Federated Learning für die multimodale Umgebungswahrnehmung im Bereich des autonomen Fahrens

Pascal Bochyn
Fakultät für Informatik
Hochschule Furtwangen
Furtwangen, Deutschland
pascal.bochyn@hs-furtwangen.de

Sven Eismann
Fakultät für Informatik
Hochschule Furtwangen
Furtwangen, Deutschland
sven.eismann@hs-furtwangen.de

Tammy Lessig
Fakultät für Informatik
Hochschule Furtwangen
Furtwangen, Deutschland
tammy.lessig@hs-furtwangen.de

***Abstract*—Autonomes Fahren ist seit einigen Jahren ein komplexes und aktives Forschungsfeld. Ständig wechselnde Umweltbedingungen im gemischten Straßenverkehr stellen für die aktuell eingesetzten Umgebungswahrnehmungssysteme für autonomes Fahren teilweise ungelöste Probleme dar. Zudem existieren technische Herausforderungen, wie beispielsweise eine hohe Netzwerkauslastung, die durch das Übertragen von vielen Trainingsdaten bei klassischen Machine-Learning-Ansätzen erzeugt wird. Diese Arbeit schlägt ein theoretisches Konzept für ein Umgebungswahrnehmungssystem in einem autonomen Fahrzeug vor. Hierbei werden bestehende Konzepte kombiniert und ergänzt, um ein System zu erhalten, das dem Stand der Technik entspricht. Durch den Einsatz eines multimodalen Sensoraufbaus bestehend aus Kameras, LiDAR-Sensoren, und Radarsensoren, soll das System robust gegenüber extremen Wetterbedingungen und Lichtverhältnissen werden. Für das effiziente, kontinuierliche Training eines Machine-Learning-Objekterkennungsmodells wird ein Peer-to-Peer-Federated-Learning-Ansatz mit Subnetwork Federated Averaging verwendet, um mit Hilfe von Federated Learning die Vorteile von Edge-Computing auszunutzen.**

Schlüsselwörter—Federated Learning, Machine Learning, autonomes Fahren, multimodale Umgebungswahrnehmung, Objekterkennung, Peer-to-Peer

I. Einleitung

Von einem autonom fahrenden Fahrzeug ist die Rede, sobald dieses in der Lage ist, ohne menschliches Zutun zu fahren. Das entspricht den Leveln 3 bis 5 des Klassifizierungssystems der „Society of Automotive Engineers" (SAE) [1], das in Bezug auf autonom fahrende Fahrzeuge den Standard definiert. Fahrzeuge des SAE-Levels 3 sind seit 2022 offiziell von Mercedes auf dem freien Markt in Deutschland für Verbraucher verfügbar [2] und werden im gemischten Verkehr eingesetzt. Ein gemischter Verkehr beschreibt in diesem Fall das Aufeinandertreffen von autonomen Fahrzeugen mit anderen, nicht-autonomen Verkehrsteilnehmern wie beispielsweise nicht-autonome Fahrzeuge oder Fahrradfahrer.

Das autonome Fahrzeug muss auf plötzlich auftretende Ereignisse reagieren und dabei weiterhin Verkehrsregeln beachten, um die Sicherheit der Insassen und der anderen Verkehrsteilnehmer zu garantieren. Dafür ist eine korrekte und präzise Wahrnehmung der Umgebung notwendig. Diese Voraussetzung ist aktuell nicht gegeben. McCarthy analysiert die Unfalldaten autonomer Fahrzeuge in Kalifornien und schlussfolgert anhand der Daten, dass die Fahrzeuge noch nicht vollständig in der Lage sind am Straßenverkehr teilzunehmen [3]. Ein konkretes Beispiel, das für das Versagen der Umgebungswahrnehmung eines autonomen Fahrzeuges herangezogen wird, ist der Unfall eines Tesla-Fahrzeuges [4], bei dem 2016 der Fahrzeugführer verstarb. Der Tesla geriet unter einen Lastkraftwagen, der die Straße kreuzte, da die Umgebungswahrnehmung des Teslas nicht zwischen dem weißen Lastkraftwagen und dem hellen Himmel unterscheiden konnte und der Tesla nicht bremste [4]. Dieser Unfall wird stellvertretend als Beispiel herangezogen, ist aber – wie McCarthy allein für Kalifornien aufzeigt [3] - nicht der Einzige, der durch autonome Fahrzeuge verursacht wurde.

Hieraus lässt sich schließen, dass autonome Fahrzeuge noch nicht in der Lage sind, die Sicherheit aller Verkehrsteilnehmer zu garantieren und die Kontrolle vollständig zu übernehmen. Bevor Fahrentscheidungen an das Fahrzeug übergeben werden können, muss ein robustes System zur Umgebungswahrnehmung umgesetzt werden, das die Grundlage für das sichere Steuern des Fahrzeuges bildet.

Für die Objekterkennung in autonomen Fahrzeugen wird Machine-Learning (ML) eingesetzt, mit Hilfe dessen Modelle trainiert werden, die Objekte erkennen können. Damit die Modelle Objekte wahrnehmen und unterscheiden können, ist es wichtig eine große Menge an Trainingsdaten zur Verfügung zu stellen. Diese Trainingsdaten sind aufwändig zu generieren und benötigen viel Speicherplatz. Wenn die Daten mit mehreren unterschiedlichen Sensoren generiert werden, handelt es sich um einem multimodalen Sensoraufbau. In einem klassischen ML-Ansatz werden Modelle auf einem zentralen Server trainiert, auf Daten, die von den Fahrzeugen gesammelt werden. Um diese Daten von den Fahrzeugen an den Server zu übertragen, wird eine hohe Netzwerkbandbreite benötigt. Zudem müssen die trainierten Modelle wieder an die Fahrzeuge verteilt werden, was wiederum Netzwerkbandbreite in Anspruch nimmt.

In dieser Arbeit wird folgendes Konzept vorgeschlagen: Autonome Fahrzeuge werden mit multimodalen Sensoren ausgestattet, um ihre Umgebung wahrzunehmen. Anhand dieser Sensordaten wird ein ML-Modell mit Hilfe von Federated-Learning (FL) trainiert. Das hieraus resultierende Modell wird zur Objekterkennung verwendet und periodisch weitertrainiert. Das Modell soll dabei mindestens eine gleichbleibende Genauigkeit im Vergleich zu einem klassisch trainierten Modell aufzeigen.

Mit dem Einsatz von FL wird auf die Probleme der begrenzten Netzwerkbandbreite, die Verteilung der Daten auf viele Clients und der hohen Last auf einen zentralen Server eingegangen und aufgezeigt, in welchem Maß diese mit den eingesetzten Mitteln gelöst werden können. Zudem werden Lösungsvorschläge präsentiert, die Probleme angehen, die sich aus dem Einsatz von FL ergeben.

Die Grundlagen zu der Umgebungswahrnehmung, dem FL-Verfahren und den Sensoren, die in autonomen Fahrzeugen zum Einsatz kommen, werden in den Abschnitten 2, 3 und 4 behandelt. Abschnitt 5 enthält verwandte Forschung zum Einsatz von FL bei autonomen Fahrzeugen. Ein allgemeiner theoretischer Ansatz zur Integrierung von FL in das System der multimodalen Umgebungswahrnehmung in einem autonomen Fahrzeug wird in Abschnitt 6 vorgestellt. In Abschnitt 7 wird das vorgeschlagene Konzept diskutiert, in Abschnitt 8 das Fazit der Arbeit gezogen. In Abschnitt 9 werden mögliche zukünftige Arbeiten beschrieben, die auf dem hier vorgeschlagenen Konzept aufbauen können.

II. Umgebungswahrnehmung im Bereich des autonomen Fahrens

Autonom fahrende Fahrzeuge benötigen einen Weg, um die Umgebung wahrzunehmen. Hierfür werden Daten gesammelt, die durch verschiedene Komponenten weiterverarbeitet werden [5]. Bilder, die mit Kameras aufgenommen werden, können beispielsweise dazu verwendet werden Objekte auf der Straße oder am Straßenrand zu erkennen. Demnach basiert die Umgebungswahrnehmung auf der Objekterkennung. Auf der Grundlage dieser Informationen werden Entscheidungen getroffen, die anschließend in Aktionen umgesetzt werden, die das Fahrzeug steuern [6]. Um diese Daten zu sammeln, setzen Hersteller autonomer Fahrzeuge auf unterschiedliche Technologien. Tesla zum einen verwendet ausschließlich Kameras, die die Umgebung aufnehmen [7]. Mercedes Benz hingegen verwendet eine Vielzahl an Sensoren, darunter auch LiDAR-Sensoren, Kameras und Radarsensoren, um mit Hilfe ihres „DRIVE PILOT" ein SAE-Level 3 umzusetzen [2].

