Carl von Ossietzky
Universität
Oldenburg

Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften
Department für Informatik

ASSESS – Anomaliesensitive State Estimation mit Streaming Systemen in Smart Grids

Von der Fakultät für Informatik, Wirtschafts- und Rechtswissenschaften der Carl von Ossietzky Universität Oldenburg zur Erlangung des Grades und Titels

Doktor der Naturwissenschaften Dr. rer. nat.

angenommene Dissertation

von Herrn Michael Brand

geboren am 13. März 1984 in Oldenburg (Oldb)

Gutachter:

Prof. Dr. Sebastian Lehnhoff
Fakultät II – Informatik, Wirtschafts- und Rechtswissenschaften, Carl von Ossietzky Universität Oldenburg

Weitere Gutachter:

FH-Prof. DI Mag. Dr. Dominik Engel
Informationstechnik & System-Management, Fachhochschule Salzburg

Tag der Disputation: 07. September 2023

Zusammenfassung

Die State Estimation ist ein essenzieller Prozess für die Lagebilderkennung in Stromversorgungssystemen, bei dem die komplexen Spannungen an den Knotenpunkten im Stromversorgungssystem auf Basis von Messwerten geschätzt werden. Die Güte der Messwerte wird dabei traditionell durch eine sogenannte Bad Data Detection ermittelt, die auf Basis des State-Estimation-Ergebnisses erkennen kann, ob einzelne Messwerte von dem Netzmodell und den übrigen Messwerten abweichen. Die Datenakquise und somit auch die State Estimation sind aber durch die zunehmende Verzahnung von IT- und OT (Netzbetriebstechnik)-Systemen und der ebenfalls zunehmenden Verwendung standardisierter Protokolle vulnerabel für Angriffe und Fehler. Darüber hinaus kann eine Bad Data Detection nicht die Güte von Messwerten ermitteln, für die es keine Redundanzen im System gibt. Es stellt sich demnach die Frage, inwiefern die akquirierten Messwerte, das ermittelte State-Estimation-Ergebnis und das daraus abgeleitete Lagebild vertrauenswürdig sind. Der Begriff der „Vertrauenswürdigkeit“ wird an dieser Stelle verwendet, um die unterschiedlichen Bedrohungen für die Lagebilderkennung, wie z.B. Angriffe und Fehlfunktionen, zu adressieren.

Auf Grundlage der Hypothese, dass es für eine vertrauenswürdige Lagebilderkennung notwendig ist, die kontextabhängige und multivariate Vertrauenswürdigkeit der akquirierten Messwerte zu erfassen, behandelt die vorliegende Arbeit die Forschungsfrage, wie das multivariate Vertrauen in physische Messwerte in einem cyber-physischen Stromversorgungssystem modelliert, geschätzt und in eine Lagebilderkennung integriert werden kann.

Es werden vier Artefakte vorgestellt, die zur Beantwortung der Forschungsfrage, wie das multivariate Vertrauen in physische Messwerte in einem cyber-physischen Energiesystem modelliert, geschätzt und in eine Lagebilderkennung integriert werden kann, entwickelt wurden. Das erste Artefakt ist ein Vertrauensmodell für die Lagebildanalyse in cyber-physischen Stromversorgungssystemen, mit dem es möglich ist, das multivariate Vertrauen und die Vertrauenserhebung für u.a. Messwerte abzubilden. Das zweite Artefakt ist eine Integrationsplattform für Vertrauensschätzer. Auf Basis externer Informationen schätzen die einzelnen Vertrauensschätzer eine Vertrauenswahrscheinlichkeit für einen Messwert und ordnen diese einem oder

mehreren Aspekten, wie z.B. der Informationssicherheit, zu. Wohldefinierte Schnittstellen erlauben ein einfaches Hinzufügen neuer Schätzer. Das dritte Artefakt ist eine anomaliesensitive State Estimation, die die multivariate Vertrauenswürdigkeit der Messwerte berücksichtigt und auf deren Basis eine ebenfalls multivariate Vertrauenswürdigkeit der Zustandsvariablen schätzt. Das vierte Artefakt ist das Gesamtsystem, das u.a. die drei genannten Artefakte integriert. Das System heißt Anomaliesensitive State Estimation mit Streaming Systemen (ASSESS). Die technologische Grundlage von ASSESS stellt ein Datenstrommanagementsystem dar, um sowohl die Messwerte als auch die externen Informationen zur Vertrauensschätzung datengetrieben verarbeiten zu können. Das bedeutet, dass umgehend auf neue Messwerte oder vertrauensrelevante Informationen reagiert werden kann. Die in ASSESS durchgeführte State Estimation ist ebenfalls datengetrieben, um jederzeit ein möglichst aktuelles Lagebild zur Verfügung zu haben. Ein weiterer wichtiger Aspekt von ASSESS ist seine Flexibilität bezüglich der Eingangsdaten, d.h. der Anzahl an Datenquellen und verwendeter Protokolle, und der Einbindung von Vertrauensschätzern.

Der Mehrwert von ASSESS wird am Beispiel eines Cyberangriffs durch eine koordinierte Einspeisung falscher Messwerte demonstriert. Durch weitere Szenarien wird im Zuge der Evaluation die demonstrierte Korrelation zwischen dem multivariaten Vertrauen in die Zustandsvariablen und deren Kompromittiertheit bestätigt. Darüber hinaus wird ASSESS bezüglich seiner Flexibilität, Skalierbarkeit, technischen sowie Prozessinteroperabilität und Aktualität evaluiert, wobei sich die Aktualität auf die zeitliche Differenz zwischen Messwertakquise und Ergebnisbereitstellung bezieht. Die Evaluation zeigt, dass ASSESS für unterschiedliche Netze einsetzbar ist und verschiedene Vertrauensschätzer integriert werden können. Außerdem wird, u.a. durch den Einsatz von Protokollen nach dem Standard IEC 60870-5-104 und der Erweiterbarkeit um weitere Protokolle, die technische und Prozessinteroperabilität gezeigt.

Abstract

State estimation is an essential process for situation awareness in power systems, in which the complex voltages at the nodes in the power system are estimated on the basis of measurements. The quality of the measurements is traditionally determined by a so-called bad data detection, which can detect whether individual measurements match the network model and the other measurements on the basis of the state estimation result. However, data acquisition and thus also state estimation are vulnerable to cyber attacks and errors due to the increasing interconnection of IT and OT (network operation technology) and the likewise increasing use of standardized protocols. In addition, bad data detection cannot determine the quality of measurements without redundancy in the system. Accordingly, the question arises to what extent the acquired measurements, the determined state estimation result and the situational awareness, derived from it, are trustworthy. The term "trustworthiness" is used here to address the various threats to the situational awareness, such as cyber attacks and malfunctions.

Based on the hypothesis that it is necessary to assess the contextual and multivariate trustworthiness of acquired mearurements for a trustworthy situational awareness, this thesis addresses the research question of how to model and estimate the multivariate trust in phyiscal measurements in a cyber-physical power system and integrate it into situational awareness.

Four artifacts are presented, developed to answer the research question, how the multivariate trust in physical measurements in a cyber-physical energy system can be modeled, estimated and integrated into situational awareness. The first artifact is a trust model for situational awareness in cyber-physical power systems, which makes it possible to represent multivariate trust and its assessment for, among other aspects, measurements. The second artifact is an integration platform for trust estimators. Based on external information, individual trust estimators estimate a trust probability for a measurement and assign it to one or more aspects, such as information security. Well-defined interfaces allow effortless addition of new estimators. The third artifact is an anomaly-sensitive state estimation, which estimates the multivariate trustworthiness of the state variables based on the multivariate trustworthiness of the measurements. The fourth artifact is the overall system, which integrates the three artifacts mentioned above, among others. The system is called anomaly sensitive state estimation with streaming systems (ASSESS). The technological basis of ASSESS is a data stream management system to be able to process both the measurements and the external information for the trust assessment in a data-driven manner. This enables to react immediately to new measurements or trust-relevant

information. The state estimation performed in ASSESS is also data-driven to have the most up-to-date situational awareness available at all times. Another important aspect of ASSESS is its flexibility in terms of input data (number of data sources and protocols) and the integration of trust estimators.

The benefit of ASSESS is demonstrated by the example of a coordinated false data injection attack. Throughout further scenarios in the course of the evaluation, the demonstrated correlation between the multivariate trust in the state variables and their level of compromise is confirmed. Furthermore, ASSESS is evaluated regarding its flexibility, scalability, technical as well as process interoperability, and timeliness. Timeliness refers to the time difference between measurement acquisition and result provision. The evaluation shows that ASSESS can be used for different networks and different trust estimators can be integrated. Furthermore, the use of protocols according to the IEC 60870-5-104 standard and the extensibility to additional protocols, among others, demonstrate the technical and process interoperability.

Danksagungen

Eine Doktorarbeit bedeutet einen langen Weg mit vielen Höhen und Tiefen, viel Euphorie für die Forschung, einigen Sackgassen und auch einigen Momenten der Verzweiflung. Umso wichtiger ist es, Menschen im beruflichen und privaten Umfeld um sich zu haben, die einen dabei unterstützen. An dieser Stelle möchte ich mich bei all jenen bedanken, die mich auf diesem langen Weg begleitet haben.

Zunächst möchte ich den leider viel zu früh verstorbenen Hans-Jürgen Appelrath (HJA) erwähnen. In der von ihm geleiteten Abteilung für Informationssysteme an der Carl von Ossietzky Universität Oldenburg habe ich nach dem Studium meine wissenschaftliche Laufbahn und die Suche nach einem Promotionsthema begonnen. Ich werde nie vergessen, wie HJA mich aufgrund seines bevorstehenden Todes an meinen späteren Doktorvater Sebastian Lehnhoff übergeben hat. Dies ist nur ein kleines Beispiel für die Größe, die er besaß.

Natürlich möchte ich mich auch bei allen damaligen Kolleg:innen in der Abteilung bedanken. Diese Zeit war stets geprägt von langen, kreativen aber auch produktiven Diskussionen im Bereich der Datenstromverarbeitung, in dem die meisten von uns geforscht haben. Allen voran möchte ich mich bei Marco Grawunder, dem „Vater“ von Odysseus, bedanken. Durch dich, lieber Marco, bin ich schon während meines Studiums auf das Thema Datenstromverarbeitung aufmerksam geworden und habe es lieben gelernt in dem Bereich zu forschen.

Nach meiner Zeit in der Abteilung wechselte ich in die Energieinformatik und bald darauf auch teilweise ans OFFIS. Sebastian Lehnhoff half mir damals sehr dabei, ein Thema in dem für mich zum damaligen Zeitpunkt nur oberflächlich bekannten Themenbereich der Energieinformatik zu finden. Wir beide hatten die Idee, meine Expertise aus dem Bereich der Datenstromverarbeitung zur Lösung von Fragestellungen in der Energieinformatik anzuwenden. Einen großen Baustein dieser Bemühungen stellt diese Dissertation dar.

Bei Sebastian Lehnhoff und Dominik Engel möchte ich mich auch für die Betreuung in den letzten Jahren bedanken. Ihr beide hattet stets ein offenes Ohr und die vielen Diskussionen mit euch haben meine Arbeit immer wieder vorangetrieben.

