Jonas Schäufler

Erkennung von Cyberangriffen auf LIDAR basierte Wahrnehmungsalgorithmen im Automobil

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang *Master of Science Informatik*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr.-Ing. Franz Korf
Zweitgutachter: Prof. Dr.-Ing. Andreas Meisel

Eingereicht am: 31. Januar 2021

Jonas Schäufler

Thema der Arbeit

Erkennung von Cyberangriffen auf LIDAR basierte Wahrnehmungsalgorithmen im Automobil

Stichworte

IT-Sicherheit, Automotive, LIDAR, Automobil, Wahrnehmung, Objekterkennung, Tracking, Anomalieerkennung, Intrusion Detection

Kurzzusammenfassung

In dieser Arbeit werden Angriffe auf LiDAR basierte Wahrnehmungsalgorithmen eines Automobils anhand einer state-of-the-art Referenzarchitektur analysiert. Basierend auf dieser Analyse werden Anforderungen an Angriffserkennungssysteme abgeleitet und Konzepte zum Schutze der Integrität der Applikationsschnittstelle durch Erkennung der Angriffe vorgestellt. Ausgewählte Konzepte werden als Prototyp implementiert und hinsichtlich ihrer funktionalen Anforderungen und erreichten Qualität ausgewertet.

Jonas Schäufler

Title of Thesis

Detection of Cyber Attacks on Lidar Based Automotive Perception Algorithms

Keywords

IT-Security, Automotive, LIDAR, Perception, Object detection, Tracking, Anomaly Detection, Intrusion Detection

Abstract

In this work, attacks on LiDAR based perception algorithms are analyzed using a reference architecture. Based on this analysis, requirements for intrusion-detection systems which aim to detect these attacks on the perception layer are identified. In the next step, concepts to protect the integrity of the application interface by detecting these attacks

are presented. Selected concepts are implemented as a prototype and evaluated regarding their functional requirements and detection quality.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungen

ADAS Advanced Driver-Assistance Systems.

CAN Controller Area Network.

CCN Convolutional Capsule Network.

CNN Convolutional Neural Network.

.

FoV Field of View.

.

GAN Generative Adversarial Network.

IDS Intrusion Detection System.

IVN Intra-Vehicular Network.

.

MEMS Microelectromechanical Systems.

OBD On-Board-Diagnose.

OPA Optical Phased Array.

.

ToF Time of Flight.

.

1 Einleitung

Neueste Trends bei der Entwicklung eines modernen Automobils erfordern die Digitalisierung und Vernetzung von dessen Komponenten als auch die Verwendung von Technologien, welche ursprünglich nicht für den Einsatz in der Automobilindustrie entworfen worden sind. Einer dieser Trends ist die Entwicklung zusätzlicher Fahrerassistenzsysteme (FAS; englisch Advanced Driver Assistance Systems, ADAS) zu welchen im weiteren Sinne auch Funktionalitäten für den autonomen Betrieb des Fahrzeugs (englisch Autonomous Driving, AD) zählen [29]. Kommunikation mit anderen Verkehrsteilnehmern (englisch Vehicle-to-Vehicle, V2V) und der Infrastruktur (englisch Vehicle-to-Infrastructure, V2I), zusammengefasst als *Vehicle-to-Everything* (V2X), sind unabdingbar für die Realisierung der vollständigen Automatisierung eines Fahrzeugs. Schlüsseltechnlogien für ADAS, wie hochauflösende LiDAR Sensoren als auch Entwicklungen im Infotainment Bereich, stellen erhöhte Ansprüche an das interne Netzwerk des Automobils bezüglich Bandbreite, Dienstgüte (englisch Quality of Service, QoS) und Routing. Dies ist einer der Gründe warum bislang eingesetzte Bus Systeme wie CAN (Controller Area Network), FlexRay oder LIN (Local Interconnect Network) vermehrt durch Gigabit-Ethernet (GigE) ersetzt werden. Dieser technische Wandel des Systems Automobil, als auch das Verlangen nach Erweiterbarkeit als Designprinzip als Folge daraus, dass Funktionalitäten durch die Digitalisierung vermehrt als Softwarekomponenten realisiert werden, stellt eine Herausforderung für die Robustheit des Systems dar. Diese bestimmt sich, zusammen mit Funktionaler Sicherheit (Safety) und Ausfallsicherheit (Reliabilitiy), maßgeblich durch die Informationssicherheit (Security). Folgen der oben genannten Entwicklungen (steigende Komplexität des Systems wie auch zunehmende Vernetzung von dessen Komponenten) sind eine erhöhte Anzahl an Angriffsvektoren und -szenarien. Eine Gruppe von neuen Angriffsszenarien, welche sich aus den genannten Trends ergibt, sind Angriffe auf die maschinelle Wahrnehmung der Umgebung des Automobils mit dem Ziel, ADAS Komponenten zu beeinträchtigen oder zu missbrauchen, um indirekt Einfluss auf das Fahrverhalten zu erlangen.

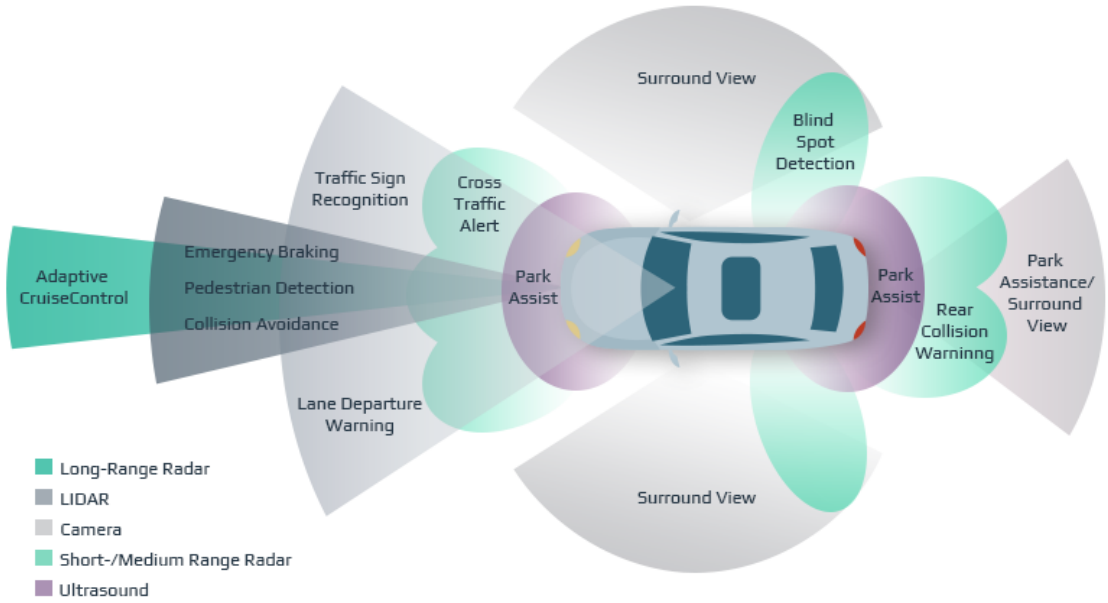

Abbildung 1.1: Überblick: ADAS Funktionalitäten und Sensorik [2]

In dieser Arbeit wird die Anwendung von Intrusion Detection System (IDS) zur Erkennung von Angriffen auf LiDAR basierte Wahrnehmungsalgorithmen erforscht. Dafür werden Angriffe identifiziert und eine Risikobewertung vorgenommen. Basierend auf einer Referenzarchitektur werden aus diesen Angriffen Anforderungen an ein IDS abgeleitet und eine Priorisierung der zu schützenden Attribute vorgenommen. Weiterhin werden Konzepte vorgestellt, wie ein IDS zum Schutz der Integrität der Wahrnehmungsalgorithmen beitragen kann. Ausgewählte Schutzkonzepte werden daraufhin prototypisch implementiert und ausgewertet. Die Arbeit ist in sieben Kapitel gegliedert. Nach der Einleitung, welche den Hintergrund zu dieser Arbeit schildert, folgt in Kapitel 2 die Vermittlung der Grundlagen, welche zum Verständnis der weiteren Kapitel nötig sind. Außerdem wird der aktuelle Stand der Forschung in Gebieten vorgestellt, die mit dem in dieser Arbeit behandelten Thema verwandt sind. Kapitel 3 befasst sich mit der Analyse der Anforderungen an ein IDS. Hier wird zunächst die Referenzarchitektur vorgestellt und im folgenden werden anhand dieser, in Verbindung mit den analysierten Angriffen, die zu schützenden Attribute der Applikationsschnittstelle identifiziert. Daraufhin werden in Kapitel 4 Lösungsansätze vorgestellt und deren Vor- und Nachteile erläutert. Das darauf folgende Kapitel 5 befasst sich mit der Realisierung des Prototypen, hier werden die ausgewählten implementierten Funktionen genannt und Softwaremodule erläutert.

Kapitel 6 befasst sich mit der Auswertung. Hier werden die Ergebnisse der qualitativen Evaluation der Kernfunktionen und der Laufzeitmessung vorgestellt und erörtert. Zuletzt wird die Arbeit in Kapitel 7 noch einmal zusammengefasst und ein Ausblick auf weitere mögliche Forschungsarbeiten gegeben.

2 Grundlagen

Dieses Kapitel dient zur Vermittlung der Grundlagen, auf welchen die weitere Bearbeitung der Problemstellung dieser Arbeit basiert. Abschnitt 2.1 widmet sich hierfür der LiDAR Technologie und beschreibt deren Funktionsweise als auch verschiedene Typen und Bauarten von LiDAR Sensoren. Zusätzlich wird, ab 2.1.3, auf die weitere Verarbeitung der LiDAR-Daten, in einem Automotive Kontext, eingegangen und in Abschnitt 2.1.4 ein Einblick in aktuelle Forschungsthemen in diesem Bereich gegeben. Darauffolgend wird in Abschnitt 2.2 die Problemstellung der Objekterkennung und des Trackings erläutert und exemplarisch dargestellt, wie eine Komponente zur Lösung dieses Problems strukturiert sein könnte. Der Fokus liegt hier nicht auf der Implementierung, sondern auf den Ein- und Ausgabedaten. Diese bilden die Grundlage für die in der Analyse verwendeten Referenzarchitektur, in welcher diese Komponente eine *Black Box* darstellt. In Abschnitt 2.3 werden Sicherheitsmuster und Prinzipien der IT-Sicherheit für cyber-physische Systeme vorgestellt und aktuelle Problematiken erläutert. Im letzten Teil des Kapitels Grundlagen werden in 2.4 die Charakteristiken von Anomalieerkennungs-Problemen dargelegt.

2.1 LiDAR Sensoren

LiDAR (light detection and ranging) Sensoren basieren, ähnlich wie RADAR, auf dem Time of Flight (ToF) Prinzip, mit dem Unterschied, dass statt Radiowellen bei LiDAR Sensoren Laserstrahlen verwendet werden [2]. In ihrer Funktionsweise unterscheidet man zwei Arten von Sensoren [54]: **Flash LiDAR** und **Scanning LiDAR** (siehe Abbildung 2.1).

2.1.1 Flash und Scanning LiDAR

Flash Sensoren beleuchten, ähnlich dem Blitz einer Kamera, das gesamte Sichtfeld (englisch Field of View (FoV)) des Sensors gleichzeitig, während Sensoren die nach dem Scan-

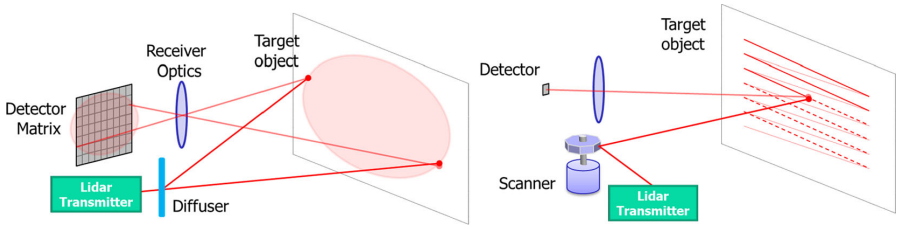

Abbildung 2.1: Funktionsweise verschiedener LiDAR Sensoren.
Flash (links), Scanning (rechts) [54]

ning Prinzip funktionieren das FoV in mehreren Schritten abtasten, entweder reihen- oder spaltenweise. Durch die Bündelung des Lichtes auf eine kleinere Fläche, bieten Scanning LiDAR Sensoren eine höhere Auflösung und liefern auch auf größere Distanzen, je nach Hersteller bis zu 250 Meter, brauchbare Messungen. Auch Kurzstrecken LiDAR Sensoren, mit einer Reichweite bis zu 50 Metern, finden Anwendung in Automobilen, zum Beispiel für *Blind Spot Detection (BSD)* oder *Forward Collision Warning (FCW)*, sind jedoch für die Objekterkennung ungeeignet. [40]

Eine technische Hürde bei der Umsetzung eines Scanning LiDAR ist die Ablenkung der Lichtstrahlen auf den zu beleuchtenden Teilabschnitt. Ein Lösungsansatz ist die Verwendung eines beweglichen Spiegels. Dieser Spiegel ist der Kategorie der Microelectromechanical Systems (MEMS) zugeordnet und wird deshalb auch MEMS-Spiegel genannt. In Abbildung 2.2 ist der prinzipielle Aufbau eines solchen MEMS basierten LiDAR Sensors dargestellt.

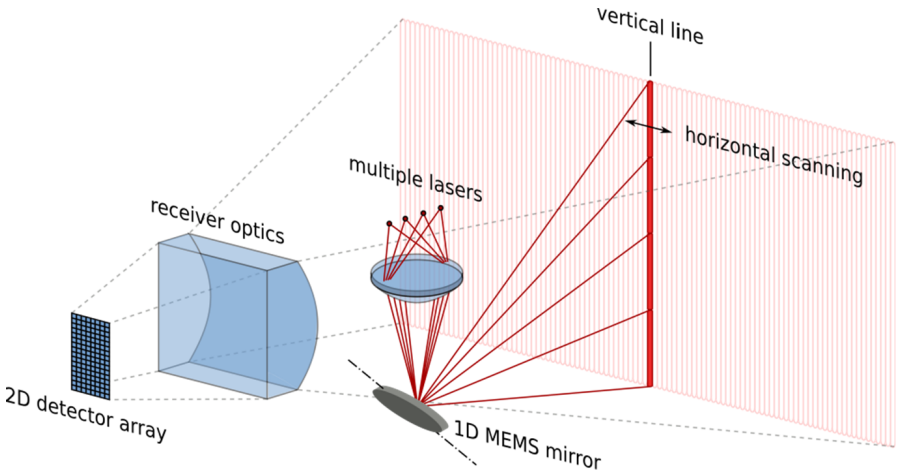

Abbildung 2.2: Aufbau eines Scanning MEMS-Spiegel LiDAR [54]

Dieser mechanische Ansatz hat jedoch den Nachteil, dass das System durch die beweglichen Bauteile weniger robust ist und die Lebenszeit durch deren Abnutzung unvermeidbar geringer ausfällt. Für die Verwendung im Automobil, welches mit einer Lebenszeit von bis zu mehreren Jahrzehnten konzipiert wird, ein großer Nachteil.

2.1.2 Solid-State LiDAR

Neue Generationen von LiDAR Sensoren werden deshalb mit dem Ziel entwickelt, ohne mechanische Bauteile auszukommen. In diesen, als *solid state* LiDAR bezeichneten Sensoren, werden nach heutigem Stand der Technik Optical Phased Arrays (OPAs) statt des Spiegels verwendet, um die Strahlen umzulenken. Das Funktionsprinzip eines OPA entstammt der RADAR Technik und ist dort durch eine Phased-Array-Antenne realisiert. Ein OPA ist also das zur Phased-Array-Antenne analoge Bauteil in der Optik. Ein weiterer Vorteil bei der Vermeidung von Mechanik ist die Möglichkeit zu höherer Messfrequenz. So kann ein Scanframe eines solchen Sensors aus mehreren Tausend Einzelmessungen erstellt werden, wodurch Messabweichungen kompensiert werden können. [13, 54]. Hsu et al. schreiben OPA-LiDARs in ihrem Review aktueller LiDAR Technologien hohes Potential zu, auch wenn die Reife des OPA-LiDAR vergleichsweise gering ist (vgl. [24]).

2.1.3 Punktwolke

Die Struktur der Rohdaten eines Sensors unterscheidet sich von Modell zu Modell. Für den Anwendungsfall der Objekterkennung werden Rohdaten üblicherweise als erstes zu einer Punktwolke aufbereitet, welche gegebenenfalls mit zusätzlichen Informationen aus den Messungen angereichert werden kann, wie zum Beispiel die Intensität des empfangenen Lichts. [28]

Abbildung 2.3 zeigt Punktwolken von vier Langstrecken LiDAR Sensoren der neuesten Generation, aus Perspektive des PKWs an welchem sie montiert sind, während einer Fahrt auf einer Autobahn. Die Punkte sind eingefärbt nach Intensität, von Rot (niedrig) bis Pink (hoch). Durch deren stark reflektierenden Oberflächen sind die in der Entfernung liegenden Straßenschilder (Mitte des Bildes) und die Rückseite eines LKWs deutlich erkennbar. Auch ein überholender PKW in der linken Hälfte des Bildes ist mit bloßem Auge leicht zu identifizieren. Auf der rechten Seite ist ein Teil einer Brücke, wie auch die Fahrbahnbegrenzung, zu sehen.