Die Umsetzung eines multimodalen Sensor-Aufbaus und der effizienten Zusammenführung der Sensordaten ist nicht trivial, wodurch sich ein aktives Feld in der Forschung gebildet hat [8], [9], [10]. Dazu kommt, dass die Wahrnehmung der Umgebung für autonome Fahrzeuge ein komplexes Unterfangen ist, da dabei eine Vielzahl an Dingen beachtet werden muss. Straßenschilder müssen erkannt werden, damit sich das Fahrzeug an alle Regeln hält [11]. Ebenso müssen sich bewegende Objekte erkannt [12], und ihre Bewegungsrichtung und Geschwindigkeit erfasst werden [13], [14]. Diese Informationen sind wichtig für die Kollisionsvermeidung. Damit die unterschiedlichen Objekte besser erkannt werden können, werden die Umgebungsbilder semantisch segmentiert [15], [16].

A. Allgemeines Systenmodell eines autonomen Fahrzeuges

Ein autonomes Fahrzeug hat die Aufgabe selbstständig und ohne Eingriff eines Menschen die Umgebung wahrzunehmen und darauf basierend Entscheidungen zur Fahrweise zu treffen und umzusetzen [1], [17], [18]. Das allgemeine System eines autonomen Fahrzeugs ist in Fig. 1 dargestellt. Dem Fahrzeug stehen die drei Steuerungselemente Bremssystem, Motor und Lenkung zur Verfügung, um Anpassungen umzusetzen. Die Steuerungsparameter für diese müssen in Echtzeit berechnet werden. Dazu werden mehrere Teilsysteme eingesetzt. Das Teilsystem Wahrnehmung setzt Sensoren wie Kameras, Radar oder LiDAR ein, um Informationen über die Objekte, die sich um das Fahrzeug herum befinden, zu erhalten. Ein ML-Algorithmus erkennt in den Sensordaten alle relevanten Objekte. Der aktuelle Standort und aktuelle interne Sensorwerte wie z.B. die Geschwindigkeit des Fahrzeugs werden im Teilsystem der Lokalisierung verwendet. Alle Informationen aus diesen beiden Teilsystemen werden in der Planung durch die Sensorfusion zusammengeführt. Hier wird ein Fahrmanöver bestimmt, das in der aktuellen Situation eine sichere Weiterfahrt verspricht [17]. Das Teilsystem Fahrzeugkontrolle setzt das aus der Planung geforderte Fahrmanöver um, indem es die Steuerungsparameter anpasst.

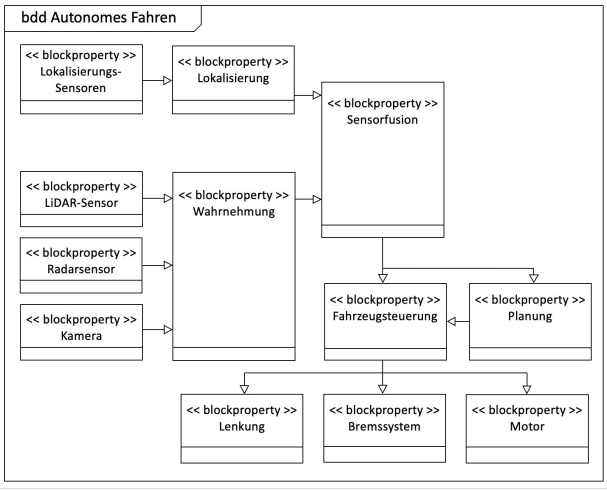

Fig. 1. Allgemeines System eines autonomen Fahrzeuges, basierend auf Jo *et al.* [18]

III. Federated Learning

FL wurde erstmals 2016 von Google als Federated-Optimization vorgestellt [19].

FL bietet eine Möglichkeit ML-Modelle mit Daten zu trainieren, die ungleichmäßig auf einer großen Anzahl an Netzwerkknoten verteilt sind. Dabei kann die Anzahl der Netzwerkknoten größer als die durchschnittliche Anzahl der Trainingsdaten auf den Clients sein [19], [20]. Aus der Arbeit von Konečný *et al.* [19] lässt sich zudem folgender allgemeiner Trainingsvorgang ableiten:

- Das Training wird direkt auf den Netzwerkknoten - die in dieser Arbeit auch als Clients bezeichnet werden - durchgeführt.
- Das Training läuft über mehrere Runden hinweg ab und beteiligt in jeder Runde eine Untermenge an Clients. Hierbei kann die Anzahl der Runden und Clients je nach Anwendungsfall und ausgewählter FL-Strategie variieren.
- Die Trainingsergebnisse der Clients werden nach jeder Runde aggregiert. Nach Abschluss des Trainings steht das resultierende globale Modell allen teilnehmenden Clients, den FL-Teilnehmern, zur Verfügung.

Wie das globale Modell trainiert und die Trainingsergebnisse aggregiert werden und welche Form die Trainingsergebnisse der einzelnen FL-Runde annehmen, ist ebenfalls abhängig von dem Anwendungsfall und der eingesetzten FL-Strategie.

A. Federated-Learning-Strategien

Federated SGD (FedSGD) [20] ist eine kommunikationsintensive Strategie, da pro Gradient-Descent-Schritt eine Menge an Clients gewählt wird, die mit ihrem lokalen Modell jeweils für diesen Schritt den durchschnittlichen Gradienten

auf ihren lokalen Trainingsdaten berechnen und an den zentralen Server schicken. Dieser berechnet den - an der Menge der Trainingsdaten auf den jeweiligen Clients gewichteten - Durchschnitt der Gradienten. Somit muss pro Gradient-Descent-Schritt nur der lokale durchschnittliche Gradient an den Server übertragen werden. Nachdem der gesamte Trainingsvorgang abgeschlossen ist, werden die Modellparameter des trainierten globalen Modells an alle Clients übertragen und diese wenden die neuen Parameter an.

Federated Averaging (FedAvg) [20] benötigt weniger Kommunikation mit dem Server. Hier werden mehrere Gradient-Descent-Schritte auf dem jeweiligen lokalen Modell ausgeführt und die daraus resultierenden Modellparameteränderungen werden an den Server übertragen. Dieser bildet den - an der Menge der Trainingsdaten auf den jeweiligen Clients gewichteten - Durchschnitt der Modellparameteränderungen und wendet diese auf das globale Modell an. Die Modellparameter des trainierten globalen Modells werden auch hier an die Clients verteilt. Aufgrund der geringeren Kommunikation, die zwischen dem Server und den Clients stattfindet, ist eine auf FedAvg basierende Strategie geeigneter für den hier behandelten Anwendungsfall des autonomen Fahrens. Das globale Modell generalisiert im besten Fall die Daten aller Clients gut. Das heißt, dass das Modell auf unbekannten Daten aller Clients eine hohe Genauigkeit erzielt. Da es sich aber in dem FL-Umfeld in der Regel um nicht unabhängig und identisch verteilte Daten handelt, ist es schwieriger ein Modell zu erhalten, das generalisiert, wodurch die lokale Genauigkeit des globalen Modells nachlassen kann [21]. Das Ziel ist jedoch, dass das globale Modell auch auf den lokalen Daten mit gleichbleibender Genauigkeit Vorhersagen erzeugt.

Subnetwork Federated Averaging (SubFedAvg) [21] basiert auf FedAvg. Die lokalen Modelle werden trainiert und bevor die Modellparameteraktualisierungen an den Server übertragen werden, wird unstrukturiertes oder hybrides Pruning [22] auf das entstandene Modell angewendet. Hierdurch wird ein komprimiertes, personalisiertes lokales Modell mit weniger Parametern auf jedem Client erstellt. Die Modellparameteraktualisierungen dieses Modells werden an den Server übertragen und dort aggregiert. Bei der Aggregation wird beachtet, dass sich die Modelle durch das Pruning im Aufbau voneinander unterscheiden, weshalb die Durchschnittsbildung auf dieses Szenario angepasst wird. Mit SubFedAvg wird erreicht, dass die lokale Genauigkeit – demnach die Genauigkeit der lokalen Modelle auf den lokalen Daten – im Vergleich zu Modellen, die keine Personalisierung zulassen, steigt.

Die hier aufgeführten Strategien sind nur eine Auswahl aller FL-Strategien. Die Aggregation von Modellen ist ein aktives Forschungsgebiet und Gegenstand zahlreicher wissenschaftlicher Arbeiten die unterschiedliche Anwendungsfälle fokussieren [23], [24], [25]. Für den Inhalt dieser Arbeit reichen diese Strategie als Grundlagen aus, weshalb keine weiteren FL-Strategien ausgeführt werden.

B. Federated-Learning-Architekturen

Es existieren mehrere Ansätze, um die Client-Server-Architektur umzusetzen. Ein zentraler und ein dezentraler Ansatz.

Der zentrale Ansatz beschreibt einen zentralen Server und mehrere dezentrale Clients. Der Server orchestriert den gesamten Lern- und Aggregationsprozess durch Auswählen der Clients, die in jeder Runde für das Training verwendet werden. Basierend auf der gewählten Strategie variieren die Aufgaben des Servers in Bezug auf das Aggregieren der Trainingsergebnisse.

