Dissertation zur Erlangung des Doktorgrades der Technischen Fakultät der Albert-Ludwigs-Universität Freiburg im Breisgau

Verteilte Datenvalidierungsnetzwerke im industriellen Umfeld

Kevin Wallis M.Sc.

04. Dezember 2023

Institut für Informatik
Technische Fakultät
Albert-Ludwigs-Universität Freiburg

Dekan
Prof. Dr. Roland Zengerle

Referenten
Prof. Dr. Christian Schindelhauer (Universität Freiburg)
Prof. Dr. Christoph Reich (Hochschule Furtwangen)

Datum der mündlichen Prüfung
15. November 2023

"Der Weg ist das Ziel"
Konfuzius (551 – 479 v. Chr.)

Eidesstattliche Erklärung

Ich erkläre eidesstattlich, dass ich die Arbeit selbständig verfasst, ganz oder in Teilen noch nicht als Prüfungs- oder Studienleistung vorgelegt und keine anderen als die angegebenen Hilfsmittel benutzt habe. Sämtliche Stellen der Arbeit, die benutzten Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellenangaben kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen sowie für Quellen aus dem Internet.

Kevin Wallis M.Sc.

Inhaltsverzeichnis

Danksagung

An erster Stelle möchte ich mich bei meinen beiden Betreuern Christoph Reich und Christian Schindelhauer bedanken, die mir die Fertigstellung dieser Dissertation ermöglicht haben. Für die grossartige Unterstützung während der Promotion durch fachliche und technische Diskussionen, die mich auf meinem wissenschaftlichen und persönlichen Weg weitergebracht haben.

Zudem bedanke ich mich bei allen Professoren und Professorinnen sowie Doktoranden und Doktorandinnen, die ich während dieser Zeit kennenlernen durfte und durch die ich Einblicke in die verschiedensten Aspekte der Wissenschaft erhalten habe. Ein besonderer Dank gilt meinem Co-Autor Fabian Schillinger, mit dem ich viele verschiedene Themen diskutieren konnte. Dadurch eröffneten sich für mich neue wissenschaftliche Wege, die in wissenschaftlichen Publikationen ausgearbeitet und veröffentlicht wurden.

Ein Dankeschön gebührt allen Unternehmen und Vorgesetzten die mir den Freiraum gegeben haben mich der Forschung zu widmen. Hierbei möchte ich namentlich Wouter Meyers, Rainer Thieringer, Matthias Bühler, Thomas Wenger und Jörg Heimoz danken. Ohne eure Unterstützung hätten einige der wissenschaftlichen Publikationen und diese Dissertation nie entstehen können.

Ich danke meinen Grosseltern Elfriede und Wilhelm Wallis. Sie haben mich während meiner gesamten schulischen und wissenschaftlichen Laufbahn unterstützt und stets ein offenes Ohr für mich gehabt.

Zuletzt möchte ich mich bei meiner Frau Geanina Wallis bedanken, sie ist meine Stütze in allen Belangen und meine Muse für neue Ideen. Ohne sie wären einige meiner Arbeiten nie entstanden. Besonders die gemeinsamen Spaziergänge mit unserer Hündin Mia haben mir frischen Wind und Energie gegeben.

Abstract

This dissertation presents a system for improving the security of data validation in an industrial environment. For this purpose, the four industrial revolutions are used, starting with the first industrial revolution with mechanisation, urbanisation and the steam engine up to the fourth industrial revolution, also called Industry 4.0, to show why cyber security has become increasingly important in recent decades. The overall picture is further emphasised by an analysis of the threat situation in the industrial environment. This shows that the threats have increased drastically, especially in recent years, as production has become increasingly networked. Previous problems, such as poor password management or public access to production computers, are now only part of the total attack surface. Above all, an invisible factor, the network, is now part of it.
In addition to the general need for increased security in the industrial environment, data quality is an essential part of this thesis. For this purpose, the terms data and data quality are defined and, based on this, the metrics for data quality are listed. The metrics are divided into four categories: intrinsic and contextual metrics, as well as metrics for representativeness and accessibility. In addition, the data pre-processing process is described in more detail in the context of data quality and predictions based on given data. One step in the data pre-processing process is validation, which is responsible for determining the quality of the data. Data pre-processing is a crucial part of the given scientific work, as the improvement of data validation security is established in this process step.
After describing the main aspects of cyber security and data quality, an abstract description of the different data validation structures in the industrial environment is given. A comparison of the different approaches based on seven categories is also provided. The main focus is on the distributed data validation approach, here called Distributed Data Validation Network (DDVN). For this approach, the individual components such as source, gateway, validator and accumulator are explained in detail and the communication between them is described using sequence diagrams and policy files.
The description of DDVN is followed by a security analysis using STRIDE and a consideration of the most common types of attacks. Causes of validator misbehaviour are also described. Not all misbehaviour is due to an attack; a node crash or network interruption is also conceivable.
Based on the STRIDE analysis, several approaches to improving the security of a verteilten Datenvalidierungsnetzwerk (DDVN) are presented. One approach is to separate identity and authorisation through certificates. This makes it easier to

revoke an authorisation without having to reissue the identity of an entity. In addition, this results in different possibilities for the allocation of authorisations. Another approach is to enrich the actual data to be validated with contextual information. This makes data validation more accurate and allows correlating errors in the data to be detected. For the third protection approach, the algorithms under investigation for the validators are first defined based on their structure (tree structure or acyclic directed graph without multiple edges) and then the degree of corruption κ and the degree of impact η are specified for the respective structure. Based on the previously given metrics, three possibilities for increasing the security of the two structures are investigated and the resulting potential for improvement is determined using the previously defined η. The fourth approach describes the classification of validators into logical clusters and thus a protection against zero-day exploits based on certain properties. The last approach introduces the two aggregation functions *t-Total Quorum* and (s,t)*-Nested Quorum* for the results of the validators. This minimises the influence of corrupted validators.

Zusammenfassung

In dieser Arbeit wird ein System zur Verbesserung der Sicherheit der Datenvalidierung im industriellen Umfeld vorgestellt. Dazu wird anhand der vier industriellen Revolutionen, beginnend mit der ersten industriellen Revolution mit Mechanisierung, Urbanisierung und Dampfmaschine bis hin zur vierten industriellen Revolution, auch Industrie 4.0 genannt, aufgezeigt, warum die Cybersicherheit in den letzten Jahrzehnten immer mehr an Bedeutung gewonnen hat. Unterstützt wird dies durch eine Analyse der Bedrohungslage im industriellen Umfeld. Dabei zeigt sich, dass insbesondere in den letzten Jahren, auch durch die zunehmende Vernetzung in der Produktion, die Bedrohungslage drastisch gestiegen ist. Bisherige Probleme, wie der schlechte Umgang mit Passwörtern oder auch der öffentliche Zugriff auf Produktionsrechner, sind nur noch ein Teil der gesamten Angriffsfläche. Vor allem ein nicht sichtbarer Faktor, das Netzwerk, gehört nun dazu.

Neben der generellen Notwendigkeit, die Sicherheit im industriellen Umfeld zu erhöhen, ist zusätzlich die Datenqualität ein wesentlicher Bestandteil der vorliegenden Dissertation. Dazu werden die Begriffe Daten und Datenqualität genau definiert und darauf aufbauend Metriken für die Datenqualität definiert. Die Metriken werden in vier Kategorien unterteilt: intrinsische und kontextuelle Metriken sowie Metriken für Repräsentativität und Zugänglichkeit. Darüber hinaus wird der Prozess der Datenvorverarbeitung im Kontext der Datenqualität und der Vorhersage mit gegebenen Daten näher beschrieben. Ein Prozessschritt der Datenvorverarbeitung ist die Validierung, die für die Bestimmung der Datenqualität verantwortlich ist. Die Datenvorverarbeitung ist ein wesentlicher Teil der vorliegenden wissenschaftlichen Arbeit, da die Verbesserung der Sicherheit der Datenvalidierung in diesem Prozessschritt etabliert ist.

Nach der Beschreibung der wesentlichen Aspekte der Cybersicherheit und Datenqualität erfolgt eine abstrakte Beschreibung der verschiedenen Datenvalidierungsstrukturen im industriellen Umfeld. Darüber hinaus erfolgt ein Vergleich der verschiedenen Ansätze anhand von sieben Kategorien. Der Schwerpunkt liegt dabei auf dem Ansatz der verteilten Datenvalidierung, hier als Distributed Data Validation Network (DDVN) bezeichnet. Für diesen werden sowohl die einzelnen Komponenten wie Source, Gateway, Validator und Akkumulator im Detail erläutert, als auch die Kommunikation untereinander anhand von Sequenzdiagrammen und Policy-Dateien beschrieben.

Die Beschreibung des DDVNs wird durch eine detaillierte Sicherheitsanalyse mittels STRIDE und einer Betrachtung der bekanntesten Angriffsarten ergänzt.

Ausserdem werden die Ursachen für Fehlverhalten von Validierern aufgezeigt. Nicht jedes Fehlverhalten ist auf einen Angriff zurückzuführen, auch ein Knotenabsturz oder eine Netzwerkunterbrechung sind möglich.
Auf der Grundlage der STRIDE Analyse werden verschiedene Ansätze zur Verbesserung der Sicherheit des DDVNs vorgestellt. Ein Ansatz ist die Trennung von Identität und Berechtigung durch Zertifikate. Dadurch kann eine Berechtigung leichter entzogen werden, ohne dass die Identität einer Entität neu ausgestellt werden muss. Ausserdem ergeben sich dadurch verschiedene Möglichkeiten der Berechtigungsvergabe. Ein weiterer Ansatz beschreibt die Anreicherung der eigentlichen Daten für die Validierung mit Kontextinformationen. Dadurch wird die Datenvalidierung genauer und es können korrelierende Fehler in den Daten erkannt werden. Für den dritten Schutzansatz werden zunächst die untersuchten Algorithmen für die Validierer anhand ihrer Struktur (Baumstruktur oder azyklischer gerichteter Graph ohne Mehrfachkanten) definiert und anschliessend der Korruptionsgrad κ und der Auswirkungsgrad η für die jeweilige Struktur angegeben. Basierend auf den zuvor angegebenen Metriken werden drei Möglichkeiten zur Erhöhung der Sicherheit der beiden Strukturen untersucht und das daraus resultierende Verbesserungspotential mit Hilfe der zuvor definierten η bestimmt. Der vierte Ansatz beschreibt die Einteilung von Validierern in logische Cluster und damit einen Schutz gegen Zero-Day Exploits basierend auf bestimmten Eigenschaften. Im letzten Ansatz werden die beiden Aggregationsfunktionen *t-Total Quorum* und (s,t)-*Nested Quorum* für die Ergebnisse der Validierer eingeführt. Dadurch wird der Einfluss korrumpierter Validierer minimiert.

1. Einleitung

Die erste industrielle Revolution begann zwischen 1700 und 1750 mit der Urbanisierung und der Entwicklung und Nutzung der Dampfmaschine. Mittlerweile, 270 Jahre später, befinden wir uns in der vierten industriellen Revolution, im deutschsprachigen Raum meist Industrie 4.0 genannt [1]. Ein zentraler Bestandteil von Industrie 4.0 ist die Digitale Transformation [2], was der Digitalisierung der Produktion entspricht. Dazu gehören die Vernetzung von Geräten (Internet of Things) [3], die Nutzung von Sensorinformationen zur Überwachung des Zustands einer Maschine (Condition Monitoring) [4], die Optimierung von Prozessen (Process Mining) [5] und die Vorhersage notwendiger Wartungsarbeiten an Maschinen (Predictive Maintenance) [6], die intelligente Auslagerung von Produktionsschritten (Smart Manufacturing) [7] und eine Losgrösse von eins und damit maximale Flexibilität für die Kunden.

Die digitale Transformation birgt einerseits ein grosses Potenzial für Hersteller, Integratoren und Betreiber von Maschinen sowie für den Endkunden der produzierten Teile, andererseits ergeben sich eine Vielzahl neuer Angriffsziele und Angriffsmöglichkeiten. Dazu gehören Remote-Angriffe, bei denen eine angreifende Person von jedem vernetzten Ort aus auf Maschinen in der Produktion zugreifen und diese manipulieren kann. Darüber hinaus ist neben der aktiven Beeinflussung auch das Abhören und damit die Betriebsspionage möglich. Die Weitergabe oder der Verkauf von Betriebsgeheimnissen ist so vereinfacht möglich. Durch die im Mai 2018 in Kraft getretene Datenschutzgrundverordnung (DSGVO) [8] führt der fahrlässige Umgang mit personenbezogenen Daten nicht nur zu einem Imageschaden, sondern kann auch zur Anzeige gebracht werden. Zwei Ziele der vorliegenden Arbeit sind es, ein erweitertes Bewusstsein für das Thema IT-Sicherheit im Bereich der digitalen Transformation zu schaffen und ein Framework zur Verfügung zu stellen, das zur Erhöhung der IT-Sicherheit im industriellen Umfeld eingesetzt werden kann.

Einer der zentralen neuen Bereiche im Umfeld von Industrie 4.0 ist das Arbeiten mit Daten. Das Senden, Beobachten und Validieren von Daten sind nur einige grobe Teilaufgaben im Umgang mit Daten. Speziellere Anwendungsfälle sind Condition Monitoring und Predictive Maintenance, bei denen Daten genutzt werden, um den Maschinenbetreibenden den Zustand der Maschine zurückzumelden und Vorhersagen über den nächsten Wartungsvorgang zu liefern. Für solche und ähnliche Anwendungen ist eine hohe Qualität der verwendeten Daten unerlässlich. Ansonsten sind hohe Ausfallzeiten, Umplanungen von

Produktionsaufträgen sowie unnötige Wartungsaufträge und damit verbundener finanzieller Mehraufwand die Folge [9, 10]. An dieser Stelle setzt die Datenvorverarbeitung an. Die Vorverarbeitung befasst sich mit der Bereinigung der gegebenen Daten, dazu gehören das Entfernen verrauschter oder inkonsistenter Daten, die Transformation der Daten in ein kompatibles Format, das Auffüllen fehlender Daten, das Zusammenführen von Daten aus verschiedenen Datenquellen und die Reduktion der Daten auf die wesentlichen Merkmale [11, 12, 13]. Ein wesentlicher Aspekt wird dabei jedoch meist ausser Acht gelassen, die IT-Sicherheit, d.h. die Datenvorverarbeitung hat einen starken Einfluss auf den nachgelagerten Prozessschritt der Datenanalyse. Wenn in der Datenvorverarbeitung eine Manipulation der Daten stattgefunden hat, sei es bewusst oder unbewusst, kann sich das Ergebnis der Datenanalyse verändern. Insbesondere im Umfeld von IoT, Edge-Computing, Fog-Computing und Cloud-Computing [14], in dem Datenvorverarbeitung und Datenanalyse nicht mehr auf einer einzigen Ressource stattfinden, sondern verteilt sind, sind Manipulationsszenarien allgegenwärtig.

Die vorliegende Dissertation greift dieses Problem auf und zeigt, wie mittels eines DDVN die Datenvorverarbeitung zur Erhöhung der Sicherheit um einen Datenvalidierungsschritt erweitert werden kann. Dabei wird die benutzenden Personen der Daten über die Qualität der zur Verfügung gestellten Daten informiert. Die Benutzer und Benutzerinnen entscheidet dann selbst, ob die Qualität für den vorliegenden Anwendungsfall ausreichend ist oder ob eine bessere Datenquelle verwendet werden soll.

1.1. Wissenschaftliche Beiträge der Dissertation

Die folgenden wissenschaftlichen Beiträge wurden im Rahmen dieser Forschungsarbeit von mir geleistet:

- Die Bedrohungslage der Cybersicherheit im industriellen Umfeld basierend auf mehreren bestehenden Analysen ermittelt.
- Eine abstrakte Beschreibung der verschiedenen Datenvalidierungsstrukturen im industriellen Umfeld erstellt.
- Die verschiedenen Schwachstellen wie Single Point of Failure oder korrumpierte Knoten in der Datenvorverarbeitung im industriellen Umfeld durch Bedrohungsanalysen (STRIDE und bekannte Angriffsarten) identifiziert.
- Für einige der oben beschriebenen Schwachstellen Schutzmechanismen entwickelt und in ein kohärentes Framework namens DDVN überführt.
- Auf Grundlage bestehender Datenqualitätskategorien eine Metrik für das Sicherheitsniveau abgeleitet.

- Eigene Sicherheitsmetriken, wie den Korruptionsgrad κ und den Auswirkungsgrad η zur Bestimmung der Sicherheit des DDVNs definiert.
- Eine detaillierte Analyse der Anwendung des DDVNs auf zwei verschiedene Algorithmen (gerichteter azyklischer Graph ohne Mehrfachkanten und Baumstruktur) zur Sicherung der Datenqualität als Teil der Datenvorverarbeitung durchgeführt.
- Zwei verschiedenen Algorithmen (t-*Total Quorum* und (s,t)-*Nested Quorum*) für die Aggregation der Validierungsergebnisse, um den Einfluss von korrumpierten Validierern zu reduzieren, eingeführt.

1.2. Mitverfasste Publikationen

Ein Teil dieser Dissertation wurde bereits in von Experten begutachteten Konferenzen veröffentlicht. Diese Veröffentlichungen sind:

- Kevin Wallis, Christoph Reich, Blesson Varghese und Christian Schindelhauer. QUDOS: Quorum-Based Cloud-Edge Distributed DNNs for Security Enhanced Industry 4.0. In IEEE/ACM 14th International Conference on Utility and Cloud Computing (UCC), 2021
- Kevin Wallis, Fabian Schillinger, Christoph Reich und Christian Schindelhauer. Protection Measurements for Distributed Decision Trees. In International Journal for Information Security Research (IJISR), 2021
- Daniel Schönle, Kevin Wallis, Jan Stodt, Christoph Reich, Dominik Welte und Axel Sikora. Industry Use Cases on Blockchain Technology. In IGI Global, Industry Use Cases on Blockchain Technology Applications in IoT and the Financial Sector, 2021
- Mohammed B. M. Kamel, Kevin Wallis, Peter Ligeti und Christoph Reich. Distributed Data Validation Network in IoT: A Decentralized Validator Selection Model. In ACM 10th International Conference on the Internet of Things (IoT), 2020
- Kevin Wallis, Fabian Schillinger, Christoph Reich und Christian Schindelhauer. Corrupted Nodes and their Impact on a Distributed Decision Tree. In World Congress on Internet Security (WorldCIS), 2020
- Kevin Wallis, Fabian Schillinger, Elias Backmund, Christoph Reich und Christian Schindelhauer. Context-Aware Anomaly Detection for the Distributed Data Validation Network in Industry 4.0 Environments. In IEEE Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4), 2020
- Kevin Wallis, Jan Stodt, Eugen Jastremskoj und Christoph Reich. Agreements between Enterprises digitized by Smart Contracts in the Domain of Industry

4.0. In 10th International Conference on Computer Science, Engineering and Applications (CCSEA), 2020

- Kevin Wallis, Fabian Schillinger, Christoph Reich und Christian Schindelhauer. Safeguarding Data Integrity by Cluster-Based Data Validation Network. In IEEE Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4), 2019
- Kevin Wallis, Marc Merzinger, Christoph Reich und Christian Schindelhauer. A Security Model based Authorization Concept for OPC Unified Architecture. In ACM 10th International Conference on Advances in Information Technology (IAIT), 2018
- Kevin Wallis, Marc Hüffmeyer, Ayhan Soner Koca und Christoph Reich. Access Rules Enhanced by Dynamic IIoT Context. In 3rd International Conference on Internet of Things, Big Data and Security (IoTBDS), 2018
- Kevin Wallis, Florian Kemmer, Eugen Jastremskoj und Christoph Reich. Adaption of a Privilege Management Infrastructure (PMI) Approach to Industry 4.0. In IEEE 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2017
- Christoph Reich, Hendrik Kuijs, Kevin Wallis und Timo Bayer. Architektur zum Schutz der Privatsphäre in AAL-Systemen. In Assistive Systeme und Technologien zur Förderung der Teilhabe für Menschen mit Hilfebedarf: Ergebnisse aus dem Projektverbund ZAFH-AAL, 2017
- Kevin Wallis und Christoph Reich. Secure Zero Configuration of IoT Devices - A Survey. In Symposium on Information and Communication Systems (SinCom), 2016

1.3. Gliederung der Dissertation

Der erste Abschnitt behandelt die Einleitung mit der Motivation, die wissenschaftlichen Beiträge und die Co-Autorenschaften. Im zweiten Kapitel werden verwandte Arbeiten vorgestellt, die Beiträge im Bereich der Datenvalidierung, der verteilten Systeme und der entsprechenden Schutzmechanismen umfassen. In Kapitel drei wird der Ursprung der Cybersicherheit über die industriellen Revolutionen hergeleitet. Darüber hinaus wird die Bedrohungslage im Bereich der Cybersicherheit aufgezeigt und ein grober Überblick über bestehende Sicherheitsstandards gegeben. Die Datenqualität sowie der Datenvalidierungsprozess werden in Kapitel vier ausführlich erläutert. Aufbauend auf den Themen Cybersicherheit und Datenqualität bzw. Datenvalidierung erfolgt in Kapitel fünf die Einführung des DDVN anhand einer abstrakten Beschreibung der verschiedenen Datenvalidierungssysteme. Die Kommunikation zwischen den verschiedenen Komponenten des DDVN wird in

Kapitel sechs erläutert. Nachdem das DDVN und seine Kommunikation im Detail erklärt wurden, wird in Kapitel sieben die Sicherheitsanalyse des verteilten Netzwerks durchgeführt. Dabei wird sowohl eine STRIDE-Analyse als auch eine Analyse auf Basis bekannter Angriffsarten durchgeführt. Für die sich aus der Analyse ergebenden Sicherheitsschwachstellen werden in Kapitel acht einige Ansätze zur Sicherheitsverbesserung aufgezeigt. Abschliessend wird im letzten Kapitel eine Zusammenfassung der Dissertation sowie ein Ausblick auf weitere Forschungsfragen für das DDVN gegeben.

2. Verwandte Arbeiten

Das folgende Kapitel beschreibt verwandte Arbeiten zum Schutz verteilter Systeme und zur Verbesserung der Validierung während der Datenvorverarbeitung.

2.1. Datenvalidierung

Im Folgenden werden einige der aktuellsten Ansätze zur Datenvalidierung aufgeführt. Der Übergang zwischen dem eigentlichen Datenvalidierungsprozess und der Bestimmung der Datenqualität ist in den meisten wissenschaftlichen Arbeiten fliessend, weshalb hier keine explizite Unterscheidung zwischen den beiden Begriffen vorgenommen wird. Der Unterschied zwischen den hier vorgestellten Ansätzen und dem Ansatz dieser Dissertation besteht darin, dass der Fokus der Dissertation auf der Sicherheit des Datenvalidierungsprozesses liegt. Insbesondere Themen wie Datenmanipulation oder ein Angriff auf das Validierungssystem werden in anderen wissenschaftlichen Arbeiten meist wenig bis gar nicht erwähnt.

Husom et al. [15] beschreiben eine Unsupervised Learning Pipeline zur Validierung von Sensordaten, kurz UDAVA, im Produktionsumfeld. Der UDAVA-Ansatz besteht aus drei Schritten: 1) Vorverarbeitung der Referenzdaten, wobei die Zeitreihen in Subsequenzen zerlegt und Merkmale extrahiert werden, 2) Ausführung des maschinellen Lernalgorithmus bzw. in diesem Fall des Clustering-Algorithmus und 3) Etikettierung und Bewertung der Validierungsergebnisse. Die Pipeline selbst kann auf einem einzelnen Computer, einem Edge Device oder in der Cloud als Container ausgeführt werden.
Der UDAVA-Ansatz kann mit dem Ansatz dieser Dissertation kombiniert werden. Dabei wird der UDAVA Validierungsansatz auf einem einzelnen Validierer installiert und führt eigene Validierungsvorgänge durch.

Taleb et al. [16] definieren ein Modell zur Bestimmung der Datenqualität unstrukturierter Daten im Big Data Umfeld. Das Modell umfasst sieben Komponenten. Wobei eine Komponente die Qualitätsanforderungen abbildet, fünf Komponenten für die Datenaufbereitung mit Datensammlung, Featureauswahl etc. zuständig sind und eine Komponente am Ende des Prozesses zur Qualitätsbestimmung verwendet wird.
Grundsätzlich ist es möglich das Modell auch im DDVN anzuwenden. Hierfür sind einige Inhalte auszuarbeiten, dazu zählen *a*) Taleb et al. fokussieren sich auf unstrukturierte Daten, das DDVN macht hierfür keine Einschränkung, *b*) sie haben

keinen verteilten Prozess aufgezeigt, somit muss definiert werden, welche Komponente die Schritte vier bis sechs (aufteilten der Daten, aufbereiten der Daten und Merkmalsauswahl) vornehmen, *c*) wo befindet sich das Data Quality Repository und wie wird es mit Metriken bzw. Reports befüllt (eine Möglichkeit ist der Akkumulator), etc.

Xu et al. [17] betrachten Daten als ein Produkt und beschreiben einen Ansatz zur Bestimmung einer Metrik für die Datenqualität, die auf mehreren Eigenschaften basiert. Die Eigenschaften bestehen aus den vier von Pipino et al. [18]: Fehlerfreiheit (FE), angemessene Datenmenge (AAD), Anpassbarkeit an verschiedene Aufgaben (EM) und Relevanz der Daten (R) sowie einer neu definierten Eigenschaft, dem Grad der Unausgewogenheit der Daten (IL). Daraus ergibt sich die kumulierte Datenqualität als gewichteter Mittelwert der fünf Eigenschaften $\overline{q} = w_1 \cdot (FE + AAD + EM + R) + w_2 \cdot IL$. Die Berechnung der Datenqualität wird verwendet, um zu entscheiden, ob vorhandene Daten für das Training eines maschinellen Lernmodells verwendet werden sollen oder ob neue Daten erfasst werden sollen.

Quevedo et al. [19, 20] befassen sich mit der Validierung von Daten im Zusammenhang mit der Wasserversorgung. Sie weisen darauf hin, dass es sich bei der Wasserversorgung um eine kritische Infrastruktur handelt und daher eine umfassende Überwachung der Daten erforderlich ist. Wenn die ermittelte Qualität der verwendeten Daten nicht den Anforderungen entspricht, wird eine Datenrekonstruktion durchgeführt. Zur Bestimmung der Qualität werden fünf Stufen, Level 0 bis Level 4 genannt, durchlaufen: 0) Kommunikation, werden alle Daten korrekt erfasst und in einer geeigneten Abtastrate ermittelt, 1) Physikalische Grenzen, entsprechen die ermittelten Daten der Sensoren den physikalisch möglichen Werten, 2) Trendentwicklung, liegen die Datenwertänderungen aufeinanderfolgender Daten im möglichen Bereich, 3) Gerätezustand, ist die Korrelation der für ein Gerät erfassten Sensorwerte korrekt und 4) Räumliche Konsistenz, ist die Korrelation der Sensorwerte, die sich am gleichen Ort befinden, korrekt.

Xin et al. [21] kombinieren einen datengetriebenen Ansatz zur Fehlererkennung in einem Sensor (Sensor Fault Detection - SFD) mit einem systemweiten probabilistischen Validierungsmodell (System Model Verification - SMV) zu einem gemeinsamen dynamischen probabilistischen Validierungsmodell. Damit bestimmen sie die Glaubwürdigkeit der gemessenen Sensordaten.
Der Ansatz von Xin et al. kann mit dem DDVN umgesetzt werden, da es keine Einschränkung bezüglich der Verteilung des dynamischen probabilistischen Validierungsmodells gibt. Darüber hinaus werden keine Angaben zur Sicherheit dieses Ansatzes gemacht.

2.2. Schutzmechanismen für die Datenvalidierung

Mohammed et al. [22] verwenden zur Sicherung der Datenqualität einen Zero Trust Ansatz, auch Zero Trust Data Quality Framework (ZTDQF) genannt. Hierbei wird keinem Datensatz von vornherein vertraut und erst durch eine erfolgreiche Validierung die Vollständigkeit und Gültigkeit nachgewiesen. Das ZTDQF besteht aus drei Teilkomponenten: 1) einem Repository mit den Datenqualitätsregeln der datenbesitzenden Person, 2) der Engine, die die Datenqualitätsregeln ausführt und 3) einer Policy und Standards Engine, die prüft, ob die Daten der Spezifikation entsprechen. Die Unterschiede zwischen dem ZTDQF und der vorliegenden Dissertation sind: *a*) Das ZTDQF ist kein verteiltes System und verwendet nur eine einzige Engine für die Datenvalidierung anstatt einer Vielzahl von Knoten, *b*) es betrachtet die Vollständigkeit und die Gültigkeit eines Datensatzes getrennt und *c*) soweit aus dem Paper entnommen werden konnte, beschränkt sich das ZTDQF nur auf eine regelbasierte Validierung.

Zan und Zhang [23] weisen darauf hin, dass das Hauptproblem bei medizinischen Daten die geringe Genauigkeit ist. Um dieses Problem zu lösen, verwenden sie eine Datenqualitätsanalyse, die aus vier Schritten besteht. Zunächst wird eine Vertrauensanalyse der Daten durchgeführt und nicht vertrauenswürdige Daten werden entfernt.
Dies ist ein wesentlicher Unterschied zur vorliegenden Dissertation, da hier die Entfernung nicht vertrauenswürdiger Daten erst durch die datenempfangende Instanz und nicht bereits bei der Datenvorverarbeitung erfolgt. Im Anschluss an die Vertrauensanalyse und die Filterung erfolgt eine umfassende Bewertung der Daten mit Hilfe des Analytic Hierarchy Process (AHP).

Foidl et al. [24] definieren den Data Smell in Anlehnung an den Code Smell als *"... context-independent, data value-based indications of latent data quality issues caused by poor practices that may lead to problems in the future"*. Der Data Smell wird in drei Kategorien unterteilt: *a*) den Glaubwürdigkeits-Smell, *b*) den Verständlichkeits-Smell und *c*) den Konsistenz-Smell. Die drei Kategorien werden in weitere 36 Arten von Data Smells unterteilt. Neben den verschiedenen Arten von Data Smells werden zwei Metriken definiert, die Data Smell Strength und die Data Smell Density. Diese werden verwendet, um die Wahrscheinlichkeit des Auftretens eines Fehlers und die relative Anzahl der gefundenen Fehler für ein definiertes Datenattribut anzugeben. Foidl et al. verwenden einen eigenen Ansatz, der aus einer Kombination eines regelbasierten und eines maschinellen Lernansatzes besteht. Sechs der 36 gegebenen Data Smells beziehen sich auf die Glaubwürdigkeit der gegebenen Daten, die restlichen 30 Data Smells beziehen sich hauptsächlich auf Encoding, Syntax und Konsistenz.
Der von Foidl et al. vorgestellte Ansatz eignet sich für eine Implementierung als Validierer im vorliegenden DDVN.

Sándor et al. [25] überwachen das Verhalten der Sensoren. Wenn sich bestimmte Sensoren merkwürdig verhalten, wird dies erkannt und ein Fachpersonal kann

entscheiden, welche weiteren Schritte zu unternehmen sind. So kann frühzeitig verhindert werden, dass bösartige Sensoren falsche Daten in das System einspeisen. Das DDVN kann um eine Verhaltensbeobachtung erweitert werden. Dadurch können neben korrumpierten Sensoren auch korrumpierte Validierer erkannt werden.

Neben der Erweiterung von Validierungsalgorithmen um externe Funktionalitäten gibt es eine Vielzahl von Ansätzen, die am Validierungsalgorithmus ansetzen und diesen mit mathematischen Funktionen möglichst robust gegen Angriffe aufbauen. Ein Problem hierbei ist, dass die Ansätze algorithmenspezifisch sind und daher nicht ohne Anpassung auf andere Algorithmen übertragen werden können. Im Folgenden werden einige bekannte Ansätze zur Erhöhung der Sicherheit bzw. Robustheit eines Algorithmus vorgestellt.

Blanchard et al. [26] betrachten ein stochastisches Gradientenverfahren, das auf n Knoten verteilt ist. Von diesen n Knoten sind bis zu f Knoten korrumpiert und verhalten sich abnormal. Bei Verwendung der bekanntesten Aggregationsfunktion (Mittelung über alle erhaltenen Ergebnisvektoren der Knoten) reicht bereits ein byzantinischer Knoten $f = 1$ aus, um die Konvergenz zum Optimum zu verhindern. Blanchard et al. führen dagegen zwei neue Aggregationsfunktionen (*Krum* und *Multi-Krum*) ein, die auch bei $2f + 2 < n$ korrupten Knoten eine Konvergenz zum Optimum erreichen.

Ein weiterer Ansatz zur Verbesserung der Sicherheit eines verteilten stochastischen Gradientenverfahrens wird von Yudong et al. [27] beschrieben. Die Ansätze von Blanchard et al. und Yudong et al. unterscheiden sich darin, dass im ersten Ansatz jedem Knoten der gesamte Datensatz zur Verfügung steht, während im zweiten Ansatz jedem Knoten nur ein Teil des gesamten Datensatzes zur Verfügung steht (basierend auf der Idee eines Federated Learning Ansatzes). Ausserdem unterscheiden sich die beiden Ansätze darin, dass das Ziel von [26] die kollektive Verbesserung der Kostenfunktion ist und das Ziel von [27] die Schätzung der optimalen Modellparameter über den wahren Vorhersagefehler.

Adversarial Attacks [28] sind eine grosse Gefahr bei der Vorhersage mittels maschineller Lernalgorithmen. Dabei wird versucht, durch kleine Änderungen in den Eingangsdaten eine falsche Vorhersage zu erzwingen. Die kleinen Änderungen an den Eingangsdaten werden von menschlichen Beobachtern meist nicht bemerkt.

Destillation ist eine Trainingsmethode für DNNs, bei der Wissen aus einem bestehenden neuronalen Netz in ein zweites neuronales Netz übertragen wird. Dadurch wird die Berechnungskomplexität von grossen DNNs auf kleinere DNNs übertragen. Die ursprüngliche Idee war, ein komplexes Neuronales Netz auf ressourcenbeschränkten Geräten wie Smartphones oder Tablets laufen zu lassen. Papernot et al. [29] überführten den Trainingsansatz der Destillation in einen defensiven Destillationsansatz, der die Klassifikatoren robuster gegenüber Störungen macht. Das destillierte Wissen des ersten DNNs wird verwendet, um die Amplitude des Gradienten des zweiten neuronalen Netzes abzuschwächen und

somit eine Glättung durchzuführen. Auf diese Weise wirken sich kleine Störungen nur geringfügig auf den Gradienten aus.

Ein ähnlicher Ansatz wird in [30] zur Erhöhung der Sicherheit eines neuronalen Netzes gegen Adversarial Attacks vorgestellt. Hierbei werden durch einen zusätzlichen Trainingsschritt nach dem eigentlichen Training starke Gradientenanpassungen der Klassifikationsfunktion bei kleinen Änderungen der Eingabedaten bestraft. Im Gegensatz zur Destillation wird kein zweites DNN zum Training verwendet.

Beim Feature Squeezing [31] bleibt das eigentliche neuronale Netz unverändert, lediglich die Eingabedaten werden für verschiedene Vorhersagen angepasst. Grundsätzlich werden drei Vorhersagen mit Neuronalen Netzen durchgeführt: 1) die normale Vorhersage $g(X_{all})$ mit allen Merkmalen X_{all}, und 2) zwei Vorhersagen $g(X_1), g(X_2)$ mit jeweils einer unterschiedlichen Untermenge X_1 und X_2 von Merkmalen. Die Vorhersageergebnisse $g(X_1), g(X_2)$ werden mittels der L_p-Norm auf ihre Abweichungen von $g(X_{all})$ untersucht. Weichen die beiden untersuchten Abweichungen zu stark voneinander ab, so liegt eine Adversarial Attack vor.

2.3. Verteilte Datenvalidierung

Shukal et al. [32] haben einen auf Blockchain und Diversität basierenden Mechanismus eingeführt, um die Robustheit des maschinellen Lernens zu verbessern. Unter Verwendung von Datenvariation und verschiedenen Algorithmen des maschinellen Lernens wurde eine Reihe von Vorhersageergebnissen berechnet. Die verschiedenen Vorhersageresultate werden mit Hilfe eines Konsensalgorithmus zu einem einzigen Ergebnis zusammengefasst. Die Forschung in dieser Dissertation verwendet keine Blockchain, hat kein Distributed Ledger als Basis und benötigt keinen Konsensmechanismus zur Entscheidungsfindung. Stattdessen unterstützt die Forschung in dieser Arbeit eine Vielzahl unterschiedlicher Validierungsalgorithmen und schützt diese mit verschiedenen Mechanismen. Zusätzlich wird die Datenqualität durch eine Validierung in der Datenvorverarbeitung bestimmt.

Liu et al. [33] beschreiben ein Framework zur Bestimmung der Datenqualität im Stromnetz bzw. in der Energieversorgung. Es wird darauf hingewiesen, dass das Stromnetz in China verschiedene Eigenschaften aufweist, die durch das Framework unterstützt werden. Zu diesen Eigenschaften gehören 1) eine mehrschichtige Struktur des Stromnetzes mit der Zentrale, dem nationalen Stromnetz, dem regionalen Stromnetz, dem städtischen Stromnetz usw., 2) eine hohe Skalierbarkeit, da viele Sensoren angeschlossen sind, 3) eine Vielzahl von unterschiedlichen Datenformaten, 4) eine Vielzahl von unterschiedlichen Datenquellen, 5) einige Informationsinseln, bei denen die Daten nur in akkumulierter Form weitergeleitet werden und 6) unterschiedliche Leistungsanforderungen für die verschiedenen Vorhersagen. Das Thema

Datenvorverarbeitung wird nur kurz im Zusammenhang mit der Datenvalidierung angesprochen. Hier wird die Datenqualität verwendet, um festzustellen, ob eine Datenbereinigung erforderlich ist. Durch die Verwendung einer verteilten Big Data Plattform für die Datenvalidierung wird die Robustheit im Vergleich zu anderen Validierungsframeworks verbessert.

Federated Learning [34, 35] ist ein Ansatz, bei dem ein gemeinsames maschinelles Lernmodell auf Basis von Trainingsvorgängen auf verschiedenen Geräten (Mobiltelefonen oder Tablets) berechnet wird. Dabei werden die Daten für die Trainingsvorgänge nicht geteilt, um den Datenschutz zu gewährleisten. Die einzelnen lokalen Trainingsschritte bzw. lokalen Updates für das Training werden mittels einer Aggregationsfunktion, dem sogenannten *FederatedAveraging*, zusammengeführt. *FederatedAveraging* kombiniert lokale stochastische Gradientenverfahren auf den verschiedenen Geräten mit einem Model Averaging auf einem Server, der ein gemeinsames Modell erzeugt.
Ein Federated-Learning-Ansatz kann zusammen mit DDVN implementiert werden. Dazu muss der Akkumulator definieren, welche Validierer für die lokalen stochastischen Gradientenverfahren ausgewählt werden, und das Gateway darf die Daten nur an die ausgewählten Validierer weiterleiten.

Ein Schema zur Datenintegritätsvalidierung in Multi-Cloud Umgebungen wird von Witanto et al. [36] eingeführt. Hierbei ist das Ziel, dass Daten die einer nicht vertrauenswürdigen Cloud Umgebung zur Verfügung gestellt werden, weiterhin auf der Cloud Umgebung im original vorhanden sind. Dadurch sollte sichergestellt werden, dass der Cloud Service Provider (CSP) keine Möglichkeit hat Fehler abzustreiten oder zu vertuschen. Mögliche Fehler sind der Verlust von Daten oder das Korrumpieren von Daten aufgrund eines Softwareproblems. Das Schema zur Datenintegritätsvalidierung basiert auf einer Blockchain in Verbindung mit Smart Contracts. Hierbei gibt es eine vertrauenswürdige Instanz, genannt Cloud Organizer (CO), diese ist für die Kommunikation mit den CSPs, den vertrauenswürdigen Verifiern und den datenbesitzenden Personen verantwortlich. Mittels Smart Contract werden die Teilnehmenden registriert sowie der wesentliche Ablauf der Integritätsvalidierung protokolliert.
Der Ansatz von Witanto et al. kann beim DDVN verwendet werden, um sicherzustellen, dass die originale Validierungslogik auf einem Validierer vorhanden ist. Hierbei kann aber nicht ohne weitere Ergänzung gewährleistet werden, dass diese Logik auch zum Einsatz kommt. Im Zusammenhang mit der Kontrolle der Integrität der Eingangsdaten beim DDVN ist der Ansatz nicht passend, da die originale Datenquelle als nicht vertrauenswürdig angesehen wird, weshalb auch eine Datenvalidierung vorgenommen werden muss.

2.4. Schutzmechanismen in verteilten Systemen

Eine Möglichkeit, ein verteiltes System zu schützen, besteht darin, einen Angriff zu erkennen und den angreifenden Knoten aus dem Netzwerk zu entfernen bzw. ihn auf dem Weg durch das Netzwerk zu umgehen. Zur Erkennung eines Angriffs kann ein Intrusion Detection System (IDS) eingesetzt werden. Für die Implementierung eines IDS gibt es nach Kumar et al. [37] vier verschiedene Ansätze: 1) anomaliebasiert, 2) signaturbasiert, 3) spezifikationsbasiert und 4) ein hybrides Verfahren. Beim anomaliebasierten Verfahren wird das aktuelle Systemverhalten mit dem Normalverhalten verglichen. Weichen die beiden Verhaltensweisen um mehr als eine definierte Schwelle voneinander ab, wird das Internet of Things (IoT) Gerät als verdächtig markiert. Wenn das ungewöhnliche Verhalten anhält, wird das Gerät als bösartig markiert und, wie oben beschrieben, von der aktuellen Kommunikation ausgeschlossen. Ein signaturbasierter IDS Ansatz erzeugt für bekannte Angriffe eine Signatur und speichert diese; beim Empfang einer oder mehrerer Nachrichten werden diese mit den bekannten Angriffssignaturen verglichen. Sind diese identisch, liegt ein Angriff vor und eine Anomalie wird erkannt. Der spezifikationsbasierte Anomalieerkennungsansatz verwendet Systemspezifikationen für die Erkennung. Hierfür gibt es eine Vielzahl von Implementierungen, eine Möglichkeit *a*) ist die Definition einer maximalen Anzahl von Anfragen, die ein Dienst erlaubt, wird diese überschritten, dann verhält sich das System abnormal [38], eine weitere Möglichkeit *b*) ist der Einsatz von Honeypots, die eingehende Nachrichten nach definierten Regeln analysieren und bei Fehlverhalten weitere Analysen durchführen [39]. Das hybride Verfahren kombiniert die drei oben genannten Verfahren auf der Basis von Anomalien, Signaturen und Spezifikationen.
Zudem ist zu beachten, dass das IDS auf drei verschiedene Arten implementiert werden kann: zentral auf bekannten zentralen Servern (CIDS), verteilt auf mehreren IoT Geräten, bei denen jedes Gerät am Anomalieerkennungsprozess beteiligt ist (DIDS), und ein hybrider Ansatz (HIDS), bei dem eine Mischung aus zentralen bekannten Servern und IoT Geräten bei der Anomalieerkennung zusammenarbeitet.
Ein IDS kann mit dem DDVN kombiniert werden, um ein Eindringen in das Netzwerk zu erkennen. Ferner ist es möglich, Validierungsknoten, die sich fehlerhaft verhalten, zu identifizieren und vom Validierungsprozess auszuschliessen.

Ein Mix Netzwerk [40] beschreibt ein Routing-Protokoll, das sicherstellt, dass eine gesendete Nachricht nur vom Sender S und vom Empfänger E gelesen werden kann. Ausserdem wird die Rückverfolgung der Kommunikation zwischen zwei Teilnehmern erschwert. Im Mix Netzwerk sind die öffentlichen Schlüssel des Senders k_s, des Empfängers k_e und der Proxies $P_1, P_2, ..., P_n$, auch Mixe genannt, $m_1, m_2, ..., m_n$ bekannt. Wenn ein S eine Nachricht N an E sendet, dann verschlüsselt S die Nachricht N mit dem öffentlichen Schlüssel k_e, $N' = enc(N, k_e)$. Damit ist sichergestellt, dass nur E N' entschlüsseln kann.

Zusätzlich wird der Weg, den die Nachricht durch das Netzwerk nimmt, für jeden Knoten verschlüsselt, d.h. das Tupel $M_{last} = (E, N')$ wird mit dem öffentlichen Schlüssel des letzten Mixes vor dem Empfänger $M'_{last} = enc(M_{last}, m_{last})$ verschlüsselt. Das Verfahren wird für jeden Knoten auf dem Pfad wiederholt, so dass $M_i = (P_{i+1}, enc(M_{i+1}, m_{i+1}))$ und $M'_i = enc(M_i, m_i)$ gilt. Beim Senden einer Nachricht wird die verschlüsselte Nachricht M'_0 an den ersten Proxy-Knoten P_0 im Netzwerk gesendet. Der Proxy-Knoten entschlüsselt die Nachricht mit seinem privaten Schlüssel p_0, $M_0 = dec(M'_0, p_0) = (P_1, M'_1)$. Damit kennt jeder Knoten seinen Folgeknoten, aber nur E weiss, dass kein weiterer Knoten folgt. Die Bestimmung von E ist somit ohne weitere Informationen nicht möglich, da kein Knoten die privaten Schlüssel der anderen Knoten kennt. Das Senden einer Antwortnachricht kann auf die gleiche Weise erfolgen.
Das DDVN hat im Prinzip kein Problem mit der Nachverfolgung von Nachrichten, da in den meisten Fällen der Validierer weiss, welcher Akkumulator die Validierung angefordert hat. Falls die Verhinderung der Nachrichtenverfolgung durch die Validierer dennoch erwünscht ist, so kann das DDVN mit dem Mix Netzwerk Ansatz kombiniert werden.

Die Blockchain [41, 42] entspricht einer Distributed-Ledger-Technologie, kurz DLT, und wird verwendet, um eine Buchführung digital nachzubilden. Grundsätzlich gibt es eine ständig wachsende Liste von zusammenhängenden Blöcken, die Daten enthalten. Jeder neue Block wird durch ein Konsensverfahren im Netzwerk bestimmt und mit einem kryptographischen Verfahren an die bestehende Liste von Blöcken angehängt. Dabei enthält der neue Block neben den neuen Daten auch eine Referenz auf den vorherigen Block und Metadaten (z.B. einen Zeitstempel).
Nach Omar et al. [43] sind die zwei wesentlichen Merkmale der Blockchain das Vertrauen (Trust) in das System und die dezentrale Architektur. Das Vertrauen wird durch *a*) den Konsensmechanismus, *b*) die Transparenz (Bereitstellung des Quellcodes, öffentliche Schnittstellen, etc.) und *c*) die Verwendung von kryptographischen Verfahren [44] gewährleistet. Eine der grössten Gefahren in Bezug auf das Vertrauen ist ein 51%-Angriff [45], bei dem ein Angriff 51% der Knoten übernimmt und damit immer die Mehrheit im Konsensverfahren besitzt. Dagegen wird unter anderem ein Zwischenprüfungsverfahren vorgeschlagen[46], das einen Knoten, der im ersten Schritt erfolgreich war, im zweiten Schritt zu einem weiteren Rechenschritt zwingt. Die dezentrale Architektur hat den Vorteil, dass sie resistenter gegen Ausfälle und DDoS-Angriffe ist.
Das DDVN verwendet kein Konsensverfahren zur Bestimmung des Validierungsergebnisses. Es obliegt dem Akkumulator selbst, eine geeignete Akkumulation der verschiedenen Validierungsergebnisse vorzunehmen. Darüber hinaus speichern die Validierer nicht die gesamte Historie der gesendeten Daten bzw. Transaktionen. Ein weiterer zentraler Unterschied zwischen Blockchain und DDVN ist die Transparenz. Ein Grossteil des Vertrauens der Blockchain basiert auf öffentlich einsehbaren Daten. Im Gegensatz dazu verhindert die DDVN bewusst, dass nicht jeder Validierer alle Daten erhält. Dadurch wird unter anderem

eine umfassende Industriespionage durch einen einzelnen korrumpierten Knoten verhindert.

Zwei zentrale Sicherheitsziele in verteilten Systemen sind Authentifizierung und Autorisierung. Um diese Sicherheitsziele zu gewährleisten, wird in der Regel OAuth 2 [47] in Verbindung mit OpenID Connect (OIDC) [48] verwendet. [49] OAuth 2 ist ein Autorisierungs-Framework, das verwendet wird, um einen eingeschränkten Zugriff auf eine oder mehrere Ressourcen zu erhalten. Grundsätzlich werden bei OAuth 2 drei verschiedene Anfragen an unterschiedliche Kommunikationsteilnehmer gestellt (siehe Abbildung 2.1).

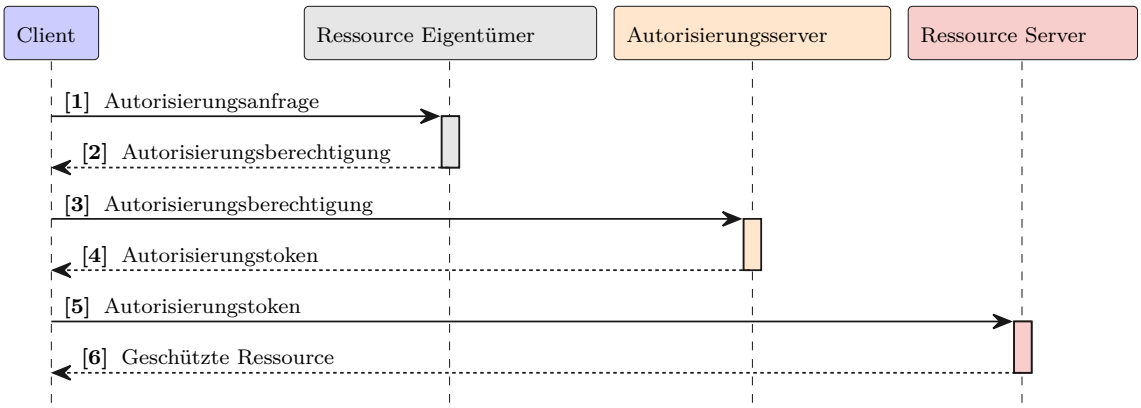


Abbildung 2.1.: Autorisierung der Ressource mittels OAuth 2

Der erste Schritt ist die Anfrage an den Eigentümer der Ressource bezüglich des Zugriffs auf die Ressource. Abhängig von der jeweiligen Implementierung wird die Ressourcenanfrage an den Eigentümer vom Autorisierungsserver als Teilschritt der zweiten Anfrage durchgeführt. Das Ergebnis der zweiten Anfrage enthält das Autorisierungstoken. Mit dem Token kann im letzten Schritt auf die geschützte Ressource zugegriffen werden. Es ist zu beachten, dass OAuth 2 selbst keine Verschlüsselung oder ähnliches durchführt. Aus diesem Grund ist eine Voraussetzung in der Spezifikation enthalten: *„Access token credentials MUST only be transmitted using TLS...“*. Ausserdem wird empfohlen, OAuth 2 zusammen mit Proof Key for Exchange (PKCE, ausgesprochen "Pixy") [50] zu verwenden. Damit wird verhindert, dass eine böswillige Anwendung trotz Autorisierung ein Access Token erhält.
OIDC ist eine Schicht über OAuth 2, die die Authentifizierung durchführt. Damit kann die Identität eines Endnutzers bzw. einer Endnutzerin validiert und allgemeine Informationen über die Identität abgefragt werden.
Im Gegensatz zu OAuth 2 wird beim DDVN die Autorisierung eines Validierers über den Akkumulator genehmigt. Dieser entspricht in diesem Fall einem Ressourcenbesitzer und dem Autorisierungsserver. Wenn der Akkumulator Zugriff von einem bestimmten Validierer möchte, teilt er dies dem Gateway mit. Das Gateway leitet die Daten dann an die autorisierten Validierer weiter. Ein Validierer weiss also grundsätzlich nicht, welche Ressourcen ihm zur Verfügung stehen.

Um die Verfügbarkeit der Daten in einem verteilten Netzwerk zu erhöhen, werden die Daten auf mehrere Instanzen repliziert. Das Schreiben und Lesen der Daten auf die verschiedenen Instanzen kann mit Hilfe von Quorum Protokollen [51] durchgeführt werden. Angenommen die Gesamtzahl der Instanzen ist n, die Anzahl der zu schreibenden Instanzen ist w und die Anzahl der zu lesenden Instanzen ist r, dann muss $r + w > n$ sein. Damit ist sichergestellt, dass mindestens eine Instanz den aktuellen Datenwert zurückliefert und auch bei Ausfall von Instanzen die Verfügbarkeit gewährleistet ist.
Die vorliegende Dissertation verwendet ebenfalls einen Quorum-Ansatz, um sicherzustellen, dass die Verfügbarkeit und Integrität von Validierungsergebnissen auch dann noch gegeben ist, wenn einzelne Knoten korrumpiert sind.

3. Industrielle Revolution und Industrielle Cybersicherheit

Im folgenden Kapitel werden die vier industriellen Revolutionen [52] kurz beschrieben. Damit wird ein Überblick über den industriellen Fortschritt gegeben und die Konsequenzen für die industrielle Sicherheit aufgezeigt. Im zweiten Teil des Kapitels wird die industrielle Cybersicherheit mit ihren Teilbereichen und Sicherheitszielen behandelt. Zunächst werden die Sicherheitsziele aufgezeigt, mögliche Angriffe erläutert und schliesslich der zentrale Punkt dieser Arbeit, die sichere Datenvorverarbeitung, behandelt.

3.1. Industrielle Revolution

Eine industrielle Revolution ist nicht das Ergebnis eines bestimmten Ereignisses oder Prozesses. Vielmehr handelt es sich um eine Anhäufung bestimmter identifizierbarer Veränderungen in den Methoden und Merkmalen der wirtschaftlichen Organisation, die zu einer wirtschaftlichen Entwicklung führen. Bisher hat es vier industrielle Revolutionen gegeben, die jeweils auf der vorhergehenden aufbauten. Eine chronologische Darstellung der industriellen Revolutionen findet sich in Abbildung 3.1.

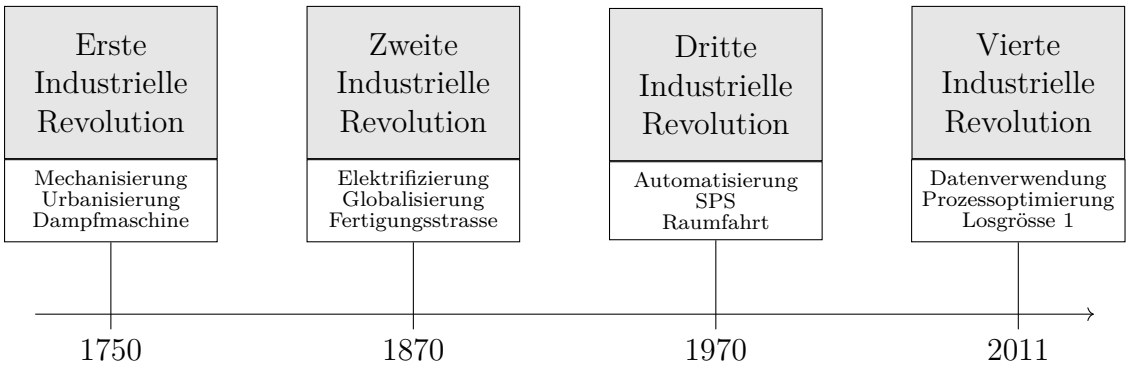


Abbildung 3.1.: Industrielle Revolution von 1700 bis heute (2023)

Die erste industrielle Revolution fand in England statt. Über den genauen Zeitpunkt der Industriellen Revolution gibt es zwei unterschiedliche Meinungen,

eine von Toynbee [53] und eine von Nef [54]. Toynbee datiert den Beginn der Industriellen Revolution auf das Jahr 1760, während Nef die Industrielle Revolution als einen langen Prozess beschreibt, der bis in die Mitte des 16. Jahrhunderts zurückreicht und gegen Ende des 19. Jahrhunderts zum endgültigen Triumph des Industriestaates führte, und nicht als ein plötzliches Phänomen, das mit dem Ende des 18. und dem Beginn des 19. Jahrhunderts verbunden ist. Aufgrund der unterschiedlichen Auffassungen über den tatsächlichen Zeitpunkt wurde in der obigen Abbildung die Mitte des 17. Jahrhunderts gewählt.
Entscheidend für die erste industrielle Revolution war die Mechanisierung, die die Landwirtschaft als Kern der Wirtschaft ablöste. Die Erfindung der Dampfmaschine [55] ersetzte den Einsatz von Tieren und Menschen und führte zu einer gleichmässigeren und effizienteren Arbeitsweise. Neben der Dampfmaschine gab es noch andere Faktoren, die zur ersten industriellen Revolution beitrugen, wie z.B. die Urbanisierung, der Einsatz von Finanzmitteln anstelle von menschlicher Arbeitskraft usw.

Die Entdeckung und Nutzung der Elektrizität und die Fliessbandproduktion markieren den Beginn der zweiten industriellen Revolution. Einer der bekanntesten Vertreter der zweiten industriellen Revolution ist Henry Ford, der die Fliessbandproduktion [56] für die Herstellung von Motoren einführte. Neben der Massenproduktion wurden auch erste Schritte in Richtung Globalisierung unternommen. Die Entwicklung des Automobils und die Verbesserung der Schifffahrt und der Luftfahrt waren Schlüsselfaktoren in diesem Prozess. Dadurch konnten Waren nicht mehr nur lokal, sondern weltweit transportiert werden. Neben dem Transport von Gütern kamen auch neue Kommunikationsmittel wie das Telefon zum Einsatz.

Die kontinuierliche Fliessbandarbeit wird durch voll- oder teilautomatisierte Prozesse ersetzt, was den Beginn der dritten industriellen Revolution markiert. Bestehende Prozesse, die menschliches Eingreifen erfordern, werden, wo immer möglich, durch Roboter ersetzt oder unterstützt. Speicherprogrammierbare Steuerungen (SPS) und Computer sind die wichtigsten Werkzeuge der Automatisierung. Hinzu kommen die ersten bemannten Weltraummissionen, wie die erste bemannte Mondlandung 1969 mit den Astronauten Neil Armstrong, Edwin Aldrin und Michael Collins (Apollo 11). Bemerkenswert ist, dass das Budget der NASA bereits drei Jahre (1966) vor der ersten bemannten Mondlandung gekürzt worden war und in den Folgejahren weiter gekürzt wurde. 1965 waren es 34 Milliarden und 1969 nur noch 23 Milliarden [57]. Dies hatte zur Folge, dass die Produktionsprozesse in der Raumfahrt optimiert werden mussten. Eine ähnliche Entwicklung war in vielen anderen Wirtschaftszweigen zu beobachten. Ein weiteres wichtiges Forschungsfeld der dritten industriellen Revolution war die Biotechnologie.

Industrie 4.0, manchmal auch als vierte industrielle Revolution bezeichnet, wurde 2011 auf der Hannover Messe als Zukunftsprojekt im Rahmen der Hightech-Strategie der Bundesregierung vorgestellt. Dabei geht es um die

umfassende Digitalisierung der industriellen Produktion. Dazu gehören *a*) die Vernetzung von Sensoren, Maschinen und Menschen über das Internet der Dinge, *b*) die Nutzung neuer Informationen, z.B. Sensordaten, zur Prozessoptimierung, *c*) die Unterstützung des Menschen bei anspruchsvollen Aufgaben, wie dem Erkennen schlechter Schweissnähte, *d*) die Automatisierung dezentraler Entscheidungen, so dass Fachkräfte nur noch in Ausnahmefällen benötigt werden und *e*) die Flexibilisierung der Produktion, so dass eine Losgrösse von eins ohne negative Auswirkungen erreicht werden kann.

Wir befinden uns mitten in der vierten industriellen Revolution. Die Fortschritte bei der Digitalisierung der Produktion sind enorm und konzentrieren sich meist auf die funktionale Erweiterung von Maschinen und Produktionssystemen. Derzeit liegt der Schwerpunkt auf der Vorhersage von Wartungsaufträgen und der Zustandsüberwachung von Maschinen, da für diese Anwendungsfälle der Mehrwert und das daraus resultierende Geschäftsmodell leicht zu realisieren sind. Ein unerwarteter Produktionsausfall kostet Zehntausende bis Hunderttausende von Euro, so dass die Anschaffung einer Wartungsvorhersage, die einen möglichen Ausfall frühzeitig erkennt, wirtschaftliche Vorteile bietet. Weitere wichtige Funktionen für Maschinenbesitzende sind die Bestimmung der Qualität der produzierten Teile sowie die optimale Einplanung von Produktionsaufträgen auf der Maschine und damit die Vermeidung von Stillstandszeiten (Zero-Downtime).