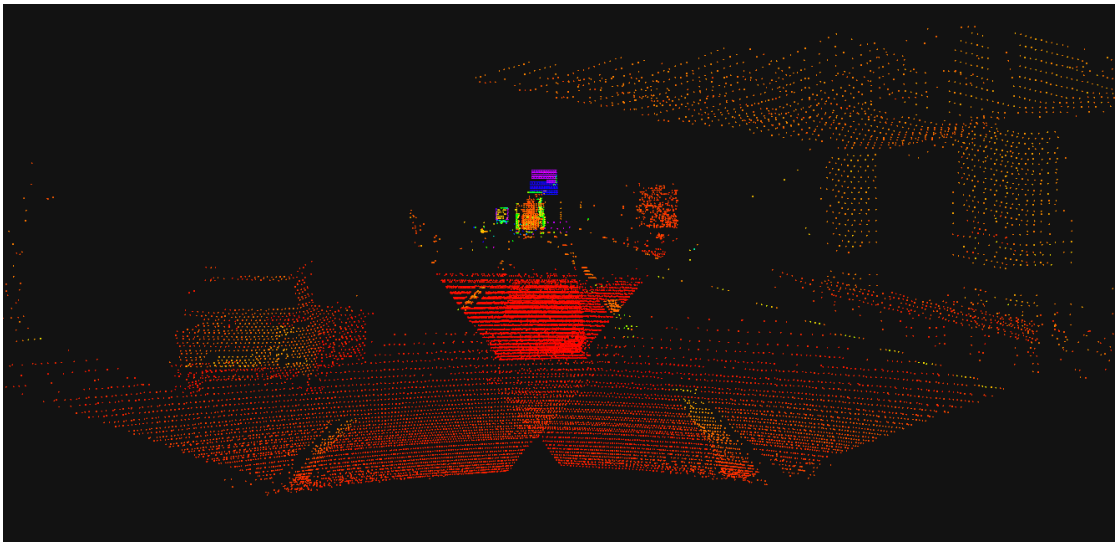

Abbildung 2.3: LiDAR Punktwolken eingefärbt nach Intensität

2.1.4 Aktuelle Forschung: Automotive LiDAR

Seit Jahrzehnten wird an der algorithmischen Erkennung von Objekten im Straßenverkehr geforscht. Damalige Techniken basierten großteils auf Kameradaten und verwendeten die, durch Farben erkannten, Schatten, Texturen und Symmetrien. Die höheren Produktionskosten von *aktiven* Sensoren, als auch technische Hürden wie die nötige Rechenleistung für die Bearbeitung der Daten in Echtzeit, waren Hindernisse beim Einsatz der LiDAR Technologie im Automobilbereich [56]. Weiterentwicklungen im Bereich des Chip-Design und der Netzwerktechnik machen den Einsatz heutzutage möglich. Standards wie 1000Base-T1 [1] und die Herstellung von *automotive grade* GPUs ebnen den Weg für einen Einsatz im Endverbrauchermarkt. Die Optimierung der LiDAR Technologie und die neu erschlossenen Anwendungsfälle bilden die Grundlage für aktuelle Forschung im Bereich der LiDAR basierten Objekterkennung. Ein erkennbarer Trend ist die Verwendung der der LiDAR Technik zu Grunde liegenden Charakteristiken in den Objekterkennungsalgorithmen [9, 28, 36, 31] und der Sensor Fusion [49, 27, 41], durch Anwendung von state of the art Techniken des Maschinellen Lernens. Convolutional Neural Networks (CNNs) können zur Erkennung von Ampeln [53, 10], Verkehrsschildern [19], Gebäuden [30, 10], Fußgängern [37, 31] und anderen Verkehrsteilnehmern [44, 4, 11, 32, 43, 10] verwendet werden. Und auch an der Anwendung neuerer Techniken wie Generative Adversarial Networks (GANs) [48] und Convolutional Capsule Networks (CCNs) [19] wird aktuell geforscht. Klassifikations- und Vorhersageprobleme existieren

nicht nur im Bereich der Objekterkennung. Maschinelles Lernen findet auch in der Sensor Fusion [15, 37, 50], Vor- und Nachverarbeitung (*Pre-/Postprocessing*) [21, 20] und auch bei der Realisierung von LiDAR basierten Tracking Algorithmen [44] und höherliegenden Assistenzfunktionen Anwendung [5].

2.2 Objekterkennung und -tracking

Die Erkennung von Objekten ist Grundvoraussetzung für verschiedene Komfort- und Sicherheitsfunktionen moderner Automobile. Im Kontext des öffentlichen Straßenverkehrs umfasst der Begriff des Objektes einerseits dessen Akteure, also andere motorisierte Verkehrsteilnehmer und Fußgänger, als auch Infrastruktur wie Fahrbahnmarkierungen/-begrenzungen und Straßenschilder sowie Anomalien, zum Beispiel Tiere und Gegenstände auf der Fahrbahn.

Eine Komponente für die Objekterkennung und des Trackings auf Basis einer Punktwolke muss grundlegend zwei verschiedene Problemstellungen lösen:

- Erkennung eines Objektes in der Punktwolke eines einzelnen Scan-Frames
- Zuordnung der Objekte über den zeitlichen und räumlichen Verlauf

Im folgenden wird exemplarisch dargestellt wie die Softwarelösung einer derartigen Komponente aufgebaut sein könnte, und dessen Module werden erläutert.

2.2.1 Module einer Tracking Komponente

Abbildung 2.4 zeigt eine zweistufige Pipeline einer solchen Komponente. In der oberen Stufe sind Module zur Erkennung der Objekte in einer Punktwolke abgebildet. Interne Zustände der einzelnen Module und nicht essentielle Module werden in dieser beispielhaften Architektur einfachheitshalber ausgelassen. Eingabe der oberen Stufe ist eine wie in Abschnitt 2.1 beschriebene Punktwolke.

1. **Clustering** Punkte werden zu Clustern zusammengefasst
2. **Feature Extraction** Cluster werden auf markante Merkmale untersucht
3. **Object Detection** Anhand der Erkennungsmerkmale werden Objekte identifiziert

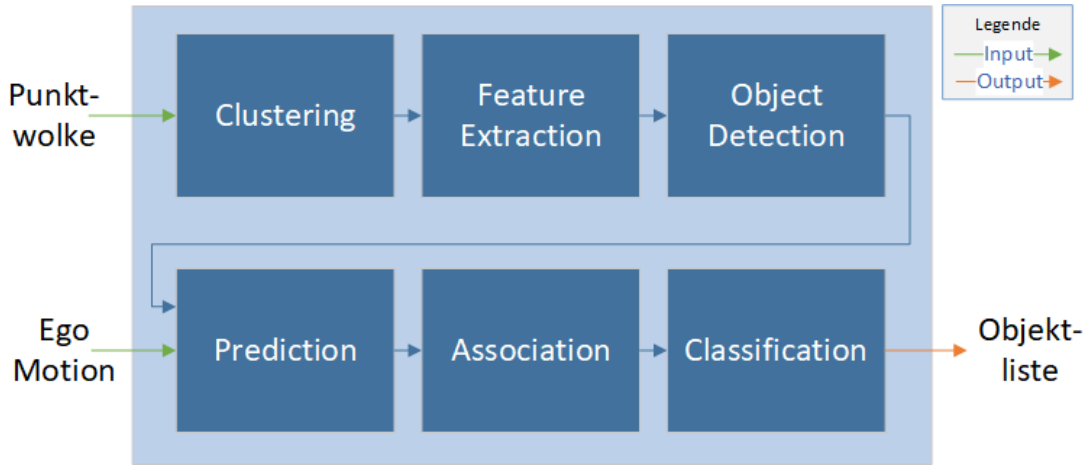

Abbildung 2.4: Module einer Objekt-Tracking-Komponente (exemplarisch)

Als letztes Zwischenergebnis der oben genannten Verarbeitungsschritte entsteht eine Liste der erkannten Objekte im derzeitigen Scan-Frame. Dieses Zwischenergebnis wird zusammen mit der Eigenbewegung des Fahrzeugs, im Schaubild beschriftet als *Ego Motion*, in der unteren Stufe weiter verarbeitet.

1. **Prediction** Unter Berücksichtigung der Eigenbewegung wird eine Vorhersage über den weiteren räumlichen Verlauf der Objekte erstellt.
2. **Association** Die Vorhersage des vorangegangenen Durchlaufes wird mit den aktuell erkannten Objekten verabeitet und einander zugeordet.
3. **Classification** Objekte werden klassifiziert, zum Beispiel als PKW, LKW, oder Motorrad.

2.2.2 Struktur der Ausgabedaten

Die Ausgabe der Komponente ist eine Liste der erkannten Objekte. Attribute eines Objektes sind zum einen dessen geometrische und kinematische Eigenschaften (z.B. Position, Dimension, Geschwindigkeit), können aber auch zusätzliche Informationen der Tracking Komponenten beinhalten (z.B. Klassifikation, Alter, IDs der Sensoren die an den Messungen des Objektes beteiligt waren), oder Kennzahlen über die Qualität der Aussagen über das Objekt (z.B. Konfidenzwert der Klassifikation oder Varianz von Position und Geschwindigkeit). In Abbildung 2.5 ist die Ausgabe einer solchen Trackingkomponente, aus der Vogelperspektive, visualisiert. Erkannte Objekte werden als *Bounding Box* angezeigt

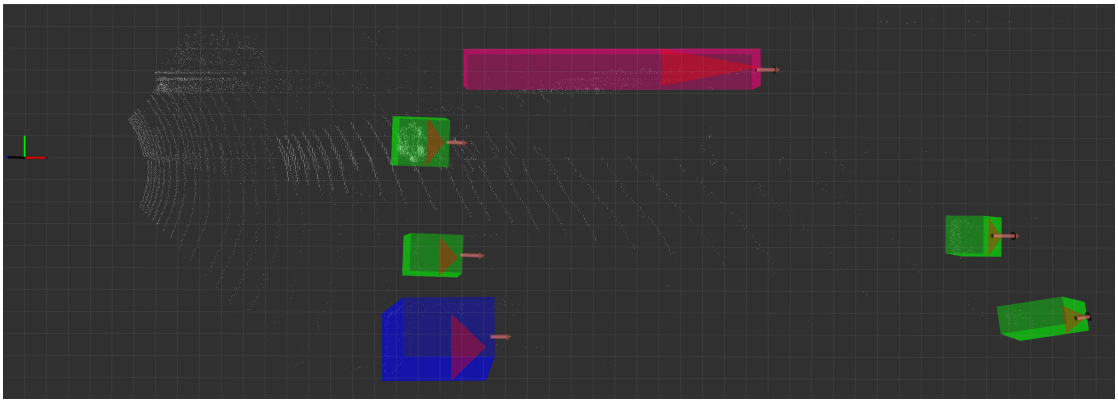

Abbildung 2.5: Visualisierung der Ausgabe der Tracking Komponente

und je nach Klassifizierung koloriert. PKWs werden grün, LKWs werden blau angezeigt. Das rot eingefärbte Objekt ist eine Leitplanke. Ebenso ist der Bewegungsvektor eines Objektes eingezeichnet. Rote Dreiecke markieren die Vorderseite der Objekte.

2.3 Informationssicherheit im Automobil

Die Kernanforderungen der Informationssicherheit Vertraulichkeit, Integrität und Verfügbarkeit müssen in einem Cyber-physischen System wie einem Automobil in einem größeren Kontext betrachtet werden. Ray et al. formulieren hierfür das *Automotive Robustness Problem* [35] um den Zusammenhang zwischen funktionaler Sicherheit, Informationssicherheit, Gerätezuverlässigkeit und Verkehrssicherheit abzubilden und sie stellen Herausforderungen vor, die eine Sicherheitsarchitektur adressieren muss. Die Erweiterbarkeit des Systems und eine geeignete Möglichkeit, einen praktikablen Ausgleich zwischen Authentizität und Anonymität in der *In-Field* Kommunikation herstellen zu können, identifizieren sie als aktuelle Problemstellungen (vgl. [35]). Zusätzlich zu diesen architekturellen Anforderungen sind Problemstellungen hauptsächlich von folgenden Domänenspezifischen Aspekten geprägt [8][35][52][33]:

- Automotive Systeme müssen mit wenig Ressourcen Echtzeitanforderungen erfüllen
- Hoher Grad des Zugangs im Vergleich zu herkömmlichen IT-Systemen
- Zuvor abgeschottetes Intra-Vehicular Network (IVN) nicht auf Sicherheit ausgelegt

Diese Faktoren beeinflussen einerseits Angriffsfläche und Schwachstellen eines solchen Systems, aber eben auch die zu realisierenden Sicherheitsmuster, wie in den folgenden Abschnitten 2.3.1 und 2.3.2 genauer erläutert wird.

2.3.1 Angriffsfläche und Schwachstellen

Der einfache Zugang als auch die ortsunabhängige Kommunikation sorgt für eine komplexe Angriffsfläche. Physische Schnittstellen (Controller Area Network (CAN) Bus oder On-Board-Diagnose (OBD)) bieten direkten Zugriff auf Kernkomponenten des Systems und Schwachstellen in Schnittstellen von Subsystemen können ausgenutzt werden, um in das System einzudringen (Bluetooth, USB). Die starke Vernetztheit mit der Infrastruktur [25] und die Kommunikation mit anderen Verkehrsteilnehmern bieten zusätzlich Möglichkeiten, das System anzugreifen [22][39]. Insbesondere Letzteres sorgt für eine Fülle von Angriffsszenarien gegen ein ortsunabhängiges System in Verbindung mit *Trust* und Authentizität. Auch eine *interne Bedrohung* ([22]) dieser Schutzziele ist gegeben. Bus Protokolle eines IVN (wie CAN, LIN, MOST oder FlexRay) verfügen über keine eigenen Sicherheitsmechanismen zur Verschlüsselung und Authentifizierung und sind anfällig für

Spoofing, Injektion und DoS Angriffe. Der fortschreitende Trend zur Automatisierung, mit dem Streben nach vollständig autonomen Fahrzeugen, verleiht eine zusätzliche Ebene an Komplexität. Die Implikationen von Angriffen auf die Navigation und die Sensorik sind zwei Beispiele zahlreicher ungeklärter Fragestellungen in diesem Bereich [33].

2.3.2 Prinzipien und Muster

V1	Secure the Weakest Link
V2	Practice Defense in Depth
V3	Fail Securely
V4	Follow the Principle of Least Privilege
V5	Compartmentalize
V6	Keep it Simple
V7	Promote Privacy
V8	Remember that Hiding Secrets is Hard
V9	Be Reluctant to Trust
V10	Use Community Resources

Tabelle 2.1: Sicherheitsprinzipien von Viega und McGraw [46]

Der von SAE International herausgegebene Leitfaden J3061 [45] versucht die Problematiken von Cyber-physischen automotive Systemen zu adressieren. Sie erweiterten die Prinzipien (2.1) von Viega und McGraw [46] um zusätzliche domainspezifische Aspekte. Basierend auf J3061 erweitern Cheng et al. [8] das Standard Template zur Beschreibung für Sicherheitsmuster [12] um *automotive* relevante Felder und stellen eine Sammlung von Mustern vor, welche sie, in Verbindung mit den aus J3061 erweiterten Sicherheitsprinzipien (Tabelle 2.2), als geeignet identifizierten.

J1	Protect Personally Identifiable Information and Sensitive Data
J2	Use Principle of Least Privilege
J3	Apply Defense in Depth
J4	Prohibit Software Changes that have not been Thoroughly Analyzed and Tested
J5	Prevent Vehicle Owners from Making Unauthorized Changes

Tabelle 2.2: Automotive Sicherheitsprinzipien aus J3061 [45]

Einschränkungen wie Echtzeitanforderung und knappe Ressourcen haben jedoch auch Auswirkungen auf die Realisierungsmöglichkeiten der Sicherheitsmuster (Tabelle 2.3).

Name	Prinzipien	Ziel
Authorization	V4, V5, V7, J1, J2	Prevention
Blacklist	V2, V5, V9, J3	Prev./Mitigation
DDoS Redundancy	V2, V3, V5, V9, J3	Prev./Mitigation
Firewall	V1, V4, V9, J2	Prev./Detection
Multi-Factor Authentication	V2, V5, V9, J3	Prev.
Multilevel Security	V4, J2, V7, J1, V8, V9,	Prev./Mitigation
Signature IDS	V9	Prev./Mit./ Det.
Symmetric Encryption	V7, J1, V9	Prev.
Tamper Resistance	V3, V4, J2, J4, J5	Prev./Mit./Det.
Third Party Validation	V7, J1, V9	Mit./Detection

Tabelle 2.3: Sicherheitsmuster [8]

Insbesondere kryptographische Verfahren zur Authentifizierung und Verschlüsselung sind teuere Operationen und sie sind auch nicht immer auf die Bus-Protokolle Anwendbar (vgl. [47][52]). Dies macht die Rolle von Erkennungsmechanismen wie Firewalls und IDSs um so wichtiger. Jedoch gilt auch hier, dass state of the art Mechanismen aufgrund einschränkender Bedingungen und hoher Anforderungen schwer realisierbar sind [8]. Auch der cyber-physikalische Aspekt spielt hier eine Rolle, da Prevention Mechanismen eines IDS eine direkte Auswirkung auf das physikalische System haben können. Die Isolation einer kompromittierten ECU durch das IDS kommt einem Systemausfall gleich und führt zu einer direkten Beeinträchtigung des Fahrzeugs, zum Beispiel durch das Abschalten von Assistenzfunktionen und Drosselung, bis hin zum Versetzen des Fahrzeugs in den *sicheren Zustand* (Funktionale Sicherheit). False-Postives sind deshalb für automotive IDS ein deutlich schwerwiegenderer negativer Faktor als in herkömmlichen Problemstellungen [52].

2.3.3 Aktuelle Forschung: Automotive IDS

IDS werden als rückwärtskompatible Lösung gesehen, um die genannten Schwächen des IVN auszugleichen. Sie werden mit den Zielen entwickelt, ein Eindringen unter Zeitgarantien erkennen zu können, vor zukünftigen Angriffen der selben Art zu schützen, und Gegenmaßnahmen einleiten zu können, um weiteren Schaden zu verhindern. (vgl. [52]). Wu et al. unterscheiden in ihrer Übersichtsstudie vier Kategorien von IDS Methodiken (Tabelle 2.4) in Kombination mit vier verschiedenen Parametersätzen: *Busebene* (Datenlänge, Signallänge, Volt, Clock), *Nachrichtenebene* (Zeitintervalle, Frequenzen), *Daten-*

Kategorie	**Technolgien**
Fingerprinting	Clock- und Voltage Fingerprinting
Parameter Monitoring	Regelbasierte Auswertung, Whitelisting, Statistische Auswertung von Frequenz und Zeitintervallen, Remote Frame (CAN)
Informationstheoretische Verfahren	Entropy, Hamming-Abstand
Machine-Learning Verfahren	LSTM, RNN, TCMA, Entscheidungsbäume, HMM, DNN, ANN, Sequenzanalyse, HTM, Petrinetze

Tabelle 2.4: Übersicht: IVN IDS Kategorien und deren Technologien [52]

flussebene (Entropie, Datenmenge, Sequenzen), *Funktionsebene* (Sequenz, Klassifikation und Vorhersage des Verhaltens).

Im Fokus stehen hier vor allem die in Abschnitt 2.3.1 genannten Schwachstellen der vorhandenen Protokolle. An IDS für automotive Ethernet wurde kaum geforscht (vgl. [52]), und auch für IDS auf Funktionsebene, für den Bereich des autonomen Fahrens (maschinelle Wahrnehmung), ist zum Zeitpunkt dieser Arbeit keine Forschung verfügbar. Diesbezügliche Forschungsfelder sind *Domain-Aware* (Infotainment, Body, ADAS, Chassis, etc.), und *Context-Aware* IVN IDS Methoden. *Context* bezeichnet hier den physikalischen Kontext des Systems basierend auf Sensorinformationen. Wasicek et al. [51] verwendeten beispielsweise Referenzmodelle physikalischer Systeme um (im Zusammenhang mit aktuellen Sensorinformationen) Chiptuning und Angriffe auf die Motorsteuerung zu erkennen. Sie weisen außerdem darauf hin, dass es in cyber-physischen Systemen nicht ausreicht, die Semantiken der Nachrichten zu überprüfen, da Angriffe auf das physikalische System durch semantisch korrekte Nachrichten erfolgen können, und deshalb für aktuelle IDS Methoden nicht zu erkennen sind. Dies gilt auch im Zusammenhang mit der maschinellen Wahrnehmung, da diese indirekt Einfluss auf die Aktoren des Systems haben kann und so ein Angriffsvektor auf das physikalische System darstellt. Durch den verhaltensbasierten Ansatz versuchen sie diese Problematik zu umgehen. Verhaltensbasierte Ansätze eignen sich jedoch nicht für Anwendungsfälle, in welchen das IDS durch Prevention unmittelbaren Schaden verhindern soll, wie im Bereich der maschinellen Wahrnehmung kombiniert mit Fahrassistenzfunktionen.