Der dezentrale Ansatz funktioniert ohne einen zentralen Server. Durch den Aufbau von Peer-to-Peer Verbindungen zwischen mehreren Clients wird ein Netzwerk aufgebaut, in welchem FL angewendet wird [26], [27], [28]. Ein Peer wird durch unterschiedliche Strategien als Server ausgewählt und startet den FL-Trainingsvorgang. Der als Server agierende Peer wählt daraufhin Clients aus dem Peer-to-Peer-Netzwerk für das Training aus. Nach Durchlaufen aller Runden wird das globale Modell an alle Clients übertragen.

C. Nachteile

Bei der Anwendung von FL müssen bestimmte Eigenschaften beachtet werden, die einen negativen Einfluss auf das Training haben können, oder je nach Anwendungsfall die Verwendung von FL ganz ausschließen.

1) Hardware: Da das Modelltraining in FL am Netzwerkrand - das heißt auf dem Endgerät - durchgeführt wird, ist die Hardware des Geräts entscheidend [29]. In Bezug auf die Rechenleistung im Anwendungsfall des autonomen Fahrens, existieren bereits Lösungen von Anbietern wie Nvidia [30] und Texas Instruments [31] die dafür eingesetzt werden können. Diese Lösungen werden zum Teil direkt für den Automobilbereich erstellt und beachten auch den Energiekonsum, der auch eine bedeutsame Rolle spielt.

2) System- und Datenheterogenität: Bei der Verwendung von FL kann es zu einem Leistungsverlust der Modelle kommen, sobald eine erhebliche Diskrepanz zwischen den generierten Daten der Netzwerkteilnehmer besteht [32]. Heterogene Daten können durch unterschiedlich umgesetzte Systeme, also Systemheterogenität, bei den FL-Teilnehmern entstehen [33], [34]. Auf das behandelte Umfeld der Umgebungswahrnehmung übertragen, besteht das Problem darin, dass Fahrzeuge unterschiedlicher Hersteller und Modelle, Sensoren mit verschiedenen Auflösungen - das heißt unterschiedlicher Qualität - und verschiedene CPUs mit unterschiedlicher Rechenleistung verwenden. Daher kann die Qualität und die Anzahl der trainierten Epochen innerhalb eines Zeitraums zwischen den FL-Teilnehmern stark variieren. Je Runde wird für eine bestimmte Anzahl an Epochen für eine Modellparameteraktualisierung, die an den Server geschickt wird, trainiert. Hierbei kann es passieren, dass ein FL-Teilnehmer die aktualisierten Modellparameter zu spät übertragt und das globale Modell schon erstellt wurde. Bei zu hohen Anforderungen an die Hardware der FL-Teilnehmer besteht die Gefahr, dass zu wenige Fahrzeuge in der Lage sind, schnell genug zu trainieren. Um das globale Modell trotzdem mit genug Daten zu trainieren, sind mehr Runden, und dadurch ein längerer Trainingsprozess nötig. Zu niedrige Anforderungen sorgen aber auch für ein langsameres Training, da die Trainingszeit pro Runde steigt. Die standardmäßige Strategie bei FL, FedAvg, löst dieses Problem nicht [32], [34]. Um den negativen Eigenschaften der Systemheterogenität entgegenzuwirken, gibt es in der Forschung den Ansatz variable lokale Arbeitsfortschritte zu tolerieren und in das globale Modell zu aggregieren [34]. Neben der Systemheterogenität der FL-Teilnehmer ist die

Umgebung, in der sich der FL-Teilnehmer befindet, eine weitere Quelle für Datenheterogenität [32], [34]. In verschiedenen Umgebungen können unterschiedliche Objekte ähnlich aussehen und deswegen mit der gleichen Bezeichnung in Verbindung gebracht werden. Bei der Aggregation im globalen Modell kommt es dann zu einer Kollision von gegensätzlichen Informationen, was die Verbesserungsrate des globalen Modells sinken lässt [32], [34].

D. Vorteile

Wenn mit den Nachteilen von FL umgegangen werden kann, bringt FL Vorteile mit sich, die sich wiederum positiv auf das Training und die Gesamtleistung des Systems auswirken können.

1) Netwerklast und Datenspeicherung: FL bedient sich dem Vorteil des Edge-Computings der physischen Nähe, die zu Skalierbarkeit und geringen Latenzen führt [35]. Auf einen klassischen ML-Ansatz übertragen, würde das Training des Modells, in dem Szenario des autonomen Fahrens, auf einem zentralen Server stattfinden. Die Trainingsdaten müssen hierfür an den Server übertragen werden. Unsicherheiten in ML-Modellen können durch die Verwendung von vielen Daten im Training teilweise beseitigt werden [36]. Folglich ist es erstrebenswert, viele Daten zu verwenden. Je mehr Daten jedoch verwendet werden, desto mehr steigt die Netzwerkauslastung, um die Daten an den zentralen Server zu übertragen. Bei der Verwendung von FL können – je nach Strategie – beispielsweise nur Modellparameteraktualisierungen übertragen werden. Durch das Verwenden von effizienten Strategien und Kompression der Modelle kann somit die hohe Bandbreitennutzung begrenzt werden [37]. Aus der dezentralen Architektur lässt sich zudem folgern, dass nicht alle Clients eine Modellparameteraktualisierung an einen Server übertragen, sondern die Last stark auf die Peer-to-Peer Netzwerke verteilt wird. Zudem entfallen die hohen Speicheranforderungen an den zentralen Server, da nicht die Trainingsdaten aller Clients auf diesem persistiert werden müssen. Der Server ist lediglich für die Aggregation der Modelle und die Verteilung des resultierenden Modells zuständig. Die Speicheranforderungen entfallen jedoch nicht komplett, sondern werden an die Clients übertragen. Ein Client muss die Trainingsdaten bis zum Training persistieren. Für das Persistieren muss jedoch kein externes, teures Rechenzentrum verwendet werden. Es kann herkömmliche Hardware zum Speichern der lokalen Daten eingesetzt werden.Ebenso ist die geringe Latenz von Bedeutung, da Entscheidungen in Echt-Zeit getroffen werden müssen, um die Sicherheit während einer autonom gesteuerten Fahrt zu garantieren [38]. Aus diesen Gründen lässt sich folgern, dass Lösungen, die Cloud-Computing – das Gegenstück von Edge-Computing – einsetzen, für den beschriebenen Anwendungsfall nicht geeignet sind. Es muss eine Lösung verwendet werden, die von den Vorteilen des Edge-Computings Gebrauch macht. Dies kann mit FL umgesetzt werden.

2) Kontinuierliches Lernen: Kontinuierliches Lernen ist ein Konzept, das auch in dem klassischen ML-Umfeld existiert [39], [40]. Ein Modell, das bereits trainiert wurde, wird mit neuen Daten erneut trainiert. Die Verwendung von FL impliziert die Verwendung von kontinuierlichem Lernen, da ohne dieses entweder nur eine einzige FL-Runde durchlaufen werden würde oder nach jeder Runde ein völlig neues Modell erstellt werden müsste. Das Szenario des autonomen Fahrens ist für die Verwendung des kontinuierlichen Lernens geeignet, da immer neue Trainingsdaten zur Verfügung stehen, nachdem das Fahrzeug benutzt wurde. Bei jeder Fahrt werden Trainingsdaten aufgezeichnet, die für das Training verwendet werden können. Somit kann das Modell immer weiter verbessert werden.

3) Datenschutz: In der Anwendung des autonomen Fahrens werden Daten gesammelt und für das Training der ML-Modelle verwendet, die als personenbezogen eingestuft werden. Hierbei kann es sich um Ortsdaten handeln, aber auch um Bilder, die im Zuge der Objekterkennung erstellt werden [5]. Dies führt mit den strikten EU-Vorlagen [41], [42] zu Konflikten in der Umsetzung von ML basierten Lösungen. Unternehmen wie Audi sind sich des Konflikts zwischen Datenschutz und Leistung bei autonomen Fahrzeugen bewusst und sehen die Hersteller in der Pflicht, Lösungen zu entwickeln [43]. FL wurde so konzipiert, dass keine personenbezogenen Daten an Dritte weitergegeben werden müssen. FL kann folglich Unternehmen in den EU-Mitgliedstaaten durch die dezentrale Datennutzung ermöglichen, ML-Modelle auf mehr Daten zu trainieren, als dies sonst möglich wäre.

IV. Sensoren

Um die Umgebung wahrzunehmen werden Sensoren benötigt, die die Daten hierfür sammeln. Die aufgezeichneten Daten können zudem fusioniert werden, um das Ergebnis der Objekterkennung durch ML-Modellen zu verbessern. Ein multimodaler Sensoraufbau ist einem Aufbau mit nur einem Sensor überlegen, da die unterschiedlichen Sensoren die Nachteile der jeweils anderen Sensoren ausgleichen können und somit ein robusteres System entsteht [6], [9], [44].