Im starken Kontrast zu den Anwendungsfällen mit offensichtlichem Mehrwert für den Kunden steht das Thema der industriellen Sicherheit. Auf den ersten Blick gibt es hier meist keinen Mehrwert und damit auch kein entsprechendes Geschäftsmodell. Erst bei genauerer Betrachtung wird der Nutzen und die Notwendigkeit von Industrial Cybersecurity deutlich. Ungeschützte Systeme ermöglichen Industriespionage und damit den Verlust von Betriebsgeheimnissen und Firmen Know-how. In der Folge verliert ein Unternehmen an Wettbewerbsfähigkeit. Andere Szenarien zeigen den Verlust von Kundendaten und daraus resultierende Konkurrenzangebote sowie Imageschäden. Auch können ungeschützte Verbindungen zu einer Maschine ausgenutzt werden, so dass diese durch eine absichtliche Fehlkonfiguration zum Ausfall gebracht wird. Diese Szenarien zeigen, wie wichtig industrielle Cybersicherheit ist und warum sie nicht am Ende der Wertschöpfungskette stehen darf.

3.2. Industrielle Cybersicherheit

Die klassische industrielle Sicherheit, auch funktionale Sicherheit (engl. Safety) genannt, befasst sich mit dem Schutz des Menschen und seiner Umwelt. Oberstes Ziel ist die Vermeidung von Unfällen und damit die Vermeidung von Personenschäden. Mögliche körperliche Verletzungen sind das Einklemmen oder Abschneiden von Körperteilen an Pressen oder der Verlust der Sehkraft bei der Arbeit mit Lasern. Um diese und andere Risiken zu vermeiden, unterzeichnete der

damalige US-Präsident Richard Nixon 1970 den Occupational Safety and Health Act (OSH Act) [58]. Das Gesetz stellt sicher, dass Arbeitnehmende in einem Umfeld arbeiten, das ihre Gesundheit und Sicherheit nicht gefährdet. Der OSH Act war somit der Beginn des gesetzlich geregelten und zertifizierbaren Arbeitsschutzes. Seitdem sind insbesondere im industriellen Bereich einige neue Zertifizierungen entstanden, wie die *IEC 61508* (Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme) [59], die *IEC 61511* (Sicherheitstechnische Systeme für die Prozessindustrie) [60] und die *IEC 62061* (Sicherheit von Maschinen - Funktionale Sicherheit sicherheitsbezogener elektrischer, elektronischer und programmierbarer elektronischer Steuerungssysteme) [61]. Sie klassifizieren den relativen Grad der Risikominderung mit Hilfe des Safety Integrity Levels (SIL). Je höher das SIL-Niveau, desto geringer ist die Wahrscheinlichkeit, dass das System die geforderten Sicherheitsfunktionen nicht erfüllen kann. Kurz gesagt ist industrielle Sicherheit die Eigenschaft, dass eine realisierte Ist-Funktion mit der entsprechenden spezifizierten Soll-Funktion übereinstimmt.

Cybersicherheit (engl. Security oder Cybersecurity) befasst sich laut BSI mit *"... allen Aspekten der Sicherheit in der Informations- und Kommunikationstechnik. Das Handlungsfeld der Informationssicherheit wird dabei auf den gesamten Cyberspace ausgeweitet. Dieser umfasst die gesamte mit dem Internet und vergleichbaren Netzen verbundene Informationstechnik einschliesslich der darauf basierenden Kommunikation, Anwendungen, Prozesse und verarbeiteten Informationen. Häufig wird bei der Betrachtung der Cybersicherheit auch ein spezieller Fokus auf Angriffe aus dem Cyberspace gelegt."* [62]
Im Vergleich zur funktionalen Sicherheit befasst sich die industrielle Cybersicherheit mit der Informations- und Kommunikationssicherheit. Die funktionale Sicherheit kann, muss aber nicht durch eine Erhöhung der Cybersicherheit verbessert werden. Daher werden diese beiden Sicherheitsbereiche grundsätzlich getrennt betrachtet. Je nach Anwendungsfall können sich jedoch aus den Anforderungen an die funktionale Sicherheit auch Anforderungen an die Cyber-Sicherheit ergeben. Ist beispielsweise in einer Fertigungsanlage ein Laser integriert, so ist eine mögliche Anforderung an die Cybersicherheit, dass das Einspielen böswilliger Steuersignale verhindert werden muss. Dies bedeutet, dass der Laser während der Anwesenheit eines Menschen nicht vorsätzlich eingeschaltet werden kann. Eine mögliche Verletzung des Menschen kann so durch Methoden der Cybersicherheit verhindert werden.
Die Cybersicherheit wurde bisher weniger beachtet als die funktionalen Sicherheit. Das liegt daran, dass Cybersicherheit erst mit der fortschreitenden Digitalisierung im Produktionsumfeld an Bedeutung gewonnen hat. Zuvor war IT-Sicherheit eine Domäne der Informationstechnologie und von den Produktionsumgebungen abgekoppelt. Durch die weitreichende Vernetzung in industriellen Umgebungen ist es Angreifenden möglich, netzwerkübergreifend in Automatisierungs- und Steuerungssysteme einzudringen und diese zu schädigen. Dabei ist nicht nur das

Netzwerk oder die Maschine gefährdet, sondern auch der Mensch, der an oder mit der Maschine arbeitet. Für die IT-Sicherheit im industriellen Umfeld gibt es im Wesentlichen die drei Sicherheitsziele Vertraulichkeit, Integrität und Verfügbarkeit (kurz CIA). Darüber hinaus gibt es weitere Sicherheitsziele, die im Zuge der Digitalisierung an Bedeutung gewonnen haben. Eine detaillierte Erläuterung der verschiedenen Sicherheitsziele findet sich unter Abschnitt 7.2.

In der vorliegenden Arbeit werden industrielle Cybersicherheit, Cybersicherheit und Sicherheit synonym verwendet. Der grundlegende Unterschied zwischen Industrieller Cybersicherheit und Cybersicherheit besteht darin, dass Industrielle Cybersicherheit ein Teilbereich von Cybersicherheit ist. Die Begriffe funktionale Sicherheit und Arbeitssicherheit werden, wenn sie gemeint sind, explizit genannt.

Im Vergleich zur funktionalen Sicherheit ist die Cybersicherheit bisher nur durch wenige Normen, Standards und Richtlinien definiert. Eine Übersicht der zentralen Normen und Standards [63] wurde von Herrn Niemann in Zusammenarbeit mit der ABB AG zusammengestellt (siehe Tabelle 3.1).

Tabelle 3.1.: Normen und Standards für die Cybersicherheit [63]

Normen Standards Richtlinien	**Hersteller-vereinigungen Behörden**	**Gesetze Verordnungen**
• ISO/IEC 27001 • NIST SP 800-53 • NIST SP 800-82 • NAMUR NA 115, NE 153 • ISO/IEC 15408: Common Criteria • IEC 62443 • VDI/VDE 62351 • Vds 10000	• PROFINET Security Guideline • Securing EtherNet/IP Networks • BSI: Industrial Control System Security und andere • BSI: Standards 200-1, 200-2, 200-3, 200-4 • SANS - Critical Controls for Effective Cyber Defense • Homeland Security / ICS-CERT	• IT-Sicherheitsgesetz • Verordnung zur Bestimmung Kritischer Infrastrukturen dem BSI Gesetz (BSI-KritisV) • Gesetz über die Elektrizitäts- und Gasversorgung (EnWG) • Sicherheitskatalog gem. §11 Abs. 1a EnWG

3.3. Bedrohungslage

Durch den Einsatz von Cybersicherheitsmethoden, wie z.B. Antivirensoftware, Passwortrichtlinien, Datenverschlüsselung etc. kann ein Teil der Cyberangriffe

verhindert werden. Eine Statistik [64] des Centraal Bureau voor de Statistiek zeigt, dass in den Niederlanden durchschnittlich 97% der produzierenden Unternehmen bereits Antiviren-Software einsetzen. Passwortrichtlinien werden dagegen nur in 72% der Unternehmen eingesetzt. Ein Vergleich der Produktionsunternehmen in den Niederlanden mit branchenunabhängigen Unternehmen in Singapur [65] zeigt, dass es zudem grosse Unterschiede bei den Cybersicherheitsmassnahmen je nach Standort gibt (siehe Tabelle 3.2).

Tabelle 3.2.: Vergleich der implementierten Cybersicherheitsmethoden zwischen Produktionsunternehmen in den Niederlanden und branchenunabhängigen Unternehmen in Singapur

Sicherheitsmethode	**Niederlande**	**Singapore**
Antiviren-Software	97%	93%
Daten physisch extern gelagert	87%	47%
Daten verschlüsselt übertragen	29%	47%
Netzwerkzugriffskontrolle	49%	44 %
Zugriffskontrolle für Geräte und Software	39%	51%

Trotz der oben genannten Cybersicherheitsmassnahmen hat die Zahl der Cyberangriffe in den letzten 15 Jahren stark zugenommen. So sind beispielsweise in Polen die dem Computer Emergency Response Team (CERT) gemeldeten Cyberangriffe [66] von 50 Meldungen im Jahr 1996 auf 10.420 Meldungen im Jahr 2020 um das 200-fache gestiegen (siehe Abbildung 3.2). In Amerika hat sich die Zahl der Datenlecks innerhalb von 15 Jahren versechsfacht [67]. Die Anzahl der offen gelegten Daten korreliert dabei wenig, da ein einzelnes Datenleck enorme Auswirkungen haben kann (siehe Abbildung 3.3). Die oben genannten Statistiken beziehen sich auf Cyberangriffe, die nicht nur auf produzierende Unternehmen abzielen. Dennoch ist ein genereller Trend zu einer steigenden Anzahl von Angriffen erkennbar. Zudem ist das verarbeitende Gewerbe mit 23.2% [68] die Branche mit dem höchsten Anteil an Cyberangriffen.

Die durch Cyberangriffe verursachten Kosten sind für deutsche Unternehmen im internationalen Vergleich am höchsten. Für deutsche Unternehmen liegen die Kosten bei 906.000 Dollar, für französische Unternehmen bei 110.000 Dollar. Dies entspricht einem Faktor von 8.2. Darüber hinaus ist auch der höchste gemeldete Schaden mit 48 Millionen Dollar bei einem deutschen Unternehmen zu verzeichnen. Die Kosten sind zudem abhängig von der Unternehmensgrösse, so muss ein kleines Unternehmen (1 − 49 Mitarbeiter) mit Kosten von 14.000 Dollar rechnen und ein Grossunternehmen (1.000+ Mitarbeiter) mit Kosten von 551.000 Dollar (siehe Abbildung 3.4) [69].

Ein weiterer Indikator für Cybersicherheit ist die Anzahl der Common

Vulnerabilities and Exposures (CVEs) [70], die sich in den letzten 20 Jahren verzwanzigfacht hat (siehe Abbildung 3.5).

Die vorliegenden Statistiken zeigen, dass die Cybersicherheit an Bedeutung gewonnen hat und diese Arbeit von zentraler Bedeutung ist. Ausserdem muss in Zukunft mehr Forschung in diesem Bereich betrieben werden, um der steigenden Anzahl von Cyberangriffen vorzubeugen. Es zeigt sich auch, dass die Analyse des Systems in Bezug auf die Sicherheit nicht vernachlässigt werden darf und bei gefundenen Schwachstellen eine Mitigation durchgeführt werden muss.

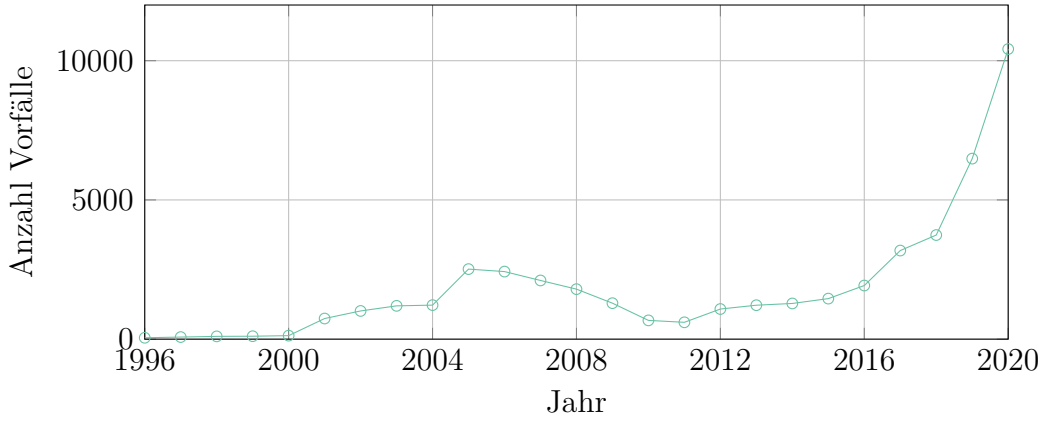


Abbildung 3.2.: Anzahl der an das CERT Polen gemeldeten Cyberangriffe

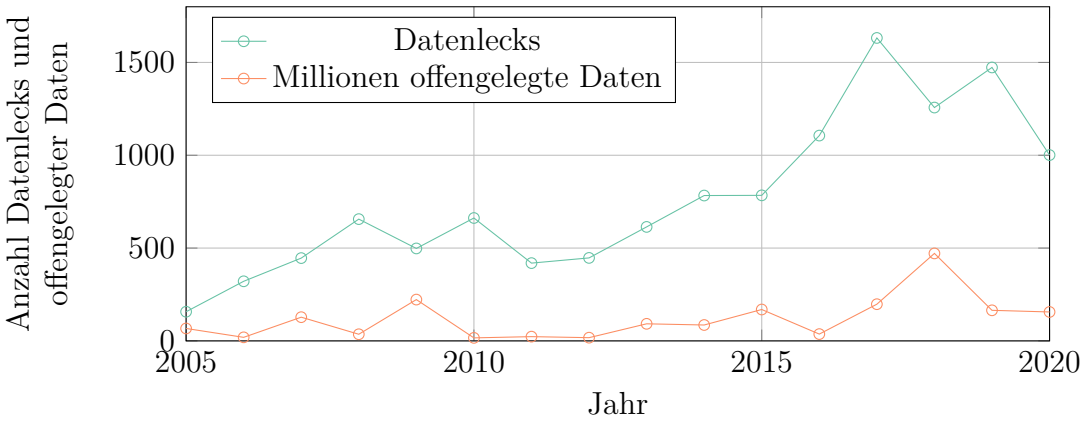


Abbildung 3.3.: Anzahl der gemeldeten Datenlecks und Menge der offengelegten Daten für Amerika

Eine zeitliche Zusammenfassung verschiedener Cyberangriffe auf unterschiedliche Sektoren wurde von Miller et al. [71] erstellt. Neben einer kurzen Beschreibung des Angriffs werden zusätzlich das Jahr des Angriffs, der erste Zugriff, die angreifende Instanz, die Branche und die Auswirkungen aufgelistet (siehe Tabelle A.1).

Nachfolgend sind einige abgeleitete Informationen aufgelistet. Unter anderem zeigt Tabelle 3.3, dass der Bereich Energie am häufigsten von Angriffen betroffen ist.

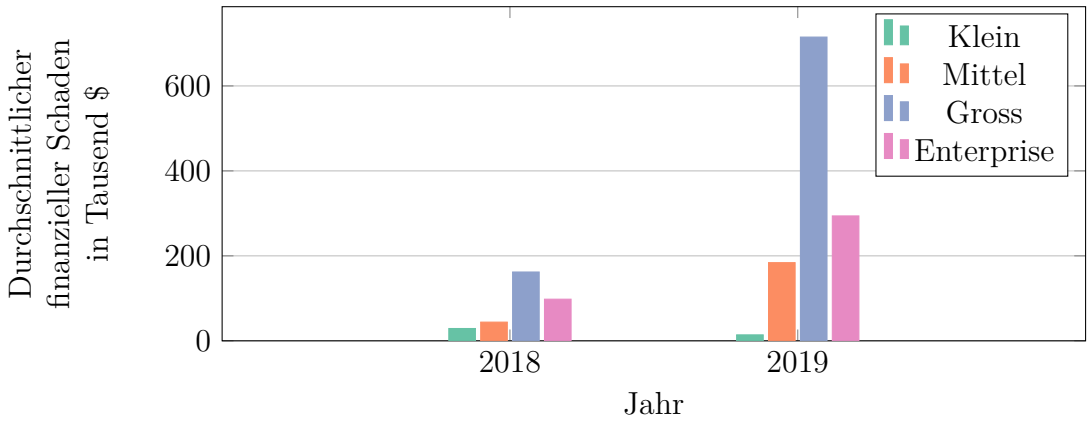


Abbildung 3.4.: Durchschnittlicher finanzieller Schaden pro Unternehmen aufgrund von Cyberangriffen (Kleines Unternehmen: $1-49$ Mitarbeiter, Mittleres Unternehmen: $50-249$ Mitarbeiter, Grosses Unternehmen: $250-999$ Mitarbeiter, Enterprise: >1000 Mitarbeiter)

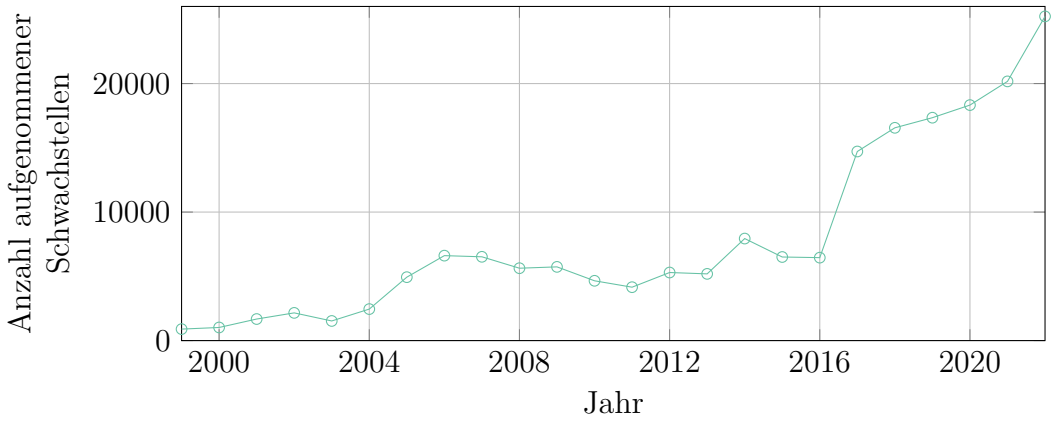


Abbildung 3.5.: Anzahl der aufgenommen Common Vulnerabilities and Exposures (CVEs) pro Jahr

Von den 43 aufgelisteten Angriffen wurde dieser in 23% der Fälle angegriffen, d.h. in jedem vierten Fall. Die Fertigung sowie der Bereich Chemie sind in 19% der Fälle betroffen, das entspricht jedem fünften Angriff. Die Bereiche Gesundheit, Lebensmittel sowie der unbekannte Anteil sind in 5% der Fälle das Angriffsziel.

Tabelle 3.3.: Anzahl Angriffe pro Sektor

Sektor	Anzahl Angriffe	Relative Anzahl
Energie	10	23%
Chemie	8	19%
Fertigung	8	19%

Tabelle 3.3.: Anzahl Angriffe pro Sektor

Sektor	Anzahl Angriffe	Relative Anzahl
Transport	6	14%
Wasser	5	12%
Kernenergie	4	9%
Gesundheit	2	5%
Lebensmittel	2	5%
Unbekannt	2	5%

Bei der Betrachtung der angreifenden Instanzen zeigt sich, dass bei 26% der Angriffe eine organisierte Gruppe hinter dem Angriff steht. Dabei können sowohl finanzielle, aktivistische als auch politische Motive eine zentrale Rolle spielen. Nur in 6 von 43 Fällen wurde der Angriff von einem internen Angreifenden durchgeführt. Insider-Angriffe sind daher unwahrscheinlicher als Angriffe durch externe Gruppen, weshalb externe Zugänge zu Systemen stark gesichert werden sollten.

Tabelle 3.4.: Arten von angreifenden Instanzen

Angreifende Instanz	Anzahl Angriffe	Relative Anzahl
Organisierte Gruppe	11	26%
Einzelperson	10	23%
Staat	8	19%
Unbekannt	8	19%
Interne Person	6	14%
Liefernde Person	1	2%

Zusätzlich wird in Tabelle 3.5 angegeben, wie sich die angreifende Person Zugang zum System verschafft hat. Die häufigsten Angriffsarten sind Workstation Access und Spear Phishing (23%). Unter Spear Phishing wird das absichtliche Versenden von gefälschten E-Mails verstanden, mit dem Ziel, direkt an geheime Daten zu gelangen oder das Opfer dazu zu verleiten, einen korrumpierten Link aufzurufen. Im Vergleich zum herkömmlichen Phishing wird beim Spear Phishing in der Regel die absendende Person gefälscht, um den Eindruck zu erwecken, dass die E-Mail von einem oder einer Vorgesetzten stammt.

Tabelle 3.5.: Zugriffe bei einem Angriff

Zugriff	Anzahl Zugriffe	Relative Anzahl
Arbeitsplatz	10	23%
Spear Phishing	10	23%
Gerät mit Internetanbindung	8	19%
Unbekannt	5	12%
Replikation durch Wechseldatenträger	4	9%
Externer Remote Service	3	7%
Ausnutzung einer öffentlichen Anwendung	1	2%
Vertrauensverhältnis	1	2%
Wireless	1	2%

Ein erfolgreicher Angriff führt zu 42% Produktivitäts- und Umsatzeinbussen (siehe Tabelle 3.6). Zu den bekanntesten Angriffen dieser Art zählt die Ransomware NotPetya [72]. Schätzungen zufolge hat NotPetya einen Gesamtschaden von 10 Milliarden Dollar verursacht. Die Unternehmen Merck KGaA (870 Millionen Dollar), FedEx (400 Millionen Dollar), Saint-Gobain (384 Millionen Dollar) und Maersk (300 Millionen Dollar) erlitten laut Hochrechnung die grössten Schäden. Der Angriff selbst verschlüsselte die Daten der betroffenen Unternehmen und machte sie unbrauchbar, bis eine Zahlung bei den Erpressenden eingegangen war. Danach wird der Entschlüsselungscode freigegeben.

Das explizite Löschen von Daten (14%) und die Beschädigung von Eigentum (9%) kamen seltener vor.

Tabelle 3.6.: Auswirkungen der Angriffe

Auswirkung	Anzahl Auswirkungen	Relative Anzahl
Produktivitätsverlust	18	42%
Umsatzverlust	18	42%
Festplattenlöschung	6	14%
Verlust der Sicherheit	6	14%

Tabelle 3.6.: Auswirkungen der Angriffe

Auswirkung	**Anzahl Auswirkungen**	**Relative Anzahl**
Diebstahl von Betriebsdaten	5	12%
Nicht veröffentlicht	5	12%
Verlust der Erreichbarkeit	5	12%
Schaden am Eigentum	4	9%
Verlust der Kontrolle	4	9%
Manipulation der Steuerung	2	5%

4. Datenqualität und Datenvorverarbeitung

Das folgende Kapitel definiert den Begriff Daten und leitet daraus die Datenqualität ab, zeigt die wesentlichen Metriken zur Bestimmung der Datenqualität auf und beschreibt den Datenvalidierungsprozess als Teil der Datenvorverarbeitung bzw. als Datenvorverarbeitung selbst.
Ein Teil der nachfolgenden Inhalte ist in komprimierter Form bereits in den Papern [73, 74, 75] publiziert.

4.1. Datenqualität

Für Anwendungsfälle im industriellen Umfeld ist die Datenqualität von entscheidender Bedeutung. Um auf die Datenqualität näher eingehen zu können, werden zunächst die Grundlagen geschaffen. Dazu ist es notwendig, die beiden Begriffe Daten und Informationen zu klären. Eine der bekanntesten Definitionen des Begriffs Daten findet sich in *Information systems: Theory and practice* von Burch et al:

„Data are language, mathematical, or other symbolic surrogates which are generally agreed upon to represent people, objects, events and concepts...“[76]

Diese Definition hat nach Fox et al. [77] den Nachteil, dass ein bestimmtes Datum nur in einer bestimmten Form vorliegen kann, was in praktischen Anwendungen meist nicht der Fall ist, z.B. bei der Verwendung von JSON, XML und Apache Avro oder bei unterschiedlichen Datumsformaten wie dem amerikanischen Format *mm/dd/yyyy* oder dem europäischen Format *dd/mm/yyyy*. Daher ist eine strikte Trennung zwischen den konzeptionellen Aspekten und den Darstellungsaspekten von Daten zu bevorzugen. Codd [78] bezeichnet diese Sichtweise der Datenmodellierung als *data independence objective*. Eine weitere Sichtweise des Datenbegriffs kommt aus dem Bereich der Datenbankforschung. Hier definieren Tsichritzis und Lochovsky [79] ein Datum bzw. ein Datenelement als ein Tripel bestehend aus $\langle e, a, v \rangle$. e entspricht einer Entität, d.h. einem Typ bestehend aus einer Menge von Attributen a und dem gesetzten Wert v des jeweiligen Attributes. Ein Beispiel ist in Abbildung 4.1 dargestellt. Hier besteht eine Entität mit dem Namen Maschine aus den Attributen Seriennummer, Maschinentyp und

Herstellungsdatum sowie den zugehörigen gesetzten Werten 12345, *Stanze* und 07.07.2022.

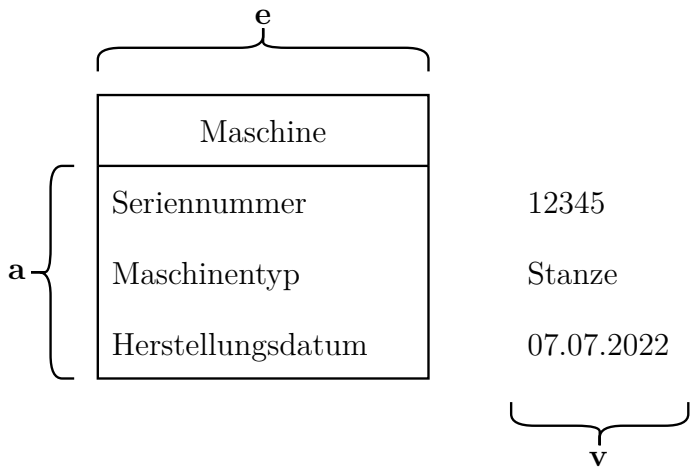


Abbildung 4.1.: Beispiel für die Darstellung eines Datenelements

Eine Menge von Datenelementen wird als Daten bezeichnet. Diese Definition hat noch einen entscheidenden Nachteil, es wird nach wie vor nicht zwischen den konzeptionellen Daten und der Speicherung bzw. Repräsentation der Daten unterschieden. An dieser Stelle greift die Ergänzung von Fox et al. [77] mit der Datenrepräsentation und dem Datensatz. Die Datenrepräsentation ist ein Regelwerk, das das Format der Daten auf einem Medium festlegt. Ein Eintrag von Daten auf einem Medium, der den Regeln der Datenrepräsentation entspricht, wird als Datensatz bezeichnet.

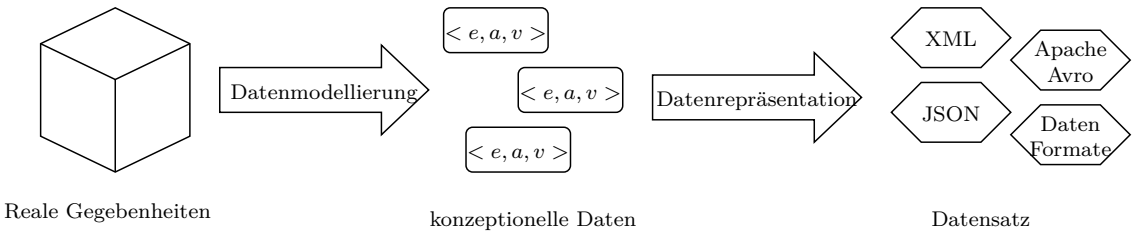


Abbildung 4.2.: Beispiel für die Darstellung eines Datenelements

Für die vorliegende wissenschaftliche Arbeit wird die erweiterte Definition von Fox et al. für den Begriff Daten verwendet. Um Verwechslungen zwischen Daten und Informationen zu vermeiden, wird im Folgenden der Begriff Information in Anlehnung an Burch et al. definiert:

„Information is the result of modelling, formatting, organizing or converting data in a way that increases the level of knowledge for its recipient.“ [76]

Die Definitionen von Daten und Informationen sind sehr allgemein gehalten und können auf verschiedenste Anwendungsfälle und Fachgebiete angewendet werden.

Übertragen auf Industrie 4.0 Szenarien entsprechen Daten meist Sensormesswerten, Auftragsdaten, Maschineneinstellungen oder Daten zu einem Fertigungsprozess. Die Verknüpfung der Daten untereinander, z.B. welcher Fertigungsauftrag auf welcher Maschine für welchen Kunden bearbeitet wird, entspricht der Information. Ein weiteres Beispiel für Information ist die Verknüpfung verschiedener Maschinendaten, um den Zustand einer Maschine zu ermitteln.

Bei der Datenqualität geht es darum, die Qualität der verfügbaren Daten anhand definierter Metriken zu bestimmen. Eine einfache Betrachtung der Datenqualität wurde von Jayawardene et al. wie folgt beschrieben: *„In its simplest form, data is a representation of objects or phenomena in the real world. Thus, when it comes to the discussion of quality of data, we can say that poor quality data is a result of poor representation of the real world.“* [80]. Eine schlechte Datenqualität ist also gleichbedeutend mit einer schlechten Abbildung der realen Welt. Um diese Aussage um eine quantitative Betrachtung zu erweitern, werden im Folgenden die zentralen Metriken zur Bewertung der Datenqualität aufgeführt und beschrieben.

4.2. Arten von Daten und die Darstellung der Datensätze

Die Datenrepräsentation konzeptioneller Daten auf einem Medium hat erheblichen Einfluss auf die Performance, die Verarbeitungsmöglichkeiten und die Verfügbarkeit. Um eine geeignete Repräsentation auf einem Medium zu finden, muss einerseits der Anwendungsfall, z.B. wird auf den Datensatz meist lesend oder schreibend zugegriffen, und andererseits die Art der konzeptionellen Daten berücksichtigt werden. Hier wird meist eine Einteilung in drei Gruppen *a*) strukturierte Daten, *b*) semistrukturierte Daten und *c*) unstrukturierte Daten vorgenommen. Eine grafische Einteilung der verschiedenen Typen wurde von Yan et al. [81] vorgenommen (siehe Abbildung Abbildung 4.3). Dabei ist zu beachten, dass für bestimmte Anwendungsfälle strukturierte Daten bevorzugt werden, weshalb die Daten mittels Semantic-Web-Technologien oder Target Tracing explizit in strukturierte Daten transformiert werden.

Im Folgenden wird eine Beschreibung der verschiedenen Datentypen in Anlehnung an Sidi et al. [82] und Batini et al. [83] gegeben:

Strukturierte Daten: Die Daten liegen in einer streng vorgegebenen Struktur für einen bestimmten Bereich vor. Die Datenmodellierung kann einem relationalen Datenbankmodell entsprechen. Bei konzeptionellen Daten besteht keine Flexibilität.

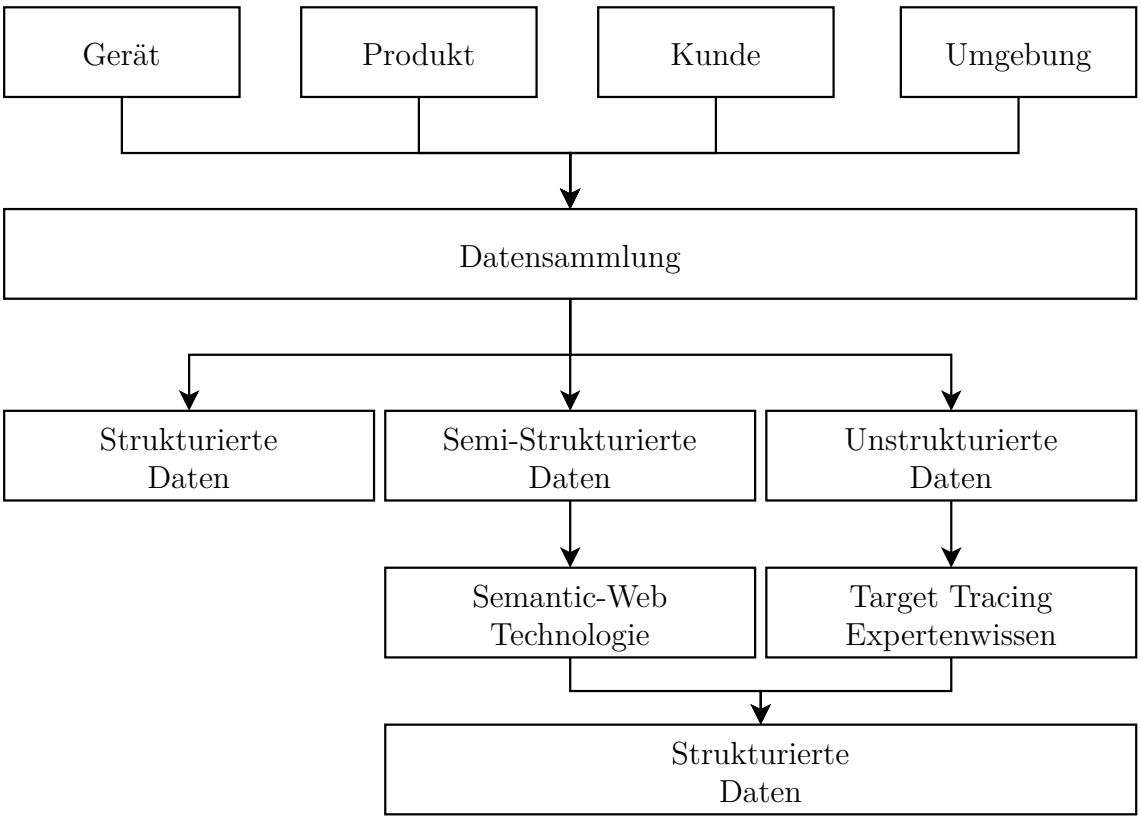


Abbildung 4.3.: Arten von Daten

Semi-Strukturierte Daten: Das Datenmodell besteht aus einem groben Schema und der Inhalt ist flexibel. Daraus ergeben sich unterschiedliche Interpretationsmöglichkeiten für bestimmte Daten, was die Verarbeitung erschweren kann.

Unstrukturierte Daten: Die Daten bestehen aus einer generischen Folge von Symbolen und sind in der überwiegenden Anzahl der Fälle in natürlicher Sprache verfasst. Eine Verarbeitung der Daten muss in der Regel in einem Kontext erfolgen, um eine sinnvolle Interpretation zu gewährleisten.

Tabelle 4.1 zeigt einige Beispiele zu den zuvor beschriebenen Datentypen. Dabei werden jeweils Anwendungsfälle im industriellen Umfeld sowie allgemein bekannte Anwendungsfälle beschrieben.

Tabelle 4.1.: Beispiele für die drei unterschiedlichen Arten von Daten

Art der Daten	Beispiel
Strukturierte Daten	*Industrie Umfeld:* Sensor Signale, Steuerung, etc. *Relationale Tabellen:* Das Modell ist vorgegeben und erlaubt keine andere Darstellung eines bestimmten Datums. *Statistische Daten:* Statistische Ergebnisse können nur dann aggregiert werden, wenn sie dieselbe Struktur aufweisen.
Semi-Strukturierte Daten	*Industrie Umfeld:* Kundenfeedback, das in einem definierten Format übermittelt wird. *Markup Sprachen:* XML, HTML, Markdown, etc.
Unstrukturierte Daten	*Industrie Umfeld:* Ton-, Bild- oder Videomaterial zur Erkennung eines Maschinenzustandes. *Freitext:* der schriftliche Inhalt einer E-Mail oder die ausdrückliche Beantwortung eines Fragebogens.

Für die Darstellung eines Datensatzes auf einem Medium gibt es eine Vielzahl von Möglichkeiten, die im industriellen Umfeld bekanntesten werden im Folgenden kurz erläutert. Dabei werden die bereits von Schmetz et al. [84] und Jirkovskỳ [85] untersuchten Datenformate aufgegriffen.

OPC UA: Open Platform Communication Unified Architecture (OPC UA) [86] ist ein Standard für den Datenaustausch in serviceorientierten Systemen. Dabei ist er eng mit den angebotenen Diensten und der verwendeten Kommunikationstechnologie verknüpft. Im Gegensatz zu seinem Vorgänger OPC erlaubt das Datenmodell eine Selbstbeschreibung durch die Integration einer Basisontologie und die Verwendung von Informationsmodellen. Diese Informationsmodelle erzeugen neue Ontologien, die auf spezielle Domänen innerhalb der Fertigung zugeschnitten sind [87].

AutomationML: Automation Markup Language (AutomationML) ist ein offenes, XML-basiertes Format, das die Vernetzung und den Datenaustausch zwischen verschiedenen Produktionswerkzeugen ermöglichen soll. Um dies zu erreichen, beschreibt AutomationML ein Top-Level-Format und eine Reihe von Unterformaten. Auf diese Weise integriert AutomationML verschiedene aufgabenspezifische Formate und Ontologien. Die allgemeine Topologie wird mit dem Computer Aided Engineering Exchange (CAEX) Format beschrieben,

während COLLADA (COLLAborative Design Activity) und PLCopen für geometrische, mechanische und logische Informationen verwendet werden [88].

B2MML und BatchML: Business to Manufacturing Markup Language (B2MML) und Batch Markup Language (BatchML) implementieren eine Reihe von internationalen Standards für die Verbindung zwischen den Unternehmen und den Steuerungssystemen. Für BatchML/B2MML-Dokumente werden allgemeine und *ISA-88/95* [89] Modelle verwendet, die durch zusätzliche Definitionen und Referenzen erweitert werden können. Beide Sprachen wurden zusammengeführt, um die Wiederverwendbarkeit von Modellen zu ermöglichen und gleichzeitig das Datenmodell vollständig kompatibel und erweiterbar zu halten [87].

MTConnect: MTConnect [90] ist ein offener XML-basierter Standard, der speziell für die Überwachung, Analyse und Konnektivität von Maschinendaten entwickelt wurde. Die Datendarstellung von MTConnect ist auf Werkzeugmaschinen ausgerichtet und bietet auch einige Erweiterungen für eine teilweise Interoperabilität mit OPC UA und B2MML. Die Spezifikation definiert allgemeine Konventionen und einige spezifische Modellierungskonventionen für Werkzeugmaschinen [91].

JSON-LD: JSON-Linked Data (JSON-LD) ist eine W3C Recommendation [92], die es ermöglicht, verknüpfte Daten in einem JSON-kompatiblen Format darzustellen. Entscheidend ist dabei, dass die verknüpften Daten von unterschiedlichen Maschinen in unterschiedlichen Umgebungen interpretiert und selbständig aufgelöst werden können, d.h. für eine Verknüpfung zu anderen Daten wird ein eingebetteter Link verwendet. Wenn die verknüpften Daten benötigt werden, kann die Maschine die Daten nur aufgrund des gegebenen Links nachladen.

4.3. Metriken für die Datenqualität

Die Metriken zur Bestimmung der Datenqualität können nach Wang und Strong [93] in vier Kategorien eingeteilt werden. Die Kategorien sind in Abbildung 4.4 aufgelistet, zusammen mit einigen Beispielen von Metriken für jede Kategorie. Eine Zusammenfassung der vier Kategorien wurde von Karkouch et al. [94] wie folgt vorgenommen:

- *Intrinsische Metriken:* beschreiben die den Daten innewohnende Qualität.
- *Kontextuelle Metriken:* beschreiben die Qualität in Bezug auf die auszuführenden Aufgaben.
- *Metriken für die Repräsentativität:* beschreiben die Qualität in Bezug auf Verständlichkeit und Datenformate.
- *Metriken für die Zugänglichkeit:* beschreiben die Qualität im Hinblick auf Zugänglichkeit und Sicherheit für die Nutzer.

Es gibt weitere Kategorisierungsvorschläge für die Einteilung von Datenqualitätsmetriken. Dazu gehört die Kategorisierung von Bovee et al. [95] in die vier Unterkategorien Verfügbarkeit, Interpretierbarkeit, Relevanz und Integrität. In der vorliegenden wissenschaftlichen Arbeit wird auf die Kategorisierung von Wang et al. zurückgegriffen, da diese auf einer Befragung unterschiedlicher Datenkonsumenten basiert und somit von praktischer Relevanz ist.

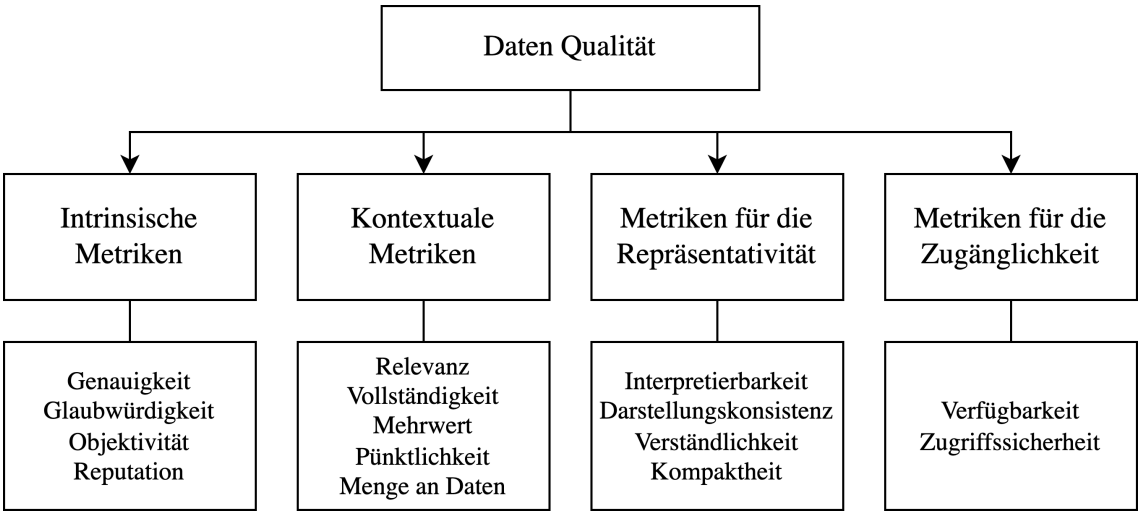


Abbildung 4.4.: Die vier Kategorien für die Metriken der Datenqualität

Eine Beschreibung der zentralen Metriken für jede der vier Kategorien findet sich in Tabelle 4.3. Darüber hinaus sind die sechzehn von Kahn et al. [96] definierten Dimensionen der Informationsqualität aufgelistet.

Tabelle 4.2.: Metriken für die Datenqualität . Dabei stehen folgende Abkürzung für jeweils eine Kategorie: Intrinsische Metrik (**I**), Kontextuelle Metrik (**K**), Metrik für die Repräsentativität (**R**) und Metrik für die Zugänglichkeit (**Z**)

Eigenschaft	Beschreibung	Kategorie
Genauigkeit	Die Genauigkeit ist ein Mass für die Nähe eines Datenwerts v, zu einem anderen Wert, v', der als als richtig angesehen wird.[97]	I
Glaubwürdigkeit	Das Ausmass, in dem die Informationen als wahr und glaubwürdig angesehen werden.[93]	I
Objektivität	Das Ausmass, in dem Informationen objektiv, unvoreingenommen und unparteiisch betrachtet werden.	I

Reputation	Das Ausmass, in dem Daten in Bezug auf ihre Quelle oder ihren Inhalt bewertet werden.[93]	I
Relevanz	Das Ausmass, in dem die verfügbaren Informationen für die gewählte Aufgabe nützlich sind.	K
Vollständigkeit	Gibt an, inwieweit die vorhandenen Daten für die definierte Aufgabe verfügbar sind.[93, 95]	K
Mehrwert	Das Ausmass, in dem die Informationen nützlich sind und durch ihre Nutzung Vorteile bieten.	K
Pünktlichkeit	Entspricht dem maximalen Alter, das ein Datum haben darf, damit Modell und Realität nicht zu sehr voneinander abweichen.	K
Menge an Daten	Das Ausmass, in dem die Menge oder der Umfang der verfügbaren Daten angemessen ist.	K
Interpretierbarkeit	Das Ausmass, in dem die Daten in geeigneten Sprachen, Symbolen und Einheiten vorliegen und die Definition eindeutig ist.[98]	R
Darstellungs-konsistenz	Gibt an, wie stark sich die Daten oder das Datenformat von zwei aufeinander folgenden Datensätzen unterscheiden. Batini et al. [83] definieren Konsistenz als Metrik zur Erkennung von semantischen Regelverletzungen. Beispielsweise sollte die Temperatur immer in $°C$ und nicht in $°F$ angegeben werden.	R
Verständlichkeit	Das Ausmass, in dem die Daten leicht nachvollziehbar sind.	R
Kompaktheit	Das Ausmass, in dem die Daten kompakt dargestellt werden.	R
Verfügbarkeit	Das Ausmass, in dem die Daten verfügbar oder schnell und einfach abrufbar sind. [93]	Z
Zugriffssicherheit	Das Ausmass, in dem der Zugang zu Informationen angemessen beschränkt wird, um die Sicherheit zu gewährleisten. [93]	Z

Es gibt eine Vielzahl weiterer Metriken, die sich den vier Kategorien zuordnen lassen (siehe Tabelle 4.3).

Tabelle 4.3.: Weitere Metriken für die Datenqualität . Dabei stehen folgende Abkürzung für jeweils eine Kategorie: Intrinsische Metrik (**I**), Kontextuelle Metrik (**K**), Metrik für die Repräsentativität (**R**) und Metrik für die Zugänglichkeit (**Z**)

Eigenschaft	Beschreibung	Kategorie
Manipulierbarkeit	Das Ausmass, in dem die Daten leicht zu manipulieren und auf verschiedene Aufgaben anwendbar sind.	R
Sicherheit (Safety)	Es ist die Fähigkeit, ein ein akzeptables Risikoniveau für Menschen, Prozesse, Eigentum oder die Umwelt zu erreichen.[99]	Z
Duplikate	Gibt an, wie viele Daten unbeabsichtigt mehrfach vorkommen.[100]	K
Daten-spezifikation	Ist eine Metrik, die angibt, ob ein adäquates Datenmodell vorhanden ist. Dies beinhaltet die Vollständigkeit eines Datenmodells mit den zugehörigen Geschäftsregeln, Metadaten und Referenzdaten sowie der zugrundeliegenden Dokumentation [100].	R
Darstellungs-qualität	Definiert die Qualität der Darstellung von Daten auf einem Datenträger. Ein wesentlicher Aspekt ist das Datenformat und die Nutzbarkeit der Daten [100].	R

4.4. Datenvorverarbeitung und Datenvalidierung

Die Datenanalyse erfordert eine hohe Qualität der verwendeten Daten. Liu et al. [101] fokussieren sich auf Big Data [102] (zusammenhängende Informationen, die aus einer grossen Menge und komplexen Datenstrukturen bestehen) und beschreiben das Problem wie folgt *„We believe that the bigness of big data not only refers to its large data volume, complex data structure, and fine granularity but also to the significance of data quality and usage problems in big data.“*. Als Gründe für eine schlechte Datenqualität nennen sie *a*) nicht authentische Daten, *b*) unvollständige Daten und Daten mit Rauschen, *c*) die Genauigkeit der Darstellung der Daten, *d*) Probleme mit der Konsistenz und Zuverlässigkeit der Daten und *e*) ethische Bedenken, wenn die Daten öffentlich zugänglich sind. Daraus folgt, dass die verwendeten Daten zunächst durch Datenvorverarbeitung für die eigentliche Datenanalyse aufbereitet werden müssen.

Xie, Gao und Tao [103, 104] beschreiben den kompletten Big Data Datenverarbeitungsprozess bestehend aus sieben Teilprozessschritten (siehe Abbildung 4.5 und die Erläuterung in Tabelle 4.4). Der Schritt der Datenzwischenspeicherung kann je nach Anwendungsszenario vernachlässigt werden, z.B. bei Szenarien mit weniger Daten. Die Datenvorverarbeitung besteht aus den Schritten 1. Sammeln von Daten bis 5. Validierung.

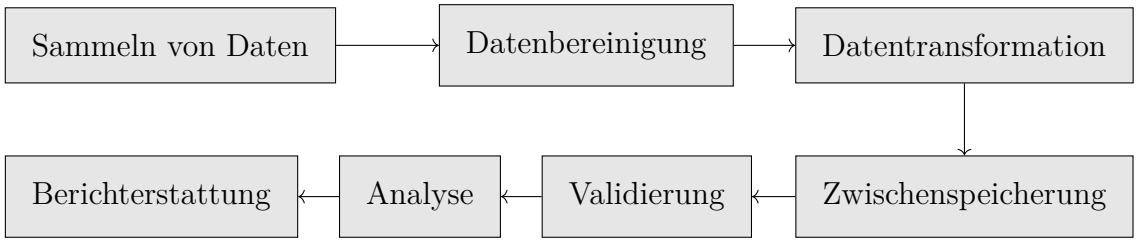


Abbildung 4.5.: Big Data Datenvalidierungsprozess [103]

Tabelle 4.4.: Beschreibung der Prozessschritte zur Datenvalidierung [103]

Nr	Schritt	Beschreibung
1.	**Sammeln von Daten**	Sammeln verschiedener Daten aus unterschiedlichen Quellen. Im industriellen Umfeld sind dies z.B. Messwerte von verschiedenen Sensoren.
2.	**Datenbereinigung**	Entfernen von ungenauen, falschen oder irrelevanten Daten.
3.	**Datentransformation**	Vereinheitlichen von unterschiedlichen Datenformaten auf ein Datenformat. Das neue Datenformat wird vom Zielsystem vorgegeben.
4.	**Zwischenspeicherung**	Speichern der Daten in ein Big Data Repository.
5.	**Validierung**	Bestimmung der Datenqualität durch Überprüfung von Datentypen, Datenformaten, einfachen Bedingungen (z. B. ein bestimmter Wert liegt über oder unter einem bestimmten Schwellenwert) usw.
6.	**Analyse**	Analyse der Datenqualität auf der Grundlage der Ergebnisse der zuvor durchgeführten Validierung.
7.	**Berichterstattung**	Rückmeldung und Dokumentation der Validierungsergebnisse (Datenqualität).

In der Beschreibung von Xie et al. erfolgt die Datenvalidierung nach der Zwischenspeicherung. Durch die Verwendung des DDVN Ansatzes (erläutert in Kapitel 5) kann die Validierung an verschiedenen Stellen der Datenvorverarbeitung erfolgen (siehe Abbildung 4.6): *a*) nach der Datenerhebung, *b*) nach der Datenbereinigung, *c*) nach der Datentransformation oder *d*) nach der Datenzwischenspeicherung. Dies ermöglicht eine grössere Flexibilität bei der Datenvalidierung.

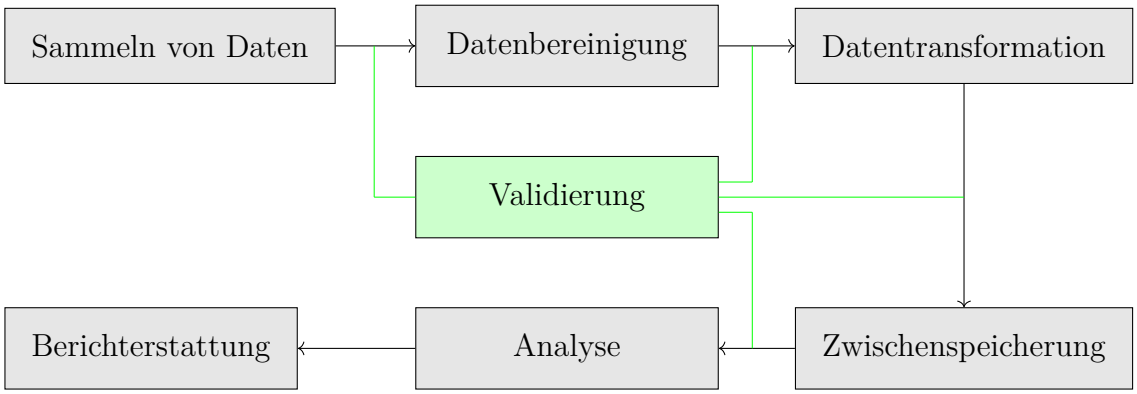


Abbildung 4.6.: Flexible Datenvalidierung in der Datenvorverarbeitung

Durch die Verschiebung der Datenvalidierung sind alternative Betrachtungen möglich. So kann eine Betrachtung auf Basis der Rohdaten oder auf Basis der bereinigten Daten erfolgen, ohne dass eventuell wichtige Daten vorher herausgefiltert wurden. Auch verrauschte Daten können zur Analyse beitragen. Verrauschte Daten sind ein Hinweis darauf, dass ein bestimmter Sensor ausgetauscht werden muss, da er die gewünschte Genauigkeit nicht mehr unterstützen kann.

Neben den unterschiedlichen Betrachtungsweisen gibt es noch einen weiteren Vorteil. Je nach Netzwerk kann es sein, dass die Maschine selbst nur die bereinigten Daten veröffentlicht und diese direkt analysiert werden müssen, ohne auf eine Zwischenspeicherung o.ä. warten zu müssen.

5. Verteilte Datenvalidierungsnetzwerke

Das folgende Kapitel befasst sich mit dem Aufbau von Datenverarbeitungssystemen im industriellen Umfeld. Drei verschiedene Ansätze für die Datenvalidierung als Datenvorverarbeitungsschritt werden erläutert. Es handelt sich um *a*) einfache Systeme ohne Datenvalidierung, *b*) Systeme mit interner Datenvalidierung und *c*) Systeme mit externem Datenvalidierungssystem. Aus dem einzelnen externen Datenvalidierungssystem ergibt sich die Verwendung eines DDVN. Eine Erläuterung des DDVN mit seinen Komponenten bildet den Abschluss dieses Kapitels.
Ein Teil der nachfolgenden Inhalte wurde bereits in komprimierter Form in den Papern [74, 75, 105, 106] publiziert.

5.1. Aufbau von produktiven Systemen

Im industriellen Umfeld werden für Predictive Maintenance, Condition Monitoring, optimierte Auftragsplanung für Maschinen etc. verschiedene Datenverarbeitungssysteme, auch Sink oder Datensenke genannt, eingesetzt. Die verwendeten Daten werden dabei von einer Source, auch Datenquelle genannt, bereitgestellt und meist direkt (siehe Abbildung 5.1) ohne Vorverarbeitung an die Sink gesendet. Bei den Daten der Source handelt es sich meist um unstrukturierte Sensordaten, die in einer NoSQL-Datenbank gespeichert werden können. Eine hohe Datenqualität ist unabdingbar, um genaue Vorhersagen mit den Datenverarbeitungssystemen zu gewährleisten. Hierzu bietet sich der Einsatz eines Datenvalidierungssystems an, welches die bereitgestellten Daten bewertet und das Ergebnis der Bewertung an den Sink weitergibt.

Die meisten Produktionssysteme sind massgeschneidert und verwenden individuelle Lösungen. Eine optimale Lösung für die Datenvorverarbeitung einschliesslich der Validierung kann daher nicht allgemein definiert werden. Vielmehr gibt es eine Vielzahl von Möglichkeiten, aus denen je nach Anwendungsfall eine Lösung ausgewählt wird.

5.1.1. Systeme ohne Datenvalidierung

Bei Systemen mit geringer Sicherheit und geringen Anforderungen an die Datenqualität kann die Source ihre Daten direkt an die Sink übermitteln (siehe Abbildung 5.1). Einerseits ist die Übertragung der Daten schnellstmöglich, da keine zusätzlichen Komponenten zwischengeschaltet sind, andererseits sind diese Systeme einfach aufgebaut, so dass die Daten der Source auch direkt zugänglich sind. Nicht validierte Daten stellen ein grosses Sicherheitsrisiko dar, da dadurch die Vorhersagen beeinträchtigt werden können.

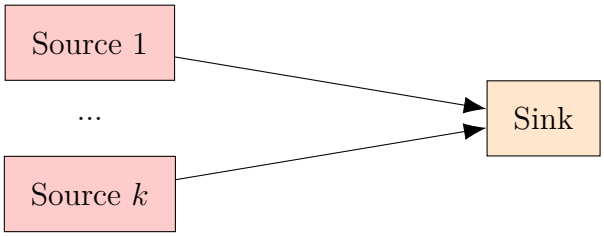


Abbildung 5.1.: Aufbau eines Systems ohne Datenvalidierung

5.1.2. Systeme mit interner Datenvalidierung

Um eine höhere Datenqualität für die Datenvorhersage zur Verfügung zu stellen, kann die Datenvalidierung Teil der internen Datenvorverarbeitung sein (siehe Abbildung 5.2). Dadurch können fehlerhafte Daten frühzeitig aussortiert und genaue bzw. korrekte Daten an die Sink weitergeleitet werden. Ist für die korrekte Validierung von Daten eine Datenkorrelation notwendig, so müssen alle dafür notwendigen Daten an die Sink weitergeleitet werden, auch wenn sie für die eigentliche Datenverarbeitung nicht benötigt werden. Die für die Korrelation benötigten Daten können sehr umfangreich sein, so dass ein grosser Overhead an Datenübertragungen entstehen kann.

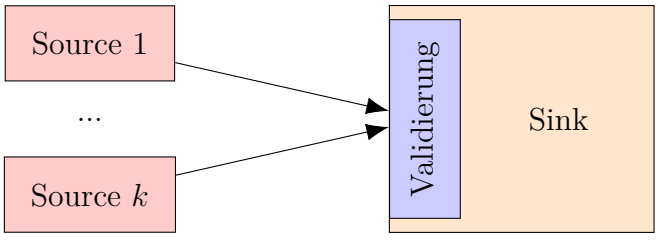


Abbildung 5.2.: Aufbau eines Systems mit interner Validierung

5.1.3. Systeme mit externer Datenvalidierung

Wenn die Wiederverwendbarkeit der Datenvalidierungsergebnisse im Vordergrund steht, kann die Datenvalidierung aus dem Sink ausgelagert werden (siehe Abbildung 5.3). Auf diese Weise können verschiedene Datenverarbeitungssysteme dieselben Datenvalidierungsergebnisse verwenden. Darüber hinaus können Datenkorrelationen durchgeführt werden, ohne dass die Autorisierung des konsumierenden Sinks berücksichtigt werden muss. Die Verwendung eines einzigen Validierungssystems zwischen Sources und Sinks hat den Nachteil, dass bei einer erfolgreichen Korruption der Validierung alle konsumierenden Sinks betroffen sind.

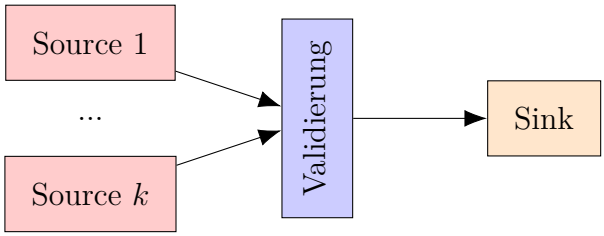


Abbildung 5.3.: Aufbau eines Systems mit einer externen Validierung

5.1.4. Systeme mit externer verteilter Datenvalidierung

Bei hohen Anforderungen an die Sicherheit der Datenvalidierung und die Wiederverwendbarkeit der Datenvalidierungsergebnisse kann eine externe verteilte Datenvalidierung (siehe Abbildung 5.4) eingesetzt werden. Dabei werden verschiedene Validierer mit der Datenvalidierung beauftragt und deren Ergebnisse im Sink akkumuliert. Die Validierer können entweder den gleichen oder unterschiedliche Algorithmen verwenden. Es muss lediglich die Möglichkeit der Akkumulation der Ergebnisse gegeben sein, d.h. entweder liefern alle Validierer ein logisches Ergebnis (ist valide, ist nicht valide) oder einen numerischen Wert, der in Relation gesetzt werden kann.

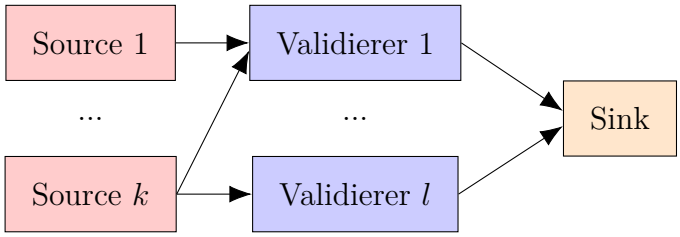


Abbildung 5.4.: Aufbau eines Systems mit einer externen verteilten Validierung

Eine Zusammenfassung der verschiedenen Datenvalidierungsansätze mit ihren Vor- und Nachteilen findet sich in Tabelle 5.1. Hierbei werden allgemeine Eigenschaften: die Performance der Kommunikation, die Einfachheit der Implementierung, die Sicherstellung der Datenqualität und die Möglichkeit des Verteilens der Validierungsergebnisse sowie sicherheitsspezifische Eigenschaften basierend auf der CIA-Triade (Vertraulichkeit, Integrität und Verfügbarkeit) aufgezeigt. Eine detaillierte Analyse findet sich in Kapitel 7.