2.4 Anomalieerkennung

Anwendungsfälle eines IDS können als Anomalieerkennungs-Problem formuliert werden, wobei ein Angriff als Abweichung des regulären Verhaltens des Systems definiert wird. Eine Anomalie, in der Statistik auch Ausreißer genannt, bezeichnet im weitesten Sinne die Abweichung von einem Erwartungswert. Die Erkennung von Anomalien ist ein Werkzeug, das in vielen verschiedenen Bereichen Anwendung findet. Chandola et al. erläutern in ihrer Forschungszusammenfassung [6] die Merkmale eines Anomalieerkennung-Problems. Die von Ihnen verwendete Art der Charaktisierung wird in diesem Abschnitt erläutert, um diese in der Analyse (Kapitel 3) auf die Anwendungsfälle dieser Arbeit anzuwenden.

2.4.1 Struktur der Eingabedaten

Ein Hauptmerkmal bei der Charakterisierung ist die Struktur und Art der Eingabedaten. Meist ist die Eingabe eine Menge an Datenpunkten, im folgenden Instanz genannt, welchen ein oder mehrere Attribute zugehörig sein können. Attribute einer einzelnen Instanz können sich wiederum in ihren Typen unterscheiden. Häufig sind diese Attribute diskrete oder stetige numerische Werte (z.B. bei Messungen), oder sie sind kategorischer Art, können aber prinzipiell jegliche Form annehmen. Ein Beispiel, welches auch in der Realität Anwendung findet, wäre die Transaktionshistorie eines Bankkontos. Attribute dieser Instanzen könnten folgende sein: Geldbetrag - diskret numerisch, Ein-/Auszahlung - Binär, Art der Transaktion (Abhebung, Überweisung, etc.) - Kategorisch. Besitzen Instanzen nur ein einzelnes Attribut bezeichnet man sie als univariat, haben Instanzen mehrere Attribute, so werden sie als multivariat bezeichnet.

Ein weiterer Aspekt ist die Beziehung zwischen den einzelnen Datenpunkten. Beispielsweise kann die Menge der Eingabedaten, wie im Falle der bereits erwähnten Messdaten, eine Sequenz (*sequential*) - also linearer Ordnung - sein. In diesem Fall wäre die Ordnung der Dateninstanzen durch die zeitliche Komponente vorgeschrieben (*temporal*). Jedoch gibt es auch andere Kriterien der Ordnung, wie zum Beispiel im Fall von Genom-Sequenzen. Beziehungen zwischen Instanzen beschränken sich auch nicht nur auf deren Ordnung. Ein weiteres Beispiel für eine Beziehung wären Daten mit einer Raumkomponente (*spatial*), dies kann ein Positionsvektor sein, welcher Instanzen auf einen gemeinsamen Raum abbildet. Auch Mischformen sind möglich, beispielweise haben Klimadaten eine räumliche als auch eine zeitliche Komponente, sie werden dann als *spatio-temporal* bezeichnet.

Beziehungen zwischen den Daten können auch abstrakter Natur sein. Liegt die Menge der Eingabedaten als Graphstruktur vor, wobei eine Instanz einem Knoten im Graphen entspricht, erschließt sich die Beziehung zwischen den Instanzen aus den Kanten.

2.4.2 Typen von Anomalien

Ein weiterer Aspekt ist die Form der zu findenden Anomalie. Besteht eine Anomalie aus einer einzelnen Instanz spricht man von einer Punktanomalie. Möchte man zum Beispiel im oben genannten Beispiel, der Transaktionshistorie eines Bankkontos, eine betrügerische Kontobewegung identifizieren, würde dies einer einzelnen Transaktion entsprechen. Besteht eine Anomalie aus mehreren Instanzen wird sie als kollektive Anomalie bezeichnet. Hier müssen die Instanzen einzeln betrachtet nicht unbedingt anomal sein, jedoch stellt das gemeinsame Auftreten eine Anomalie dar. Instanzen dieser Art sind also nur in einem gewissen Kontext anomal, weshalb man sie als Kontextanomalie bezeichnet. Auch Punktanomalien können kontextabhängig sein. Im oben angeführten Beispiel könnte das Abheben einer größeren Summe Geld im Ausland als Betrug gewertet werden, wenn gleichzeitig normale Geldabhebungen im Inland stattfinden - ein Indikator, dass der Kontoinhaber nicht einfach nur im Urlaub ist, sondern dass mit einer geklonten Bankkarte auf das Konto zugegriffen wird. Der Kontext ist also abhängig von der Problemstellung und muss zusammen mit dieser definiert werden. Man unterscheidet deshalb Attribute einer Instanz in verhaltensbezogene (*behavioural*) und kontextbezogene (*contextual*). Kontextuelle Attribute sind oft die im vorherigen Abschnitt genannten Beispiele für Attribute, die Beziehungen zwischen Instanzen herstellen: Positionsvektor, Zeitstempel, et cetera. Nicht jede Anomalie ist einem einzelnen Typen zuzuordnen, so können zum Beispiel kontextuelle Punktanomalien in kollektive Anomalien transformiert werden. Die Wahl des Typus muss also auf den Anwendungsfall und das Vorhandensein, oder gegebenenfalls der Qualität, der Attribute abgestimmt werden.

2.4.3 Trainingsdaten

Der Grad an Abdeckung des Problemraumes der bereits vorliegenden Daten ist ausschlaggebend beim Entwurf einer Lösung, da je nach Datenlage verschiedene Ansätze möglich sind. Generell kann der Modus Operandi einer Anomalieerkennung in drei verschiedene Kategorien unterteilt werden:

1. **Supervised** Anhand gelabelter Daten für anomales und normales Verhalten wird ein Vorhersagemodell erstellt, das als Klassifikator verwendet wird.
2. **Semi-Supervised** Hier liegen keine gelabelten Daten für anomales Verhalten vor. Das Vorhersagemodell wird nur anhand des Normalverhaltens trainiert.
3. **Unsupervised** Training erfolgt mit ungelabelten Daten, unter der Annahme, dass Anomalien in den Trainingsdaten kaum oder garnicht vorhanden sind.

2.4.4 Ausgabe

Die Ausgabe der Anomalieerkennung kann in zwei verschiedenen Formen erfolgen. Eine Möglichkeit ist, Instanzen in anomal und normal zu klassifizieren (*labeling*). Die andere Option ist, Instanzen eine Wertung zuzuweisen, welche das Maß der Abnormalität abbildet (*scoring*). Welche Art der Ausgabe gewählt wird ist abhängig vom jeweiligen Anwendungsfall. [6]

3 Analyse

Changalvala und Malik haben [7] eine, auf Watermarking und Data-Hiding basierte, Methode entwickelt, die Bedrohung auf die Integrität von LiDAR Rohdaten in Form einer Punktwolke zu entschärfen. In ihrer Arbeit betrachteten sie ein Angriffsmodell mit zwei Kategorien *Regular-channel* (1) und *Transmission-channel* (2), unter der Annahme, dass Advanced Driver-Assistance Systems (ADAS) Systeme als Eingabe die Punktwolke erhalten (siehe Abbildung 3.1). In modernen Architekturen ist dies jedoch nicht der Fall. Algorithmen zur Objekterkennung werden auf Domain-spezifischen ECUs integriert. Diese verarbeiten die Rohdaten der LiDAR Sensoren, mit speziell zu diesem Zweck optimierter Hardware und bilden den *Perception* Layer. Um diesen Unterschied der Architektur im Angriffsmodell zu berücksichtigen, erweitern wir es um die Kategorie der *perception-channel* (3) Angriffe. In Abbildung 3.1 ist der Unterschied zwischen den Architekturen und die für diese Arbeit vorgenommenen Erweiterung des Angriffsmodells dargestellt.

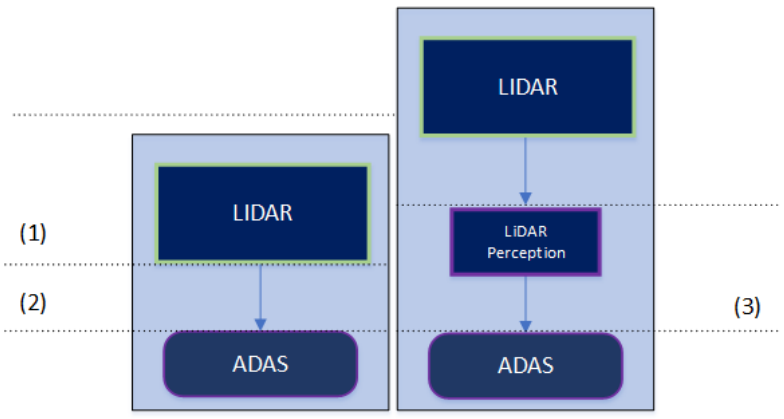

Abbildung 3.1: Architekturausschnitte im Vergleich

Während LiDAR Rohdaten wegen der großen Menge an Daten über Ethernet oder Low Voltage Differential Signaling (LVDS) übertragen werden, ist eine Übertragung der erkannten Objekte per CAN durchaus möglich. Aber auch im Falle von Ethernet sind durch Man-in-the-middle (MITM) oder Replay, Angriffe auf die Nachrichtenübertragung

vorstellbar. Interne Angriffe auf die LiDAR Perception Electronic Control Unit (ECU), zum Beispiel durch das Ausnutzen einer Sicherheitslücke in anderen Funktionen des Systems, oder durch einen Supply-Chain-Angriff können dazu führen, dass die Integrität der Ausführungsumgebung der Algorithmen nicht mehr gegeben ist. Selbst im Falle einer kryptographisch abgesicherten Transportschicht wäre also eine Bedrohung vorhanden, die hinsichtlich dessen, dass die Wahrnehmung des Automobils einen direkten Einfluss auf die Unversehrtheit deren Insassen und anderer Verkehrsteilnehmer hat, ein Risiko birgt. Die steigende Anzahl der am System beteiligten Zulieferer, als auch der Trend zu kürzeren Lieferzyklen mit In-Field Updates, sind Faktoren, die in Zukunft zu einer größeren Anzahl an ausnutzbaren Schwachstellen innerhalb des IVN führen können. Hier können IDS als Methoden zur Anwendung der Prinzipien V2, J3 und V9 (siehe Tabelle 2.2 in Abschnitt 2.3.2) verwendet werden, um die Integrität der Ausführungsumgebung und der Ergebnisse der Wahrnehmungsalgorithmen vor Angriffen innerhalb des *perception-layer* zu schützen.

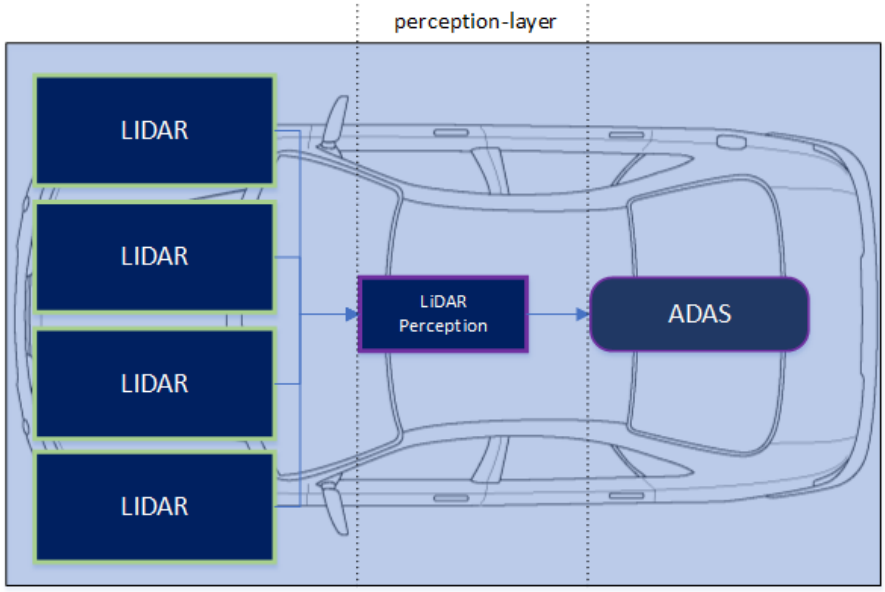

Abbildung 3.2: LiDAR Domain Architektur mit *perception-layer*

Abbildung 3.2 zeigt den Aufbau einer modernen LiDAR Domain Architektur mit vier frontalen Sensoren, die als Referenz für die weitere Analyse der Problemstellung herangezogen wird. Deren Komponenten und deren Nachrichten werden im folgenden Abschnitt 3.1 genauer erläutert.

3.1 Referenzarchitektur

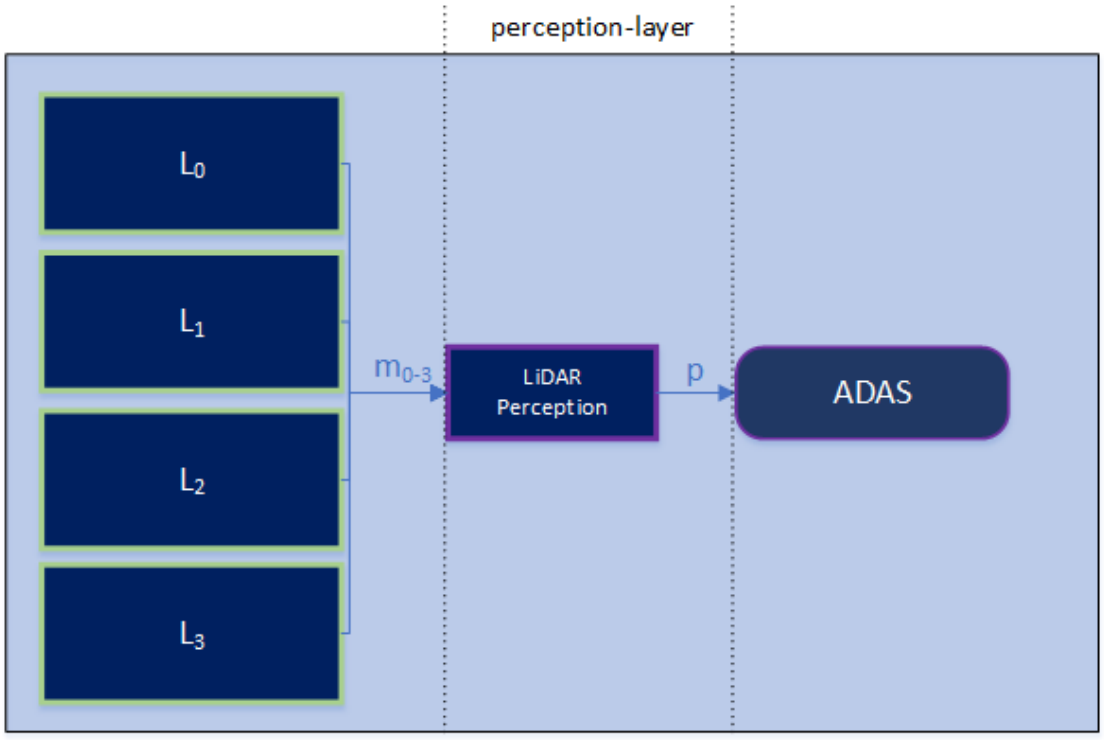

Abbildung 3.3: Beispiel einer modernen LiDAR Domain Architektur

Abbildung 3.3 zeigt die Referenzarchitektur, welche eine wie in Abschnitt 2.2 beschriebene Komponente für Objekterkennung und -tracking integriert. Deren Eingaben sind die Punktwolken ($m_{0\text{-}3}$) vierer LiDAR Sensoren ($L_{0\text{-}3}$). Fahrassistenzsysteme, welche die Objektliste (p) als Eingabe verwenden, werden im Schaubild unter ADAS zu einer einzelnen Komponente zusammengefasst. Diese implementieren die Logik der Assistenzfunktionen und steuern die verwendeten Aktoren an. Spurhalte-/Spurwechselassistent, Notbremsassistent, Abstandsregeltempomat sind Anwendungsfälle für konventionelle Assistenzfunktionen die auf Basis einer solchen Architektur implementiert werden können. Attribute der Nachrichten *m* und *p* sind in den folgenden Tabellen 3.1, 3.2, 3.3, 3.4 aufgelistet. Diese dienen im folgenden Abschnitt als Referenz für die Identifizierung von Angriffsvektoren im *perception-layer*.

Bezeichner	Typ	Beschreibung
mId	Sequenznummer	Identifikator der Messung eines Sensors
$\mathrm{mT}_{\mathrm{begin}}$	Zeitstempel	Zeitpunkt bei Begin der Messung
$\mathrm{mT}_{\mathrm{end}}$	Zeitstempel	Zeitpunkt bei Ende der Messung
mL	Liste aus *mP* (siehe 3.2)	Liste der Punkte
$\mathrm{sFov}_{\mathrm{v}}$	Radiant	Vertikaler Öffnungswinkel
$\mathrm{sFov}_{\mathrm{h}}$	Radiant	Horizontaler Öffnungswinkel
$\mathrm{sRes}_{\mathrm{v}}$	Radiant	Vertikale Auflösung
$\mathrm{sRes}_{\mathrm{h}}$	Radiant	Horizontale Auflösung
sPos	Kartesisches Tripel	Einbaulage des Sensors
sRot	Radiant Tripel	Ausrichtung des Sensors
sId	Identifikator	Identifikator des Sensors

Tabelle 3.1: Attribute der *m*-Nachricht

Bezeichner	Typ	Beschreibung
pPos	Kartesisches Tripel	Koordinaten des Punktes
pOff	Zeitstempel Offset	Zeitpunkt innerhalb $\mathrm{mT}_{\mathrm{begin}}$ und $\mathrm{mT}_{\mathrm{end}}$ (siehe 3.1)
pDist	Distanz zum Sensor	Distanz zum Sensor Ursprung

Tabelle 3.2: Attribute von *mP* (siehe 3.1)

3.1.1 Angriffsvektoren

Die von Changalvala und Malik identifizierten Angriffsvektoren FOI (Fake Object Insertion) und TOD (Target Object Deletion), durch Manipulation der Punktwolke *m*, [7] sind auf den *perception-channel* übertragbar und entsprechen dem Löschen eines Objektes aus *oList* in *p*. Da es sich bei *p*, im Gegensatz zu *m*, um bereits prozessierte Daten mit wesentlich höherem Informationsgehalt handelt, sind auf *perception-layer* deutlich komplexere Angriffe möglich. Ein zusätzlicher Angriffsvektor ist die Manipulation objekteigener Attribute (OM, *Object Manipulation*). Für die Identifizierung der Anforderungen an Intrusion Detection (ID) Mechanismen wird im folgenden Abschnitt eine Analyse der Bedrohung vorgenommen.