In meiner Zeit am OFFIS habe ich in unterschiedlichen Gruppen und Forschungsprojekten mit einer Vielzahl von Kolleg:innen arbeiten dürfen. Viele davon haben mich, vor allem durch ihre Expertise im Energiebereich, sehr unterstützt. Dafür bin ich euch allen ausgesprochen dankbar. Hervorheben möchte ich an dieser Stelle die Gruppe „Resilient Monitoring and Control (ROC)“ im Bereich Energie am OFFIS, der ich, mit einer kurzen Unterbrechung, bereits einige Jahre angehöre. Ohne eure fachliche und auch freundschaftliche Unterstützung wäre mein Weg mit Sicherheit noch deutlich länger und anstrengender geworden. Insbesondere die Diskussionen mit dir, lieber Anand, haben mir sehr geholfen und ich bin froh, dass wir an der Schnittstelle unserer beiden Doktorarbeiten zusammen forschen können.

Die meiste und sicherlich auch wichtigste Unterstützung habe ich von meinen Eltern, Christa und Heinz, und meiner Lebensgefährtin Christin erfahren. Ich danke euch von Herzen für eure Unterstützung und Geduld mit mir in den letzten Jahren. Und ohne dich, lieber Heinz, hätte ich gar nicht erst mein Informatikstudium aufnehmen können. Auch insbesondere dir, meine liebe Christin, danke ich von ganzem Herzen. Es war in einigen Phasen der Doktorarbeit sicher nicht immer leicht mit mir. Auch danke ich dir sehr für das Lektorieren dieser Arbeit.

Oldenburg, den 12. Oktober 2023

Michael Brand

Inhaltsverzeichnis

1 Einleitung

> *Was wir wissen, ist ein Tropfen, was wir nicht wissen, ein Ozean.*
>
> **— Sir Isaac Newton**

Dieses Kapitel führt in die vorliegende Arbeit ein, indem in Abschnitt 1.1 zunächst die Problemstellung herausgearbeitet wird. Die daraus abgeleitete Forschungsfrage wird zusammen mit Forschungszielen und nichtfunktionalen Anforderungen in Abschnitt 1.2 formuliert und erläutert. In Abschnitt 1.3 folgt die Vorstellung der Vorgehensweise zur Beantwortung der Forschungsfrage. Das Kapitel endet mit einer Zusammenfassung und Informationen über die restliche Struktur der restlichen Arbeit in Abschnitt 1.4.

1.1 Motivation

Anmerkung: Große Teile der folgenden Motivation sind ebenfalls in [Bra+19b; Bra+20; BBL21; BEL23] veröffentlicht.

Operateuren, die in Leitwarten von Stromnetzbetreibern arbeiten, kommt eine wichtige Aufgabe zu. Sie müssen, trotz sich verändernder Bedingungen während des operativen Alltags, einen normalen, sicheren Zustand für das Stromnetz als Gesamtsystem bewahren. Die Steuerungen, die ein Operateur dabei durchführt, basieren auf einer umfangreichen Lagebilderkennung des elektrotechnischen Systems. Ein Beispiel dafür ist die automatische Überprüfung aller Spannungsniveaus, auf Basis derer ein Operateur gegebenenfalls reagiert.

Eine Grundlage für eine Lagebilderkennung stellt die State Estimation (deutsch: Zustandsschätzung[1]) dar. Die Aufgabe einer State Estimation in einem Stromversorgungssystem[2] ist es, diejenigen physikalischen Größen in dem System zu schätzen, mit denen in Kombination mit einem Netzmodell das gesamte System vollständig

[1] In dieser Arbeit wird der englische Fachbegriff State Estimation verwendet.
[2] In dieser Arbeit werden Stromversorgungssysteme kurz Stromsysteme genannt. Außerdem werden die Begriffe Stromnetz und -system synonym verwendet.

beschrieben werden kann [AE04]. Diese zu schätzenden Größen sind die komplexen Spannungen an allen Netzwerkknoten. Die Schätzung basiert auf redundanten Messwerten und einem angenommenen Netzmodell. Des Weiteren wird bei der State Estimation berücksichtigt, dass einzelne Messwerte fehlerhaft sein können. Der Prozess, solche zufallsverteilten, fehlerhaften Messwerte zu erkennen, zu identifizieren und herauszurechnen wird Bad Data Detection (deutsch: Erkennung falscher Messwerte[3]) genannt [AE04].

Im Vergleich zu traditionellen Stromsystemen sind heutige und vor allem zukünftige Smart Grids deutlich stärker mit Informations- und Kommunikationstechnik (IKT) verwoben. Man nennt solche Systeme auch cyber-physische Energiesysteme (CPESs). Dies schafft auf der einen Seite notwendige Voraussetzungen, um Stromsysteme mit viel dezentraler Erzeugung überwachen und steuern zu können [Pan13]. Auf der anderen Seite resultieren aus ihnen eine erhöhte Komplexität und gegenseitige Abhängigkeiten [Pan13], sowie eine gesteigerte Gefahr durch Softwarefehler und Cyberangriffe [PB14]. Im Falle von Softwarefehlern ist es in anderen Domänen ein gängiges Vorgehen das System herunterzufahren, neu zu konfigurieren und anschließend neu zu starten. Dieses Vorgehen ist allerdings für Stromsysteme nicht anwendbar, da es sich um kritische Infrastrukturen handelt und die Stromversorgung zu jeder Zeit garantiert werden muss [PB14]. Demzufolge ist der Zustand der IKT-Systeme als eine zusätzliche Grundlage für die State Estimation anzusehen.

Abbildung 1.1 stellt Beispiele aus der Literatur für Bedrohungen für CPESs und insbesondere die State Estimation dar, wobei die Liste keinen Anspruch auf Vollständigkeit erhebt. Grundsätzlich existieren Bedrohungen durch Angriffe, natürliche Phänomene und Fehlfunktionen. Die Auswirkungen, rechts in Abbildung 1.1, lassen sich mit Hinblick auf eine State Estimation in Asset- sowie Kommunikationsausfälle, Kommunikationsverzögerungen und kompromittierte Prozessdaten kategorisieren. Als ein Assetausfall wird in dieser Arbeit der Ausfall eines Gerätes der Fernwirktechnik bezeichnet. Dies kann u.a. durch einen physischen Angriff auf das Gerät [Xin20], durch eine Überlastung des Gerätes [Xin20] oder durch Softwarefehler [KSZ13] geschehen. Weitere mögliche Ursachen für einen Assetausfall sind natürliche Phänomene wie beispielsweise Temperaturschwankungen oder extreme Wetterereignisse [Hum+17; Xin20].

Ein Kommunikationsausfall bedeutet, dass die Prozessdaten nicht aus dem Feld ins Kontrollzentrum oder andersherum kommuniziert werden können. Ursachen dafür können Denial-of-Service- [GH15; Hum+17; WS18; Cai+19; DFD19; MHB19; AA20; FBY20; GD20; Kar+20; Xin20] oder Wurmlochangriffe [AA20; GD20] sein.

[3] In dieser Arbeit wird der englische Fachbegriff Bad Data Detection verwendet.

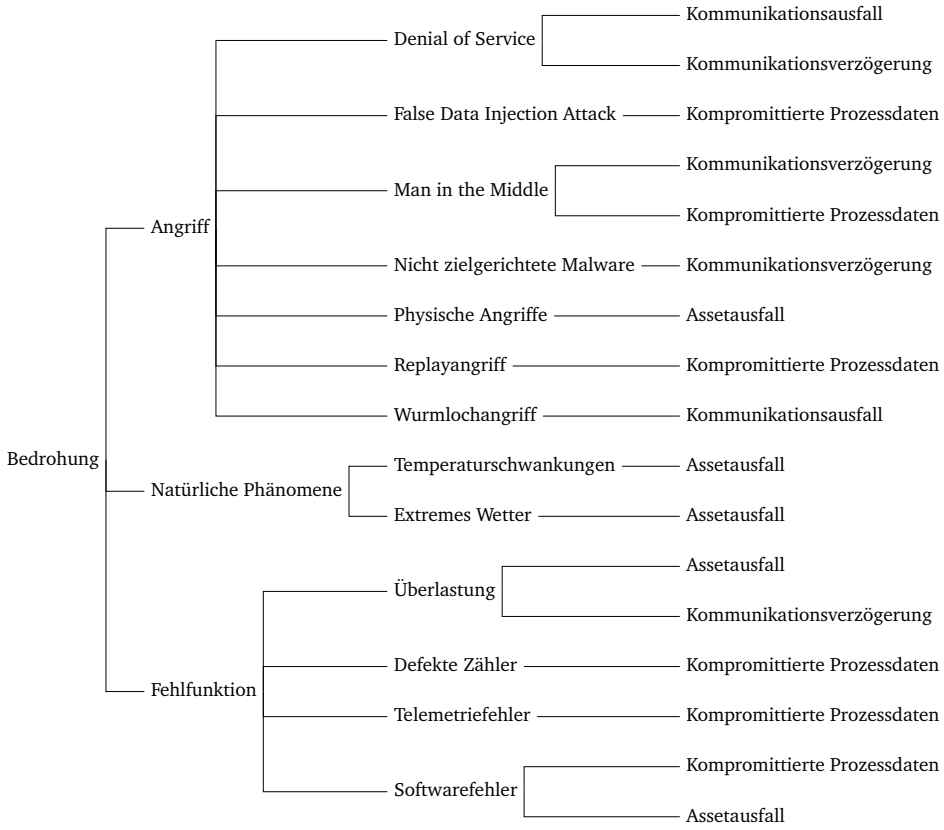


Abb. 1.1.: Beispiele für Bedrohungen für cyber-physische Energiesysteme und insbesondere die State Estimation.

Bei einem Denial-of-Service-Angriff wird die Kommunikationsschnittstelle eines IKT-Gerätes durch das Senden von Unmengen von Datenpaketen an diese Schnittstelle stark beeinträchtigt oder vollständig blockiert [WS18]. Die Idee eines Wurmlochangriffs ist es, Routinganfragen von einem Gerät zu beantworten, das unter der Kontrolle von Angreifern ist. Dadurch können Angreifer verhindern, dass Datenpakete ihr gewünschtes Ziel erreichen [AA20].

Neben Kommunikationsausfällen kann es noch zu Kommunikationsverzögerungen kommen, d.h. die Übertragung von Prozessdaten ist langsamer als gewöhnlich. Dies kann bei zeitkritischen Anwendungen zu großen Problemen führen. Eine Kommunikationsverzögerung kann u.a. durch Denial-of-Service- [GH15; Hum+17; WS18; Cai+19; DFD19; MHB19; AA20; FBY20; GD20; Kar+20; Xin20] oder Man-in-the-Middle-Angriffe [AA20; DFD19; FBY20; GD20], durch den Einsatz nicht zielgerichteter Malware [Hum+17] sowie durch eine Überlastung des Gerätes [Xin20] hervorgerufen werden. Bei einem Man-in-the-Middle-Angriff wird die Kommunikation zwischen zwei Endpunkten zu einem dritten Gerät, das unter der Kontrolle der Angreifer ist, umgeleitet. Die Angreifer können dann z.B. Datenpakete verwerfen

(Wurmlochangriff) oder sie manipulieren und anschließend an das ursprüngliche Ziel weiterleiten [DFD19]. Nicht zielgerichtete Malware ist Malware, die sich im IKT-System befindet, zwar nicht den ausdrücklichen Zweck eines Angriffs auf das Stromsystem hat, aber dennoch die Kommunikation beeinträchtigt [Hum+17].