Tabelle 5.1.: Vergleich verschiedener Ansätze von Datenvalidierungssystemen

Keine Datenvalidierung
Performance: ist am besten, da keine Validierung durchgeführt wird. *Implementierung:* ist am einfachsten, da keine zusätzlichen Kommunikationskanäle benötigt werden. *Datenqualität:* kann nicht direkt, sondern nur anhand der fachlichen Ergebnisse bestimmt werden. *Verteilen der Validierungsergebnisse:* der Ergebnisse der Validierung kann nicht erfolgen, da keine Ergebnisse vorliegen. *Vertraulichkeit:* ist grundsätzlich hoch, da keine anderen Komponenten die Daten erhalten. *Integrität:* ist nicht gegeben, da korrupte Sensorwerte unkontrolliert in das System gelangen können. *Verfügbarkeit:* die Sources sind direkt angeschlossen, daher ist die Verfügbarkeit gut.
Interne Datenvalidierung
Performance: hängt von der Anzahl der Validierungen ab. Für einfache Validierungen ist eine interne Datenvalidierung möglich. Steigt jedoch das Validierungsvolumen, ist eine externe Datenvalidierung sinnvoll. *Implementierung:* ist einfach. Allerdings müssen die internen Ergebnisse an die Sink übergeben werden. Ausserdem wird das Separation of Concerns Prinzip verletzt. *Datenqualität:* kann direkt bestimmt werden. *Verteilen der Validierungsergebnisse:* der Validierungsergebnisse kann nicht erfolgen, da die Ergebnisse erst im Zielsystem vorliegen. *Vertraulichkeit:* ist grundsätzlich hoch, da die Validierung im Sink erfolgt. *Integrität:* ist gegeben, da eine direkte Validierung erfolgt. *Verfügbarkeit:* die Sources sind direkt verbunden, daher ist die Verfügbarkeit gut. Performanceprobleme können auftreten, wenn zu viele Validierungen durchgeführt werden.

Externe Datenvalidierung
Performance: hängt von der Anzahl der Validierungen ab. Bei komplexeren Validierungen ist eine externe Datenvalidierung sinnvoll. *Implementierung:* ist komplex, da eine neue externe Komponente bereitgestellt werden muss. *Datenqualität:* kann durch das Validierungssystem bestimmt und an die Sink weitergeleitet werden. *Verteilen der Validierungsergebnisse:* der Validierungsergebnisse kann erfolgen, da die Ergebnisse bereits vor der Sink vorliegen. *Vertraulichkeit:* ist niedriger, da die Validierung in einer zusätzlichen Komponente stattfindet und somit die Daten an eine zusätzliche Komponente weitergegeben werden. Diese Komponente könnte korrumpiert werden. *Integrität:* ist gegeben, da eine Validierung durchgeführt wird. Die Gewährleistung der Integrität wird aufwendiger, da eine neue Komponente in das System eingefügt wird. *Verfügbarkeit:* die Sources sind mit einem anderen Validierer verbunden, daher ist die Verfügbarkeit geringer. Allerdings können die Daten von den Sources zusätzlich zum Validierungssystem auch an den Sink weitergeleitet werden. Es kann zu Performance-Problemen kommen, wenn zu viele Validierungen durchgeführt werden.
Externe verteilte Datenvalidierung
Performance: hängt von der Anzahl der Validierungen ab. Bei komplexeren Validierungen ist eine externe verteilte Datenvalidierung sinnvoll. *Implementierung:* ist am komplexesten, da neue externe Komponenten bereitgestellt werden müssen. *Datenqualität:* kann von den Validierern bestimmt und an die Sink weitergeleitet werden. Dabei können verschiedene Kombinationen von Validierungsergebnissen weitergeleitet und somit verschiedene Qualitätsmetriken verwendet werden. *Verteilen der Validierungsergebnisse:* der Validierungsergebnisse kann erfolgen, da die Ergebnisse bereits vor der Sink vorliegen. Ausserdem können spezifischere Validierungsergebnisse übermittelt werden, da nicht jeder Validierer alle Daten für eine Validierung erhält. *Vertraulichkeit:* ist am niedrigsten, da die Validierung in mehreren zusätzlichen Komponenten stattfindet und somit die Daten an zusätzliche Komponenten weitergegeben werden. Diese Komponenten könnten korrumpiert werden. *Integrität:* ist gegeben, da eine Validierung stattfindet. Die Sicherstellung der Integrität wird aufwendiger, da neue Komponenten zum System hinzugefügt werden. Allerdings ist die Abhängigkeit von einem einzelnen Validierungssystem nicht mehr gegeben, so dass ein Angriff je nach Aggregierungsalgorithmus mehrere Validierer korrumpieren muss.

Verfügbarkeit: die Sources sind mit anderen Validierern verbunden, so dass die Verfügbarkeit kaum eingeschränkt ist, da Redundanzen vorhanden sind.

5.2. Verteilte Datenvalidierungsnetzwerke

Ein DDVN wird für Anwendungen verwendet, die eine hohe Datenintegrität erfordern. Dazu gehören das autonome Fahren und die industrielle Fertigung von Produkten. Diese Dissertation konzentriert sich auf Anwendungen im industriellen Umfeld. Das DDVN führt Korrektheits- und Plausibilitätsprüfungen für verschiedene industrielle Daten durch. Beispiele sind die Anzahl der produzierten Teile, die gemessene Temperatur an einer bestimmten Stelle der Maschine oder die Drehzahl bei einem Fräsvorgang. Der in den meisten Datenverarbeitungssystemen vorhandene Single Point of Failure wird so entschärft und durch einen verteilten Ansatz ersetzt. Abbildung 5.5 zeigt die Struktur eines DDVN mit den verschiedenen Komponenten: Source, Gateways, Validierer-Cluster, Validierer (bezeichnet mit v_1 bis v_m) und Sinks (im folgenden Akkumulatoren genannt, da sie die Ergebnisse der Validierer akkumulieren). Nachfolgend werden alle Komponenten des DDVN im Detail beschrieben. Der Kommunikationsprozess und die Sicherheitsbetrachtung werden in den folgenden Kapiteln erläutert.

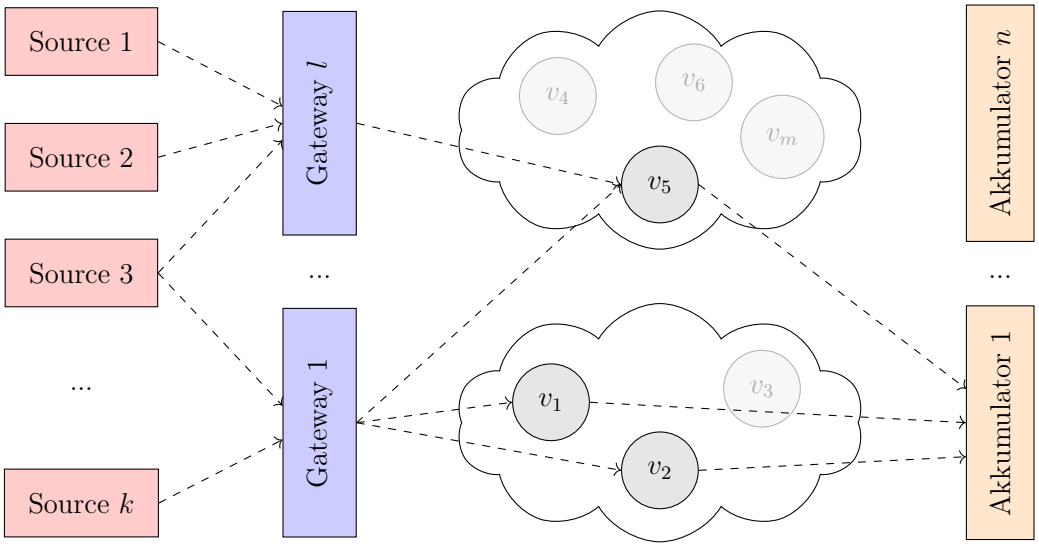


Abbildung 5.5.: Struktur des DDVN

5.2.1. Source

Im Folgenden wird davon ausgegangen, dass die Source $\{s_1, s_2, ..., s_k\}$ entweder ein passiver oder ein aktiver Sensor ist. Der Unterschied zwischen beiden besteht

darin, dass ein aktiver Sensor aufgrund des verwendeten Messprinzips ohne zusätzliche elektrische Energieversorgung ein elektrisches Signal erzeugt. Ein passiver Sensor benötigt eine Primärelektronik und eine externe Stromversorgung, um ein elektrisches Signal zu erzeugen. Der Nachteil aktiver Sensoren besteht darin, dass in den meisten Fällen nur Änderungen der Messgrösse erfasst werden, jedoch keine statischen oder quasistatischen Werte gemessen werden können. Aufgrund dieses Nachteils werden passive Sensoren häufiger eingesetzt als aktive Sensoren.

In Industrie 4.0-Umgebungen werden Sensorwerte hauptsächlich für relevante geschäftliche Anwendungsfälle gemessen oder berechnet. Mögliche Beispiele für Geschäftsanwendungen sind Condition Monitoring, Predictive Maintenance, Auftragsplanung an Maschinen etc. Die meisten Anwendungsfälle sind anwendungsspezifisch, z.B. haben Lasersysteme ganz andere Parameter als Abfüllanlagen und umgekehrt. Eine Auflistung der verschiedenen Sensoren und ihrer Anwendungsbereiche findet sich unter Tabelle 5.2. Neben aktiven und passiven Sensoren wird die Sensorliste um Kameras und Zähler erweitert, da auch diese Geräte mit ihren generierten Daten einen grossen Mehrwert für die aktuellen Anwendungsfälle bieten.

Tabelle 5.2.: Sensoren und deren Anwendungsbereiche

Sensor Typ	Anwendungsbereiche
Temperatur Sensor	- Kontrolle der Maschinentemperatur - Beobachten der Kunststofftemperatur während Gussvorgängen - Überwachung zum Schutz vor der Überhitzung einzelner Komponenten
Beschleunigungssensor	- Beschleunigung von Transportrobotern in einer Lagerhalle - Sicherstellung eines langsamen Anflaufens mit einer niedrigen Spindeldrehzahl - Kontrolle der Steuerung von Industrie Roboterarmen [107]
Drucksensor	- Aufnehmen und Platzieren von Objekten mit einem Roboterarm [108] - Dichtheitsprüfung für Flaschen und Geräte - Überwachung des Kompressionsdrucks

Feuchtigkeitssensor	- Sicherstellung einer hohen Sauerstoffkonzentration - Feuchtigkeitskontrolle bei der Waferfertigung [109] - Feuchtigkeitskontrolle bei der chemischen Gasdestillation
Durchflussensor	- Kontrolle des Füllstands in einem Tank - Sicherstellung des Verhältnisses zwischen Chemikalien - Überwachung einer Versorgungsleitung
Kamera	- Bewertung der Schweissnahtqualität - Korrekte Positionierung eines Bauteils für die Bearbeitung - Bestimmung der visuellen Qualität eines Bauteils (z.B. keine Haarrisse)
Zähler	- Überprüfung der Anzahl der produzierten Teile - Messung des Verschleisses der Komponenten bis zum Austausch - Vergleich der Produktionsgeschwindigkeit von Maschinen

5.2.2. Gateways

Sensoren und Aktoren verfügen über eine Vielzahl von Schnittstellen zur Übertragung von Messdaten, Befehlen oder Zuständen. Generell werden drei Arten der Kommunikation unterschieden [110]:

- *Analoge Signale:* Analogsignale liegen in der Regel im Bereich von $4 - 20mA$ und werden entweder direkt oder über einen Analog-Digital-Wandler (ADC) an einen Datenprozessor angeschlossen. Im Datenprozessor erfolgt die Zuordnung zwischen dem übertragenen Stromwert und der tatsächlich gemessenen physikalischen Grösse (z.B. Temperatur).
- *Feldbusse:* Feldbusse sind in der *IEC 61158* [111] definiert und bieten eine Bustopologie, so dass mehrere Sensoren und Aktoren über die dieselbe Leitung kommunizieren können. Dadurch wird die Anzahl der benötigten Leitungen reduziert, eine bessere Wartbarkeit gewährleistet und ein einheitlicher Kommunikationsstandard geschaffen.
- *Ethernet:* definiert durch *IEEE 802.3* [112], ist ein LAN-Standard und kann alternativ zu Feldbussen verwendet werden. Der Hauptvorteil des Einsatzes von Ethernet besteht darin, dass alle Komponenten im Intranet oder Internet eingesetzt werden können. Dies ermöglicht neue Anwendungsszenarien wie z.B. vernetzte Smart Factories.

Bei kostengünstigen Sensoren oder Aktoren und älteren Maschinen ohne eigene Netzwerkfähigkeit werden Gateways zur Ergänzung der Netzwerkfähigkeit eingesetzt [113]. Insbesondere im Maschinenumfeld und bei der vorausschauenden Wartung werden Nachrüstungen vorgenommen und die Maschinen mit internetfähigen Gateways ausgestattet. Neben der Kommunikationsfähigkeit wird auch sichergestellt, dass nur ein Kommunikationskanal zwischen einer Maschine und einem Netzwerk genutzt wird. Dadurch werden Datenlecks an einer nicht berücksichtigten Kommunikationsschnittstelle vermieden.

Im DDVN akkumulieren die Gateways $\{g_1, g_2, ..., g_l\}$ die empfangenen Sensordaten und ermöglichen eine sichere Kommunikation der Sensordaten zu den Validierern oder Akkumulatoren. Darüber hinaus werden die Gateways als Vertrauensanker im DDVN akzeptiert und sind somit vertrauenswürdig.

5.2.3. Validierer Knoten

Zur Validierung der Daten verwendet das DDVN Validator Knoten, die auch als Validierer oder Validatoren $(v_1, ..., v_m)$ bezeichnet werden. Ein Validierer kann jede netzwerkfähige Hardware im Produktionsnetzwerk sein, die über ungenutzte Rechenkapazität verfügt. Dazu gehören Maschinensteuerungen, Sensoren mit eigener CPU, Tabletts, PCs usw. (siehe Abbildung 5.6). Die Hardware des Validierers ist für die meisten Validierungsoperationen irrelevant, mit Ausnahme von Validierungsoperationen, die besondere Anforderungen stellen, wie z.B. die Verwendung einer Graphical Processing Unit (GPU). Alternativ kann ein Validator auch in der Edge oder in der Cloud betrieben werden. Abhängig von der verfügbaren Rechenkapazität kann ein Validierer mehr oder weniger unterschiedliche Datentypen validieren, jedoch immer mindestens einen. Beispiele für zu validierende Daten sind die Temperatur am Punkt A der Maschine Z oder verwandte Daten, wie z. B. die gesamten gemessenen Temperaturen der Maschine Z.
Neben der Validierung der Daten ist ein Validierer auch für die Übermittlung des Ergebnisses und gegebenenfalls der Daten an einen Akkumulator verantwortlich.

5.2.4. Cluster von Validierer Knoten

Die logische Gruppierung von Validierern wird als Validierunger Knoten Cluster, Validierer Cluster, Validatorencluster oder einfach als Cluster $c_1 = \{v_1, v_2, v_3\}$ und $c_2 = \{v_4, v_5, v_6, ..., v_m\}$ bezeichnet. Mögliche Gruppen können auf der Grundlage von a) Validierungsmerkmalen, wie Temperatur- oder Feuchtigkeitsvalidierung, b) physikalischen Merkmalen, wie lokaler Anordnung oder Speichergrösse, oder c) Merkmalen der Validierer, wie dem Betriebssystem des Validierers, gebildet werden. Ein Validator ist nicht auf ein Cluster beschränkt und kann daher auf der Grundlage einer Vielzahl von Merkmalen mehrfach gruppiert werden. Die Cluster

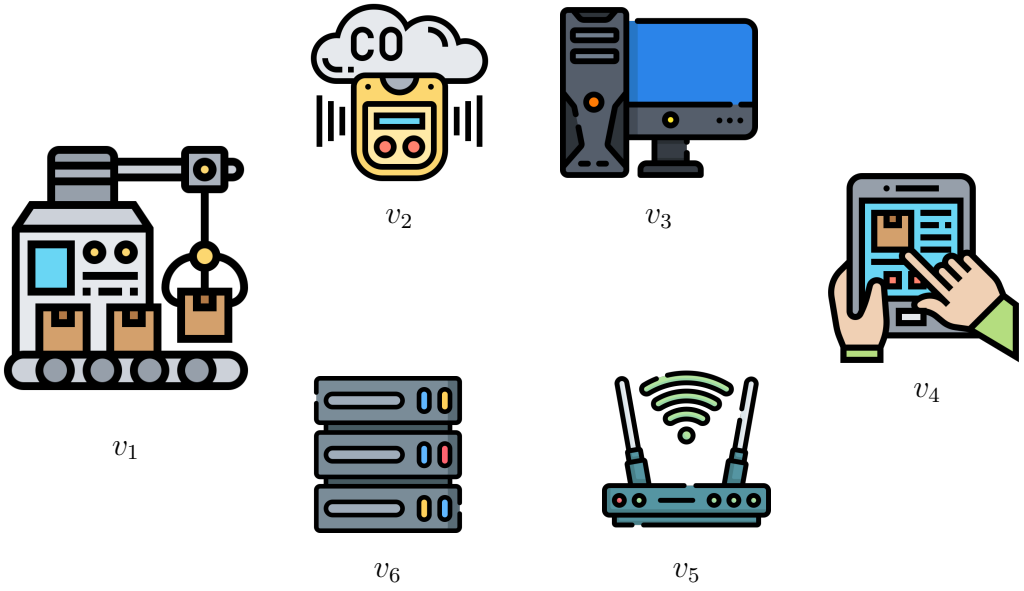


Abbildung 5.6.: Positionierung der Validiererknoten auf unterschiedlicher Hardware

werden vom Akkumulator verwendet, um eine möglichst heterogene Auswahl von Validierern sicherzustellen. Dadurch wird die Auswirkung von Zero-Day Exploits basierend auf bestimmten Eigenschaften, zum Beispiel dem verwendeten Betriebssystem, minimiert. Eine detaillierte Beschreibung der Sicherheitsvorteile des Cluster Ansatzes findet sich in Abschnitt 8.8.

5.2.5. Akkumulatoren

Der Akkumulator $a \in \{a_1, ..., a_n\}$ hat die Aufgabe, eine Vielzahl von Validierungsergebnissen zu sammeln und zu einer Validierungsentscheidung zusammenzufassen. Es ist Aufgabe des Akkumulators zu entscheiden, welche Validierer verwendet werden sollen (z.B. auf Basis ihrer Fähigkeiten oder Clusterzugehörigkeit) und ob die empfangenen Daten vertrauenswürdig sind. Ein Akkumulator ist ein zentraler Dienst, der Geschäftsanwendungen unterstützt oder selbst Geschäftsanwendungen ausführt.

5.3. Positionierung der Komponenten

Im industriellen Umfeld (siehe Abbildung 5.7) befinden sich die Sensoren an der Maschine oder an anderen Geräten im Produktionsprozess. Das Gateway ist entweder der Sensor selbst, sofern er netzwerkfähig ist, über ausreichende Rechenleistung verfügt und vertrauenswürdig ist, oder eine spezielle

Hardwarekomponente in der Nähe des Sensors. Der Validator kann jede in der Produktion verfügbare Hardware sein, z.B. ein Terminal in der Nähe der Maschine oder ein Tablet des Mitarbeiters, auf dem die Produktionsaufträge angezeigt werden. Zusätzlich zu den in der Fabrik verfügbaren Ressourcen kann ein Validator auch auf Edge-Ressourcen (z. B. Router oder Micro-Clouds) und auf Ressourcen in Cloud-Rechenzentren laufen. Da die Cluster nur logische Grenzen darstellen, ist keine zusätzliche Hardware erforderlich. Typischerweise befinden sich die Cluster in einem geschützten Bereich, z. B. in der firmeninternen Cloud, in der sensible Produktions-, Kunden- und Unternehmensdaten verarbeitet und gespeichert werden.

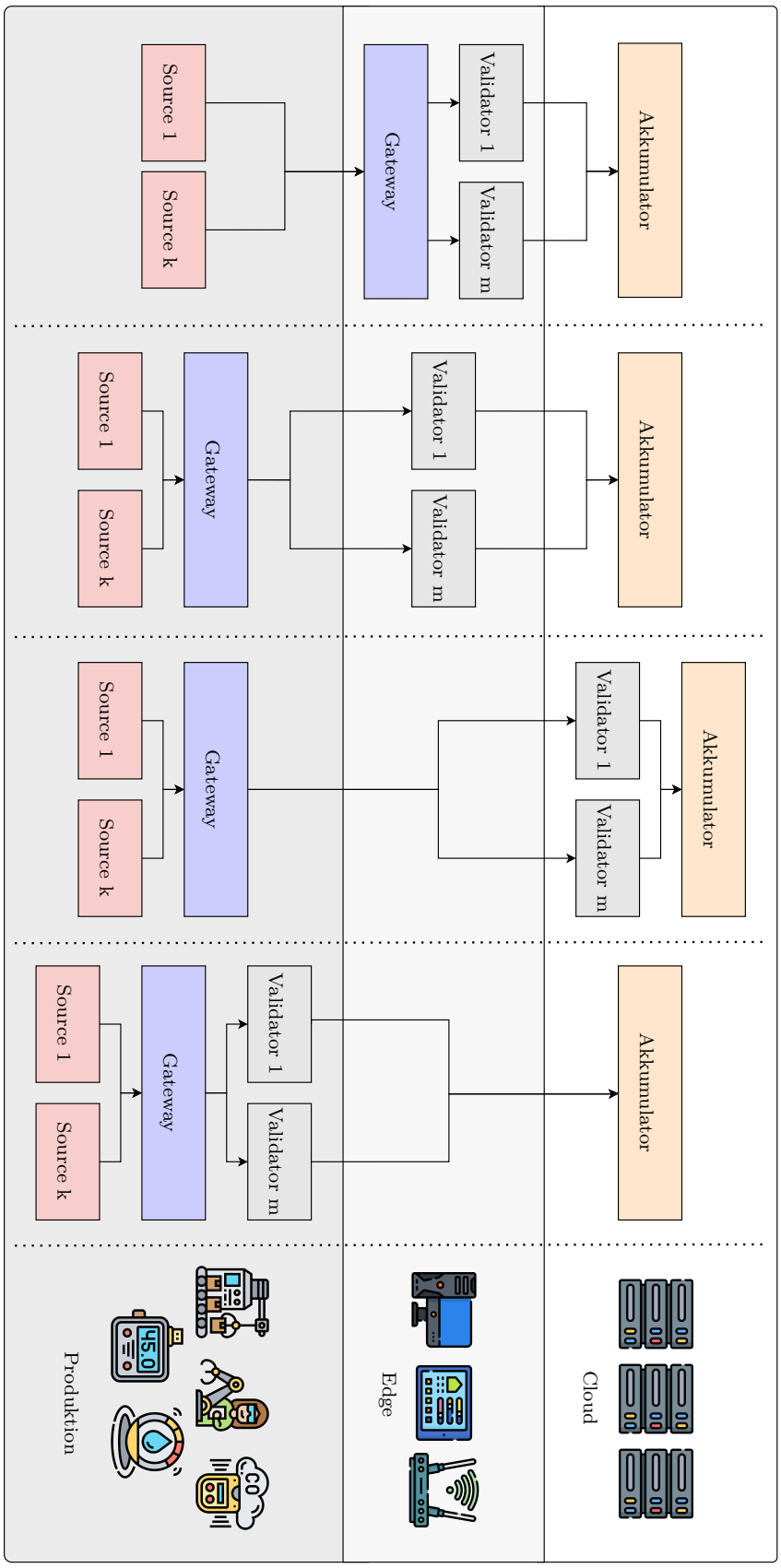


Abbildung 5.7.: Positionierung der DDVN Komponenten in Cloud, Edge und Produktionsumgebung

6. Kommunikation

Die Kommunikation im DDVN ist ein zentraler Bestandteil und wird im folgenden Abschnitt erläutert. Grundsätzlich sind nicht alle Komponenten miteinander verbunden, da es einerseits hardwaretechnische Einschränkungen gibt, z.B. hat ein herkömmlicher Sensor keine Netzwerkfunktionalität und muss mittels Gateway um diese Funktionalität erweitert werden. Bei den Validierern verhält es sich meist umgekehrt, d.h. diese haben zwar Netzwerkfunktionalität, sind aber meist nicht im direkten Umfeld des Sensors platziert, weshalb sie Daten nur über das Netzwerk selbst beziehen können.
Ein Teil der folgenden Inhalte wurde bereits in komprimierter Form in den Papern [74, 105] publiziert.

6.1. Arten von Umgebungen

Im Vergleich zu einem einzelnen Datenvalidierungssystem erfordert das clusterbasierte DDVN einen zusätzlichen Kommunikationsaufwand. Der zusätzliche Kommunikationsaufwand entsteht einerseits bei der Übertragung der Daten vom Gateway zu den Validierern und andererseits bei der Übertragung der Validierungsergebnisse von den Validierern zum Akkumulator.

Grundsätzlich gibt es acht verschiedene Möglichkeiten, eine DDVN-Umgebung aufzubauen (siehe Abbildung 6.1). In jeder dieser Strukturen muss der Akkumulator vertrauenswürdig sein, da sonst die eigentliche Geschäftsprognose nicht verwendet werden kann. Von den sieben verschiedenen Strukturen gibt es 1) eine vertrauenswürdige (a), 2) sechs halb-vertrauenswürdige ($\{b, c, d, e, f, g\}$) und 3) eine nicht vertrauenswürdige (h).

Der Fall (a) wird im Folgenden nicht weiter betrachtet, da in diesem Szenario alle Komponenten vertrauenswürdig sind, was in Industrie 4.0 Umgebungen zumeist nicht der Fall ist. Die Semi-Vertrauenswürdigen Fälle ($\{b, c, d\}$) werden ebenfalls nicht weiter betrachtet, da jeweils nur ein Teilbereich nicht vertrauenswürdig ist. Nicht-Vertrauenswürdige Umgebung (h) und Semi-Vertrauenswürdige Fälle ($\{e, f\}$) werden von der weiteren Betrachtung ausgeschlossen, da die Implementierung von sicheren Gateways für industrielle Anwendungsfälle [114, 115] ein eigenes wissenschaftliches Gebiet darstellt. Aufgrund der aufgezeigten Punkte wird für die vorliegende wissenschaftliche Arbeit die Semi-Vertrauenswürdige Umgebung (g) untersucht.

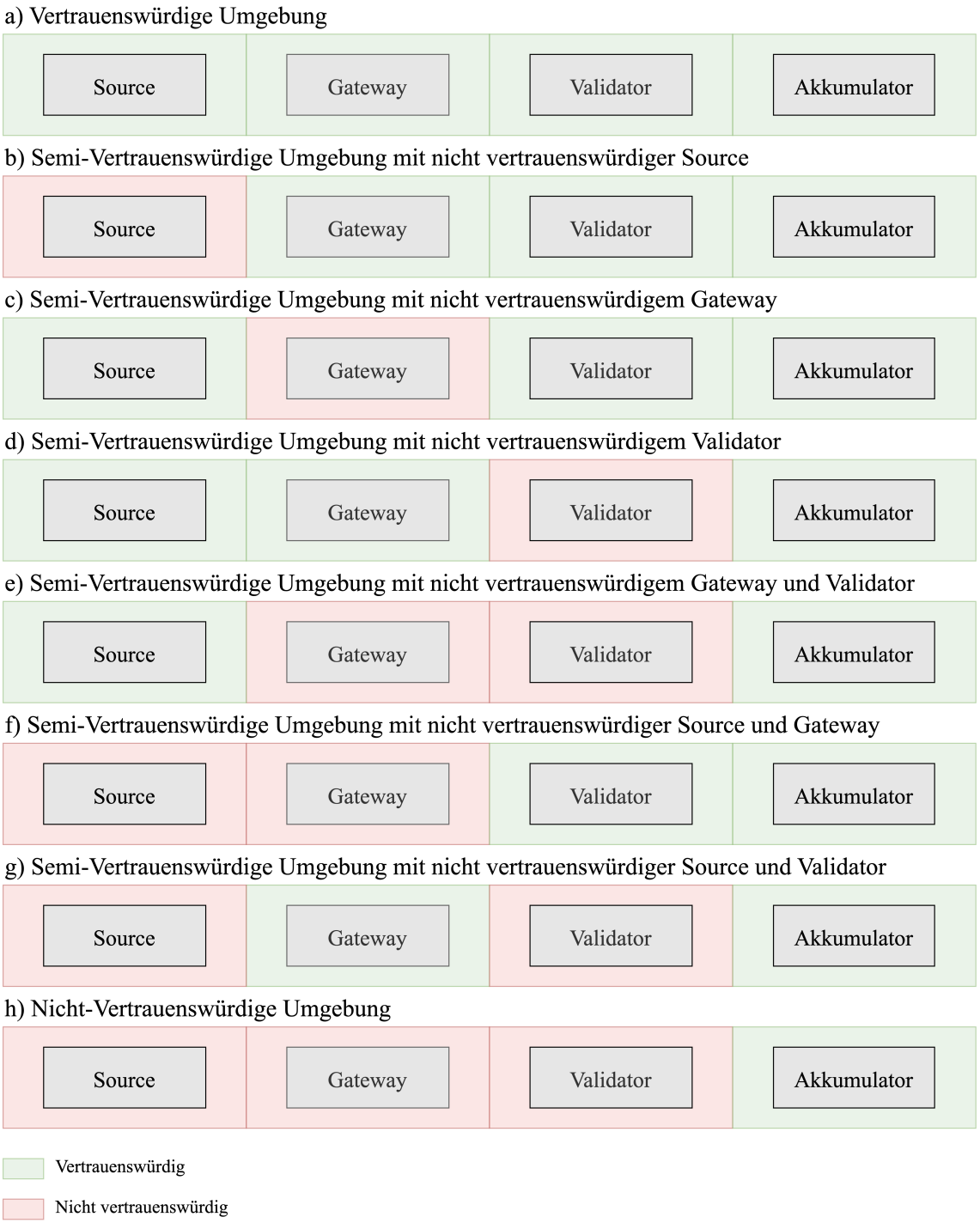


Abbildung 6.1.: Unterschiedliche Arten von Umgebungen

Zusammenfassend wird in dieser Arbeit davon ausgegangen, dass Gateways und Akkumulatoren vertrauenswürdig sind. Alle anderen Komponenten wie Quellen und Validierer sind nicht vertrauenswürdig. Daher sind für die Datenübertragung

zwischen den Komponenten verschiedene Sicherheitsmechanismen wie digitale Signaturen und Datenverschlüsselung erforderlich. Im Folgenden werden diese Sicherheitsmechanismen für die Kommunikation beschrieben.

6.2. Allgemeiner Kommunikationsablauf

Beim DDVN kommunizieren mehrere Komponenten miteinander (siehe Abbildung 5.5). Der Kommunikationsprozess gliedert sich in drei Teile: 1) Auswahl der zu verwendenden Validatoren, 2) Validierungsvorgang und 3) Datenübertragung zum Akkumulator.

Auswahl der Validierer

Die Auswahl der zu verwendenden Validierer (siehe Abbildung 6.2) wird durch den vertrauenswürdigen Akkumulator vorgenommen. Dabei sammelt der Akkumulator zunächst alle Attribute der Validierer. Die Attributdatei enthält zum einen die Validierungsfähigkeiten des Validierers und zum anderen den globalen Identifikator (z.B. die IP des Validierers) des Validierers (siehe Listing 6.1). Falls sich die Validierer in unterschiedlichen Netzwerken befinden, kann ein alternativer globaler Identifier, z.B. mittels UUID, verwendet werden.

Die Validierungsfähigkeiten bestehen aus einem fachlichen Hauptidentifikator und einem Subidentifikator. Der Hauptidentifikator kann zum Beispiel einer bestimmten Maschine mit einer Versionsnummer entsprechen (*Maschinentyp-112233*, *Maschinentyp-445566* oder *Maschinentyp-778899*) und der Subidentifikator den Parametern oder Eigenschaften, die der Validator kontrollieren kann (*Temperatur*, *Feuchtigkeit*, *Stückzahl* oder *Steuersignale*). Die Attributdatei enthält neben der globalen Identifikation und den Attributen auch den öffentlichen Schlüssel des jeweiligen Validators. Damit werden die Validierungsergebnisse signiert und die Rückverfolgbarkeit sichergestellt. Zusätzlich wäre eine Erweiterung der Attributdatei um Clusterinformationen möglich. Damit könnte der Akkumulator die Validierer anhand der Clustereigenschaften auswählen.

Der Akkumulator erstellt eine Policy-Datei aus den von den Validierern erhaltenen Fähigkeiten. Ein Beispiel ist in Listing 6.2 dargestellt. Sie besteht aus einer Menge von Registrierungen für die Hauptidentifier mit ihren Subidentifiern sowie den zugeordneten Validierern, die eine Validierung für die jeweilige Registrierung durchführen sollen. Zusätzlich sind der öffentliche Schlüssel und die IP des Akkumulators enthalten. Damit wird dem Validierer der Zielakkumulator für die Validierungsergebnisse über eine Subpolicy-Datei mitgeteilt (siehe Listing 6.3).

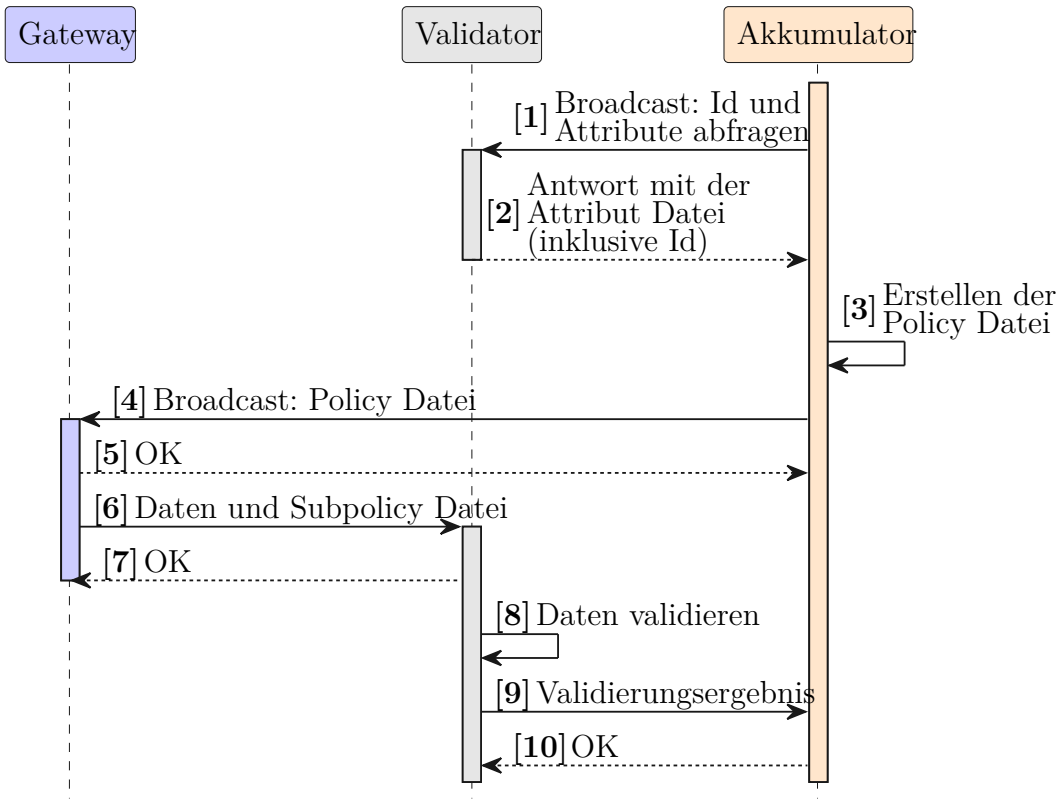


Abbildung 6.2.: Auswahlverfahren der Validierer

```
Globaler Identifier:
38.246.231.241

Attribute:
    Maschine Typ-112233: Temperatur, Feuchtigkeit
    Maschine Typ-445566: Stückzahl
    Maschine Typ-778899: Steuersignale
    ...

Öffentlicher Schlüssel:
-----BEGIN PUBLIC KEY-----
MIGeMA0GCSqGSIb3DQEBAQUAA4GMADCBiAKBgGnEByG2ckQAv51tfFek
YfB0sTvfJhygc8b3f34smz76dzLIPTqyxyNmMaGLPFG9/YiR/kqtnB59
a42xTjeNJmRTokSomqsEQs8Ou32cfXA4e4TGF7ofkEV5LEdYwKbKRgEc
8E2R3ceVL9dwJBTnF+M3c/E2MEaqp7/nQt/c4I2vAgMBAAE=
-----END PUBLIC KEY-----
```

Listing 6.1: Beispiel einer Attributdatei mit globalem Identifikator

Damit hat der Akkumulator die Kontrolle über die Datenvalidierung und kann diese ohne Eingreifen anderer Komponenten in Auftrag geben. Wenn ein Validierer sich ungewöhnlich verhält, kann der Akkumulator ihn sofort aus der Policy-Datei entfernen. Alternativ kann der Akkumulator einen Validierer beobachten und eine Verhaltensanalyse durchführen. Je nach Ergebnis kann ein Validierer aus dem Netzwerk entfernt werden.

Als Alternative zur Verwendung von Broadcasts, Attributdateien und Policy-Dateien könnte die Verwendung eines Service Mesh mit Service Discovery in Betracht gezogen werden. Ein mögliches Service Mesh hierfür ist Consul [116] von HashiCorp. Consul besteht aus einer Service Registry, bei der sich Services beim Hochfahren registrieren. Sollte ein Service A Zugriff auf einen anderen Services B benötigen, dann kann A eine Query an die Service Registry senden und die Adresse von B erhalten. Dadurch wird ein ähnliches Verhalten wie von einem Load Balancer erreicht, jedoch ohne multiple Instanzen von Load Balancern. Neben der Service Discovery Fähigkeit bietet Consul noch weitere Funktionalitäten, wie die zentrale Konfiguration von Services und die Segmentation von Services. Umgelegt auf das DDVN kann sich ein Validierer beim Hochfahren in der Service Registry als Service registrieren und der Akkumulator kann die zu verwendenden Validierer für eine Validierungsvorgang selbst ermitteln. Einen praktischen Ansatz Consul für ein verteiltes Datenvalidierungsnetzwerk zu verwenden ist in der Bachlor Thesis von Elias Backmund [117] enthalten.

Validierungsvorgang

Der zweite Schritt des Kommunikationsprozesses, die Validierung, ist in Abbildung 6.3 dargestellt. Das Gateway leitet die empfangenen Daten an die in der Policy-Datei definierten Validierer weiter. Die Zusammenführung der Eingangsdaten und der semantischen Einträge in der Policy-Datei kann z.B. über Schlüsselwörter in den Daten erfolgen.

Die Validierer sammeln die für den Validierungsprozess erforderlichen Daten. Dabei kann es sich entweder um einzelne Eingangsdaten oder um einen Datensatz von verschiedenen Sensoren handeln. Liegen alle erforderlichen Daten vor, führt der Validierer die Validierung durch und übermittelt das Ergebnis an den Akkumulator. Der Akkumulator akkumuliert die Validierungsergebnisse und führt auf dieser Grundlage seine weiteren Berechnungen durch.

Bei der Validierung kann neben der Verwendung eines einzelnen Validierers auch eine Korrelation zwischen verschiedenen Validierern erforderlich sein. Ein Beispiel hierfür ist die Verwendung von Validierungsalgorithmen, die auf mehreren Knoten basieren. Eine Auflistung der verschiedenen Möglichkeiten der Korrelation von Validierern findet sich in Abschnitt 6.6. Abhängig von der jeweiligen Korrelation muss die Policy-Datei um Informationen zum Algorithmus erweitert werden.

```
Registriert:
    Maschine Typ-112233: Temperatur
Validierer:
    - 38.246.231.241
    - 96.232.54.189
    ...

Registriert:
    Maschine Typ-778899: Steuersignale
Validierer:
    - 60.112.35.57
    - 96.94.233.56
    ...

Öffentlicher Schlüssel:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjlCCXDpuw1pf/G6BL
pYN8YEhxvNUUUWynFM6Xgtsx0Me+VvBAGu4tIOj/IBtkVz46evhMA0Fk
S9ampfDU1hPK6p/To9D4M7+uvaTpKHiOqt8rP4kcRPgwqVZ0b0MX/seY
FIzUQfMZOMmymCAfDK08VjQc9Sa3ek0061bmka+z+wIDAQAB
-----END PUBLIC KEY-----

Akkumulator Endpunkt:
192.168.7.7
```

Listing 6.2: Beispiel für eine Policy-Datei

```
Registriert:
    Maschine Typ-112233: Temperatur
    Maschine Typ-778899: Steuersignale
    ...

Öffentlicher Schlüssel:
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDjlCCXDpuw1pf/G6BL
pYN8YEhxvNUUUWynFM6Xgtsx0Me+VvBAGu4tIOj/IBtkVz46evhMA0Fk
S9ampfDU1hPK6p/To9D4M7+uvaTpKHiOqt8rP4kcRPgwqVZ0b0MX/seY
FIzUQfMZOMmymCAfDK08VjQc9Sa3ek0061bmka+z+wIDAQAB
-----END PUBLIC KEY-----

Akkumulator Endpunkt:
192.168.7.7
```

Listing 6.3: Beispiel für eine Subpolicy-Datei

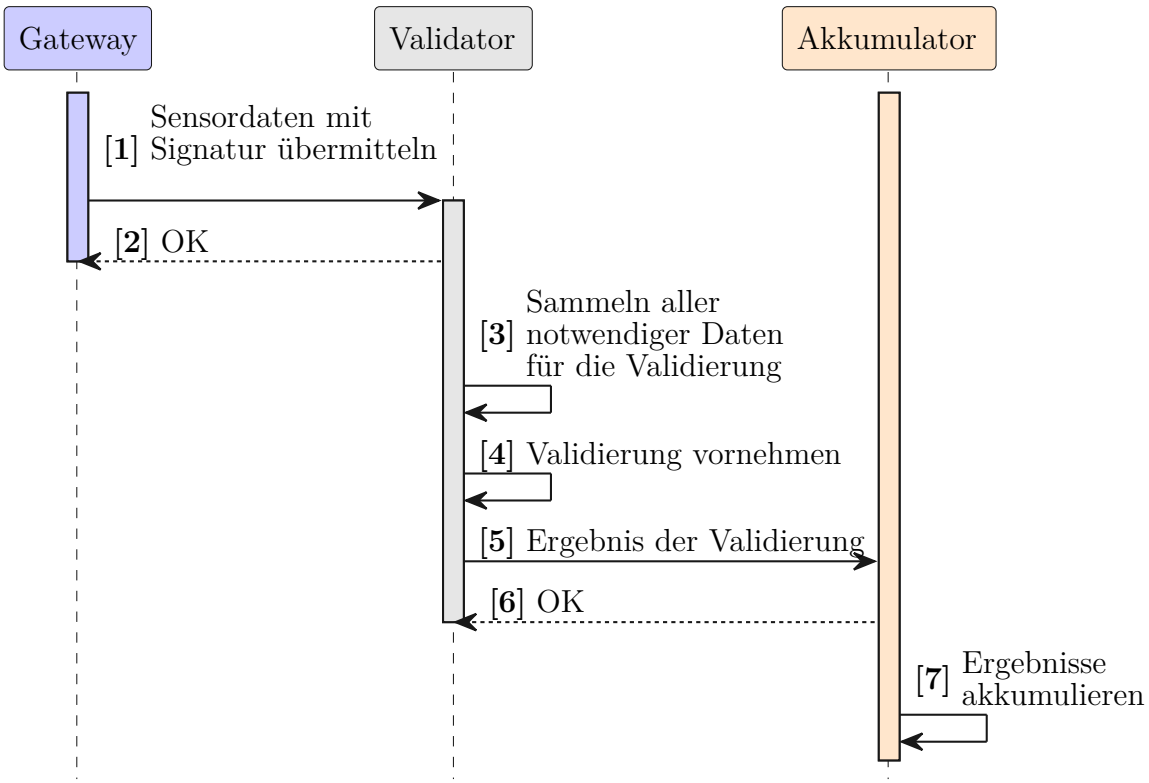


Abbildung 6.3.: Validierungsvorgang

Datenübertragung zum Akkumulator

Der letzte Teil des Kommunikationsprozesses (siehe Abbildung 6.4) umfasst die Weiterleitung der Eingangsdaten an den Akkumulator, damit dieser die Daten auch dann verwenden kann, wenn keine Validierungsergebnisse vorliegen. Dazu gibt es zwei Möglichkeiten: *a*) das Gateway sendet die Daten direkt an den Akkumulator und *b*) die Validierer senden ihre Daten an den Akkumulator. Der Nachteil der Möglichkeit *b*) ist, dass der Akkumulator die Daten von den Validierern selbstständig aggregieren muss und wenn bestimmte Validierungsergebnisse fehlen, dann fehlen auch die Daten. Ein weiteres Problem ist, dass ein korrumpierter Validierer falsche Daten weitergeben kann. Die Verwendung einer digitalen Signatur der Sensordaten könnte die Datenintegrität schützen und die Weitergabe falscher Daten verhindern.

In der vorliegenden wissenschaftlichen Arbeit wird die Möglichkeit *a*) verwendet, bei der das Gateway die Daten direkt an den Akkumulator weiterleitet. Im Anhang sind zwei Sequenzdiagramme (Abbildung A.1 und Abbildung A.2) beigefügt, die die Kommunikationsschritte 2) und 3) zusammenfassend darstellen.

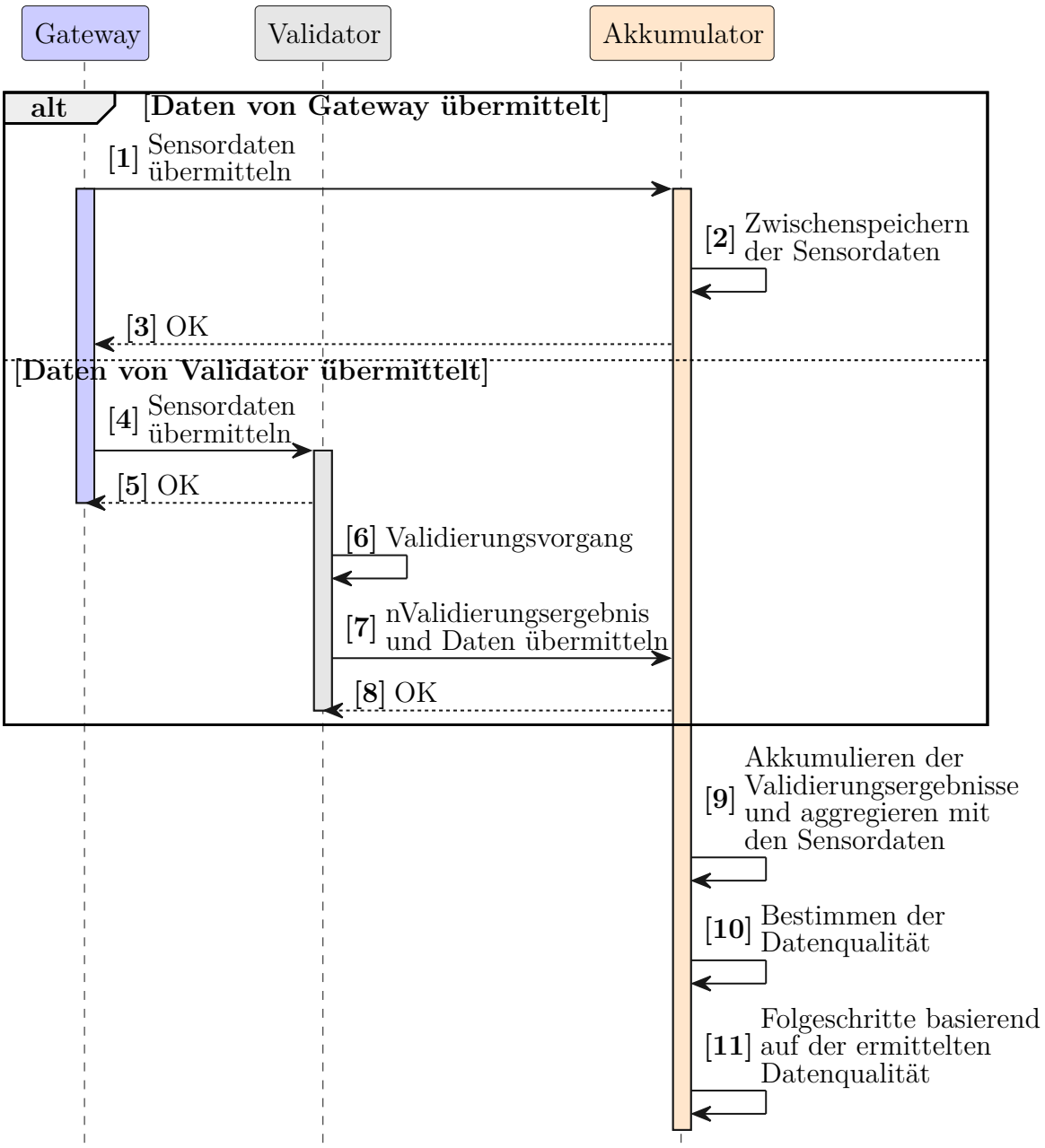


Abbildung 6.4.: Datenübertragung zum Akkumulator

Im Folgenden werden die Kommunikationsabläufe zwischen den einzelnen Komponenten im Detail beschrieben. Dazu gehört die Kommunikation zwischen:

- *Source zu Gateway (Abschnitt 6.3):* die Eingangsdaten werden unter Verwendung des Gateways netzwerkfähig aufbereitet.
- *Gateway zu Validierer (Abschnitt 6.4):* die Eingangsdaten werden an die ausgewählten Validierer weitergeleitet.
- *Gateway zu Akkumulator (Abschnitt 6.5):* die Eingangsdaten werden direkt

zum Akkumulator weitergeleitet.

- *Validierer zu Akkumulator (Abschnitt 6.6):* die Validierungsergebnisse werden an den Akkumulator zur Aggregation weitergeleitet.
- *Akkumulator zu Enterprise (Abschnitt 6.7):* das aggregierte Validierungsergebnis wird für die Bestimmung der Datenqualität herangezogen.

6.3. Source zu Gateway

Die Sourcen senden ihre Daten meist unverschlüsselt, im Klartext an das Gateway, da die Sensoren oft keine Verschlüsselungsmöglichkeiten besitzen. Die semantische Zuordnung zwischen Sensormesswert und Sensorposition an der Maschine, Datentyp des Sensorwertes usw. wird vom Gateway vorgenommen. OPC Unified Architecture [86] kann verwendet werden, um ein semantisches Datenmodell der Maschine und der gemessenen Sensorwerte zu erstellen. Zusätzlich zum semantischen Datenmodell unterstützt OPC Unified Architecture Defense-In-Depth Sicherheitsfunktionen und kann um ein auf dem Sicherheitsmodell basierendes Berechtigungskonzept [118] erweitert werden.

Die Verbindung zwischen Source und Gateway wird meist physikalisch direkt an der Maschine realisiert, da es sich bei der Source um einen Sensor handeln kann, der nicht netzwerkfähig ist. Die Verschlüsselung der Sensordaten sowie die digitale Signatur der Daten erfolgt grundsätzlich durch das Gateway, da dieses ein Kryptomodul enthält.

6.4. Gateway zu Validierer

Ein vertrauenswürdiges Gateway verfügt über ein signiertes Public-Key-Zertifikat des Gateway-Herstellers, einen entsprechenden privaten Schlüssel und eine Liste der Endpunkte der verschiedenen Akkumulatoren, für die Policy-Dateien verfügbar sind. Wenn ein Gateway Sensordaten empfängt, fügt es den Daten semantische Informationen hinzu und leitet die erweiterten Daten an die registrierten Validierer weiter. Um zu verhindern, dass gefälschte Daten eingeschleust werden, signiert das Gateway die ausgehenden Daten. Anschliessend überprüft ein Validierer die Identität des Gateways und die digitale Signatur der empfangenen Daten. Wenn die Identitätsprüfung und die Signaturprüfung korrekt sind, wird die Validierung der Daten ausgelöst, andernfalls werden die empfangenen Daten verworfen.

Grundsätzlich erlaubt der DDVN Ansatz nicht authentifizierten Validierern, Teil des Netzwerks zu werden. Daher kann eine angreifende Person einen neuen, nicht authentifizierten Validierer in das Netzwerk einschleusen. Wenn ein Akkumulator

diesen neuen korrumpierten Validierer verwendet, erhält die angreifende Instanz einen Teil der Daten und kann diese an nicht autorisierte Dritte weiterleiten.

Eine Verschlüsselung der Daten wird dennoch empfohlen, um ein einfaches Mitlesen der Kommunikation zwischen Gateway und Validierer zu verhindern.

6.5. Gateway zu Akkumulator

Das Gateway hat eine direkte verschlüsselte Verbindung zum Akkumulator. Diese Verbindung wird verwendet, um die von der Source erhaltenen Daten an den Akkumulator weiterzuleiten. Damit der Akkumulator die Daten der Source mit den Validierungsergebnissen in Beziehung setzen kann, gibt es mehrere Möglichkeiten:

1. *Identifikation über UUID:* Sowohl den an den Akkumulator gesendeten Eingangsdaten als auch den Validierungsergebnissen wird eine UUID hinzugefügt. Dies ermöglicht dem Akkumulator eine direkte Zuordnung zwischen den Daten und dem Validierungsergebnis.

2. *Relation über Zeitstempel:* Die Daten werden mit den Validierungsergebnissen des Akkumulators anhand von Zeitstempeln in Beziehung gesetzt und zugeordnet. Um einen eindeutigen Zeitstempel für Daten und Validierungsergebnisse zu ermöglichen, kann ein Vektoruhr-Algorithmus verwendet werden [119, 120]. Bei einem Validierungsergebnis, das mehrere Daten korreliert, muss der Vektoruhr-Algorithmus erweitert werden, da Zeitstempel mehrfach verwendet werden können. Hierzu wird auf die Arbeit von Zao et al. [121] verwiesen. Der aktualisierte Vektoruhrenalgorithmus ist nicht Teil dieser Arbeit.

3. *Eindeutige Validierer pro Datum:* Bei jeder Validierung werden neue Validierer vom Akkumulator ausgewählt. Dadurch weiss der Akkumulator, welcher Validierer zu welchem Zeitpunkt das Ergebnis liefert. Dies sollten keine zeitkritischen Daten sein, da die Verteilung von Aktualisierungen der Policy-Datei einen zeitlichen Overhead hat.

Die Verwendung eines eindeutigen Identifikators ist der zu bevorzugende Ansatz für einfache Daten. Wenn die Daten eine höhere Komplexität aufweisen, z.B. weil mehrere Daten von verschiedenen Gateways benötigt werden, ist eine Beziehung über Zeitstempel und Validierer sinnvoll. Die Verwendung von eindeutigen Validierern pro Datum ist vor allem in einer Hochsicherheitsumgebung anwendbar, da selbst bei einem Datenabfluss aus einem Datensatz die nachfolgenden Daten an einen anderen Validierer gesendet werden und somit die Korrelation der Daten stark erschwert wird.

6.6. Validierer zu Akkumulator

Die Initialisierung der Verbindung zwischen Validierer und Akkumulator wird durch den Akkumulator ausgelöst (siehe Sequenzdiagramm in Abbildung 6.2). Dazu wird ein Broadcast an alle Validierer gesendet. Die adressierten Validierer antworten auf den Broadcast des Akkumulators mit ihren Attributdateien (siehe Listing 6.1). Falls die IP zur Identifizierung des Validierers nicht ausreicht, kann einer der beiden folgenden Ansätze verwendet werden *a*) Ould-Ahmed-Vall et al. [122] distributed unique global ID assignment for sensor networks oder *b*) Lin et al. [123] SIDA: Self-organized ID Assignment. Um ein höheres Sicherheitsniveau zwischen einem Validierer und einem Akkumulator zu initialisieren, kann ein Virtual Private Network (VPN) eingerichtet werden.

Die Registrierung der Validierer dient zum Schutz vor dem Einschleusen falscher Validierungsergebnisse, da nur die Validierer, die zu einem bestimmten Zeitpunkt in der Policy-Datei enthalten sind, ihre Ergebnisse senden können. Wenn ein anderer Validierer sein Ergebnis sendet, wird das Ergebnis verworfen.

Für die Akkumulation von Validierungsergebnissen muss ein Akkumulator nur die Ergebnisse einer Teilmenge aller in der Policy-Datei aufgeführten Validierer erhalten. Daher kann eine bestimmte Anzahl von Akkumulatoren, die den Cluster verlassen haben, kompensiert werden. Dasselbe gilt für den Ausfall eines ganzen Clusters. Für die Aggregation der Validierungsergebnisse gibt es verschiedene Möglichkeiten, wie z.B. Mittelwert, logische Operatoren (AND, OR), etc. (zwei clusterabhängige Funktionen sind in Kapitel 8 aufgezeigt). Die Wahl der Aggregationsfunktion hängt von der geforderten Sicherheit des DDVNs sowie von der Darstellung des Validierungsergebnisses ab. Für die Darstellung des Validierungsergebnisses stehen zwei Darstellungsformen zur Verfügung: 1) ein logischer Wert mit dem Ergebnis OK oder NICHT OK und 2) ein numerischer Wert im Bereich $[0, 100]$.

Bei der Verwendung von Validierern gibt es eine Vielzahl von Korrelationsmöglichkeiten, d.h. bei der Verwendung von mehr als einem Validierer müssen die Validierungsergebnisse miteinander in Beziehung gesetzt werden. Zum einen kann die Aggregation der einzelnen Ergebnisse im Akkumulator erfolgen, zum anderen gibt es Zwischenergebnisse in einem Algorithmus, die für die weiteren Berechnungen verwendet werden müssen.

Die einfachste Struktur ist in Abbildung 6.5 dargestellt. Das Gateway übermittelt die Daten an alle Validierer $(v_1, v_2, ..., v_n)$, die in der Policy-Datei aufgeführt sind. Die Validierer berechnen dann die Ergebnisse parallel und übermitteln ihre Ergebnisse unabhängig voneinander an den Akkumulator. Der Akkumulator sammelt die mindestens erforderliche Anzahl von Validierungsergebnissen und berechnet daraus den Datenqualitätswert.

Bei der sequentiellen Validierung (siehe Abbildung 6.6) werden die Validierer hintereinander geschaltet. Wenn ein Validierungsergebnis ausbleibt, kann der

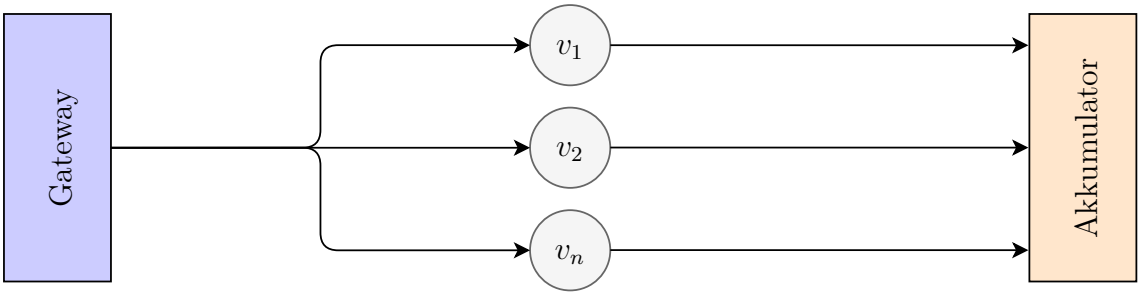


Abbildung 6.5.: Parallele Validierung mit direkt Übertragung an den Akkumulator

nachfolgende Validierer keine Validierung durchführen. Ein Beispiel hierfür ist ein Distributed Deep Neuronal Network (DDNN), bei dem ein Layer seine Validierungsergebnisse nicht mehr weitergibt. Um dieses Problem zu vermeiden, kann die sequentielle Validierung mit der parallelen Validierung kombiniert werden. Eine solche Kombination ist in Abbildung 6.7 dargestellt. Hier gibt es mehrere Validierer v_1 und v_2, die parallel eine Berechnung durchführen und das Ergebnis an v_n übergeben. v_n kann eine bestimmte Zeit t oder eine bestimmte Anzahl vorhergehender Validierungsergebnisse v_{result} abwarten und dann seine eigene Validierung auf Basis der erhaltenen Ergebnisse durchführen. Wenn ein Validierer seine Ergebnisse nicht rechtzeitig weitergibt (z.B. aufgrund eines Ausfalls), kann der nachfolgende Validierer ohne dessen Ergebnisse weiterarbeiten.

Wenn sequentielle oder aufbauende Validiererstrukturen verwendet werden, muss jedes Teilergebnis signiert und an den nächsten Validierer weitergegeben werden. Dadurch wird die Nachvollziehbarkeit und Integrität der Validierungsergebnisse sichergestellt.