Bezeichner	Typ	Beschreibung
mId	Sequenznummer	Identifikator der Messung eines Sensors
oId	Identifikator	Identifikator des Objektes
oList	Liste aus o siehe 3.4)	Objektliste

Tabelle 3.3: Attribute der p Nachricht

Bezeichner	Typ	Beschreibung
oId	Identifikator	Identifikator des Objektes
oPos	Kartesisches Tripel	Koordinaten des Objekts
oVel	Geschwindigkeitsvektor	Geschwindigkeit des Objekts
oType	Kategorie	Klassifikation (LKW, PKW, etc.)
oDim	Kartesisches Tripel	Bounding Box
oRot	Radiant	Ausrichtung des Objektes
oAge	Alter	Dauer des bisherigen Trackings
tMeas	Zeitstempel	Zeitpunkt der letzten Messung

Tabelle 3.4: Attribute von o

3.2 Bedrohung

Für die Ableitung der Anforderungen und deren Priorisierung muss die Bedrohung durch die in Abschnitt 3.1.1 identifizierten Angriffsvektoren analysiert werden. Hierfür dient ein von Henniger et al [23] beschriebener Prozess für die Analyse von Sicherheitsanforderungen an ein IVN. Im Folgenden wird eine Auswahl an Angriffen hinsichtlich folgender Aspekte (nach Henniger et al) bewertet: Schaden S (Tabelle 3.6), Kontrollierbarkeit K (Tabelle 3.5), Angriffswahrscheinlichkeit P (Tabelle 3.7). Die Angriffswahrscheinlichkeit wird durch eine Metrik beschrieben, die sich aus dem Angriffspotential erschließt. Dieses wird durch eine Punktwertung verschiedener Aspekte eines Angriffs bestimmt. Zu diesen Aspekten zählen der benötigte Zeitaufwand um den Angriff durchzuführen, Expertise und Systemkenntnisse des Angreifers, das Zeitfenster für die Möglichkeit eines Angriffs und die benötigte Ausstattung. Bei der Bewertung muss berücksichtigt werden, dass es sich bei Angriffen auf das *perception-layer* um Folgeangriffe handelt, und ein Angreifer durch das Ausnutzen anderer Schwachstellen bereits in das IVN oder sogar in die LiDAR ECU eingedrungen ist. Aus diesem Grund wird die Ausgangslage für Angriffe, hinsichtlich des Angriffspotentials, wie folgt bewertet und fließt in die Auswertung mit ein.

Zeitaufwand: 3 Monate (10), Expertise: Geübt (3), Systemkenntnis: Begrenzt (3)

Die Abschätzung des Schades eines Angriffes erfolgt im Zusammenhang mit den, in Abschnitt 3.1 genannten, ADAS Funktionen, die auf Basis einer solchen Architektur realisierbar sind. Da Assistenzfunktionen, wie Spurhalteassistent oder Abstandsregeltempomat, in weiterentwickelter Form auch Teil des Funktionsumfangs eines autonomen Fahrzeugs sein können, werden für manche Angriffe zusätzlich die Auswirkungen in diesem Anwendungsfall betrachtet.

K	**Bedeutung**
1	Es ist mit einer normalen menschlichen Reaktion möglich einen Unfall zu verhindern
2	Das Verhindern eines Unfalls ist schwer, aber mit einer richtigen Reaktion möglich.
3	Das Verhindern eines Unfalls ist sehr schwer, aber unter guten Umständen oder für erfahrene Menschen möglich
4	Die Situation kann von einem Menschen nicht beeinflusst werden

Tabelle 3.5: Kontrollierbarkeitsklassen [23]

Tabelle 3.6: Schadensklassen [23]

S	**Beschreibung**
0	Keine Verletzungen
1	Leichte bis mittlere Verletzungen
2	Schwere Verletzungen
3	Lebensgefährliche Verletzungen oder schwere Verletzungen mehrerer Verkehrsteilnehmer
4	Lebensgefährliche Verletzungen mehrerer Verkehrsteilnehmer

Tabelle 3.7: Relation der Angriffswahrscheinlichkeit P zum Angriffspotential A [23]

A	**Beschreibung**	**P**
0-9	Klein	5
10-13	Klein bis Mittel	4
14-19	Mittel	3
20-24	Hoch	2
≥ 25	Sehr Hoch	1

3.2.1 Angriffe

A1 - Auslösung der Notbremse im Falle einer Kollision verhindern

Durch das Löschen eines Objektes aus der Objektliste (Attribut *oList* in Nachricht p) kann das Auslösen der Notbremse durch den Notbremsassistent verhindert werden. Hierfür darf sich das Zielobjekt, zum Zeitpunkt des regulären Auslösens des Notbremsassistenten, nicht in der Objektliste befinden. Es muss also identifiziert und, entweder zu einem exakten Zeitpunkt oder dauerhaft, aus den Nachrichten gelöscht werden. Dies setzt voraus, dass der Angreifer Kenntnis über das Anwendungsprotokoll und die Struktur der

Nachrichten hat. Dies sind zwar keine öffentlichen Informationen, können aber durch *Reverse Engineering* des Netzwerkverkehrs oder durch Einsicht in die Spezifikation der LiDAR ECU, welche beispielsweise durch einen Insider oder *Social-Engineering* Angriff erlangt werden kann, gewonnen werden (Expertise 3, Systemkenntnis 7). Der Notbremsassistent wurde hauptsächlich dafür entwickelt, um Auffahrunfälle zu verhindern und ist nur bis zu einer gewissen Geschwindigkeit aktiv, da ein starkes Bremsmanöver bei hoher Geschwindigkeit wiederum selbst ein Sicherheitsrisiko für andere Verkehrsteilnehmer darstellen kann. Moderne Systeme reagieren jedoch auch auf die bevorstehende Kollision mit nicht motorisierten Verkehrsteilnehmern, in welchem Fall der Personenschaden deutlich höher ausfällt. In diesem Fall ist mit schweren bis lebensgefährlichen Verletzungen zu rechnen (S3). Da es sich hierbei um Denial of Service (DoS) einer Sicherheitsfunktion handelt, welche nur im Falle einer Gefahrensituation aktiv wird, können Menschen durch eine normale Reaktion einen Unfall verhindern. (K1)

Schaden: 3
Kontrollierbarkeit: 1
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 ⇒ 20

A2 - Kollision durch Verstecken eines Objektes herbeiführen

Dieser Angriff gleicht auf technischer Ebene dem Angriff A1. Durch das Löschen des Objektes aus der Objektliste wird ein Verkehrsteilnehmer versteckt. Ziel des Angreifers ist hier jedoch das Herbeiführen einer Kollision eines Fahrzeugs im autonomen Betrieb. Während das Angriffspotential das gleiche ist, ist hier jedoch von einem höheren möglichen Schaden und einer geringeren Kontrollierbarkeit auszugehen. Dieser Angriff basiert im Gegensatz zu A1 nicht auf DoS einer Sicherheitsfunktion sondern stellt einen Eingriff in den normalen Betrieb dar. Ein Schaden entsteht also nicht nur im Falle eines bestimmten Szenarios wie bei A1, sondern er wird aktiv herbeigeführt. Auch die Kontrollierbarkeit ist bei diesem Angriff geringer, da sich der Fahrer normalerweise auf Kernfunktionen des Betriebs, im Gegensatz zu Notfallmechanismen, verlässt.

Schaden: 4
Kontrollierbarkeit: 2
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 ⇒ 20

A3 - Abschaltung des Abstandsregeltempomaten herbeiführen

Durch einmaliges Löschen des Zielobjektes kann eine Abschaltung des Abstandsregeltempomaten herbeigeführt werden. Dies gleicht dem Abschalten des Abstandsregeltempomaten im normalen Betrieb, beispielsweise durch manuellen Eingriff oder durch das Verlassen der Fahrbahn des vorausfahrenden Fahrzeugs. In diesem Fall wird der Fahrer über das Abschalten des Assistenzsystems informiert. Da es sich hier um eine Komfortfunktion handelt, ist kein Schaden durch das Abschalten der Assistenzfunktion zu erwarten, und der Fahrer behält jederzeit die Kontrolle über das Fahrzeug.

Schaden: 0
Kontrollierbarkeit: 1
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 $\Rightarrow$ 20

A4 - Abschaltung des Spurhalteassistenten herbeiführen

Durch das Löschen des Fahrspur-Objektes kann eine Abschaltung des Spurhalteassistenten herbeigeführt werden. Auch dies gleicht dem Abschalten des Spurhalteassistenten im normalen Betrieb, beispielsweise durch manuelles Übersteuern oder durch das Nichtvorhandensein einer Fahrspurmarkierung. Auch in diesem Fall wird der Fahrer über das Abschalten des Assistenzsystems informiert. Spurhalteassistenten als Komfortfunktion erkennen, ob der Fahrer die nötigen Lenkradbewegungen durchführt, und weisen ihn gegebenenfalls darauf hin diese durchzuführen. Der Fahrer ist in diesem Fall immer noch aktiv an der Lenkung beteiligt. Also selbst in einem unerwarteten Moment der Abschaltung kann dieser mit einer normalen Reaktion einen Unfall verhindern.

Schaden: 0
Kontrollierbarkeit: 1
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 $\Rightarrow$ 20

A5 - Auslösung der Notbremse durch Einfügen eines Objektes vor dem Fahrzeug

Durch das Einfügen eines Objektes in die Objektliste (Attribut *oList* in Nachricht p) kann das Auslösen der Notbremse durch den Notbremsassistenten herbeigeführt werden. Das Auslösen der Notbremse in einem unerwarteten Moment ist eine Gefahrensituation, jedoch kann ein Mensch durch eine angemessene Reaktion einen Unfall verhindern.

Schaden: 2
Kontrollierbarkeit: 2
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 $\Rightarrow$ 20

A6 - Einleitung eines Ausweichmanövers durch Einfügen eines Objektes vor dem Fahrzeug

Dieser Angriff gleicht auf technischer Ebene A5, jedoch hier mit dem Ziel, ein Manöver eines autonom betriebenen Fahrzeugs herbeizuführen. In Zukunft könnte es der Fall sein, dass autonome Fahrzeuge im Falle einer bevorstehenden Kollision Ausweichmanöver einleiten. Der Angreifer versucht hier ein derartiges Manöver zu provozieren. Hier gilt wiederum, dass sich der Fahrer, im Gegensatz zu Notfallmechanismen, auf den autonomen Normalbetrieb verlässt und weniger wachsam ist. Unter guten Umständen kann ein Fahrer jedoch auch hier mit einer angemessenen Reaktion einen Unfall verhindern.

Schaden: 3
Kontrollierbarkeit: 2
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 $\Rightarrow$ 20

A7 - Richtungsänderung durch Einfügen einer höher priorisierten Fahrbahnmarkierung herbeiführen

Durch das Einfügen eines zweiten Fahrspur-Objektes mit höherer Priorität kann eine Richtungsänderung durch den Spurhalteassistenten herbeigeführt werden. Auch hier gilt aus den selben Gründen wie bei A4, dass der Fahrer mit einer angemessenen Reaktion einen Unfall verhindern kann. Da bei diesem Angriff ein vollständig neues Objekt erzeugt

werden muss, ist die Bewertung der nötigen Expertise jedoch höher als bei der Änderung eines einzelnen Attributs.

Schaden: 4
Kontrollierbarkeit: 2
Potential: Zeitaufwand: 10, Expertise: 6, Systemkenntnis: 7 ⇒ 23

A8 - Bremsung oder Beschleunigung durch Geschwindigkeitsänderung des Zielobjektes herbeiführen

Durch die Manipulation von $oVel$ in o kann eine Beschleunigung oder Abbremsung durch den Abstandsregeltempomat herbeigeführt werden. Da eine Beschleunigung vom Menschen schwerer festzustellen ist als eine Richtungsänderung, ist die Kontrollierbarkeit ist in diesem Fall mit 3 eingestuft.

Schaden: 4
Kontrollierbarkeit: 3
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 ⇒ 23

A9 - Ausweichmanöver durch Änderung des Bewegungsvektors eines Objektes herbeiführen

Durch die Änderung des Bewegungsvektors $oVel$ eines vorhanden Objektes wird eine bevorstehende Kollision vorgetäuscht und ein Ausweichmanöver des autonom betriebenen Fahrzeugs herbeigeführt. Dieser Angriff bezieht sich auf autonom betriebene Fahrzeuge, Ausweichmanöver sind aktuell nicht Bestandteil des Funktionsumfangs von ADAS. Aus diesem Grund ist die Kontrollierbarkeit hier mit 4 eingestuft.

Schaden: 4
Kontrollierbarkeit: 4
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 ⇒ 20

A10 - Richtungsänderung durch Änderung des Verlaufes der Fahrbahnmarkierung herbeiführen

Durch die Manipulation des Fahrspur-Objektes kann eine Richtungsänderung durch den Spurhalteassistenten herbeigeführt werden. Dieser Angriff gleicht in den Schaden- und Kontrollierbarkeitsklassen A7, mit dem Unterschied, dass hier eine Manipulation von *oPos* einer vorhandenen Fahrspur stattfindet.

Schaden: 4
Kontrollierbarkeit: 3
Potential: Zeitaufwand: 10, Expertise: 6, Systemkenntnis: 7 ⇒ 23

A11 - Folgen der falschen Fahrspur durch Änderung der Priorität

Durch die Manipulation des Typs eines Fahrspur-Objektes wird der falschen Fahrspur gefolgt. Dieser Angriff hat die selben Auswirkungen wie bei A10, jedoch wird hier, anstatt einer Einfügung, das *oType* Attribut einer vorhandenen Fahrspur manipuliert um eine höhere Priorisierung zu erzwingen. Die Priorisierung der Fahrspuren ist notwendig für den Fall des Vorhandenseins mehrerer Fahrspurmarkierungen wie beispielsweise bei Baustellen.

Schaden: 4
Kontrollierbarkeit: 3
Potential: Zeitaufwand: 10, Expertise: 3, Systemkenntnis: 7 ⇒ 23

3.2.2 Risikobewertung

In der folgenden Tabelle 3.8 sind die Angriffe und das Risikolevel gemäß des von Henninger et al vorgschlagenen Schlüssels aufgelistet. Zusätzlich sind die relevanten Attribute der Objekte identifiziert worden, welche später als Grundlage für die Anforderungen an ein IDS herangezogen werden.

Angriff	Vektor	S	K	A	P	Risikolevel	Vorgehen
A1	TOD	3	1	20	2	2	Löschen aus oList
A2	TOD	4	2	20	2	4	Löschen aus oList
A3	TOD	0	1	20	2	-	Löschen aus oList
A4	TOD	0	1	20	2	-	Löschen aus oList
A5	FOI	2	2	20	2	2	Einfügen in oList
A6	FOI	3	2	20	2	4	Einfügen in oList
A7	FOI	4	2	23	2	5	Einfügen in oList
A8	OM	4	3	23	2	5	Änderung von oVel
A9	OM	4	4	20	2	5	Änderung von oVel
A10	OM	4	3	23	2	5	Änderung von oPos
A11	OM	4	3	23	2	5	Änderung von oType

Tabelle 3.8: Risikobewertung der Angriffe

3.2.3 Anforderungen an IDS

Aus der Risikobewertung der Angriffe in Abschnitt 3.2.1 ergeben sich folgende Anforderungen an ein IDS mit hoher Priorität, basierend auf einem Risikolevel von 4 oder höher:

#	Anforderung
R1	Erkennung des Einfügen in oList
R2	Erkennung des Löschen aus oList
R3	Erkennung der Manipulation von oVel
R4	Erkennung der Manipulation von oPos
R5	Erkennung der Manipulation von oType

Tabelle 3.9: Funktionale Anforderungen

Zusätzlich zu diesen funktionalen Anforderungen muss der Entwurf eines IDS folgende Aspekte berücksichtigen:

- **Ressourcenverbrauch**: Interne Zustände sollten so weit wie möglich vermieden werden, um Ressourcenverbrauch zu minimieren.
- **Antwortzeit**: Damit ein Unfall vermieden werden kann, muss das IDS Angriffe erkennen, bevor das System die Aktoren ansteuert.
- **Aufrechterhaltung essentieller Funktionen**: Um einen Unfall zu verhindern, müssen essentielle Funktionen aufrecht erhalten werden. Präventionsmechanismen

dürfen nicht zu einem Ausfall dieser Funktionen führen. Dies ist insbesondere im Falle von autonom betriebenen Fahrzeugen relevant, da Kernfunktionalitäten des autonomen Betriebs direkt an die maschinelle Wahrnehmung gekoppelt sind.

4 Lösungsansätze

In diesem Kapitel werden mögliche Lösungsansätze vorgestellt. Dabei wird zwischen zwei verschiedenen Kategorien von Ansätzen unterschieden, wobei die Kategorisierung auf den dem IDS zur Verfügung stehenden Daten basiert. Der Fokus dieser Arbeit liegt auf der LiDAR Domäne. Es wird zwischen Ansätzen die rein auf der Anwendungsschnittstelle (4.1) arbeiten und Ansätzen welche die Rohdaten der Sensoren berücksichtigen (4.2) unterschieden.

4.1 Anwendungsschnittstelle

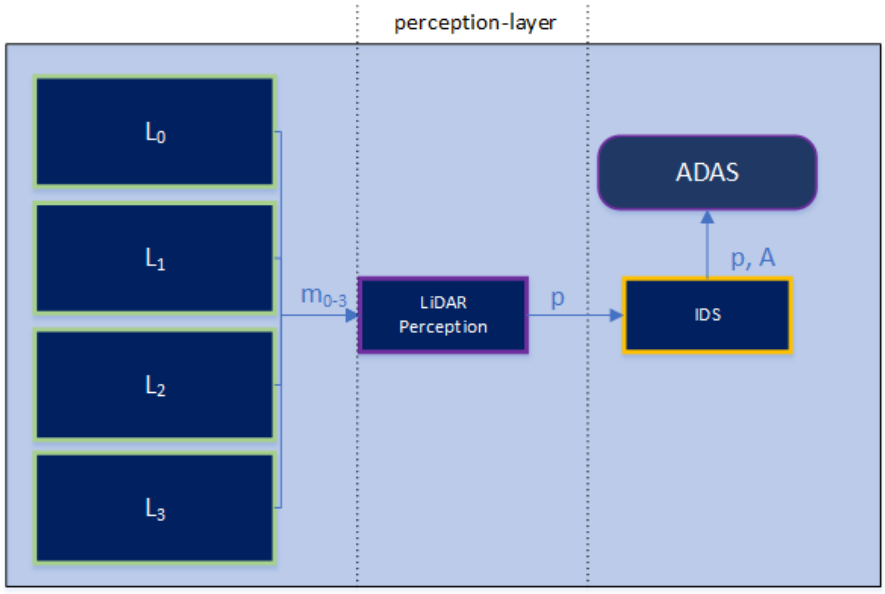

Abbildung 4.1: IDS basierend auf der LiDAR Perception Anwendungsschnittstelle

Basierend auf Nachricht p können verschiedene Anomalieerkennungs-Probleme definiert werden. Hierfür werden die Attribute der Eingangsdaten, gemäß der im Kapitel 2.4 erläuterten Methode, untersucht um gezielt Problemstellungen zu formulieren, welche die in Abschnitt 3.2 identifizierten Angriffe abbilden. In den Tabellen 4.1 und 4.2 sind die Attribute der Nachrichten und deren Art der Relation aufgelistet.