A. Arten von Sensoren

Eine Kamera für das Erfassen der Umgebung zu verwenden ist kein neuer Ansatz. Die Bilder können von ML-Modellen analysiert werden und Aufschluss über Objekte in der Umgebung geben. Jedoch lässt die Bildqualität von Kameras in dunklen Umgebungen – beispielsweise bei Nacht – nach, weshalb eine Kombination mit unter anderem einem LiDAR-Sensor sinnvoll ist [8], [45]. Die Leistung dieses Sensors ist unabhängig der Lichtverhältnisse der Umgebung. LiDAR (Abkürzung für en. „Light Detection and Ranging") ist eine Fernerkundungs-Technik, die sich über viele Jahre hinweg entwickelt hat [46] und in der jetzigen Zeit, vor allem im Bereich des autonomen Fahrens, etabliert ist [10]. Ein LiDAR-Sensor sendet Infrarot- oder Laserstrahlen aus, die auf Objekte treffen und reflektiert werden. Die reflektierten Strahlen werden von dem Sensor aufgenommen und über den Zeitunterschied zwischen Aussenden und Empfangen der Strahlen wird die Distanz zu dem Objekt gemessen. Aus den Distanzen wird eine dreidimensionale Darstellung der Umgebung erstellt. Hierbei ist die Rede von Punktwolken.

Aus der Funktionsweise von LiDAR-Sensoren lässt sich ein Problem ableiten, das aufkommt, wenn ein LiDAR-Sensor bei Regen, Nebel oder Schnee verwendet wird. Die Strahlen

können durch Wetterbedingungen gestört werden. Dies wurde unter anderem von Bijelic *et al.* untersucht und bestätigt [47]. Des Weiteren merken sie an, dass dieses Problem die Entwicklung des autonomen Fahrens beeinflusst und gelöst werden muss, damit autonomes Fahren über dem SAE-Level 4 existieren kann [47].

Ein Sensor, der nicht anfällig für Wetterbedingungen ist, ist ein Radarsensor. Dies lässt sich aus der Funktionsweise eines Radarsensors schließen. Die Funktionsweise unterscheidet sich von der eines LiDAR-Sensors dahingehend, dass Radarwellen anstelle von Laser- oder Infrarotstrahlen eingesetzt werden. Radarwellen sind nicht anfällig gegenüber schlechten Wetterbedingungen [48]. Zhou *et al.* zeigen auf, dass Radarsysteme inzwischen durchaus hochauflösende dreidimensionale Darstellungen der Umgebung erzeugen können [49]. Somit ist es zwar nicht trivial, aber möglich, semantische Informationen aus den Daten zu generieren und für die Umgebungskennung einzusetzen [49]. Auch Li *et al.* setzen eine Kombination von Radar und Kameras ein, um eine dreidimensionale Objekterkennung für Fahrzeuge umzusetzen, die eine höhere Genauigkeit erzielt als ein System, das nur aus Kameras besteht [50].

In Bezug auf ein Fahrzeug gehören LiDAR, Radar und auch Kamera zu äußeren Sensoren, die die dynamische Umgebung eines Fahrzeuges wahrnehmen. Yeong *et al.* [10] teilen Sensoren in zwei Kategorien ein: Sensoren zur Ermittlung und Messung des internen Zustands und die zuvor erwähnten Sensoren zur Ermittlung und Messung der dynamischen Umgebung. Zu den internen Sensoren gehören unter anderem Positionssensoren wie GPS, Sensoren zur Beschleunigungsbestimmung wie Gyro-Sensoren, sowie einer Trägheitsmesseinheiten. Neben den bereits genannten äußeren Sensoren, werden beispielsweise auch Ultraschall-Sensoren in Anwendungen des autonomen Fahrens verwendet [51].

B. Sensorfusion

Für die Zusammenführung der Informationen aus den Sensordaten kommen zwei Zeitpunkte in Frage. Entweder werden die Daten mittels einer Merkmalfusion vor der Entscheidungsfindung zusammengeführt oder mittels einer Entscheidungsfusion nach der Entscheidungsfindung [52]. Die Entscheidungsfindung bezieht sich hier auf das Verwenden der Objektdaten in ML-Modellen, um Fahrentscheidungen zu treffen. Aktuell ist der Einsatz der Merkmalsfusion in der multimodalen Objekterkennung weit verbreitet [25], [53]. Die extrahierten Merkmale aus den Sensordaten werden vor der Objekterkennung so fusioniert, dass sie sich gegenseitig ergänzen. Zum Beispiel können dreidimensionale Punktwolken aus dem LiDAR-Sensor mit den hochauflösenden Pixeldaten aus Kameras angereichert werden, um die Erkennungsrate zu erhöhen [25]. Ein Problem bei der Merkmalsfusion ist, dass die Sensordaten in verschiedenen Perspektiven vorliegen und aufeinander projiziert werden müssen. Dabei kann es zu Informationsverlusten durch Quantisierungsfehler kommen [52]. Die Entscheidungsfusion hat den Vorteil, dass etablierte und erforschte Objekterkennungsalgorithmen für die einzelnen Sensoren eingesetzt werden können. Daher ist der Informationsverlust bei der Objekterkennung gering. Die Ergebnisse aus den Objektdaten der einzelnen Sensoren müssen jedoch noch verglichen und in ein zusammenhängendes Ergebnis umgewandelt werden.

Um die Sensordaten sinnvoll fusionieren zu können, müssen die Sensoren zuvor kalibriert werden. Die Sensoren können durch Erschütterungen oder Temperaturänderungen im Fahrzeug beeinflusst werden und müssen automatisch rekalibriert werden [10]. Die Methoden zur Kalibrierung der Sensoren stellen ein aktives Feld der Forschung dar und werden hier nicht weiter behandelt.

V. Stand der Technik

In der Arbeit von Agarwal *et al.* [54] wurde eine Flotte von Fahrzeugen der Marke Ford mit einer Reihe an Sensoren ausgestattet, darunter mehrere Kameras, LiDAR-Sensoren und GPS-Sensoren. Die autonom fahrende Flotte fuhr durch die Stadt Michigan in den USA, um an verschiedenen Tagen und Tageszeiten eine Gesamtstrecke von 66 Kilometern abzufahren. Die dabei gesammelten Daten wurden anschließend in einem öffentlich zugänglichen Datenset online gestellt.

Posner *et al.* stellen das Konzept eines intelligenten, stationären Netzwerks, „Federated Vehicular Network“ (z. Dt. föderales Fahrzeug-Netzwerk), vor [55]. Das föderale Fahrzeug-Netzwerk setzt sich zusammen aus einer Infrastruktur von Standorten, die Sammelstellen für Netzwerkteilnehmer und deren ML-Modell sind, den Netzwerkteilnehmer - Fahrzeuge auf dem aktuellen Stand der Technik - und ein System für die Kommunikation zwischen den Netzwerkteilnehmern basierend auf Blockchain-Technologien. Das Problem der Vertrauenswürdigkeit anderer Netzwerkteilnehmer in FL wird von Posner *et al.* demnach mittels der auf Blockchain basierenden Kommunikation gelöst [55]. Ebenfalls wird das Problem einer instabilen Verbindung gelöst, indem vorgeschlagen wird das lokale ML-Modell nur an den festgelegten Standorten zu trainieren und anschließend das Resultat an den Standort zu übertragen, der die einzelnen Modelle aggregiert und austeilt [55].

Anstelle des klassischen, serverbasierten Ansatzes finden dezentrale Ansätze von FL in vielen Arbeiten im Bereich des autonomen Fahrens Einzug [26], [27], [28].

Yu *et al.* schlagen einen Entwurf für einen proaktiven Zwischenspeicher (en. „Caching Scheme”) vor, das auf Peer-to-peer-Federated-Deep-Learning basiert [26]. Ziel der Arbeit ist es, eine Lösung für die begrenzte Zwischenspeicherkapazität auf Randgeräten aufzuzeigen, dabei aber auch sensitive Daten des Endgerät-Benutzers zu schützen. Aus diesem Grund wird von Yu *et al.* vorgeschlagen, dass die aktualisierten Modelle aller Netzwerk-Peers in einem zentralen Parameter-Server aggregiert werden [26]. Als mobiler zentraler Server wird ein Fahrzeug anstelle von Straßenrandeinheiten gewählt, um einen häufigen Datenaustausch zwischen den Peers und den Straßenrandeinheiten zu vermeiden und somit dem Problem der begrenzten Zwischenspeicherkapazität entgegenzuwirken.

