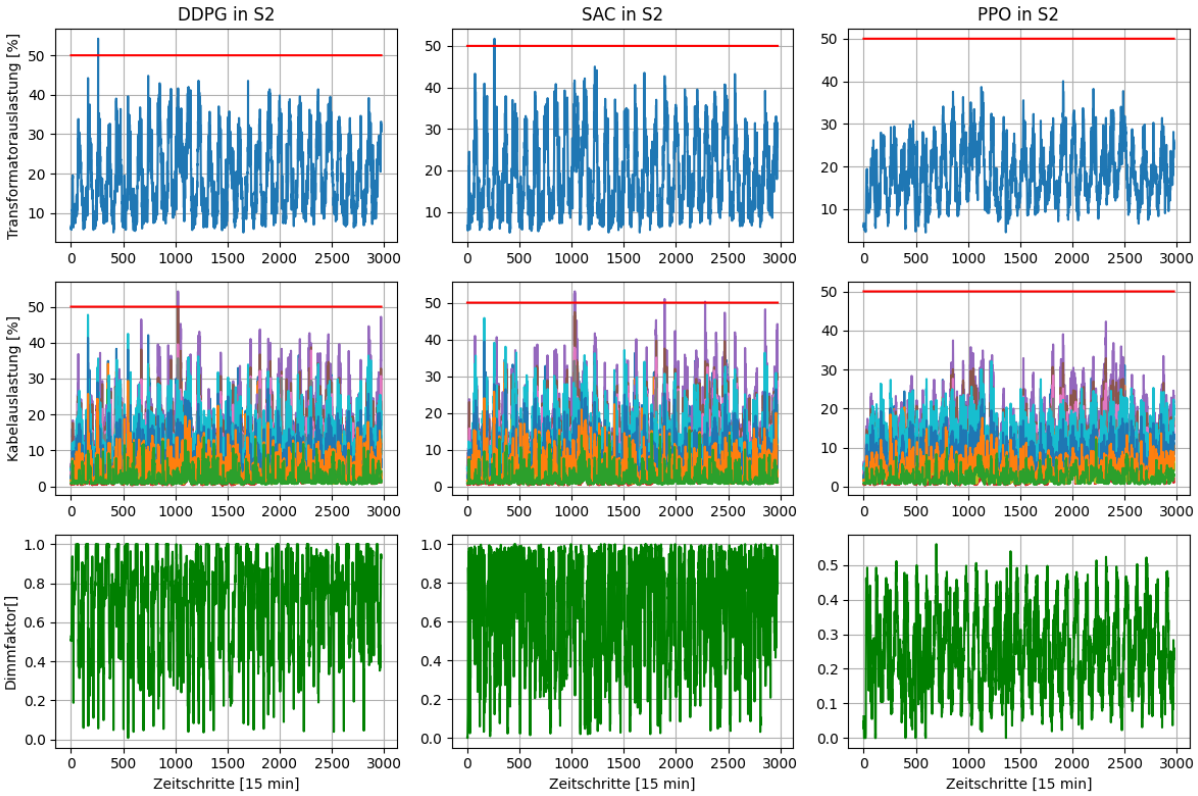

Abbildung 7: Darstellung der Transformatorauslastung, der Kabelauslastung und des Dimmfaktors im Test-Szenario 2

Tabelle 2: Auswertung der Grenzwertverletzungen und des Ladestroms in S2

Algorithmus	Transformator-Verletzungen	Kabel-Verletzungen	Max. Transformatorlast [%]	Max. Kabellast [%]	Ladeenergie [%]
DDPG	1 (≈3,5%) *	2 (≈ 5,6%)	54.2 (24,3%)	54,3 (23,3%)	**96,44**
SAC	1 (≈3,5%)	5 (≈13,9%)	51.7 (27,8%)	53,1 (25,0%)	**96,72**
PPO	0	0	40,1 (43,9%)	42,3 (40,3%)	90,94

() - Klammern: Verhältnis zum ungeregelten Zustand

Auch im zweiten Szenario lässt sich bei allen Kandidaten eine Generalisierung feststellen. Alle Agenten finden auf den Testdaten die Strategie, die die Lasten an besagten Betriebsmitteln unter die Lastgrenze regelt und dabei nicht den Ladestrom vernachlässigt (Abb. 8). Auch hier findet der PPO eine sehr zurückhaltende Strategie, wenn es um die Erfüllung des Ladestroms (90,94 %) geht (Tab. 2). Auch hier wird wieder deutlich, dass das Potenzial nicht gänzlich ausgeschöpft wird, da die Auslastungswerte in den meisten Fällen weit unter der Auslastungsgrenze sind.