Attribut	**Typ**	**Relation**
mId	contextual	sequential
oList	siehe Tabelle 4.2	-

Tabelle 4.1: Typen und Relationen der Attribute von p

Attribut	**Typ**	**Relation**
oId	behavioural	-
oPos	contextual	spatial
oVel	contextual	spatio-temporal
oType	behavioural	
oDim	behavioural	-
oRot	behavioural	-
oAge	contextual	temporal
tMeas	contextual	temporal

Tabelle 4.2: Typen und Relationen der Attribute von o

Nun gilt es den Angriff als Abweichung eines Erwartungswertes, i.e. des Normalverhaltens des Systems, zu definieren. Zur Modellierung des Normalverhaltens können folgende Erwartungen an das System gestellt werden:

1. Syntaktisch korrekte Nachrichten
2. Attribute sind gemäß ihren Relationen konsistent
3. Einhaltung von Naturgesetzen

Da es sich bei den in dieser Arbeit betrachteten Angriffen um die syntaktisch korrekte Manipulation der Nachrichten handelt, sind die korrespondierenden Anomalieerkennungs-Probleme großteils kontextueller Natur. Beispielsweise kann bezüglich des Angriffes *A1 - Auslösung der Notbremse im Falle einer Kollision verhindern* durch Löschen eines Objektes *oList*, folgende Erwartung an das Normalverhalten formuliert werden:

Das Verschwinden eines Objektes aus oList ist nur dann zu erwarten, wenn das Objekt das FoV der Sensoren verlässt oder durch ein anderes Objekt verdeckt wird.

Dieser Zusammenhang kann nun mit den Attributen *oPos* und *oVel* modelliert werden und ist ein Beispiel für eine Kontextanomalie basierend auf Erwartung 3. Zusätzlich können basierend auf den identifizierten Relationen der Attribute (Tabelle 4.2) basierend auf Erwartung 2 weitere Regeln formuliert werden. Beispielsweise kann für Angriff *A9 - Ausweichmanöver durch Änderung des Bewegungsvektor eines Objektes herbeiführen* der Zusammenhang der Attribute *oPos* und *oVel* modelliert werden, um Manipulationen der Positions- und Bewegungsvektoren zu erkennen. Dieser Ansatz entspricht einem Regelbasierten IDS unter Verwendung von physikalischen Modellen. Es muss sich jedoch nicht jede Regel auf den physischen Raum beziehen, andere modellierbare Relationen zwischen den Instanzen wären die der Attribute *mId* und *tMeas* (Sequenzen). Wichtig ist bei diesem Ansatz, dass Regeln Angriffe abdecken, ohne dass unerwartete Situationen, die keine Angriffe sind, ein *false-positive* produzieren, weshalb statistische Methoden bei der Integritätsprüfung der Objektliste weniger geeignet sind.

Ein IDS, das diesen Ansatz verfolgt, würde es einem Angreifer erheblich erschweren, unbemerkt einen Angriff durchzuführen, bietet jedoch keinen vollständigen Schutz und wäre ohne zusätzliche Mechanismen unzureichend. Ein Angreifer, mit Expertenwissen über das System, wäre theoretisch in der Lage Nachrichten zu erzeugen, die ein Szenario darstellen, das mit den Modellen des IDS übereinstimmen. [26]

4.2 Punktwolke

Die im vorherigen Abschnitt formulierten Erwartungen basieren auf a priori Wissen, ohne direkten Bezug zur derzeitigen Situation des cyber-physikalischen Systems. Durch die Berücksichtigung der Sensor-Rohdaten fließt ein direkter Erfahrungswert mit ein, der es dem IDS erlaubt, die Limitationen des vorherigen Ansatzes zu umgehen. Dies erlaubt es Angriffe zu erkennen, welche konsistent mit Modellen sind, die auf a priori Wissen basieren. Hier können Erwartungen in beide Richtungen der Verarbeitungskette formuliert werden. Erwartungen an die Applikationsschnittstelle, basierend auf Sensor-Rohdaten, entsprichen einem redundanten Pfad des Informationsflusses, in welchem das IDS die gleiche Problemstellung (Objekterkennung) löst und Ergebnisse miteinander verglichen werden. Dieser Ansatz hat den Vorteil, dass die Prozessierung der Rohdaten durch das IDS parallel zur LiDAR ECU erfolgen kann.

Eine weitere Möglichkeit ist es, basierend auf der Liste der erkannten Objekte, einen Rückschluss auf die Sensordaten zu ziehen. Hierfür kann die Relation zwischen den Mess-

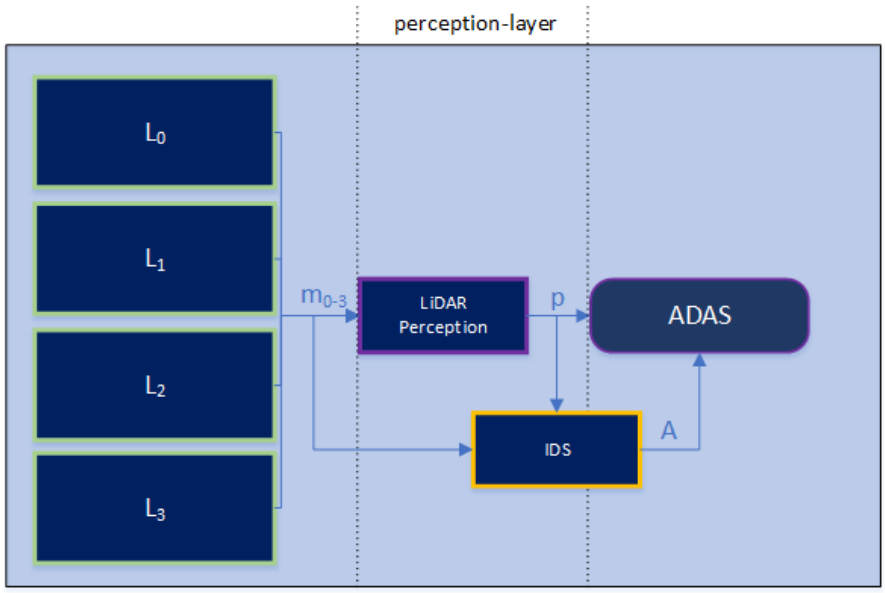

Abbildung 4.2: IDS mit Punktwolke als zusätzliche Eingabe

punkten, unter Berücksichtigung der Sensorkonfiguration (sFov, sRes, sPos und sRot), und der Liste der erkannten Objekte modelliert werden, um Vorhersagen über die Sensordaten zu treffen. Ein Angriff wäre dann als fehlende Übereinstimmung zwischen den gemessenen Punkten und der Vorhersage zu erkennen.

Attribut	**Typ**	**Relation**
mId	contextual	sequential
mT_{begin}	contextual	temporal
mT_{end}	contextual	temporal
mL	behavioural	-
$sFov_v$	behavioural	-
$sFov_h$	behavioural	-
$sRes_v$	behavioural	-
$sRes_h$	behavioural	-
sPos	behavioural	-
sRot	behavioural	-
sId	behavioural	-

Tabelle 4.3: Typen und Relationen der Attribute von m

In den Tabellen 4.3 und 4.4 sind die Typen der Attribute der Objektliste und deren Art der Relation aufgelistet. Da die Sensorkonfiguration keine direkte Relation zu anderen Instanzen beschreibt, ist diese an dieser Stelle als *behavioural* klassifiziert. Der Zusam-

Attribut	Typ	Relation
pPos	behavioural	-
pOff	contextual	temporal
pDist	behavioural	-

Tabelle 4.4: Typen und Relationen der Attribute von *mP*

menhang von Sensorkonfiguration und Messpunkten (*mL*) ist nur durch die Liste der erkannten Objekte gegeben. Während eine exakte Vorhersage der gemessenen Punkte im Rückschluss nicht möglich ist, können Metriken definiert werden (wie die Anzahl der gemessenen Punkte oder deren Dichte) welche die Messung eines Objektes charakterisieren. Auch Hybride aus den hier beschriebenen Lösungsansätzen wären möglich. So muss sich die Vorhersage nicht nur auf die aktuellen Messungen beziehen, sondern es können durch den Einsatz eines physikalischen Modells auch Vorhersagen über zukünftige Messungen getätigt werden.

4.2.1 Reverse-Measurement-Model

Die Modellierung der Messung basierend auf den erkannten Objekten wird als *Reverse-Measurement-Model* bezeichnet. Dieses Prinzip entstammt der Problemstellung des *Obstacle Mapping* [17]. Eine einfache, aber fehlerbehaftete, Variante eines Modells dieser Art ist die Vorhersage der Anzahl der erwarteten Messpunkte eines Objektes basierend auf dem Öffnungswinkel, der sich aus den Polarkoordinaten der Bounding-Box ergibt. In Abbildung 4.3 ist dieses Prinzip vereinfacht dargestellt. Durch die Berechnung des Winkels kann anhand der Auflösung des Sensors eine Vorhersage über die Anzahl der zu erwarteten Punkte gemacht werden. Dieses Modell ist fehlerbehaftet, da sich die Geometrien der Objekte unterscheiden. Außerdem ist die Streuung der Messpunkte auf der Oberfläche abhängig von der Distanz zum Sensor, die Vorhersage durch den Öffnungswinkel ist lediglich eine Approximation. Um die Streuung zu berücksichtigen, kann die Bounding Box auf das FoV des Sensors projiziert werden und mittels der Punktdichte eine genauere Schätzung der zu erwartenden Messpunkte erfolgen.

Ohne weitere Mechanismen ist die Anzahl der Messpunkte in einem Objekt nicht hinreichend, um alle Manipulationen zu erkennen, da Objekte in einer Art und Weise eingefügt oder manipuliert werden können, so dass weiterhin genügend Messpunkte vorhanden sind. Beispielsweise durch das Einfügen eines Objektes direkt vor dem Fahrzeug, so dass es die Bodenebene dort schneidet, wo Sensoren Punkte auf der Fahrbahn messen.

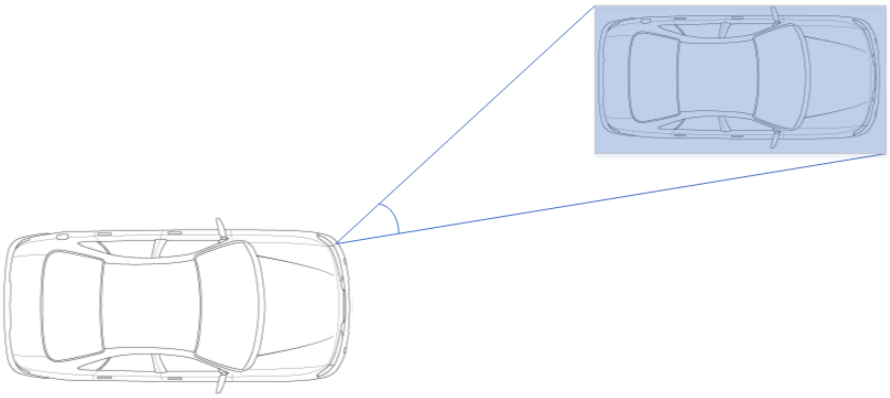

Abbildung 4.3: Öffnungswinkel der Bounding-Box mit Ursprung im Sensor

4.2.2 Objekt-Modell

Um das *Reverse-Measurement-Model* zu verfeinern und die genannten Nachteile zu umgehen, können die berücksichtigten Punkte durch ein *Objekt-Model* eingeschränkt werden. Das *Objekt-Modell* beschreibt die Geometrie des Objektes. Je nach Objekt Typ (*oType*) können zur Laufzeit verschiedene Modelle für PKWs, LKWs oder Zweiräder verwendet werden, welche die Charakteristiken der Fahrzeuge abbilden. So können geringere Rotationen oder Translationen der Objekte erkannt werden, obwohl sich ein manipuliertes Objekt mit echten Messpunkten überschneidet.

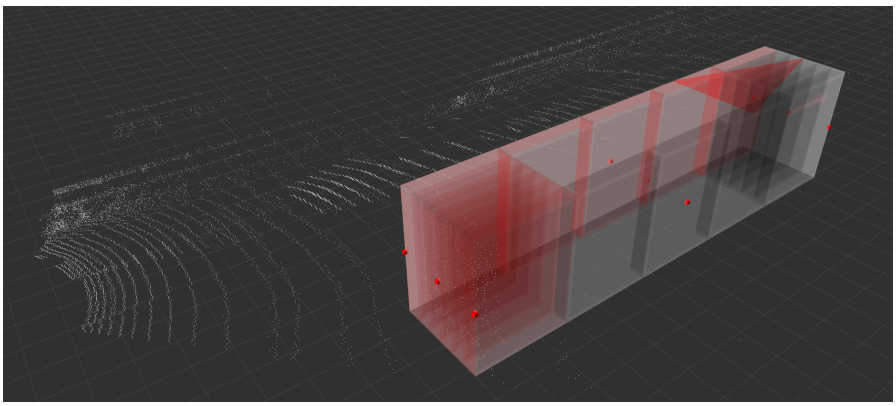

Abbildung 4.4: An einem LKW visualisiertes Object-Model

In Abbildung 4.4 ist ein einfaches Objekt-Modell visualisiert. Das Objekt wurde in mehrere Quader unterteilt. Nur Messpunkte in den äußeren Schichten werden berücksichtigt,

da die meisten Messpunkte eines Objektes an der Außenhülle des Fahrzeuges lokalisiert sind. Weiterhin kann durch den Richtungswinkel berechnet werden, welche Teile der Geometrie durch das Objekt selbst verdeckt werden. Auch das Verdecken durch andere Objekte in der Objekteliste kann in das geometrische Objekt-Modell mit einbezogen werden. [18]

4.2.3 Freiraum-Erkennung

Eine Limitierung eines auf Rückschluss basierten Ansatzen ist es, dass keine gelöschten Objekte erkannt werden können. Das Erkennen gelöschter Objekte lässt sich im Umkehrschluss als die Erkennung von fehlendem Freiraum formulieren. Wird in einer Region of Interest (ROI) kein Freiraum erkannt, wo sich in der Objektliste keine Objekte befinden, handelt es sich hierbei mit hoher Wahrscheinlichkeit um ein gelöschtes Objekt. Dieser Ansatz impliziert die Anforderung an die Objekterkennung, dass alle Objekte in der ROI erkannt und als Objekt markiert werden. Ein von Pomerleau et al. [34] [55] verwendetes Konzept für die LiDAR basierte Freiraum-Erkennung ist die Unterteilung der ROI in ein dreidimensionales Gitter. Basierend auf der Strahlverfolgung werden die Messpunkte in den sogenannten Bins des Gitters lokalisiert. Wird ein vorgegebener Schwellwert an Messpunkten überstiegen gilt der Bin als belegt. Überschneidet sich der Bin nicht mit einem Objekt, wird davon ausgegangen, dass es sich hierbei um ein gelöschtes Objekt handelt. Da für die vorherigen Ansätze die Assoziation der Messpunkte mit den Objekten hergestellt wird, können alternativ nur die Punkte gezählt werden, die mit keinem Objekt in Verbindung stehen.

4.2.4 Scoring

In den Grundlagen wurde zwischen *scoring* und *labeling* Methoden bei der Art der Ausgabe einer Anomalieerkennungs-Komponente unterschieden. Um Komponenten welche die Objektliste weiterverarbeiten, die Möglichkeit zu geben, essentielle Funktionen im Falle eines Angriffes aufrechtzuerhalten, sollte möglichst jedes Objekt einzeln bewertet werden. Hier eignet sich ein *scoring* Mechanismus, welcher die Konfidenz einer Manipulation eines Objektes repräsentiert. Ein weiterer Aspekt, der bei einem Mechanismus dieser Art berücksichtigt werden muss, sind Unsicherheiten durch Mess- oder Übertragungsfehler. In verwandten Problemstellungen werden hierfür Filter verwendet, um fehlerbehaftete

Beobachtungen zu verarbeiten. Ein Beispiel aus der Robotik ist die Erstellung von *Occupancy Grids*, das Erstellen einer Karte basierend auf rauschbehafteten Messungen und der Position des Roboters [16]. Dort ist die Zufallsvariable das Existenzmaß eines Hindernisses, beziehungsweise einer belegten Zelle in der *Grid Map*. Dieses Konzept wird auf den Anwendungsfall dieser Arbeit übertragen, in der die Zufallsvariable ein Maß für den Grad an Übereinstimmung mit dem vorhergesagten Objekt darstellt. Ein geringer Grad an Übereinstimmung entspricht einer hohen Wahrscheinlichkeit für eine Anomalie.

Die bedingte Wahrscheinlichkeit einer Anomalie zum Zeitpunkt t, basierend auf den Sequenzen der Messungen $\mathrm{m_{1:t}}$ und der erkannten Objekte $\mathrm{o_{1:t}}$, lässt sich wie folgt formulieren:

$$p(A_t|m_{1:t}, o_{1:t}) \tag{4.1}$$

$$p(m_t|A, m_{1:t-1}, o_{1:t}) = p(m_t|A, o_t) \tag{4.2}$$

$$p(A|m_{1:t-1}, o_{1:t}) = p(A|m_{1:t-1}, o_{1:t-1}) \tag{4.3}$$

Durch das mehrfache Erweitern dieser Gleichung durch den Satz von Bayes und der Annahmen 4.2 (Markow-Eigenschaft) und 4.3 (Annahme der Unabhängigkeit) lässt sich ein Bayesscher Filter ableiten. Dieser wird zur Schätzung der Wahrscheinlichkeit einer Anomalie (als unbeobachteter Zustand des Systems bei gegebenen Objektlisten und Messungen) verwendet:

$$l(A|m_{1:t}, o_{1:t}) = \underbrace{l(A|m_t, o_t)}_{\text{model}} + \underbrace{l(A|m_{1:t-1}, o_{1:t-1})}_{\text{recursion}} - \underbrace{l(A)}_{\text{prior}} \tag{4.4}$$

Die in der Gleichung als *model* deklarierte Komponente kann durch Bildung des Jaccard-Koeffizienten der vorhergesagten und echten Anzahl der Messpunkte gebildet werden. Sie beschreibt den Grad an örtlicher Übereinstimmung des Objektes mit der Messung. Auch wenn für das System in der Realität keine echte Markow-Eigenschaft vorliegt, lässt sie sich für diesen speziellen Anwendungsfall verwenden, da Objektlisten unabängig voneinander betrachtet werden und die Vorhersage nur auf der letzten empfangenen Objektliste basiert. Zudem beinhalten Objektlisten eine Historie in Form von abgeleiteten

Attributen wie der Geschwindigkeit und der Beschleunigung. So sind vorangegangene Zustände des Positionsattributs Teil der aktuellen Zustandsdefinition. Die Annahme der Unabhängigkeit ist ebenso eine Vereinfachung der realen Zusammenhänge, kann jedoch in diesem Fall angewandt werden, da ohne eine zur Objektliste korrespondierende Messung keine Aussage über die Wahrscheinlichkeit einer Anomalie getroffen werden kann. Die rekursive Auswertung dient der Verringerung der internen Zustände, da keine Historie der Objektlisten nötig ist. Durch die Umformung in ein Logit können zudem teuere Fließkomma-Multiplikationen eliminiert werden.

5 Realisierung

Dieses Kapitel beschreibt die Realisierung des Prototyps. Folgende Funktionalitäten wurden implementiert:

1. Approximation der gemessenen Punkte, berechnet durch den Öffnungswinkel und die Auflösung des Sensors.
2. Vorhersage der gemessenen Punkte durch Projektion der Bounding Box und Punktdichte des Sensors
3. Berechnung der echten Messpunkte auf dem Objekt und Auswertung des geometrischen Modells
4. Ausgleichsrechnung bei der Vorhersage der Messpunkte
5. Bayes-Filter als Scoring Mechanismus für die Objekte, basierend auf der vorhergesagten Anzahl an Punkten und den echten Messpunkten
6. Freiraum-Erkennung durch die Erstellung eines 3D-Gitters und Lokalisierung der Messpunkte in den Bins

Im folgenden Abschnitt 5.1 werden die verwendeten Technologien und der vorhandene Datenbestand erläutert. Darauf folgt eine Übersicht über den Aufbau der Softwarekomponente und eine Erläuterung der Module. Im letzten Abschnitt dieses Kapitels wird auf die Verifikation der Kernfunktionen eingegangen.