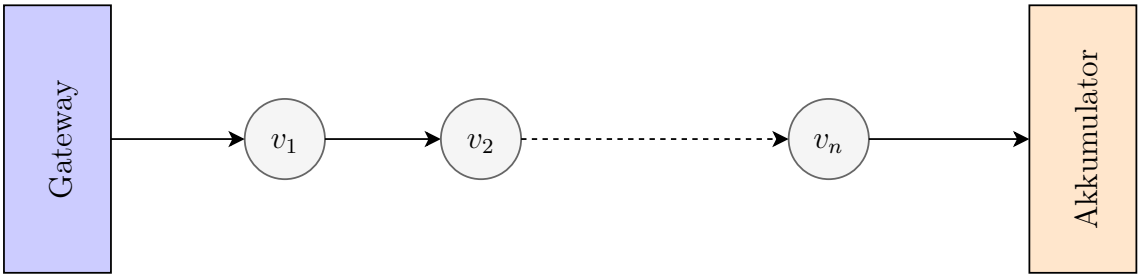


Abbildung 6.6.: Sequentielle Validierung aufbauend auf vorherigen Validierungsergebnissen

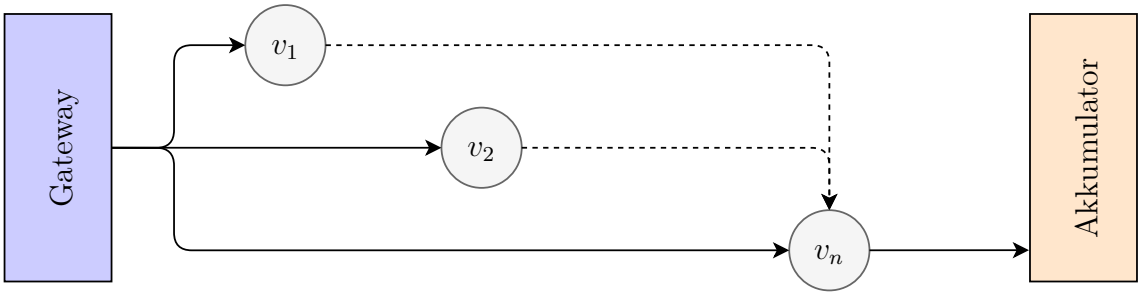


Abbildung 6.7.: Parallele Validierung mit Übertragung der Validierungsergebnisse an die nachfolgenden Validierer

Wenn ein Angriff alle Validierer einer Validierungsschicht korrumpiert hat, wird das Validierungsergebnis nicht an den Akkumulator weitergegeben. Um dies zu verhindern, können neben dem endgültigen Validierungsergebnis auch die jeweiligen Zwischenergebnisse der Validierer an den Akkumulator weitergeleitet werden (siehe Abbildung 6.8). Dadurch wird sichtbar, welche Ebene ihre Ergebnisse nicht weitergibt bzw. welche Ebene ein Problem hat. Übermittelt v_1 Ergebnisse an den Akkumulator, v_2 aber nicht, dann überträgt entweder v_1 seine Ergebnisse nicht an v_2 oder v_2 überträgt keine Ergebnisse.

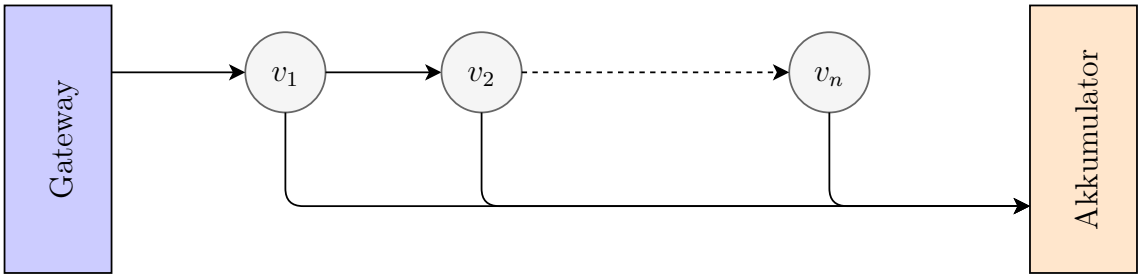


Abbildung 6.8.: Zwischenergebnisse der Validierung an den Akkumulator übermitteln

Um sicherzustellen, dass kein Validierer seine Ergebnisse absichtlich zurückhält oder grössere Manipulationen vornimmt, kann eine Trusted Third Party (TTP) verwendet werden. In Abbildung 6.9 ist eine TTP grün dargestellt. In diesem Fall läuft die gesamte Kommunikation zwischen den Validierern über die TTP. Es wäre auch denkbar, dass die TTP die Kommunikation zum Akkumulator durchführt. Damit wäre sichergestellt, dass der Validierer v_n dem Akkumulator keine Validierungsergebnisse vorenthält.

Der Aufbau einer TTP ist sehr aufwendig, daher kann der Akkumulator selbst als TTP verwendet werden. Die Performance und der Kommunikationsaufwand müssen bei dieser Implementierung genauer betrachtet werden.

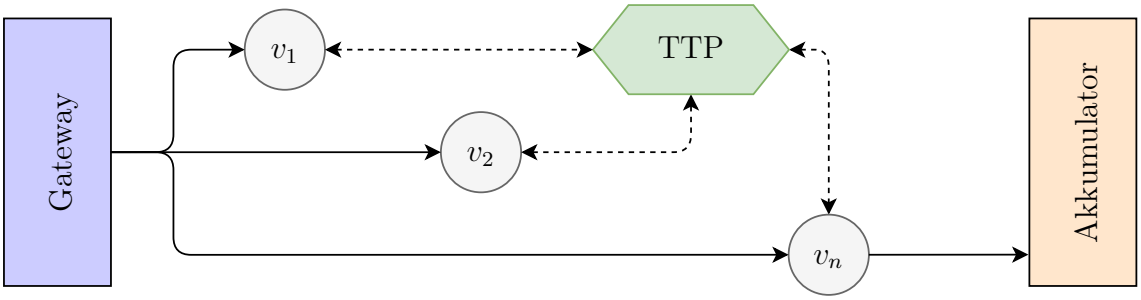


Abbildung 6.9.: Kommunikation der Validierer über eine Trusted Third Party

6.7. Akkumulator zu Enterprise

Ein Akkumulator ist ein Server (siehe Unterabschnitt 5.2.5), der bereits Teil des Unternehmensnetzwerks sein kann. Die Kommunikation zwischen dem Akkumulator und anderen Servern im Unternehmensnetzwerk kann über Maschine-Maschine-Protokolle wie MQTT und OPC Unified Architecture oder über Protokolle wie HTTP und HTTPS erfolgen.

Sollte das Unternehmen Zweifel an den akkumulierten Validierungsergebnissen der Akkumulatoren haben, können mehrere Akkumulatoren herangezogen werden. Die Akkumulatoren können dann selbstständig einen Konsens finden und dieses Ergebnis weitergeben. Für die Konsensfindung gibt es eine Vielzahl von Konsensverfahren wie *a*) Proof of Work (PoW), *b*) Proof of Stack (PoS), *c*) Delegated Proof of Stack (DPoS), *d*) Practical Byzantine Fault Tolerance (PBFT) und *e*) Raft. Ein Vergleich der verschiedenen Algorithmen wurde von Mingxiao et al. [124] (siehe Tabelle 6.1) durchgeführt.

Tabelle 6.1.: Vergleich der Konsensverfahren (PoW, PoS, DPoS, PBFT und RAFT) [124]

	Konsensverfahren				
Charakteristik	**PoW**	**PoS**	**DPoS**	**PBFT**	**RAFT**
Byzantinische Fehlertoleranz	50%	50%	50%	33%	N/A
Resilienz	50%	50%	50%	33%	50%
Verifikations-geschwindigkeit	$> 100s$	$< 100s$	$< 100s$	$< 10s$	$< 100s$
Durchsatz	< 100	< 1000	< 1000	< 2000	> 10000
Skalierbarkeit	hoch	hoch	hoch	niedrig	niedrig

Alternativ haben Bach et al. [125] weitere Konsensverfahren wie Ripple Protocol Consensus Algorithm (RPCA), Cardano's Ouroboros, Stellar Consensus Protocol (SCP) und Proof of Importance (PoI) untersucht. Abhängig vom Anwendungsfall

kann eines der unteren Konsensverfahren ausgewählt werden. Das Verwenden eines reinen PoW Ansatzes wird im Umfeld von privaten bzw. firmeninternen Umgebungen nicht empfohlen, da es einen sehr rechenintensiven Ansatz darstellt [126]. Eine Möglichkeit den Energieverbrauch von PoW zu minimieren bietet Green PoW [127]. Hierbei wird der eigentlich Mining Vorgang in zwei Epochen aufgeteilt wodurch der Energieverbrauch halbiert wird.

7. Sicherheitsanalyse

Im Folgenden wird die Bedeutung von Sicherheitsanalysen anhand der Auswirkungen erfolgreicher Angriffe auf Unternehmen aufgezeigt. Anschliessend werden verschiedene Arten der Bedrohungsmodellierung und Werkzeuge zur Bedrohungsmodellierung erläutert. Für die Analyse der DDVN wird eine STRIDE Analyse beschrieben und durchgeführt. Ausserdem werden verschiedene Angriffsarten und deren Auswirkungen auf das DDVN aufgelistet. Weiterhin werden DDVN-spezifische Faktoren 1) die Ursachen für Fehlverhalten, 2) die Arten von Algorithmenstrukturen, 3) die gefährdeten Bereiche bei einem Angriff und 4) das vorhandene Wissen bei einem Angriff und dessen Auswirkungen beschrieben. Abschliessend beschäftigt sich das Kapitel mit DDVN-spezifischen Metriken. Zunächst im Bereich der Datenqualität und nachfolgend im Bereich der Sicherheitsfaktoren. Dazu gehören der Korruptionsfaktor und der Auswirkungsfaktor sowie die Erfolgswahrscheinlichkeit des Angriffs und die Auswahlwahrscheinlichkeit der verwendeten Validierer. Die Sicherheitsfaktoren werden im Kapitel 8 zur Bestimmung der Schutzfähigkeit verwendet.
Ein Teil der folgenden Inhalte ist in komprimierter Form bereits in den Papern [74, 128, 129] publiziert.

7.1. Arten von Sicherheitsanalysen

Die Bestimmung des Sicherheitsniveaus eines Systems erfordert eine objektive Betrachtung. Hierzu gibt es eine Vielzahl von Risikoanalysen und Bedrohungsmodellierungen, die auf unterschiedlichen Sichtweisen basieren (z.B. Motivation und Ziele des Angriffs oder Umgang mit den Daten).

Eine Einteilung in vier Kategorien der Gefährdungsmodellierung wurde von Tatam et al. [130] vorgenommen. Die Kategorien sind: Vermögenszentriert, Bedrohungszentriert, Systemzentriert und Datenzentriert. Abbildung Abbildung 7.1 zeigt die vier Betrachtungsweisen mit den wichtigsten Unterpunkten.

Bei der vermögenszentrierten Betrachtung stehen die finanziellen Auswirkungen im Vordergrund. Ein zentraler Punkt ist dabei die Risikobetrachtung anhand einer Risikoanalyse mit der Ermittlung der Eintrittswahrscheinlichkeit bestimmter Szenarien [131], deren Folgen bei Eintritt sowie deren Auswirkungen auf andere Komponenten.

Die bedrohungszentrierte Sichtweise stellt den Angriff in den Mittelpunkt und versucht anhand verschiedener Profile von angreifenden Personen mögliche Schwachstellen im System zu identifizieren. Als Vorlage für die Profile können die sechs Definitionen: 1) NutzerIn, 2) InsiderIn, 3) HacktivistIn, 4) TerroristIn, 5) cyberkriminelle Person und 6) Staat von Rocchetto et al. [132] verwendet werden. Für diese Definition wurde eine Literaturrecherche durchgeführt.

Eine weitergehende Betrachtung erfolgt auf Basis des bestehenden Systems. Dazu gehören die Umsetzung des Netzwerkes mit der entsprechenden Segmentierung, die installierten Hardwarekomponenten sowie die Aktualisierung der Systemkomponenten. Darüber hinaus wird die Software einer Analyse unterzogen. Dazu gehören das eigentliche Design (Monolith/Microservice, ereignisbasierte/datenbasierte Architektur, Verschlüsselungsalgorithmen, etc.) und die eigentliche Implementierung (wurde der Programmcode kopiert, entspricht der Programmcode den vorgegebenen Standards, etc.) [133, 134]. Ein weiterer wichtiger Punkt bei der systemzentrierten Betrachtung ist das zur Verfügung gestellte Betriebssystem. Werden verschiedene Betriebssysteme vom System angeboten, so muss sichergestellt werden, dass jedes davon stets aktuell und mit den neuesten Patches ausgestattet ist.

Die vierte Sichtweise stellt die vorhandenen Daten in den Mittelpunkt. Hier werden der Zugriff auf die Daten, z.B. über ein Rollenkonzept, die sichere Übertragung (Verschlüsselung) und die resiliente Speicherung (Replikation) im Detail analysiert. Ziel ist es, dass Unbefugte keinen Zugriff auf die Daten haben und diese nicht manipulieren oder löschen können.

Für die gegebenen Ansätze zur Modellierung von Gefährdungen haben Tatam et al. [130] zusätzliche Modellierungswerkzeuge aufgenommen. Diese Modellierungswerkzeuge wurden in der vorliegenden Arbeit in Abbildung Abbildung 7.2 zusammengefasst. Darüber hinaus wurden die vier Betrachtungsweisen Vermögenszentriert, Bedrohungszentriert, Systemzentriert und Datenzentriert verwendet.

Für das vorliegende DDVN bieten sich vor allem system- und datenzentrierte Modellierungen an, da es sich um ein Datenvalidierungsnetz handelt. Darüber hinaus wird das DDVN nur theoretisch behandelt, weshalb anwendungsfallbasierte Modellierungen ebenfalls wegfallen. Daher wird eine allgemeine STRIDE Modellierung [135] durchgeführt, um aufzuzeigen, welche Risiken das System bietet und wie diese gemindert werden können. Darüber hinaus werden wichtige Punkte für das DDVN identifiziert und spezifische Definitionen für die Datenqualität und den Korruptionsfaktor vorgenommen.

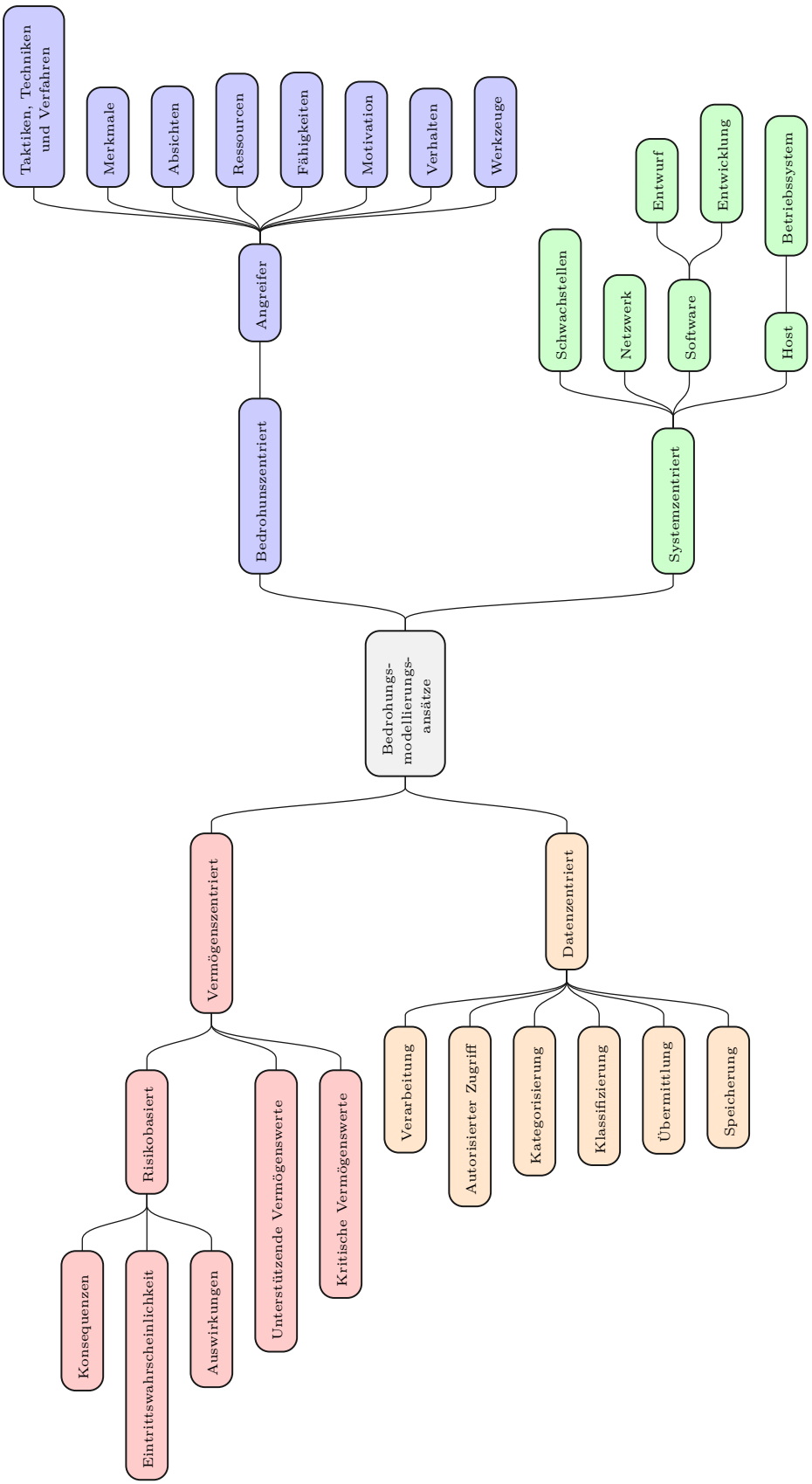


Abbildung 7.1.: Bedrohungsmodellierungsansätze[130]

Abbildung 7.2: Bedrohungsmodellierungswerkzeuge

7.2. STRIDE Analyse

Die Definition und Analyse von Sicherheitszielen für ein System ist notwendig, um das System zu entwerfen und eine Risikobewertung durchzuführen. Die Firma Microsoft schlägt hierfür die Methode STRIDE [135] vor. Dabei wird ein System auf die sechs Bedrohungen 1) Verschleierung (Spoofing), 2) Verfälschung (Tampering), 3) Verleugnung (Repudiation), 4) Offenlegung von Geheimnissen (Information Disclosure), 5) Dienstverweigerung (Denial of Service) und 6) Erhöhung der Privilegien (Elevation of Privilege) untersucht. Die Bedrohungen entsprechen den Sicherheitszielen: Authentifizierung, Integrität, Verbindlichkeit, Vertraulichkeit, Verfügbarkeit und Autorisierung. Im Folgenden werden die Sicherheitsziele kurz beschrieben und in Bezug auf DDVN analysiert.

7.2.1. Authentifizierung

Authentifizierung beschreibt den Nachweis der Identität. Nach erfolgreicher Authentisierung ist sichergestellt, dass das Gegenüber identifiziert ist.

Der Nachweis der Identität von Kommunikationspartnern dient dazu, Wirtschaftsspionage und das Einschleusen gefälschter Daten zu verhindern. Je mehr Teilnehmer ein Netzwerk hat, desto mehr Identitätsnachweise sind erforderlich. In einem DDVN gibt es viele verschiedene Teilnehmer: Sources, Gateways, Validators und Sinks, deren Anzahl je nach Produktionsgrösse stark variieren kann. Die aktuelle Architektur unterstützt nur die Identifizierung eines Gateways. Ein Sybil-Angriff [136] kann daher nicht direkt verhindert werden, weshalb der Einsatz einer Sybil-Angriffserkennung [137, 138, 139] notwendig ist. Die Angriffserkennung kann Teil eines Intrusion Detection Systems (IDS) oder eines Intrusion Prevention Systems (IPS) sein.

Eine Auflistung der verschiedenen Bedrohungen für die Authentifizierung und deren Gegenmassnahmen sind in Tabelle 7.1 und Tabelle 7.2 gegeben.

Tabelle 7.1.: Gefahren für die Authentifizierung und deren Auswirkung

Nr.	Gefahr
AtG-1	Die Source ist nicht authentifiziert und spielt falsche Daten ein.
AtG-2	Der Validierer ist nicht authentifiziert und spielt falsche Validierungsergebnisse ein.
AtG-3	Der Validierer ist nicht authentifiziert und berechnet kein Validierungsergebnis.

Tabelle 7.1.: Gefahren für die Authentifizierung und deren Auswirkung

Nr.	Gefahr
AtG-4	Der Validierer ist nicht authentifiziert und gibt die erhaltenen Daten, für die Validierung, an Dritte weiter.
AtG-5	Der Akkumulator vertraut dem Validierungsergebnis nicht, da die verwendeten Validierer nicht authentifiziert sind.
AtG-6	Es wird eine Vielzahl an korrumpierten Validierer verbunden und dadurch das Netzwerk korrumpiert.

Tabelle 7.2.: Gegenmassnahmen für die Gefahren der Authentifizierung

Nr.	Gefahren-behandlung	Gegenmassnahme
AtM-1	**AtG-{1,2,3,4,5}**	Die Source und der Validierer müssen zuvor authentifiziert werden z.B. mittels PMI und Zertifikaten. Dadurch besitzt jede Komponente eine eindeutige Identität die in Kombination mit digitalen Signaturen einen Schutz gegen Identitätsdiebstahl dargstellt.
AtM-2	**AtG-1**	Aufgrund einer Verhaltensanalyse wird festgestellt, dass sich die Source anormal verhält. Daraus abgeleitet kann die Source von der Vorhersage beim Akkumulator ausgeschlossen werden. Alternativ kann die Source ausgetauscht werden.
AtM-3	**AtG-{2,3}**	Aufgrund einer Verhaltensanalyse wird festgestellt, dass sich der Validierer anormal verhält. Daraus abgeleitet kann der Validierer von der Aggregation der Datenqualität beim Akkumulator ausgeschlossen werden.

Tabelle 7.2.: Gegenmassnahmen für die Gefahren der Authentifizierung

Nr.	Gefahren-behandlung	Gegenmassnahme
AtM-4	**AtG-1**	Die gelieferten Daten der Source sind ausserhalb des erlaubten Bereichs und werden vom Validierer als nicht korrekt eingestuft. Hierbei kann ein regelbasierter Ansatz verwendet werden. Alternativ können Neuronale Netze, Entscheidungsbäume, etc. verwendet werden.
AtM-5	**AtG-{1,2,3,5}**	Die Daten werden mit anderen Daten gemittelt, sodass ein verfälschter Wert einen geringeren Einfluss besitzt. Sollte trotz Mittelwertbildung der Ausreisser einen zu starken Einfluss auf das Ergebnis aufweisen kann stattdessen der Median oder ein Mittelwert exklusive einer definierten Menge an maximalen und minimalen Werten verwendet werden.
AtM-6	**AtG-{2,5}**	Es werden Testdaten eingespielt, um zu kontrollieren, ob der Validierer die korrekten Ergebnisse liefert. Sollten die Ergebnisse des Validierers abweichen, dann kann ein Rückschluss auf ein anormales Verhalten erfolgen.
AtM-7	**AtG-4**	Es wird nur das Minimum an benötigten Daten an den jeweiligen Validierer weitergeleitet. Auch bei einer erfolgreichen Korruption eines Validierers ist dadurch nur ein minimaler Informationsabfluss gegeben.
AtM-8	**AtG-4**	Es wird homomorphe Verschlüsselung verwendet. Dabei führt der Validierer seine Validierungsoperationen auf verschlüsselten Daten aus und bekommt keinen Einblick in die rohen Daten. Das Ergebnis der Validierungsoperation auf die verschlüsselten Daten ist dabei das Ergebnis der Validierung.

Tabelle 7.2.: Gegenmassnahmen für die Gefahren der Authentifizierung

Nr.	Gefahren-behandlung	Gegenmassnahme
AtM-9	**AtG-4**	Sollte der Validierer Teil eines mehrstufigen Algorithmus (z.B. ein Neuronales Netzwerk mit mehreren Layern) sein, dann sind die Daten an einem mittleren Knoten meistens nicht interpretierbar, da diese von den vorherigen Knoten bereits abgeändert wurden.
AtM-10	**AtG-6**	Es wird ein Konsensmechanismus oder Ähnliches verwendet, sodass der Aufwand und die Kosten für einen Angriff erheblich steigen. Der Angriff muss einerseits eine Mindestanzahl an Validierern korrumpieren und andererseits ausreichend korrumpierte Validierer vom Akkumulator für einen bestimmten Validierungsprozess ausgewählt bekommen.
AtM-11	**AtG-6**	Es wird ein geschütztes firmeninternes Netzwerk verwendet, sodass der Angriff zusätzlich die allgemeinen Schutzmechanismen überwinden muss.
AtM-12	**AtG-6**	Die Möglichkeiten sich in das DDVN zu verbinden wird minimal gehalten, zum Beispiel Netzwerkanschlüsse werden hardwaretechnisch verschlossen und WLAN Verbindungen sind mit Passwörtern sowie MAC Filtern geschützt.

7.2.2. Integrität

Die Integrität stellt sicher, dass die Daten im System nicht manipuliert wurden. Dies schliesst das Ändern, Löschen oder Einfügen von Daten ein.

Um sicherzustellen, dass ein Angriff nicht alle Daten der Sources löscht, wird in dieser Arbeit das Gateway als vertrauenswürdig betrachtet. Dies wird erreicht, indem nur zertifizierte und signierte Gateways verwendet werden, die jeweils ein bekanntes Zertifikat für die Signatur der Daten verwenden. Die erfassten Daten der Sources sowie die Validierungsergebnisse können jedoch weiterhin korrumpiert werden, sofern

keine zusätzlichen Sicherheitsmechanismen in Betracht gezogen werden.

Eine Auflistung der unterschiedlichen Gefahren auf die Integrität sowie deren Gegenmassnahmen sind in Tabelle 7.3 und Tabelle 7.4 gegeben.

Tabelle 7.3.: Gefahren für die Integrität und deren Auswirkung

Nr.	Gefahr
ItG-1	Die Source spielt falsche Daten ein (siehe auch AtG-1). Hierbei ist die Source entweder korrumpiert worden oder eine korrupte Source angeschlossen worden.
ItG-2	Der Validierer spielt falsche Validierungsergebnisse ein (siehe auch AtG-2).

Tabelle 7.4.: Gegenmassnahmen für die Gefahren der Integrität

Nr.	Gefahren-behandlung	Gegenmassnahme
ItM-1	**ItG-{1,2}**	Die versendeten Daten müssen digital signiert und mit einer Checksumme erweitert sein. Das Signieren von Daten spielt stark mit der Authentifizierung zusammen.

7.2.3. Verbindlichkeit

Durch die Verbindlichkeit wird sichergestellt, dass die Teilnehmer des Systems nicht in der Lage sind, bestimmte Handlungen im Nachhinein zu leugnen oder zurückzuziehen.

Die Validierungsergebnisse und die Eingabedaten sind die wesentlichen Informationen im Netzwerk. Daher muss der Urheber dieser Informationen eindeutig identifiziert werden, z.B. durch digitale Signaturen. Ein mögliches Angriffsszenario wäre, die Verbindung zum Netzwerk zu trennen und sich mit einem neuen öffentlichen/privaten Schlüsselpaar wieder anzumelden. Die Rückverfolgung eines Angriffs ist somit ohne weitere Sicherheitsmassnahmen nicht möglich. Aus diesem Grund können Vertrauensstufen verwendet werden, die sich auf die Dauer der Teilnahme eines Validierers beziehen.

Eine Auflistung der verschiedenen Bedrohungen für die Verbindlichkeit und deren Gegenmassnahmen sind in Tabelle 7.5 und Tabelle 7.6 gegeben.

Tabelle 7.5.: Gefahren für die Verbindlichkeit und deren Auswirkung

Nr.	Gefahr
VbG-1	Eine Source liefert konstant falsche Werte und es ist nicht nachvollziehbar, von welcher Source die falschen Werte stammen.
VbG-2	Ein Validator liefert konstant falsche Ergebnisse und es ist nicht nachvollziehbar, von welchem Validator die falschen Ergebnisse stammen.

Tabelle 7.6.: Gegenmassnahmen für die Gefahren der Verbindlichkeit

Nr.	Gefahren-behandlung	Gegenmassnahme
VbM-1	**VbG-{1,2}**	Die Werte und Ergebnisse sind von der jeweiligen Komponente digital signiert, so dass keine Möglichkeit der Anfechtung besteht.

7.2.4. Vertraulichkeit

Die Vertraulichkeit stellt sicher, dass kein unbefugter Zugriff auf schutzwürdige Daten erfolgt.

Die Kommunikation zwischen Validierer und Akkumulator sollte verschlüsselt und für Dritte unlesbar sein, da sonst die Möglichkeit des Know-how-Diebstahls besteht. Ein Angriff darf nicht in der Lage sein, aus den gesammelten Daten Rückschlüsse auf Validierungsergebnisse und Maschinenkonfigurationen zu ziehen. Derzeit ist dieses Sicherheitsziel nicht erfüllt, da bei einem Angriff ein eigener Validator implementiert und alle vom Gateway übertragenen Werte ausgelesen werden können. Die Verbindung von einem nicht authentifizierten Validator wird nicht abgewiesen. Für spätere Versionen des DDVN wird ein Token-basiertes Wiederverschlüsselungsschema vorgesehen [140, 141]. Neben der Authentifizierung aller eingesetzten Validierer kann auch ein Zero-Knowledge-Validierungsverfahren eingesetzt werden. Dabei kann die Validierung an verschlüsselten Daten durchgeführt werden, ohne dass der Validierer die eigentlichen Daten erhält.

Eine Auflistung der verschiedenen Bedrohungen der Vertraulichkeit und deren Gegenmassnahmen sind in Tabelle 7.7 und Tabelle 7.8 gegeben.

Tabelle 7.7.: Gefahren für die Vertraulichkeit und deren Auswirkung

Nr.	Gefahr
VtG-1	Die Kommunikation zwischen den einzelnen Komponenten wird abgefangen und ausgelesen. Betriebsgeheimnisse wie Maschineneinstellungen oder Informationen über Aufträge werden an unbefugte Dritte weitergegeben.
VtG-2	Ein Validierer ist korrumpiert und gibt die erhaltenen Daten zur Validierung an nicht autorisierte Dritte weiter.

Tabelle 7.8.: Gegenmassnahmen für die Gefahren der Vertraulichkeit

Nr.	Gefahren-behandlung	Gegenmassnahme
VtM-1	**VtG-1**	Die Kommunikation zwischen den einzelnen Komponenten wird verschlüsselt.
VtM-2	**VtG-1**	Bei Komponenten, die keine Verschlüsselung der Daten durchführen können, z.B. Sensoren ohne Verschlüsselungshardware, kann ein Gateway zur Verschlüsselung verwendet werden. Dabei muss die Hardware so installiert werden, dass ein direkter Zugriff auf die Hardware verhindert wird. Darüber hinaus sollten nur authentifizierte / vertrauenswürdige Gateways verwendet werden, da sonst über die neue Sicherheitskomponente ein neues Sicherheitsrisiko entstehen kann.
VtM-3	**VtG-2**	Es werden nur authentifizierte und autorisierte Validierer verwendet. Das kann mittels Zertifikaten in Kombination mit einer PMI umgesetzt werden.

7.2.5. Verfügbarkeit

Verfügbarkeit stellt sicher, dass Funktionen und Daten jederzeit zur Verfügung stehen, wenn sie von den benutzenden Personen oder den Systemen benötigt werden.

In Produktionsunternehmen ist der unterbrechungsfreie Betrieb von Maschinen absolut notwendig. Im Vereinigten Königreich fiel 2018 eine Maschine durchschnittlich fünf Mal aus, wodurch Kosten von mehr als 60.000£ [142] entstanden. Die Kosten hängen stark von der Unternehmensgrösse ab. Im Allgemeinen hat ein grösseres Unternehmen höhere Verluste als ein kleineres. Um die Ausfallzeit aufgrund eines Ausfalls des Validierungssystems zu minimieren, verwendet das vorgeschlagene clusterbasierte Validierungsnetzwerk mehrere Validierer und Akkumulatoren. Zusätzlich werden redundante Validierer und Akkumulatoren unterstützt.

Eine Auflistung der verschiedenen Bedrohungen für die Verfügbarkeit und deren Gegenmassnahmen sind in Tabelle 7.9 und Tabelle 7.10 gegeben.

Tabelle 7.9.: Gefahren für die Verfügbarkeit und deren Auswirkung

Nr.	Gefahr
VfG-1	Komponenten werden hardwaremässig so manipuliert, dass sie nicht mehr Teil des Netzes sind.
VfG-2	Eine korrumpierte Source wird so gesteuert, dass keine Daten mehr gesendet werden. Dadurch stehen keine Daten mehr für die Validierung und die Akkumulatoren zur Verfügung.
VfG-3	Ein korrumpierter Validator sendet keine Validierungsergebnisse, so dass der Akkumulator nicht mehr feststellen kann, ob bestimmte Daten vertrauenswürdig sind.
VfG-4	Ein Gateway fällt aus und sendet keine Daten mehr. Dieser Fall ist unabhängig vom DDVN, da auch in anderen Systemen die Sink ausfallen kann.
VfG-5	Ein Akkumulator fällt aus und kann keine Aggregation oder ähnliches durchführen. Dieser Fall ist unabhängig vom DDVN, da auch in anderen Systemen der Sink ausfallen kann.

Tabelle 7.10.: Gegenmassnahmen für die Gefahren der Verfügbarkeit

Nr.	Gefahren-behandlung	Gegenmassnahme
VfM-1	**VfG-1**	Die Komponenten sind hardwaremässig geschützt (z.B. abgeschlossener Schaltschrank), so dass eine Manipulation durch unbefugte Dritte nicht möglich ist.
VfM-2	**VfG-{2,4,5}**	Eine Komponente wird gespiegelt, sodass bei einem Ausfall einer Komponente auf eine alternative Komponente gewechselt werden kann.
VfM-3	**VfG-3**	Der Akkumulator verwendet mehrere Ergebnisse von verschiedenen Validierern. Wenn ein Validator keine Validierungsergebnisse sendet, können die anderen Validierungsergebnisse aggregiert werden. Darüber hinaus kann der Akkumulator Validatoren vollständig ersetzen.
VfM-4	**VfG-3**	Um ein Stillstand in der Produktion zu verhindern kann der Akkumulator zwischenzeitlich komplett auf Validierungsergebnisse verzichten oder eine eigene grobe Validierung vornehmen.

7.2.6. Autorisierung

Autorisierung beschreibt den Prozess der Vergabe von Rechten an Benutzerinnen, Benutzer oder Systemkomponenten. Besitzt ein System die erforderliche Berechtigung, kann es eine Funktion ausführen oder auf bestimmte Daten zugreifen.

Grundsätzlich arbeiten Authentifizierung und Autorisierung zusammen. Ohne Authentisierung werden meist keine Berechtigungen vergeben bzw. ohne Authentisierung werden die Berechtigungen nicht überprüft. Für die Zugriffskontrolle gibt es fünf verschiedene Varianten: 1) Erforderliche Zugriffskontrolle (MAC), 2) Benutzerdefinierte Zugriffskontrolle (DAC), 3) Rollenbasierte Zugriffskontrolle (RBAC), 4) Attributbasierte Zugriffskontrolle (ABAC) und 5) Risikobasierte Zugriffskontrolle (RBAC) [143].

Eine Auflistung der verschiedenen Bedrohungen für die Autorisierung und deren Gegenmassnahmen sind in Tabelle 7.11 und Tabelle 7.12 gegeben.

Tabelle 7.11.: Gefahren für die Autorisierung und deren Auswirkung

Nr.	Gefahr
AtG-1	Ein Komponente (Gateway,Validierer) erhält Daten für die sie keine Autorisierung besitzt.
AtG-2	Ein Validierer sendet Validierungsergebnisse für Daten, für die er keine Autorisierung besitzt.
AtG-3	Ein Validierer wird in einem Cluster ergänzt, für das der Validierer keine Autorisierung besitzt.
AtG-4	Der Akkumulator wählt einen falschen Validierer für die Validierung der Daten aus.

Tabelle 7.12.: Gegenmassnahmen für die Gefahren der Autorisierung

Nr.	Gefahren-behandlung	Gegenmassnahme
AtM-1	**AtG-1**	Der Sender sollte identifiziert und kontrolliert werden, weshalb dieser die Daten an einen falschen Empfänger versendet hat.
AtM-2	**AtG-1**	Die versendeten Daten werden so verschlüsselt, dass nur Komponenten die Daten entschlüsseln können, die auch hierfür autorisiert sind.
AtM-3	**AtG-2**	Der empfangende Akkumulator löscht die Ergebnisse ohne diese weiter in Betracht zu ziehen. Zudem könnten Rückschlüsse für eine Fehleranalyse auf den Sender erfolgen.
AtM-4	**AtG-{3,4}**	Die Cluster Validierer kontrollieren sich gegenseitig und entfernen, wenn nötig, unautorisierte Validierer aus dem Cluster.
AtM-5	**AtG-{2,3,4}**	Ein PMI mit Attribute Certificate (AC) wird verwendet, um sicherzustellen, dass ein Validierer die richtigen Berechtigungen für eine Validierungsoperation hat.

Zusammenfassend hat die STRIDE Bedrohungsanalyse 21 Bedrohungen und 26 Schutzmechanismen identifiziert. Die meisten Bedrohungen betreffen die Authentisierung mit 6 und die Verfügbarkeit mit 5 Bedrohungen. Im Gegenzug verfügt die Authentisierung mit 12 Schutzmechanismen auch über die meisten Möglichkeiten, um Verletzungen der Authentisierung zu begegnen. Es ist zu beachten, dass ein Teil der Schutzmechanismen gegen mehrere Bedrohungen eingesetzt werden kann. Beispielsweise kann die Verwendung eines PMI sowohl für die Authentifizierung als auch für die Autorisierung verwendet werden.

7.3. Angriffsarten und deren Auswirkung

Im Folgenden wird eine Menge an Cyberangriffen auf das DDVN betrachtet und deren Auswirkungen aufgezeigt. Hierfür wird die Liste der Cyberangriffe aus [144] als Referenz herangezogen und mit einigen weiteren Angriffen aus [145] ergänzt. Einige Angriffe, wie Cross-Site Scripting, Password Cracking oder Phishing, wurden weggelassen, da diese ohne Anwendungsfall nicht direkt dem DDVN in Verbindung gebracht werden können. Durch das Aufzeigen der Angriffe wird ein Verständnis über die verhinderten und erfolgreichen Angriffe geschaffen. Die vorliegende Aufzählung muss abhängig vom jeweiligen Anwendungsfall angepasst werden, weshalb diese keinen Anspruch auf Vollständigkeit aufweist.

Tabelle 7.13.: Angriffsart

Name	Beschreibung
	Eavesdropping
Angriff	Beschreibt das Ausspionieren einer Kommunikation mit dem Ziel Informationen zu erhalten. Dabei können die Informationen als geheim oder öffentlich eingestuft sein. Es gibt zwei verschiedene Arten von Eavesdropping *a*) *passiv:* hierbei werden die Daten mitgehört bzw. mitgelesen und *b*) *aktiv:* hierbei wird versucht manipulierte Daten einzuspielen und als rechtmässig auszugeben.
DDVN Auswirkungen	Der Ursprung des DDVNs sind manipulierte Daten seitens Sources. Somit ist das DDVN gegen falsche Eingangsdaten geschützt. In Bezug auf korrumpierte Validierungsergebnisse hat das DDVN ebenfalls eine Menge an Schutzmechanismen eingeführt (siehe Kapitel 8).

	SQL Injection
Angriff	Beschreibt das Einspeisen von bösartigen SQL Abfragen und daraus resultierend das Ändern und Löschen von bestehenden Daten oder das Einspielen falscher Daten.
DDVN Auswirkungen	Das DDVN macht keine Vorgaben bezüglich dem Verwenden von Datenbanken. Sollten solche auf den Validierern vorhanden sein, dann kann nicht ausgeschlossen werden, dass eine SQL Injection (SQLi) stattfinden kann. Hierfür kann der Ansatz *Prepared Statements* verwenden werden. Dabei wird der Angriff bei seinen SQL Abfragen limitiert und kann nur noch in einer vorbereiteten Abfragen seine Parameter setzen.
	Replay Attack
Angriff	*„Das Mitlesen der versandten Daten ermöglicht Replay aber auch Replay-Attacken, bei denen einmal zur Authentisierung benutzte Daten, wie z.B. verschlüsselte Passwörter, von einem Angreifer bei einem späteren Zugangsversuch wieder eingespielt werden.“* [146]
DDVN Auswirkungen	Die Komponenten des DDVN verhalten sich beim erneuten Einlesen der Daten unterschiedlich. Die Auswirkungen von Datenduplikaten werden durch die Aggregation mehrerer Ergebnisse minimiert.
	Denial of Service
Angriff	Beschreibt den Angriff auf ein System, bei dem das System unbrauchbar wird. Ein bekannter Angriff hierfür ist das Überladen eines Services mit Anfragen. Sollte die Anzahl der Anfragen zu gross werden, kann der Service nicht mehr zeitnah antworten und die Queue mit Anfragen wächst bis der Service nicht mehr fähig ist zu antworten.
DDVN Auswirkungen	Ein Denial of Service Angriff wird im DDVN unter Verwendung von mehreren Validatoren entgegengewirkt.
	Malware
Angriff	Beschreibt Software die dafür ausgelegt wurde einen Schaden zu erzeugen. Zu Malware zählen Viren [146], Würmer, Tojaner, etc.

DDVN Auswirkungen	Das DDVN ist grundsätzlich nicht vor Malware geschützt. Es können jedoch eine Firewall, ein Anomalieerkennungssystem, etc. verwendet werden. Zudem können mithilfe der Cluster (zum Beispiel anhand des Betriebssystems) Gruppierungen vorgenommen werden, die verhindern, dass alle Validierer gleichzeitig betroffen sind.
Daten Manipulation	
Angriff	Beschreibt das vorsätzliche Manipulieren von Daten zum eigenen Vorteil.
DDVN Auswirkungen	Beim vorliegenden DDVN können keine Daten manipuliert werden, da diese jeweils vom Sender signiert und mit einer Checksumme ergänzt werden.
Einspeisen korrumpierter Daten	
Angriff	Beschreibt das Einspielen von bösartigen oder fehlerhaften Daten in ein System.
DDVN Auswirkungen	Beim DDVN können korrumpierte Validierer falsche Ergebnisse an den Akkumulator weitersenden. Durch die Verwendung einer Aggregationsfunktion beim Akkumulator kann der Einfluss von manipulierten Ergebnissen minimiert werden. Ähnliche verhält es sich mit korrumpierten Sources, diese können falsche Daten weitergeben und dadurch einen wesentlichen Einfluss auf die Business Logik haben. Gerade im Bereich von korrumpierten Sources triggern die Validierer und markieren die falschen Daten als nicht vertrauenswürdig.
Spoofing	
Angriff	Beschreibt das Verschleiern der eigenen Identität, sodass andere Teilnehmer getäuscht werden. Häufig wird Spoofing verwendet, um geheime Informationen von anderen Teilnehmern zu erhalten.
DDVN Auswirkungen	Abhängig vom Aufbau des DDVN können unbekannte Validierer dem Netzwerk beitreten. Dadurch kann ein korrumpierter einige Daten im Netzwerk auslesen und an unerlaubte Dritte weitergeben. Im Fall von authentifizierten und autorisierten Validierern, ist das DDVN besser gegen Spoofing geschützt.

	Zero-Day Exploit
Angriff	Beschreibt einen Angriff auf ein System, bei dem der Hersteller des Systems entweder noch nichts von dem Angriff wusste oder die Sicherheitslücke noch nicht geschlossen hat. Die Sicherheitslücke selbst wird jedoch bereits bei Angriffen ausgenutzt oder ist öffentlich bekannt.
DDVN Auswirkungen	Das DDVN verwendet einen Cluster basierten Ansatz, um sich vor Zero-Day Exploits bei den Sources und den Validierern zu schützen.
	Manipulierte Hardware
Angriff	Beschreibt eine manipulierte Hardware, die für einen Angriff entwickelt wurde [147, 148]. Dabei kann es sich um externe Hardware wie Maus, Tastatur, USB-Stick etc. oder um direkt im Gerät eingebaute Hardware handeln. In den meisten Fällen wird die manipulierte Hardware für Spionagezwecke verwendet, z.B. um ein Passwort auszuspähen.
DDVN Auswirkungen	Werden die Sensoren hardwaretechnisch manipuliert, dann wird das durch den Validierungsvorgang ersichtlich, da die eingespielten Daten nicht mehr valide sind. Bei Angriffen auf die Validierer wird mittels Quorum Ansatz, etc. der Einfluss von korrumpierten Validierern minimiert.
	Adversarial Attack
Angriff	Beschreibt die Eingabe von speziell konstruierten Eingabedaten, so dass ein Klassifizierungssystem ein falsches Ergebnis vorhersagt, ohne dass die Eingabedaten offensichtlich beeinflusst wurden. Ein Beispiel ist die Eingabe eines veränderten Bildes, bei dem ein menschlicher Experte den Unterschied nicht erkennen kann [149].
DDVN Auswirkungen	Das DDVN verwendet unterschiedliche Validierer Ergebnisse, die auf unterschiedlichen Algorithmen basieren. Dadurch ist es möglich Adversarial Attacks zu erkennen, da bei einem Angriff eine bestimmte Menge Algorithmen getäuscht werden müssen.

7.4. Ursachen für Fehlverhalten der Validierer

Bei der Betrachtung eines DDVN befindet sich jeder Validierer auf einem anderen Knoten. Ein fehlerhaftes Verhalten eines Knotens muss nicht unbedingt bösartig sein, sondern kann auch durch fehlerhafte Konfigurationen etc. verursacht werden. Eine Auflistung der verschiedenen Fehlverhalten und deren Auswirkungen ist nachfolgend aufgeführt:

- *Absturz von Knoten:* Ein oder mehrere Knoten fallen aus und liefern keine Ergebnisse mehr. Je nach zugrundeliegender Netzwerkstruktur kann dies unterschiedliche Auswirkungen haben. Grundsätzlich sollten einzelne Ausfälle in einem verteilten Netzwerk durch Redundanzen aufgefangen werden. Ist dies nicht der Fall, muss der Akkumulator ohne Datenqualitätsbestimmung arbeiten.
- *Unterbrechung von Knoten:* Ein oder mehrere Knoten sind ausgefallen. Dies kann entweder durch einen Absturz oder durch ein Problem im Netzwerk verursacht werden. Wie bei einem Knotenabsturz muss der Akkumulator im schlimmsten Fall ohne Datenqualitätsbestimmung arbeiten.
- *Byzantinischer Fehler:* Die Knoten verhalten sich willkürlich. Das bedeutet, dass Daten richtig oder falsch klassifiziert werden können. In diesem Fehlerszenario kann der Akkumulator stark beeinträchtigt werden, d.h. gute Daten können verworfen und schlechte Daten für die Vorhersage verwendet werden. Ein willkürliches Verhalten der Validierer führt also zu einem willkürlichen Verhalten der Akkumulatoren.
- *Angriff:* Ein oder mehrere Knoten werden korrumpiert und liefern absichtlich manipulierte Daten. Wenn ein Angriff genügend Wissen (siehe Abschnitt 7.6) über das Gesamtsystem hat und genügend Validierer korrumpiert, kann er aktiv steuern, welche Daten von den Akkumulatoren für die Vorhersage verwendet werden. Damit kann auch das gesamte Vorhersageergebnis gesteuert werden.

7.5. Angriffsbereiche auf Validierer

Aufgrund unterschiedlicher Algorithmusstrukturen können unterschiedliche Bereiche als gefährdet definiert werden. Die verschiedenen Bereiche können sich auch überschneiden. Je nachdem, auf welchen Bereich ein Angriff Zugriff hat, ändert sich sein Einfluss auf den Algorithmus. Mögliche Gründe, warum nur ein bestimmter Bereich von einem Angriff betroffen ist, sind *a*) die Knoten sind über das Netzwerk verteilt und ein Angriff hat nur Zugriff auf einen bestimmten Teil des Netzwerks, *b*) die Knoten sind auf verschiedenen Betriebssystemen installiert und der Angriff kann nur auf einem bestimmten Betriebssystem durchgeführt

werden und *c*) einige der Knoten sind durch einen zusätzlichen Schutz, wie z.B. eine stärkere Firewall, geschützt und dadurch weniger anfällig für den Angriff. Im Folgenden werden verschiedene Angriffsarten aufgelistet.

- *None:* Der Validierer sind gut geschützt, so dass diese nicht angegriffen werden können.
- *Single:* Ein einzelner Validierer v_i kann angegriffen werden.
- *Level:* Eine ganze Ebene einer Baumstruktur (siehe v_2' und v_2'' in Abbildung 8.10) hat Schwachstellen und kann angegriffen werden. Wenn mehrere Ebenen angegriffen werden können, werden sie als *Subset* kategorisiert.
- *Subtree:* Ein vollständiger Teilbaum (siehe v_1, v_2' und v_2'' in Abbildung 8.10) ist angreifbar. Das kann ein Elternknoten und seine beiden Kindknoten sein, also ein Teilbaum über zwei Ebenen oder ein Elternknoten mit Kindknoten über mehrere Ebenen.
- *Teilmenge:* Eine Teilmenge von Knoten weist Schwachstellen auf und ist angreifbar. Die Teilmenge kann aus aufeinander folgenden Knoten oder aus Knoten bestehen, die nicht miteinander verbunden sind. Somit kann jeder mögliche Angriffsbereich (*None*, *Single*, *Level*, *Subtree* oder *All*) dargestellt werden.
- *All:* Der Angriff nutzt eine Sicherheitslücke, die jeden Knoten betrifft.

7.6. Auswirkung von Wissen bei einem Angriff

Ein Angriff auf ein System beinhaltet eine Vielzahl von Faktoren, die einen Vorteil verschaffen und daher aus sicherheitstechnischer Sicht berücksichtigt werden müssen. Faktoren, die vor einem Angriff berücksichtigt werden müssen, sind *a*) die Fähigkeiten einer angreifenden Instanz, *b*) die für einen Angriff verfügbare Hardware und *c*) das Wissen bei einem Angriff. Insbesondere bei der Betrachtung der Sicherheit des DDVN hat das Wissen bei einem Angriff einen erheblichen Einfluss. Das mögliche Wissen umfasst:

- *Oblivious:* Der Angriff besitzt keine Kenntnis über die Schicht, die Struktur, das erwartete Ergebnis oder die Daten innerhalb des DDVN. Daher kann ein erfolgreicher Angriff ein beliebiges Ergebnis liefern, und die Auswirkungen auf das des Resultat eines Akkumulators sind unbekannt.
- *Erwartetes Ergebnis eines Layers ist bekannt:* Der Angriff kennt das erwartete Ergebnis, indem er zum Beispiel die üblichen Operationen der beschädigten Schicht verwendet. Der Angriff kann somit das erwartete Ergebnis umkehren oder weiterleiten, kennt aber nicht die Auswirkungen auf das Gesamtergebnis, solange die beschädigte Schicht nicht die letzte Schicht des Algorithmus ist.

- *Erwartetes Ergebnis des DDVN ist bekannt:* Der Angriff kennt das erwartete Endergebnis und kann versuchen, die nachfolgenden Schichten nachzubauen. Im schlimmsten Fall für das DDVN kann der Angriff die Ergebnisse zwischen den Schichten erfolgreich reproduzieren und damit herausfinden, welche Validiererergebnisse an welcher Stelle im System eingefügt werden müssen, um das gewünschte Vorhersageergebnis zu erhalten.
- *Eingabedaten des DDVN sind bekannt:* Der Angriff kennt die Eingabedaten für das DDVN. Solange der Angriff keine zusätzlichen Informationen hat, sind die Auswirkungen auf das Resultat eines Akkumulators unbekannt.
- *Algorithmus-Struktur ist bekannt:* Eine angreifende Instanz ist in der Lage die geeigneten Knoten für einen Angriff zu bestimmen. Zum Beispiel ist die letzte Schicht eines DDNN das beste Ziel für einen Angriff, da diese Schicht das endgültige Validierungsresultat liefert. Handelt es sich bei der korrumpierten Schicht um eine Zwischenschicht eines DDNN, kann der Angriff den gesamten Algorithmus kontrollieren, wenn die Gewichte und die vollständige Struktur der nachfolgenden Schichten bekannt sind. Bei Algorithmen, die unterschiedlichen sequentiellen Abläufen folgen, zum Beispiel ein Distributed Decision Tree (DDT), muss der Angriff auch wissen, welcher Pfad im Entscheidungsprozess verfolgt wird.
- *Alles bekannt:* Die angreifende Instanz kann alle der zuvor genannten Inhalte nutzen um einen Angriff zu planen und durchzuführen.

In diesem Kapitel wird eine Sicherheitsanalyse des DDVN vorgestellt. Dabei werden vier verschiedene Kategorien der Gefährdungsmodellierung *a*) Vermögenszentriert, *b*) Bedrohungszentriert, *c*) Systemzentriert und *d*) Datenzentriert beschrieben. Eine detaillierte Analyse des DDVNs erfolgt mit Hilfe der STRIDE-Modellierung, die zu den systemzentrierten Ansätzen zählt. Insgesamt werden mit STRIDE 21 Bedrohungen und 26 Schutzmechanismen identifiziert. Neben der systemzentrierten Bedrohungsmodellierung werden 11 Cyberangriffe aus [144, 145] als Referenzangriffe herangezogen und deren Auswirkungen auf DDVNs beschrieben. Dadurch wird ein Verständnis für die zentralen Schutzvektoren des DDVNs geschaffen und mögliche Schwachstellen aufgezeigt.
Neben der STRIDE Modellierung und der Betrachtung von Referenzangriffen werden drei zentrale Sicherheitsmerkmale des DDVNs beschrieben. Dies sind *a*) die Ursache für das Fehlverhalten eines Validierers, die nicht ausschliesslich auf einem Angriff beruhen kann, *b*) die Angriffsbereiche, die eine Abhängigkeit vom verwendeten Validierungsalgorithmus aufweisen und *c*) das verwendete Wissen eines Angriffs.

8. Ansätze zur Erhöhung der Sicherheit

In einem System gibt es verschiedene Komponenten, die geschützt werden müssen. Die STRIDE Sicherheitsanalyse aus Abschnitt 7.2 enthält eine Analyse anhand von Bedrohungen bzw. den zugehörigen Sicherheitszielen. Im Folgenden werden vier Schutzmechanismen für das DDVN erarbeitet.
Ein Teil der folgenden Inhalte wurde bereits in komprimierter Form in den Papern [150, 118, 73, 74, 75, 128, 129, 105] publiziert.

Die Schutzmechanismen sind in Tabelle 8.1 mit einer kurzen Beschreibung aufgelistet.

Tabelle 8.1.: Zusammenfassung der untersuchten Schutzmechanismen

Schutzmechanismus	Erklärung
Trennung von Identität und Validierungsfähigkeiten	Um die Identität sowie die Fähigkeiten der einzelnen Validierer im DDVN sicherzustellen wird bei diesem Schutzmechanismus die Verwendung einer Privilege Management Infrastructure (PMI) vorgestellt. Hierbei übernimmt die Public Key Infrastructure (PKI) das Ausstellen eines Public Key Certificate (PKC)s und die PMI die Verwaltung der einzelnen Validiererfähigkeiten mittels ACs.
Erweiterung der Daten für die Validierung	Um die Validierungslogik zu verbessern, werden während des Validierungsprozesses zusätzliche Daten hinzugefügt. In der vorliegenden Arbeit handelt es sich bei den zusätzlichen Daten um Kontextinformationen.

Tabelle 8.1.: Zusammenfassung der untersuchten Schutzmechanismen

Schutzmechanismus	Erklärung
Schutz des Algorithmus	Eine Vielzahl von Algorithmen zur Validierung der Eingangsdaten mehrerer Validierer. Damit ein Angriff hier eine geringe Erfolgswahrscheinlichkeit hat, werden neue Ansätze wie *a*) das Zusammenfassen mehrerer Validiererergebnisse zu einem gemeinsamen Ergebnis mittels Quorumsentscheidung, *b*) das Aufteilen eines Validierers mit hohem Einfluss in mehrere Validierer mit geringerem Einfluss und *c*) das Hinzufügen von nicht verwendeten Validierern vorgestellt.
Einteilung von Validierern in Cluster	Um die Auswahl der verwendeten Validatoren durch den Akkumulator zu verbessern, werden die Validatoren in Cluster eingeteilt. Dadurch wird sichergestellt, dass ein erfolgreicher Zero-Day-Exploit oder ein ähnlicher Angriff weniger Auswirkungen auf das DDVN hat.
Aggregierungsentscheid mittels Cluster und Quorum Ansatz	Die Ergebnisse der verschiedenen Validierer werden mittels einer Aggregationsfunktion zusammengeführt. Um den Einfluss von korrumpierten Validierern zu minimieren, werden zwei unterschiedliche Aggregationsansätze 1) *t-Total Quorum* und 2) (s,t)-*Nested Quorum* eingeführt.

Die neuen Ansätze werden verwendet, um die in der STRIDE-Analyse identifizierten Bedrohungen zu mindern. In Tabelle 8.2 werden die Auswirkungen der neuen Schutzmechanismen auf die Sicherheitsziele der STRIDE-Analyse abgebildet und beschrieben. Drei der vier genannten Schutzmechanismen verbessern die Verfügbarkeit. Verfügbarkeit ist eines der zentralen Sicherheitsziele im industriellen Umfeld.

Tabelle 8.2.: Auswirkungen der neuen Schutzmechanismen auf die gefundenen Bedrohungen der STRIDE Analyse

Sicherheitsziel nach STRIDE	Erklärung
Trennung von Identität und Validierungsfähigkeiten	
Authentifizierung **Integrität** **Verbindlichkeit** **Vertraulichkeit** **Autorisierung**	Das Ausstellen von Zertifikaten stellt die Authentifizierung und Autorisierung sicher. Darüber hinaus werden die Zertifikate zusammen mit dem privaten und dem öffentlichen Schlüssel verwendet, um eine digitale Signatur der Validierungsergebnisse vorzunehmen und somit die Integrität zu gewährleisten. Durch die digitale Signatur der Validierungsergebnisse kann ein Validierer seine Ergebnisse nicht mehr abstreiten, wodurch die Verbindlichkeit umgesetzt wird. Öffentliche Schlüssel ermöglichen die verschlüsselte Übertragung von Daten, wodurch die Vertraulichkeit gewährleistet wird.
Erweiterung der Daten für die Validierung	
Verfügbarkeit	Durch die Erweiterung der Eingangsdaten einer Validierung wird die Verfügbarkeit erhöht, da die Validierungsergebnisse genauer und damit besser verwendbar werden. Unzureichende Daten wirken sich stark negativ auf die Anwendbarkeit des DDVN aus, da die Validierungsergebnisse nicht verwendet werden können.
Schutz des Algorithmus	
Verfügbarkeit **Vertraulichkeit**	Der Schutz des verwendeten Algorithmus durch verschiedene Mechanismen erhöht die Verfügbarkeit von brauchbaren Validierungsergebnissen. Der Schutz des Algorithmus und die Erweiterung der Validierungsdaten verhalten sich dabei gleich. Zusätzlich wird durch bestimmte Schutzmechanismen, wie z.B. das Einfügen von Validierern mit einem Auswirkungsgrad von 0, die Vertraulichkeit des DDVNs erhöht, da die angreifende Person im besten Fall keine Validierungsdaten erhält.

Tabelle 8.2.: Auswirkungen der neuen Schutzmechanismen auf die gefundenen Bedrohungen der STRIDE Analyse

Sicherheitsziel nach STRIDE	Erklärung
Einteilung von Validierern in Cluster	
Verfügbarkeit **Autorisierung**	Durch die Einteilung der Validierer in Cluster wird die Verfügbarkeit des DDVN erhöht, da bei bestimmten Angriffen nur ein Teil der Validierer korrumpiert wird. Eine gute Validiererauswahl stellt sicher, dass immer funktionierende Validierer am Validierungsprozess beteiligt sind. Sollten bestimmte Validierer aufgrund ihrer Eigenschaften als zu unsicher für einen bestimmten Validierungsprozess eingestuft werden, so können diese mittels der merkmalsbasierten Validiererauswahl ausgeschlossen werden.
Aggregierungsentscheid mittels Cluster Ansatz und Quorum Ansatz	
Integrität	Durch die Aggregation der Validierungsergebnisse mittels Cluster Ansatz und Quorum Ansatz wird eine höhere Integrität des Validierungsergebnisses erreicht. Ausserdem kann ein einzelner Validierer in den meisten Fällen nicht allein ein Validierungsergebnis liefern.