4.2.3 Szenario 3

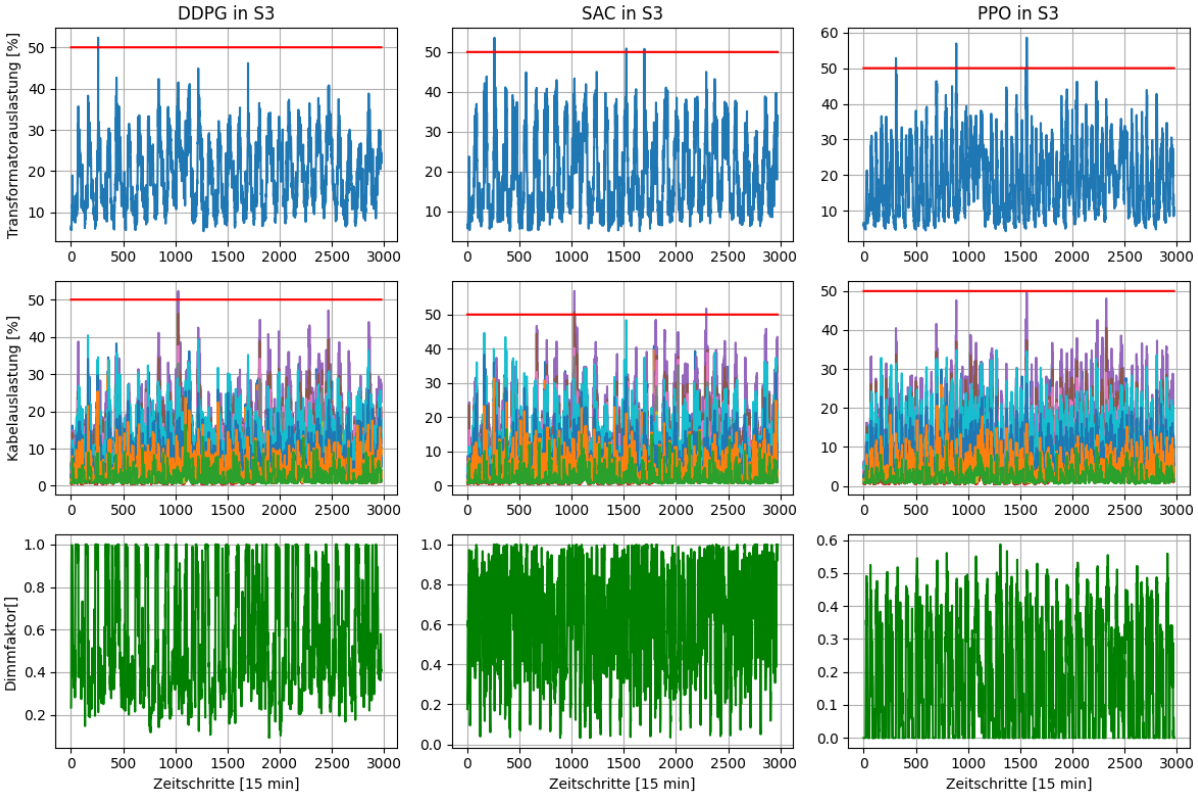

Abbildung 8: Darstellung der Transformatorauslastung, der Kabelauslastung und des Dimmfaktors im Test-Szenario 3

Tabelle 3: Auswertung der Grenzwertverletzungen und des Ladestroms in S3

Algorithmus	Transformator-Verletzungen	Kabel-Verletzungen	Max. Transformatorlast [%]	Max. Kabellast [%]	Ladeenergie [%]
DDPG	1 (≈3,5%) *	2 (≈ 5,6%)	52.3 (27,0%)	52,3 (26,1%)	**94,66**
SAC	3 (≈10.7%)	4 (≈11,1%)	53,6 (25,1%)	56,9 (19,6%)	**97,29**
PPO	4 (≈14,3%)	0	58,5 (18,3%)	49,5 (30,0%)	89,57