5.1 Technologien und Daten

Zur Realisierung des Prototyps wird das Robot Operating System (ROS) verwendet. Dieses Framework bietet die Möglichkeit, einen ereignisgesteuerten (*event-triggered*) Prozess zu implementieren, welcher durch Nachrichtenaustausch (*message-passing*) angestoßen

werden kann. ROS bietet, zusätzlich zum Software-Framework, Werkzeuge um den Nachrichtenaustausch der Prozesse (ROS-Nodes) aufzuzeichnen und gegebenenfalls wieder in das Netzwerk einzuspeisen. In dieser Form liegen Aufzeichnungen eines Netzwerks gemäß der in Kapitel 3.1 beschrieben Architektur vor. Diese Aufzeichnungen beinhalten die Rohdaten der Sensoren in Form von Punktwolken und die Ergebnisse der Objekterkennungs-Komponente, welche während einer Autobahnfahrt aufgenommen worden sind. Zusätzlich stehen Simulationsdaten einer Autobahnfahrt zur Verfügung, welche mit einem Sensor-Modell des selben Sensors generiert worden sind. Auch dieser Datensatz wurde mit der Objekterkennungs-Komponente prozessiert und enthält entsprechende Objektlisten. Die in dieser Arbeit verwendeten Objekterkennungsalgorithmen unterstützen die Erkennung von PKWs, LKWs und Zweirädern. Für Leitplanken und Fahrspurmarkierungen sind keine Daten verfügbar.

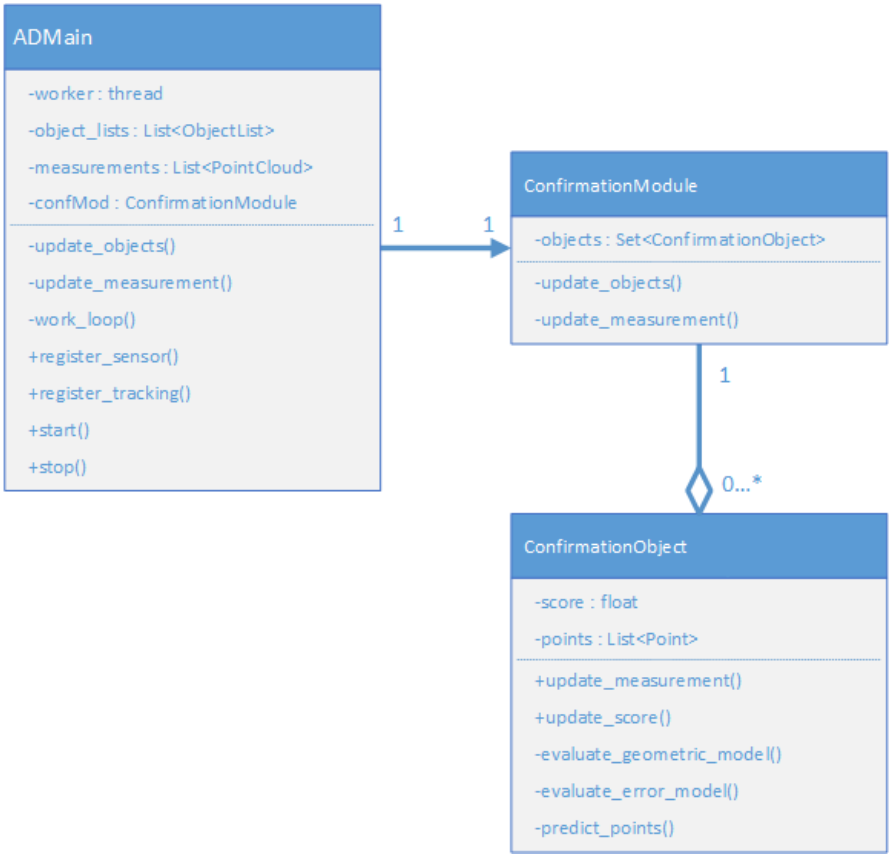

Abbildung 5.1: UML Klassendiagramm des Prototyps

5.2 Software-Komponente

Der Prototyp ist in C++ implementiert worden und besteht im wesentlichen aus drei Klassen. Die Kernfunktionalitäten der Klassen, deren Multiplizitäten und interne Zustände, sind in 5.1 abgebildet und werden im folgenden näher erläutert.

ADMain Diese Klasse dient der Verwaltung eingehender Nachrichten und dem Anstoßen der Anomalieerkennung. Sie implementiert Callbacks, welche die Events der Aktualisierung der Objektliste und der eingehenden Messungen bearbeiten. Weiterhin implementiert die Klasse das Starten und Stoppen des Verarbeitungsthreads und dessen Einstiegspunkt. Die Aktualisierungen der Objektliste werden in eine Warteschlange eingefügt und gemäß der Sequenznummer verarbeitet. Messungen werden in einer Liste verwaltet. Die Anomalieerkennung wird durch die Aktualisierung der Objektliste angestoßen. Eine Aktualisierung der Objektliste wird publiziert, wenn die vier Scan-Frames der Sensoren an der Objekterkennungskomponente eingegangen und verarbeitet worden sind. Durch die Unvorhersagbarkeit der Nachrichtenverteilung an die Prozesse durch ROS, sind hier Synchronisationsmechanismen nötig, welche in einem echten IVN nicht nötig sind. Außerdem kann es aus diesem Grund vorkommen, dass zum Zeitpunkt einer Aktualisierung der Objektliste nicht alle Messungen vorhanden sind. Die Komponente wurde so konzipiert, dass sie mit variabler Anzahl an Messungen funktioniert. Dies kann auch in einem echten Anwendungsszenario vorkommen, wenn beispielsweise ein Sensor defekt ist. Deshalb wird die Anomalieerkennung in jedem Fall angestoßen, wenn eine neue Objektliste eingeht. Der Verarbeitungsthread identifiziert alle korrespondierenden Messungen anhand der Zeitstempel und stößt das *ConfirmationModule* mit der aktuellen Objektliste und den letzten Messungen, welche in die Berechnung der Objektliste eingeflossen sind, an.

ConfirmationModule Das ConfirmationModule verwaltet die aktuelle Liste der Objekte. Im Falle einer Aktualisierung der Objektliste werden die Objekte anhand ihrer Identifier assoziiert und der aktuelle Score auf das neue Objekt übertragen. Die eingehenden Messungen werden daraufhin für jedes Objekt verarbeitet. Weiterhin dient diese Klasse zur Implementierung von Funktionen, die nicht auf Attributen erkannter Objekte basieren. Das Erstellen des Gitters und die Aktivierung der Bins für die Freiraum-Erkennung ist hier implementiert.

ConfirmationObject In dieser Klasse sind die objektbasierten Kernfunktionen realisiert. Dazu gehören die Implementationen des Objekt-Modells und des *Reverse-Measurement-Modells*, als auch des Scoring Mechanismus durch den Bayesschen Filter. Interne Zustände des ConfirmationObject sind die pro Frame kumulierten Ergebnisse aus den Berechnungen der einzelnen Messungen der verschiedenen Sensoren sowie der aktuelle Score der Anomalieerkennung.

5.3 Verifikation

Zur Verifikation der Kernfunktionen wurden, basierend auf den Simulationsdaten, verschiedene Angriffsszenarien erstellt. Hierfür wurden die Nachrichten der Objekterkennungs-Komponente in den Aufnahmen manipuliert. So konnte während der Entwicklung, durch Abspielen dieser manipulierten Aufnahmen, eine laufende Überprüfung der Funktionen stattfinden. In den Abbildungen 5.2, 5.3 und 5.4 sind die Ergebnisse dieser manuellen Verifikation abgebildet. In Abbildung 5.2 ist die Detektion eines gelöschten Autos zu erkennen. Auf der rechten Seite des Schaubildes 5.2 sind aktive Bins, an der Stelle wo sich das gelöschte Auto befindet, in rot eingefärbt. Für die Verifikation wurden Bins als Würfel mit einer Seitenlänge von 50cm eingestellt. Auf der linken Seite des Schaubildes ist das simulierte Kamerabild zu sehen, in welchem das gelöschte Auto zu sehen ist.

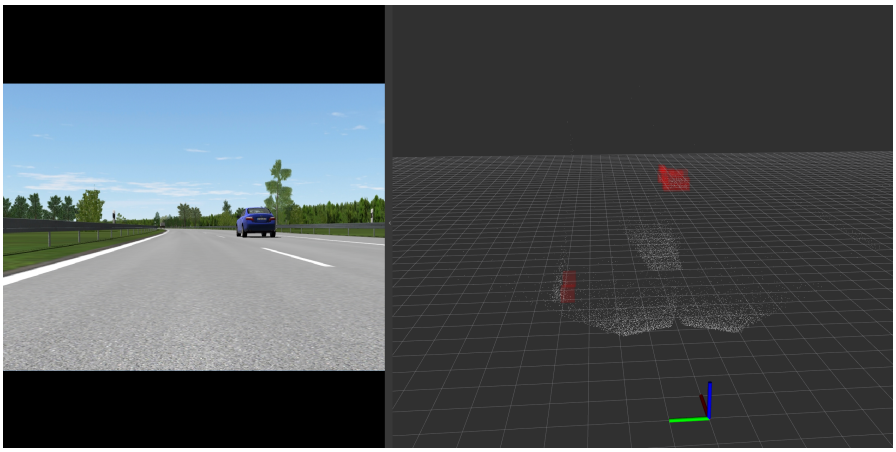

Abbildung 5.2: Erkennung eines gelöschten Objektes

Abbildung 5.3 zeigt die Bewertung eines eingefügten Objektes durch die Anomalie-Erkennung. Das Objekt ist hier mit dem Wert 0.01 annotiert. Dieser sagt aus, dass das Objekt mit der geringsten Wahrscheinlichkeit existiert.

Abbildung 5.3: Erkennung eines eingefügten Objektes

Im letzten Szenario wurde das Objekt um 90 Grad rotiert. Auch hier bewertet die Anomalieerkennung das Objekt mit einem Score von 0.01, das Objekt wurde also mit sehr hoher Konfidenz als Anomalie eingestuft.

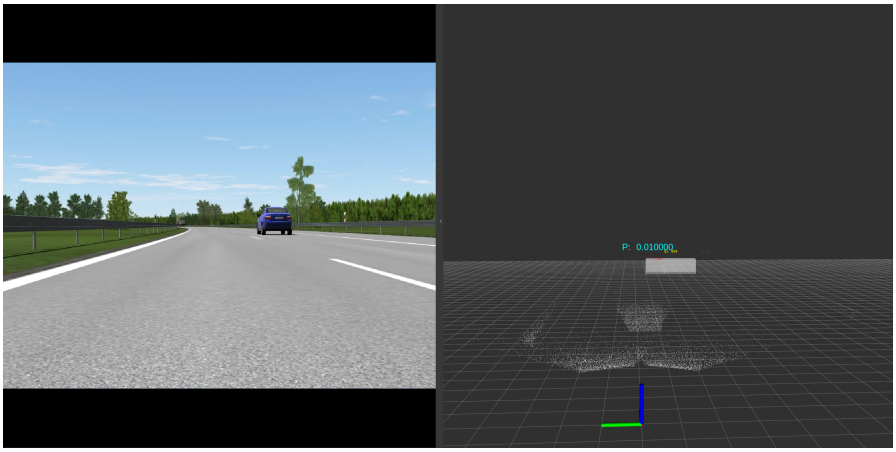

Abbildung 5.4: Erkennung eines um 90 Grad rotierten Objektes

6 Auswertung

Nachdem in der Verifikation die Funktionalität der Kernfunktionen überprüft wurde, wird anhand weiterer Szenarien die Genauigkeit der Anomalieerkennung ausgewertet. Hierfür wurden die Aufnahmen Manipuliert. Danach wurde der Prototyp gestartet und die Ausgabe Anomalieerkennung und der Objekterkennungs-Software mitgeschnitten. Daraufhin wurde die untersuchten Datenpunkte mittels Scripten extrahiert und ausgewertet. Ein weiteres Hilfsmittel ist die Visualisierung der Objekterkennung und der Rohdaten. So können Auffälligkeit und deren Zusammenhänge genauer untersucht werden.

6.1 Translation

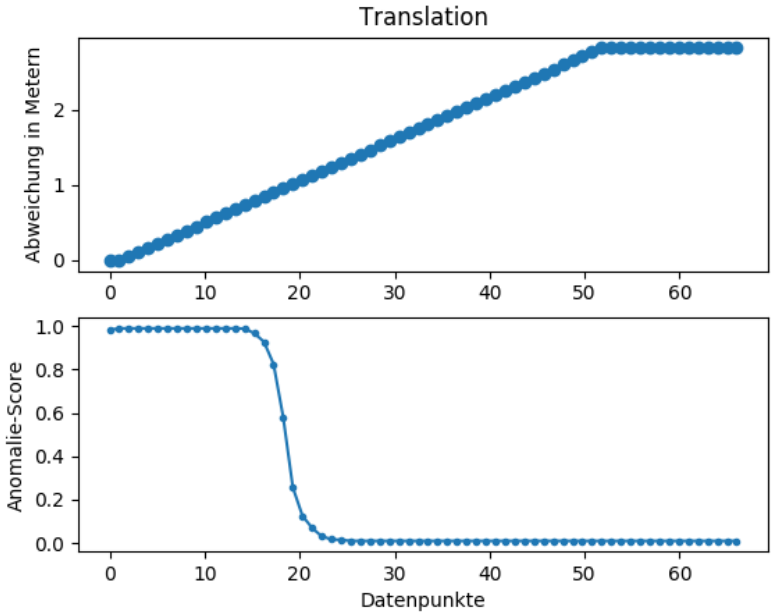

Abbildung 6.1: Translation auf Y Achse

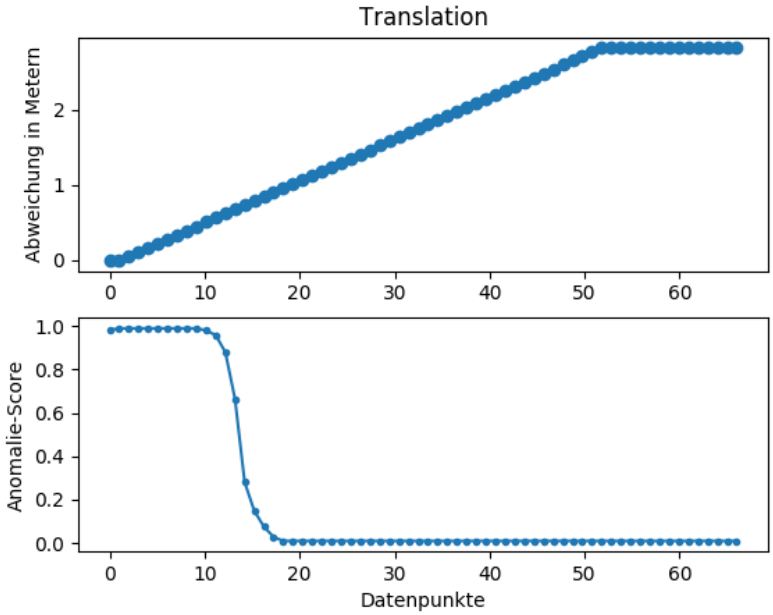

Abbildung 6.2: Translation auf X und Y Achse

Um das Verhalten der Anomalie-Score im Vergleich zum Grad der Manipulation zu untersuchen, wurde ein Objekt Frame für Frame über einen Zeitraum von 2 Sekunden um 3 Meter verschoben. Die ursprüngliche und veränderte Position des Objektes und dessen Anomalie-Score wurden mitgeschnitten und es wurden die Datenpunkte der Aufnahme extrahiert.

Die Abweichung von der echten Position und die Bewertung der Anomalie-Erkennung ist in 6.1 dargestellt. Im Schaubild ist zu erkennen, dass die Translation auf der Y-Achse erst nach einem Meter dafür sorgt, dass das Objekt als Anomalie eingestuft wird.

Der Grund dafür ist, dass sich noch ausreichend Messpunkte im Objekt befinden, wenn sich das manipulierte und das echte Objekt überschneiden. Im Vergleich ist in Abbildung 6.2 das Ergebnis einer Translation in zwei Dimensionen abgebildet. Hier ist bereits nach 50 cm eine Erkennung möglich. Eine Translation auf zwei Achsen ist leichter möglich, da das verwendete geometrische Modell in diesem Fall eine geringere Überschneidung mit den Messpunkten hat.

6.2 Rotation

Um zu überprüfen, in welchem Maße eine Rotation erkannt werden kann, wurde ein Objekt nach dem selben Prinzip über einen Zeitraum von 2 Sekunden um 180 Grad gedreht. In Schaubild 6.3 sind die Datenpunkte dieses Manövers visualisiert. Man erkennt dass eine Rotation von ungefähr 50 Grad zu einem starken Abfall des Anomalie-Score führt. Nähert sich das Objekt an eine symmetrische Rotation an, steigt der Anomalie-Wert wieder wie erwartet auf den höchsten Konfidenzwert von 1.0. Dies ist bei ungefähr 145 Grad der Fall.