Die letzte in Abbildung 1.1 dargestellte Auswirkung stellen kompromittierte Prozessdaten, z.B. kompromittierte Messwerte, dar. Ursachen dafür können Manipulationen von Topologie- oder Messdaten sein [Cai+19; Deh+19; FBY20; GD20; Xin20]. Darüber hinaus können durch Replayangriffe zuvor aufgezeichnete Prozessdaten erneut in das IKT-System gespeist werden, um Operateure in die Irre zu führen [GH15; WS18; DFD19; MHB19; AA20; GD20; Kar+20]. Koordinierte Cyberangriffe stellen eine besondere Bedrohung dar. Die Autoren der NISTIR-7628-Richtlinien stellen fest, dass (aus dem Englischen übersetzt, Satzbau angepasst) „es klar ist, dass Cyberangriffe oder kombinierte cyber-physische Angriffe eine signifikante Bedrohung für das Energienetz darstellen“ [PB14]. Liu et al. [LNR11] haben 2009 begonnen False Data Injection Attacks (FDIAs) (deutsch: Einspeisung falscher Daten[4]) auf State Estimatoren zu untersuchen. Sie haben gezeigt, dass es unter der Bedingung, dass die Angreifer mehrere Messgeräte unter ihrer Kontrolle und Kenntnis über das elektrotechnische Modell des Stromsystems (Netztopologie und Leitungsimpedanzen) haben, möglich ist, den geschätzten Systemzustand unbemerkt zu beeinflussen und so den Operateuren einen falschen Systemzustand glauben zu machen. Etwaige Ziele können schädliche Kontrollaktionen oder das Verbergen schädlicher Manipulationen sein.

Zwar gibt es in der Literatur Lösungen für das Problem der FDIAs (z.B. [Cui+12; MS15; XN15]), diese funktionieren allerdings nicht notwendigerweise für alle Bedrohungsszenarien. Die meisten Lösungen setzen eine Messredundanz im Stromnetz voraus, die aber nicht in allen Netzen vorhanden ist. Vor allem seien hier Verteilnetze erwähnt, für welche in der Regel nicht genügend Messwerte für eine State Estimation vorliegen. Aber auch dort steigt mit der steigenden Komplexität der Stromsysteme auch die Notwendigkeit einer Lagebilderkennung [Hua+12]. In solchen Netzen ohne Messredundanz müssten kompromittierte Prozessdaten z.B. durch simulierte ersetzt werden.

Die Quintessenz ist, dass traditionelle Bad Data Detection nicht ausreicht, um koordinierte Angriffe in allen vorstellbaren Konstellationen zu erkennen. Darüber hinaus würde eine Absicherung aller Prozessdaten durch z.B. kryptografische Maßnahmen ebenfalls nicht ausreichen, da sie lediglich die Integrität der Prozessdaten sicherstellen würde. Dabei bedeutet Integrität, dass es Subjekten nicht möglich ist,

[4] In dieser Arbeit wird der englische Fachbegriff False Data Injection Attack verwendet.

zu schützende Daten unautorisiert und unbemerkt zu manipulieren [Eck14]. Die Prozessdaten können allerdings auch aus anderen Gründen fehlerhaft sein (siehe Abbildung 1.1). Die Informationssicherheit, die durch Cyberangriffe bedroht wird, ist nur ein Aspekt einer zuverlässigen und vertrauenswürdigen Lagebilderkennung. Neben der Frage, ob Messwerte manipuliert wurden, spielen unter anderem auch die folgenden Fragen eine bedeutende Rolle: Arbeiten alle (Mess-) Geräte korrekt? Sind sie zuverlässig? Kommt es zu Fehlern durch Wechselwirkungen im CPES? Aus diesen Feststellungen lassen sich die folgenden Bedrohungen bei der Arbeit mit Prozessvariablen bei der Lagebilderkennung in einem CPES im Vergleich zu einem früheren Stromsystem ableiten[5]:

- Das Risiko, dass die Integrität der Prozessvariablen verletzt wurde, ist höher.
- Die Möglichkeit, dass Fehler in einem vorgelagerten System die Korrektheit oder Genauigkeit der Prozessvariablen beeinflusst haben, ist wahrscheinlicher.
- Die Gefahr, dass entweder Dritte oder Fehler in vorgelagerten Systemen die Verfügbarkeit der Prozessvariablen verringert haben, ist größer.

Aber wie ist damit umzugehen, dass nicht definitiv festgestellt werden kann, ob eine Prozessvariable kompromittiert ist? Was ist zu tun, wenn es nur eine Wahrscheinlichkeit gibt, dass die Prozessvariable kompromittiert ist? Bei welchen Wahrscheinlichkeiten ist z.B. ein Ersetzen der Variablen sinnvoll? In Anbetracht solcher Fragestellungen ist ein umfassenderer Begriff als zum Beispiel die Integrität vonnöten, um die Güte von Prozessvariablen zu beschreiben. Dafür wurde weiter oben bereits der Begriff der Vertrauenswürdigkeit verwendet. Vertrauenswürdigkeit beschreibt eine Eigenschaft einer Entität, während Vertrauen, oder Trust[6], einer Entität entgegengebracht und für diese Arbeit wie folgt definiert wird:

Definition 1 (Trust)**.** *„Trust ist ein subjektives, kontextabhängiges und multivariates Empfinden gegenüber einer Entität bezüglich ihrer funktionalen Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit“ [Bra+20] (aus dem Englischen übersetzt)*[7].

Trust ist demnach subjektiv, immer in einem Kontext zu betrachten und zeichnet sich durch verschiedene Aspekte aus. In Bezug auf die Problematiken in modernen

[5] Aus den genannten Gründen weichen die Bedrohungen von den in der Informationssicherheit geläufigen Zielen der Vertraulichkeit (engl. confidentiality), Integrität und Verfügbarkeit (engl. availability), kurz CIA, ab.
[6] In dieser Arbeit wird für das technische Vertrauen der englische Fachbegriff Trust verwendet.
[7] Diese Definition basiert auf einer Definition von Trust im Bereich des Organic Computings [SR12].

CPESs wird die Hypothese aufgestellt, dass es für eine vertrauenswürdige Lagebilderkennung notwendig ist, den kontextabhängigen und multivariaten Trust in Prozessvariablen zu erfassen.

Ist es möglich, den Trust in Prozessvariablen zu erfassen, stellen sich weitere Fragen: Wie kann der Trust in eine Lagebilderkennung integriert werden? Wie kann ein möglichst vertrauenswürdiges Lagebild gewährleistet werden? Vor dem Hintergrund solcher Fragestellungen ist eine umfassendere Herangehensweise vonnöten, die auf Basis eines Trust-Modells die Möglichkeit eröffnet, Trust zu erfassen und in den Prozess der Lagebilderkennung einzubinden. Das Ergebnis ist zum einen eine neue Perspektive auf die Korrektheit und Zuverlässigkeit der Prozessvariablen sowie das Vertrauen in das Lagebild. Zum anderen ist es ein Schritt in die Richtung zuverlässigerer netzdienlicher Operationen, indem diese auf einer Trust-sensitiven Lagebilderkennung und einem möglichst vertrauenswürdigen Lagebild basieren. Eine solche Herangehensweise wird in dieser Arbeit vorgestellt.

1.2 Forschungsfrage

Die Fragen, die in der Motivation diskutiert wurden, werden in diesem Abschnitt in einer Forschungsfrage aufbereitet und zusammengefasst. Darüber hinaus werden Forschungsziele abgeleitet und nichtfunktionale Anforderungen identifiziert, die sich aus dem Umfeld des zu entwickelnden Systems ergeben. Die Forschungsfrage lautet wie folgt:

Forschungsfrage. *Wie kann der multivariate Trust in physische Messwerte in einem cyber-physischen Energiesystem modelliert, geschätzt und in eine Lagebilderkennung integriert werden?*

Das erste Forschungsziel stellt ein kontextsensitives, multivariates Trust-Modell dar, das Trust gemäß Definition 1 aus Abschnitt 1.1 begreift. Dies beinhaltet insbesondere die sechs Aspekte funktionale Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit. Auch wenn nicht jeder dieser Aspekte die gleiche Relevanz für eine Lagebilderkennung hat, so hat doch das Trust-Modell den Anspruch, allgemein für Trust in CPESs anwendbar zu sein.

Einen weiteren wesentlichen Aspekt zur Beantwortung der Forschungsfrage stellt eine Trust-Schätzung dar, die für einen Messwert den multivariaten Trust kontextabhängig einschätzt. Dies kann durch unterschiedliche Metriken oder Detektoren

geschehen, wobei nicht neue Detektoren entwickelt, sondern vornehmlich Detektoren aus der Fachliteratur integriert werden sollen. Somit ist die Bereitstellung einer Integrationsplattform für Trust-Schätzer das zweite Forschungsziel. Das Ergebnis sind mit komplexen, da multivariaten, Trust-Werten annotierte Messwerte.

Das dritte Forschungsziel ist die Durchführung einer Trust-sensitiven State Estimation zur Lagebilderkennung. Dabei sollte das Ergebnis der State Estimation den Trust in die Messwerte berücksichtigen. Letztere münden in einen ebenfalls multivariaten Trust in die Zustandsvariablen, also einer Aussage, dass eine Zustandsvariable ggf. nicht vertrauenswürdig ist und welche Aspekte betroffen sind.

Zusammenfassend soll die Forschungsfrage durch das Erreichen der folgenden Forschungsziele beantwortet werden:

1. Entwicklung eines kontextsensitiven, multivariaten Trust-Modells,
2. Entwicklung einer Integrationsplattform für Trust-Schätzer und
3. Entwicklung einer Trust-sensitiven State Estimation.

Neben den Forschungszielen, die als funktionale Anforderungen angesehen werden können, muss ein System, das diese Anforderungen erfüllt, auch nichtfunktionale Anforderungen erfüllen, die sich aus dem Umfeld von Stromsystemen ergeben.

Stand der Technik bei der State Estimation ist ein zyklischer Prozess, bei dem eine State Estimation z.B. alle fünf Minuten durchgeführt wird. Heutige und vor allem zukünftige CPESs unterliegen aber, vor allem durch vermehrt dezentrale und weniger vorhersag- bzw. steuerbare Erzeuger, stärkeren Dynamiken, wodurch die Aktualität der Lagebilderkennung und somit auch der State Estimation einen hohen Stellenwert bekommt. Darüber hinaus erfordern auch die, teilweise für die Domäne in der Tragweite neuen, Herausforderungen (vgl. Abbildung 1.1 in Abschnitt 1.1) eine sofortige Berücksichtigung möglicherweise kompromittierender Ereignisse.

Darüber hinaus ist sowohl eine technische als auch eine Prozessinteroperabilität erforderlich, damit ein System, das die funktionalen Anforderungen erfüllt, auch perspektivisch von Stromnetzbetreibern eingesetzt werden kann. Bei der technischen Interoperabilität stehen vor allem sogenannte Supervision, Control, and Data Acquisition (SCADA)-Systeme und die in ihrem Kontext verwendeten Kommunikations- und Datenprotokolle im Fokus, da der Prozess der Lagebilderkennung und auch jener der State Estimation fester Bestandteil von SCADA-Systemen sind. SCADA-Systeme sind „eine Sammlung von Gerätschaften, die einen Operateur mit genügend Informationen versorgt, um den Status eines bestimmten Gerätes oder Prozesses in einer Fernwirkstation zu bestimmen und Aktionen bzgl. dieses Gerätes oder Prozesses

zu veranlassen, ohne körperlich anwesend zu sein“ [TM15] (aus dem Englischen übersetzt).