8.1. Trennung von Identität und Validierungsfähigkeiten

Systeme mit bestimmten Sicherheitsstufen erfordern ein Identitätsmanagement und ein Berechtigungsschema für die Nutzung bestimmter Funktionen oder für den Nachweis von Fähigkeiten. Im DDVN ist dies z.B. die Identität eines einzelnen Validierers und seine Fähigkeit, bestimmte Daten zu validieren. Unter Kapitel 6 wird gezeigt, welche Daten in einer Attributdatei und in einer Policy-Datei enthalten sind. Auf die Sicherung der Identität oder der Eigenschaften durch Zertifikate oder ähnliches wird hier nicht näher eingegangen. Im Folgenden wird dieser Punkt aufgegriffen und eine Lösung mittels eines PMIs vorgestellt [151, 152].

Für den Identitätsnachweis kann der *X.509-Standard* [153], auch PKI genannt, verwendet werden (siehe grauer Bereich Abbildung 8.1). Die PKI besteht aus drei

Teilen, einer Zertifizierungsstelle (CA), einer Registrierungsstelle (RA) und einer Validierungsstelle (VA). Zur Überprüfung der Gültigkeit eines Zertifikats kann eine Certificate Revocation List (CRL), ein Online Certificate Status Protocol (OCSP) oder NOVOMODO [154] verwendet werden. Die PKI stellt Identitätsnachweise mittels PKCs aus. Für den Identitätsnachweis sind folgende Angaben erforderlich: Version, Seriennummer, Signaturalgorithmus, Aussteller, Gültigkeit, Betreff, öffentlicher Schlüssel und Aussteller. Ein gültiges PKC ist im Anhang in Abbildung A.3 dargestellt.

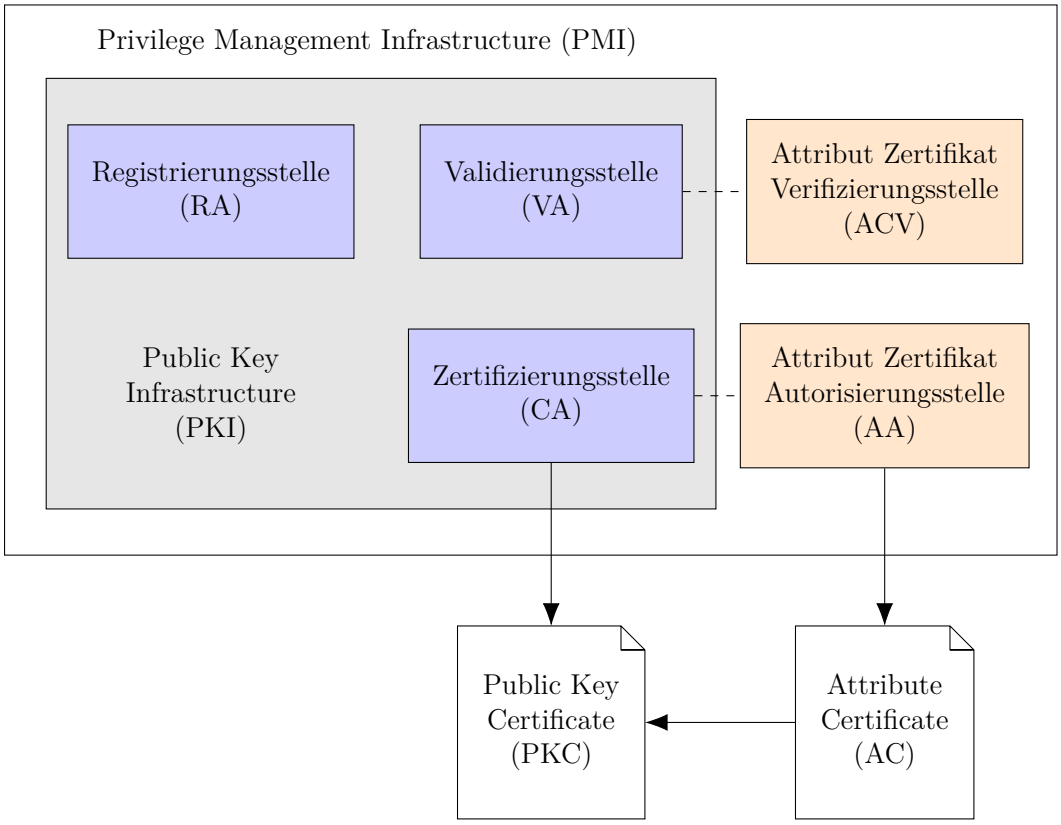


Abbildung 8.1.: Allgemeiner Aufbau einer Privilege Management Infrastructure

Die PKI kann neben der Identität auch Berechtigungen verwalten. Dabei wird das PKC neben dem Identitätsnachweis (in Zusammenarbeit mit einem privaten Schlüssel) um die Berechtigungen erweitert: *"The X.509 v3 certificate format also allows communities to define private extensions to carry information unique to those communities."* [153] Damit sind Identität und Berechtigungen stark gekoppelt, wodurch sich Probleme ergeben, wenn bestimmte Berechtigungen entzogen werden sollen. In diesem Fall muss der gesamte Identitätsnachweis widerrufen und neu ausgestellt werden.

Um die Verantwortung für die Autorisierung von der PKI zu entkoppeln, wird eine PMI verwendet [155]. Die PKI verwaltet nur die Identität für ein zu identifizierendes Objekt mittels PKC. Die PMI (siehe Abbildung 8.1) ist dafür

zuständig, die Berechtigungen für die identifizierten Objekte sicherzustellen. Dazu werden Zugriffsrechte mit ACs [156] implementiert und mit der PKC der zu autorisierenden Identität verknüpft. Ein AC wird durch das Attribut Autorisierungsstelle (AA) signiert. Dadurch können einzelne ACs zurückgezogen werden, ohne die PKC zu beeinflussen.

Neben den Zugriffsrechten und Fähigkeiten können ACs auch alternative Zugriffskontrollansätze wie Role Based Access Control (RBAC) oder Attribute Based Access Control (ABAC) implementieren. Zusammengefasst stellt die PKI die PKCs aus. Das mittels PKC identifizierte Objekt kann anschliessend einen Fähigkeitennachweis (AC) bei einer PMI beantragen. Die PMI prüft den Antrag und stellt, sofern zulässig, die beantragten ACs aus.

Unter Verwendung eines DDVN kann jeder Validierer ein PKC als Identitätsnachweis und mehrere ACs für die Validierungsfähigkeiten beantragen. Wenn der Akkumulator die Fähigkeiten anfordert, kann der Validierer mit den ausgestellten Zertifikaten antworten. Dadurch wird einerseits sichergestellt, dass es böswilligen Validierern erschwert wird, in das DDVN einzudringen, und andererseits, dass Validierer Fähigkeiten vortäuschen können, die sie nicht besitzen.

Bei der Verwendung von Zertifikaten muss die Datenmenge berücksichtigt werden. In diesem Zusammenhang kann einerseits ein leichtgewichtiges Zertifikat wie das Card Verifiable Certificate (CVC) [157, 158] in Betracht gezogen werden und andererseits die Verwendung eines alternativen Berechtigungskonzepts.

8.2. Erweiterung der Daten für die Validierung

Damit der Validierer ein korrektes Validierungsergebnis liefern kann, müssen die erforderlichen Daten zur Verfügung stehen. Bei einer regelbasierten Validierung kann ein einziger Eingabewert ausreichend sein. Ein Beispiel hierfür ist ein Temperatursensor, der in bestimmten Bereichen arbeitet; wenn diese Bereiche verlassen werden, sind die Daten nicht mehr vertrauenswürdig. Wenn der vorgegebene Bereich des Sensors (aus dem Datenblatt) als Regel hinterlegt ist und beim Empfang eines Temperaturwertes dieser überprüft wird, ist die Validierung vollständig.

In Szenarien, in denen bestimmte Eingangsdaten korreliert werden müssen, z. B. in einem Machine Learning (ML) Ansatz, muss sichergestellt werden, dass die geeigneten Daten in den Algorithmus eingegeben werden. Die Auswahl der passenden Daten wird als Merkmalsauswahl oder Feature Selection bezeichnet. Einige wichtige Punkte für die Feature Selection wurden von Guyon et al. [159] untersucht und werden im Folgenden kurz erläutert:

- *Hinzufügen von scheinbar redundanten Merkmalen:* Eine Verringerung des Rauschens und damit eine bessere Klassentrennung kann durch Hinzufügen

von Merkmalen erreicht werden, die scheinbar redundant sind. Merkmale, die unabhängig und identisch verteilt sind, sind nicht wirklich redundant. Zusammenfassend verbessert das Hinzufügen von nicht wirklich redundanten Merkmalen das Validierungsergebnis.

- *Entfernen von vollständig korrelierenden Merkmalen:* Vollständig korrelierte Merkmale sind wirklich redundant, so dass ihre Addition keine zusätzliche Information liefert und sie bei der Merkmalsauswahl eliminiert werden können. Umgekehrt bedeutet eine sehr hohe Korrelation (oder Anti-Korrelation) zwischen Merkmalen nicht, dass keine Komplementarität zwischen den Variablen besteht, und sollte daher nicht vernachlässigt werden.
- *Bestimmte Merkmale generieren nur Mehrwert mit anderen Merkmalen:* Merkmale, die für sich allein betrachtet werden, können nutzlos sein und nur in Verbindung mit anderen Merkmalen einen signifikanten Vorteil für die Vorhersage bringen.

In dieser wissenschaftlichen Arbeit wird vorgeschlagen, dass es sich bei den zusätzlichen Daten um Kontextinformationen handelt. Gerade im Industrie 4.0 Umfeld gibt es viele Sensoren, die Daten produzieren und somit viele Kontextinformationen gesammelt werden können.

Grundsätzlich gibt es drei verschiedene Ansätze für die Integration der Merkmalsauswahl in das System [160, 161, 162]:

- *Filter:* die Auswahl der verwendeten Merkmale erfolgt vor der eigentlichen Ausführung des Algorithmus und basiert auf den intrinsischen Eigenschaften der Merkmale. Die Merkmale werden hierbei entweder einzeln (univariat) oder im Zusammenhang mit anderen Merkmalen (multivariat) betrachtet.
- *Wrapper:* für die Auswahl der Merkmale wird der ML Algorithmus als Black Box angenommen und die Merkmalsauswahl umhüllt den Algorithmus.
- *Embedded:* die Auswahl der Merkmale erfolgt innerhalb des ML Algorithmus und braucht daher weniger Rechenleistung als der *Wrapper* Ansatz.

Beim DDVN erfolgt die Merkmalsselektion auf der Ebene der Validierer, d.h. unabhängig von der eigentlichen Implementierungsart (Filter, Wrapper, Embedded). Der Filter-Ansatz könnte durch zusätzliche Validierer realisiert werden, die eine Filterung vornehmen und erst dann die gefilterten Daten weitergeben. Bei Wrapper und Embedded ist die Merkmalsauswahl näher an der eigentlichen Validierung integriert. Für die Anreicherung der eigentlichen Daten mit Kontextinformationen gibt es drei verschiedene Möglichkeiten: 1) ein bestehendes Gateway wird wiederverwendet und sendet neben den Sensordaten auch Kontextdaten (siehe Abbildung 8.2), 2) ein alternatives (Kontext-)Gateway wird verwendet (siehe Abbildung 8.3) und 3) die Kontextinformationen werden direkt an den Validator übergeben (siehe Abbildung 8.4).

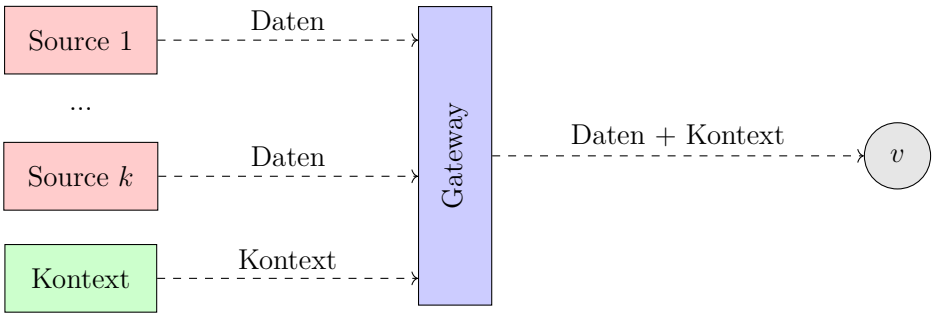


Abbildung 8.2.: Hinzufügen der Kontextinformation mit einem bereits verwendeten Gateway

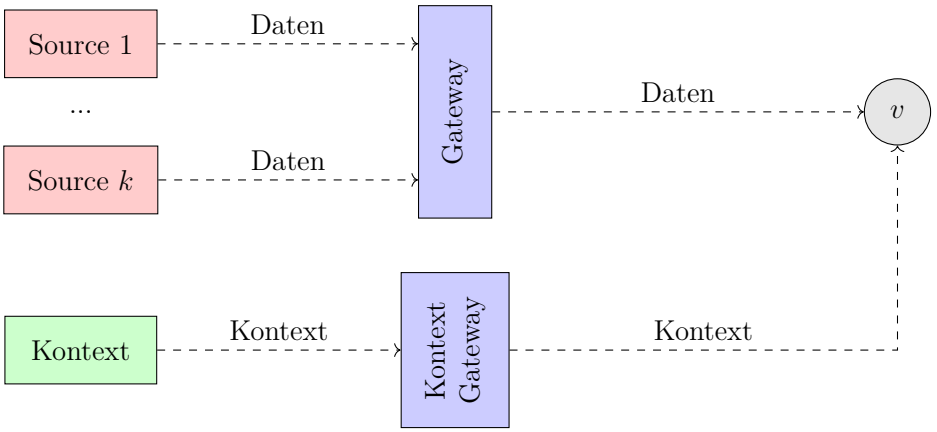


Abbildung 8.3.: Hinzufügen der Kontextinformation mit einem weiteren (Kontext-) Gateway

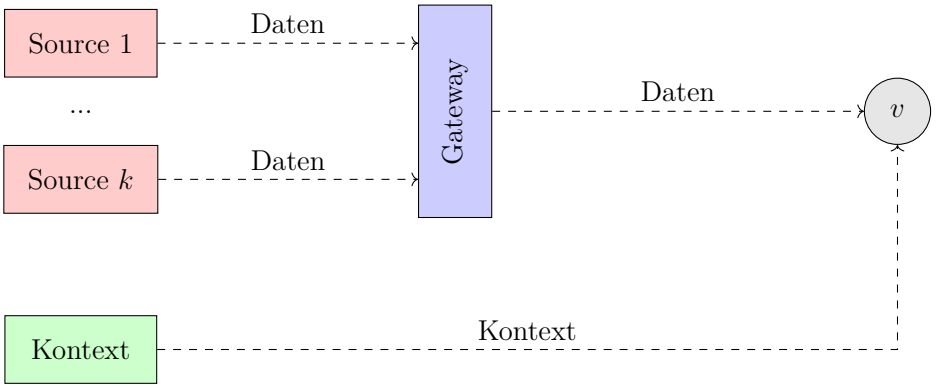


Abbildung 8.4.: Hinzufügen der Kontextinformation direkt beim Validierer

Die Qualität der Validierungsergebnisse kann anhand verschiedener Metriken gemessen werden. Dazu gehören Genauigkeit (ACC), Sensitivität (TPR), Spezifität (TNR), Präzision (PPV) sowie die Anzahl der False-Positive (FP), False-Negative (FN), True-Positive (TP) und True-Negative (TN) [163]. Dabei

entspricht FP der Anzahl der Validierungsergebnisse, die fälschlicherweise als gut bewertet wurden, und FN der Anzahl der Validierungsergebnisse, die als falsch oder schlecht bewertet wurden, obwohl sie gut waren. TP und TN entsprechen den korrekt erkannten guten und schlechten Ergebnissen. Ausgehend von diesen Zahlen können die verbleibenden Werte wie folgt bestimmt werden

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{8.1}$$

$$TPR = \frac{TP}{TP + FN} \tag{8.2}$$

$$TNR = \frac{TN}{TN + FP} \tag{8.3}$$

$$PPV = \frac{TP}{TP + FP} \tag{8.4}$$

Für die Untersuchung des Einflusses von Zusatzinformationen auf die Vorhersageergebnisse wurde ein anonymisierter Datensatz der Firma TRUMPF GmbH & Co. KG zur Verfügung gestellt. Bei den Daten handelt es sich um Maschinendaten, die für die Vorhersage verwendet werden. Ein Ausschnitt ist in Tabelle 8.3 dargestellt. Der Datensatz besteht aus 24 Merkmalen und 243 Einträgen.

Tabelle 8.3.: Ausschnitt aus dem Datensatz der Firma TRUMPF GmbH & Co. KG

Datum	F_1	F_2	F_i	F_{24}	**Anomalie**
2017-01-04	-0.39076	-0.83337	...	-0.05751	Nein
2017-01-05	-0.40015	-0.79404	...	1.84592	Nein
2017-01-06	-0.42676	-0.79761	...	-0.05751	Nein
...	...	...	...	...	...
2017-12-23	0.01410	0.15785	...	-0.80233	Ja
2017-12-27	15.25476	13.75834	...	1.68040	Ja

Im ersten Schritt wird die Genauigkeit von fünf verschiedenen Algorithmen *a*) Decision Tree (DT), *b*) k-nearest Neighbors (KNN), *c*) Support Vector

Classification (SVC), *d*) Stacking und *e*) Voting bestimmt. Es werden keine Merkmale entfernt, d.h. stark korrelierende Merkmale sind noch enthalten und es erfolgt keine Trennung in normale Daten und Kontextinformationen. Für die Analyse wird der Datensatz im Verhältnis $\frac{1}{4}$ zu $\frac{3}{4}$ aufgeteilt. Das Training erfolgt anschliessend mit $\frac{1}{4}$ der Daten und das Testen mit den verbleibenden $\frac{3}{4}$. Die Ergebnisse werden über fünf Runden gemittelt, um den Einfluss von Ausreissern zu minimieren. In Tabelle 8.4 sind die einzelnen Algorithmen mit ihren Ergebnissen aufgeführt. Für die verschiedenen Optimierungen: Genauigkeit, Anzahl von TN/FP und Anzahl von TP/FN wurden jeweils die Hyperparameter der Algorithmen variiert und die beste Konfiguration ausgewählt. Zudem wurden die Ergebnisse für unterschiedliche Seeds (7, 42, 77, 128, 256) bestimmt und anschliessend gemittelt. Die untersuchten Bereiche der Hyperparameter sind wie folgt definiert:

- *DT*: die maximale Baumtiefe wurde zwischen $[1, 25]$ variiert
- *KNN*: die Anzahl der Nachbarn wurde zwischen $[1, 15]$ variiert
- *SVC*: der verwendete Kernel wurde zwischen $\{poly, rbf, sigmoid\}$ variiert

Tabelle 8.4.: Vorhersageergebnisse bei Verwendung des rohen Datensatzes

Algorithmus	Genauigkeit	Anzahl TN	Anzahl FP	Anzahl FN	Anzahl TP
Optimierung auf Genauigkeit					
DT	0.944	34.8	1.6	1.8	22.8
KNN	0.879	31.8	4.6	2.8	21.8
SVC	0.866	32.4	4.0	4.2	20.4
Stacking	0.928	33.4	3.0	1.4	23.2
Voting	0.908	33.0	3.4	2.2	22.4
Optimierung auf die Anzahl von TN/FP					
DT	0.928	35.0	1.4	3.0	21.6
KNN	0.823	33.8	2.6	8.2	16.4
SVC	0.800	33.4	3.0	9.2	15.4
Stacking	0.921	33.2	3.2	1.6	23.0
Voting	0.866	34.2	2.2	6.0	18.6

Tabelle 8.4.: Vorhersageergebnisse bei Verwendung des rohen Datensatzes

Algorithmus	Genauigkeit	Anzahl TN	Anzahl FP	Anzahl FN	Anzahl TP
Optimierung auf die Anzahl von TP/FN					
DT	0.918	34.0	2.4	1.8	22.8
KNN	0.852	30.2	6.2	2.8	21.8
SVC	0.862	32.2	4.2	4.2	20.4
Stacking	0.915	32.4	4.0	1.2	23.4
Voting	0.895	32.4	4.0	2.4	22.2

Für die Auswahl der zentralen Merkmale wird der Pearson Standardkorrelationskoeffizient [164] berechnet und alle Merkmale, die einen Koeffizienten von > 0.8 aufweisen, werden eliminiert. Dadurch reduziert sich die Anzahl der verwendeten Merkmale von 24 auf 16, was einer Reduktion um 33% entspricht. In Tabelle 8.5 sind die Ergebnisse nach der Merkmalsauswahl dargestellt, wobei die Verbesserungen (33 Einträge) grün und die Verschlechterungen (27 Einträge) rot eingefärbt sind. Die Sensitivität hat sich durch die Erweiterung der Merkmalsauswahl um 2.73% verschlechtert. Selektivität und Präzision wurden um 0.99% bzw. 0.87% verbessert. Daraus ist ersichtlich, dass bei einer Merkmalsauswahl genau kontrolliert werden muss, welche Metriken am wichtigsten eingestuft werden. Dies geschieht in der Regel durch Expertenwissen oder eine fachliche Vorgabe.

Tabelle 8.5.: Vorhersageergebnisse bei Verwendung des Datensatzes nach der Feature Selection

Algorithmus	Genauigkeit	Anzahl TN	Anzahl FP	Anzahl FN	Anzahl TP
Optimierung auf Genauigkeit					
DT	0.947	34.8	1.6	1.6	23.0
KNN	0.856	33.0	3.4	5.4	19.2
SVC	0.869	33.0	3.4	4.6	20.0
Stacking	0.905	33.4	3.0	2.8	21.8
Voting	0.898	34.2	2.2	4.0	20.6

Tabelle 8.5.: Vorhersageergebnisse bei Verwendung des Datensatzes nach der Feature Selection

Algorithmus	Genauigkeit	Anzahl TN	Anzahl FP	Anzahl FN	Anzahl TP
Optimierung auf die Anzahl von TN/FP					
DT	0.944	35.0	1.4	2.0	22.6
KNN	0.800	34.2	2.2	10.0	14.6
SVC	0.757	34.0	2.4	12.4	12.2
Stacking	0.941	34.4	2.0	1.6	23.0
Voting	0.866	34.8	1.6	6.6	18.0
Optimierung auf die Anzahl von TP/FN					
DT	0.938	34.0	2.4	1.4	23.2
KNN	0.823	29.8	6.6	4.2	20.4
SVC	0.849	31.2	5.2	4.0	20.6
Stacking	0.928	33.2	3.2	1.2	23.4
Voting	0.908	33.2	3.2	2.4	22.2

Unter der Annahme, dass die Kontextinformationen zusammen mit den Daten als Input in den Vorhersagealgorithmus einfliessen, ist in Abbildung 8.5 zu sehen, wie die Genauigkeit der Vorhersagen von anfänglich unter 71% unabhängig vom jeweiligen Algorithmus mit steigender Anzahl der verwendeten Features auf über 86% ansteigt. Die bereits beschriebenen Anforderungen von Guyon et al. [159] gelten auch hier.

Für die Analyse der beiden Punkte *a*) Auswirkung einer guten Merkmalsauswahl und *b*) Auswirkung der Verwendung mehrerer unabhängiger Merkmale wurden die Python-Bibliotheken scikit-learn [165], pandas [166], numpy [167], matplotlib [168] und seaborn [169] verwendet.

Die Datenqualität setzt sich aus einer Vielzahl unterschiedlicher Metriken zusammen, die in vier Kategorien unterteilt sind: 1. intrinsische Daten, 2. kontextbezogene Daten, 3. repräsentative Daten und 4. zugängliche Daten. Eine detaillierte Beschreibung der Kategorien und Metriken findet sich in Abschnitt 4.3.

Die Betrachtung der Datenqualität in Bezug auf Sicherheit kann auf drei Arten erfolgen:

1. *Metrik Ebene:* Eine einzelne Metrik wird herangezogen und basierend auf den jeweiligen Anwendungsfall bewertet.

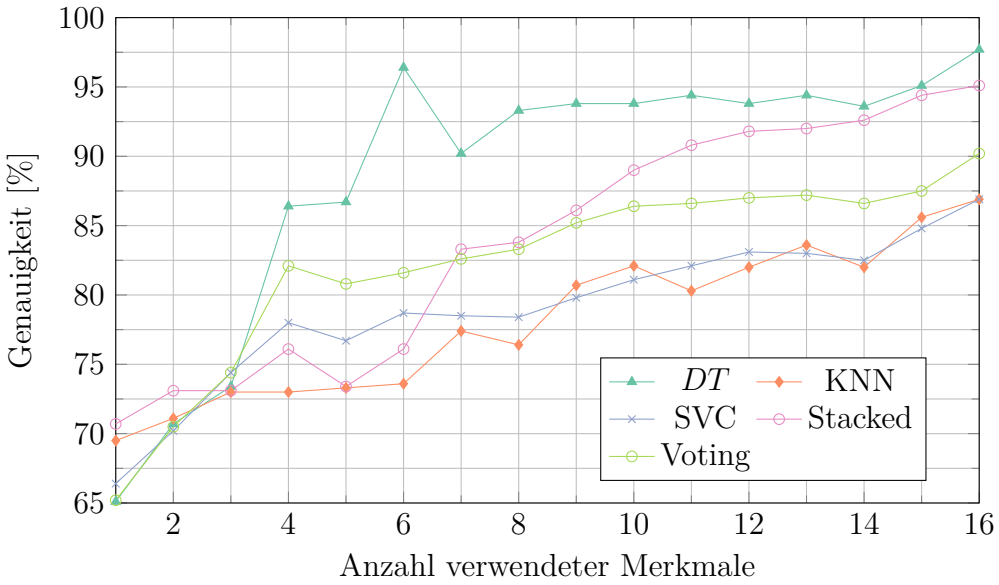


Abbildung 8.5.: Verbesserung des Vorhersageergebnisses durch zusätzliche Merkmale

2. *Kategorie Ebene:* Die Metriken einer einzelnen Kategorie werden herangezogen und basierend auf den jeweiligen Anwendungsfall zusammengefasst bewertet.
3. *Kategorie Kombination:* Die Kategorien werden in Relation zu einander gesetzt und und die Kombination wird bewertet (siehe Abbildung 8.6). Bei einer sehr oberflächlichen Betrachtung kann davon ausgegangen werden, dass desto mehr Kategorien zusammengeführt werden, desto höher ist die Sicherheitsanforderung an ein gegebenes System.

Im Folgenden wird ein Beispiel für die vier Sicherheitsstufen der Datenqualitätskategorien beschrieben. Es werden Daten gegeben, die in Bezug auf Genauigkeit (I), Relevanz (K), Kompaktheit (R) und Verfügbarkeit (Z) betrachtet werden. Unter der Annahme, dass alle Werte in einen Bereich überführt wurden, der miteinander in Beziehung gesetzt werden kann, ergibt sich das Rechenbeispiel aus Tabelle 8.6. Der Wertebereich ist $[0 - 100]$ (100 entspricht der höchsten Qualität) und als Funktion für die Aggregation der Kategorien wird das arithmetische Mittel $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ verwendet. Alternativ kann ein gewichteter Mittelwert verwendet werden, bei dem bestimmte Kategorien stärker gewichtet werden als andere.

Die Sicherung der Qualität mehrerer Metriken ist schwieriger als die Sicherung der Qualität einer einzelnen Metrik. Daher ist es auch schwieriger, eine hohe aggregierte Qualität $\bar{x}$ zu erreichen.

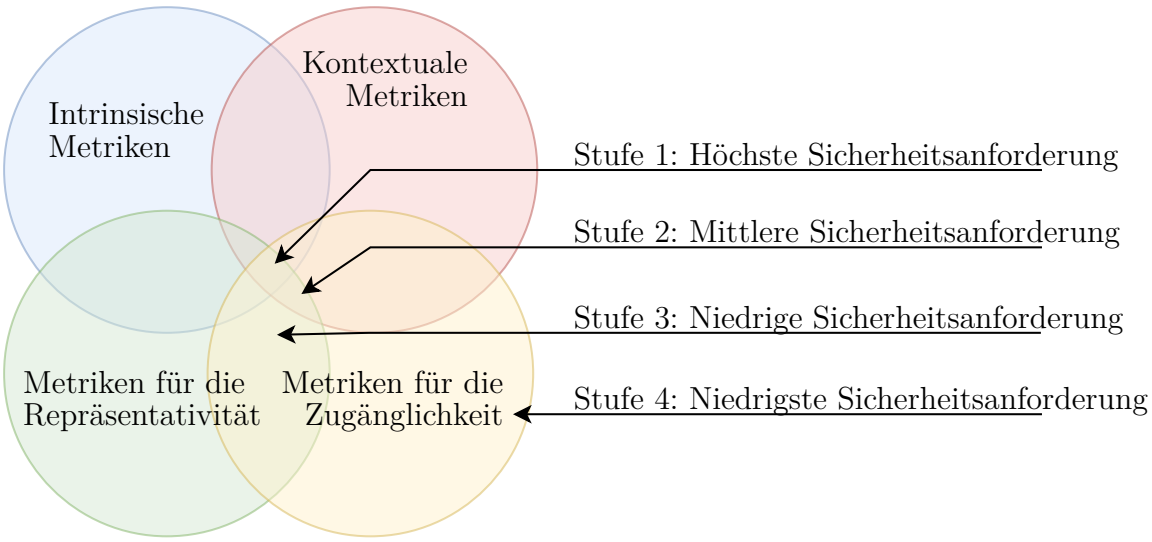


Abbildung 8.6.: Die vier Sicherheitsstufen der Datenqualitätskategorien

Sollte ein System eine hohe Sicherheit erfordern, ist es wichtig, mehr als eine Kategorie zur Bestimmung der Qualität zu verwenden. Es ist auch möglich, verschiedene Metriken zu aggregieren. Ein einfacher Ansatz ist die Verwendung der Stufe 1 (höchste Sicherheitsanforderung), bei der mindestens eine Metrik aus jeder Kategorie verwendet wird.

Tabelle 8.6.: Bestimmung der aggregierten Datenqualität

Nr.	Genauigkeit	Relevanz	Kompaktheit	Verfügbarkeit	$\bar{x}$
1.	20	20	20	20	20
2.	20	20	20	80	35
3.	20	20	80	80	50
4.	80	80	80	80	80

8.3. Definition der untersuchten Algorithmen

Je nach Struktur des verwendeten Algorithmus lassen sich verschiedene Rückschlüsse auf das System und mögliche Angriffe ziehen. Ausserdem können mögliche gefährdete Bereiche (siehe Abschnitt 7.5) abgeleitet werden. Die Validierer und ihre Verbindungen untereinander bzw. zu Gateway und Akkumulator werden als gerichteter azyklischer Graph (azyklischer Digraph) ohne Mehrfachkanten definiert. Damit kann die Struktur eines verteilten Algorithmus in eine bereits bekannte Darstellung überführt werden (siehe Abbildung 8.7). Das Gateway entspricht immer dem ersten Knoten v_1 und der Akkumulator immer

dem letzten Knoten v_n. Die Knoten dazwischen $\{v_2, v_3, v_4, ...\}$ entsprechen der Struktur des verteilten Algorithmus.

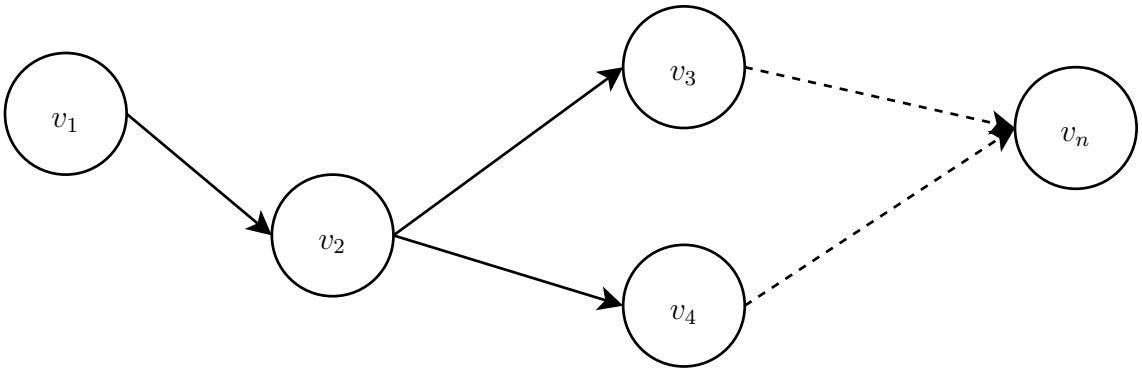


Abbildung 8.7.: Graphenstruktur der Verteilten Algorithmen

Durch die Betrachtungsweise des DDVN als Graph, werden folgende Eigenschaften abgeleitet:

- *Gerichtet:* Jeder Knoten, der mit einem anderen Knoten in Verbindung steht, wird über einen Pfeil gekennzeichnet und kann nur in eine Richtung durchlaufen werden.
- *Azyklisch:* Es gibt keinen Pfad im Aufbau der einem Zyklus entspricht, d.h. es gibt nur eine endliche Menge an Pfaden durch den Graph.
- *Ohne Mehrfachkanten:* Zwischen zwei Knoten gibt es maximal eine gerichtete Kante, d.h. es gibt eine Verbindung zwischen v_2 zu v_3 aber nicht in die entgegengesetzte Richtung. Zudem gibt es auch nicht mehrere gleiche Kanten in eine Richtung.

Mögliche Validiererstrukturen umfassen die Verwendung eines einzelnen Validierers (siehe Abbildung 8.8). Es kann ein regelbasierter Algorithmus verwendet werden, der Validierungsregeln basierend auf einem Datenblatt oder Expertenwissen erstellt. Das Gateway sendet die Daten der Source direkt an den v_1, der die Daten validiert und das Ergebnis der Validierung an den Akkumulator zurückmeldet.

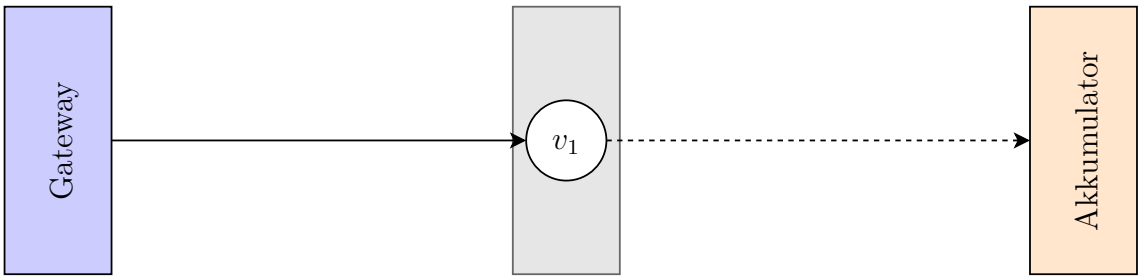


Abbildung 8.8.: Graph mit einem einzelnen Validierer

Neben der Betrachtung eines Graphen mit einem Knoten v_1 werden auch Graphen betrachtet, bei denen ein Knoten einen Ausgangsgrad von maximal zwei besitzt und die Anzahl der verwendeten Validierer gleich n ist. Ausserdem laufen die Folgeknoten unmittelbar auf den nächsten Knoten zu. Dieser Ansatz ist inspiriert von einem einfachen Branched DDNN [170, 171, 172, 173] (siehe Abbildung 8.9). Die Ergebnisse der vorhergehenden Validierer werden an alle nachfolgenden Validierer, auch Branches genannt, weitergegeben. So gibt v_1 sein Ergebnis an v_2' und v_2'' weiter. Die Ergebnisse von $\{v_2', v_2''\}$ können anschliessend in einem Validierer v_3 zusammengeführt werden.

Die Validierer im DDVN sind in Layer eingeteilt. Ein Layer besteht aus ein bis maximal zwei Knoten, die sich auf der gleichen Ebene befinden, d.h. die Knoten können sich nie gegenseitig erreichen und sind beide gleich weit (gleich viele Kanten) vom Wurzelknoten entfernt. In der Abbildung sind die Layer grau dargestellt. Die Knoten v_2' und v_2'' sind weiss umrandet, um zu zeigen, dass beide gleichzeitig verwendet werden.

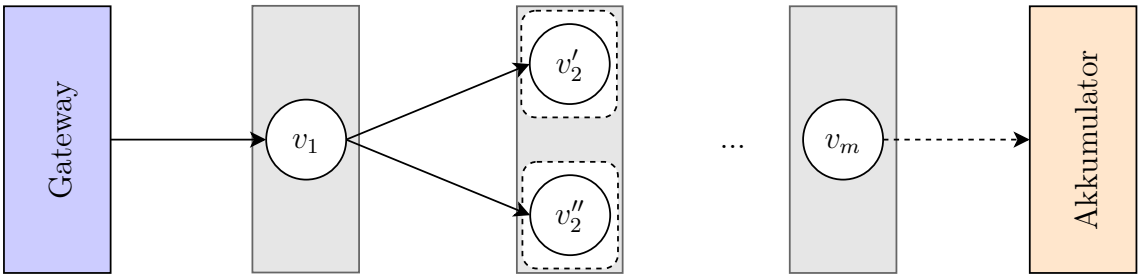


Abbildung 8.9.: Graph mit m Layern und n Knoten, die einen maximalen Ausgangsgrad von zwei besitzen

Die dritte Validiererstruktur entspricht einer Baumstruktur, in einem verteilten System kann dies ein DDT [174, 175, 176] sein. Dabei werden mehrere Validierer zu einem verteilten Baum zusammengefasst (siehe Abbildung 8.10). Der Validierer v_1 erhält die Eingabedaten und validiert diese. Anhand seines Ergebnisses entscheidet v_1, an welchen nachfolgenden Validierer v_2' bzw. v_2'' er die Validierung weitergibt. Es wird immer nur ein nachfolgender Validierer ausgewählt. Die graue Markierung zeigt die jeweilige Baumebene an, in DDVN wird die Ebene auch als Layer bezeichnet.

Tan et al. [177] beschreiben die Erstellung eines Entscheidungsbaums und definieren hierfür vier zentrale Methoden:

- *create_node:* Die Erzeugung eines Knotens
- *find_best_split:* Das Feststellen des besten Aufteilungskriteriums
- *classify:* Definiert das Ergebnis für einen Blattknoten
- *stopping_cond:* Sobald eine Abschlussbedingung erfüllt ist, wird der aktuelle Knoten zu einem Blattknoten

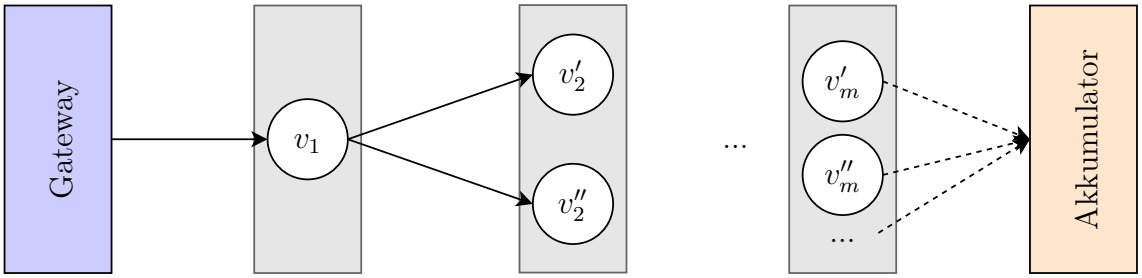


Abbildung 8.10.: Verteilte Baumstruktur

Ein Pseudocode, der den Ablauf der Entscheidungsbaumgenerierung zeigt, ist in Listing 8.1 angegeben. Es handelt sich um einen rekursiven Algorithmus, der zunächst das Abbruchkriterium mittels *stopping_cond* prüft. Ist das Kriterium erfüllt, wird der aktuelle Knoten als Blattknoten klassifiziert und eine *classify*-Klassifizierung des Knotens durchgeführt. Ist das Abbruchkriterium nicht erfüllt, wird ein normaler Entscheidungsknoten angenommen und die beste Teilungsmöglichkeit *find_best_split* ermittelt.

```
def tree_growth(training_data,attributes):
    if stopping_cond(training_data,attributes):
        # Erzeugen eines Blattknotens
        # mit Kategorie
        leaf = create_node()
        leaf.label = classify(training_data)
        return leaf
    else:
        # Erzeugung eines Entscheidungknotens
        # mit dem Entscheidungskriterium
        root = create_node()
        root.test_cond =
        find_best_split(training_data,attributes)

        # Iterieren aller Kategorien
        # und Aufteilen der Trainingsdaten
        for label in possible_labels():

            # Kombinieren von Kategorie und
            # Trainingsdaten
            training_data_label = []
            for data in training_data:
                if root.test_cond(data) == label:
                    training_data_label.push(data)

            # Rekursiver Aufrufe mit den
```

```
            # Trainingsdaten und Kategoriesubset
            child =
            tree_growth(training_data_label,attributes)
            # Unterknoten mit ermittelten
            # Kategorie speichern
            root.childs[label] = child

        return root
```

Listing 8.1: Rekursiver Pseudocode zur Erstellung eines Entscheidungsbaums

Für die Feststellung der Aufteilung gibt es eine Vielzahl an Algorithmen, hierzu gehören die Betrachtung der Entropy oder der Gini Index [178, 179, 180]. Ein Auflistung von unterschiedlichen Aufteilungsalgorithmen ist in Tabelle 8.7 gegeben. Die Auswahl des Aufteilunsalgorithmus ist Anwendungsfall spezifisch weswegen in dieser Arbeit keine Aussage bezüglich der Eignung eines spezifischen Algorithmus erfolgt.

Tabelle 8.7.: Unterschiedliche Aufteilungsalgorithmen für die Baumstruktur

Eigenschaft	ID3	C4.5	CART
Unterstützte Attribut Typen	Kategorien	Kategorien und Werte	Kategorien und Werte
Aufteilungs-kriterien	Entropie und Informationsgewinn	Informationsgewinn und Gewinnverhältnis	Entropie, Gini oder Towing und Informationsgewinn
Behandeln von fehlenden Werten	Nein	Ja	Ja
Unterstützt Entscheidungsbaum Pruning	Nein	Ja	Ja
Typ des Ergebnisses	Numerisch	Numerisch	Binär

8.4. Definition von Korruptionsgrad κ und Auswirkungsgrad η

Im Folgenden werden die wesentlichen Metriken für das DDVN a) der Korruptionsgrad κ, b) der Auswirkungsgrad η, c) die Angriffserfolgswahrscheinlichkeit γ für eine Baumstruktur und einen azyklischer gerichteter Graph ohne Mehrfachkanten sowie d) die Gleichung zur Berechnung der Selektionswahrscheinlichkeit von korrupter Knoten definiert. Der *Korruptionsgrad* definiert, wie viele Knoten in einem System korrumpiert sind. Alle zu berücksichtigenden Knoten im System sind $n = |N|$ ($n \in \mathbb{N}$) und die Anzahl der korrumpierten Knoten von N wird durch $c = |C|$ ($C \subseteq N$) dargestellt. Daraus folgt für den Korruptionsgrad κ und die Integrität ι:

$$\kappa = \frac{c}{n} \tag{8.5}$$

$$\iota = 1 - \kappa \tag{8.6}$$

In bisherigen Publikationen wurde κ meist mit dem Auswirkungsgrad gleichgesetzt. Um Verwirrung vorzubeugen, wird in dieser wissenschaftlichen Arbeit der Auswirkungsgrad η separat angegeben. Der *Auswirkungsgrad* beschreibt, wie gross der Einfluss von c auf den verwendeten Algorithmus ist. Abhängig von der jeiligen Algorithmusstruktur kann die Auswirkung geringer oder grösser sein. Gegeben ist M, ein Multiset, das alle durch den jeweils verwendeten Algorithmus klassifizierten Eingangsdaten x enthält.

$$m = |M| = \sum_{x \in M} x \tag{8.7}$$

Jeder Layer l hat einen maximalen Auswirkungsgrad von 1, d.h. wenn der Layer vollständig korrumpiert ist, ist $\eta = 1$ und unabhängig von der Anzahl m kann jedes Ergebnis manipuliert werden. Die Aufteilung des Auswirkungsgrades auf die einzelnen Knoten eines Layers ist sehr spezifisch und muss über Bedingungen, Expertenwissen oder ähnliches hergeleitet werden. So kann ein Knoten 90% der Auswirkungen haben und ein anderer 10%. Im Allgemeinen kann der Auswirkungsgrad eines Layers l als die Summe aller Teilauswirkungen der korrupten Knoten $C_l = \{c_{1l}, c_{2l}, ...c_{il}\}$ eines Layers betrachtet werden:

$$\eta(C_l) = \sum_{j \leq i} \eta(c_{jl}) \tag{8.8}$$

Schwieriger ist die Betrachtung des Auswirkungsgrades über alle Layer Λ. Hier kann es je nach gewähltem Algorithmus zu einer Korrelation zwischen den Teilauswirkungen kommen. Beispielsweise haben in einem Baumstruktur (siehe Abbildung 8.11) die Folgeknoten eines korrumpierten Elternknotens keine Teilauswirkung mehr, da der Elternknoten sicherstellt, dass der Angriff bereits erfolgreich war. Wenn v_2' korrumpiert ist, müssen v_3' und v_3'' nicht mehr berücksichtigt werden, da der Pfad bereits korrumpiert ist. Gleiches gilt für die Korrumpierung des Wurzelknotens v_1. Wenn dieser korrumpiert ist, dann ist jeder Pfad korrumpiert und das System wurde von der angreifenden Instanz übernommen. Aus diesem Grund hat der Wurzelknoten auch die maximale Auswirkung eines einzelnen Knotens von 1.

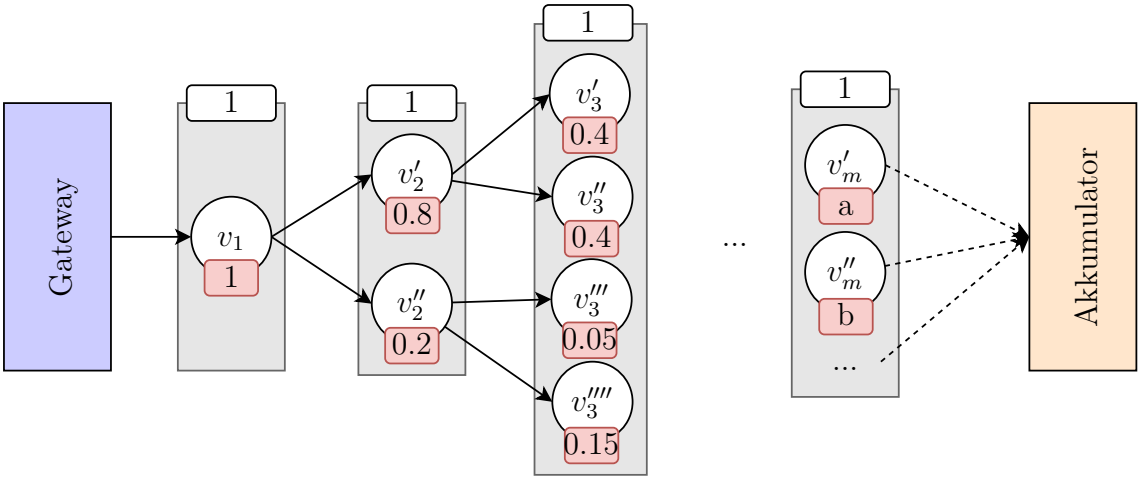


Abbildung 8.11.: Baumstruktur mit einem exemplarischen Auswirkungsgrad η pro Knoten

Gegeben ist eine Menge an korrumpierten Elternknoten $C_p = \{c_{p1}, c_{p2}, ...c_{pi}\}$ in einer Baumstruktur. Daraus können der Auswirkungsgrad $\eta(\Lambda)$ und der Schutzgrad $\rho(\Lambda)$ bestimmt werden:

$$\eta(\Lambda) = \eta(C_p) = \sum_{j \leq i} \eta(c_{pj}) \tag{8.9}$$

$$\rho(\Lambda) = 1 - \eta(\Lambda) \tag{8.10}$$

Eine Simulation des Schutzgrades für eine Baumstruktur ist in Abbildung 8.12 und Abbildung 8.13 dargestellt. Es werden jeweils vier unterschiedlich grosse Entscheidungsbäume mit $|N| = \{3, 7, 15, 31\}$ bzw. $|N| = \{63, 127, 255, 511\}$ Knoten untersucht. Dabei wird die Anzahl der korrumpierten Knoten erhöht und die korrumpierten Knoten jeweils auf unterschiedliche Positionen im

Entscheidungsbaum verteilt. Anschliessend wird jeweils $\rho(\Lambda)$ berechnet. Es ist ersichtlich, dass ohne zusätzliche Schutzmechanismen der Schutzgrad schnell unter 80% sinkt.

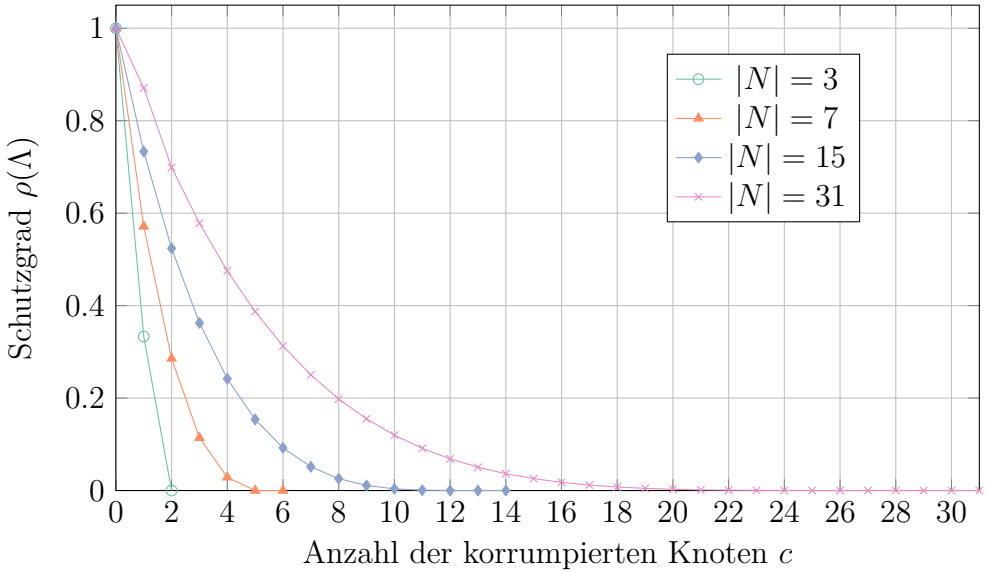


Abbildung 8.12.: Durchschnittlicher Schutzgrad $\rho(\Lambda)$ für eine Baumstruktur mit einer steigenden Anzahl an korrumpierten Knoten c

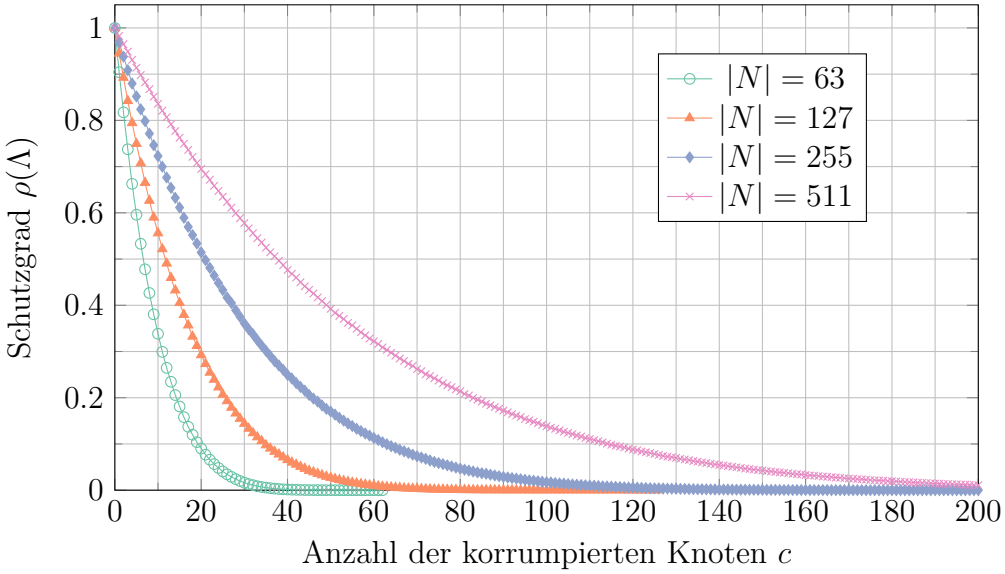


Abbildung 8.13.: Durchschnittlicher Schutzgrad $\rho(\Lambda)$ für eine Baumstruktur mit einer steigenden Anzahl an korrumpierten Knoten c

Die Auswirkung der Anzahl an korrumpierten Knoten auf eine Baumstruktur ist in Tabelle 8.8 und Tabelle 8.9 gegeben. Hierbei wird zwischen einer konstanten

Anzahl an korrumpierten Knoten 2 und einem prozentualen Anteil von 20% an korrumpierten Knoten unterschieden. In der ersten Abbildung ist ersichtlich, dass zwei korrumpierte Knoten bei einer Baumstruktur mit 511 Knoten eine mittlere Erfolgswahrscheinlichkeit von mehr als 3% besitzt.

Tabelle 8.8.: Korrelation zwischen einer konstanten Anzahl an korrumpierten Knoten und des Schutzgrads ρ der Baumstruktur

Anzahl Knoten	**Anzahl korrumpierter Knoten**	**Prozentualer Anteil korrumpierter Knoten**	**Schutzgrad** ρ
7	2	28.6%	28.57%
15	2	13.3%	52.38%
31	2	6.5%	69.89%
63 (Simulation)	2	3.2%	81.74%
127 (Simulation)	2	1.6%	89.29%
255 (Simulation)	2	0.8%	93.79%
511 (Simulation)	2	0.4%	96.48%

Tabelle 8.9.: Korrelation zwischen einer prozentualen Anzahl (ca. 20%) an korrumpierten Knoten und des Schutzgrads ρ der Baumstruktur

Anzahl Knoten	**Anzahl korrumpierter Knoten**	**Prozentualer Anteil korrumpierter Knoten**	**Schutzgrad** ρ
7	1	14.3%	57.14%
15	3	20%	36.26%
31	6	19.4%	31.27%
63 (Simulation)	13	20.6%	23.45%
127 (Simulation)	25	19.7%	20.70%
255 (Simulation)	51	20%	16.35%
511 (Simulation)	102	20%	13.27%

Ein wesentlicher Unterschied zwischen dem azyklischer gerichteter Graph ohne Mehrfachkanten und der Baumstruktur ist die Abhängigkeit zu den vorherigen

Knoten. Die Baumstruktur hat eine starke Abhängigkeit zu den Elternkoten, so werden nicht alle Knoten pro Entscheidungsfindung durchlaufen. Die Auswahl des nachfolgenden Knotens findet jeweils auf Basis der Entscheidung des Elternknoten statt. Bei einem azyklischer gerichteter Graph ohne Mehrfachkanten erhalten stets alle Knoten eines Layers die Ergebnisse des vorherigen Layers. Dadurch beeinflusst das Ergebnis der Elternknoten nicht den Pfad der durchlaufenen Knoten. In Abbildung 8.14 ist ersichtlich, dass bei einem Graph mit einem Ausgangsgrad von zwei (v_1 nach v_2' und v_2'') die Auswirkung auf das Ergebnis aufgeteilt wird.

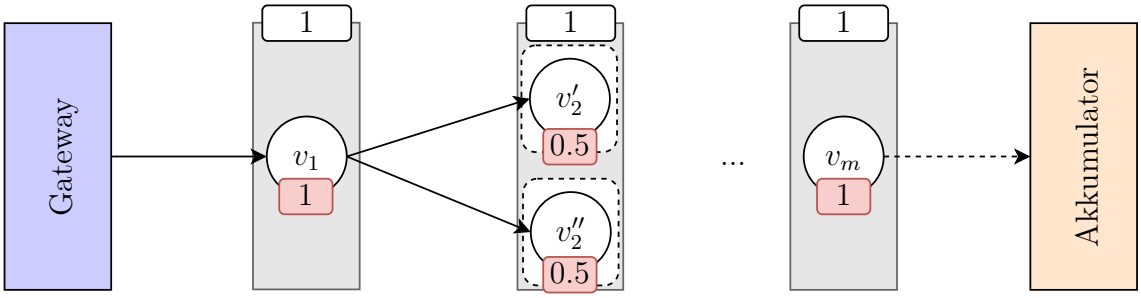


Abbildung 8.14.: azyklischer gerichteter Graph ohne Mehrfachkanten mit einem exemplarischen Auswirkungsgrad η pro Knoten

Der Gesamt Auswirkungsgrad $\eta(\Lambda)$ bei einem azyklischer gerichteter Graph ohne Mehrfachkanten wird über das Produkt der korrumpierten Knoten eines Layers berechnet. Der Schutzgrad $\rho(\Lambda)$ entspricht weiterhin $1 - \eta(\Lambda)$.

$$\eta(\Lambda) = 1 - \prod_{C_l \in \Lambda} 1 - \eta(C_l) \tag{8.11}$$

Durch die Betrachtung des Auswirkungsgrades η können die wichtigsten Knoten im Algorithmus identifiziert werden. Diese entsprechen auch den am meisten zu schützenden Knoten. Der Auswirkungsgrad kann nur verringert werden, indem bestimmte Knoten nicht mehr korrumpiert werden oder indem die Auswirkungsverteilungen auf die korrumpierten Knoten verringert werden. Wenn nicht klar ist, welche Knoten korrumpiert werden, ist die Gleichverteilung der Auswirkungen auf alle Knoten eines Layers das erreichbare Optimum.

Die Angriffserfolgswahrscheinlichkeit γ gibt an, wie erfolgreich ein Angriff auf einen gegeben Algorithmus ist. Diese ist dem Auswirkungsgrad η gleichzusetzen:

$$\gamma(C) = \eta(C) \tag{8.12}$$

Neben den bereits beschriebenen Metriken *a*) dem Korruptionsgrad κ, *b*) der Integrität ι, *c*) dem Auswirkungsgrad η, *d*) dem Schutzgrad ρ und *e*) der

Angriffserfolgswahrscheinlichkeit γ ist auch die Auswahlwahrscheinlichkeit zentral. Diese beschreibt die Wahrscheinlichkeit, dass ein Angriff einen für einen bestimmten Algorithmus verwendeten Knoten c in einem DDVN korrumpiert.

In einem DDVN können 100 Validatoren vorhanden sein, aber nur 20 für einen bestimmten Algorithmus verwendet werden. Auf diese Weise kann ein Angriff einen nicht verwendeten Knoten korrumpieren, ohne die Sicherheit des Systems zu gefährden. Sei v die Anzahl aller Validierer im DDVN (die Validierer werden als voneinander unabhängig angenommen), c ($c \leq v$) die Anzahl aller korrumpierten Validierer im DDVN, m die Anzahl der für den Algorithmus verwendeten Validierer und x ($x \leq m$) die Anzahl der korrumpierten Knoten im Algorithmus ist, dann folgt, dass die Auswahlwahrscheinlichkeit für x korrumpierte Knoten in m der hypergeometrischen Verteilung entspricht:

$$p = \frac{\binom{c}{x} \cdot \binom{v-c}{m-x}}{\binom{v}{m}} \tag{8.13}$$

Die Wahrscheinlichkeit, dass ein korrumpierter Knoten beim Durchlaufen eines Algorithmus gewählt wird, hängt von drei Faktoren ab: der Datenstruktur des Algorithmus, der Eingangsdaten und den Regeln in den durchlaufenen Knoten. Zusammengefasst entsprechen die drei Faktoren dem gewählten Pfad durch den Algorithmus.

Bei der Verwendung eines azyklischer gerichteter Graph ohne Mehrfachkanten mit einem maximalen Ausgangsgrad von eins, bei dem jeder Layer einem eigenen Knoten entspricht, verhält sich die Auswahlwahrscheinlichkeit wie in Abbildung 8.15 dargestellt. Hierbei besteht das gesamte DDVN aus 100 Knoten und die Anzahl der korrumpierten Knoten wird zwischen $[1, 10]$ variiert. Es ist ersichtlich, dass bereits bei einer niedrigen Anzahl an korrumpierten Knoten (5, 10) die Auswahlwahrscheinlichkeit über 20% bei 5 Layern liegt.

In Abbildung 8.16 wird die Anzahl der Knoten im DDVN mit 100.000 angenommen. Die Anzahl der korrumpierten Knoten bleibt bei den zuvor verwendeten $\{1, 5, 10, 50\}$. Die Wahrscheinlichkeit, dass mindestens ein korrupter Knoten im Graphen enthalten ist, nimmt mit $c = 50$ stark zu. Sobald in einem gerichteten Graphen ein korrumpierter Knoten enthalten ist, ist der Auswirkungsgrad $\eta = 1$. Aus beiden Analysen geht ausserdem hervor, dass bei konstanter Anzahl korrupter Knoten c und steigender Anzahl verwendeter Layer im azyklischer gerichteter Graph ohne Mehrfachkanten die Wahrscheinlichkeit, einen korrupten Knoten zu enthalten, steigt. Daher sollte die Anzahl der Layer so gering wie möglich gehalten werden, da $\lim_{l \to n} \eta(l) = 1$.