Der von Wink und Nochta vorgeschlagene dezentrale FL-Ansatz basiert auf der Erstellung eines geteilten Geheimnisses in Form eines ML-Modells zwischen den Peers des FL-Netzwerkes [27]. Ein Initiator verteilt ein initiales Modell an alle Netzwerk-Peers, die auf der Grundlage der lokalen Daten das jeweilige lokale Modell weiter trainieren. Die resultierenden Modelle unterscheiden sich voneinander, da jeder Peer seine eigenen Daten für das Training verwendet. Anschließend führen alle Peers eine sichere Durchschnittsberechnung durch, um ein gemeinsames Modell zu erhalten. Der Ansatz von Wink und Nochta [27] unterscheidet sich dabei maßgeblich von anderen FL-Ansätzen, da keine Modellparameter ausgetauscht werden. Es wird ein Algorithmus angewendet, der jedem Peer eine Mittelwerts-Berechnung der Modellparameter aller Modelle ermöglicht, ohne dass diese ausgetauscht werden müssen. Chen *et al.* stellen einen Ansatz vor, bei dem

Peer-to-Peer-FL mit einer Methode zur Byzantinischen Fehlertoleranz kombiniert wird [28]. Die FL-Teilnehmer nutzen ein Schema für das öffentlich verifizierbare Teilen eines Geheimnisses um die erhaltenen, verschlüsselten Daten zu verifizieren. Damit wird eine sichere Aggregation der Modellparameter gewährleistet - auch beim Ausfallen von Peers. Zudem werden durch die Verwendung von FL die Privatsphäre der FL-Teilnehmer geschützt. Im Gegensatz zu anderen Ansätzen wird von Chen *et al.* [28] auf die Verwendung von Blockchain für die Verifizierung der Kommunikation verzichtet.

Ye *et al.* behandeln das Problem der Datenheterogenität in FL [33]. Um zu vermeiden, dass Modellparameteraktualisierungen an dem globalen Modell mit Modellparametern durchgeführt werden, die die Genauigkeit des globalen Modells senken, wird eine gewisse Modellqualität der Modelle vorausgesetzt. Die Modellqualität wird z.B. über die Bildqualität festgelegt, bei der wiederum die Verzerrung des Bildes die Hauptrolle spielt [33]. Vor dem Prozess der Modellaggregation wird demnach überprüft, ob der FL-Teilnehmer, dessen Modellparameter verwendet werden sollen, eine bestimmte Qualität aufweist.

VI. Konzept

Es werden bestimmte Voraussetzungen in dem hier vorgeschlagenen Konzept angenommen.

- Die autonom fahrenden Fahrzeuge agieren in einem gemischten Verkehr
- Alle Fahrzeuge sind in der Lage miteinander zu kommunizieren.

Basierend auf den genannten Grundlagen wird ein robustes Konzept für ein Umgebungswahrnehmungssystem für autonomes Fahren vorgeschlagen.

Die Architektur des vorgeschlagenen Konzepts beschreibt den Einsatz von multimodalen Sensoren. Hierdurch kann die Umgebung trotz sich verändernden Bedingungen wie den Lichtverhältnissen oder dem Wetter zuverlässig erfasst werden. Die Sensordaten werden so fusioniert, dass ein ML-Modell aus den gesammelten Daten dreidimensionale Objekte erkennen kann. Die erkannten Objekte werden als Ergebnis an das Teilsystem Planung im autonomen Fahrzeug weitergegeben. Für das kontinuierliche Training des ML-Modells wird ein Peer-to-Peer-FL-Ansatz verwendet.

Für die Kommunikation zwischen den Fahrzeugen in dem FL-Netzwerk wird der Einsatz von 5G vorgeschlagen [29].

Da sich die Verwendung von Blockchain für den Daten-Austausch bei FL bewährt [55], [56], [57], wird für dieses Konzept ebenfalls Blockchain zum sicheren Datenaustausch verwendet. Zwar fällt bei der Verwendung von Blockchain eine höhere Rechenlast auf die einzelnen Teilnehmer, jedoch wird dadurch eine sichere Kommunikation zwischen den Teilnehmer sichergestellt. Zusätzlich wird basierend auf Chen *et al.* eine Kompression der Daten durchgeführt, um die Netzwerklast gering zu halten [37].

A. Sensoren

Die Fahrzeuge sind mit einer Anzahl an Sensoren ausgestattet. Die Sensoren befinden sich an strategisch sinnvollen Stellen an dem Fahrzeug. Es existieren sowohl Sensoren an der Vorderseite als auch an der Rückseite und den Fahrzeugseiten [54]. Alle eingesetzten Sensoren werden zuvor kalibriert. Dies ist ein bedeutsamer Schritt, da sonst die resultierenden Daten mitunter unbrauchbar wären [10], [54].

1) Ausgewählte Sensoren: Die Sensoren, die für das Konzept ausgewählt werden, sind

- Kameras
- LiDAR
- Radarsensor

Die Entscheidung für den Einsatz von multimodalen Sensordaten basiert auf einer Steigerung der Erkennungsrate von Objekten bei erschwerten Bedingungen. Durch diese Kombination der Sensoren wird den aufgezeigten Problemen der Wetterbedingungen entgegengewirkt und zudem kann ein Ausfall eines Sensors besser kompensiert werden. Damit entsteht insgesamt ein robusteres System.

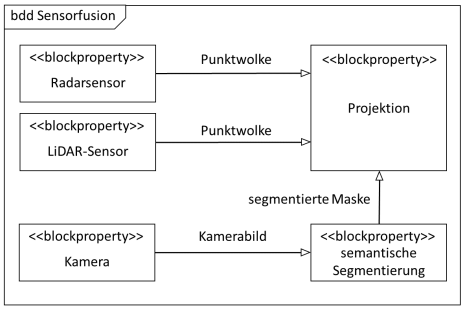

Fig. 2. Blockdefinitionsdiagramm für die Sensorfusion

Für die Zusammenführung der Informationen aus den multimodalen Sensoren soll eine Merkmalsfusion eingesetzt werden. Der Grund dafür ist, dass die fusionierten zweidimensionalen Bilder der Kameras und dreidimensionalen Punktwolken des LiDAR-Sensors einen höheren Informationsgehalt bieten als die Daten der einzelnen Sensoren. Wie auch in vielen umgesetzten autonomen Fahrzeugen soll eine punktweise Fusion von Kamerabildern und LiDAR-Punktwolken eingesetzt werden. Eine vielversprechende Methode dafür ist PointPainting [25]. Die Kamerabilder werden zuerst semantisch segmentiert. Anschließend erfolgt die Projektion der dreidimensionalen Punkte aus dem LiDAR-Sensor auf die segmentierte Maske, um angereicherte Punktwolken zu erhalten. Ein Blockdefinitionsdiagramm für den beschriebenen Prozess der Sensorfusion wird in Fig. 2 gezeigt. Als letzter Schritt wird ein für LiDAR-Daten passender Objekterkennungsalgorithmus eingesetzt, um aus den angereicherten Punktwolken dreidimensionale Objektdaten zu erhalten [25]. Das Radarsystem dient in diesem Konzept als zusätzliche Informationsquelle, die je nach Bedarf als Ersatz bzw. Ergänzung zum LiDAR-Sensor eingesetzt wird. Das ist möglich, weil der Radarsensor – wie von Zhou *et al.* [49] aufgezeigt – wie der LiDAR-Sensor dreidimensionale Daten liefert, die mit den Bildern der Kamera fusioniert werden können.

B. Federated Learning

Das Training der Modelle wird mit FL und SubFedAvg in einem dezentralen Umfeld - also Peer-to-Peer - durchgeführt. Somit wird eine Personalisierung der Modelle per Client zugelassen. Durch die hieraus erwartete entstehende Steigerung

der lokalen Genauigkeit der Modelle, ist eine genauere Wahrnehmung der Umgebung zu erwarten.

Das einzusetzende Modell ist ein vortrainiertes Basismodell, das auf bestehenden Datensätzen, die dem Stand der Technik entsprechen [58], mit konventionellen – nicht föderalen - Trainingsverfahren trainiert wird. Durch den Einsatz von Transferlernen [59] soll die Genauigkeit der lokalen Modelle weiter gesteigert werden. Dies wird von Hsu *et al.* aufgezeigt [60]. Darüber hinaus zeigen sie, dass durch den Einsatz von Transferlernen zusätzlich die Anzahl der benötigten FL-Runden sinkt [60]. FL wird für das Konzept des kontinuierlichen Lernens verwendet. Aufgrund der vorgeschlagenen Strategie des SubFedAvg wird erwartet, dass das globale Modell auf den Daten aller Clients gut generalisiert. Somit wird eine robuste Objekterkennung erwartet, die sowohl eine hohe Genauigkeit auf den Daten aller Clients als auch auf unbekannten Daten hat.