Eine adäquate Prozessinteroperabilität sollte zwei Aspekte umfassen. Zum einen sollte es die Gewährleistung eines möglichst vertrauenswürdigen Lagebildes umfassen, um den Operateuren nicht nur ein Lagebild zur Verfügung zu stellen, das zwar Trust-sensitiv aber im Zweifel nicht vertrauenswürdig ist. Dies kann zum Beispiel durch das Bereitstellen eines alternatives Lagebildes, das zum Teil auf Pseudomesswerten beruht, geschehen, wenn das Trust-sensitive Lagebild als nicht vertrauenswürdig genug eingeschätzt wird. Zum anderen sollte ein Lagebild nur dann an Operateure oder weiterverarbeitende Systeme weitergeleitet werden, wenn es als wichtig eingestufte Änderungen am Systemzustand oder Trust gibt. Dies ist wichtig, da eine verbesserte Aktualität zu einer wesentlich erhöhten Updaterate führen kann, was zu einer Überforderung der Operateure oder einer geringeren Aufmerksamkeit für die Lagebildänderungen führen kann.

Ferner bewegt sich das zu entwickelnde System vor allem in den Forschungsfeldern von Stromsystemen als komplexe CPESs, Bedrohungen gegen sie und möglichen Gegenmaßnahmen. Diese Forschungsfelder sind sehr komplex und hochgradig dynamisch. Ziel sollte daher vielmehr ein technisches Rahmenwerk (engl. framework) sein, das als Integrationsplattform für z.B. zukünftige Metriken, Detektoren oder State-Estimation-Algorithmen dienen kann. Aus diesem Grund ist eine weitere nichtfunktionale Anforderung Flexibilität. Neben Flexibilität ist aber auch Skalierbarkeit vonnöten. Das zu entwickelnde System muss mit unterschiedlichen Stromnetzgrößen und Anzahlen von Trust-Schätzern umgehen können.

Ein System, das die Forschungsziele erfüllt, sollte also, um eine praktische Relevanz zu haben und zukunftsfähig zu sein, die folgenden nichtfunktionalen Anforderungen erfüllen:

1. Aktualität,
2. technische Interoperabilität,
3. Prozessinteroperabilität,
4. Flexibilität und
5. Skalierbarkeit.

1.3 Methodologie

Als Vorgehensweise zum wissenschaftlichen Arbeiten wird der Design Science Research Process [Pef+06] ausgewählt. Designwissenschaftliche Forschung ist eine Methode, die Forschung etabliert und operationalisiert, wenn das gewünschte Ziel ein Artefakt oder eine Empfehlung ist.

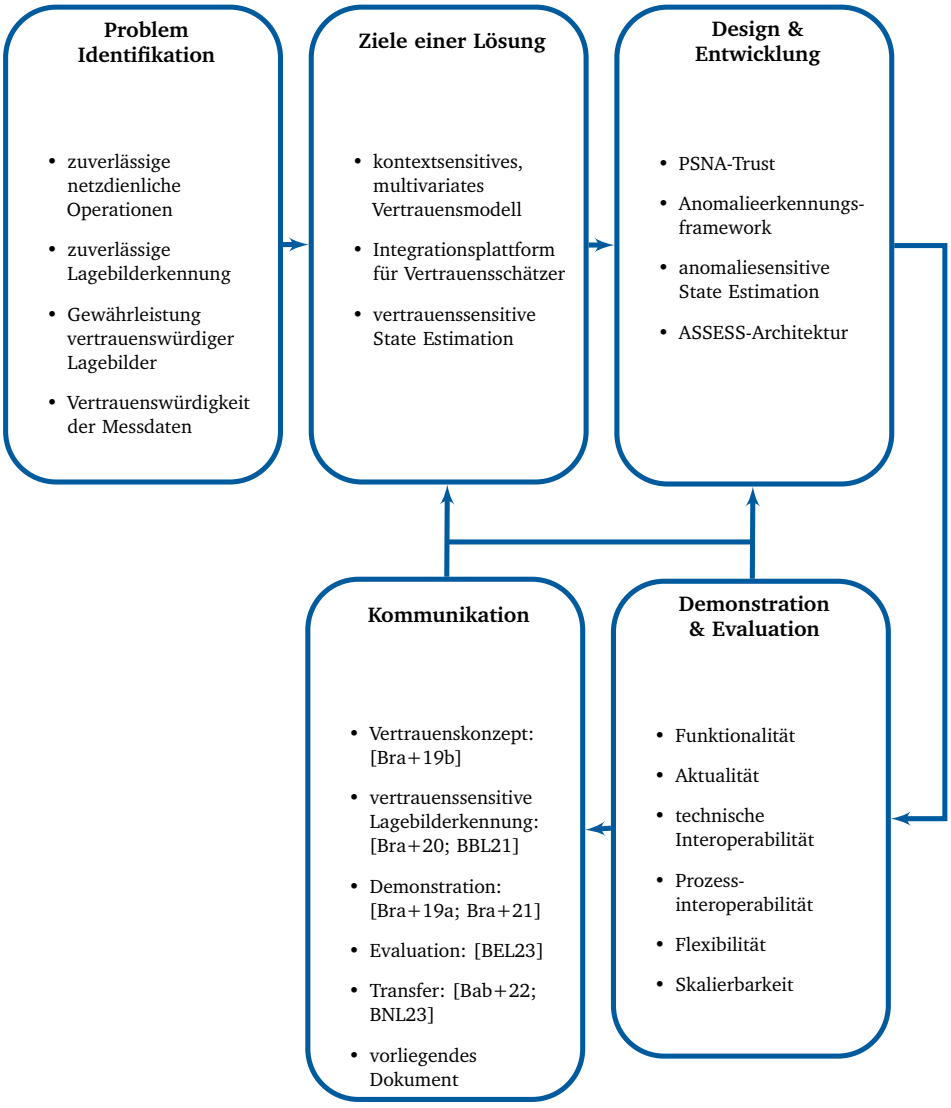


Abb. 1.2.: Das Vorgehen bei der vorliegenden Arbeit basierend auf dem Design Science Research Process [Pef+06].

Der Design Science Research Process ist entsprechend ein Prozess, um designwissenschaftliche Forschung zu operationalisieren [DLA15], und besteht aus sechs Phasen [Pef+06]: Problemidentifikation, Ziele einer Lösung, Design und Entwicklung, Demonstration, Evaluation, sowie Dissemination. Dabei kann der Einstiegspunkt

für eine wissenschaftliche Arbeit bei der Problemidentifikation, bei den Zielen einer Lösung oder bei dem Design und der Entwicklung liegen. Die Phasen bauen zwar auf einander auf, allerdings kann es vorkommen, dass Schleifen in dem Prozess vorgenommen werden müssen. Schleifen können dabei entweder von der Demonstration, der Evaluation oder der Kommunikation ausgehen und als Wiedereinstiegspunkte in den Prozess sind sowohl das Design und die Entwicklung als auch die Ziele einer Lösung möglich [Pef+06].

Abbildung 1.2 zeigt den Design Science Research Process, instantiiert für das wissenschaftliche Vorgehen bei dieser Arbeit. Der Einstiegspunkt in den Prozess ist dabei die Problemidentifikation. In den folgenden Unterabschnitten werden die Instanziierungen der einzelnen Prozessschritte beschrieben: Design und Entwicklung in Abschnitt 1.3.1, Demonstration und Evaluation in Abschnitt 1.3.2, sowie Kommunikation bzw. Dissemination in Abschnitt 1.3.3. Problemidentifikation und Ziele einer Lösung wurden bereits in den Abschnitten 1.1 und 1.2 ausführlich beschrieben.

1.3.1 Design und Entwicklung

In diesem Unterabschnitt werden die im Rahmen dieser Arbeit entwickelten Modelle, Konzepte und Systemkomponenten vorgestellt, um die Forschungsziele zu erreichen. Das erste Artefakt, ein Trust-Modell für die Lagebildanalyse in CPESs, Power System Network Assessment (PSNA)-Trust, stellt dabei die Basis dar. PSNA-Trust basiert dabei auf einem Trust-Modell aus dem Organic Computing (OC) namens OC-Trust [Ste+10; SR12] und modelliert Trust kontextsensitiv sowie multivariat, wie es in Definition 1 definiert ist.

Das zweite Artefakt ist ein Anomalieerkennungsframework (engl. anomaly detection framework) (ADF). Die Funktion des ADF ist es, unterschiedliche Anomaliedetektoren zur Trust-Schätzung zu integrieren. Die Anomalieerkennung stellt dabei eine allgemeine Form dar, Trust-Einbußen zu erkennen. Mit dem ADF wird die Flexibilität und Skalierbarkeit des Systems bzgl. verschiedenster Anomaliedetektoren gewährleistet. Die Schnittstelle zu den Detektoren wird dabei durch Datenmodelle definiert und die Ausgabe des ADF stellen Messwerte annotiert mit multivariaten Trust-Werten dar.

Eine anomaliesensitive State Estimation (ASSE), die die Trust-sensitive State Estimation umsetzt, stellt das dritte Artefakt dar. Als Eingabe erhält sie Messwerte, annotiert mit multivariaten Trust-Werten. Ihre Ausgabe sind geschätzte Zustandsvariablen, die komplexen Spannungen an den Netzknoten, ebenfalls annotiert mit multivariaten

Trust-Werten. Der Trust in die Zustandsvariablen berechnet sich dabei aus dem Trust in die Messwerte. Die ASSE arbeitet ferner ereignisgetrieben und nicht in festen Zyklen, um eine Aktualität zu gewährleisten.

Das vierte Artefakt ist die Gesamtarchitektur des Systems, das die oben beschriebenen Komponenten beinhaltet und im Folgenden vorgestellt wird. Der Name des Systems ist anomaliesensitive State Estimation mit Streaming Systemen (ASSESS). Während die ASSE die Kernkomponente des Systems ist, wird die Bedeutung der Streaming Systeme[8] nachfolgend erläutert. ASSESS umfasst allerdings nicht nur die in diesem Unterabschnitt beschriebenen Komponenten, sondern gewährleistet durch die Verwendung von Pseudomesswerten auch ein möglichst vertrauenswürdiges Lagebild. Ein Streaming System wird für diese Arbeit wie folgt definiert:

Definition 2. *Ein Streaming System ist ein System, das Daten ereignisbasiert und im Hauptspeicher verarbeitet. Dabei werden die Daten nur so lange vorgehalten, wie unbedingt notwendig, und nur einmal verarbeitet.*[9]

Die Messdaten werden in einem SCADA-System durch aktive Datenquellen zyklisch oder spontan übertragen. Dadurch erleichtert die Verwendung eines Streaming Systems mit seiner ereignisbasierten Verarbeitung als technologische Grundlage für ASSESS die Interoperabilität des Systems mit den vorhandenen Systemen und Prozessen. Darüber hinaus sollten sowohl Prozessdaten als auch Ereignisse von Anomaliedetektoren so schnell wie möglich, also ereignisgetrieben, verarbeitet werden, um eine Aktualität zu gewährleisten.

Um die nichtfunktionale Anforderung der Flexibilität zu erfüllen, eignen sich aber speziellere Streaming Systeme, nämlich Datenstrommanagementsysteme. Ein Datenstrommanagementsystem wird dabei wie folgt definiert:

Definition 3. *Ein Datenstrommanagementsystem (DSMS) ist ein Streaming System erweitert um Eigenschaften von Datenbankmanagementsystemen wie Anfrageverwaltung, vordefinierte Operatoren, Anfrageoptimierung und Zugriffskontrolle.*[10]

[8] Da der deutsche Begriff der (Daten-) Stromsysteme im Kontext dieser Arbeit mehrdeutig ist, wird der englische Fachbegriff Streaming System verwendet.
[9] Diese Definition basiert auf den Anforderungen an eine (Daten-) Stromverarbeitung aus [SÇZ05].
[10] Diese Definition basiert auf den Definitionen von DSMSs aus [CM12] und [Gei13].