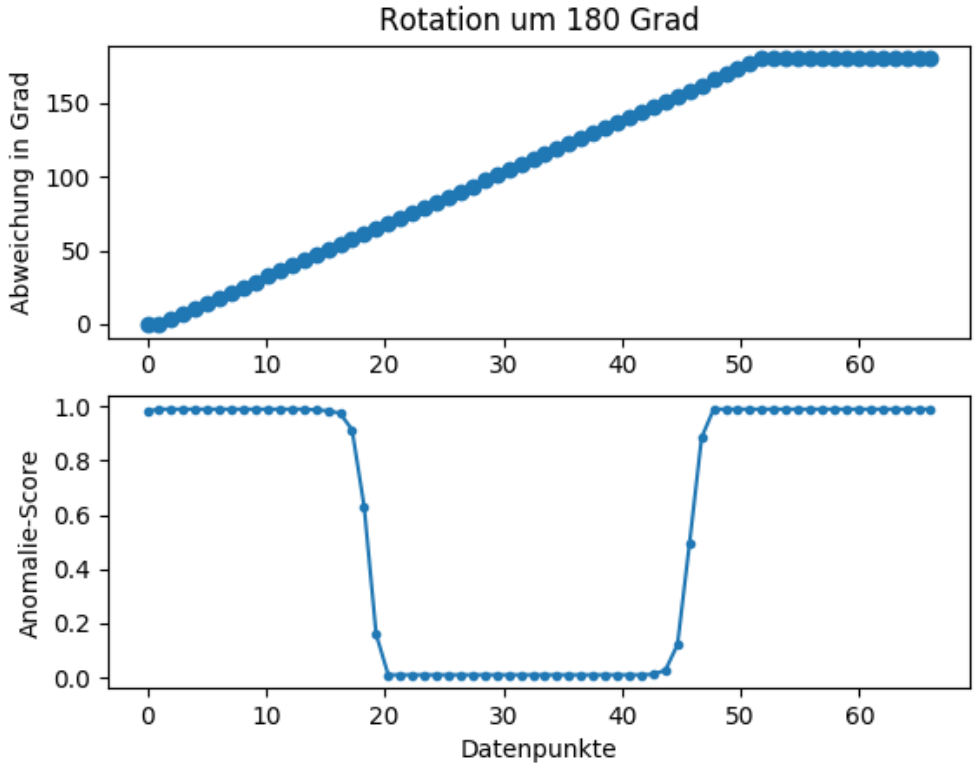

Abbildung 6.3: Verlauf einer Rotation um 180 Grad

6.3 Dimension

Weiterhin wurde untersucht, wie sich eine Änderung der Größe auswirkt. Ein Objekt wurde für diesen Zweck graduell verkleinert. Ähnlich der Translation ist auch hier erst nach einem Meter die Erkennung der Manipulation eingetreten. Da das verwendete geometrische Objekt-Modell achsensymmetrisch, aber nicht rotationssymmetrisch ist, war zu erwarten dass Rotationen leichter erkannt werden als Translationen, insbesondere wenn diese nur in eindimensionaler Richtung stattfinden. Eine Verbesserungsmöglichkeit um diesem Umstand entgegenzuwirken wäre, wenn das geometrische Modell nicht nur verwendet wird, um die gezählten Punkte einzuschränken, sondern wenn auch fehlende Messpunkte in der Geometrie als Malus berücksichtigt werden. Der Verlauf der Datenpunkte bei der Verkleinerung des Objektes ist in 6.4 dargestellt. In Abbildung 6.5 ist ein Ausschnitt aus der Visualisierung dieser manipulierten Aufnahme dargestellt. Hier ist zu erkennen, dass trotz Verkleinerung des Objektes genug Messpunkte in den hinteren Bins vorhanden sind, um diese zu aktivieren.

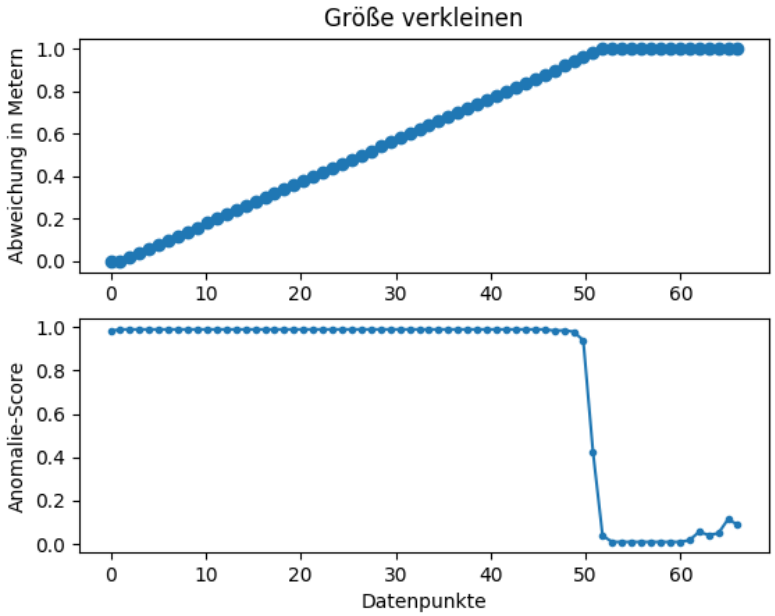

Abbildung 6.4: Verkleinerung eines Objektes

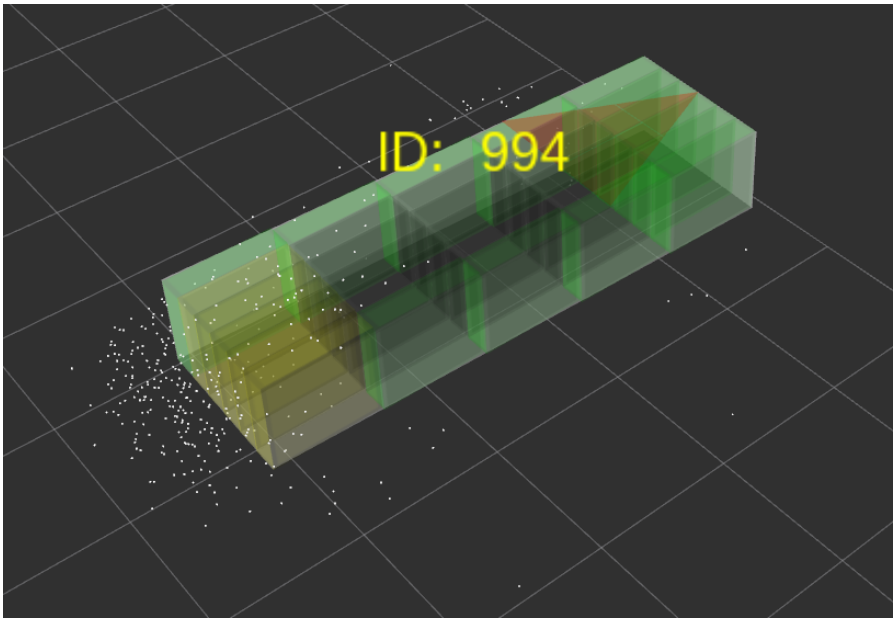

Abbildung 6.5: Visualisierung der Verkleinerung

6.4 Reale Daten

Zusätzlich zu dieser qualitativen Auswertung basierend auf Simulationsdaten wurde der Prototyp mit den Aufzeichnung einer realen Autobahnfahrt ausgewertet. Im Zuge dieser Auswertung wurde ersichtlich, dass die anhand der Simulationsdaten erprobten Konzepte auf echte Daten übertragbar sind. Jedoch wurde hier deutlich, dass Objekte in großer Entfernung nahezu in jedem Fall als wahrscheinliche Anomalie eingestuft werden. Die verwendete Objekterkennungs-Software ist imstande, schon anhand weniger Messpunkte Objekte zu erkennen. So hat ein Objekt in ca. 120 Metern Entfernung oft weniger als 10 Messpunkte. Da die Abweichung der vorhergesagten und der gemessenen Punkte in dieser Größenordnung relativ groß ist, wird das Objekt als Anomalie erkannt.

Weiterhin wurde anhand der echten Daten eine Laufzeitmessung des Prototyps durchgeführt. Hierfür wurden die Zeitstempel der Publikation der Objektliste und der Ausgabe der Anomalieerkennung miteinander verglichen. In Abbildung 6.6 ist das Laufzeitverhalten in Abhängigkeit zur Anzahl der erkannten Objekte dargestellt. Wie zu erwarten steigt die Laufzeit linear zur Anzahl der vorhanden Objekte. Für die Messung wurde die Freiraumerkennung deaktiviert. Während die durchschnittliche Laufzeit bei 14 Objekten ca 30ms beträgt, kann ein Durchlauf im Worst-Case selbst bei 10 Objekten bis

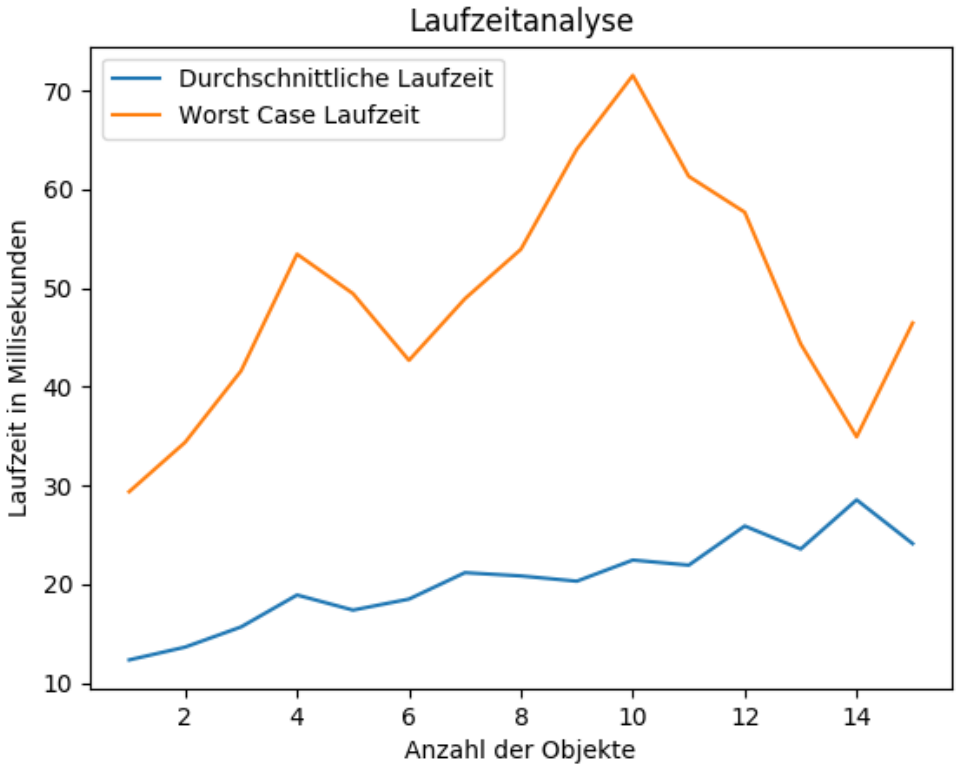

Abbildung 6.6: Laufzeit der Anomalieerkennung in Abhängigkeit zur Anzahl der Objekte

zu 70ms dauern. Bei einer Geschwindigkeit von 120 Kilometern pro Stunde würde ein Auto in dieser Zeit ungefähr 2,3 Meter zurücklegen. Die Lösungsansätze wurden zwar mit dem Ziel einer möglichst geringen Laufzeit entworfen, jedoch ist der Prototyp nicht hinsichtlich der geringstmöglichen Laufzeit optimiert. Basierend auf diesen Ergebnissen scheint es jedoch auch für höhere Geschwindigkeiten möglich zu sein, Angriffe rechtzeitig zu erkennen, um das Ansteueren der Aktoren in diesem Falle zu verhindern.

6.5 Regression

Um zu untersuchen, ob die Vorhersage der gemessenen Punkte durch Regressionverfahren verbessert werden kann, wurde ein Datensatz aus 21.340 Frames zusammengestellt, der für jeden Datenpunkt zusätzliche Zwischenergebnisse der verwendeten Algorithmen beinhaltet, die als Merkmale dienen. Die ausgewählten Merkmale sind:

- Distanz (*distance*)
- Vorhersage der Punkte basierend auf Öffnungswinkel und Auflösung (*nb_points_angular*)
- Vorhersage der Punkte basierend auf projizierter Fläche und Dichte (*nb_points*)
- Horizontaler Anteil des FoV (*hfov*)
- Vertikaler Anteil des FoV (*vfov*)
- Winkel in Blinkrichtung (*direction_angle*)
- Fläche der projizierten Bounding Box im FoV (*area*)

In Abbildung 6.7 sind die Relationen der Merkmale bezüglich der echten Anzahl der gemessenen Punkte visualisiert. X und Y Achse der Subplots entsprechen den normalisierten Werten der einzelnen Merkmale. Die Kolorierung der Punkte entspricht der gemessenen Anzahl der Messpunkte. Aus der Grafik erschließt sich, dass sich die Merkmale *nb_points_angular*, *hfov* und *vfov* recht gut zur Unterscheidung der tatsächlichen Anzahl an Messpunkten eignen. Je größer *nb_points_angular*, desto größer ist wie erwartet auch die tatsächliche Anzahl an gemessenen Messpunkten. Da die ausgewählten Merkmale nicht unabhängig sind, sind hier auch deutliche Relationen zu erwarten. Auffällig ist jedoch, dass die Relation der Merkmale *area* und der aus diesem Zwischenergebnis vorhergesagten Anzahl an Punkten *nb_points* weniger korreliert als *nb_points_angular* mit *hfov* und *vfov*. Merkmale deren Korrelation eine gute Vorhersage liefern, sind im Schaubild daran zu erkennen, dass der farbliche Verlauf der Punkte möglichst glatt ist.

Um zu testen, ob sich Hauptkomponeten für das Trainieren eines Regressionsmodells eignen, wurde eine Hauptkomponenten-Analyse durchgeführt. Aus der Abbildung 6.8 ist zu erschließen, dass ein größerer Wert von Hauptkomponente 1 (PC 1) einer höheren Anzahl an tatsächlichen Messpunkten entspricht. Jedoch beträgt die kumulative Summe der abgedeckten Varianz des verwendeten Datensatzes erst mit den 5 Hauptkomponenten 95%.

In Zusammenhang mit der zweiten Hauptkomponente, dargestellt in Abbildung 6.9, sind drei Korrelationen zu sehen. Die erste entspricht einer Verbindung der horizontalen und vertikalen FOV-Merkmalen, die zweite einer Verbindung aus den beiden berechneten Anzahlen der möglichen Messpunkte (*nb_points* und *nb_points_angular*). Die dritte

Abbildung 6.7: Relationen der Merkmale

Korrelation entspricht einer Verbindung der Distanz und den FOV-Merkmalen. Die ersten beiden Korrelationen ermöglichen eine gute Diskrimination von niedrigen und hohen (> 1500) tatsächlichen Messpunkten. Hingegen reichen die ersten beiden Hauptkomponenten nicht aus, um kleinere Anzahlen von tatsächlichen Messpunkten (0-500) gut zu unterscheiden. Daher resultiert die eher niedrige abgedeckte Varianz (ca. 63%) der gesamten Daten. Insgesamt ist durch die Hauptkomponentenanalyse ersichtlich, dass keine signifikante Reduktion der Dimensionalität erreicht werden kann bzw. die Hauptkomponenten sich nicht explizit als Eingangsmerkmale für die Regression anbieten. Aus diesem Grund werden im Folgenden die beschriebenen Eingangsmerkmale für das Training der Regression verwendet.

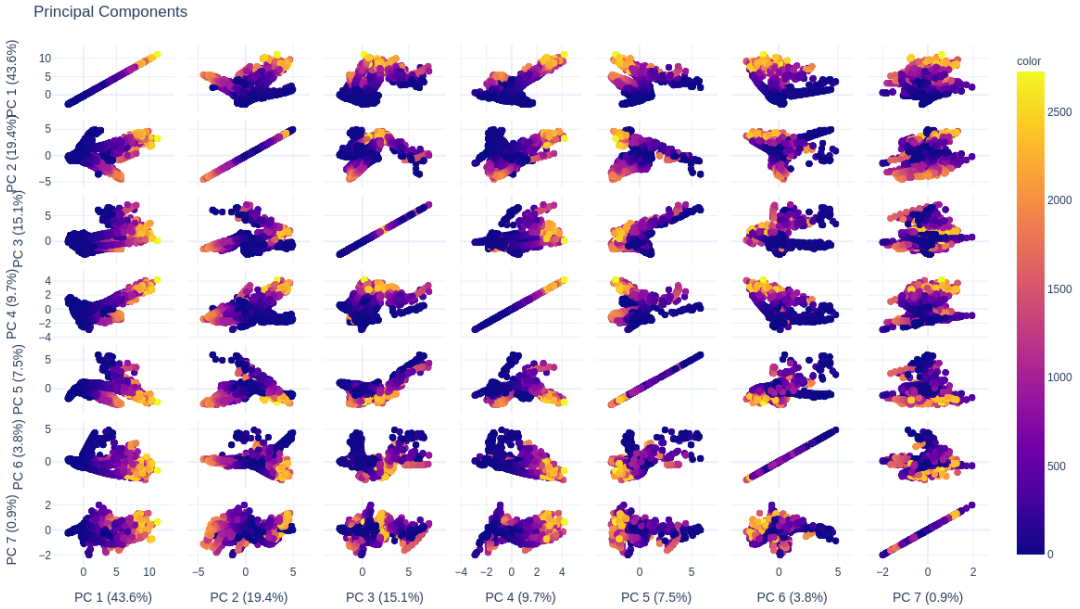

Abbildung 6.8: Hauptkomponenten

Metrik	**Ergebnis**
Mittlerer absoluter Trainingsfehler [Punkte]	44,16
Mittlerer absoluter Testfehler [Punkte]	44,69

Tabelle 6.1: Ergebnisse: Lineare Regression

Für das Experiment zur Verbesserung der Vorhersage der erwarteten Messpunkte werden die beschriebenen Eingangsmerkmale verwendet. Die vorhandenen Samples werden in ein Trainings- und ein Testset im Verhältnis 90% (19.206) zu 10% (2.134) unterteilt. Als Metrik für die Bewertung der Qualität der Vorhersage der erwarteten Messpunkte wird der mittlere absolute Fehler (engl. *mean absolute error*, MAE) in Messpunkten gewählt:

$$\text{MAE} = \frac{\sum |\hat{y} - y|}{\hat{y}}$$

Für die Regression wurden drei gängige Verfahren miteinander verglichen:

- Lineare Regression (Tabelle 6.1) [38]
- Regression durch einen Random Forest (Tabelle 6.2) [42, 3]
- Regression durch Extremely Randomized Trees (Tabelle 6.3) [14]

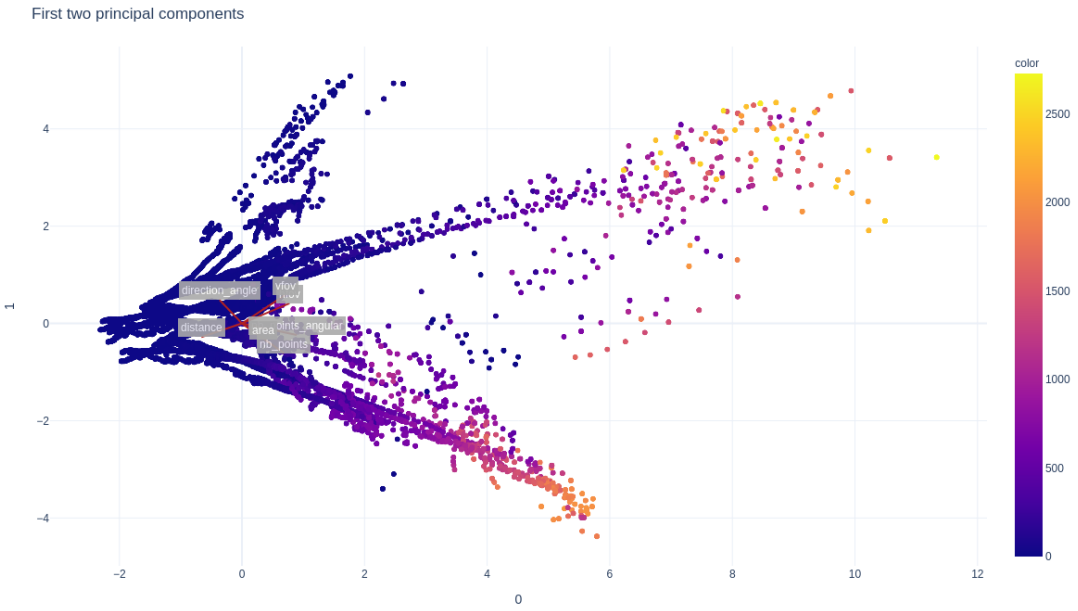

Abbildung 6.9: Korrelationen der Merkmale für die ersten zwei Hauptkomponenten

Metrik	**Ergebnis**
Mittlerer absoluter Trainingsfehler [Punkte]	2,2
Mittlerer absoluter Testfehler [Punkte]	7,87

Tabelle 6.2: Ergebnisse: Random Forest Regression

Die Extremely Randomized Trees zeigen die beste Performance hinsichtlich des Trainingsfehlers und des Testfehlers sowie eine leichte Verbesserung gegenüber der Regression mit einem herkömmlichen Random Forest. Insgesamt wurde ein durchschnittlicher Testfehler von ca. fünf Messpunkten erreicht. Die lineare Regression zeigt die schlechtesten Ergebnisse, da der Zusammenhang von Eingangsvariablen und Zielgröße durch eine Linearkombination nicht ausreichend modelliert werden kann und Eingangsvariablen nicht unabhängig voneinander sind. Obwohl ein Fehler von 5 Punkten auf großen Distanzen noch zu großen Einfluss auf die in dieser Arbeit verwendeten Konzepte hat, lässt sich

Metrik	**Ergebnis**
Mittlerer absoluter Trainingsfehler [Punkte]	0,11
Mittlerer absoluter Testfehler [Punkte]	4,88

Tabelle 6.3: Ergebnisse: Extremely Randomized Trees

sagen, dass Verfahren dieser Art durchaus verwendet werden können, um die Vohersage der gemessenen Punkte zu verbessern.