**() - Klammern: Verhältnis zum ungeregelten Zustand*

Im dritten Szenario (Abb. 9) ergibt sich ein vergleichbares Bild. Der SAC-Algorithmus erzielt mit 97,29 % den höchsten Anteil geladener Energie, weist jedoch gleichzeitig die höchste Anzahl an Grenzwertverletzungen auf (Tab. 3). Dies verdeutlicht den Zielkonflikt zwischen einer maximalen Erfüllung der Ladeanforderungen und der strikten Einhaltung netzseitiger Beschränkungen.

Die Evaluationsergebnisse zeigen, dass alle drei untersuchten Reinforcement-Learning-Algorithmen in der Lage sind, wirksame und auf unbekannte Daten generalisierende Lastmanagementstrategien zu erlernen. Der SAC-Algorithmus erzielt mit 96–97 % der angeforderten Ladeenergie die höchsten Erfüllungsgrade, gefolgt von DDPG (93–96 %) und PPO (90–92 %). Darüber hinaus konnte gezeigt werden, dass bereits unter Verwendung ausschließlich netzseitiger Informationen (Szenario 1) ein effektives Lastmanagement realisiert werden kann. Die Einbeziehung zusätzlicher kundenseitiger Informationen führt

lediglich zu marginalen Verbesserungen der Performanz, was die Anwendbarkeit des Ansatzes unter Datenschutz- und Umsetzungsaspekten unterstreicht.

5 Fazit und Ausblick

Diese Arbeit untersucht den Einsatz tiefer Reinforcement-Learning-Verfahren (DRL) zur präventiven Regelung von Lastspitzen in Niederspannungsnetzen unter Variation der verfügbaren Beobachtungsräume. Hierzu wurden die Algorithmen DDPG, SAC und PPO systematisch miteinander verglichen. Zusätzlich wurden unterschiedliche Kombinationen aus netzseitigen und kundenseitigen Messdaten, wie sie einem Netzbetreiber realistisch zur Verfügung stehen, analysiert. Ziel war es zu bestimmen, welche Informationsbasis ein datengetriebener Regelalgorithmus benötigt, um Lastspitzen in Echtzeit wirksam zu begrenzen. Als Datengrundlage dienten reale Haushaltslastprofile in Kombination mit Ladeinfrastruktur aus dem Pecan Street Datensatz. Die Ergebnisse zeigen, dass DRL-Algorithmen auch in informationsarmen Szenarien, die ausschließlich auf netzseitigen Messdaten basieren, in der Lage sind, ein effektives Last- und Lademanagement zu realisieren. Dabei werden sowohl Transformator- als auch Kabelauslastungen zuverlässig reduziert. Zudem konnte nachgewiesen werden, dass die Agenten auf unbekannten Datensätzen generalisieren und somit keine reine Anpassung an die Trainingsdaten vorliegt. Die Analyse ergab weiterhin, dass die Einbeziehung zusätzlicher kundenseitiger Informationen die Performanz der Agenten nur marginal verbessert. Dies unterstreicht die praktische Umsetzbarkeit des Ansatzes unter Berücksichtigung von Datenschutz- und Implementierungsaspekten. Gleichzeitig zeigte sich, dass die Agenten die angeforderte Ladeenergie nicht vollständig zur Verfügung stellen und vorhandene Netzkapazitäten nicht in allen Fällen vollständig ausnutzen, was auf weiteres Optimierungspotenzial hinweist. Zu berücksichtigen ist, dass DRL-Verfahren inhärent stochastisch sind und ihre Ergebnisse sowohl von der Initialisierung als auch von den gewählten Trainingsparametern abhängen. Entsprechend können die Resultate zwischen einzelnen Trainingsläufen variieren. Darüber hinaus bilden die verwendeten Datensätze das zukünftige Ladeverhalten elektrischer Verbraucher und Fahrzeuge nur eingeschränkt ab. Eine regelmäßige Aktualisierung der Trainingsdaten ist daher erforderlich, um eine langfristig robuste Performanz sicherzustellen. In dieser Arbeit nicht berücksichtigt wurden unter anderem bidirektionales Laden, der Einfluss dynamischer Strompreise sowie die Einbindung lokaler dezentraler Erzeuger. Zukünftige Arbeiten sollten untersuchen, inwiefern eine Anpassung der Trainingsparameter sowie eine Erweiterung des Beobachtungs- und Aktionsraums die Performanz der Agenten weiter verbessern kann.