1.3.2 Demonstration und Evaluation

In diesem Unterabschnitt wird die Strategie für die durchgeführte Demonstration und Evaluation vorgestellt. Da es sich bei Stromsystemen um kritische Infrastrukturen handelt, ist es nicht möglich, ASSESS in einem realen Setup zu testen. Aus diesem Grund wird ASSESS in einer Laborumgebung demonstriert und evaluiert. In der Laborumgebung werden ein Stromsystem, Informationsquellen für die Trust-Schätzung, sowie Angriffs- und Fehlerszenarien simuliert.

Ziel der Demonstration ist es, Trust-Einbußen zu erkennen und zu zeigen, dass durch ASSESS, im Gegensatz zu einer normalen Lagebildanalyse, Lagebilder geliefert werden, die diese Trust-Einbußen widerspiegeln, und, wenn gewünscht, alternative, möglichst vertrauenswürdige Lagebilder. Als Evaluationsmetrik wird der Abstand der Zustandsvariablen zur realen physikalischen Größe in Relation zum Trust in die Zustandsvariablen gewählt. Bei korrekten, fehlerfreien Messwerten ist der Abstand der Zustandsvariablen zur realen physikalischen Größe minimal und der Trust in die Zustandsvariablen maximal. Neben der Funktionalität wird ASSESS auch bzgl. der nichtfunktionalen Anforderungen evaluiert. Für die Aktualität wird die Latenz als Metrik verwendet, d.h. die Zeitspanne zwischen dem Eintreffen von Messwerten im System und der Ausgabe von Ergebnissen. Die technische Interoperabilität wird argumentativ durch die verwendete Technologie gezeigt. Für die Prozessinteroperabilität, Flexibilität und Skalierbarkeit wird ein morphologischer Kasten mit entsprechenden Variationspunkten erstellt.

1.3.3 Dissemination

Dieser Unterabschnitt widmet sich der betriebenen Dissemination der erzielten Forschungsergebnisse. Die einzelnen Artefakte werden dabei in teilweise unterschiedlichen Publikationen behandelt. PSNA-Trust ist Gegenstand einer Publikation im Rahmen des internationalen Workshops zu Flexiblitäts- und Resilienzproblemen elektrischer Energiesysteme 2019 [Bra+19b]. Die Arbeit zum Thema Trust-sensitive State Estimation ist Teil der Tagungsbänder der neunten DACH+-Energieinformatik-Konferenz 2020 [Bra+20] und der IEEE-PowerTech-Konferenz 2021 [BBL21]. Die Demonstration ist Gegenstand von Veröffentlichungen auf der IEEE-PowerTech-Konferenz 2019 [Bra+19a] und der zehnten DACH+-Energieinformatik-Konferenz [Bra+21]. Eine Publikation über das Gesamtsystem mit den Evaluationsergebnissen ist Teil der Tagungsbänder der zwölften DACH+-Energieinformatik-Konferenz [BEL23]. Ferner gibt es eine Veröffentlichung zum Beitrag der vorliegenden Arbeit zur Resilienzforschung [Bab+22] und eine Einreichung zur Forschung an IKT-abhängigen

Funktionalitäten in CPESs [BNL23]. Die vorliegende Dissertation dient als abschließende und umfassende Publikation. Eine Publikationsliste ist in den Verzeichnissen dieses Dokuments zu finden.

1.4 Zusammenfassung und Struktur der Arbeit

In dieser Einleitung wurde die Arbeit zunächst in Abschnitt 1.1 motiviert. Das übergeordnete Ziel sind zuverlässige netzdienliche Operationen in modernen CPESs. Die Grundlage für dieses Ziel stellt eine Trust-sensitive Lagebilderkennung dar, die auf einer Erhebung und Berücksichtigung des Trusts in Messdaten basiert. In Abschnitt 1.2 wurden die in dieser Arbeit behandelte Forschungsfrage, wie der multivariate Trust in physische Messwerte in einem CPES modelliert, geschätzt und in eine Lagebilderkennung integriert werden kann, sowie die abgeleiteten Forschungsziele und nichtfunktionalen Anforderungen vorgestellt. Die drei Forschungsziele sind ein kontextsensitives, multivariates Trust-Modell, eine Integrationsplattform für Trust-Schätzer und eine Trust-sensitive State Estimation. Aktualität, technische und Prozessinteroperabilität sowie Flexibilität und Skalierbarkeit wurden als nichtfunktionale Anforderungen identifiziert. Die in Abschnitt 1.3 vorgestellte Methodologie zur Beantwortung der Forschungsfrage folgt dem Design Science Research Process [Pef+06]. Dabei wurden als Artefakte einer Lösung das Trust-Modell PSNA-Trust, ein Anomalieerkennungsframework (engl. anomaly detection framework) (ADF), eine anomaliesensitive State Estimation (ASSE) sowie die Gesamtarchitektur des zu entwickelnden Systems präsentiert. Darüber hinaus wurden in dem Abschnitt die Demonstration, Evaluation und Dissemination kurz vorgestellt.

Der Rest der Arbeit ist wie folgt gegliedert: Kapitel 2 behandelt mit SCADA-Systemen, State Estimation, Bad Data Detection, multivariatem Trust und DSMSs die für diese Arbeit wichtigen Grundlagen. In den folgenden Kapiteln werden die relevanten Forschungsziele inkl. verwandter Arbeiten behandelt: das Trust-Modell in Kapitel 3, die Integrationsplattform für Trust-Schätzer in Kapitel 4 und die Trust-sensitive Lagebilderkennung in Kapitel 5. Das Gesamtsystem wird in Kapitel 6 vorgestellt und demonstriert sowie in Kapitel 7 evaluiert. Abgeschlossen wird die vorliegende Arbeit in Kapitel 8 mit einer Vorstellung bereits existierender Verwertungen, Diskussionen über Limitierungen und etwaige zukünftige Arbeiten sowie einem Fazit.

2 Grundlagen

> *Der Beginn der Weisheit ist die Definition der Begriffe.*
>
> **— Sokrates**

In diesem Kapitel werden die wesentlichen Grundlagen für die vorliegende Arbeit aufbereitet. Die Abschnitte 2.1 und 2.2 beschreiben mit SCADA-Systemen bzw. der State Estimation und der Bad Data Detection die Zielsysteme und -anwendungen für ASSESS. Mit multivariatem Trust setzt sich Abschnitt 2.3 auseinander, während in Abschnitt 2.4 DSMSs als technologische Grundlage von ASSESS eingeführt werden. Abschnitt 2.5 fasst dieses Grundlagenkapitel zusammen.

2.1 Supervisory, Control, and Data Acquisition Systeme

Ein Supervision, Control, and Data Acquisition (SCADA)-System ist „eine Sammlung von Gerätschaften, die einen Operateur mit genügend Informationen versorgt, um den Status eines bestimmten Gerätes oder Prozesses in einer Fernwirkstation zu bestimmen und Aktionen bzgl. dieses Gerätes oder Prozesses zu veranlassen, ohne körperlich anwesend zu sein“ [TM15] (aus dem Englischen übersetzt). SCADA-Systeme haben demnach die Aufgabe, menschliche Operateure in Kontrollzentren zu unterstützen, und verfügen über Komponenten im Feld[1] sowie im Kontrollzentrum, die miteinander kommunizieren.

Neben höheren Funktionen, die in Unterabschnitt 2.1.2 beschrieben werden, lassen sich die grundlegenden Aufgaben von SCADA-Systemen in zwei Prozesse strukturieren: einen Prozess zur Datenakquise und Überwachung und einen zur Steuerung eines Stromsystems [TM15].

Der Prozess zur Datenakquise und Überwachung ist wie folgt (teilweise wörtlich aus dem Englischen übersetzt) [TM15]. Zunächst werden Daten im Feld gesammelt, in ein Übertragungsformat konvertiert und in Datenpakete zusammengefasst. Die

[1] Als Feld wird in dieser Arbeit der Teil des verteilten Stromsystems bezeichnet, der mittels Fernwirktechnik überwacht und gesteuert wird.

Datenpakete werden anschließend an das Kontrollzentrum übertragen, um dort dekodiert zu werden und um die enthaltenen Daten den Operateuren adäquat anzuzeigen. Die Steuerung wird entsprechend durch den folgenden Prozess ermöglicht (teilweise wörtlich aus dem Englischen übersetzt) [TM15]. Zunächst wird ein Steuerungskommando im Kontrollzentrum durch einen Operateur initiiert und in ein Datenpaket konvertiert. Das Datenpaket wird anschließend an das Feld übertragen, um dort dekodiert zu werden und um das enthaltene Steuerungskommando durch eine geeignete Geräteoperation umzusetzen. Anhand dieser Prozesse ist zu erkennen, dass ein SCADA-System aus unterschiedlichen Komponenten besteht, die in Unterabschnitt 2.1.1 näher erläutert werden. Unterabschnitt 2.1.2 widmet sich den Funktionen von SCADA-Systemen und Unterabschnitt 2.1.3 den Protokollen, mit denen Datenpakete zwischen Feld und Kontrollzentrum übertragen werden. Da bei der prototypischen Implementierung von ASSESS das in Europa weit verbreitete Protokoll IEC 60870-5-104 Verwendung findet, wird dieses in Unterabschnitt 2.1.4 detaillierter als die anderen Protokolle vorgestellt. Zudem bietet [TM15] umfassendere Literatur zu SCADA-Systemen.

2.1.1 Komponenten

Wie bereits erwähnt sind SCADA-Systeme verteilte Systeme mit Komponenten im Feld und im Kontrollzentrum, die miteinander kommunizieren. Die wesentlichen Komponenten sind dabei Fernwirkstationen (engl. Remote Terminal Units (RTUs)[2]) im Feld, die Masterstation im Kontrollzentrum und das IKT-System, das die RTUs und Masterstation verbindet [TM15].

Eine RTU dient zum einen bei der Datenakquise und Überwachung als Sammelpunkt für Daten von Geräten im Feld. Sensoren erfassen physikalische Größen, wie etwa Spannung oder Strom, und eine RTU versendet die von diesen Geräten gesammelten Daten gebündelt mithilfe des IKT-Systems. Zum anderen erfüllt eine RTU bei der Steuerung den Zweck eines Verteilungspunktes für Steuerbefehle, die mithilfe des IKT-Systems übertragen wurden. Sie interpretiert die Befehle und setzt sie durch Aktoren, wie z.B. einen Leistungsschalter, um [TM15].

Im Kontrollzentrum, auch Netzleitwarte genannt, befindet sich die (häufig zentrale) Masterstation. Sie stellt im wesentlichen eine Infrastruktur dar, die Operateure bei der Überwachung und Steuerung des Stromsystems unterstützt. Dies umfasst vor allem auch eine Mensch-Maschine-Schnittstelle, die dem Operateur alle Mess-

[2] In dieser Arbeit wird der englische Fachbegriff der Remote Terminal Unit verwendet.

und Systemdaten grafisch aufbereitet zur Verfügung stellt und mittels derer der Operateur Steuerbefehle ausführen kann [TM15].

Die Datenübertragung erfolgt durch ein IKT-System mit Sendern und Empfängern, zwischen denen Nachrichten (Messwerte oder Steuerbefehle verpackt in Datenpakete) übermittelt werden. Zu diesem Zweck können unterschiedliche Medien, wie z.B. Kabel, und Protokolle, wie z.B. das IEC 60870-5-104, zum Einsatz kommen.