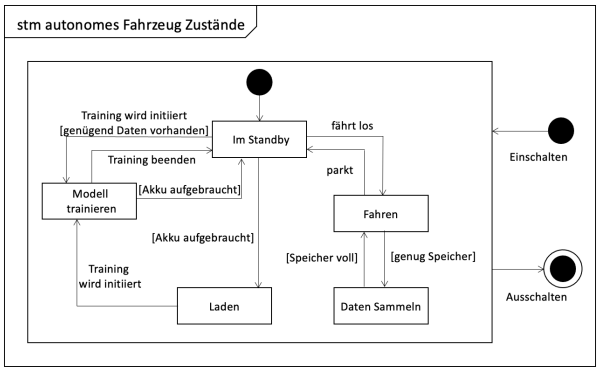

Fig. 3. Zustandsdiagramm autonomes Fahrzeug

In Fig. 3 wird ein vereinfachter Überblick über die Zustände aufgezeigt, die ein autonomes Fahrzeug einnehmen kann. Der FL-Trainingsprozess findet dabei ausschließlich im Zustand „Modell trainieren" statt. Bevor ein Fahrzeug an FL teilnehmen kann, muss es zunächst genügend Daten gesammelt haben. Dies wird ausschließlich während des Fahrens gemacht. Sobald sich das Fahrzeug im Standby befindet bzw. an einem geeigneten Ort befindet, kann das Fahrzeug das Training seines Modells beginnen.

Für die Durchführung von FL gibt es definierte FL-Treffpunkte. Geeignete FL-Treffpunkte sind Orte mit einer bestehenden Infrastruktur für einen großen Abstellplatz für Fahrzeuge, z.B. große Parkplätze. Die Treffpunkte sind entsprechend ausgerüstet, das heißt sie bieten Möglichkeiten zum Aufladen des autonomen Fahrzeugs, damit diesem während des Modelltrainings nicht die Energie ausgeht [55]. Als zusätzliche Sicherung gegen einen zu hohen Energieverlust wird sowohl vor dem Training als auch während des Trainings geprüft, ob das Fahrzeug über ausreichend Energie verfügt, und bei Bedarf wird das Fahrzeug aus dem Training herausgenommen [55], wie in Fig. 3 dargestellt. Sobald sich genügend autonome Fahrzeuge an einem Treffpunkt befinden, können diese gemeinsam ein FL-Netzwerk bilden. In diesem FL-Netzwerk stellt jedes autonome Fahrzeug ein FL-Teilnehmer dar. Unter allen FL-Teilnehmern wird ein FL-Koordinator gewählt, der der FL-Teilnehmer bzw. das Fahrzeug mit den leistungsstärksten Ressourcen ist [55]. Der FL-Koordinator stößt den FL-Trainingsprozess an und empfängt im Anschluss an das Training die Modelle der FL-Teilnehmer und aggregiert diese. Anschließend wird das entstandene Modell anhand eines Testdatensatzes getestet. Mit diesem zusätzlichen Schritt soll sichergestellt werden, dass das neu entstandene Modell eine mindestens gleichbleibende Genauigkeit im Vergleich zum Vorgängermodell aufweist. Anschließend überträgt der FL-Koordinator das globale Modell an die FL-Teilnehmer. Diese entscheiden, ob sie das neue Modell annehmen oder ablehnen, falls die Genauigkeit des globalen Modells der ihres lokalen Modells unterliegt. Der gesamte FL-Trainingsprozess wird wiederholt, bis der Koordinator das Training beendet. Nach jeder abgeschlossenen FL-Runde kann jeder Teilnehmer das Training verlassen und bei Bedarf wird ein neuer FL-Koordinator gewählt.

Die Netzwerkgröße wird beschränkt. Dies ist zum einen, da somit einer Überlastung des Koordinators verhindert wird, wenn eine große Menge an Fahrzeugen an einem Trainingsprozess teilnehmen. Zum anderen soll die Bildung von Subnetzwerken gefördert werden.

Das heißt konkret, dass ein Fahrzeug an einem FL-Treffpunkt verschiedenen FL-Netzwerken beitreten kann und somit eine größere Varianz entsteht.

Jeder FL-Teilnehmer nimmt in regelmäßigen Abständen an einem FL-Trainingsprozess teil. Dabei muss der Treffpunkt nicht immer derselbe sein, sondern kann variieren. Auch müssen die FL-Teilnehmer eines Netzwerkes nicht immer dieselben sein.

Das bei diesem Konzept resultierende Modell wird im Fahrbetrieb bzw. im Zustand „Fahren" auf Fig. 3 von jedem autonomen Fahrzeug des Netzwerkes zur Objekterkennung eingesetzt und bei der nächsten FL-Iteration weiter trainiert. Weitere Komponenten des Fahrzeugs treffen auf der Grundlage der Objekterkennung Entscheidungen und ermöglichen dadurch das autonome Fahren.

VII. Diskussion

Das erstellte Konzept zeigt auf, dass FL einige der Probleme löst, die ein klassischer ML-Ansatz zur Modellerstellung im Bereich des autonomen Fahrens mit sich bringen würde. Darunter fällt die hohe Bandbreitennutzung, die entsteht, wenn viele Fahrzeuge zur selben Zeit ihre Sensordaten an einen zentralen Trainings-Server übertragen. Lediglich Modellparameter werden über das Netzwerk an FL-Teilnehmer ausgeteilt. Netzwerk-Latenzen, die sich aufgrund hoher Netzwerklast ergeben, können somit ebenfalls mittels der Verwendung von FL umgangen werden.

Aufgrund der Architektur von FL, ist es zudem ohne weiteres Zutun in der Lage mit verteilten Daten umzugehen.

Darüber hinaus ergeben sich Vorteile aufgrund der Nutzung von FL im Bereich des autonomen Fahrens. Richtlinien bezüglich des europäischen Datenschutzes können mittels FL eingehalten werden, da die gesammelten Trainingsdaten ausschließlich von dem Fahrzeug genutzt werden, das die Daten generiert hat, und somit niemals das Fahrzeug verlassen. Des Weiteren ermöglicht FL die Anwendung des Konzepts des kontinuierlichen Lernens und den Umgang mit verteilten Daten auf vielen Clients.

Die in diesem Konzept vorgeschlagene Bildung von Subnetzwerken während des FL-Trainingsprozesses soll zu einem effizienteren Training führen. Es wird davon ausgegangen,

dass durch kleinere FL-Netzwerke sowohl die Kommunikation als auch die Modellaggregation profitieren. Zudem soll somit die Last auf den Koordinator minimiert werden.

Es wird erwartet, dass die Kombination von verschiedenen Sensoren eine zuverlässigere Wahrnehmung von Objekten ermöglicht, gegenüber Ansätzen, in denen lediglich ein Sensor zur Umgebungswahrnehmung verwendet wird.

Jedoch bringt FL neue Probleme mit sich, die in weiterführenden Arbeiten bearbeitet werden müssen. Zum einen stellt die clientseitige Hardware ein Problem dar. Der Client muss in der Lage sein, ein ML-Modell zu trainieren, dabei jedoch den Energiekonsum gering zu halten.

Der Einsatz von multimodalen Sensoren führt, neben dem Vorteil eines robusteren Systems und dem Vorteil der Erfassung der Umgebung aus verschiedenen Perspektiven, auch Probleme mit sich, die beachtet werden müssen. Die punktweise Fusionierung der Kamera und des LiDAR-Sensors ist speicherintensiv [52]. Daher steigen die Anforderungen an die Hardware im Fahrzeug. Allerdings ist zu erwarten, dass in naher Zukunft die Hardware ausreichend Leistung bietet. Bei der Merkmalsfusion ist der Informationsverlust durch Quantisierungsfehler bei der Projektion der Sensordaten zu beachten. Eine Möglichkeit den Quantisierungsfehler zu minimieren ist der von Liang *et al.* vorgeschlagene Einsatz von bilinearer Interpolation [53].

VIII. Fazit

Diese Arbeit stellt basierend auf mehreren Ansätzen [18], [25], [26], [52], [55] die im Bereich des autonomen Fahrens vorgeschlagen und belegt wurden, ein Konzept für ein robustes Umgebungswahrnehmungssystem innerhalb eines autonomen Fahrzeugs vor. Für die Umgebungswahrnehmung wird ein multimodaler Sensoraufbau eingesetzt. Zusätzlich zum gängigen Einsatz von Kameras und LiDAR-Sensoren wird der Einsatz eines Radarsystems integriert. Das Radarsystem wird ergänzend zum LiDAR-Sensor eingesetzt, um das System robuster gegenüber extremen Wetterbedingungen zu gestalten. Das eingesetzte ML-Basismodell zur Objekterkennung ist mit bestehenden Datensätzen vortrainiert. Es wird mit Peer-to-Peer-FL und SubFedAvg kontinuierlich verbessert. Aufgrund des dezentralen Peer-to-Peer Ansatzes und da die FL-Teilnehmer nur die aktualisierten Modellparameter austauschen, wird eine niedrigere Netzwerklast erwartet als bei klassischen ML-Ansätzen. Zudem kann mit der Verteilung der Daten auf viele Klienten umgegangen werden. Durch die Verteilung der Modell- und Trainingsdaten auf die leistungsfähige Hardware der FL-Teilnehmer ist auch kein kostenintensiver zentraler Server notwendig.