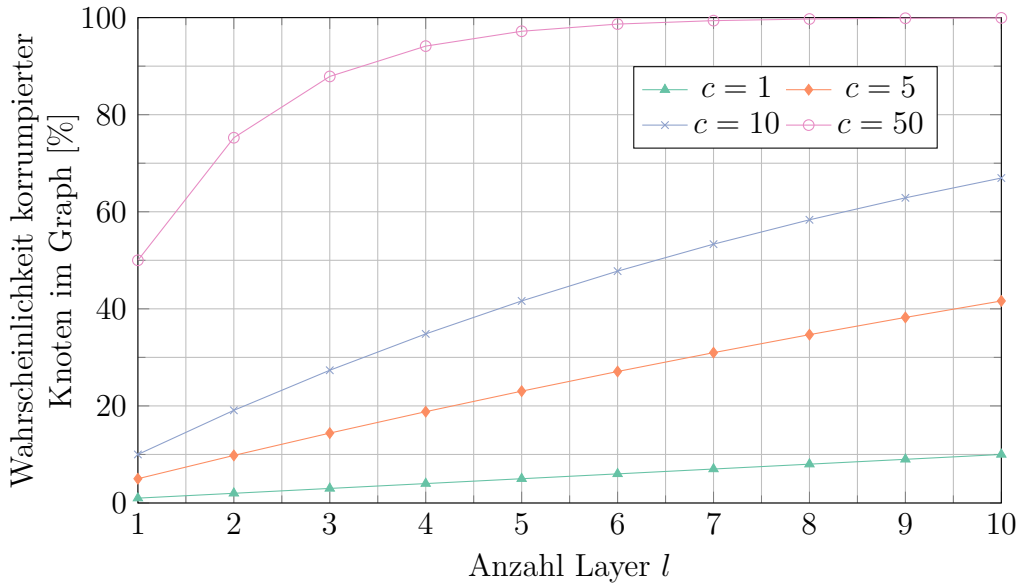


Abbildung 8.15.: Wahrscheinlichkeit, dass ein oder mehr korrumpierte Knoten c für den azyklischer gerichteter Graph ohne Mehrfachkanten mit der Layeranzahl $l = [1, 10]$ und der Gesamtknotenanzahl $n = 100$ ausgewählt werden

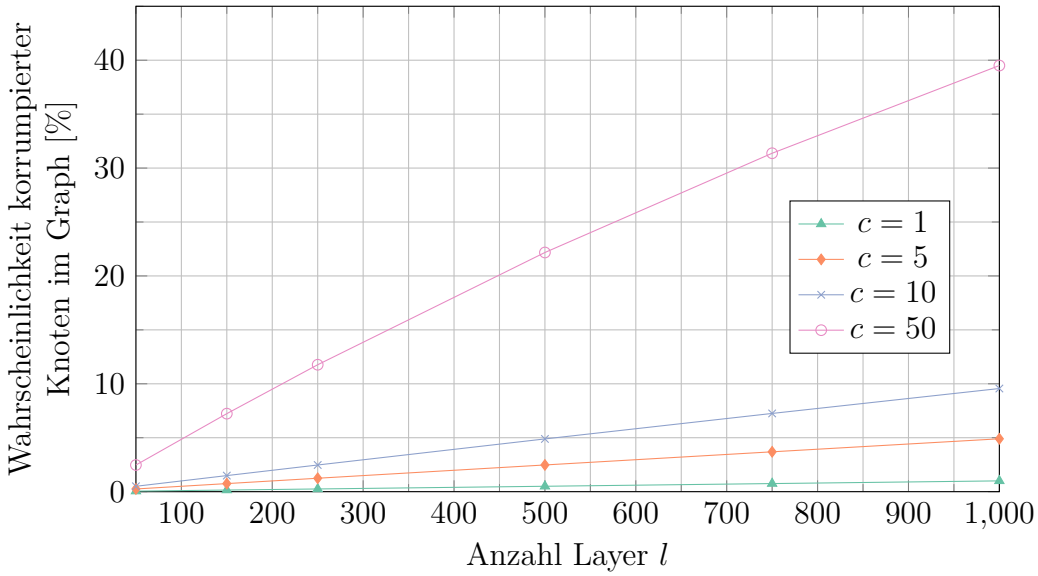


Abbildung 8.16.: Wahrscheinlichkeit, dass ein oder mehr korrumpierte Knoten c für den azyklischer gerichteter Graph ohne Mehrfachkanten mit der Layeranzahl $l = [50, 150, 250, 500, 750, 1000]$ und der Gesamtknotenanzahl $n = 100.000$ ausgewählt werden

8.5. Ersetzen von Knoten mit einem Quorum

Bei einem Quorum Ansatz wird die Verantwortung für die Entscheidungsfindung auf eine Menge von Validierern $Q = \{q'_1, q'_2, ...\}$ verteilt (siehe Abbildung 8.17 und Abbildung 8.24). Durch die Bildung eines Quorums, d.h. eine definierte Anzahl von Stimmen muss der gleichen Meinung sein, wird die Entscheidung bestimmt. Kann kein Quorum gebildet werden, so wird kein Validierungsergebnis erzielt und der Akkumulator muss ohne eine eindeutige Entscheidung des Quorums auskommen. Durch die Verwendung eines Quorums wird implizit die Anzahl der verwendeten Validierer erhöht. Dadurch sinkt γ bei gleicher Anzahl korrupter Knoten c.

8.5.1. Betrachtung eines Graphen

Im Folgenden werden einige Theoreme für die Verwendung von Quoren mit einer Graphenstruktur vorgestellt und anhand von durchgeführten Analysen erläutert. Der grundsätzliche Aufbau einer Graphenstruktur mit einem Quorum ist in Abbildung 8.17 dargestellt.

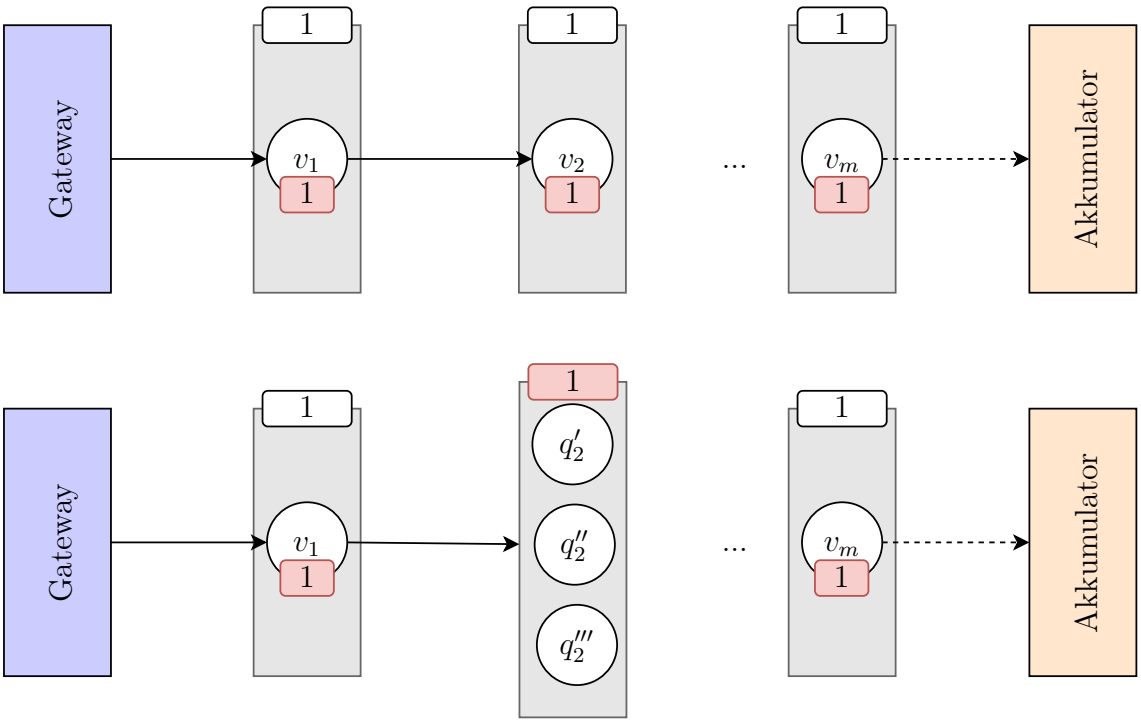


Abbildung 8.17.: Ersetzen einer Validiererentscheidung durch eine Quorumsentscheidung für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Theorem 8.5.1. *Die Verwendung eines Quorums verringert die Erfolgswahrscheinlichkeit γ eines Angriffs, solange die Anzahl der korrumpierten Knoten gleich bleibt.*

Wenn eine Quorum Entscheidung verwendet wird, kann der Angriff einen Knoten im Quorum korrumpieren, ohne dass γ steigt. Liegt die Anzahl der korrumpierten Knoten c unter einem bestimmten Schwellenwert $t > \frac{c}{|Q|}$, kann der Angriff keine unabhängige Entscheidung treffen. Die Verwendung eines Quorums verringert also die Erfolgswahrscheinlichkeit eines Angriffs. Abbildung 8.18 und Abbildung 8.19 zeigen die Auswirkungen der Verwendung von Quorum Entscheidungen für einen azyklischen gerichteten Graphen ohne Mehrfachkanten mit einer Layeranzahl l von 10. Die Anzahl der verwendeten Quorum Entscheidungen ist q. Das Maximum für die Anzahl der Validiererentscheidungen, die durch eine Quorum Entscheidung geschützt sind, entspricht der Anzahl der Layer $q_{max} = l$. Für beide Abbildungen werden drei verschiedene Anzahlen von korrumpierten Layern $c = \{1, 5, 10\}$ und 100.000 Iterationen pro q und l betrachtet.

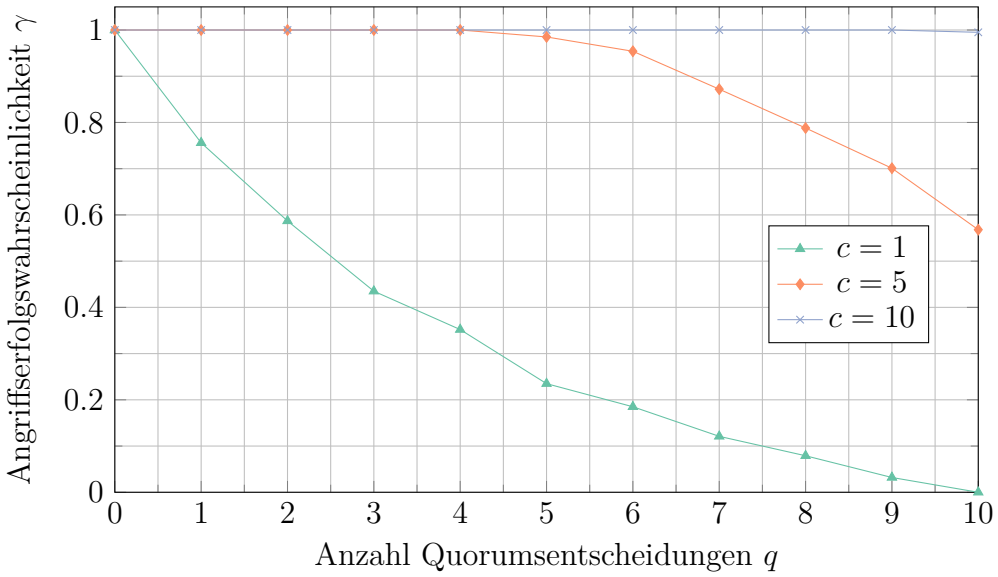


Abbildung 8.18.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer Layeranzahl $l = 10$, einer Quorumsgrösse $|Q| = 3$ und einer variablen Quorumsanzahl q für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Theorem 8.5.2. *In einem azyklischen gerichteten Graphen ohne Mehrfachkanten, in dem jede Validiererentscheidung durch eine Quorumsentscheidung geschützt ist, ist die Wahrscheinlichkeit eines erfolgreichen Angriffs $\gamma = 0$, wenn nur ein Knoten korrumpiert wird.*

Unter der Annahme, dass der Schwellwert für eine Quorumsentscheidung $t = \frac{1}{2}$ ist, kann eine Quorumsentscheidung bei einer Quorumsgrösse $|Q| = 3$ mit 2 gleichen

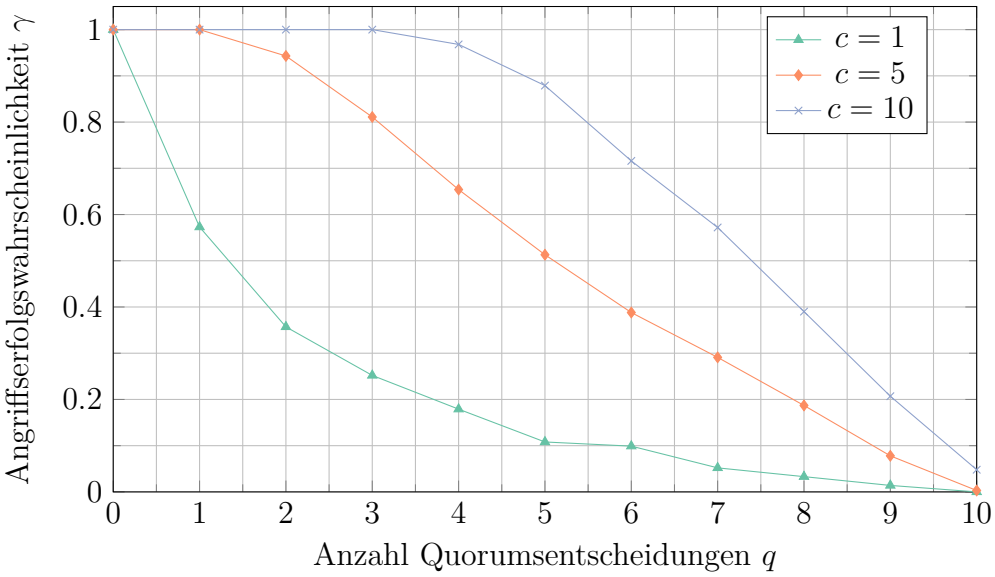


Abbildung 8.19.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer Layeranzahl $l = 10$, einer Quorumsgrösse $|Q| = 7$ und einer variablen Quorumsanzahl q für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Stimmen erreicht werden. Wenn nur ein Validierer erfolgreich korrumpiert wird, kann der Angriff niemals die minimale Anzahl von Stimmen für eine Quorumsentscheidung erreichen. Dies ist in Abbildung 8.18 und Abbildung 8.19 für $c = 1$ und $q = 10$ dargestellt. In beiden Abbildungen ist $\gamma = 0$.

Theorem 8.5.3. *Die Verwendung eines Quorums wirkt sich negativ auf den Schutzgrad ρ aus, wenn die Anzahl der korrupten Knoten c proportional zur Anzahl der Schichten l steigt.*

Die zusätzliche Korrumpierung von Validierern in einer konstanten Datenstruktur führt immer zu einer Erhöhung oder Beibehaltung des Auswirkungsgrades η. Sofern die Quorumsgrösse $|Q|$ konstant bleibt, die Datenstruktur gleich bleibt und sich nur die Anzahl der Layer erhöht, hat der Angriff eine höhere Wahrscheinlichkeit, ein Quorum zu übernehmen, da sich die Quorumsgrösse nicht proportional zur Anzahl der korrumpierten Knoten verhält. Die Abbildung 8.20 zeigt vier verschiedene Layeranzahlen $l = [10, 20, 40, 60]$ mit verschiedenen Gesamtknotenzahlen $n = [30, 60, 120, 180]$ und einer korrumpierten Knotenzahl $c = \frac{n}{3}$, die proportional zur Gesamtknotenzahl wächst. Es ist ersichtlich, dass die geringste Angriffserfolgswahrscheinlichkeit γ bei $l = 10$ und die höchste bei $l = 60$ vorliegt.

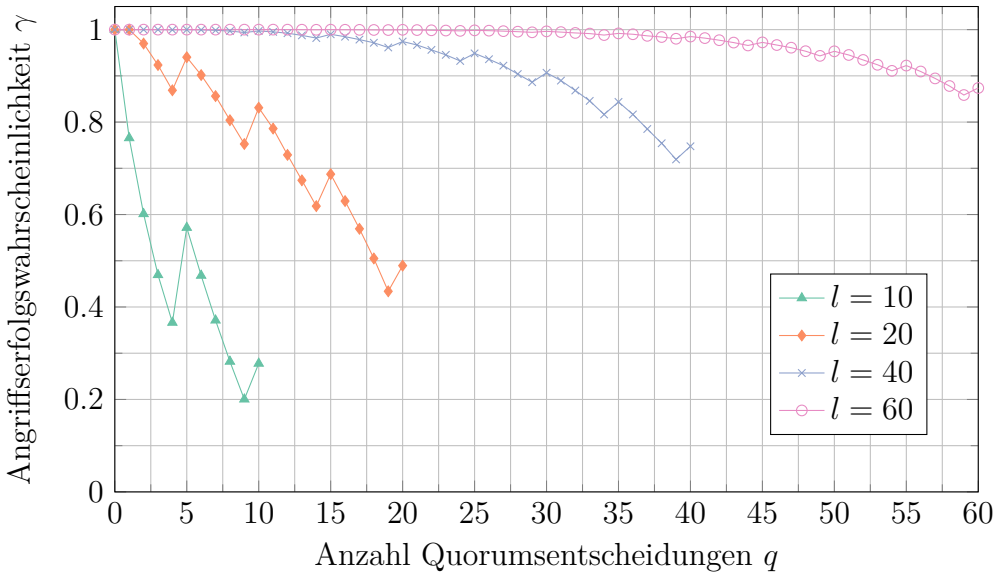


Abbildung 8.20.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer Layeranzahl $l = [10, 20, 40, 60]$, einer Quorumsgrösse $|Q| = 3$, einer variablen Quorumsanzahl q im Bereich von $[0, l]$ und einer proportionalen Anzahl korrumpierter Knoten von $c = \frac{n}{10}$ für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Theorem 8.5.4. *Eine Erhöhung des Quorums $|Q|$ führt zu einer grösseren Verbesserung des Schutzgrades ρ als eine Erhöhung der Anzahl der Layer l.*

Durch die Erhöhung des Quorums $|Q|$ wird es für eine angreifende Instanz schwieriger, die Entscheidungsfindung zu seinen oder ihren Gunsten zu beeinflussen. Abbildung 8.22 und Abbildung 8.21 vergleichen die beiden Ansätze der Erhöhung der Quorumsgrösse und der Erhöhung der Anzahl der Layer. Aus den beiden Abbildungen geht hervor, dass bei einer Gesamtknotenanzahl von $n = l \cdot |Q| = 75$ die Erhöhung der Quorumsgrösse $|Q|$ zu einer grösseren Verbesserung führt. Beim Quorumsgrössenansatz ist die Erfolgswahrscheinlichkeit des Angriffs bei $|Q| = 25$ kleiner als 1%, beim Layeranzahlansatz ist die Erfolgswahrscheinlichkeit des Angriffs bei $l = 25$ stark von der Anzahl der korrumpierten Knoten abhängig. Im Vergleich dazu beträgt bei $c = 25$ die Angriffserfolgswahrscheinlichkeit mehr als 81%. Bei $c = 10$ beträgt $\gamma_l > 22\%$ (Layeranzahlansatz) und $\gamma_s < 1\%$ (Quorumsgrössenansatz), was einem Unterschied von mehr als 20% entspricht.

Im Bereich von $|Q| = [1, 5]$ bzw. $l = [1, 5]$ ist der Layeranzahl Ansatz dem Quorumsgrössen Ansatz überlegen. Der Grund dafür ist, dass durch die Annahme einer festen Quorumsgrösse $|Q| = 5$ beim Layeranzahl Ansatz die Quorumsgrösse höher ist als beim Quorumsgrössen Ansatz. Erst ab $|Q| > 5$ hat der Quorumsgrössenansatz tatsächlich eine höhere Quorumsgrösse.

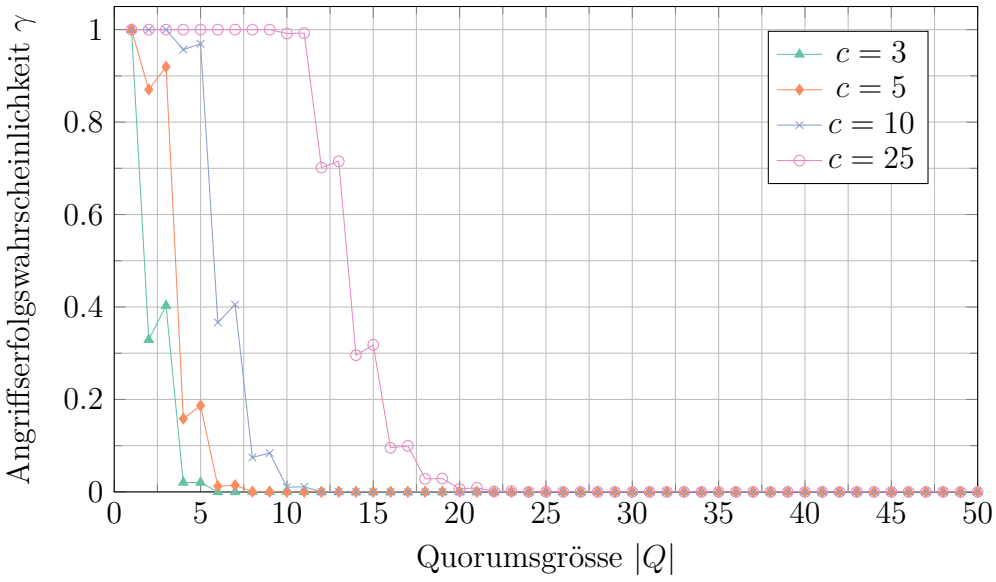


Abbildung 8.21.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer variablen Quorumsgrösse im Bereich von $|Q| = [0, 50]$, einer fixen Layeranzahl $l = 5$, einer Quorumsanzahl $q = l$ und einer unterschiedlichen Anzahl an korrumpierten Knoten $c = \{3, 5, 10, 25\}$ für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

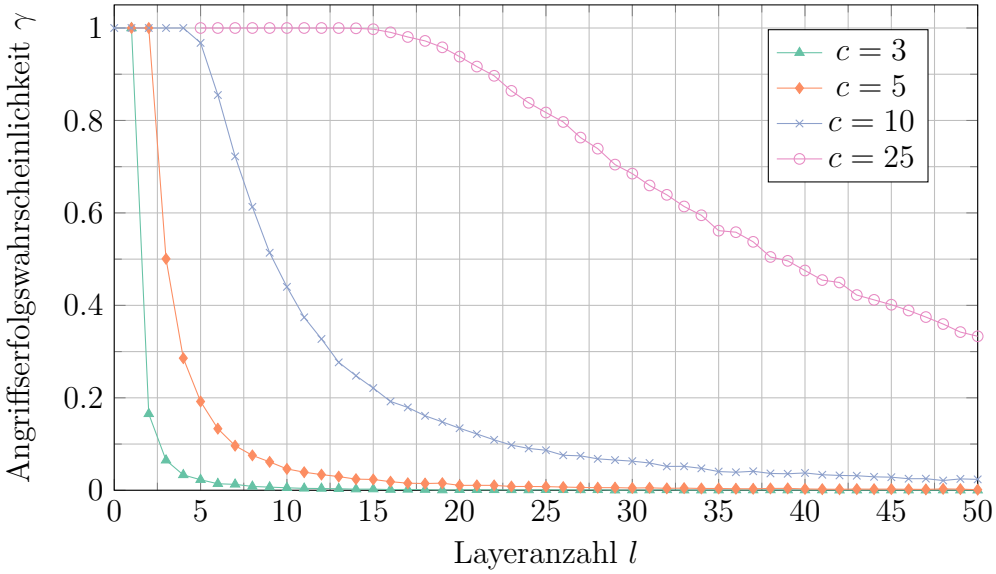


Abbildung 8.22.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer variablen Layeranzahl im Bereich von $l = [0, 50]$, einer Quorumsgrösse $|Q| = 5$, einer Quorumsanzahl $q = l$ und einer unterschiedlichen Anzahl an korrumpierten Knoten $c = \{3, 5, 10, 25\}$ für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

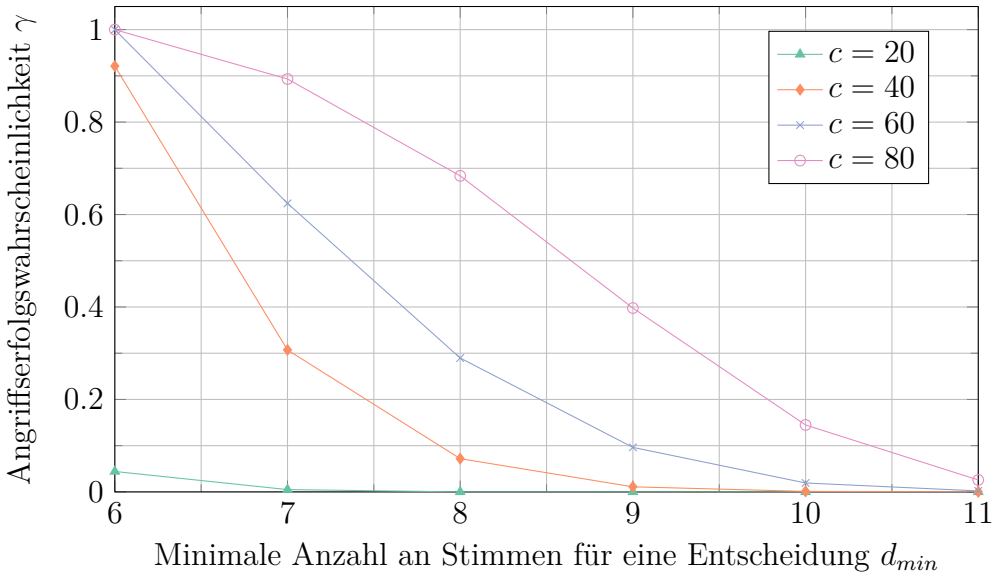


Abbildung 8.23.: Vergleich der Angriffserfolgswahrscheinlichkeit γ bei einer Quorumsgrösse $|Q| = 11$, einer Layeranzahl $l = 10$, einer Quorumsanzahl $q = l$ und einer unterschiedlichen Anzahl an korrumpierten Knoten $c = \{20, 40, 60, 80\}$ für einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Theorem 8.5.5. *Das Erhöhen des Schwellwerts t bei einer Quorumsentscheidung erniedrigt die Angriffserfolgswahrscheinlichkeit γ.*

Damit ein Quorum eine Entscheidung treffen kann, muss eine bestimmte Anzahl von Stimmen für die gleiche Entscheidung abgegeben werden. Die minimale Anzahl gleicher Stimmen für eine Entscheidung wird als d_{min} bezeichnet. In der vorliegenden Analyse wird immer ein $d_{min} > \frac{1}{2} \cdot |Q|$ an. Mit steigender Anzahl der Teilnehmenden in einem Quorum steigt üblicherweise auch d_{min}. In Abbildung 8.23 wird ein azyklischer gerichteter Graph ohne Mehrfachkanten mit einer Layeranzahl von $l = 10$ und einer Quorumsgrösse von $|Q| = 11$ untersucht. Es wird gezeigt, mit welcher Wahrscheinlichkeit ein Angriff genügend Validierer in einem Quorum überzeugen oder übernehmen kann, um die Entscheidung des Quorums zu kontrollieren. Bei $c = 20$ hat der Angriff selbst bei $d_{min} = 6$ nur eine sehr geringe Erfolgswahrscheinlichkeit $\gamma < 5\%$. Bei einem hohen Verhältnis von $\frac{c}{n} = \frac{80}{110}$ kann die Sicherheit im DDVN hoch gehalten werden, da alle Teilnehmenden im Quorum der gleichen Meinung sein müssen. Die Erfolgswahrscheinlichkeit eines Angriffs *gamma* ist $< 3\%$, wenn angenommen wird, dass der Angriff in mindestens einem der 10 Quoren eine Entscheidung erzwingen muss.

8.5.2. Betrachtung einer Baumstruktur

Wie bei einem azyklischer gerichteter Graph ohne Mehrfachkanten wird auch bei einer Baumstruktur ein Validierer durch eine Menge von Validierern ersetzt (siehe Abbildung 8.24). Der wesentliche Unterschied zum Graphen besteht einzig im Durchlaufen der Pfade, bei der Baumstruktur gibt es pro Layer nur einen Validierer, der besucht wird. Der Validierer selbst kann durch eine Menge von Validierern mit einer Quorumsbildung ersetzt werden.

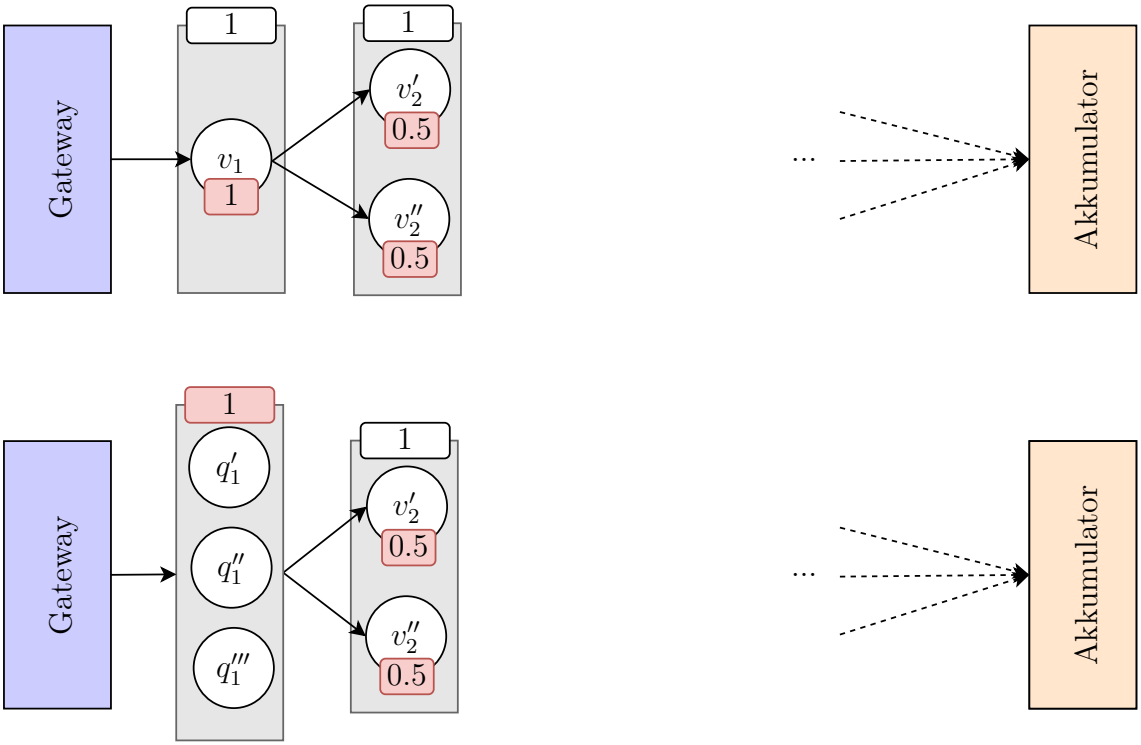


Abbildung 8.24.: Ersetzen einer Validiererentscheidung durch eine Quorumentscheidung bei einer Baumstruktur

Durch das Ersetzen eines Validierers durch ein Quorum erhöht sich der Rechenaufwand pro Validierer, d.h. die Anzahl der Berechnungen beträgt $calc_n = |Q|$. Dieser Mehraufwand darf bei der Erstellung von DDVN mit Quoren nicht vernachlässigt werden.

Theorem 8.5.6. *Die Verwendung eines Quorums verringert die Erfolgswahrscheinlichkeit γ eines Angriffs, solange die Anzahl der korrumpierten Knoten gleich bleibt.*

In Abbildung 8.25 und Abbildung 8.26 werden zwei verschiedene Baumstruktur mit und ohne Quoren betrachtet. Die Kennlinien entsprechen jeweils einer Baumstruktur, die analysiert wird. Dabei ist $|N|$ die Gesamtzahl der Knoten in der

Baumstruktur. Die Indizes N_{Q0}, N_{Q1} und N_{Q2} geben an, wie viele Quoren in der Baumstruktur verwendet werden ($Q0 = 0$, $Q1 = 1$ und $Q2 = 2$). Die Anzahl der Quoren ist unabhängig von der Anzahl der Knoten und beträgt immer drei.

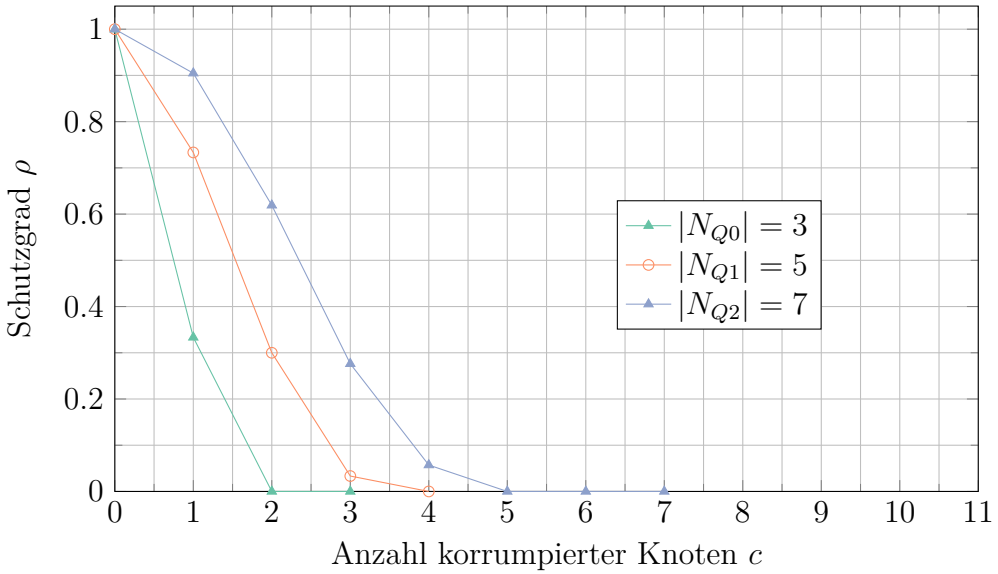


Abbildung 8.25.: Vergleich des Schutzgrad ρ für eine Baumstruktur mit drei Knoten ohne Schutzmechanismus und mit zwei bzw. drei Quoren mit der Quorumsgrösse $|Q| = 3$

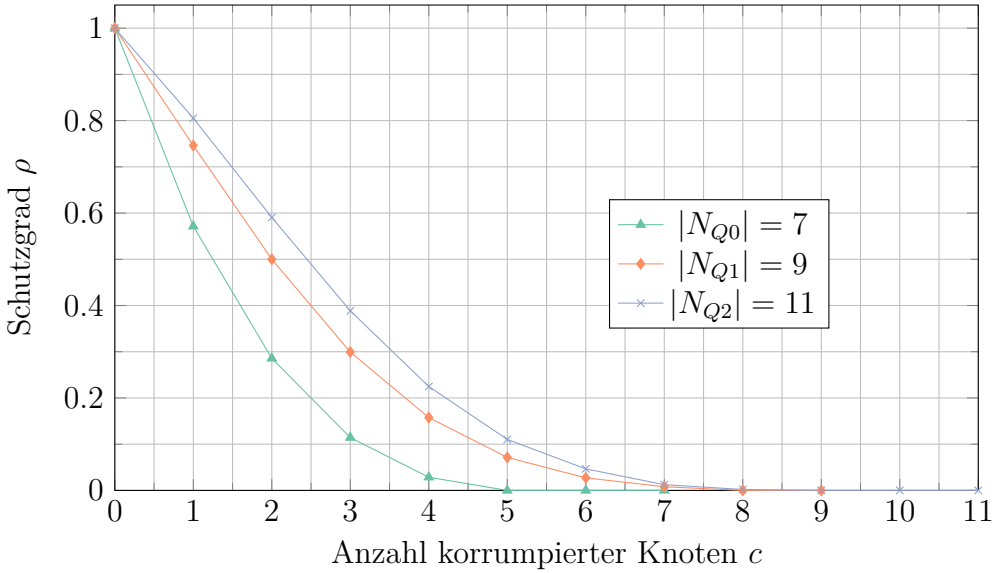


Abbildung 8.26.: Vergleich des Schutzgrad ρ für eine Baumstruktur mit sieben Knoten ohne Schutzmechanismus und mit zwei bzw. drei Quoren mit der Quorumsgrösse $|Q| = 3$

Eine weitere Analyse beinhaltet eine tabellarische Betrachtung der Auswirkungen unterschiedlicher Anzahlen korrupter Knoten auf unterschiedlich grosse Baumstrukturen. In Tabelle 8.10 ist die Auswirkung auf den Schutzgrad ρ dargestellt. Für eine kleine Baumstruktur mit 7 oder 15 Knoten führt das Hinzufügen eines kompletten Layers zu einer Verbesserung von 16% bzw. 10%. Je grösser die Baumstruktur, desto geringer ist die Verbesserung des Schutzgrades durch Hinzufügen eines weiteren Layers. Von 31 Knoten auf 63 Knoten beträgt die Verbesserung nur 6%, obwohl nun 32 Knoten mehr vorhanden sind.
Die Untersuchung zeigt zudem, dass die Verwendung eines kleinen Quorums mit $|Q| = 3$ vor allem bei einer kleinen Baumstruktur zu einer Verbesserung führt. Hier verhält es sich ähnlich wie bei der Layerergänzung.
Im dritten Abschnitt sind jeweils ca. 20% der Baumstruktur korrupt. Daraus folgt, dass sich der Schutzgrad ρ trotz steigender Anzahl der verwendeten Knoten stark verschlechtert. Bei 63 Knoten liegt der Schutzgrad bei 25%, d.h. nur jede vierte Validierung des DDVN ist nicht manipuliert.
Im letzten Abschnitt werden jeweils 20% der geschützten Knoten und 20% der korrumpierten Knoten betrachtet. Es zeigt sich, dass die Verwendung von mehreren Quoren einen positiven Effekt auf den Schutzgrad ρ der Baumstruktur hat. Bei 63 Knoten mit 13 quorumsgeschützten Knoten beträgt die Verbesserung gegenüber einem quorumsgeschützten Knoten 18%.

Tabelle 8.10.: Auswirkung der Verwendung von Quoren auf den Schutzgrad ρ bei einer Baumstruktur

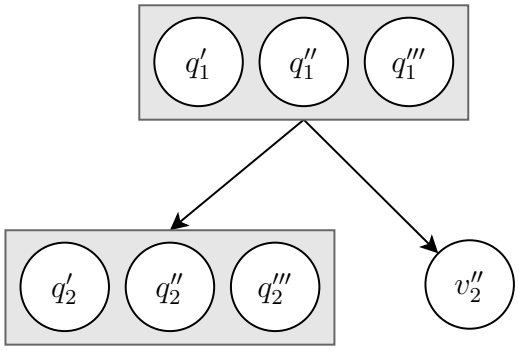
Anzahl Knoten	**Anzahl Quoren**	**Anzahl korrumpierter Knoten**	ρ
Ohne Quorum und ein korrupter Knoten			
7	0	1	57.16%
15	0	1	73.49%
31	0	1	83.93%
63	0	1	90.42%
Ein Quorum und ein korrupter Knoten			
7	1	1	71.43%
15	1	1	78.04%
31 (Simulation)	1	1	85.36%
63 (Simulation)	1	1	90.91%

Tabelle 8.10.: Auswirkung der Verwendung von Quoren auf den Schutzgrad ρ bei einer Baumstruktur

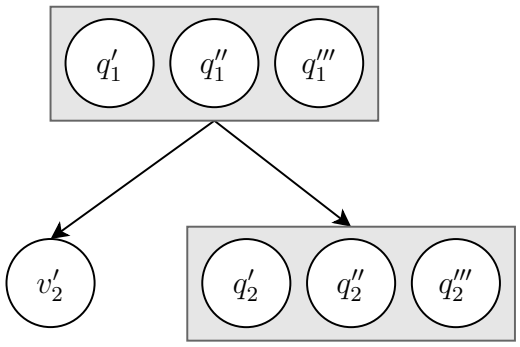
Anzahl Knoten	**Anzahl Quoren**	**Anzahl korrumpierter Knoten**	ρ
Ein Quorum und 20% korrupte Knoten			
7	1	1	71.43%
15	1	3	43.78%
31 (Simulation)	1	6	34.76%
63 (Simulation)	1	13	24.98%
20% an Quoren und korrupten Knoten			
7	1	1	71.43%
15	3	3	56.68%
31 (Simulation)	6	6	50.71%
63 (Simulation)	13	13	43.13%

Ein wichtiger Faktor bei der Verwendung eines Quorums im Zusammenhang mit einer Baumstruktur ist die Positionierung des Quorums. Das Quorum kann entweder in der Nähe des Wurzelknotens oder in der Nähe der Blattknoten positioniert werden. Gegeben ist eine Baumstruktur mit drei Knoten, auf diese drei Knoten werden zwei Quoren mit einer Quorumsgrösse von jeweils drei Knoten verteilt. Es gibt drei mögliche Positionierungen 1) des Wurzelknotens und des linken Blattknotens (siehe 8.27a), 2) des Wurzelknotens und des rechten Blattknotens (siehe 8.27b) und 3) des linken Blattknotens und des rechten Blattknotens (siehe Abbildung 8.28). Ausserdem werden zwei korrumpierte Knoten angenommen, die beliebig positioniert sein können. Für die Positionierung gibt es $\binom{7}{2} = 21$ Möglichkeiten. Neben den 21 Positionierungsmöglichkeiten gibt es jeweils zwei Pfade durch die Baumstruktur. Entweder durch den Wurzelknoten zum linken Blattknoten oder durch den Wurzelknoten zum rechten Blattknoten. Zusammengefasst entspricht dies 42 zu untersuchenden Fällen pro Quorumspositionierung.

Eine detaillierte Analyse der Positionierungsmöglichkeiten sowie deren Korruptionsmöglichkeiten ist in Tabelle 8.11 und in Tabelle 8.12 gegeben. Dort sind alle Positionierungsmöglichkeiten der korrumpierten Knoten sowie die beiden einzigen Traversierungsmöglichkeiten durch die Baumstruktur aufgelistet: Linker Pfad und Rechter Pfad. Bei der Positionierung eines Quorums am Wurzelknoten



(a) Einfügen der Quoren am Wurzelknoten und am linken Blattknoten



(b) Einfügen der Quoren am Wurzelknoten und am rechten Blattknoten

Abbildung 8.27.: Einfügen von Quoren am Wurzelknoten und an einem Blattknoten

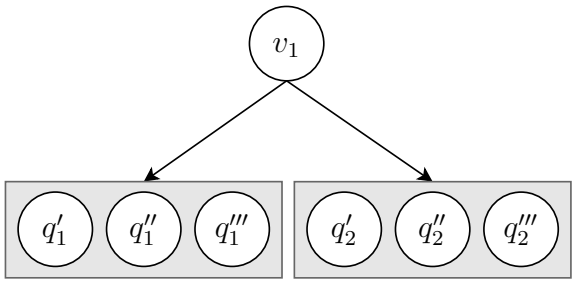


Abbildung 8.28.: Einfügen der Quoren am linken Blattknoten und am rechten Blattknoten

sind $\frac{15}{42}$ der möglichen Pfade korrumpiert, dies entspricht einem ρ von 35.71%. Bei einer Positionierung der Quoren an den beiden Blattknoten ergibt sich ein $\rho = \frac{18}{42} = 42.86\%$. Aus den beiden vorliegenden Analysen geht hervor, dass bei gleicher Verteilung der Traversierungen der Baumstruktur die Positionierung des Quorums am Wurzelknoten einen um 7% höheren Schutzgrad aufweist.

Tabelle 8.11.: Permutation von zwei korrumpierten Knoten mit zwei Quoren an den Knoten v_1 (q'_1, q''_1, q'''_1) und v_2 (q'_2, q''_2, q'''_2), die beiden Pfade (links und rechts) geben an, ob eine Korruption bei einem Durchlauf der Baumstruktur möglich ist

Nr.	q'_1	q''_1	q'''_1	q'_2	q''_2	q'''_2	v_3	**Linker Pfad**	**Rechter Pfad**
1.	0	0	0	0	0	1	1	nein	ja
2.	0	1	0	0	1	0	0	nein	nein
3.	1	0	1	0	0	0	0	ja	ja
4.	0	0	0	0	1	0	1	nein	ja
5.	0	0	0	1	1	0	0	ja	nein
6.	1	0	0	1	0	0	0	nein	nein

Tabelle 8.11.: Permutation von zwei korrumpierten Knoten mit zwei Quoren an den Knoten v_1 (q_1', q_1'', q_1''') und v_2 (q_2', q_2'', q_2'''), die beiden Pfade (links und rechts) geben an, ob eine Korruption bei einem Durchlauf der Baumstruktur möglich ist

Nr.	q_1'	q_1''	q_1'''	q_2'	q_2''	q_2'''	v_3	**Linker Pfad**	**Rechter Pfad**
7.	1	0	0	0	0	1	0	nein	nein
8.	1	1	0	0	0	0	0	ja	ja
9.	1	0	0	0	0	0	1	nein	ja
10.	0	0	1	1	0	0	0	nein	nein
11.	0	1	0	0	0	1	0	nein	nein
12.	0	1	0	0	0	0	1	nein	ja
13.	0	1	1	0	0	0	0	ja	ja
14.	0	0	0	1	0	1	0	ja	nein
15.	1	0	0	0	1	0	0	nein	nein
16.	0	1	0	1	0	0	0	nein	nein
17.	0	0	1	0	0	1	0	nein	nein
18.	0	0	0	1	0	0	1	nein	ja
19.	0	0	1	0	0	0	1	nein	ja
20.	0	0	1	0	1	0	0	nein	nein
21.	0	0	0	0	1	1	0	ja	nein
		Gesamt:						**6**	**9**

Tabelle 8.12.: Permutation von zwei korrumpierten Knoten mit zwei Quoren an den Knoten v_2 (q_2', q_2'', q_2''') und v_3 (q_3', q_3'', q_3'''), die beiden Pfade (links und rechts) geben an, ob eine Korruption bei einem Durchlauf der Baumstruktur möglich ist

Nr.	v_1	q_2'	q_2''	q_2'''	q_3'	q_3''	q_3'''	**Linker Pfad**	**Rechter Pfad**
1.	0	0	0	0	0	1	1	nein	ja
2.	0	1	0	0	1	0	0	nein	nein
3.	1	0	1	0	0	0	0	ja	ja
4.	0	0	0	0	1	0	1	nein	ja
5.	0	0	0	1	1	0	0	nein	nein
6.	1	0	0	1	0	0	0	ja	ja
7.	1	0	0	0	0	1	0	ja	ja
8.	1	1	0	0	0	0	0	ja	ja
9.	1	0	0	0	0	0	1	ja	ja
10.	0	0	1	1	0	0	0	ja	nein
11.	0	1	0	0	0	1	0	nein	nein
12.	0	1	0	0	0	0	1	nein	nein
13.	0	1	1	0	0	0	0	ja	nein
14.	0	0	0	1	0	1	0	nein	nein

Tabelle 8.12.: Permutation von zwei korrumpierten Knoten mit zwei Quoren an den Knoten v_2 (q'_2, q''_2, q'''_2) und v_3 (q'_3, q''_3, q'''_3), die beiden Pfade (links und rechts) geben an, ob eine Korruption bei einem Durchlauf der Baumstruktur möglich ist

Nr.	v_1	q'_2	q''_2	q'''_2	q'_3	q''_3	q'''_3	**Linker Pfad**	**Rechter Pfad**
15.	1	0	0	0	1	0	0	ja	ja
16.	0	1	0	1	0	0	0	ja	nein
17.	0	0	1	0	0	1	0	nein	nein
18.	0	0	0	1	0	0	1	nein	nein
19.	0	0	1	0	0	0	1	nein	nein
20.	0	0	1	0	1	0	0	nein	nein
21.	0	0	0	0	1	1	0	nein	ja
		Gesamt:						**9**	**9**

In Tabelle 8.13 ist ein Vergleich zwischen einer Positionierung oben (in der Nähe des Wurzelknotens) und einer Positionierung unten (in der Nähe der Blattknoten) mit einer grösseren Anzahl von Quoren und korrumpierten Knoten dargestellt. Bei Verwendung eines Quorums und eines korrupten Knotens ist die beste Positionierung des Quorums immer direkt auf dem Wurzelknoten, da so jeder Durchlauf durch die Baumstruktur geschützt ist. Je weiter unten das Quorum gesetzt wird, desto weniger Durchläufe werden geschützt. Auch bei einer steigenden Anzahl korrupter Knoten (20%) ist eine Positionierung oben besser als eine Positionierung unten. Im letzten Abschnitt 20% *an Quoren und korrupten Knoten* ist ersichtlich, dass die Oben-Positionierung in der vorliegenden Simulation zu einem Unterschied von $> 14\%$ im Schutzfaktor ρ führt. Bei einer kleinen Baumstruktur von 15 Knoten beträgt die Differenz mehr als 20% im Vergleich zur Unten-Positionierung.

Tabelle 8.13.: Auswirkung der Position des Quorums auf den Schutzgrad ρ in einer Baumstruktur

Anzahl Knoten	**Anzahl Quoren**	**Anzahl korrumpierter Knoten**	ρ
Ein Quorum und ein korrupter Knoten			
15 (Oben)	1	1	82.3%
15 (Unten)	1	1	77.18%
31 (Oben)	1	1	87.98%
31 (Unten)	1	1	84.97%

Tabelle 8.13.: Auswirkung der Position des Quorums auf den Schutzgrad ρ in einer Baumstruktur

Anzahl Knoten	**Anzahl Quoren**	**Anzahl korrumpierter Knoten**	ρ
63 (Oben)	1	1	92.37%
63 (Unten)	1	1	90.42%
Ein Quorum und 20% korrupte Knoten			
15 (Oben)	1	3	48.62%
15 (Unten)	1	3	42.63%
31 (Oben)	1	6	38.75%
31 (Unten)	1	6	34.34%
63 (Oben)	1	13	27.90%
63 (Unten)	1	13	24.93%
20% an Quoren und korrupten Knoten			
15 (Oben)	3	3	65.25%
15 (Unten)	3	3	53.94%
31 (Oben)	6	6	59.98%
31 (Unten)	6	6	47.11%
63 (Oben)	13	13	54.06%
63 (Unten)	13	13	39.37%

Anhand der aufgezeigten Analysen ist ersichtlich, dass bei einer Baumstruktur ohne Schutzmechanismus, der Schutzgrad ρ niedriger ist, als bei einer mit einem Quorum.

8.6. Aufteilung der Validierer mit hoher Auswirkung

Im Folgenden wird ein Ansatz vorgestellt, um die Auswirkungen eines erfolgreichen Angriffs auf einen Validierer zu minimieren. Einige Validierer im DDVN haben einen höheren Auswirkungsgrad η als andere Validierer. Um die grösste Sicherheitsverbesserung zu erzielen, sollten diese Knoten zuerst aufgeteilt werden.

Dadurch wird ihr Einfluss auf die neuen Knoten verteilt (siehe Abbildung 8.29 und Abbildung 8.30) und die Korruption eines einzelnen Knotens hat einen geringeren Einfluss.

8.6.1. Betrachtung eines Graphen

In Abbildung 8.29 wird die Aufteilung des Knotens v_2 in die beiden Knoten v_2' und v_2'' dargestellt. Dabei kann entweder der ursprüngliche Knoten v_2 durch einen Splitknoten s_2 ersetzt werden, oder der vorherige Knoten v_1 muss genügend Informationen über den Splitvorgang erhalten, um die Daten selbständig an v_2' und v_2'' weiterzuleiten.

Die einfachste Art der Aufteilung nimmt eine gleichmässige Aufteilung des Auswirkungsgrades vor, d.h. der Auswirkungsgrad wird halbiert $\eta_{split} = \frac{\eta}{2}$. In der Abbildung ist dies durch die beiden gleichen Werte 0.5 dargestellt. Eine Voraussetzung für das Aufteilen ist, dass der verwendete Algorithmus das Aufteilen erlaubt, hier ist meistens eine Baumstruktur besser geeignet, da bei einer azyklischen gerichteten Graphen ohne Mehrfachkanten eine Aggregationsfunktion am Knoten nach dem Aufteilen vorhanden sein muss. Ausserdem ist bei einem azyklischen gerichteten Graphen ohne Mehrfachkanten der Auswirkungsgrad für alle Layer bzw. Knoten gleich, solange nicht an mehreren Knoten eine Aufteilung stattfindet. Um bei einem azyklischen gerichteten Graphen ohne Mehrfachkanten dennoch eine signifikante Verbesserung durch die Validiereraufteilung zu erreichen, können die aufzuteilenden Knoten durch Expertenwissen bestimmt werden. Auf diese Weise können Knoten, die sich in einer weniger sicheren Umgebung befinden (z.B. leicht zugänglich für die mitarbeitende Person), identifiziert werden und ihre Auswirkungen aufgeteilt werden.

8.6.2. Betrachtung einer Baumstruktur

Bei der Betrachtung einer Baumstruktur ist der Schutzgrad ρ meist deutlich höher, wenn die Aufteilung in verschiedene Validierer oberhalb der Blattknoten erfolgt. Dabei werden auch die jeweiligen Kindknoten aufgeteilt, wodurch der Auswirkungsgrad der Kindknoten sinkt. In Abbildung 8.30 wurde v_2'' durch s_2 ersetzt. Der Knoten s_2 wird als Load Balancer verwendet, der die Aufgabe hat, die Eingangsdaten möglichst gleichmässig auf die nachfolgenden Knoten v_2'' zu verteilen. Die beiden grün markierten Knoten v_2'' sind absichtlich gleich benannt, um zu verdeutlichen, dass es sich um die gleiche Validierungslogik handelt. Es ist auch ersichtlich, dass die Kindknoten v_3'' und v_3'''' ebenfalls dupliziert wurden. Dies führt dazu, dass sie im Falle einer erfolgreichen Korrumpierung einen geringeren Einfluss haben.

Durch das Hinzufügen einer neuen Aufteilung erhöht sich die Anzahl der Knoten von $2^l - 1$ auf $2^{l+1} - 1$, wobei l die Anzahl der Layer ist. Bei einer Baumstruktur mit drei

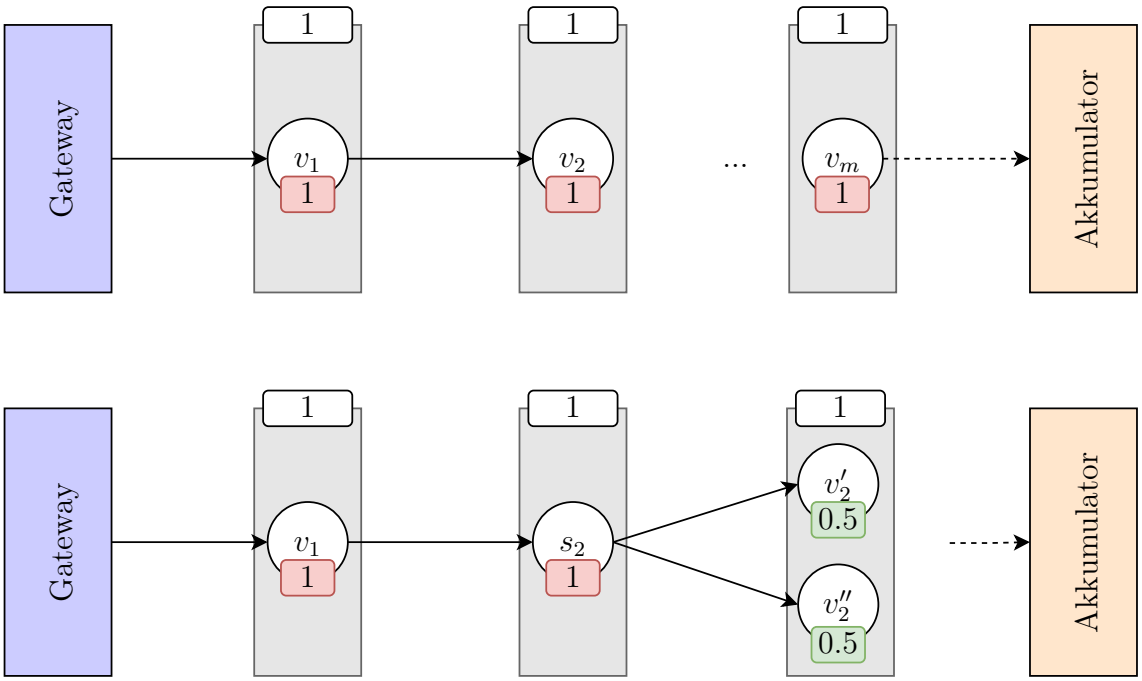


Abbildung 8.29.: Aufteilung eines Validierers mit grosser Auswirkung in einem azyklischen gerichteten Graphen ohne Mehrfachkanten

Layern und sieben Knoten erhöht sich die Anzahl der Knoten um vier, wenn nur der mittlere Knoten v_2'' geteilt wird. Unter der Annahme, dass die Baumstruktur immer gleich verteilt ist, müsste ein kompletter Layer hinzugefügt werden. Dies bedeutet, dass an einer bisher nicht berücksichtigten Stelle (z.B. v_3' und v_3'') eine zusätzliche Aufteilung vorgenommen werden muss.

Je grösser und ausgewogener ein Baumstruktur ist, desto weniger nützlich ist dieser Ansatz. Am besten funktioniert dieser Ansatz, um die Auswirkungen einiger weniger Validierer zu verteilen. Unter der Annahme, dass es in einer Baumstruktur mit drei Layern und insgesamt sieben Knoten einen korrumpierten Validierer gibt, ergibt sich ein mittlerer Auswirkungsgrad $\overline{\eta} = \frac{3}{7}$. Dieser errechnet sich aus dem Mittelwert der gesamten η:

$$\begin{aligned} \eta(v_1) &= \frac{4}{4} = 1 \\ \eta(v_2') = \eta(v_2'') &= \frac{2}{4} = \frac{1}{2} \\ \eta(v_3') = \eta(v_3'') &= \ldots = \frac{1}{4} \\ \overline{\eta} &= \frac{1 \cdot 1 + 2 \cdot \frac{1}{2} + 4 \cdot \frac{1}{4}}{7} = \frac{3}{7} \end{aligned} \tag{8.14}$$

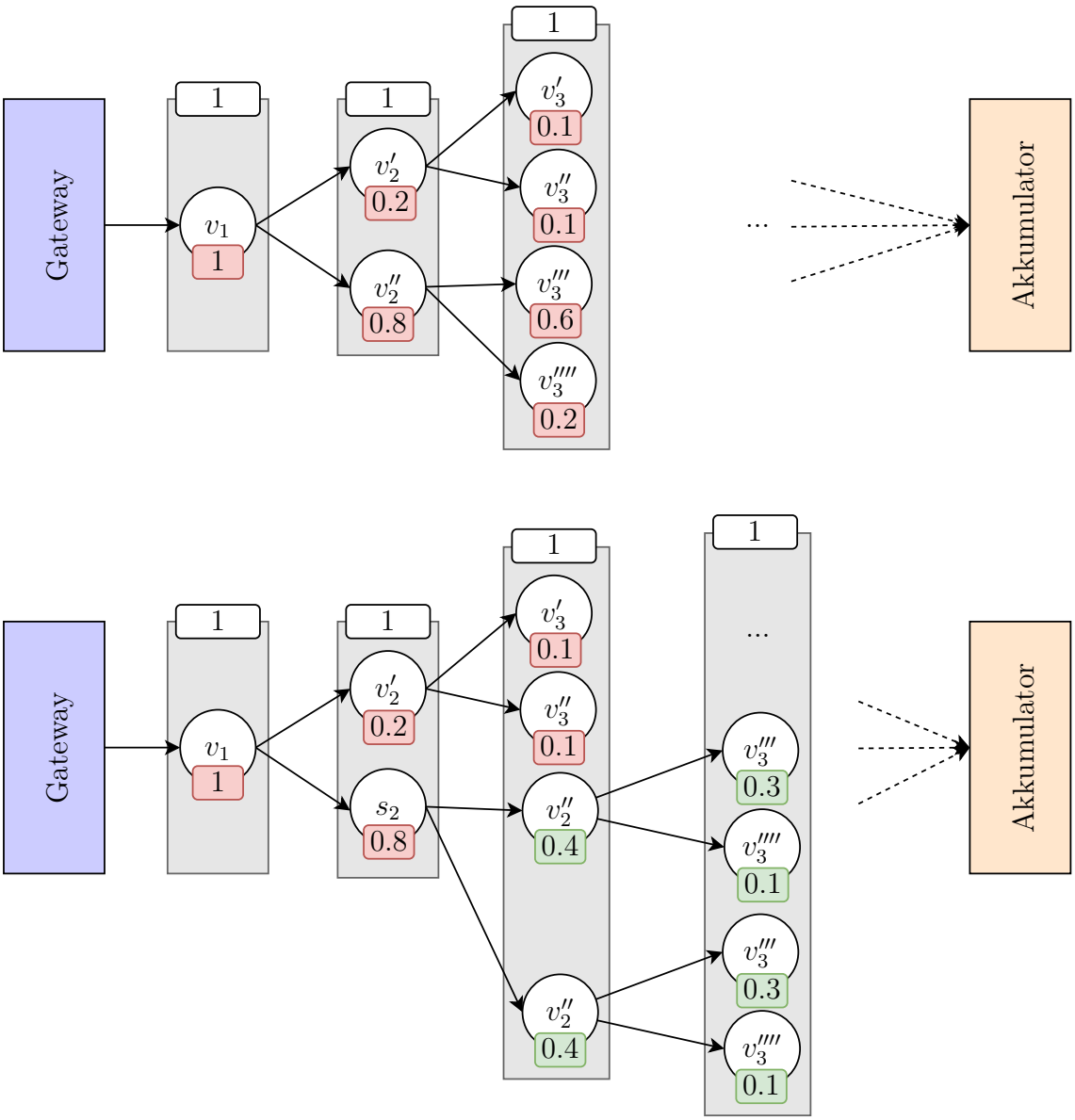


Abbildung 8.30.: Aufteilen eines Validierers mit einem $\eta = 0.8$ in einen neuen Unterbaum in einer Baumstruktur

Zusammenfassend kann der mittlere Auswirkungsgrad für eine Baumstruktur mit einem korrumpierten Knoten wie folgt berechnet werden:

$$\overline{\eta} = \frac{1}{2^{l-1}} \cdot l \tag{8.15}$$

Durch die Aufteilung von v_2'' in zwei Validierer mit dem Aufteilungsknoten s wird der Auswirkungsgrad des Angriffs auf $\frac{4}{15}$ reduziert.