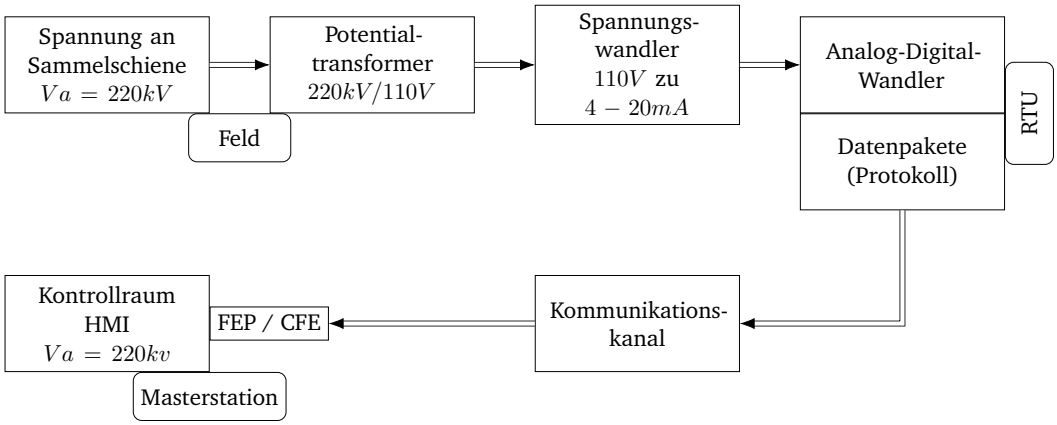


Abb. 2.1.: Das Zusammenwirken der SCADA-Komponenten anhand einer beispielhaften Instantiierung des Datenakquise- und Überwachungsprozesses [TM15] (aus dem Englischen übersetzt). Va ist die Spannung an Phase a, FEP ein Frontendprozessor, CFE ein Kommunikationsfrontend und HMI eine Mensch-Maschine-Schnittstelle.

Abbildung 2.1 stellt das Zusammenwirken der genannten SCADA-Komponenten anhand einer beispielhaften Instanz des Prozesses zur Datenakquise und Überwachung dar (aus [TM15] entnommen und aus dem Englischen übersetzt). Im Feld, oben links in Abbildung 2.1, wird eine Spannung von $220kV$ an einer Phase gemessen, auf $110V$ transformiert und in einen Stromwert gewandelt. Dieser analoge Stromwert von $4 - 20mA$ wird in der RTU (rechts in Abbildung 2.1) in einen digitalen Wert gewandelt und gemäß dem in Abschnitt 2.1 beschriebenen Prozess zur Datenakquise und Überwachung in ein Übertragungsformat konvertiert und ggf. mit anderen Daten zu Datenpaketen zusammengefasst. Über einen Kommunikationskanal (Teil des IKT-Systems) werden die Datenpakete an die Masterstation (unten links in Abbildung 2.1) übertragen. Mithilfe eines Frontendprozessors und eines Kommunikationsfrontends werden die Datenpakete entgegengenommen und dekodiert. Die Mensch-Maschine-Schnittstelle sorgt für eine adäquate Darstellung der ursprünglich gemessenen $220kV$ Phasenspannung [TM15].

2.1.2 Funktionen

SCADA-Systeme werden in unterschiedlichen Bereichen von Stromsystemen eingesetzt und erfüllen je nach Bereich auch unterschiedliche Funktionen [TM15].

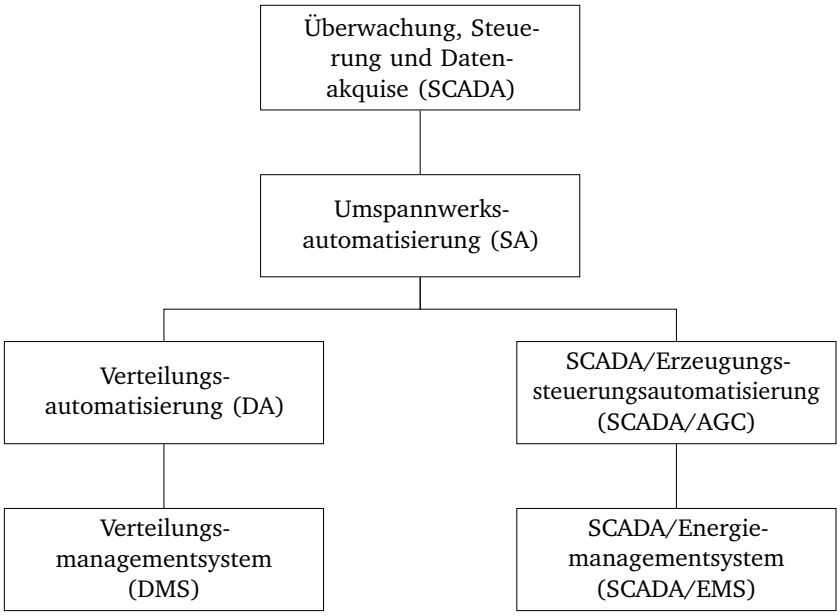


Abb. 2.2.: Die Verwendung von SCADA- in Stromsystemen [TM15] (aus dem Englischen übersetzt). Im Englischen steht SA für Substation Automation, DA für Distribution Automation, DMS für Distribution Management System und AGC für Automation Generation Control.

Abbildung 2.2 gibt einen Überblick über die Einsatzbereiche von SCADA [TM15]. Der oberste Block wird durch die Basisfunktionen von SCADA-Systemen gefüllt. Diese beinhalten „Datenakquise, Fernwirkung, Mensch-Maschine-Schnittstelle, Analyse historischer Daten und Berichtswesen“ [TM15] (aus dem Englischen übersetzt). Der nächste Block umfasst Funktionen zur Umspannwerksautomatisierung. Die linken Blöcke werden in Verteilnetzen zur Automatisierung und Verwaltung eingesetzt. Funktionen, die über die Basisfunktionen und die der Umspannwerksautomatisierung hinausgehen, umfassen u.a. eine Einspeisungs- und Kundenautomatisierung. Der rechte obere Block der Erzeugungssteuerungsautomatisierung wird in Erzeugungssteuerungszentren umgesetzt und beinhaltet u.a. Funktionen zur Überwachung der Frequenz und Interferenzen, sowie zur Fahrplanerstellung. Der Block wird gefolgt von einem Block, der Funktionen für das Energiemanagementsystem enthält. Diese Funktionen werden in Übertragungsnetzen eingesetzt und sind zum Teil hochgradig komplex. Sie umfassen u.a. die State Estimation und Ausfallrechnung. Insgesamt werden die SCADA-Funktionen in Abbildung 2.2 von oben nach unten gesehen immer komplexer [TM15].

Tab. 2.1.: Vergleich von Protokollen, die in SCADA-Systemen zum Einsatz kommen [Chr19] (aus dem Englischen übersetzt und angepasst).

Protokoll	Einsatzgebiet	Funktion	Sicherheit
Modbus/TCP	• weltweit • verschiedene Sektoren (u.a. Gas und Strom)	Kommunikation zwischen Geräten im selben Netzwerk (z.B. zwischen RTU und Masterstation)	nicht eingebaut
IEC 60870	• Europa und Nordafrika • Energiesysteme	Kommunikation zwischen RTU und Masterstation	nicht eingebaut
DNP3	• Amerika und Asien (Energiesysteme) • Europa (andere Sektoren)	Kommunikation zwischen RTU und Masterstation	nicht eingebaut
IEC 61850	• weltweit • Umspannwerksautomatisierung	Kommunikation von intelligenten elektronischen Geräten in Umspannwerken	nicht eingebaut
OPC	• weltweit • Übersetzungsprotokoll	• Operabilitätsstandard • Umwandlung von SCADA-Protokollen ins OPC-Protokoll	keine Angabe

2.1.3 Protokolle

Bei der Datenübertragung in einem SCADA-System können unterschiedliche Protokolle zum Einsatz kommen. Dieser Unterabschnitt gibt einen kurzen Überblick über die Protokolle, während der folgende Unterabschnitt das IEC 60870-5-104-Protokoll detaillierter beschreibt.

Tabelle 2.1 listet die gängigsten Protokolle im Kontext von SCADA-Systemen, ihre Einsatzgebiete, Funktionen und Sicherheitsmechanismen auf [Chr19]. Einige Protokolle werden weltweit eingesetzt, während sich die Einsatzgebiete anderer auf bestimmte Regionen oder Kontinente beschränken. Die Protokolle unterscheiden sich auch hinsichtlich der Domänen, in denen sie eingesetzt werden. IEC 60870, zum Beispiel, wird nur bei der Umspannwerksautomatisierung verwendet. OPC hingegen ist ein domänenunabhängiges Übersetzungsprotokoll. Drei der fünf aufgelisteten Protokolle werden zur Kommunikation zwischen RTUs und Masterstationen eingesetzt. Dort liegt durch den Prozess der State Estimation auch der Fokus dieser Arbeit. Insbesondere liegt der Fokus auf dem Protokoll IEC 60870-5-104 aus der Standardreihe IEC 60870, da es in europäischen Energiesystemen zum Einsatz kommt. Erwähnenswert ist außerdem, dass Sicherheitsmechanismen protokollübergreifend nicht eingebaut sind [Chr19; TM15].

2.1.4 IEC 60870-5-104

Zur Kommunikation zwischen RTUs und Masterstationen wird in Europa zumeist der Standard IEC 60870-5-104 (104er) [Int16] eingesetzt. Dieser basiert auf dem Standard IEC 60870-5-101 [Int15], bei dem die Daten noch seriell übertragen werden. Im 104er-Standard geschieht dies über TCP/IP. Ein im 104er-Standard übertragenes Telegramm besteht aus einer Anwendungsprotokolldateneinheit (engl. Application Protocol Data Unit (APDU)). Deren Aufbau ist in Abbildung 2.3 dargestellt. Eine APDU setzt sich aus einer Anwendungssteuerungsinformation (engl. Application Protocol Control Information (APCI)) und je nach Steuerungsinformationen einer Anwendungsservicedateneinheit (engl. Application Service Data Unit (ASDU)) zusammen. Alle APCIs fangen mit demselben Byte an, 64 hexadezimal, gefolgt von der Restlänge des Telegramms (Länge abzüglich Start- und Längenbyte). Es folgen vier Bytes, die je nach Steuerungsinformationsformat variieren [Int16].