IX. Ausblick

Das hier vorgeschlagene Konzept gilt in weiteren Arbeiten zu bestätigen und soll als Grundstein für zukünftige Forschungen auf diesem Gebiet fungieren. Des Weiteren soll das Konzept praktisch umgesetzt, evaluiert und ergänzt werden. Dazu gehören die Ermittlung und die Spezifikation einer angemessenen Hardware für das autonome Fahren, sowie die Findung einer passenden Modellarchitektur für die Modelle.

Zudem soll geprüft werden, ob sich die Peer-to-Peer-Architektur in dem vorgeschlagenen Konzept bewährt, oder ob es einer Anpassung der Architektur bedarf.

Im Zuge dessen soll ebenfalls geprüft werden, ob die gewählte Kommunikationsmethode in der Praxis sinnvoll ist.

Eine denkbare Erweiterung des Konzepts würde beinhalten, dass der Sicherheitsaspekt mehr in den Fokus gerückt wird. Der Aspekt möglicher böswilliger FL-Teilnehmer ist nicht außer Acht zu lassen und bedarf weiterer Sicherung.

In weiteren Arbeiten kann genauer auf die Fusionierung der multimodalen Sensoren eingegangen werden. Das Verbesserungspotential liegt hier in der Minimierung des Speicherbedarfs und der benötigten Rechenleistung bei der Sensorfusionierung. Die Erkennungsrate soll dabei mindestens gleichbleiben. Mit den aktuell eingesetzten Verfahren zur Merkmalsfusionierung werden Erkennungsraten von ca. 80% erreicht [52]. Ebenfalls interessant für weitere Arbeiten ist der Einsatz einer Entscheidungsfusionierung. Die Motivation für ein System mit Entscheidungsfusionierung liegt in der Vermeidung von Informationsverlust bei der Fusionierung der Sensordaten auf Merkmalebene. Das System mit Entscheidungsfusionierung kann ähnlich aufgebaut sein wie ein System mit nur einer Art von Sensor. Durch die Parallele Auswertung von z.B. LiDAR-Sensoren und Kameras kann die Robustheit des Systems erhöht werden.

References

[1] Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles, J3016_202104, 2022.

[2] „Vorreiter bei automatisierten Fahr- und Sicherheitstechnologien," mercedenz-benz.com, April 2022. [Online]. Available: https://group.mercedes-benz.com/innovation/case/autonomous/drive-pilot.html. [Zugriff am 9. Januar 2023].

[3] R. L. McCarthy, „Autonomous Vehicle Accident Data Analysis: California OL 316 Reports: 2015–2020," ASME J. Risk Uncertainty Part B, Bd. 8, Nr. 3, September 2022, doi: 10.1115/1.4051779.

[4] The Tesla Team, „A Tragic Loss," tesla.com, [Online]. Available: https://www.tesla.com/blog/tragic-loss. [Zugriff am 22. Januar 2023].

[5] K. Ren, Q. Wang, C. Wang, Z. Qin und X. Lin, „The Security of Autonomous Driving: Threats, Defenses, and Future Directions," in Proceedings of the IEEE, vol. 108, no. 2, pp. 357-372, Feb. 2020, doi: 10.1109/JPROC.2019.2948775.

[6] Z. Huang, C. Lv, Y. Xing und J. Wu, „Multi-Modal Sensor Fusion-Based Deep Neural Network for End-to-End Autonomous Driving With Scene Understanding," IEEE Sensors Journal, Bd. 21, Nr. 10, pp. 11781-11790, 2021, doi: 10.1109/JSEN.2020.3003121.

[7] Tesla, „Tesla Vision Update: Replacing Ultrasonic Sensors with Tesla Vision," Tesla, [Online]. Available: https://www.tesla.com/de_de/support/transitioning-tesla-vision. [Zugriff am 2. Januar 2022].

[8] G. Rizzoli, F. Barbato und P. Zanuttigh, „Multimodal Semantic Segmentation in Autonomous Driving: A Review of Current Approaches and Future Perspectives," Technologies, Bd. 10, Nr. 4, p. 90, Juli 2022, doi: 10.3390/technologies10040090.

[9] D. Khan, M. Baek, M. Y. Kim und D. S. Han, „Multimodal Object Detection and Ranging Based on Camera and Lidar Sensor Fusion for Autonomous Driving," in 2022 27th Asia Pacific Conference on Communications (APCC), Jeju Island, Republic of Korea, 2022, pp. 342-343, doi: 10.1109/APCC55198.2022.9943618.

[10] D. Yeong, G. Velasco-Hernandez, J. Barry und J. Walsh, „Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review," Sensors, Bd. 21, Nr. 6, p. 2140, 2021, doi: 10.3390/s21062140 .

[11] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li und S. Hu, „Traffic-Sign Detection and Classification in the Wild," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2110-2118, doi: 10.1109/CVPR.2016.232.

[12] A. Pfeuffer und K. Dietmayer, „Optimal Sensor Data Fusion Architecture for Object Detection in Adverse Weather Conditions," in 2018 21st International Conference on Information Fusion (FUSION), Cambridge, UK, 2018, pp. 1-8, doi: 10.23919/ICIF.2018.8455757.

[13] N. Wojke, A. Bewley und D. Paulus, „Simple Online and Realtime Tracking with a Deep Association Metric," März 2017. [Online]. Available: arXiv:1703.07402v1.

[14] Y. Xiang, A. Alahi und S. Savarese, „Learning to Track: Online Multi-object Tracking by Decision Making," in 2015 IEEE International

Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 4705-4713, doi: 10.1109/ICCV.2015.534.

[15] Q. Ha, K. Watanabe, T. Karasawa, Y. Ushiku und T. Harada, „MFNet: Towards real-time semantic segmentation for autonomous vehicles with multi-spectral scenes," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 2017, pp. 5108-5115, doi: 10.1109/IROS.2017.8206396.

[16] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu und A. M. López, „Multimodal End-to-End Autonomous Driving," IEEE Transactions on Intelligent Transportation Systems, Bd. 23, Nr. 1, pp. 537-547, Januar 2022, doi: 10.1109/TITS.2020.3013234.

[17] K. Jo, J. Kim, D. Kim, C. Jang und M. Sunwoo, „Development of Autonomous Car—Part I: Distributed System Architecture and Development Process," IEEE Transactions on Industrial Electronics, Bd. 61, Nr. 12, pp. 7131-7140, 2014, doi: 10.1109/TIE.2014.2321342.

[18] K. Jo, J. Kim, D. Kim, C. Jang und M. Sunwoo, „Development of Autonomous Car—Part II: A Case Study on the Implementation of an Autonomous Driving System Based on Distributed Architecture," IEEE Transactions on Industrial Electronics, Bd. 62, Nr. 8, pp. 5119-5132, 2015, doi: 10.1109/TIE.2015.2410258.

[19] J. Konecný, H. B. McMahan, D. Ramage und P. Richtárik, „Federated Optimization: Distributed Machine Learning for On-Device Intelligence," Oktober 2016. [Online]. Available: arXiv:1610.02527v1.

[20] H. B. McMahan, E. Moore, D. Ramage, S. Hampson und B. A. y. Acras, „Communication-Efficient Learning of Deep Networks from Decentralized Data," in 20 th International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Ft. Lauderdale, FL, USA, vol: 54, pp. 1273-1282, Februar 2017.

[21] S. Vahidian, M. Morafah und B. Lin, „Personalized Federated Learning by Structured and Unstructured Pruning under Data Heterogeneity," in 2021 IEEE 41st International Conference on Distributed Computing Systems Workshops (ICDCSW), Washington, DC, USA, 2021, pp. 27-34, doi: 10.1109/ICDCSW53096.2021.00012.

[22] X. Xu, M. S. Park und C. Brick, „Hybrid Pruning: Thinner Sparse Networks for Fast Inference on Edge Devices," November 2018. [Online]. Available: arXiv:1811.00482v1 .

[23] M. Asad, A. Moustafa und T. Ito, „FedOpt: Towards Communication Efficiency and Privacy Preservation in Federated Learning," Applied Science, Bd. 10, Nr. 8, p. 2864, April 2020, doi: 10.3390/app10082864.

[24] D. Li und J. Wang, „FedMD: Heterogenous Federated Learning via Model Distillation," Oktober 2019. [Online]. Available: arXiv:1910.03581v1.

[25] S. Vora, A. H. Lang, B. Helou und O. Beijbom, „PointPainting: Sequential Fusion for 3D Object Detection," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020, pp. 4603-4611, doi: 10.1109/CVPR42600.2020.00466.

[26] Z. Yu, J. Hu, G. Min, H. Xu und J. Mills, „Proactive Content Caching for Internet-of-Vehicles based on Peer-to-Peer Federated Learning," in 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS), Hong Kong, 2020, pp. 601-608, doi: 10.1109/ICPADS51040.2020.00083.