Zusammenfassend lässt sich sagen, dass die in dieser Arbeit vorgestellten Konzepte geeignet sind, um Manipulationen der Attribute eines Objektes zu erkennen. Schwächen sind Objekte in großer Entfernung, die nur eine geringe Anzahl an Messpunkten besitzen. Hier können *Machine-Learning* Verfahren für das Modellieren der Regression verwendet werden, um die Reichweite zu optimieren. Das geometrische Objekt-Modell ist der entscheidende Faktor für den Grad der erkennbaren Manipulationen. Die Optimierung dieses Modells würde direkt in einer verbesserten Erkennung minimaler Manipulationen resultieren. Das Laufzeitverhalten des Prototypen ist linear in Abhängigkeit zur Länge der Objektliste. Durch Optimierung der Laufzeit könnte es möglich sein, Angriffe auch bei höheren Geschwindigkeiten rechtzeitig zu erkennen, so dass Gegenmaßnahmen eingeleitet werden können.

7 Fazit und Ausblick

In diesem Kapitel wird die Arbeit noch einmal zusammengefasst und es wird ein Fazit gezogen. Zudem wird ein Ausblick über weitere mögliche Vorschungsrichtungen gegebenen.

7.1 Zusammenfassung

Moderne Komfort- und Sicherheitsfunktionen sind in der neuesten Generation von Automobilen durch LiDAR basierte maschinelle Wahrnehmung realisiert. Zusätzlich gilt LiDAR als Schlüsseltechnologie für zukünftige Entwicklungen im Bereich des autonomen Fahrens. Angriffe auf die Wahrnehmung können schweren Schaden zur Folge haben und gefährden die Sicherheit der am Straßenverkehr beteiligten Menschen.
In dieser Arbeit wurden Angriffe auf LiDAR-basierte Wahrnehmungsalgorithmen identifiziert und analysiert. Es wurden IDS Systeme als möglicher Schutzmechanismus vor dieser Art von Angriffen vorgestellt und es wurden Anforderungen formuliert und priorisiert. Basierend auf diesen Anforderungen wurden Lösungsansätze zum Schutze der Integrität der wichtigsten Attribute der Applikationsschnittstelle vorgestellt. Eine Auswahl an vorgestellten Konzepten wurden prototypisch realisiert. Der Prototyp wurde durch die Verwendung von Simulationsdaten und realen Daten hinsichtlich der funktionalen Anforderungen und der Qualtität der Funktionalitäten ausgewertet.

7.2 Ergebnisse

1. Die LiDAR basierte maschinelle Wahrnehmung ist integraler Bestandteil moderner Assistenzsysteme und eine essentielle Kernfunktion für den autonomen Betrieb eines Fahrzeugs.

2. Moderne Architekturen sind verteilte Systeme, in welchen Funktionalitäten der maschinellen Wahrnehmung und die der Assistenzsysteme auf verschiedenen Steuergeräten integriert sind.

3. Schwächen der Bus-Protokolle bedrohen die Integrität der Informationen, welche innerhalb des IVNs übertragen werden, und ermöglichen eine neue Art von Angriffsszenarien auf den *perception-layer*.

4. State-of-the-art Schutzmechanismen, um die Integrität der Transportschicht zu gewährleisten und Angriffe zu verhindern, sind aufgrund von limitierten Resourcen und Echtzeitanforderung nicht immer realisierbar.

5. Auf Anomalieerkennungsverfahren basiergende IDS, die in dieser Arbeit vorgestellt wurden, sind ein geeignetes Werkzeug um Angriffe auf das *perception-layer* zu erkennen.

6. Die Implementation dieser Konzepte muss hinsichtlich ihrer Laufzeit so optimiert werden, dass die Erkennung eines Angriffes auch bei höheren Geschwindigkeiten innerhalb eines Zeitrahmens erfolgt, der es erlaubt, rechtzeitig Schutzmaßnahmen einzuleiten um Schaden zu verhindern.

7. Verfahren der Regressionsanalyse aus dem Bereich des maschinellen Lernens eignen sich, um die in dieser Arbeit vorgestellten Konzepte zu optimieren.

7.3 Ausblick

Die in dieser Arbeit identifizierten Angriffe und vorgestellten Schutzkonzepte liefern eine Grundlage für mögliche weiterführende Forschungsarbeiten. Zum einen könnte das Verhalten realer ADAS Systeme hinsichtlich der hier identifizierten Angriffe untersucht werden, um quantifizierbare Anforderungen an die Laufzeit der IDS Mechanismen abzuleiten. Weiterhin kann untersucht werden, in welchem Maße die vorgestellten Mechanismen, hinsichtlich der identifizierten Schwäche bei der Erkennung von Objekten in großen Entfernungen, durch ein *Machine-Learning* Verfahren verbessert werden können. Als Alternative zu dem in dieser Arbeit vorgestellten Scoring-Mechanismus kann erforscht werden, ob sich *Machine-Learning* basierte Klassifikationsverfahren für die Ausgabe in Form von Labels eignen würden.

Literaturverzeichnis

[1] IEEE Standard for Ethernet Amendment 4: Physical Layer Specifications and Management Parameters for 1 Gb/s Operation over a Single Twisted-Pair Copper Cable. In: *IEEE Std 802.3bp-2016* (2016), S. i –22

[2] Alam Bhuiyan, Ifte K.: LiDAR Sensor for Autonomous Vehicle. (2017), 09

[3] Breiman, Leo: In: *Machine Learning* 45 (2001), Nr. 1, 5–32 S

[4] Börcs, A. ; Nagy, B. ; Benedek, C.: Instant Object Detection in Lidar Point Clouds. In: *IEEE Geoscience and Remote Sensing Letters* 14 (2017), Nr. 7, S. 992–996

[5] Caltagirone, L. ; Bellone, M. ; Svensson, L. ; Wahde, M.: LIDAR-based driving path generation using fully convolutional neural networks. In: *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, 2017, S. 1–6

[6] Chandola, Varun ; Banerjee, Arindam ; Kumar, Vipin: Anomaly Detection: A Survey. In: *ACM Comput. Surv.* 41 (2009), 07

[7] Changalvala, R. ; Malik, H.: LiDAR Data Integrity Verification for Autonomous Vehicle. In: *IEEE Access* 7 (2019), S. 138018–138031

[8] Cheng, B. H. C. ; Doherty, B. ; Polanco, N. ; Pasco, M.: Security Patterns for Automotive Systems. In: *2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2019, S. 54–63

[9] Elmadawi, K. ; Abdelrazek, M. ; Elsobky, M. ; Eraqi, H. M. ; Zahran, M.: End-to-end sensor modeling for LiDAR Point Cloud. In: *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, S. 1619–1624

[10] Fan, Y. ; Wu, B. ; Huang, C. ; Bai, Y.: Environment Detection of 3D LiDAR by Using Neural Networks. In: *2019 IEEE International Conference on Consumer Electronics (ICCE)*, 2019, S. 1–2

[11] Feng, D. ; Rosenbaum, L. ; Dietmayer, K.: Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection. In: *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, S. 3266–3273

[12] Fernández, Eduardo ; Jurjens, Jan ; Van Hilst, Michael ; Pernul, Günther: Using Security Patterns to Develop Secure Systems. In: *Software Engineering for Secure Systems: Industrial and Research Perspectives* (2010), 01. ISBN 9781615208388

[13] Galle, C. ; Amelung, J. ; Dallmann, T. ; Brueggenwirth, S.: Vehicle environment recognition for safe autonomous driving: Research focus on Solid-State LiDAR and RADAR. In: *AmE 2020 - Automotive meets Electronics; 11th GMM-Symposium*, 2020, S. 1–3

[14] Geurts, Pierre ; Ernst, Damien ; Wehenkel, Louis: Extremely randomized trees. In: *Machine Learning* 63 (2006), März, Nr. 1, S. 3–42

[15] Ghamisi, P. ; Höfle, B. ; Zhu, X. X.: Hyperspectral and LiDAR Data Fusion Using Extinction Profiles and Deep Convolutional Neural Network. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10 (2017), Nr. 6, S. 3011–3024

[16] Gindele, Tobias ; Brechtel, Sebastian ; Schroder, Joachim ; Dillmann, Rüdiger: Bayesian Occupancy Grid Filter for Dynamic Environments Using Prior Map Knowledge, 07 2009, S. 669 – 676

[17] Gonzalez-Ruiz, Alejandro ; Ghaffarkhah, Alireza ; Mostofi, Yasamin: An Integrated Framework for Obstacle Mapping With See-Through Capabilities Using Laser and Wireless Channel Measurements. In: *Sensors Journal, IEEE* 14 (2014), 01, S. 25–38

[18] Granström, Karl ; Lundquist, Christian ; Orguner, Umut: Tracking rectangular and elliptical extended targets using laser measurements, 07 2011, S. 1–8. – ISBN 978-1-4577-0267-9

[19] Guan, H. ; Yu, Y. ; Peng, D. ; Zang, Y. ; Lu, J. ; Li, A. ; Li, J.: A Convolutional Capsule Network for Traffic-Sign Recognition Using Mobile LiDAR Data With

Digital Images. In: *IEEE Geoscience and Remote Sensing Letters* 17 (2020), Nr. 6, S. 1067–1071

[20] Guidolini, R. ; Carneiro, R. V. ; Badue, C. ; Oliveira-Santos, T. ; De Souza, A. F.: Removing Movable Objects from Grid Maps of Self-Driving Cars Using Deep Neural Networks. In: *2019 International Joint Conference on Neural Networks (IJCNN)*, 2019, S. 1–8

[21] Gupta, A. ; Byrne, J. ; Moloney, D. ; Watson, S. ; Yin, H.: Tree Annotations in LiDAR Data Using Point Densities and Convolutional Neural Networks. In: *IEEE Transactions on Geoscience and Remote Sensing* 58 (2020), Nr. 2, S. 971–981

[22] Haas, R. E. ; Möller, D. P. F.: Automotive connectivity, cyber attack scenarios and automotive cyber security. In: *2017 IEEE International Conference on Electro Information Technology (EIT)*, 2017, S. 635–639

[23] Henniger, O. ; Apvrille, L. ; Fuchs, A. ; Roudier, Y. ; Ruddle, A. ; Weyl, B.: Security requirements for automotive on-board networks. In: *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, 2009, S. 641–646

[24] Hsu, C. P. ; Li, B. ; Solano-Rivas, B. ; Gohil, A. R. ; Chan, P. H. ; Moore, A. D. ; Donzella, V.: A Review and Perspective on Optical Phased Array for Automotive LiDAR. In: *IEEE Journal of Selected Topics in Quantum Electronics* 27 (2021), Nr. 1, S. 1–16

[25] Islam, Mhafuzul ; Chowdhury, Mashrur ; Li, Hongda ; Hu, Hongxin: Cybersecurity Attacks in Vehicle-to-Infrastructure (V2I) Applications and their Prevention. In: *Transportation Research Record: Journal of the Transportation Research Board* 2672 (2017), 11

[26] Izo, T. ; Grimson, W. E. L.: Unsupervised Modeling of Object Tracks for Fast Anomaly Detection. In: *2007 IEEE International Conference on Image Processing* Bd. 4, 2007, S. IV – 529–IV – 532

[27] Kocić, J. ; Jovičić, N. ; Drndarević, V.: Sensors and Sensor Fusion in Autonomous Vehicles. In: *2018 26th Telecommunications Forum (TELFOR)*, 2018, S. 420–425

[28] Kraemer, S. ; Bouzouraa, M. E. ; Stiller, C.: Utilizing LiDAR Intensity in Object Tracking. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, S. 1543–1548

[29] Levinson, J. ; Askeland, J. ; Becker, J. ; Dolson, J. ; Held, D. ; Kammel, S. ; Kolter, J. Z. ; Langer, D. ; Pink, O. ; Pratt, V. ; Sokolsky, M. ; Stanek, G. ; Stavens, D. ; Teichman, A. ; Werling, M. ; Thrun, S.: Towards fully autonomous driving: Systems and algorithms. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, S. 163–168

[30] Maltezos, E. ; Doulamis, A. ; Doulamis, N. ; Ioannidis, C.: Building Extraction From LiDAR Data Applying Deep Convolutional Neural Networks. In: *IEEE Geoscience and Remote Sensing Letters* 16 (2019), Nr. 1, S. 155–159

[31] Matti, D. ; Ekenel, H. K. ; Thiran, J.: Combining LiDAR space clustering and convolutional neural networks for pedestrian detection. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, S. 1–6

[32] Muhammad, S. ; Kim, G.: Visual Object Detection Based LiDAR Point Cloud Classification. In: *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2020, S. 438–440

[33] Parkinson, S. ; Ward, P. ; Wilson, K. ; Miller, J.: Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges. In: *IEEE Transactions on Intelligent Transportation Systems* 18 (2017), Nr. 11, S. 2898–2915

[34] Pomerleau, F. ; Krüsi, P. ; Colas, F. ; Furgale, P. ; Siegwart, R.: Long-term 3D map maintenance in dynamic environments. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, S. 3712–3719

[35] Ray, S. ; Wen Chen ; Bhadra, J. ; Al Faruque, M. A.: Extensibility in automotive security: Current practice and challenges. In: *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, 2017, S. 1–6

[36] Sato, S. ; Hashimoto, M. ; Takita, M. ; Takagi, Kiyokazu ; Ogawa, T.: Multilayer Lidar-Based Pedestrian Tracking in Urban Environments, 07 2010, S. 849 – 854

[37] Schlosser, J. ; Chow, C. K. ; Kira, Z.: Fusing LIDAR and images for pedestrian detection using convolutional neural networks. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, S. 2198–2205

[38] Schönfeld, P.: *Methoden der Ökonometrie: Stochastische Regressoren und simultane Gleichungen*. Vahlen, 1969 (Methoden der Ökonometrie). – ISBN 9783800601899

[39] Singh, A. ; Singh, M.: An empirical study on automotive cyber attacks. In: *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, 2018, S. 47–50

[40] Sivaraman, S. ; Trivedi, M. M.: Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis. In: *IEEE Transactions on Intelligent Transportation Systems* 14 (2013), Nr. 4, S. 1773–1795

[41] Song, H. ; Choi, W. ; Kim, H.: Robust Vision-Based Relative-Localization Approach Using an RGB-Depth Camera and LiDAR Sensor Fusion. In: *IEEE Transactions on Industrial Electronics* 63 (2016), Nr. 6, S. 3725–3736

[42] Tin Kam Ho: Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition* Bd. 1, 1995, S. 278–282 vol.1

[43] Ulagamuthalvi ; Felicita, J. B. J. ; Abinaya, D.: An Efficient Object Detection Model Using Convolution Neural Networks. In: *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, S. 142–147

[44] Vaquero, V. ; Pino, I. d. ; Moreno-Noguer, F. ; Solà, J. ; Sanfeliu, A. ; Andrade-Cetto, J.: Dual-Branch CNNs for Vehicle Detection and Tracking on LiDAR Data. In: *IEEE Transactions on Intelligent Transportation Systems* (2020), S. 1–12

[45] Vehicle Cybersecurity Systems Engineering Committee: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems. In: *SAE International* (2016)

[46] Viega, John ; McGraw, Gary: Building Secure Software: How to Avoid Security Problems the Right Way. (2001), 01

[47] Wampler, D. ; Fu, H. ; Zhu, Y.: Security Threats and Countermeasures for Intravehicle Networks. In: *2009 Fifth International Conference on Information Assurance and Security* Bd. 2, 2009, S. 153–157

[48] Wang, A. ; Xue, D. ; Wu, H. ; Iwahori, Y.: LiDAR Data Classification Based on Improved Conditional Generative Adversarial Networks. In: *IEEE Access* 8 (2020), S. 209674–209686

[49] Wang, Z. ; Wu, Y. ; Niu, Q.: Multi-Sensor Fusion in Automated Driving: A Survey. In: *IEEE Access* 8 (2020), S. 2847–2868

[50] Warakagoda, N. ; Dirdal, J. ; Faxvaag, E.: Fusion of LiDAR and Camera Images in End-to-end Deep Learning for Steering an Off-road Unmanned Ground Vehicle. In: *2019 22th International Conference on Information Fusion (FUSION)*, 2019, S. 1–8

[51] Wasicek, A. ; Burakova, Yelizaveta: Context-aware Intrusion Detection in Automotive Control Systems, 2017

[52] Wu, W. ; Li, R. ; Xie, G. ; An, J. ; Bai, Y. ; Zhou, J. ; Li, K.: A Survey of Intrusion Detection for In-Vehicle Networks. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), Nr. 3, S. 919–933

[53] Yeh, T. ; Lin, S. ; Lin, H. .. ; Chan, S. ; Lin, C. ; Lin, Y.: Traffic Light Detection using Convolutional Neural Networks and Lidar Data. In: *2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, 2019, S. 1–2

[54] Yoo, Han W. ; Druml, Norbert ; Brunner, David ; Schwarzl, Christian ; Thurner, Thomas ; Hennecke, Marcus ; Schitter, Georg: MEMS-based lidar for autonomous driving. In: *e & i Elektrotechnik und Informationstechnik* 135 (2018), Oct, Nr. 6, S. 408–415. – ISSN 1613-7620

[55] Yoon, D. ; Tang, T. ; Barfoot, T.: Mapless Online Detection of Dynamic Objects in 3D Lidar. In: *2019 16th Conference on Computer and Robot Vision (CRV)*, 2019, S. 113–120

[56] Zehang Sun ; Bebis, G. ; Miller, R.: On-road vehicle detection: a review. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), Nr. 5, S. 694–711

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „— bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] — ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen."

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende Masterarbeit – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

Erkennung von Cyberangriffen auf LIDAR basierte Wahrnehmungsalgorithmen im Automobil

ohne fremde Hilfe selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

_____	_____	_____
Ort	Datum	Unterschrift im Original