Es gibt drei Formate: das I-Format, das S-Format und das U-Format, welche in Tabelle 2.2 bzgl. der in diesen Formaten übersandten Informationen aufgelistet sind. Im I-Format werden Informationen transferiert. Dies ist das einzige Format, bei dem

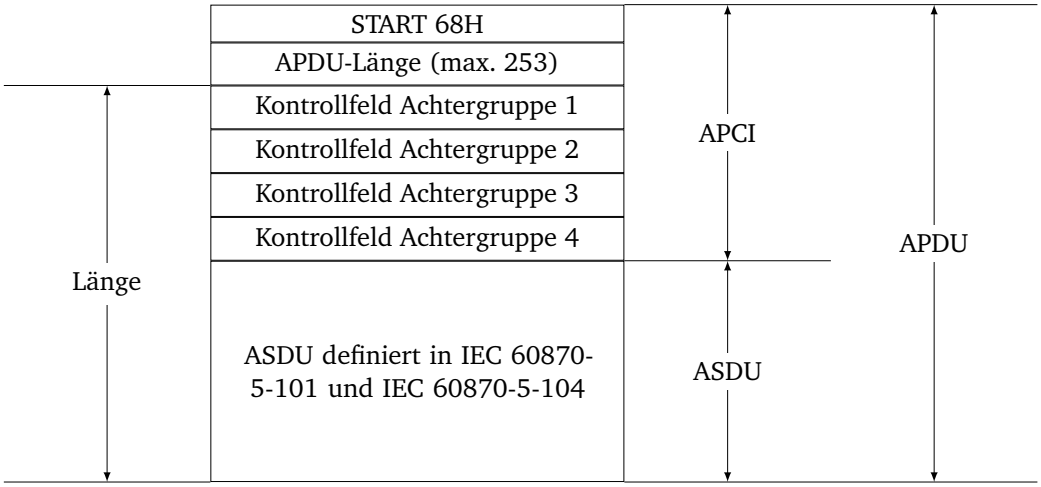


Abb. 2.3.: Der Aufbau einer APDU nach IEC 60870-5-104 [Int16].

die APDU auch eine ASDU enthält. Letztere enthält die übertragenen Informationen. Die Steuerungsfelder in der APCI enthalten dabei Sendungs- und Empfangssequenznummern mit denen Sender und Empfänger eine vollständige Übertragung aller Telegramme überprüfen können. Das S-Format wird als Antwort auf Telegramme im I-Format verwendet. In den Steuerungsfeldern der APCI wird dabei die Empfangssequenznummer des Telegramms übermittelt, bis zu dem alle Telegramme erhalten wurden. Das U-Format wird nicht für die eigentliche Informationsübertragung verwendet. Es dient zur Initiierung oder Beendigung des Datentransfers oder zum Versenden von Testnachrichten [Int16].

Tab. 2.2.: Auflistung der übersandten Informationen je 104er-Format [Int16].

Format	**Übersandte Informationen**
I-Format	Sendungssequenznummer, Empfangssequenznummer und ASDU
S-Format	Empfangssequenznummer
U-Format	Initiierung oder Bestätigung von Beginn Datenübertragung, Ende Datenübertragung oder Test

Der Aufbau einer ASDU ist in Abbildung 2.4 dargestellt. Eine ASDU besteht aus einem Dateneinheitsidentifizierer und n Informationsobjekten. Der Dateneinheitsidentifizierer gibt dabei Aufschluss über den Typ der ASDU, z.B. Übertragung normierter Messwerte, der Anzahl an enthaltenen Informationsobjekten und deren Struktur, dem Grund der Übertragung sowie der Adresse der ASDU zum Zwecke der Zuordnung und Interpretation. Ein Informationsobjekt umfasst ebenfalls eine Adresse, n Informationselemente und optional einen Zeitstempel [Int15]. Um eine ASDU und die enthaltenen Informationsobjekte korrekt zu interpretieren und zuzuordnen, bedarf es demnach Kontextinformationen.

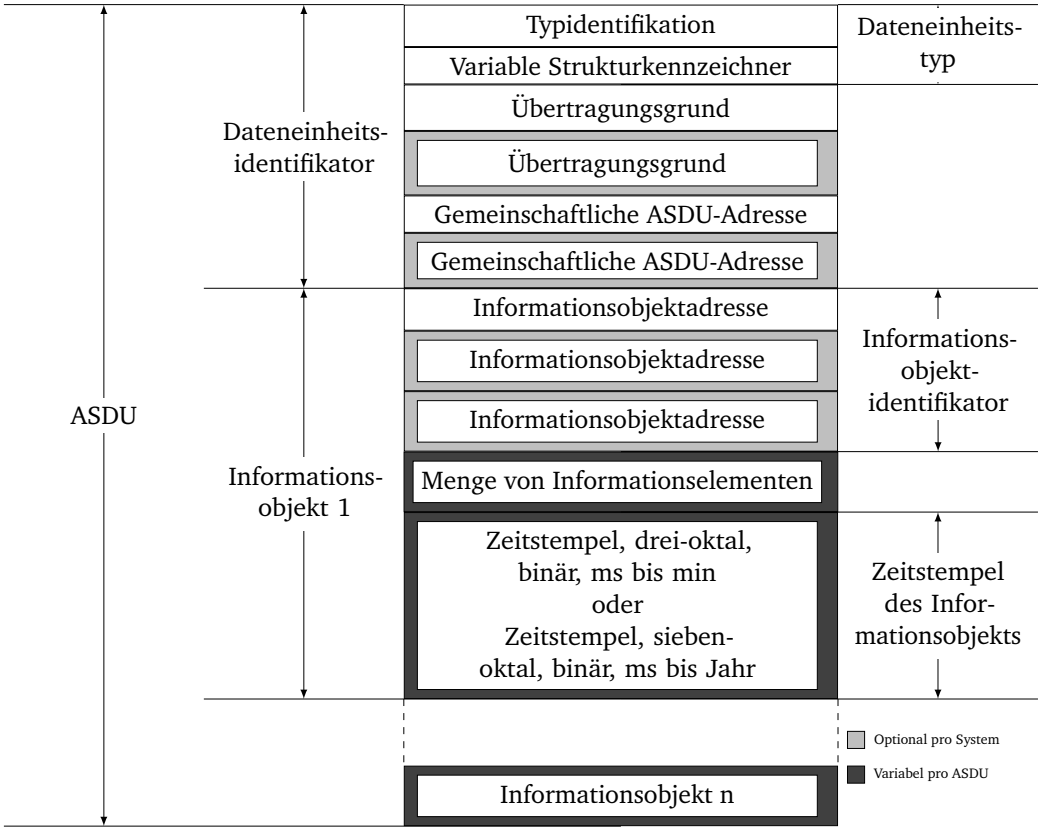


Abb. 2.4.: Der Aufbau einer ASDU nach IEC 60870-5-101 [Int15].

2.2 State Estimation und Bad Data Detection

Der operative Betrieb eines Stromsystems lässt sich mit fünf Zuständen klassifizieren: Normal-, Alarm-, Notfall-, Extrem- und Erholungszustand [SG16]. Der Normalzustand ist der angestrebte Zustand, in dem alle angeschlossenen Lasten mit Strom versorgt werden können und dabei keine Bedingungen des operativen Betriebs (z.B. Spannungsbänder) verletzt werden. Kommt es zu solchen genannten Verletzungen, die allerdings noch tolerierbar sind, befindet sich das Stromsystem im Alarmzustand [SG16].

Können die angeschlossenen Lasten beim Auftreten von Eventualitäten nicht mehr mit Strom versorgt werden, wechselt das System in den Notfall- oder Extremzustand, abhängig von der Schwere der Störung [SG16]. Die dabei betrachteten Eventualitäten (engl. contingencies) werden vorab in einer Analyse bestimmt. In solch einem Fall ist unmittelbares Handeln der Operateure vonnöten. Dieses Handeln kann dazu führen, dass die Bedingungen des operativen Betriebs wieder erfüllt, aber nicht mehr

alle Lasten mit Strom versorgt werden. Der Zustand, bei dem versucht wird wieder alle Lasten mit Strom zu versorgen, wird Erholungszustand genannt [AE04].

Den Prozess der Identifizierung des aktuellen Zustands wird auch Sicherheitsanalyse [AE04] genannt. Man spricht aber häufig auch vom Situationsbewusstsein (engl. situation(al) awareness) [PK15]. In dieser Arbeit werden diese unterschiedlichen Perspektiven und Aspekte unter dem Begriff der Lagebilderkennung zusammengefasst. Als Voraussetzung müssen die relevanten Systemvariablen, insbesondere die Messwerte, überwacht (siehe Abschnitt 2.1) und daraus die Zustandsvariablen geschätzt werden [AE04]. Die Zustandsvariablen sind die komplexen Spannungen an den Netzknoten und der Prozess zur Schätzung dieser Variablen ist die State Estimation. Konkreter umfasst die Funktionalität einer State Estimation Folgendes:

- Topologieverarbeitung: Es werden Statusdaten über Leistungs- sowie Lastschalter gesammelt und ein Onlinezustand der Systemtopologie zur Verfügung gestellt [AE04].
- Beobachtbarkeitsanalyse: Es wird bestimmt, ob für alle Bereiche des Systems der Zustand geschätzt werden kann, oder ob für einige Bereiche die zur Verfügung stehenden Messwerte nicht ausreichen. Es werden die nicht beobachtbaren Bereiche und etwaige beobachtbare Inseln im System identifiziert [AE04].
- State-Estimation-Lösung: Auf Basis des Netzwerkmodells und der zur Verfügung stehenden Messwerte wird eine Schätzung der Zustandsvariablen vorgenommen [AE04].
- Bad Data Detection: Grobe Fehler in den Messwerten werden erkannt, identifiziert und herausgerechnet [AE04].
- Verarbeitung von Parameter- und strukturellen Fehlern: Netzwerkparameter, wie z.B. Parameter von Stufentransformatoren, werden geschätzt, strukturelle Fehler in der Netzwerkkonfiguration ausfindig gemacht und fehlerhafte Trennschalterzustände identifziert, sofern eine ausreichende Messredundanz gegeben ist [AE04].

Dieser Abschnitt fokussiert sich auf die State-Estimation-Lösung und Bad Data Detection. In Unterabschnitt 2.2.1 wird zunächst der Zusammenhang zwischen Messwerten und Zustandsvariablen erörtert. In Unterabschnitt 2.2.2 folgt die Vorstellung eines weit verbreiteten Vorgehens zur Schätzung der Zustandsvariablen: die State Estimation mit gewichteten kleinsten Quadraten. Unterabschnitt 2.2.3 widmet sich der Bad Data Detection. Der Abschnitt schließt mit einer Betrachtung

von State-Estimation-Verfahren für unterbestimmte Systeme in Unterabschnitt 2.2.4. Umfassendere Literatur zu den Themen State Estimation und Bad Data Detection bietet zudem [AE04].

2.2.1 Zusammenhang zwischen Messwerten und Zustandsvariablen

Den Zusammenhang zwischen den Zustandsvariablen und den Messwerten zeigt Gleichung 2.1 [AE04].

$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{e} \tag{2.1}$$

$\boldsymbol{z}$ ist der m-elementige Messwertevektor und $\boldsymbol{x}$ der 2n-elementige Zustandsvektor mit $x^T = [\theta_1\theta_2\theta_3 \cdots \theta_n V_1 V_2 \cdots V_n]$. θ_i ist der Spannungsphasenwinkel und V_i die Spannungsmagnitude an Sammelschiene i. Dabei bezeichnet θ_1 den Spannungsphasenwinkel an einer Referenzsammelschiene und wird daher beliebig festgelegt, oft mit 0. $\boldsymbol{h}(\boldsymbol{x})$ ist eine nichtlineare Funktion, die den komplexen Zustandsvektor auf den komplexen Messwertevektor abbildet, und $\boldsymbol{e}$ der m-elementige Fehlervektor. Letzterer bildet Messfehler ab und es werden die folgenden beiden Annahmen über $\boldsymbol{e}$ getroffen [AE04]. Erstens folgt ein Fehler e_j der Normalverteilung mit einer Standardabweichung σ_j (i.d.R. bekannt vom Messgerät) und einem Erwartungswert von $E(e_j) = 0$. Zweitens sind die Fehler statistisch unabhängig, d.h. $E(e_j e_k) = 0$.

$$Cov(\boldsymbol{e}) = \boldsymbol{R} = diag\{\sigma_1^2, \sigma_2^2, \ldots, \sigma_m^2\} \tag{2.2}$$

Daraus ergibt sich die Kovarianzmatrix von $\boldsymbol{e}$ wie in Gleichung 2.2 [AE04].

$$\boldsymbol{Y} = \begin{bmatrix} Y_{11} & \ldots & Y_{1n} \\ \ldots & \ddots & \ldots \\ Y_{n1} & \ldots & Y_{nn} \end{bmatrix} \quad \text{mit} \tag{2.3}$$

$$Y_{ik} = G_{ik} + j \cdot B_{ik} \quad \forall i,k \in \{1,\ldots,n\} \tag{2.4}$$

Als Grundlage für die Erläuterung von $\boldsymbol{h}(\boldsymbol{x})$ wird zunächst die Admittanzmatrix $\boldsymbol{Y}$ eines Netzwerkes in den Gleichungen 2.3 und 2.4 vorgestellt [AE04]. Die Admittanz ist der Kehrwert einer Impedanz und komplexwertig. Die Konduktanz G bildet den Real- und die Suszeptanz B den Imaginärteil. $\boldsymbol{Y}$ stellt die Admittanzen zwischen

allen n Knoten untereinander in einem Netzwerk dar und hat die folgenden vier Eigenschaften [AE04]: $\boldsymbol{Y}$ ist eine $n \times n$-Matrix, komplexwertig, spärlich besetzt, und nicht singulär.