[27] T. Wink und Z. Nochta, „An Approach for Peer-to-Peer Federated Learning," in 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), Taipei, Taiwan, 2021, pp. 150-157, doi: 10.1109/DSN-W52860.2021.00034.

[28] J. -H. Chen, M. -R. Chen, G. -Q. Zeng und J. -S. Weng, „BDFL: A Byzantine-Fault-Tolerance Decentralized Federated Learning Method for Autonomous Vehicle," IEEE Transactions on Vehicular Technology, Bd. 70, Nr. 9, pp. 8639-8652, September 2021, doi: 10.1109/TVT.2021.3102121.

[29] M. Bennis, „Federated Learning and Control at the Wireless Network Edge," GetMobile: Mobile Computing and Communications, Bd. 24, Nr. 3, pp. 9-13, September 2021, doi: 10.1145/3447853.3447857.

[30] „Lösungen für selbstfahrende Autos und autonome Fahrzeuge," nvidia.com, [Online]. Available: https://www.nvidia.com/de-de/self-driving-cars/. [Zugriff am 07 Januar 2023].

[31] „Arm-based processors," ti.com, [Online]. Available: https://www.ti.com/microcontrollers-mcus-processors/processors/arm-based-processors/overview.html. [Zugriff am 08 Januar 2023].

[32] L. Fantauzzo, E. Fanì, D. Caldarola, A. Tavera, F. Cermelli, M. Ciccone und B. Caputo, „FedDrive: Generalizing Federated Learning to Semantic Segmentation in Autonomous Driving," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022, pp. 11504-11511, doi: 10.1109/IROS47612.2022.9981098.

[33] D. Ye, R. Yu, M. Pan und Z. Han, „Federated Learning in Vehicular Edge Computing: A Selective Model Aggregation Approach," IEEE Access, Bd. 8, pp. 23920-23935, 2020, doi: 10.1109/ACCESS.2020.2968399.

[34] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar und V. Smith, „Federated Optimization in Heterogeneous Networks," in Proceedings of Machine Learning and Systems 2020, Austin, TX, USA, 2020.

[35] M. Satyanarayanan, „The Emergence of Edge Computing," Computer, Bd. 50, Nr. 1, pp. 30-39, 2017, doi: 10.1109/MC.2017.9.

[36] A. Kendall und Y. Gal, „What uncertainties do we need in Bayesian deep learning for computer vision?," in Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, Dezember 2017, pp. 5580–5590, doi: 10.48550/arXiv.1703.04977.

[37] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar und S. Cui, „Communication-efficient federated learning," Proceedings of the National Academy of Sciences, Bd. 118, Nr. 17, p. e2024789118, April 2021, doi: 10.1073/pnas.2024789118.

[38] S.-C. Lin, Y. Zhang, C.-H. Hsu, M. Skach, M. E. Haque, L. Tang und J. Mars, „The Architectural Implications of Autonomous Driving: Constraints and Acceleration," in Twenty-Third International Conference on Architectural Support for Programming Languages and Operating Systems, Williamsburg, VA, USA, 2018, pp. 751–766, doi: 10.1145/3173162.3173191.

[39] G. M. v. d. Ven und A. S. Tolias, „Three scenarios for continual learning," April 2019. [Online]. Available: arXiv:1904.07734v1.

[40] G. i. Parisi, R. Kemker, J. L. Part, C. Kanan und S. Wermter, „Continual lifelong learning with neural networks: A review," Neural Networks, Bd. 113, pp. 54-71, Mai 2019, doi: 10.1016/j.neunet.2019.01.012.

[41] „Vorschriften für Unternehmen und Organisationen," europa.eu, [Online]. Available: https://commission.europa.eu/law/law-topic/data-protection/reform/rules-business-and-organisations_de. [Zugriff am 22. Januar 2023].

[42] G. Sartor und F. Lagioia, „The impact of the General Data Protection Regulation (GDPR) on artificial intelligence," Juni 2020. [Online]. Available: https://www.europarl.europa.eu/thinktank/de/document/EPRS_STU(2020)641530.

[43] „Summary: The most important insights," audi.com, Februar 2021. [Online]. Available: https://www.audi.com/en/company/research/and-audi-initiative/audi-autonomous-vehicles-socalty-study-2021-summary.html. [Zugriff am 22. Januar 2023].

[44] D. Feng et al., „Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," IEEE Transactions on Intelligent Transportation Systems, Bd. 22, Nr. 3, pp. 1341-1360, März 2021, doi: 10.1109/TITS.2020.2972974..

[45] S. Muhammad und G. -W. Kim, „Visual Object Detection Based LiDAR Point Cloud Classification," in 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), Busan, Korea (South), 2020, pp. 438-440, doi: 10.1109/BigComp48618.2020.00-32.

[46] J. C. F. Diaz, W. E. Carter, R. L. Shrestha und C. L. Glennie, „ Lidar Remote Sensing," in Handbook of Satellite Applications , New York, NY, Springer, 2013, p. 757–808.

[47] M. Bijelic, T. Grube und W. Ritter, „A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down?," in 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu China, 2018, pp. 760-767, doi: 10.1109/IVS.2018.8500543.

[48] M. Sheeny, E. D. Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang und A. Wallace, „RADIATE: A Radar Dataset for Automotive Perception in Bad Weather," in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 021, pp. 1-7, doi: 10.1109/ICRA48506.2021.9562089.

[49] Y. Zhou, L. Liu, H. Zhao, M. López-Benítez, L. Yu und Y. Yue, „Towards Deep Radar Perception for Autonomous Driving: Datasets, Methods, and Challenges," Sensors, Bd. 22, Nr. 11, p. 4208, 2022, doi: 10.3390/s22114208 .

[50] Z. Li, M. Yan, W. Jiang und P. Xu, „Vehicle Object Detection Based on RGB-Camera and Radar Sensor Fusion," in 2019 International Joint Conference on Information, Media and Engineering (IJCIME), Osaka, Japan, 2019, pp. 164-169, doi: 10.1109/IJCIME49369.2019.00041.

[51] S. Campbell, N. O'Mahony, L. Krpalcova, D. Riordan, J. Walsh, A. Murphy und C. Ryan, „Sensor Technology in Autonomous Vehicles : A review," in 2018 29th Irish Signals and Systems Conference (ISSC), Belfast, UK, 2018, pp. 1-4, doi: 10.1109/ISSC.2018.8585340.

[52] Y. Wang, Q. Mao, H. Zhu, Y. Zhang, J. Ji und Y. Zhang, „Multi-Modal 3D Object Detection in Autonomous Driving: a Survey," Juni 2021. [Online]. Available: arXiv:2106.12735v2.

[53] M. Liang, B. Yang, S. Wang und R. Urtasun, „Deep Continuous Fusion for Multi-sensor 3D Object Detection," in Computer Vision – ECCV 2018. ECCV 2018, Oktober 2018, pp. 663–678, doi: 10.1007/978-3-030-01270-0_39.

[54] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous und J. McBride, „Ford Multi-AV Seasonal Dataset," International Journal of Robotics Research (IJRR), Bd. 39, Nr. 12, pp. 1367-1376, März 2020, doi: 10.48550/arXiv.2003.07969.

[55] J. Posner, L. Tseng, M. Aloqaily und Y. Jararweh, „Federated Learning in Vehicular Networks: Opportunities and Solutions," IEEE Network, Bd. 35, Nr. 2, pp. 152-159, 2021, doi: 10.1109/MNET.011.2000430.

[56] H. Liu, P. Zhang, G. Pu, T. Yang, S. Maharjan und Y. Zhang, „Blockchain Empowered Cooperative Authentication With Data Traceability in Vehicular Edge Computing," in IEEE Transactions on Vehicular Technology, Bd. 69, Nr. 4, pp. 4221-4232, April 2020, doi: 10.1109/TVT.2020.2969722.

[57] Y. Lu, X. Huang, K. Zhang, S. Maharjan und Y. Zhang, „Blockchain Empowered Asynchronous Federated Learning for Secure Data Sharing in Internet of Vehicles," in IEEE Transactions on Vehicular Technology, Bd. 69, Nr. 4, pp. 4298-4311, April 2020, doi: 10.1109/TVT.2020.2973651.

[58] M. Menze und A. Geiger, „Object scene flow for autonomous vehicles," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Bosten, MA, USA, 2015, pp. 3061-3070, doi: 10.1109/CVPR.2015.7298925.

[59] J. Yosinski, J. Clune, Y. Bengio und H. Lipson, „How transferable are features in deep neural networks?," in NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems, Dezember 2014, pp. 3320–3328.

[60] T. Hsu, H. Qi und M. Brown, „Federated Visual Classification with Real-World Data Distribution," in Computer Vision – ECCV 2020. ECCV 2020, November 2020, doi: 10.1007/978-3-030-58607-2_5.