Der Aufteilungsansatz unterscheidet sich vom Quorum Ansatz in drei wesentlichen Punkten: 1) es wird für beide Validierer die gleiche Logik verwendet, beim Quorum Ansatz können die Validierer unterschiedliche Algorithmen verwenden, 2) es wird nur in zwei Validierer aufgeteilt, beim Quorum werden mehr Validierer benötigt, um ein Quorum zu bilden, 3) beim Aufteilen eines Validierers in einer Baumstruktur werden alle notwendigen Unterknoten mit kopiert.

Theorem 8.6.1. *Der Auswirkungsgrad eines Validierers v_p, der ein Elternknoten für zwei Validierer v_{c1} und v_{c2} darstellt, hat einen grösseren oder mindestens gleichen Auswirkungsgrad η als die beiden Kinderknoten.*

Beweis. Ein Elternknoten v_p besitzt stets zwei Kinderknoten $\{v_{c1}, v_{c2}\}$. Unter der Annahme, dass $\eta(v_p) = \eta(v_{c1}) = \alpha \cdot \eta(v_{c2})$ $(\eta(v_p) > 0,\ \alpha \geq 0)$ und $\eta(v_p) = \eta(v_{c1}) + \eta(v_{c2}) = \eta(v_p) \cdot (1 + \alpha)$ folgt das $\eta(v_p) \geq \eta(v_c)$. □

Der Ansatz, Knoten mit einem hohen η aufzuteilen, kann verwendet werden, wenn die angreifende Instanz die Baumstruktur bzw. den Algorithmus nicht kennt.

8.7. Einfügen von Validierern ohne Auswirkung

Eine Möglichkeit, die Erfolgswahrscheinlichkeit γ eines Angriffs zu verringern, besteht darin, die eigentliche Datenstruktur um nicht verwendete Validierer, sogenannte Stubs, zu erweitern. Bei der Erstellung des Algorithmus sind die nicht verwendeten Knoten bereits bekannt. Sollte ein Validierer mit $\eta = 0$ korrumpiert werden, so hat dies keine negativen Auswirkungen auf den eigentlichen Algorithmus. Beim Hinzufügen von Stubs sind nur Algorithmen geeignet, die eine Verzweigung haben, da es nicht möglich ist, einen Validierer mit $\eta = 0$ in einem azyklischen gerichteten Graphen mit einem Anfangsgrad von eins zu positionieren. Der Schutzmechanismus: *Einfügen von Validierern ohne Auswirkung* hat die positive Eigenschaft, dass der ursprüngliche Algorithmus nicht angepasst werden muss.

8.7.1. Betrachtung eines Graphen

Durch das Hinzufügen eines Knotens, der keinen Einfluss auf den Gesamtablauf hat (siehe Abbildung 8.31), wird aus einem azyklischen Graphen mit dem Anfangsgrad 1 ein azyklischer Graph mit dem Anfangsgrad > 1.

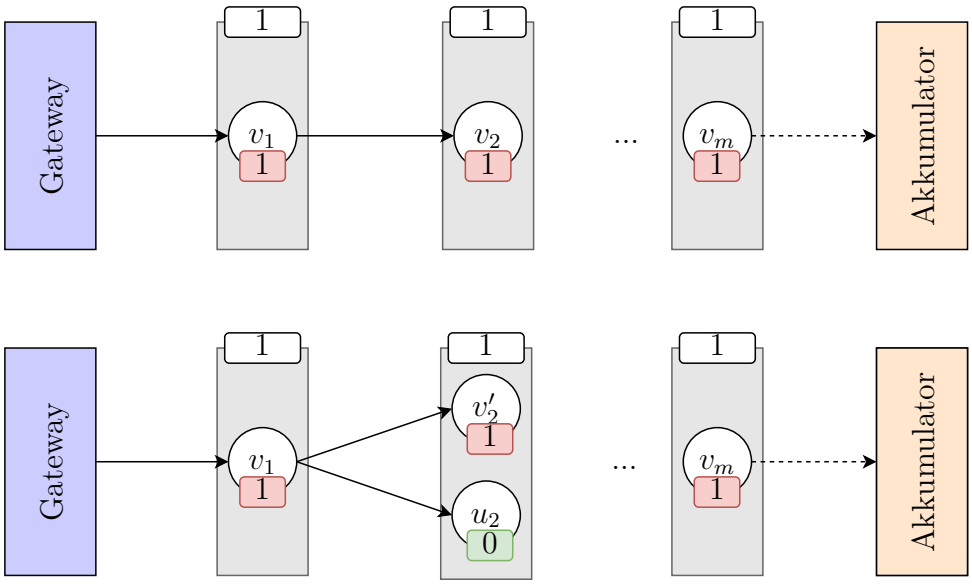


Abbildung 8.31.: Hinzufügen von Validierern mit keiner Auswirkung bei einem azyklischen gerichteten Graphen ohne Mehrfachkanten

8.7.2. Betrachtung einer Baumstruktur

In Abbildung 8.32 ist eine Baumstruktur gegeben. Hier ist zu erkennen, dass ein ganzer Teilbaum den eigentlichen Algorithmus vor Korruption schützt. Die Wahrscheinlichkeit, dass ein Angriff einen Stub korrumpiert, hängt also von der Anzahl der Stub-Knoten ab $p_{stub} = \frac{n_{stub}}{n}$, wobei n_{stub} die Anzahl der vorkommenden Stubs und n die Gesamtanzahl der Knoten des Algorithmus ist.

Dieser Schutzansatz ist gut geeignet, wenn bei einem Angriff keine Kenntnisse über die Struktur des Algorithmus und des Stubs bekannt sind. Der Nachteil ist, dass ein Stub nicht für den eigentlichen Validierungsprozess verwendet werden kann.

8.8. Einteilung von Validierern in Cluster

Die Validierer eines DDVNs haben bestimmte Eigenschaften bzw. Merkmale. In Kapitel 5 werden diese Eigenschaften in drei logische Kategorien *a*) Validierungseigenschaften, *b*) physikalische Eigenschaften und *c*) plattformspezifische Eigenschaften eingeteilt. Eine Auflistung verschiedener Einteilungsmöglichkeiten ist in Tabelle 8.14 gegeben. Ein Validierer kann in beliebig vielen verschiedenen Clustern enthalten sein. Bei bestimmten physikalischen oder plattformspezifischen Merkmalen schliessen sich die Cluster jedoch logisch aus. So kann z.B. ein Validierer mit dem Standort Deutschland nicht in den Standortclustern für Österreich und die Schweiz enthalten sein.

Durch die Einteilung der Validierer in logische Cluster wird der Akkumulator bei

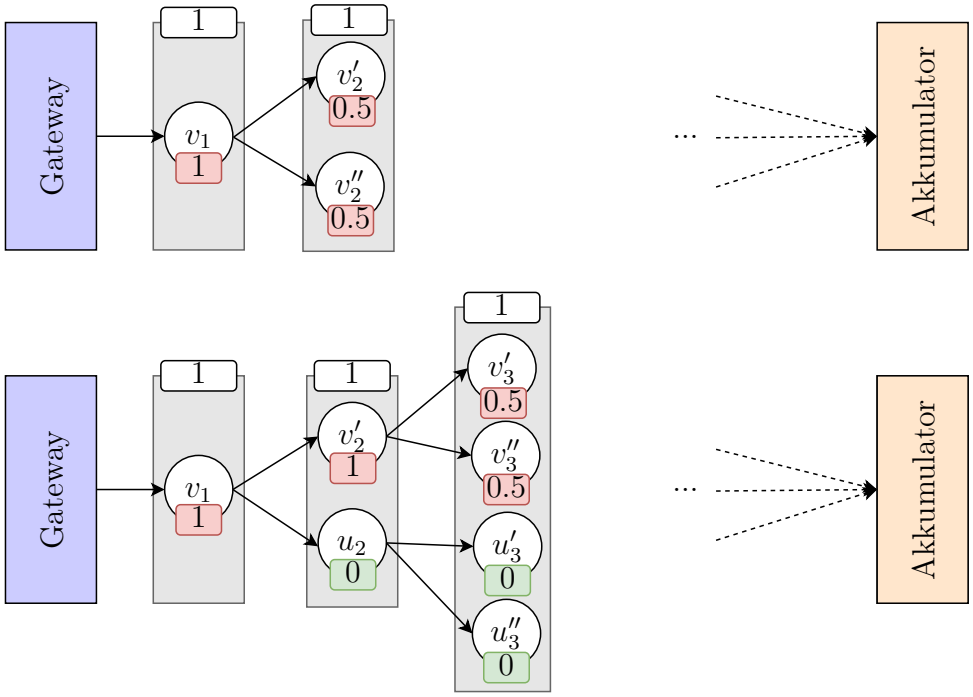


Abbildung 8.32.: Hinzufügen von Validierern mit keiner Auswirkung bei einer Baumstruktur

der Auswahl der Validierer unterstützt. Der Akkumulator kann anhand bestimmter Merkmale die für ihn geeigneten Validierer auswählen, so dass ein erhöhter Schutz im DDVN gegeben ist. Die Verwendung von Validierungseigenschaften wird üblicherweise nicht zur Erhöhung der Sicherheit verwendet, da diese bereits implizit durch die Validierungsanforderung gegeben ist. Wenn die Validierungsanforderung der Temperaturkontrolle entspricht, muss ein Validierer aus dem entsprechenden Cluster genommen werden.

Sei M die Menge aller Clustermerkmale $M = \{m_1, m_2, ...\}$ ohne die Validierungseigenschaften. Daraus ergeben sich mögliche Mengen von Gruppierungsmerkmalen $V \subseteq M$. Für eine gute Validiererauswahl seitens des Akkumulators ist zu beachten, dass $V_a \cap V_b = \{\}$. Somit haben beide Validierer unterschiedliche Eigenschaften und können gut in einer gemeinsamen Validiererauswahl auftreten. Validierer, für die $V = V_a = V_b$ gilt, sind dagegen nur bedingt geeignet. Bei einem erfolgreichen Angriff auf einen Validierer v mit den Eigenschaften V kann der Angriff meist auch auf alle anderen Validierer mit den Eigenschaften V angewendet werden.

Im Folgenden ist ein Beispiel für die Auswahl von Validierern aus verschiedenen Clustern beschrieben. Bei der Validierung einer definierten Maschine ist darauf zu achten, dass mindestens ein Validierer das Betriebssystem Windows und mindestens ein Validierer das Betriebssystem Linux hat. So kann auch bei einem Zero-Day-Exploit verhindert werden, dass alle Validierer gleichzeitig korrumpiert werden. Je

mehr unterschiedliche Cluster bei der Auswahl der Validierer berücksichtigt werden, desto mehr unterschiedliche Eigenschaften haben die Validierer und desto schwieriger ist es, sie mit dem gleichen Angriff zu korrumpieren.

Tabelle 8.14.: Mögliche Merkmale für das logische Einteilen von Validierern in Cluster

Cluster Name	Beschreibung
	Validierungsmerkmale
Temperatur	Der Validierer besitzt die Fähigkeit die Temperatur zu validieren. Abhängig vom jeweiligen DDVN Aufbau, muss hier noch die Maschine oder das Objekt, für das die Validierungsfähigkeit vorhanden ist, angegeben werden.
Feuchtigkeit	Der Validierer besitzt die Fähigkeit die Feuchtigkeit zu validieren. Abhängig vom jeweiligen DDVN Aufbau, muss hier noch die Maschine oder das Objekt, für das die Validierungsfähigkeit vorhanden ist, angegeben werden.
Gesundheit	Der Validierer besitzt die Fähigkeit die Gesundheit einer definierten Maschine über die Korrelation mehrere Daten festzustellen. Mögliche Daten hierfür sind zum Beispiel der Stromverbrauch und die Anzahl der produzierten Teile in einer definierten Zeit.
	Physische Merkmale
Standort	Der Validierer ist an einem bestimmten Standort (Land, Produktionshalle, etc.) installiert und ist deswegen ein grösseres oder geringeres Risiko bei der Verwendung für einen Validierungsvorgang. Gerade im Zusammenhang mit Industriespionage oder -sabotage ist dies von zentraler Bedeutung.
Gerät	Der Validierer ist auf einem bestimmten Gerät (Tablet, Notebook, PC, Maschine, etc.) installiert. Ähnlich, wie beim Standort Merkmal, können verschiedene Geräte unterschiedlich in Bezug auf Sicherheit eingestuft werden.

Zeitzone	Der Validierer besitzt eine bestimmte Zeitzone, diese könnte auch über den Standort ermittelt werden. Die Verwendung der Zeitzone bei der Validiererauswahl kann dazu verwendet werden, Leistungsspitzen in bestimmten Regionen zu verhindern.
	Plattform Merkmale
Algorithmus	Der Validierer verwendet für die Validierung eines bestimmten Datensatzes einen bestimmten Algorithmus.
Betriebssystem	Der Validierer ist auf einem definierten Betriebssystem installiert.
Container-virtualisierung	Der Validierer verwendet eine bestimmte Containervirtualisierungslösung, wie Docker oder LXC.
Netzwerkbereich	Der Validierer ist Teil eines bestimmten Netzwerkbereichs des Firmennetzwerks und deswegen mehr oder weniger vertrauenswürdig.

8.9. Aggregationsentscheidung mittels Cluster Ansatz und Quorum Ansatz

Abhängig von der verwendeten Aggregationsfunktion des Akkumulators kann der Angriff auf unterschiedliche Arten erfolgreich sein. Die einfachste Art der Aggregation besteht darin, alle gleichen Validierungsergebnisse zusammenzufassen und ihre Anzahl zu bestimmen. Wenn ein bestimmter Grenzwert t mit $\{t \in \mathbb{R} \mid 0 \leq t \leq 1\}$, von einer Gruppe überschritten, so gilt das Ergebnis dieser Gruppe. Dies entspricht der Vorgehensweise bei einem Quorum Ansatz für Validierer. Unter der Annahme $t = \frac{1}{2}$ wird ein Ergebnis akzeptiert, sobald $\lfloor t \cdot m \rfloor + 1 = \lfloor \frac{m}{2} \rfloor + 1$, wobei $m = |M|$ die Anzahl aller für eine Aggregation ausgewählten Validierer ist. Dies ergibt die erste mögliche Aggregationsfunktion, die als *t-Total Quorum* bezeichnet wird. Im obigen Beispiel entspricht dies einem $\frac{1}{2}$-*Total Quorum*, da mehr als 50% der Validierer der gleichen Meinung sein müssen, um den Akkumulator zu überzeugen. Je höher t ist, desto geringer ist die Wahrscheinlichkeit eines erfolgreichen Angriffs γ. Wenn jedoch die Aggregationsfunktion zu oft ein ungültiges oder unzuverlässiges Ergebnis liefert, wird es mehr falsch positive Ergebnisse geben.

Eine weitere Aggregationsfunktion wird als (s, t)-*Nested Quorum* bezeichnet. Hierbei wird der *t-Total Quorum* Ansatz pro Cluster durchgeführt und sobald eine

bestimmte Anzahl von Clustern k die gleiche Meinung hat, wird das Aggregationsergebnis ermittelt. s entspricht dem Verhältnis der zu überzeugenden Cluster k zur Gesamtanzahl der Cluster ℓ. Auf diese Weise bezieht der Akkumulator die Cluster in die Aggregation ein. Bei einem $(\frac{1}{2}, \frac{3}{4})$-*Nested Quorum* ist ein gültiges Aggregationsergebnis erreicht, sobald mehr als 50% der Cluster der gleichen Meinung sind und in jedem dieser Cluster mehr als 75% der Validierer in ihrer Entscheidung übereinstimmen.

Auch wenn die Bezeichnungen *t-Total Quorum* und (s,t)-*Nested Quorum* auf eine bestimmte Netzstruktur hinweisen, ist diese für die Aggregationsfunktion ohne Bedeutung.

8.9.1. Wahrscheinlichkeit eines erfolgreichen Angriffs auf ein *t-Total Quorum*

In der folgenden Analyse wird ein hoher Anteil an fehlerhaften Validierern $> 50\%$ angenommen. Sobald der Threshold überschritten wird, kann die weitere Aggregation gestoppt werden, um Ressourcen zu sparen.

Theorem 8.9.1. *Die Wahrscheinlichkeit eines erfolgreichen Angriffs bei einem t-Total Quorum Ansatz ist unabhängig von der Anzahl der verwendeten Cluster ℓ.*

Beweis. Die Wahrscheinlichkeit p_t einen Akkumulator mit mindestens c korrumpierten Validierern unter Verwendung eines *t-Total Quorum* Ansatzes zu überzeugen, wird wie folgt berechnet:

$$p_t = \sum_{i=0}^{m-\lfloor t\cdot m+1\rfloor} \frac{\binom{c}{m-i}\binom{n-c}{i}}{\binom{n}{m}}, \text{ wenn } m \leq c. \tag{8.16}$$

Wie aus Gleichung 8.16 hervorgeht, hat die Anzahl der Cluster bei einem *t-Total Quorum* Ansatz keinen Einfluss. m ist die Anzahl der für die Aggregation berücksichtigten Validierer und n ist die Anzahl aller Validierer im gesamten DDVN. □

Tabelle 8.15.: Angriffserfolgswahrscheinlichkeit auf ein *t-Total Quorum*

Knoten m	**1 Cluster**	**2 Cluster**	**16 Cluster**	p_t
64	0.4430	0.4431	0.4432	0.4427
32	0.4257	0.4255	0.4263	0.4252

Tabelle 8.15.: Angriffserfolgswahrscheinlichkeit auf ein *t-Total Quorum*

Knoten m	**1 Cluster**	**2 Cluster**	**16 Cluster**	p_t
16	0.3984	0.3983	0.4001	0.3986

Tabelle 8.15 zeigt das Ergebnis einer Simulation, bei der 50% der Validierer korrumpiert sind. Die Anzahl der ausgewählten Validierer entspricht $m = \ell \cdot v$, wobei ℓ die Anzahl der Cluster und v die Anzahl der Validierer pro Cluster ist. Die letzte Spalte in der Tabelle zeigt das Ergebnis der Berechnung p_t mittels Gleichung 8.16.

Theorem 8.9.2. *Die Wahrscheinlichkeit eines erfolgreichen Angriffs bei einem* $\frac{1}{2}$*-Total Quorum hat ein Minimum bei* $m = 2$ *für jede mögliche Anzahl korrumpierter Knoten.*

Beweis. Gleichung 8.16 hat ein lokales Minimum an der Stelle $m = 2$ für alle Werte von c. □

In Abbildung 8.33 ist die Erfolgswahrscheinlichkeit eines Angriffs für vier verschiedene Mengen korrupter Knoten bei einem $\frac{1}{2}$*-Total Quorum* dargestellt. Jedes untersuchte DDVN besteht aus einem Cluster $\ell = 1$ und einer Gesamtanzahl von $n = 256$ Knoten. Es ist ersichtlich, dass bei $m = 1$ ein lokales Maximum existiert. Ausserdem ist bei $m = 2$ ein lokales Minimum für eine korrupte Knotenanzahl $c \geq 0.3 \cdot n$ zu erkennen. Dies kann dadurch erklärt werden, dass bei $m = 2$ alle Validierer korrumpiert werden müssen, um das Ergebnis zu bestimmen. Ein Teil der Simulation ist in Unterabschnitt A.4.4 aufgezeigt.

Für $m = 1$ ist bei einem Verhältnis von $c = \frac{1}{2} \cdot n$ die Wahrscheinlichkeit, einen korrupten Knoten auszuwählen, $p_c = \frac{1}{2}$. Wenn $m = 2$ ist, ist die Wahrscheinlichkeit, zwei korrupte Knoten auszuwählen, geringer, da $p_{c1} = \frac{1}{2}$, $p_{c2} < \frac{1}{2}$ und $p_c = p_{c_1} p_{c_2} < \frac{1}{4}$, wobei p_{ci} der Wahrscheinlichkeit entspricht, dass das $i$te Ergebnis korrupt ist.

Für $m > 2$ müssen nicht alle Validierer korrumpiert werden, weshalb die Angriffserfolgswahrscheinlichkeit zunächst ansteigt. Abhängig vom Korruptionsgrad κ, wird ein unterschiedliches Maximum erreicht bevor die Angriffserfolgswahrscheinlichkeit wieder abnimmt. So kann ein globales Minimum je nach κ unter $m = 2$ liegen.

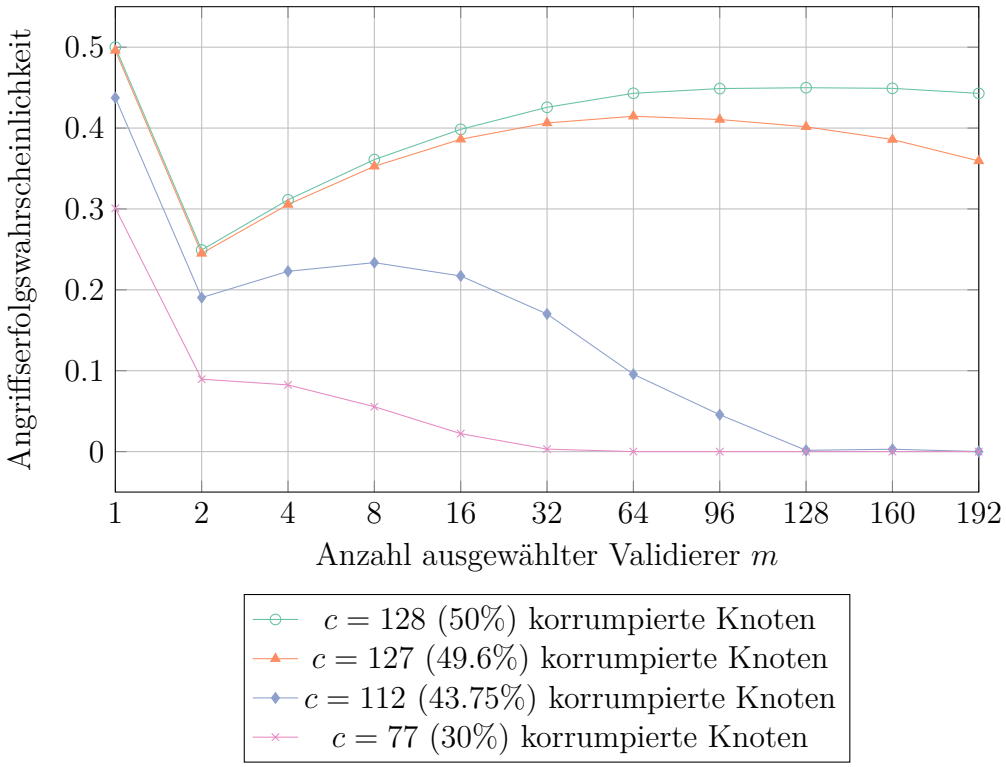


Abbildung 8.33.: Angriffserfolgswahrscheinlichkeit für eine unterschiedliche Anzahl an korrumpierten Knoten bei einem $\frac{1}{2}$-*Total Quorum Ansatz*. Jedes DDVN besteht aus $n = 256$ Knoten. Ein lokales Minimum ist bei $m = 2$ für eine korrumpierte Knotenanzahl $c \geq 0.3 \cdot n$.

8.9.2. Wahrscheinlichkeit eines erfolgreichen Angriffs auf ein (s, t)-*Nested Quorum*

Ein (s, t)-Nested Quorum besteht aus mehreren (kleineren) Quoren, daher ist die Wahrscheinlichkeit das Produkt der Wahrscheinlichkeiten aller Quoren. Daraus folgt, dass die Gesamtwahrscheinlichkeit umso geringer ist, je mehr Cluster beteiligt sind.

In Abbildung 8.34 wird die Auswirkung unterschiedlicher Korruptionsgrade κ auf ein $(\frac{1}{2}, \frac{1}{2})$-*Nested Quorum* gezeigt. Es werden 32 Cluster mit jeweils acht Validierern angenommen. Die Anzahl der ausgewählten Validierer m pro Cluster wird bei konstantem κ variiert. Die korrumpierten Validierer werden gleichmässig auf alle Cluster verteilt. Wie bei einem $\frac{1}{2}$-*Total Quorum* liegt ein lokales Minimum an der Stelle $m = 2$. Es ist ersichtlich, dass bereits 0.4% weniger korrupte Knoten einen signifikanten Effekt (teilweise bis zu 100%) auf die Angriffserfolgswahrscheinlichkeit haben. Weiterhin wird der Vorteil einer geraden Auswahl von Knoten in einem Cluster aufgezeigt (siehe $m \in \{2, 4, 6, 8\}$). Bei einem Vergleich mit Abbildung 8.33 muss die Anzahl der ausgewählten Validierer mit 32

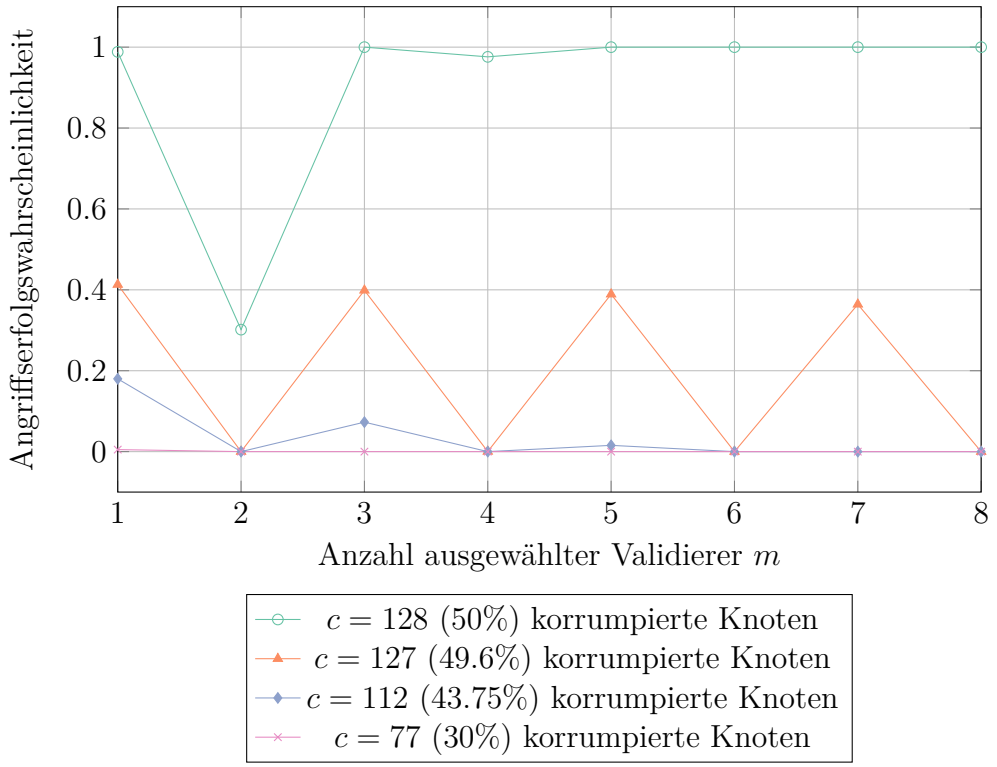


Abbildung 8.34.: Angriffserfolgswahrscheinlichkeit für eine unterschiedliche Anzahl an korrumpierten Knoten bei einem $(\frac{1}{2}, \frac{1}{2})$-*Nested Quorum Ansatz*. Jedes DDVN besteht aus $n = 256$ Knoten. Ein lokales Minimum ist bei $m = 2$ für eine korrumpierte Knotenanzahl $c \geq 0.3 \cdot n$.

multipliziert werden, da dies der Anzahl der verwendeten Cluster entspricht.

8.9.3. Auswirkung unterschiedlicher (s, t) Werte bei einem (s, t)-*Nested Quorum*

In Abbildung 8.35 wird die Auswirkung verschiedener (s, t) Thresholds simuliert. Durch Erhöhung der Thresholds kann die Wahrscheinlichkeit eines erfolgreichen Angriffs drastisch reduziert werden. Bei (s, t)-Werten von $(\frac{1}{4}, \frac{1}{4})$ liegt die Angriffserfolgswahrscheinlichkeit jeweils über 0.9. Wird der (s, t) Wert auf $(\frac{3}{4}, \frac{3}{4})$ gesetzt, so ist die Angriffserfolgswahrscheinlichkeit kleiner als 0.01.

s-Werte unter $\frac{1}{2}$ führen dazu, dass ein Angriff eine viel grössere Chance hat, einen Akkumulator zu überzeugen, wenn t nicht erhöht wird. Dies kann dadurch erklärt werden, dass eine Verringerung von s effektiv zu einer Verringerung der Anzahl der für die Entscheidungsfindung verwendeten Cluster führt. Mehr Cluster verringern die Wahrscheinlichkeit eines erfolgreichen Angriffs (siehe Theorem 8.9.2). Der Effekt ist stärker, wenn der t-Wert von $\frac{1}{2}$ auf $\frac{1}{4}$ reduziert wird, ohne gleichzeitig s zu erhöhen.

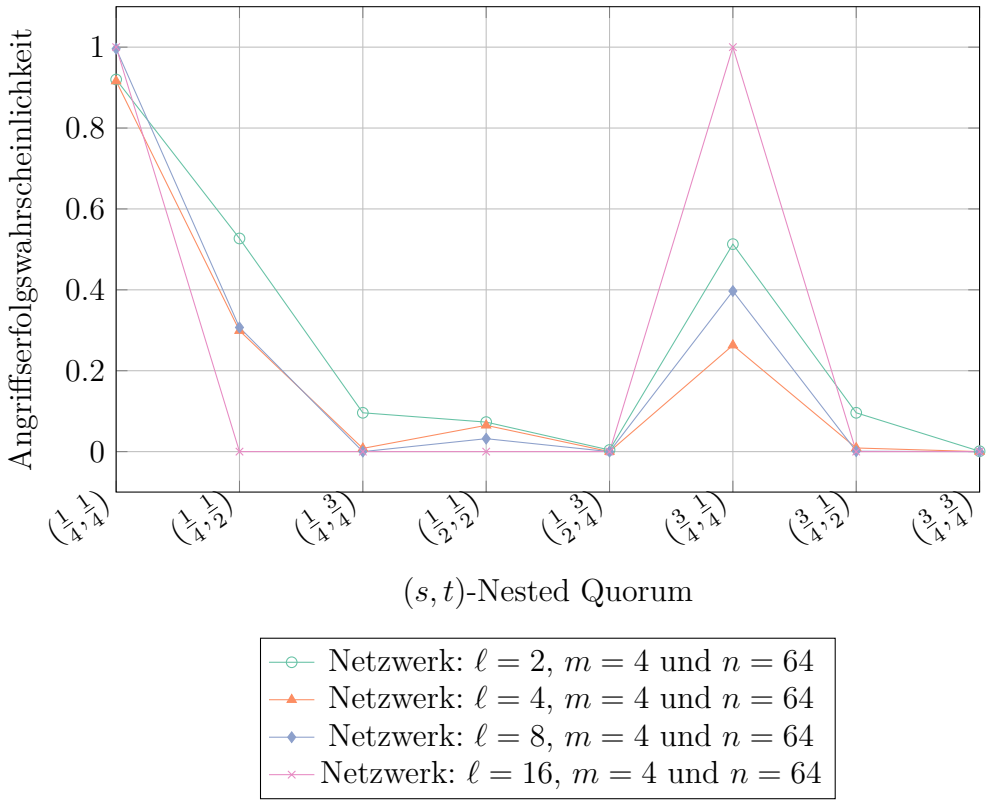


Abbildung 8.35.: Angriffserfolgswahrscheinlichkeit für *Nested-Quoren* mit einer unterschiedlichen Anzahl an Clustern $\ell \in \{2, 4, 8, 16\}$ und unterschiedlichen (s, t) Werten. Der Korruptionsgrad κ ist bei 50% und gleichverteilt zwischen den Clustern

Das Kapitel befasst sich mit den spezifischen Schutzmechanismen für das DDVN. Als erstes wird hierfür die Verwendung einer PMI für die Ausstellung von PKCs und ACs eingeführt. Damit wird eine Trennung zwischen Identität und Berechtigung ermöglicht, sodass ein Validierer seine Berechtigung bzw. Fähigkeit ohne eine Wechsel der Identität ausführen kann. Ein AC entspricht hierbei der Validierungsfähigkeit die einem bestimmten PKC zugewiesen wird.
Ein weiterer Schutzmechanismus beschreibt das Anreichern der eigentlichen Daten zur Validierung mittels Kontextinformation. Daraus resultierend sollte die Genauigkeit einer Validierung und daraus resultierend die Genauigkeit der Bestimmung der Datenqualität erhöht werden. Bei einem anonymisierten Datensatz der Firma TRUMPF GmbH & Co. KG konnte aufgezeigt werden, dass durch die Erhöhung der verwenden Merkmale für eine Vorhersage die Genauigkeit um mehr als 30% steigen konnte (siehe DT in Abbildung 8.5).
Um die Auswirkung von unterschiedlichen Schutzmechanismen auf die beiden Algorithmenstrukturen Decision Tree und azyklischer gerichteter Graph ohne Mehrfachkanten (GRAPH) zu bestimmen wurden eigene Metriken eingeführt. Dazu zählen *a*) der Korruptionsgrad κ, der dem Verhältnis zwischen den korrupten

Knoten und den gesamten Knoten entspricht, b) der Auswirkungsgrad η, der angibt wie stark bestimmte korrumpierte Knoten die Vorhersage beeinflussen können und c) die Angriffserfolgswahrscheinlichkeit γ, die angibt wir erfolgreich ein Angriff ist. Für die gegebenen Algorithmenstrukturen werden drei unterschiedliche Ansätze zur Erhöhung der Sicherheit untersucht. Beim ersten Ansatz werden einzelne Knoten durch ein Quorum ersetzt. Hierbei sinkt γ bei fünf korrumpierten Knoten $c = 5$ in einem GRAPH mit einer Layeranzahl $l = 10$ und einer Quorumsgrösse $|Q| = 3$ von anfänglich 100% auf unter 60%, wenn jeder Knoten durch ein Quorum ersetzt wird. Durch die Erhöhung von $|Q| = 7$ sinkt γ unter 1%. Zudem wird mittels Simulation gezeigt, dass die Erhöhung der Quorumsgrösse gegenüber der Erhöhung der Layeranzahl in Bezug auf Sicherheit zu bevorzugen ist. Weitere Ansätze, wie das Aufteilen von Validierern mit einer grosser Auswirkung sowie das Einfügen von Validierern ohne Auswirkung werden ebenfalls beschrieben.
Zero-Day Exploits sind eine Gefahr für viele Systeme. Damit die Auswirkung eines solchen Angriffs für das DDVN minimiert werden, können Validierer in Cluster eingeteilt werden. Ein Cluster entspricht einer logischen Gruppierung basierend auf Validiereigenschaften (z.B. das Betriebssystem). Bei der Auswahl der verwendeten Validierer für einen Validierungsvorgang sollte der Akkumulator möglichst Validierer wählen, die nicht in selben Clustern enthalten sind. Dadurch wird sichergestellt, dass die Validierer unterschiedliche Eigenschaften besitzen und bei einem Zero-Day Exploit für eine bestimmte Eigenschaft, nur ein Teil der Validierer eines Validierungsvorgangs betroffen sind. Somit erkennt der Akkumulator, dass die Validierer unterschiedliche Ergebnisse liefern und eine Anomalie vorliegt.
Der letzte Ansatz zur Erhöhung der Sicherheit beschreibt die zwei unterschiedliche Funktionen, um eine Aggregierung im Akkumulator vorzunehmen. Bei einem t-Total Quorum muss eine bestimmte Anzahl an Validierungsergebnissen das gleiche Ergebnis liefern, damit eine Entscheidung getroffen werden kann. Eine Erweiterung hiervon ist das (s, t)-Nested Quorum. Hierbei muss in einer bestimmten Menge an Clustern ein bestimmter Schwellwert ein gleichen Ergebnissen überschritten werden, um ein erfolgreiche Aggregierung der Validierungsergebnis vorzunehmen.

9. Zusammenfassung und Ausblick

Das folgende Kapitel fasst die vorangegangenen Abschnitte zusammen und gibt einen Ausblick auf offene Forschungsfragen im Umfeld des DDVNs.

9.1. Zusammenfassung

Der erste Abschnitt dieser Arbeit gibt eine Einführung in das Thema und zeigt die Motivation auf, in diesem Bereich zu forschen. Darüber hinaus wird aufgezeigt, warum Datensicherheit bzw. Datenqualität für Vorhersagen im industriellen Umfeld unerlässlich ist.

Im zweiten Abschnitt werden verwandte Arbeiten in den Bereichen *a*) Datenvalidierung, *b*) Schutzmechanismen für die Datenvalidierung, *c*) verteilte Datenvalidierung und *d*) Schutzmechanismen in verteilten Systemen vorgestellt. Es zeigt sich, dass ein grosser Teil der Forschung im Bereich der Datenvalidierung die Sicherheit ausser Acht lässt oder als gegeben voraussetzt.
In der Forschung zur Sicherheit der Datenvalidierung werden zwei Ansätze unterschieden. Der interne Ansatz, bei dem versucht wird, den verteilten Algorithmus selbst zu schützen, und der externe Ansatz, bei dem der Algorithmus selbst durch Erweiterungen von aussen geschützt wird. Für den internen Ansatz spricht, dass der Algorithmus nur schwer zu korrumpieren ist und somit eine In-Depth-Sicherheit gewährleistet ist, d.h. auch wenn ein Angriff einen Teil des Systems korrumpiert, funktioniert der Algorithmus weiterhin. Allerdings sind diese Schutzmechanismen meist algorithmusspezifisch und können, wenn überhaupt, nur mit grossem Aufwand auf einen anderen Algorithmus übertragen werden. Demgegenüber steht ein externer Ansatz, bei dem mittels Netzwerkschutzmechanismen, Hardwareschutzmechanismen etc. ein allgemein anwendbarer Schutzansatz aufgezeigt wird. Der Nachteil hierbei ist, dass allgemeine Angriffe auch den externen Schutzansatz stark beeinflussen können.
Das DDVN gehört zur Kategorie der externen Schutzansätze und kann daher auf alle in dieser Arbeit definierten Algorithmen (azyklischen gerichteten Graphen ohne Mehrfachkanten und Baumstruktur) angewendet werden. Eine Kombination mit einem internen Schutzansatz ist jedoch nicht ausgeschlossen und kann die Gesamtsicherheit des Systems erhöhen.

Der dritte Abschnitt umfasst eine historische Beschreibung der industriellen Revolution von der ersten bis zur vierten industriellen Revolution (Industrie 4.0).

Die historische Betrachtung dient als Ausgangspunkt für die Beschreibung der industriellen Cybersicherheit. Es wird aufgezeigt, warum die Personensicherheit derzeit weiter fortgeschritten ist als die Cybersicherheit. Ein Grund dafür ist, dass der Einsatz von vernetzten Komponenten und Vorhersagen auf Basis von Sensordaten im industriellen Umfeld erst seit wenigen Jahren als De-facto-Standard gilt.
Darüber hinaus enthält das Kapitel eine Zusammenfassung der aktuellen Bedrohungslage im Bereich der Cybersicherheit. Dabei wird deutlich, dass mit zunehmender Digitalisierung im industriellen Umfeld auch die Anzahl der Cyberangriffe steigt und somit mehr finanzielle Mittel in die Cybersicherheit investiert werden müssen.

Eine detaillierte Beschreibung der Datenqualität anhand der vier Kategorien *a*) intrinsische Metriken, *b*) kontextuelle Metriken, *c*) Metriken zur Repräsentativität und *d*) Metriken zur Zugänglichkeit findet sich in Kapitel 4. Neben der Definition der Datenqualität wird der Prozess der Datenvorverarbeitung [103, 104], bestehend aus sieben Teilschritten, beschrieben und aufgezeigt, an welchen Prozessschritten die Datenvalidierung eingesetzt werden kann, um eine Validierung durchzuführen.

Im fünften Abschnitt wird das selbst entwickelte DDVN anhand einer abstrakten Beschreibung der verschiedenen Strukturen von Datenvalidierungsnetzwerken vorgestellt. Diese sind *a*) ein System ohne Datenvalidierung, *b*) ein System mit interner Datenvalidierung, *c*) ein System mit externer Datenvalidierung und *d*) ein System mit externer verteilter Datenvalidierung (kurz DDVN). Die verschiedenen Systeme werden beschrieben und mit ihren Vor- und Nachteilen in bestimmten Bereichen wie Performance, Implementierung, Datenqualität etc. verglichen. Darüber hinaus werden die einzelnen Komponenten des DDVN im Detail erläutert.

Der nächste Abschnitt behandelt die Kommunikation im DDVN. Zunächst werden alle acht Arten von Umgebungen vorgestellt: eine vertrauenswürdige Umgebung, sechs halb-vertrauenswürdige Umgebungen und eine nicht vertrauenswürdige Umgebung. Anschliessend wird erläutert, welche Art von Umgebung (nicht vertrauenswürdige Quelle, vertrauenswürdiges Gateway und nicht vertrauenswürdiger Validator) in dieser Dissertation untersucht wird.
Der Aufbau der Kommunikation über den Broadcast des Akkumulators sowie die verwendeten Response Policy Dateien werden ebenfalls beschrieben. Das Kapitel umfasst auch den Validierungsprozess, bei dem die Sources die Sensordaten über das Gateway an den Akkumulator übermitteln. Das Gateway kann anhand der Subpolicy Dateien entscheiden, welcher Validator die Daten zur Validierung erhält. Schliesslich aggregiert der Akkumulator die Ergebnisse und entscheidet anhand des aggregierten Ergebnisses, wie die erhaltenen Daten zu klassifizieren sind. Auf diese Weise wird verhindert, dass korrupte Validierer die Auswahl der Validierer beeinflussen können.
Darüber hinaus enthält das Kapitel eine detaillierte Beschreibung der verschiedenen Kommunikationsmöglichkeiten zwischen Gateway, Validierer und Akkumulator. Die fünf wichtigsten Arten sind: 1) parallele Validierung mit

direkter Übertragung an den Akkumulator, 2) sequentielle Validierung auf der Grundlage vorhergehender Validierungsergebnisse, 3) parallele Validierung mit Übertragung der Validierungsergebnisse an nachfolgende Validierer, 4) Übertragung von Zwischenergebnissen der Validierung an den Akkumulator und 5) Kommunikation zwischen Validierern über TTP.

Im siebten Abschnitt werden zwei Sicherheitsanalysen auf das DDVN angewendet. Dabei handelt es sich um eine STRIDE-Analyse [135] und eine Analyse, die auf einer Reihe bekannter Angriffe basiert. Die STRIDE-Analyse basiert auf sechs bekannten Bedrohungen (Verschleierung, Verfälschung, Abstreitbarkeit, Enthüllung von Geheimnissen, Dienstverweigerung und Privilegienerweiterung), denen die jeweiligen Sicherheitsziele gegenübergestellt werden, z.B. Enthüllung von Geheimnissen entspricht Vertraulichkeit. Das System wird jeweils im Hinblick auf die Bedrohung betrachtet und mögliche Schutzmechanismen gegen die Bedrohung aufgezeigt. Insgesamt werden 21 Bedrohungen und 26 Schutzmechanismen dargestellt und erläutert.
Neben den Sicherheitsanalysen werden drei weitere wichtige Punkte erläutert, diese sind *a*) die Ursachen für das Fehlverhalten der Validierer, wobei nicht jedes Fehlverhalten auf einen Angriff zurückzuführen ist, *b*) die Angriffsbereiche auf Validierer, diese zeigen auf, welche Kombinationen von Knoten korrumpiert werden können und *c*) die Auswirkungen von Wissen bei einem Angriff.

Im letzten Abschnitt werden fünf eigene Ansätze zur Erhöhung der Sicherheit im DDVN auf der Grundlage der zuvor durchgeführten Sicherheitsanalyse erläutert.
Der erste Ansatz beschreibt die Trennung von Identität und Validierungsfähigkeit, wodurch Validierer ihre Validierungsfähigkeiten ändern können, während sie sich im Netzwerk befinden. Zusätzlich werden PKC und AC verwendet, um die Authentifizierung und Autorisierung der einzelnen Validierer sicherzustellen.
Ein weiterer Ansatz beschreibt die Anreicherung der Validierungsdaten mit Kontextinformationen. Dadurch verbessern sich die Vorhersageergebnisse für einen realen Anwendungsfall hinsichtlich Selektivität (0.99%) und Genauigkeit (0.87%). Allerdings sinkt die Empfindlichkeit um 2.73%. Die Entscheidung, welche Metrik zu optimieren ist, muss daher immer für den konkreten Anwendungsfall getroffen werden.
Der dritte Ansatz zeigt drei verschiedene Möglichkeiten auf, den Schutz des für die Validierung verwendeten Algorithmus zu erhöhen. Dazu werden einige neue Metriken eingeführt, wie der Korruptionsgrad κ, der Schutzgrad ρ und die Angriffserfolgswahrscheinlichkeit γ. Es wird gezeigt, dass bei Verwendung eines einzelnen Validierers die Angriffserfolgswahrscheinlichkeit 1 beträgt, sobald der Validierer korrumpiert ist. Durch den Einsatz mehrerer verteilter Validierer kann die Angriffserfolgswahrscheinlichkeit drastisch reduziert werden. Die positive Auswirkung auf das Schutzniveau hängt von mehreren Faktoren ab, wie z.B. der gewählten Algorithmenstruktur, der Anzahl der verwendeten Validierer, etc. und kann nicht zusammengefasst werden.
Die Einteilung der Validierer in verschiedene logische Cluster auf Basis ihrer

Eigenschaften beschreibt den vierten Schutzmechanismus. Durch die Verwendung von Clustern können verschiedene Validierer mit unterschiedlichen Eigenschaften kombiniert und Zero Day Exploits mitigiert werden.
Der letzte Ansatz verwendet den Cluster Ansatz und führt darauf aufbauend zwei verschiedene Aggregationsfunktionen (*t-Total Quorum* und (s,t)-*Nested Quorum*) ein, die zur Aggregation der Validierer-Ergebnisse verwendet werden. Das *t-Total Quorum* gruppiert alle Validiererergebnisse und sobald eine Gruppe einen bestimmten Threshold überschreitet, wird dieses Validiererergebnis als korrekt oder vertrauenswürdig angenommen. Ein (s,t)-*Nested Quorum* verwendet zusätzlich zum *t-Total Quorum* eine Gruppierung mit Threshold Kontrolle in den jeweiligen Clustern. Dadurch wird die Vertrauenswürdigkeit der einzelnen Cluster berücksichtigt und korrupte Cluster werden von vornherein ausgeschlossen.

9.2. Ausblick

Im Folgenden sind einige vielversprechende Erweiterungen und Themen für weitere Forschungsarbeiten aufgeführt:

Die vorliegende Dissertation hat sich mit der Erhöhung der Sicherheit für die Datenvalidierung mittels DDVN beschäftigt. Dabei kann das DDVN so erweitert werden, dass im Falle eines erfolgreichen Angriffs der erfolgreiche Angriff auf das Netzwerk erkannt wird. Dieses Verfahren entspräche einem IDS und würde das DDVN um eine zusätzliche Sicherheitsfunktionalität erweitern. Zusätzlich sollte untersucht werden, ob es eine Möglichkeit gibt, den korrumpierten Knoten im Netzwerk zu identifizieren.

Eine weitere Forschungsfrage betrifft die Anbindung von Validierern und Komponenten im DDVN ohne manuelle Konfiguration. In diesem Zusammenhang sind [181] Zero Configuration Ansätze und deren Sicherheit näher zu untersuchen. Bereits existierende Zero Configuration Lösungen wie Avahi oder Mono.Zeroconf können für die Analyse herangezogen werden.

Der Akkumulator aggregiert die einzelnen Validierungsergebnisse zu einem Gesamtergebnis und verwendet dieses, um über die Qualität der erhaltenen Daten zu entscheiden. In dieser Arbeit werden zwei Möglichkeiten der Aggregation vorgestellt. Bei der Untersuchung der Aggregationsfunktionen müssen die verschiedenen Arten von Validierungsergebnissen (Wahrheitswert, Ganzzahl, Fliesskommazahl, Vektor usw.) sowie deren Normalisierung bzw. Zusammenführung berücksichtigt werden. Weiterhin ist zu definieren, welches Ergebnis nach der Qualitätsaggregation vorliegt und wie das jeweilige Aggregationsergebnis zu interpretieren ist. Je nach Anwendungsfall können bestimmte Ergebnisse als kritisch oder unkritisch eingestuft werden.

Eine weitere offene Forschungsfrage ist die Fehlerbehandlung. Wie soll das DDVN reagieren, wenn Datenübertragungen ausbleiben oder in einem Quorum keine

Mehrheitsentscheidung getroffen werden kann? Wird in einem solchen Fall der Akkumulator informiert oder gibt es eine unabhängige Fehlerbehandlungskomponente, die diesen Fehler behandelt? Wenn es eine Fehlerbehandlungskomponente gibt, kann diese auch entscheiden, ob eine erneute Validierung durchgeführt wird. Gegebenenfalls kann auch ein Support-Personal informiert werden. Die verantwortliche Person kann dann reagieren und weitere Schritte planen.
Neben der Fehlerbehandlung sollte auch eine Zustandsüberwachung in Betracht gezogen werden. Diese kann frühzeitig auf notwendige Wartungen der Komponenten hinweisen und bei Änderungen der zu überwachenden Daten, z.B. Austausch eines Sensors an der Maschine, auch eine Anpassung der Validierungsfunktion veranlassen. Eine eigenständige Aktualisierung der Validierungsfunktion durch die Zustandsüberwachungskomponente ist zu prüfen.

Die zeitlichen Unterschiede zwischen den für die Validierung erhaltenen Daten wurden in der vorliegenden Arbeit nicht weiter berücksichtigt. Es ist zu untersuchen, wie die Daten am besten zueinander in Beziehung gesetzt werden können. Eine Möglichkeit ist die Generierung eines Zeitstempels beim Empfang der Daten, eine andere Möglichkeit ist, dass die sendende Komponente einen eigenen Zeitstempel hinterlegt. Weiterhin ist zu definieren, wie lange Daten gesammelt werden, bis sie für eine Validierung herangezogen werden. Wenn bestimmte Daten erst mit Verzögerung eintreffen, kann die Vernachlässigung dieser Daten zu einem falschen Validierungsergebnis führen.

A. Anhang

Nachfolgend sind einige wichtige Auszüge der Vollständigkeit wegen aufgelistet. Dazu gehören die Darstellung des gesamten Kommunikationsflusses im DDVN, eine Zusammenfassung der Cyberangriffe von 1988 bis 2020, ein gültiges Beispiel eines PKCs und einige zentrale Codeausschnitte für die Analyse des azyklischen gerichteten Graphen ohne Mehrfachkanten, der Baumstruktur und der Aggregationsfunktionen der Akkumulatoren.

A.1. Gesamter Kommunikationsablauf

Der gesamte Kommunikationsablauf im DDVN is in Abbildung A.1 und Abbildung A.2 dargestellt. Der Unterschied zwischen den beiden besteht darin, dass in Abbildung A.2 der Validator die Daten an Akkumulator weiterleitet und nicht das Gateway.

Abbildung A.1.: Kommunikationsablauf, bei dem das Gateway die Daten an den Akkumulator weiterleitet

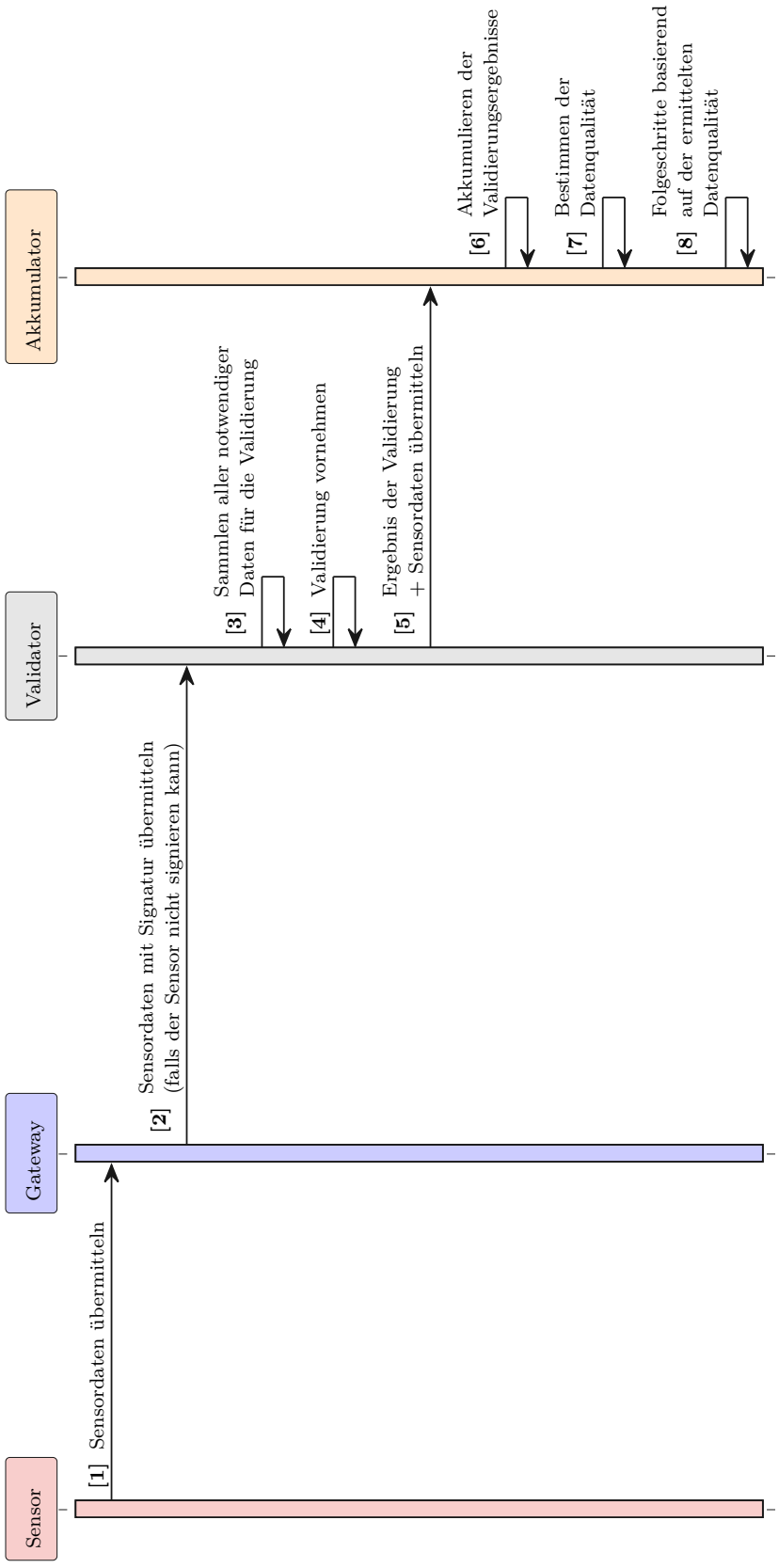


Abbildung A.2.: Kommunikationsablauf, bei dem der Validator die Daten an den Akkumulator weiterleitet

A.2. Cyberangriffe von 1988 bis 2020

Miller et al. [71] fassen die Cyberangriffe von 1988 bis 2020 zusammen. In Tabelle A.1 ist diese Zusammenfassung dargestellt.