Zusammenfassend zeigt diese Arbeit, dass Deep Reinforcement Learning einen vielversprechenden Beitrag zur Bewältigung steigender elektrischer Lasten in Niederspannungsnetzen leisten kann. Der vorgestellte Ansatz ermöglicht ein wirksames Lastmanagement bei gleichzeitigem Verzicht auf sensible Kundendaten und stellt damit eine skalierbare und datenschutzkonforme Alternative zum konventionellen Netzausbau dar.

6 References

[1] „Umwelt Bundesamt,“ 15 04 2025. [Online]. Available: https://www.umweltbundesamt.de/themen/klima-energie/internationale-klimapolitik/uebereinkommen-von-paris/.

[2] „European Parliament,“ 08 08 2023. [Online]. Available: https://www.europarl.europa.eu/topics/en/article/20180920STO14027/reducing-car-emissions-new-co2-targets-for-cars-and-vans-explained.

[3] B. d. Justiz, *Gesetz über die Elektrizitäts- und Gasversorgung (Energiewirtschaftsgesetz - EnWG),* EnWG: https://www.gesetze-im-internet.de/enwg_2005/__14a.html, 2005 (Jahr der Bekanntmachung).

[4] S. Schey, D. Scoffield und J. Smart, „A First Look at the Impact of Electric Vehicle Charging on the Electric Grid in The EV Project," in *EVS26 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*, Los Angeles, California, 2012.

[5] Nationale Leitstelle Ladeinfrastruktur , „Einfach zu Hause laden – Studie zum Ladeverhalten von Privatpersonen mit Elektrofahrzeug und eigener Wallbox," 2025.

[6] V. F. Netztechnik/Netzbetrieb, „Ein Meilenstein auf dem Weg zu intelligenten Verteilnetzen," Bundesnetzagentur, Berlin, 2023.

[7] „Bundesnetzagentur," [Online]. Available: https://www.bundesnetzagentur.de/DE/Vportal/Energie/SteuerbareVBE/artikel.html.

[8] Y. Fu und D. Versen, „Electric Vehicle Charging Management for Avoiding Transformer Congestion Using Policy-based Reinforcement Learning," in *IEEE*, 2023.

[9] D. Versen, „Entwicklung eines netzseitigen Lastmanagements auf Basis intelligenter nichtlinearer Systemidentifikation zur Vermeidung von Transformator-Lastspitzen in einem Niederspannungsnetz," in *TKB*, 2025.

[10] J. Zhang, X. Zhan, T. Li, L. Jiang, J. Yang, Y. Zhang und X. Diao, „A Convex Optimization Algorithm for Electricity Pricing of Charging Stations," *MDPI,* Bd. 12, Nr. 10 - Recent Advances in Nonsmooth Optimization and Analysis, 2019.

[11] H. Ameer, Y. Wang, X. Fan und Z. Chen, „Hybrid optimization of EV charging station placement and pricing using Bender's decomposition and NSGA-II algorithm," *Science Direkt ,* Bd. 397, Nr. Applied Science, 2025.

[12] H. Sun, P. Yuan, Z. Sun, S. Hu, F. Peng und W. Zhou, „Distribution Network Congestion Dispatch Considering Time-Spatial Diversion of Electric Vehicles Charging," *Energies (MDPI),* Bd. 11, Nr. 10, pp. 1-17, 2018.

[13] D. Liu, P. Zeng, S. Cui und C. Song, „Deep Reinforcement Learning for Charging Scheduling of Electric Vehicles Considering Distribution Network Voltage Stability," *MDPI,* Bd. 23, Nr. Optimal Planning, Integration and Control of Smart Grids and Microgrids Systems, p. 1618, 2022.

[14] S. Sykiotis, C. Menos-Aikateriniadis, A. Doulamis, N. Doulamis und P. S. Georgilakis, „A self-sustained EV charging framework with N-step deep reinforcement learning," *Elsevier,* Bd. 35, Nr. Sustainable Energy, Grids and Networks, p. 101124, 2023.