Tabelle A.1.: Zusammenfassung von Angriffen (1988-2020)[71]

Angriff auf	Jahr	Zugriff	Angreifer	Sektor	Auswirkung
PLC Passwort Änderung	1988	Arbeitsplatz	Insider	Fertigung	Verlust der Kontrolle
Ignalina Kernkraftwerk	1992	Arbeitsplatz	Insider	Kernkraft	Produktivitäts- und Umsatzverluste
Notfallalarmsystem der Chevron-Raffinerie	1992	Arbeitsplatz	Einzelperson	Chemie	Produktivitäts- und Umsatzverluste
Salt River Projekt	1994	Gerät mit Internetanbindung	Einzelperson	Energie und Wasser	Produktivitäts- und Umsatzverluste, Festplattenlöschung
Omega Engineering	1996	Arbeitsplatz	Einzelperson	Fertigung	Festplattenlöschung
Worcester, MA Flughafen	1997	Gerät mit Internetanbindung	Einzelperson	Transport	Produktivitäts- und Umsatzverluste, nicht erreichbar und Sicherheitsprobleme
Gazprom	1999	Unbekannt	Organisierte Gruppe und Insider	Chemie und Energie	Produktivitäts- und Umsatzverluste
Bradwell Kernkraftwerk	1999	Arbeitsplatz	Insider	Kernkraft	Festplattenlöschung
Maroochy Wassersystem	2000	Wireless	Insider	Wasser	Schaden am Eigentum
Cal-ISO System	2001	Unbekannt	Staat	Energie	Nicht veröffentlicht
Virus im Fertigungssystem	2001	Spear Phishing	Staat	Fertigung	Produktivitäts- und Umsatzverluste

Tabelle A.1.: Zusammenfassung von Angriffen (1988-2020)[71]

Angriff auf	Jahr	Zugriff	Angreifer	Sektor	Auswirkung
Hafen von Houston	2001	Gerät mit Internetanbindung	Einzelperson	Transport	Produktivitäts- und Umsatzverluste
Gasaufbereitung	2001	Vertrauensverhältnis	Zulieferer	Chemie	Produktivitäts- und Umsatzverluste
PDVSA	2002	Gerät mit Internetanbindung	Organisierte Gruppe	Chemie	Produktivitäts- und Umsatzverluste, Festplattenlöschung
Flugplanungssystem	2003	Unbekannt	Einzelperson	Transport	Produktivitäts- und Umsatzverluste
CSX-Zugsignalisierungssystem	2003	Spear Phishing	Unbekannt	Transport	Produktivitäts- und Umsatzverluste
Infektion eines SCADA Netzwerks	2004	Replikation durch Wechseldatenträger	Unbekannt	Lebensmittel	Produktivitäts- und Umsatzverluste
Daimler Chrysler Werke	2005	Externer Remote Service	Einzelperson	Fertigung	Produktivitäts- und Umsatzverluste
Tehama-Colusa Canal	2007	Arbeitsplatz	Einzelperson	Wasser	Schaden am Eigentum
Lodz Strassenbahnsystem	2008	Externer Remote Service	Einzelperson	Transport	Verlust der Sicherheit
US Stromnetz	2009	Gerät mit Internetanbindung	Staat	Energie	Nicht veröffentlicht
HVAC Krankenhaus	2009	Arbeitsplatz	Insider	Gesundheit	Verlust der Sicherheit

Tabelle A.1.: Zusammenfassung von Angriffen (1988-2020)[71]

Angriff auf	Jahr	Zugriff	Angreifer	Sektor	Auswirkung
Night Dragon	2009	Ausnutzung einer öffentlichen Anwendung	Organisierte Gruppe	Energie	Diebstahl von Betriebsdaten
Sality Virus infiziert DVS Server	2009	Unbekannt	Unbekannt	Chemie	Verlust der Erreichbarkeit
Stuxnet	2010	Replikation durch Wechseldatenträger	Staat	Kernkraft	Schaden am Eigentum, Manipulation der Erreichbarkeit und Steuerung
Shionogi	2011	Arbeitsplatz	Einzelperson	Gesundheit	Festplattenlöschung
Niagra AX	2012	Gerät mit Internetanbindung	Unbekannt	Fertigung	Manipulation der Steuerung
Spionage des Iranischen CI	2012	Replikation durch Wechseldatenträger	Staat	Chemie	Diebstahl von Betriebsdaten und ungewollte Festplattenlöschung
Turbinen Steuerungssystem	2012	Replikation durch Wechseldatenträger	Organisierte Gruppe	Energie	Produktivitäts- und Umsatzverluste, Diebstahl von Betriebsdaten
Rye Brook Damm	2013	Gerät mit Internetanbindung	Organisierte Gruppe	Wasser und Energie	Nicht veröffentlich

Tabelle A.1.: Zusammenfassung von Angriffen (1988-2020)[71]

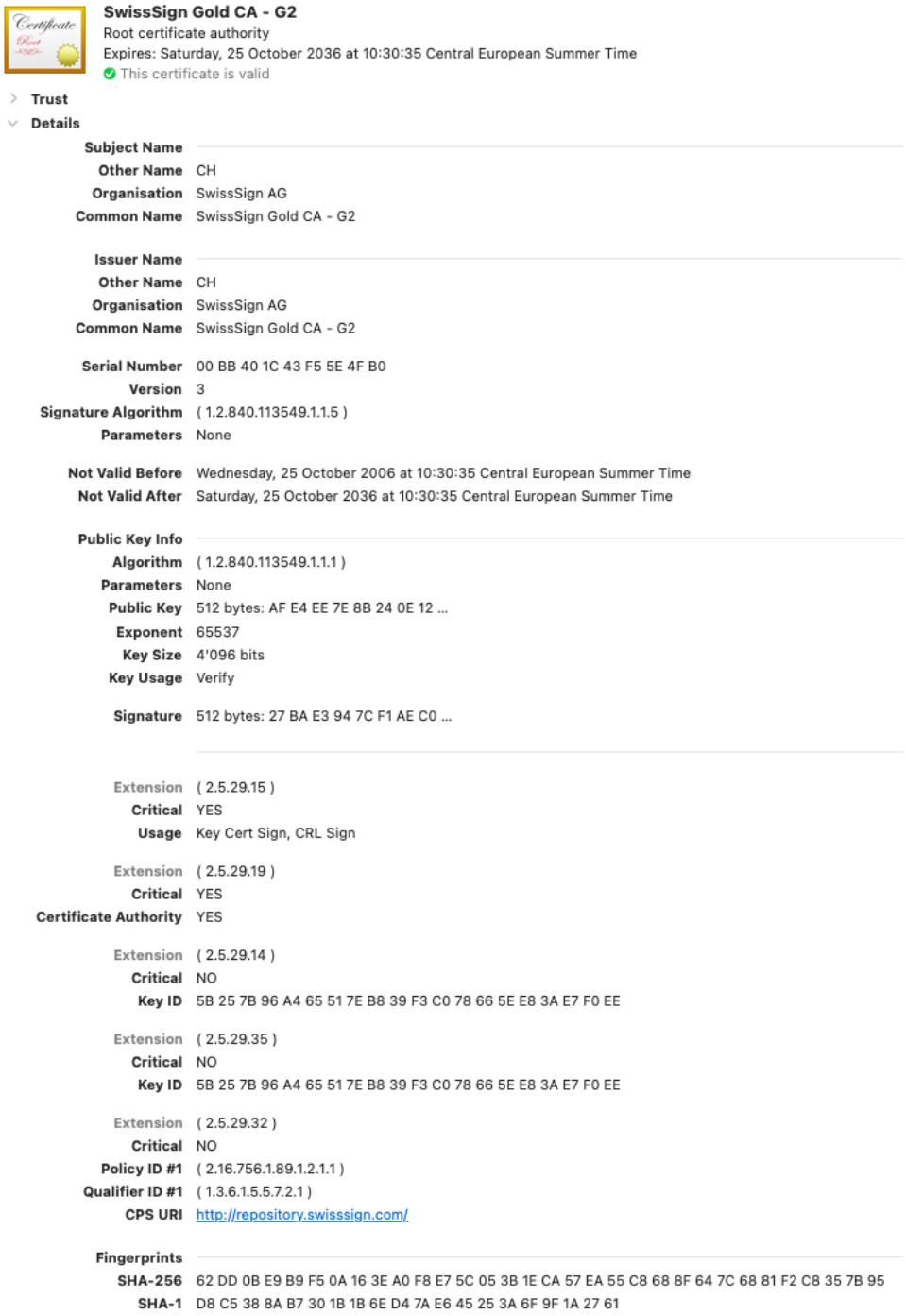
Angriff auf	Jahr	Zugriff	Angreifer	Sektor	Auswirkung
Europäische Versorgungsdienste angegriffen	2014	Spear Phishing	Organisierte Gruppe	Verschiedene	Verlust der Erreichbarkeit, Diebstahl von Betriebsdaten
Deutsches Stahlwerk	2014	Spear Phishing	Unbekannt	Fertigung	Schaden am Eigentum
Ukrainische Stromerzeugung	2015	Spear Phishing	Organisierte Gruppe	Energie	Produktivitäts- und Umsatzverluste
Ukrainische Stromerzeugung	2016	Spear Phishing	Organisierte Gruppe	Energie	Festplattenlöschung, Produktivitäts- und Umsatzverluste, Verlust der Sicherheit
Wolf Creek	2017	Spear Phishing	Organisierte Gruppe	Kernkraft	Nicht veröffentlicht
Cadbury Fertigung	2017	Externer Remote Service	Organisierte Gruppe	Lebensmittel	Produktivitäts- und Umsatzverluste
Triton/Petro Rabigh	2017	Arbeitsplatz	Staat	Chemie	Verlust der Kontrolle, Verlust der Sicherheit
Norsk Hydro	2019	Spear Phishing	Unbekannt	Fertigung und Energie	Verlust der Erreichbarkeit
Triton/Nicht veröffentlicht	2019	Arbeitsplatz	Staat	Nicht veröffentlicht	Verlust der Kontrolle, Schadem am Eigentum, Verlust der Sicherheit

Tabelle A.1.: Zusammenfassung von Angriffen (1988-2020)[71]

Angriff auf	Jahr	Zugriff	Angreifer	Sektor	Auswirkung
Ölproduzenten	2020	Spear Phishing	Unbekannt	Chemie	Diebstahl von Betriebsdaten
Israelische Wasserversorgung	2020	Gerät mit Internetanbindung	Organisierte Gruppe	Wasser	Nicht veröffentlicht
Hafen von Shahid Rajaie	2020	Unbekannt	Staat	Transport	Produktivitäts- und Umsatzverluste
Honda Fertigungsanlagen	2020	Spear Phishing	Unbekannt	Fertigung	Verlust der Kontrolle

A.3. Beispiel für ein Public Key Certificate

In Abbildung A.3 ist ein Beispiel für ein gültiges PKC aufgezeigt.



SwissSign Gold CA - G2
Root certificate authority
Expires: Saturday, 25 October 2036 at 10:30:35 Central European Summer Time
This certificate is valid

Trust
Details

Subject Name	
Other Name	CH
Organisation	SwissSign AG
Common Name	SwissSign Gold CA - G2
Issuer Name	
Other Name	CH
Organisation	SwissSign AG
Common Name	SwissSign Gold CA - G2
Serial Number	00 BB 40 1C 43 F5 5E 4F B0
Version	3
Signature Algorithm	(1.2.840.113549.1.1.5)
Parameters	None
Not Valid Before	Wednesday, 25 October 2006 at 10:30:35 Central European Summer Time
Not Valid After	Saturday, 25 October 2036 at 10:30:35 Central European Summer Time
Public Key Info	
Algorithm	(1.2.840.113549.1.1.1)
Parameters	None
Public Key	512 bytes: AF E4 EE 7E 8B 24 0E 12 …
Exponent	65537
Key Size	4'096 bits
Key Usage	Verify
Signature	512 bytes: 27 BA E3 94 7C F1 AE C0 …
Extension	(2.5.29.15)
Critical	YES
Usage	Key Cert Sign, CRL Sign
Extension	(2.5.29.19)
Critical	YES
Certificate Authority	YES
Extension	(2.5.29.14)
Critical	NO
Key ID	5B 25 7B 96 A4 65 51 7E B8 39 F3 C0 78 66 5E E8 3A E7 F0 EE
Extension	(2.5.29.35)
Critical	NO
Key ID	5B 25 7B 96 A4 65 51 7E B8 39 F3 C0 78 66 5E E8 3A E7 F0 EE
Extension	(2.5.29.32)
Critical	NO
Policy ID #1	(2.16.756.1.89.1.2.1.1)
Qualifier ID #1	(1.3.6.1.5.5.7.2.1)
CPS URI	http://repository.swisssign.com/
Fingerprints	
SHA-256	62 DD 0B E9 B9 F5 0A 16 3E A0 F8 E7 5C 05 3B 1E CA 57 EA 55 C8 68 8F 64 7C 68 81 F2 C8 35 7B 95
SHA-1	D8 C5 38 8A B7 30 1B 1B 6E D4 7A E6 45 25 3A 6F 9F 1A 27 61

Abbildung A.3.: Beispiel für ein gültiges Public Key Zertifikat

A.4. Programmcode für die Analysen

Im Folgenden sind einige Ausschnitte aus dem Programmcode für die verschiedenen Analysen dargestellt. Es handelt sich dabei um die Analysen für die Erfolgswahrscheinlichkeit eines Angriffs auf einen azyklischen gerichteten Graphen ohne Mehrfachkanten, die Berechnungen für die optimale Positionierung von Quoren in einer Baumstruktur und die Bestimmung des Korruptionsgrades in einer Baumstruktur.

A.4.1. Ermittlung der Erfolgswahrscheinlichkeit eines Angriffs auf einen azyklischen gerichteten Graphen ohne Mehrfachkanten

Zur Bestimmung der Erfolgswahrscheinlichkeit eines Angriffs auf einen azyklischen gerichteten Graphen ohne Mehrfachkanten wurden zahlreiche Simulationen durchgeführt. Ein Ausschnitt des zugehörigen Programmcodes[1] ist in Listing A.1, Listing A.2 und Listing A.3 gegeben. Grundsätzlich besteht die Simulation aus drei Schritten: 1) Erzeugen einer beliebigen korrupten Kombination von Knoten, 2) Aufteilen der erzeugten Kombination in beliebige Layer und 3) Ausführen der Simulation mit definierten Parametern.

```
def fixed_corrupted_generator(number_of_validators,
                              iterations,
                              number_of_corrupted=0):
# Erzeugen einer fixen Anzahl an korrumpierten
# Validierern pro Iteration. Entweder fix oder
# variabel, um Proportionen darzustellen.
return 1

def create_random_corrupted_combination(
  number_of_validators, number_of_corrupted):
# Erzeugen einer zufälligen Kombination an normalen
# und korrumpierten Validierern
# Mögliche Ergebnisse (number_of_corrupted=2):
#   [0, 1, 0, 0, 1]
#   [1, 0, 0, 1, 0]
if number_of_corrupted >= number_of_validators:
  return [1] * number_of_validators

combination = [0] * number_of_validators
for corrupted_index in random.sample(
```

[1] https://github.com/wake-0/dnn-corruption-simulation

```
                            range(number_of_validators),
                            number_of_corrupted):
  combination[corrupted_index] = 1
return combination

def create_random_corrupted_combinations(
  number_of_validators, iterations):
# Erzeugt eine Menge an zufälligen Kombination an
# normalen und korrumpierten Validierern
for number_of_corrupted in fixed_corrupted_generator(
                              number_of_validators,
                              iterations,
                              number_of_validators):
  combination = create_random_corrupted_combination(
                              number_of_validators,
                              number_of_corrupted)
  yield combination
```

Listing A.1: Erzeugen einer zufälligen Kombination an korrumpierten Knoten

```
def split_into_layers(validators, number_of_layers,
                      quorum_size, quorum_count):
# Aufteilen eines Validierer Arrays in Validierer
# pro Layer; das ist für die Quoren notwendig.
layers = [0] * number_of_layers
for quorum_index in random.sample(
                            range(0, number_of_layers),
                            quorum_count):
  # Alle Layer die einem Quorum entsprechen
  # werden markiert
  layers[quorum_index] = 1

# Knoten pro Layer ermitteln und zurückgeben.
current_index = 0
for layer in layers:
  if layer == 0:
    yield [validators[current_index]]
    current_index += 1
  else:
    yield validators[current_index:
                     current_index+quorum_size]
    current_index += quorum_size
```

Listing A.2: Erzeugen einer definierten Anzahl von Layern mit zufällig positionierten Quoren

```
def simulate(iterations, number_of_layers, quorum_size):

result = {
    'quorum_count': [],
    'attack_success_p': []
}
number_of_corrupted = {}

# Alle möglichen Anzahlen an Quoren simulieren
for quorum_count in range(0, number_of_layers + 1):
  validators_count = (quorum_size*quorum_count) +
                     (number_of_layers - quorum_count)
  number_of_corrupted[quorum_count] = 0
  min_quorum_threshold = quorum_size // 2 + 1

  for combination in create_random_corrupted_combinations(
    validators_count, iterations):

    for quorum in split_into_layers(combination,
                                    number_of_layers,
                                    quorum_size,
                                    quorum_count):
      current_threshold = sum(quorum)

      # Kein Quorum
      if len(quorum) == 1:
        # Kontrolle, ob korrumpiert
        if current_threshold == 1:
          number_of_corrupted[quorum_count] =
            number_of_corrupted[quorum_count] + 1
          break

      # Quorum
      else:
        # Kontrolle, ob korrumpiert
        if current_threshold >= min_quorum_threshold:
          number_of_corrupted[quorum_count] =
            number_of_corrupted[quorum_count] + 1
          break

# Alle unterschiedlichen Quoren untersuchen
for quorum_count in range(0, number_of_layers + 1):
  successful_corrupted = number_of_corrupted[quorum_count]
```

```
  attack_success_p = successful_corrupted / iterations
  print("Quorum count:" + str(quorum_count))
  print("(Successful corrupted count:" +
        str(successful_corrupted) + ")")
  print("(Quorum count:" + str(quorum_count) +
        "; Attack success probability:" +
        str(attack_success_p) + ")")
  print("-"*50)

  result['quorum_count'].append(quorum_count)
  result['attack_success_p'].append(attack_success_p)
  return result

# Aufruf der Simulation
iterations = 1000
quorum_size = 7
number_of_layers = 10
result = simulate(iterations,
                  number_of_layers,
                  quorum_size)
print(result)

quorum_count = result['quorum_count']
attack_success_p = result['attack_success_p']

plt.xlabel('Quorum count')
plt.ylabel('Attack success probability')
plt.title('Quorum attack success probability')
plt.bar(quorum_count, attack_success_p)
plt.show()
```

Listing A.3: Durchführung einer Simulation zur Ermittlung der Erfolgswahrscheinlichkeit eines Angriffs bei unterschiedlichen Quorumszahlen

A.4.2. Positionierung von zwei Quoren in einer Baumstruktur mit drei Knoten

In Listing A.4 ist der Programmcode[2] zur Bestimmung der optimalen Positionierung eines Quorums in einer Baumstruktur mit zwei Quoren dargestellt. Die Baumstruktur besteht aus drei Knoten, einem Wurzelknoten sowie einem linken und einem rechten Blattknoten. Die Quoren werden entweder am

[2]https://github.com/wake-0/dt-simple-quorum-positioning

Wurzelknoten und an einem Blattknoten oder an beiden Blattknoten platziert. Die Grösse des Quorums beträgt drei Knoten. Im vorliegenden Code wird davon ausgegangen, dass im gesamten Algorithmus zwei Knoten korrupt sind.

```
# Hier werden zwei korrumpierte Knoten angenommen
# und alle möglichen Permutationen ermittelt
permutations = list(itertools.permutations(
      [1,1,0,0,0,0,0]
    ))

unique = set(permutations)

paths = 0
corrupted = 0

# Die Quoren werden am Wurzelknoten und
# dem linken Blattknoten angenommen
for entry in unique:
  # Linker Pfad korrumpiert?
  paths = paths + 1
  is_corrupted = sum(entry[0:3]) == 2 or
                 sum(entry[3:6]) == 2
  if is_corrupted: corrupted = corrupted + 1

  # Rechter Pfad korrumpiert?
  paths = paths + 1
  is_corrupted = sum(entry[0:3]) == 2 or
                 entry[6] == 1
  if is_corrupted: corrupted = corrupted + 1

print(f'{paths}-{corrupted}:{corrupted/paths}')

paths = 0
corrupted = 0

# Die Quoren werden an beiden Blattknoten angenommen
for entry in unique:
  # Linker Pfad korrumpiert?
  paths = paths + 1
  is_corrupted = entry[0] == 1 or
                 sum(entry[1:4]) == 2
  if is_corrupted: corrupted = corrupted + 1

  # Rechter Pfad korrumpiert
```

```
  paths = paths + 1
  is_corrupted = entry[0] == 1 or
                 sum(entry[4:7]) == 2
  if is_corrupted: corrupted = corrupted + 1

print(f'{paths}-{corrupted}:{corrupted/paths}')
```

Listing A.4: Berechnung der Position von zwei Quoren mit je drei Knoten in einer Baumstruktur mit drei Knoten

A.4.3. Ermittlung des Korruptionsgrads bei einer Baumstruktur

Nachfolgend ein Ausschnitt aus dem Programmcode[3] (siehe Listing A.5) zur Bestimmung des Korruptionsgrades für eine Baumstruktur. Die vier zentralen Komponenten sind 1) ein Generator für die Baumkombinationen *combinations_generator*, 2) ein Generator für die verschiedenen Pfade durch die gegebene Baumstruktur *paths_generator*, 3) eine Kontrollfunktionalität *corruption_validator*, die für die Feststellung einer erfolgreichen Korruption verantwortlich ist, und 4) eine Berechnungsmethode für den Korruptionsgrad *corruption_factor_calculator*.

```
def analyse_corruption_factors(
  combinations_generator,
  combinations_params,
  paths_generator,
  path_params,
  corruption_factor_calculator,
  corruption_factor_calculator_params,
  corruption_validator):

corruption_factors = {}
paths = list(paths_generator(**path_generator_params))

# Durchlaufen der definierten Baumstrukturkombinationen
# In jedem tree wurde bereits ein bestimmte Anzahl
# an Knoten korrumpiert; z.B. per Zufall
for tree in combinations_generator(
  **combinations_generator_params):

  # Bei einer Baumstruktur zählen nur die korrumpierten
  # Elternknoten, da alle Pfade mit diesen Knoten
  # automatisch korrumpiert sind
```

[3] https://github.com/wake-0/Distributed-Decision-Trees-Corruption-Analysis

```
  corrupted_parent_nodes = set()
  for path in paths:
    for node in path:
      if corruption_validator(tree[node]):
          corrupted_parent_nodes.add(node)
          break

  # Berechnen des Korruptionsgrads für die
  # aktuelle Baumstruktur
  corruption_factor = {
    corruption_factor_calculator(
      corrupted_parent_nodes,
      **corruption_factor_calculator_params
      ): 1}

  # Zusammenführen des neuen Korruptionsgrads mit
  # den bereits berechneten
  corruption_factors = dict(
    collections.Counter(corruption_factors) +
    collections.Counter(corruption_factor)
    )

  yield corruption_factors
```

Listing A.5: Algorithmus zur Analyse des Korruptionsgrades bei einer Baumstruktur

A.4.4. Ermittlung der Aggregation beim Akkumulator

Nachfolgend ein Ausschnitt aus dem Programmcode[4] (siehe Listing A.6) zur Ermittlung der Aggregation von Validierungsergebnissen auf einen Akkumulator. Dabei werden verschiedene Thresholds für ein (s, t)-*Nested Quorum* untersucht.

```
# Anzahl aller Knoten im Netzwerk
nodes_number = 64
# Anzahl der korrumpierten Knoten
corrupted_number = 32
# Anzahl der Cluster im Netzwerk
cluster_number = 16
# Anzahl der ausgewählten Nodes pro Cluster
picked_nodes_number = 4
```

[4] https://github.com/wake-0/accumulator-aggregations

```
for t in [[0.25,0.25],[0.25,0.5],[0.25,0.75],[0.5,0.5],
          [0.5,0.75],[0.75,0.25],[0.75,0.5],[0.75,0.75]]:
  successful_corrupted_number = 0
  # Anzahl der gleichen Ergebnisse im gesamten Quorum
  threshold = t[0]
  # Anzahl der gleichen Ergebnisse im Nested Quorum
  nested_threshold = t[1]

  for _iteration in range(100000):
    # Erzeugen von gleichgrossen Clustern
    clusters = utils.create_clusters(nodes_number,
                                     cluster_number)

    # Korrumpieren von Knoten in den Clustern
    corrupted_clusters = utils.
      create_equaly_corrupted_clusters(clusters,
                                       corrupted_number)

    # Cluster mit korrumpierten Knoten
    corrupted_clusters_after_selection = utils.
      select_nodes_in_clusters(corrupted_clusters,
                               picked_nodes_number)

    successful_corrupted_cluster_number = 0
    min_corrupted = nested_threshold * picked_nodes_number
    for cluster in corrupted_clusters_after_selection:
      if sum(cluster) > min_corrupted:
        successful_corrupted_cluster_number += 1

    successful_corrupted_ratio =
      successful_corrupted_cluster_number / cluster_number
    if successful_corrupted_ratio > threshold:
      successful_corrupted_number += 1

  success_rate =
    successful_corrupted_number / iteration_number
  print('Corrupted: ' + str(successful_corrupted_number))
  print('Ccorrupted ratio: ' + str(success_rate))
```

Listing A.6: Algorithmus zur Analyse eines (s,t)-Nested Quorum Ansatzes mit unterschiedlichen Thresholds

Literaturverzeichnis

[1] H. Kagermann, W.-D. Lukas, und W. Wahlster, „Industrie 4.0: Mit dem Internet der Dinge auf dem Weg zur 4. industriellen Revolution," *VDI Nachrichten*, Vol. 13, Nr. 1, S. 2–3, 2011.

[2] G. Oswald und H. Krcmar, *Digitale Transformation: Fallbeispiele und Branchenanalysen*. Springer Nature, 2018.

[3] F. Wortmann und K. Flüchter, „Internet of Things," *Business & Information Systems Engineering*, Vol. 57, Nr. 3, S. 221–224, 2015.

[4] L. Wang und R. X. Gao, *Condition monitoring and control for intelligent manufacturing*. Springer Science & Business Media, 2006.

[5] W. Van Der Aalst, „Process mining," *Communications of the ACM*, Vol. 55, Nr. 8, S. 76–83, 2012.

[6] R. K. Mobley, *An introduction to predictive maintenance*. Elsevier, 2002.

[7] F. Tao, Q. Qi, A. Liu, und A. Kusiak, „Data-driven smart manufacturing," *Journal of Manufacturing Systems*, Vol. 48, S. 157–169, 2018.

[8] European Parliament und Council of the European Union, „Regulation (EU) 2016/679 of the European Parliament and of the Council." [Online]. Verfügbar: https://data.europa.eu/eli/reg/2016/679/oj

[9] H. B. Braiek und F. Khomh, „On testing machine learning programs," *Journal of Systems and Software*, Vol. 164, 2020. [Online]. Verfügbar: https://www.sciencedirect.com/science/article/pii/S0164121220300248

[10] J. Bogner, R. Verdecchia, und I. Gerostathopoulos, „Characterizing Technical Debt and Antipatterns in AI-Based Systems: A Systematic Mapping Study," *CoRR*, Vol. abs/2103.09783, 2021. [Online]. Verfügbar: https://arxiv.org/abs/2103.09783

[11] A. Famili, W.-M. Shen, R. Weber, und E. Simoudis, „Data preprocessing and intelligent data analysis," *Intelligent data analysis*, Vol. 1, Nr. 1, S. 3–23, 1997.

[12] S. García, J. Luengo, und F. Herrera, *Data preprocessing in data mining*. Springer, 2015, Vol. 72.

[13] S. A. Alasadi und W. S. Bhaya, „Review of data preprocessing techniques in data mining," *Journal of Engineering and Applied Sciences*, Vol. 12, Nr. 16, S. 4102–4107, 2017.

[14] M. De Donno, K. Tange, und N. Dragoni, „Foundations and evolution of modern computing paradigms: Cloud, iot, edge, and fog," *IEEE Access*, Vol. 7, 2019.

[15] E. J. Husom, S. Tverdal, A. Goknil, und S. Sen, „UDAVA: An Unsupervised Learning Pipeline for Sensor Data Validation in Manufacturing," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, Serie CAIN '22. New York, NY, USA: Association for Computing Machinery, 2022, S. 159–169. [Online]. Verfügbar: https://doi.org/10.1145/3522664.3528603

[16] I. Taleb, M. A. Serhani, und R. Dssouli, „Big Data Quality Assessment Model for Unstructured Data," in *2018 International Conference on Innovations in Information Technology (IIT)*, 2018, S. 69–74.

[17] D. Xu, Z. Zhang, und J. Shi, „A Data Quality Assessment and Control Method in Multiple Products Manufacturing Process," in *2022 5th International Conference on Data Science and Information Technology (DSIT)*, 2022, S. 1–5.

[18] L. L. Pipino, Y. W. Lee, und R. Y. Wang, „Data Quality Assessment," *Commun. ACM*, Vol. 45, Nr. 4, S. 211–218, April 2002. [Online]. Verfügbar: https://doi.org/10.1145/505248.506010

[19] J. Quevedo, J. Pascual, S. Espin, und J. Roquet, „Data Validation and Reconstruction for Performance Enhancement and Maintenance of Water Networks," *IFAC-PapersOnLine*, Vol. 49, Nr. 28, S. 203–207, 2016, 3rd IFAC Workshop on Advanced Maintenance Engineering, Services and Technology AMEST 2016. [Online]. Verfügbar: https://www.sciencedirect.com/science/article/pii/S2405896316324594

[20] M. A. Cugueró-Escofet, D. García, J. Quevedo, V. Puig, S. Espin, und J. Roquet, „A methodology and a software tool for sensor data validation/reconstruction: Application to the Catalonia regional water network," *Control Engineering Practice*, Vol. 49, S. 159–172, 2016. [Online]. Verfügbar: https://www.sciencedirect.com/science/article/pii/S0967066115300459

[21] X. Xin, S. L. Keoh, M. Sevegnani, und M. Saerbeck, „Dynamic probabilistic model checking for sensor validation in Industry 4.0 applications," in *2020 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2020, S. 43–50.

[22] M. Mohammed, J. R. Talburt, S. Dagtas, und M. Hollingsworth, „A Zero Trust Model Based Framework For Data Quality Assessment," in *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2021, S. 305–307.

[23] S. Zan und X. Zhang, „Medical Data Quality Assessment Model Based

on Credibility Analysis," in *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, 2018, S. 940–944.

[24] H. Foidl, M. Felderer, und R. Ramler, „Data Smells: Categories, Causes and Consequences, and Detection of Suspicious Data in AI-Based Systems," in *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*, Serie CAIN '22. New York, NY, USA: Association for Computing Machinery, 2022, S. 229–239. [Online]. Verfügbar: https://doi.org/10.1145/3522664.3528590

[25] H. Sándor, B. Genge, und Z. Szántó, „Sensor data validation and abnormal behavior detection in the Internet of Things," in *2017 16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, September 2017, S. 1–5.

[26] P. Blanchard, E. M. El Mhamdi, R. Guerraoui, und J. Stainer, „Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, und R. Garnett, Hrsgg., Vol. 30. Curran Associates, Inc., 2017. [Online]. Verfügbar: https://proceedings.neurips.cc/paper/2017/file/f4b9ec30ad9f68f89b29639786cb62ef-Paper.pdf

[27] Y. Chen, L. Su, und J. Xu, „Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent," *Proc. ACM Meas. Anal. Comput. Syst.*, Vol. 1, Nr. 2, Dezember 2017. [Online]. Verfügbar: https://doi.org/10.1145/3154503

[28] N. Dalvi, P. Domingos, S. Sanghai, und D. Verma, „Adversarial classification," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, S. 99–108.

[29] N. Papernot, P. McDaniel, X. Wu, S. Jha, und A. Swami, „Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE symposium on security and privacy (SP)*. IEEE, 2016, S. 582–597.

[30] D. Jakubovitz und R. Giryes, „Improving DNN Robustness to Adversarial Attacks Using Jacobian Regularization," in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu, und Y. Weiss, Hrsgg. Cham: Springer International Publishing, 2018, S. 525–541.

[31] W. Xu, D. Evans, und Y. Qi, „Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

[32] R. M. Shukla, S. Badsha, D. Tosh, und S. Sengupta, *Robust Machine Learning Using Diversity and Blockchain*. Cham: Springer International Publishing, 2021, S. 103–121. [Online]. Verfügbar: https://doi.org/10.1007/978-3-030-55692-1_6

[33] H. Liu, F. Huang, H. Li, W. Liu, und T. Wang, „A Big Data Framework for Electric Power Data Quality Assessment," in *2017 14th Web Information Systems and Applications Conference (WISA)*, 2017, S. 289–292.

[34] B. McMahan, E. Moore, D. Ramage, S. Hampson, und B. A. y Arcas, „Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. PMLR, 2017, S. 1273–1282.

[35] T. Li, A. K. Sahu, A. Talwalkar, und V. Smith, „Federated learning: Challenges, methods, and future directions," *IEEE signal processing magazine*, Vol. 37, Nr. 3, S. 50–60, 2020.

[36] E. N. Witanto, B. Stanley, und S.-G. Lee, „Distributed Data Integrity Verification Scheme in Multi-Cloud Environment," *Sensors*, Vol. 23, Nr. 3, 2023. [Online]. Verfügbar: https://www.mdpi.com/1424-8220/23/3/1623

[37] S. V. N. S. Kumar, M. Selvi, und A. Kannan, „A Comprehensive Survey on Machine Learning-Based Intrusion Detection Systems for Secure Communication in Internet of Things," *Computational Intelligence and Neuroscience*, Vol. 2023, S. 1–24, Januar 2023. [Online]. Verfügbar: https://doi.org/10.1155/2023/8981988

[38] S. Misra, P. V. Krishna, H. Agarwal, A. Saxena, und M. S. Obaidat, „A Learning Automata Based Solution for Preventing Distributed Denial of Service in Internet of Things," in *2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing*, 2011, S. 114–122.

[39] Q. D. La, T. Q. S. Quek, J. Lee, S. Jin, und H. Zhu, „Deceptive Attack and Defense Game in Honeypot-Enabled Networks for the Internet of Things," *IEEE Internet of Things Journal*, Vol. 3, Nr. 6, S. 1025–1035, 2016.

[40] D. L. Chaum, „Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, Vol. 24, Nr. 2, S. 84–90, 1981.

[41] Z. Zheng, S. Xie, H.-N. Dai, X. Chen, und H. Wang, „Blockchain challenges and opportunities: A survey," *International journal of web and grid services*, Vol. 14, Nr. 4, S. 352–375, 2018.

[42] R. Zhang, R. Xue, und L. Liu, „Security and privacy on blockchain," *ACM Computing Surveys (CSUR)*, Vol. 52, Nr. 3, S. 1–34, 2019.

[43] O. Ali, A. Jaradat, A. Kulakli, und A. Abuhalimeh, „A comparative study: Blockchain technology utilization benefits, challenges and functionalities," *IEEE Access*, Vol. 9, S. 12 730–12 749, 2021.

[44] G. Zyskind, O. Nathan *et al.*, „Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*. IEEE, 2015, S. 180–184.

[45] J. H. Park und J. H. Park, „Blockchain security in cloud computing: Use cases, challenges, and solutions," *Symmetry*, Vol. 9, Nr. 8, S. 164, 2017.

[46] Bastiaan, Martijn, „Preventing the 51%-attack: a stochastic analysis of two phase proof of work in bitcoin," https://fmt.ewi.utwente.nl/media/175.pdf, 2015, [Online; abgerufen 28.11.2023].

[47] D. Hardt, „The OAuth 2.0 Authorization Framework," https://www.rfc-editor.org/info/rfc6749.txt, Oktober 2012, [Online; abgerufen 28.11.2023].

[48] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros, und C. Mortimore, „Openid connect core 1.0," *The OpenID Foundation*, 2014.

[49] J. V. Chandra, N. Challa, und S. K. Pasupuletti, „Authentication and authorization mechanism for cloud security," *International Journal of Engineering and Advanced Technology*, Vol. 8, Nr. 6, S. 2072–2078, 2019.

[50] N. Sakimura, J. Bradley, und N. Agarwal, „Proof Key for Code Exchange by OAuth Public Clients," https://www.rfc-editor.org/info/rfc7636.txt, September 2015, [Online; abgerufen 28.11.2023].

[51] Schadek, Robert, „Analysis and Development of Quorum Protocols for Real-World Network Topologies," PhD Thesis, Universität Oldenburg, 2021.

[52] A. Sharma und B. J. Singh, „Evolution of industrial revolutions: A review," *International Journal of Innovative Technology and Exploring Engineering*, Vol. 9, Nr. 11, S. 66–73, 2020.

[53] A. Toynbee und B. Jowett, *Lectures on the Industrial Revolution in England: Popular Addresses, Notes and Other Fragments*. Rivingtons, 1884. [Online]. Verfügbar: https://books.google.ch/books?id=APxUr3FCuccC

[54] J. U. Nef, „The Progress of Technology and the Growth of Large-Scale Industry in Great Britain, 1540-1640," *The Economic History Review*, Vol. 5, Nr. 1, S. 3–24, 1934. [Online]. Verfügbar: http://www.jstor.org/stable/2589915

[55] P. M. Deane, *The First Industrial Revolution*. Cambridge University Press, 1979.

[56] J. M. Wilson, „Henry Ford vs. assembly line balancing," *International Journal of Production Research*, Vol. 52, Nr. 3, S. 757–765, 2014.

[57] Thomas G. Roberts, „History of the NASA Budget," https://aerospace.csis.org/data/history-nasa-budget/, 2021, [Online; abgerufen 24.01.2021].

[58] A. Act, „Occupational safety and health act of 1970," *Public Law*, Vol. 91, S. 596, 1970.

[59] „Functional safety of electrical/electronic/programmable electronic safety-related systems - Part 1 to 7," International Electrotechnical Commission (IEC), Geneva, CH, Standard IEC 61508 1-7:2010, 2010. [Online]. Verfügbar: https://www.vde-verlag.de/iec-normen/217177/iec-61508-1-2010.html

[60] „Functional safety - Safety instrumented systems for the process industry sector - ALL PARTS," International Electrotechnical Commission

(IEC), Geneva, CH, Standard IEC 61511 ALL PARTS:2020, 2020. [Online]. Verfügbar: https://www.vde-verlag.de/iec-normen/211436/iec-61511-2020-ser.html

[61] „Safety of machinery - Functional safety of safety-related electrical, electronic and programmable electronic control systems," International Electrotechnical Commission (IEC), Geneva, CH, Standard IEC 62061:2005+AMD1:2012+AMD2:2015, 2015. [Online]. Verfügbar: https://www.vde-verlag.de/iec-normen/221921/iec-62061-2005-amd1-2012-amd2-2015-csv.html

[62] Bundesamt für Sicherheit in der Informationstechnik (BSI), „Cyber-Sicherheit," https://www.allianz-fuer-cybersicherheit.de/Webs/ACS/DE/Informationen-und-Empfehlungen/Informationen-und-weiterfuehrende-Angebote/Glossar-der-Cyber-Sicherheit/Functions/glossar.html?cms_lv2=132798, 2022, [Online; abgerufen 13.07.2022].

[63] Karl-Heinz Niemann und ABB AG, „Abgrenzungen der IT-Sicherheitsnormenreihen ISO 27000 und IEC 62443," https://library.e.abb.com/public/05167520631b4451b2c9c9cd2df4cb24/3ADR010839_Differentiation_ISO_27001_IEC_62443_REV_C_de_DE.pdf, 2022, [Online; abgerufen 13.07.2022].

[64] Centraal Bureau voor de Statistiek, „Most used cyber security protection measures among manufacturing companies in the Netherlands as of 2018," https://www.statista.com/statistics/1084009/cybersecurity-measures-among-manufacturing-companies-in-the-netherlands/, 2018, [Online; abgerufen 14.07.2022].

[65] IMDA, „Leading cybersecurity measures that were taken by enterprises in Singapore in 2019," https://www.statista.com/statistics/1180696/singapore-leading-cybersecurity-measures-taken-by-enterprises/, 2019, [Online; abgerufen 14.07.2022].

[66] CERT Polska, „Number of cyber security incidents handled by CERT* in Poland from 1996 to 2020," https://www.statista.com/statistics/1028557/poland-cybersecurity-incidents/, 2020, [Online; abgerufen 14.07.2022].

[67] Identity Theft Resource Center, „Annual number of data breaches and exposed records in the United States from 2005 to 2020," https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed/, 2020, [Online; abgerufen 14.07.2022].

[68] IBM, „Distribution of cyber attacks across leading industries worldwide in 2021," https://www.statista.com/statistics/1315805/cyber-attacks-top-industries-worldwide/, 2022, [Online; abgerufen 15.07.2022].

[69] Hiscox, „Average cost of all cyber attacks and largest single cyber attack to European firms in 2019," https://www.statista.com/statistics/1008178/european-firms-cyberattack-target-cost/, 2019, [Online; abgerufen 15.07.2022].

[70] CVE Details, „Number of common IT security vulnerabilities and exposures (CVEs) worldwide from 1999 to 2022," https://www.cvedetails.com/browse-by-date.php, 2022, [Online; abgerufen 24.07.2022].

[71] T. Miller, A. Staves, S. Maesschalck, M. Sturdee, und B. Green, „Looking back to look forward: Lessons learnt from cyber-attacks on Industrial Control Systems," *International Journal of Critical Infrastructure Protection*, Vol. 35, 2021.

[72] A. Greenberg, „The untold story of NotPetya, the most devastating cyberattack in history," *Wired, August*, Vol. 22, 2018.

[73] K. Wallis, M. Hüffmeyer, A. S. Koca, und C. Reich, „Access Rules Enhanced by Dynamic IIoT Context," in *Proceedings of the 3rd International Conference on Internet of Things, Big Data and Security: March 19-21, 2018, in Funchal, Madeira, Portugal*, 2018, S. 204–211.

[74] K. Wallis, F. Schillinger, C. Reich, und C. Schindelhauer, „Safeguarding data integrity by cluster-based data validation network," in *2019 Third World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)*. IEEE, 2019, S. 78–86.

[75] K. Wallis, F. Schillinger, E. Backmund, C. Reich, und C. Schindelhauer, „Context-Aware Anomaly Detection for the Distributed Data Validation Network in Industry 4.0 Environments," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, S. 7–14.

[76] J. Burch und G. Grudnitski, *Information systems: Theory and practice*. John Wiley & Sons, Inc., 1989.

[77] C. Fox, A. Levitin, und T. Redman, „The notion of data and its quality dimensions," *Information processing & management*, Vol. 30, Nr. 1, S. 9–19, 1994.

[78] E. F. Codd, „Data models in database management," in *Proceedings of the 1980 workshop on Data abstraction, databases and conceptual modeling*, 1980, S. 112–114.

[79] D. C. Tsichritzis und F. H. Lochovsky, *Data models*. Prentice Hall Professional Technical Reference, 1982.

[80] V. Jayawardene, S. Sadiq, und M. Indulska, „An analysis of data quality dimensions," 2015.

[81] J. Yan, Y. Meng, L. Lu, und L. Li, „Industrial big data in an industry 4.0 environment: Challenges, schemes, and applications for predictive maintenance," *IEEE Access*, Vol. 5, S. 23 484–23 491, 2017.

[82] F. Sidi, P. H. S. Panahy, L. S. Affendey, M. A. Jabar, H. Ibrahim, und A. Mustapha, „Data quality: A survey of data quality dimensions," in *2012 International Conference on Information Retrieval & Knowledge Management*. IEEE, 2012, S. 300–304.

[83] C. Batini, C. Cappiello, C. Francalanci, und A. Maurino, „Methodologies for data quality assessment and improvement," *ACM computing surveys (CSUR)*, Vol. 41, Nr. 3, S. 1–52, 2009.

[84] A. Schmetz, T. H. Lee, M. Hoeren, M. Berger, S. Ehret, D. Zontar, S.-H. Min, S.-H. Ahn, und C. Brecher, „Evaluation of Industry 4.0 Data formats for Digital Twin of Optical Components," *International Journal of Precision Engineering and Manufacturing-Green Technology*, Vol. 7, Nr. 3, S. 573–584, Mar. 2020. [Online]. Verfügbar: https://doi.org/10.1007/s40684-020-00196-5

[85] Jirkovskỳ, Václav, „Semantic integration in the context of cyber-physical systems," PhD Thesis, Czech Technical University, 2017.

[86] TC 65/SC 65E - Devices and integration in enterprise systems, „ISO/IEC 62541: OPC Unified Architecture - Part 1 to 13," International Electrotechnical Commission (IEC), Technischer Report, 2010–2020. [Online]. Verfügbar: https://webstore.iec.ch/publication/68039

[87] S.-H. Leitner und W. Mahnke, „OPC UA–service-oriented architecture for industrial applications," *ABB Corporate Research Center*, Vol. 48, Nr. 61-66, S. 22, 2006.

[88] R. Draht und R. Drath, *Datenaustausch in der Anlagenplanung mit AutomationML*. Springer, 2010.

[89] TC 65/SC 65A - System aspects, „IEC 61512: Batch Control," International Electrotechnical Commission (IEC), Standard, August 1997. [Online]. Verfügbar: https://webstore.iec.ch/publication/5528

[90] „ANSI/MTC1.6-2021: MTConnect Standard," MTConnect Institute, Standard, Mai 2021.

[91] A. Vijayaraghavan, W. Sobel, A. Fox, D. Dornfeld, und P. Warndorf, „Improving machine tool interoperability using standardized interface protocols: MT connect," 2008.

[92] Kellogg, Gregg and Champin, Pierre-Antoine and Longley, Dave, „JSON-LD 1.1 A JSON-based Serialization for Linked Data." W3C, Technischer Report, 2019.

[93] R. Y. Wang und D. M. Strong, „Beyond accuracy: What data quality means to data consumers," *Journal of management information systems*, Vol. 12, Nr. 4, S. 5–33, 1996.

[94] A. Karkouch, H. Mousannif, H. Al Moatassime, und T. Noel, „Data quality in internet of things: A state-of-the-art survey," *Journal of Network and Computer Applications*, Vol. 73, S. 57–81, 2016.

[95] M. Bovee, R. P. Srivastava, und B. Mak, „A conceptual framework and belief-function approach to assessing overall information quality," *International journal of intelligent systems*, Vol. 18, Nr. 1, S. 51–74, 2003.

[96] B. K. Kahn, D. M. Strong, und R. Y. Wang, „Information Quality Benchmarks: Product and Service Performance," *Commun. ACM*, Vol. 45, Nr. 4, S. 184–192, April 2002. [Online]. Verfügbar: https://doi.org/10.1145/505248.506007

[97] T. C. Redman, *Data quality for the information age*. Artech House, Inc., 1997.

[98] L. L. Pipino, Y. W. Lee, und R. Y. Wang, „Data quality assessment," *Communications of the ACM*, Vol. 45, Nr. 4, S. 211–218, 2002.

[99] M. Heravizadeh, J. Mendling, und M. Rosemann, „Dimensions of business processes quality (QoBP)," in *International Conference on Business Process Management*. Springer, 2008, S. 80–91.

[100] D. McGilvray, *Executing data quality projects: Ten steps to quality data and trusted information (TM)*. Academic Press, 2021.

[101] J. Liu, J. Li, W. Li, und J. Wu, „Rethinking big data: A review on the data quality and usage issues," *ISPRS journal of photogrammetry and remote sensing*, Vol. 115, S. 134–142, 2016.

[102] M. J. Khoury und J. P. A. Ioannidis, „Big data meets public health," *Science*, Vol. 346, Nr. 6213, S. 1054–1055, 2014. [Online]. Verfügbar: https://www.science.org/doi/abs/10.1126/science.aaa2709

[103] C. Xie, J. Gao, und C. Tao, „Big data validation case study," in *2017 IEEE third international conference on big data computing service and applications (BigDataService)*. IEEE, 2017, S. 281–286.

[104] J. Gao, C. Xie, und C. Tao, „Big Data Validation and Quality Assurance–Issuses, Challenges, and Needs," in *2016 IEEE symposium on service-oriented system engineering (SOSE)*. IEEE, 2016, S. 433–441.

[105] K. Wallis, C. Reich, B. Varghese, und C. Schindelhauer, „QUDOS: quorum-based cloud-edge distributed DNNs for security enhanced industry 4.0," in *Proceedings of the 14th IEEE/ACM International Conference on Utility and Cloud Computing*, 2021, S. 1–10.

[106] M. B. Kamel, K. Wallis, P. Ligeti, und C. Reich, „Distributed Data Validation Network in IoT: A Decentralized Validator Selection Model," in *Proceedings of the 10th International Conference on the Internet of Things*, 2020, S. 1–8.

[107] P. Neto, J. N. Pires, und A. P. Moreira, „Accelerometer-based control of an industrial robotic arm," in *RO-MAN 2009 - The 18th IEEE International Symposium on Robot and Human Interactive Communication*, 2009, S. 1192–1197.

[108] A. M. Almassri, W. Z. Wan Hasan, S. A. Ahmad, A. J. Ishak, A. M. Ghazali, D. N. Talib, und C. Wada, „Pressure Sensor: State of the Art, Design, and Application for Robotic Hand," *Journal of Sensors*, Vol. 2015, 2015. [Online]. Verfügbar: https://doi.org/10.1155/2015/846487

[109] A. Yadav, „Classification and Applications of Humidity Sensors: A Review," *International Journal for Research in Applied Science and Engineering Technology*, Vol. 6, Nr. 4, S. 3686–3699, April 2018. [Online]. Verfügbar: https://doi.org/10.22214/ijraset.2018.4616

[110] A. Flammini, P. Ferrari, E. Sisinni, D. Marioli, und A. Taroni, „Sensor interfaces: from field-bus to Ethernet and Internet," *Sensors and Actuators A: Physical*, Vol. 101, Nr. 1-2, S. 194–202, 2002.

[111] „Industrial communication networks - Fieldbus specifications – Part 1: Overview and guidance for the IEC 61158 and IEC 61784 series-1," International Electrotechnical Commission (IEC), Geneva, CH, Standard IEC 61158-1:2019, 2019. [Online]. Verfügbar: https://webstore.iec.ch/publication/59890#

[112] „IEEE Standard for Ethernet," *IEEE Std 802.3-2018 (Revision of IEEE Std 802.3-2015)*, S. 1–5600, 2018.

[113] T. Lojka, M. Bundzel, und I. Zolotová, „Industrial Gateway for Data Acquisition and Remote Control," 2015.

[114] S. Gupta, „An edge-computing based Industrial Gateway for Industry 4.0 using ARM TrustZone technology," *Journal of Industrial Information Integration*, Vol. 33, S. 100441, 2023.

[115] F. Fraile, T. Tagawa, R. Poler, und A. Ortiz, „Trustworthy industrial IoT gateways for interoperability platforms and ecosystems," *IEEE Internet of Things Journal*, Vol. 5, Nr. 6, S. 4506–4514, 2018.

[116] HashiCorp, Inc., „Identity-based networking," https://www.consul.io/, 2022, [Online; abgerufen 29.12.2022].

[117] Elias Backmund, „Secure Routing for Data Validation Networks," Bachelor Thesis, Hochschule Furtwangen University, 2019, Nicht veröffentlicht.

[118] K. Wallis, M. Merzinger, C. Reich, und C. Schindelhauer, „A Security Model based Authorization Concept for OPC Unified Architecture," in *Proceedings of the 10th International Conference on Advances in Information Technology*, 2018, S. 1–8.

[119] C. J. Fidge, „Timestamps in message-passing systems that preserve the partial ordering," *Proceedings of the 11th Australian Computer*

Science Conference, Vol. 10, Nr. 1, S. 56–66, 1988. [Online]. Verfügbar: http://sky.scitech.qut.edu.au/~fidgec/Publications/fidge88a.pdf

[120] F. Mattern, „Virtual Time and Global States of Distributed Systems," in *PARALLEL AND DISTRIBUTED ALGORITHMS*. North-Holland, 1988, S. 215–226.

[121] W. Zhao und M. Babi, „Byzantine fault tolerant collaborative editing," in *IET International Conference on Information and Communications Technologies (IETICT 2013)*, April 2013, S. 233–240.

[122] E. M. Ould-Ahmed-Vall, D. M. Blough, B. S. Heck, und G. F. Riley, „Distributed unique global ID assignment for sensor networks," in *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, 2005.*, November 2005.

[123] J. Lin, Y. Liu, und L. M. Ni, „SIDA: Self-organized ID Assignment in Wireless Sensor Networks," in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, Oktober 2007, S. 1–8.

[124] D. Mingxiao, M. Xiaofeng, Z. Zhe, W. Xiangwei, und C. Qijun, „A review on consensus algorithm of blockchain," in *2017 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE, 2017, S. 2567–2572.

[125] L. M. Bach, B. Mihaljevic, und M. Zagar, „Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018, S. 1545–1550.

[126] S. Fahim, S. Rahman, und S. Mahmood, „Blockchain: A Comparative Study of Consensus Algorithms PoW, PoS, PoA, PoV," *Int. J. Math. Sci. Comput*, Vol. 3, S. 46–57, 2023.

[127] N. Lasla, L. Al-Sahan, M. Abdallah, und M. Younis, „Green-PoW: An energy-efficient blockchain Proof-of-Work consensus algorithm," *Computer Networks*, Vol. 214, S. 109–118, 2022.

[128] K. Wallis, F. Schillinger, C. Reich, und C. Schindelhauer, „Corrupted Nodes and their Impact on a Distributed Decision Tree," in *World Congress on Internet Security (WorldCIS-2020)*. http://dx.doi.org/10.20533/WorldCIS.2020.0001, 2020.

[129] ——, „Protection Measurements for Distributed Decision Trees," in *International Journal for Information Security Research (IJISR)*, 2021.

[130] M. Tatam, B. Shanmugam, S. Azam, und K. Kannoorpatti, „A review of threat modelling approaches for APT-style attacks," *Heliyon*, Vol. 7, Nr. 1, 2021.

[131] Y.-G. Kim und S. Cha, „Threat scenario-based security risk analysis using use case modeling in information systems," *Security and Communication Networks*, Vol. 5, Nr. 3, S. 293–300, 2012.

[132] M. Rocchetto und N. O. Tippenhauer, „On attacker models and profiles for cyber-physical systems," in *European Symposium on Research in Computer Security*. Springer, 2016, S. 427–449.

[133] A. Sachitano, R. O. Chapman, und J. Hamilton, „Security in software architecture: a case study," in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.* IEEE, 2004, S. 370–376.

[134] L. Futcher und R. von Solms, „Guidelines for Secure Software Development," in *Proceedings of the 2008 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries: Riding the Wave of Technology*, Serie SAICSIT '08. New York, NY, USA: Association for Computing Machinery, 2008, S. 56–65. [Online]. Verfügbar: https://doi.org/10.1145/1456659.1456667

[135] M. Howard und S. Lipner, *The security development lifecycle*. Microsoft Press Redmond, 2006, Vol. 8.

[136] J. R. Douceur, „The Sybil Attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Serie IPTPS '01. London, UK, UK: Springer-Verlag, 2002, S. 251–260. [Online]. Verfügbar: http://dl.acm.org/citation.cfm?id=646334.687813

[137] J. Newsome, E. Shi, D. Song, und A. Perrig, „The Sybil attack in sensor networks: analysis amp; defenses," in *Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004*, April 2004, S. 259–268.

[138] S. Abbas, M. Merabti, D. Llewellyn-Jones, und K. Kifayat, „Lightweight Sybil Attack Detection in MANETs," *IEEE Systems Journal*, Vol. 7, Nr. 2, S. 236–248, June 2013.

[139] M. A. Jan, P. Nanda, X. He, und R. P. Liu, „A Sybil Attack Detection Scheme for a Centralized Clustering-Based Hierarchical Network," in *2015 IEEE Trustcom/BigDataSE/ISPA*, Vol. 1, August 2015, S. 318–325.

[140] M. Blaze, G. Bleumer, und M. Strauss, „Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology — EUROCRYPT'98*, K. Nyberg, Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, S. 127–144.

[141] G. Ateniese, K. Fu, M. Green, und S. Hohenberger, „Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," *ACM Trans. Inf. Syst. Secur.*, Vol. 9, Nr. 1, S. 1–30, Februar 2006. [Online]. Verfügbar: http://doi.acm.org/10.1145/1127345.1127346

[142] Oneserve, „The impact machine downtime has on business: A peer to peer comparison," Oneserve, Technischer Report, Februar 2018. [Online]. Verfügbar: https://www.oneserve.co.uk/wp-content/uploads/2018/02/The-impact-machine-downtime-has-on-business.pdf

[143] Y. A. Younis, K. Kifayat, und M. Merabti, „An access control model for cloud computing," *Journal of Information Security and Applications*, Vol. 19, Nr. 1, S. 45–60, 2014. [Online]. Verfügbar: https://www.sciencedirect.com/science/article/pii/S2214212614000222

[144] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, und M. Malli, „Cyber-physical systems security: Limitations, issues and future trends," *Microprocessors and microsystems*, Vol. 77, 2020.

[145] The MITRE Corporation, „MITRE ATT&CK®," https://attack.mitre.org/, 2023, [Online; abgerufen 10.04.2023].

[146] C. Eckert, *IT-Sicherheit: Konzepte - Verfahren - Protokolle*. De Gruyter Oldenbourg, 2014. [Online]. Verfügbar: https://doi.org/10.1515/9783486859164

[147] ProSoft, „Angriffe durch manipulierte Hardware," https://www.prosoft.de/loesungen/software/angriffe-durch-manipulierte-hardware/, 2022, [Online; abgerufen 02.10.2022].

[148] Pilz GmbH & Co. KG, „Schutz vor Manipulation," https://www.pilz.com/de-DE/support/knowhow/law-standards-norms/manufacturer-machine-operators/manipulation-protection, 2022, [Online; abgerufen 02.10.2022].

[149] N. Akhtar und A. Mian, „Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, Vol. 6, S. 14 410–14 430, 2018.

[150] K. Wallis, F. Kemmer, E. Jastremskoj, und C. Reich, „Adaption of a Privilege Management Infrastructure (PMI) Approach to Industry 4.0," in *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 2017, S. 101–107.

[151] D. W. Chadwick und A. Otenko, „The PERMIS X. 509 role based privilege management infrastructure," in *Proceedings of the seventh ACM symposium on Access control models and technologies*, 2002, S. 135–140.

[152] D. Chadwick, A. Otenko, und E. Ball, „Role-based access control with X. 509 attribute certificates," *IEEE Internet Computing*, Vol. 7, Nr. 2, S. 62–69, 2003.

[153] Cooper, D. and Santesson, S. and Farrell, S. and Boeyen, S. and Housley, R. and Polk, W., „Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," DOI: 10.17487/rfc5280. [Online]. Verfügbar: https://www.rfc-editor.org/info/rfc5280

[154] S. Micali, „Scalable certificate validation and simplified pki management," *1st Annual PKI research workshop*, Vol. 15, 2002.

[155] Dawson, Ed and Lopez, Javier and Montenegro, Jose A. and Okamoto, Eiji, „A New Design of Privilege Management Infrastructure for Organizations Using Outsourced PKI," in *Proceedings of the 5th International Conference*

on Information Security, Serie ISC '02. London, UK, UK: Springer-Verlag, 2002, S. 136–149. [Online]. Verfügbar: http://dl.acm.org/citation.cfm?id=648026.744524

[156] S. Turner, S. Farrell, und R. Housley, „RFC5755: An Internet Attribute Certificate Profile for Authorization," https://www.rfc-editor.org/info/rfc5755, Januar 2010, [Online; abgerufen 28.11.2023].

[157] J. A. Buchmann, E. Karatsiolis, und A. Wiesmaier, *Introduction to Public Key Infrastructures*. Springer Berlin Heidelberg, DOI: 10.1007/978-3-642-40657-7. [Online]. Verfügbar: http://link.springer.com/10.1007/978-3-642-40657-7

[158] ISO/IEC 7816-8:2016, „ISO/IEC 7816-8:2016: Identification cards – Integrated circuit cards – Part 8: Commands and mechanisms for security operations," International Organization for Standardization, Geneva, CH, Standard, November 2016.

[159] I. Guyon und A. Elisseeff, „An Introduction to Variable and Feature Selection," *J. Mach. Learn. Res.*, Vol. 3, Nr. null, S. 1157–1182, März 2003.

[160] A. L. Blum und P. Langley, „Selection of Relevant Features and Examples in Machine Learning," *Artif. Intell.*, Vol. 97, Nr. 12, S. 245–271, Dezember 1997. [Online]. Verfügbar: https://doi.org/10.1016/S0004-3702(97)00063-5

[161] Z. M. Hira und D. F. Gillies, „A Review of Feature Selection and Feature Extraction Methods Applied on Microarray Data," *Advances in Bioinformatics*, Vol. 2015, S. 1–13, 2015. [Online]. Verfügbar: https://doi.org/10.1155/2015/198363

[162] Y. Saeys, I. n. Inza, und P. Larrañaga, „A Review of Feature Selection Techniques in Bioinformatics," *Bioinformatics*, Vol. 23, Nr. 19, S. 2507–2517, September 2007. [Online]. Verfügbar: https://doi.org/10.1093/bioinformatics/btm344

[163] A. Khraisat, I. Gondal, P. Vamplew, und J. Kamruzzaman, „Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, Vol. 2, Nr. 1, Juli 2019. [Online]. Verfügbar: https://doi.org/10.1186/s42400-019-0038-7

[164] J. Lee Rodgers und W. A. Nicewander, „Thirteen ways to look at the correlation coefficient," *The American Statistician*, Vol. 42, Nr. 1, S. 59–66, 1988.

[165] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, „Scikit-learn: Machine learning in Python," *Journal of machine learning research*, Vol. 12, Nr. Oktober, S. 2825–2830, 2011.

[166] W. McKinney *et al.*, „Data structures for statistical computing in python," in *Proceedings of the 9th Python in Science Conference*, Vol. 445. Austin, TX, 2010, S. 51–56.

[167] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, und T. E. Oliphant, „Array programming with NumPy," *Nature*, Vol. 585, S. 357–362, 2020.

[168] J. D. Hunter, „Matplotlib: A 2D graphics environment," *Computing in science & engineering*, Vol. 9, Nr. 3, S. 90–95, 2007.

[169] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. De Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, , Brian, C. Fonnesbeck, A. Lee, und A. Qalieh, „mwaskom/seaborn: v0.8.1 (September 2017)," https://zenodo.org/record/883859, 2017, [Online; abgerufen 04.06.2023]. [Online]. Verfügbar: https://zenodo.org/record/883859

[170] Y. Kang, J. Hauswald, C. Gao, A. Rovinski, T. Mudge, J. Mars, und L. Tang, „Neurosurgeon: Collaborative Intelligence Between the Cloud and Mobile Edge," in *Proceedings of the 22nd International Conference on Architectural Support for Programming Languages and Operating Systems*, 2017, S. 615–629.

[171] L. Lockhart, P. Harvey, P. Imai, P. Willis, und B. Varghese, „Scission: Performance-driven and Context-aware Cloud-Edge Distribution of Deep Neural Networks," in *2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2020, S. 257–268.

[172] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, und J. Zhang, „Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing," *Proceedings of the IEEE*, Vol. 107, Nr. 8, S. 1738–1762, 2019.

[173] Fitzgerald, Simon, „Design and Implementation of a Distributed Neural Network Platform Utilising Crowdsource Processing," PhD Thesis, University of Dublin, 2018.

[174] L. Breiman, J. Friedman, C. J. Stone, und R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[175] A. Desai und S. Chaudhary, „Distributed Decision Tree," in *Proceedings of the 9th Annual ACM India Conference*, 2016, S. 43–50.

[176] ——, „Distributed decision tree v. 2.0," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, S. 929–934.

[177] P.-N. Tan, M. Steinbach, und V. Kumar, *Introduction to data mining*. Pearson Education India, 2016.

[178] L. E. Raileanu und K. Stoffel, „Theoretical comparison between the gini

index and information gain criteria," *Annals of Mathematics and Artificial Intelligence*, Vol. 41, Nr. 1, S. 77–93, 2004.

[179] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[180] B. Hssina, A. Merbouha, H. Ezzikouri, und M. Erritali, „A comparative study of decision tree ID3 and C4. 5," *International Journal of Advanced Computer Science and Applications*, Vol. 4, Nr. 2, S. 13–19, 2014.

[181] F. Siddiqui, S. Zeadally, T. Kacem, und S. Fowler, „Zero configuration networking: Implementation, performance, and security," *Computers & electrical engineering*, Vol. 38, Nr. 5, S. 1129–1145, 2012.