[15] B. d. J. u. Verbraucherschutz, *Gesetz über den Messstellenbetrieb und die Datenkommunikation in intelligenten Energienetzen1 (Messstellenbetriebsgesetz - MsbG),* Regelungen zur Datenkommunikation in intelligenten Energienetzen §§ 49 bis 52 .

[16] F. L. D. Silva, C. E. H. Nishida, D. M. Roijers und A. H. R. Costa, „Coordination of Electric Vehicle Charging Through Multiagent Reinforcement Learning," *IEEE,* Bd. 11, Nr. 3, pp. 2347 - 2356, 2019.

[17] F. L. D. Silva, C. E. H. Nishida, D. M. Roijers und A. H. R. Costa, „Coordination of Electric Vehicle Charging Through Multiagent Reinforcement Learning," *IEEE Transactions on Smart Grid,* Bd. 11, Nr. 3, pp. 2347 - 2356, 2020.

[18] A. Poddubnyy, P. Nguyen und H. Slootweg, „Online EV charging controlled by reinforcement learning with experience replay," *Elsevier,* Bd. 36, Nr. Sustainable Energy, Grids and Networks, p. 101162, 2023.

[19] pandapower, „pandapower.readthedocs.io," 2016-2024 . [Online]. Available: https://pandapower.readthedocs.io/en/latest/. [Zugriff am 2024].

[20] „PECAN STREET," [Online]. Available: https://www.pecanstreet.org/dataport/.

[21] M. Lapan, Deep Reinforcement Learning, mitp, 2020.

[22] T. P. Lillicrap, J. J. Hunt, A. Pritzel und N. Heess, „CONTINUOUS CONTROL WITH DEEP REINFORCEMENT," in *arXiv*, London, 2016.

[23] A. Z. P. A. S. L. Tuomas Haarnoja, „Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *arXiv*, 2018.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford und O. Klimov, „Proximal Policy Optimization Algorithms," in *arXiv*, 2017.

[25] N. Andrenacci und M. P. Valentini, „A Literature Review on the Charging Behaviour of Private Electric Vehicles," *MDPI,* p. 29, 2023.

[26] D. An, F. Cui und X. Kang, „Optimal scheduling for charging and discharging of electric vehicles based on deep reinforcement learning," *Frontiers,* Bd. 11, Nr. Sec. Energy Efficiency, 2023.

[27] Verbraucherzentrale, „www.verbraucherzentrale.de," 25 06 2025. [Online]. Available: https://www.verbraucherzentrale.de/wissen/energie/preise-tarife-anbieterwechsel/.

[28] S. Ayyadi, H. Bilil und M. Maaroufi, „Optimal charging of Electric Vehicles in residential area," *Science Direkt,* Bde. %1 von %2Sustainable Energy, Grids and Networks, Nr. 19, 2019.

[29] R. Zhang, Z. Li, C. Wei, Y. Li, J. Yang und S. Su, „Optimization and Solution Method for Electric Vehicle Charging and Discharging Load," in *IEEE 4th International Electrical and Energy Conference*, Wuhan, China , 2021.

7 Abbildungsverzeichnis

8 Tabellenverzeichnis

9 Anhang

Für diese Arbeit wurden die Deep Reinforcement Algorithmen der Plattform Stable-Baselines3 verwendet. Die folgenden Tabellen zeigen die Parametereinstellung der gewählten Algorithmen:

Tabelle 4: Parameterinitialisierung DDPG

Policy_Model	MlpPolicy
Learning_Rate	0,0001
Buffer_Size	1000000
Batch_Size	256
Train_Frequence	1
Gradient_Steps	1
Learning_starts	1000

Tabelle 5: Parameterinitialisierung SAC

Policy_Model	MlpPolicy
Learning_Rate	0,0003
Buffer_Size	1000000
Batch_Size	256
Train_Frequence	1
Gradient_Steps	1
Learning_starts	100

Tabelle 6: Parameterinitialisierung PPO

N_Steps	2048
Learning_Rate	0,001
N_Epochs	15
Batch_Size	128
Clip_Range	0,3
Ent_Coef	0,005
Vf_Coef	0,5