Die Einträge $h_i(\boldsymbol{x})$ aus $\boldsymbol{h}(\boldsymbol{x})$, die $\boldsymbol{x}$ auf ein z abbilden, hängen von der Art des entsprechenden Messwertes ab.

$$P_i = V_i \cdot \sum_{k \in N_i} V_k \cdot (G_{ik} \cdot \cos(\theta_i - \theta_k) + B_{ik} \cdot \sin(\theta_i - \theta_k)) \tag{2.5}$$

$$Q_i = V_i \cdot \sum_{k \in N_i} V_k \cdot (G_{ik} \cdot \sin(\theta_i - \theta_k) - B_{ik} \cdot \cos(\theta_i - \theta_k)) \tag{2.6}$$

Die Gleichungen 2.5 und 2.6 zeigen den Zusammenhang zwischen den Zustandsvariablen und der Wirk- (P_i) bzw. Blindleistungseinspeisung (Q_i) an einer Sammelschiene i [AE04]. N_i bezeichnet dabei die Menge der benachbarten Sammelschienen von i, also jener Sammelschienen, die direkt mit der Sammelschiene i verbunden sind und für deren Verbindung zu i es entsprechend einen Eintrag in $\boldsymbol{Y}$ gibt. P_i und Q_i hängen über $\boldsymbol{h}(\boldsymbol{x})$ von den komplexen Spannungen an der Sammelschiene i und den Sammelschienen in der Nachbarschaft von i ab [AE04].

$$P_{ik} = V_i^2 \cdot (g_{si} + g_{ik}) - V_i \cdot V_k \cdot (g_{ik} \cdot \cos(\theta_i - \theta_k) + b_{ik} \cdot \sin(\theta_i - \theta_k)) \tag{2.7}$$

$$Q_{ik} = -V_i^2 \cdot (b_{si} + b_{ik}) - V_i \cdot V_k \cdot (g_{ik} \cdot \sin(\theta_i - \theta_k) - b_{ik} \cdot \cos(\theta_i - \theta_k)) \tag{2.8}$$

Der Zusammenhang zwischen den Zustandsvariablen und dem Wirk- (P_{ik}) und Blindleistungsfluss (Q_{ik}) von einer Sammelschiene i zu einer Sammelschiene k wird in den Gleichungen 2.7 bzw. 2.8 gezeigt [AE04]. g_{ik} und b_{ik} bezeichnen die Konduktanz bzw. Suszeptanz der Serienverbindung (engl. series branch) zwischen den Sammelschienen i und k, während g_{si} und b_{si} die Konduktanz bzw. Suszeptanz der Nebenschlussverbindung (engl. shunt branch) bei Sammelschiene i bezeichnen. P_{ik} und Q_{ik} hängen über $\boldsymbol{h}(\boldsymbol{x})$ von den komplexen Spannungen an den verbundenen Sammelschienen i und k ab [AE04].

$$I_{ik} = \frac{\sqrt{P_{ik}^2 + Q_{ik}^2}}{V_i} \tag{2.9}$$

Die Gleichung 2.9 zeigt den Zusammenhang zwischen den Zustandsvariablen und der Stromflussmagnitude (I_{ik}) von einer Sammelschiene i zu einer Sammelschiene k [AE04]. I_{ik} hängt demnach über $\boldsymbol{h}(\boldsymbol{x})$ von den komplexen Spannungen an den verbundenen Sammelschienen i und k ab [AE04].

2.2.2 State Estimation mit gewichteten kleinsten Quadraten

Das Ziel der State Estimation ist es, die Zustandsvariablen derart zu schätzen, dass sie, auf Basis des Netzwerkmodells, am besten zu den Messwerten passen. Dies bedeutet, dass der Betrag des Fehlervektors $\boldsymbol{e}$ minimiert werden muss, um eine optimale Schätzung zu erreichen [AE04]. Bei der State Estimation mit gewichteten kleinsten Quadraten wird dazu eine Maximum-Likelihood-Schätzung [AE04] angewandt. Nach dieser Methode wird die gewichtete Summe der quadrierten Residuen (Einträge im Fehlervektor $\boldsymbol{e}$) minimiert.

$$\text{minimiere } \boldsymbol{J}(\boldsymbol{x}) = \sum_{j=1}^{m}\left(\frac{(z_j - h_j(\boldsymbol{x}))^2}{R_{jj}}\right) = [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T \cdot \boldsymbol{R}^{-1} \cdot [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{2.10}$$

Die Zielfunktion $\boldsymbol{J}(\boldsymbol{x})$ ist in Gleichung 2.10 dargestellt [AE04]. $z_j - h_j(\boldsymbol{x})$ ergibt sich dabei aus Gleichung 2.1 und R_{jj} entspricht nach Gleichung 2.2 der Varianz von e_j.

$$\boldsymbol{g}(\boldsymbol{x}) = \frac{\partial \boldsymbol{J}(\boldsymbol{x})}{\partial \boldsymbol{x}} = -\boldsymbol{H}^T(\boldsymbol{x}) \cdot \boldsymbol{R}^{-1} \cdot [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \stackrel{!}{=} \boldsymbol{0} \quad \text{mit} \tag{2.11}$$

$$\boldsymbol{H}(\boldsymbol{x}) = \frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}} \tag{2.12}$$

Eine notwendige Voraussetzung für ein Minimum ist, dass die erste Ableitung an der entsprechenden Stelle $\boldsymbol{0}$ ist. Diese Bedingung ist in den Gleichungen 2.11 und 2.12 beschrieben [AE04]. $\boldsymbol{H}$ ist dabei die Jacobi-Matrix von $\boldsymbol{h}(\boldsymbol{x})$.

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{g}(\boldsymbol{x}^k) + \boldsymbol{G}(\boldsymbol{x}^k) \cdot (\boldsymbol{x} - \boldsymbol{x}^k) + \ldots \stackrel{!}{=} \boldsymbol{0} \quad \text{mit} \tag{2.13}$$

$$\boldsymbol{G}(\boldsymbol{x}^k) = \frac{\partial \boldsymbol{g}(\boldsymbol{x}^k)}{\partial \boldsymbol{x}^k} = \boldsymbol{H}^T(\boldsymbol{x}^k) \cdot \boldsymbol{R}^{-1} \cdot \boldsymbol{H}(\boldsymbol{x}^k) \tag{2.14}$$

Da $\boldsymbol{g}(\boldsymbol{x})$ nichtlinear ist, wird i.d.R. eine Taylorreihe um einen Entwicklungspunkt $\boldsymbol{x}^k$, wie in den Gleichungen 2.13 und 2.14 dargestellt, entwickelt [AE04]. $\boldsymbol{G}(\boldsymbol{x}^k)$ wird Verstärkungsmatrix (engl. gain matrix) genannt. Sie ist spärlich besetzt, positiv definit, und symmetrisch, sofern das Netz vollständig beobachtbar ist [AE04].

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k - \frac{\boldsymbol{g}(\boldsymbol{x}^k)}{\boldsymbol{G}(\boldsymbol{x}^k)} \tag{2.15}$$

Werden die Terme höherer Ordnung in Gleichung 2.13 ignoriert, lassen sich die Nullstellen mit dem Gauß-Newton-Verfahren lösen [AE04]. Gleichung 2.15 zeigt den entsprechenden Gauß-Newton-Iterationsschritt [AE04], wobei $\boldsymbol{x}^k$ den Lösungsvektor in der $k$ten Iteration bezeichnet. Normalerweise wird $\boldsymbol{G}(\boldsymbol{x}^k)$ nicht invertiert, sondern

mittels einer Cholesky-Zerlegung zerlegt [AE04]. Der Grund ist, dass $\boldsymbol{G}(\boldsymbol{x}^k)$ zwar relativ spärlich ist, $\boldsymbol{G}(\boldsymbol{x}^k)^{-1}$ aber generell voll besetzt ist.

$$[\boldsymbol{G}(\boldsymbol{x}^k)] \cdot \Delta\boldsymbol{x}^{k+1} = \boldsymbol{H}^T(\boldsymbol{x}^k) \cdot \boldsymbol{R}^{-1} \cdot [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x}^k)] \quad \text{mit} \tag{2.16}$$

$$\Delta\boldsymbol{x}^{k+1} = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k \tag{2.17}$$

Stellt man Gleichung 2.15 um und ersetzt $\boldsymbol{g}(\boldsymbol{x}^k)$ gemäß Gleichung 2.11, erhält man die sogenannten Normalgleichungen (Gleichungen 2.16 und 2.17), die zu lösen sind [AE04]. Das Lösen der Normalgleichungen liefert ein $\Delta\boldsymbol{x}$, also eine Aussage darüber, wie stark sich $\boldsymbol{x}$ in der aktuellen Iteration verändert hat. Dieses iterative Lösungsverfahren wird fortgeführt, bis es keine nennenswerte Verbesserung mehr gibt.

```
1  k ← 0
2  init x^k
3  do
4     calc G(x^k)
5     calc H^T(x^k) · R^-1 · [z − h(x^k)]
6     decompose G(x^k) and solve for Δx^k
7     if max|Δx^k| > ϵ
8        x^(k+1) ← x^k + Δx^k
9        k ← k + 1
10 while max|Δx^k| > ϵ
```

Skript 2.1: Der iterative Lösungsalgorithmus für die State Estimation mit gewichteten kleinsten Quadraten [AE04].

Skript 2.1 zeigt den Lösungsalgorithmus als Pseudocode. Der Lösungsvektor in der ersten Iteration ($k = 0$) kann beliebig gewählt werden. In einer Iteration werden zunächst $\boldsymbol{G}(\boldsymbol{x}^k)$ und die rechte Seite von Gleichung 2.16 berechnet. Im Anschluss wird $\boldsymbol{G}(\boldsymbol{x}^k)$ zerlegt und Gleichung 2.16 wird für ein $\Delta\boldsymbol{x}^k$ gelöst. Ist der Unterschied für mindestens eine Zustandsvariable zu ihrem Wert aus der vorigen Iteration größer als ein Schwellwert ϵ, so wird mit einer weiteren Iteration fortgefahren. Ist das nicht der Fall, terminiert der Algorithmus [AE04].

2.2.3 Bad Data Detection

Im Anschluss an die State Estimation geht es bei der Bad Data Detection um das Erkennen, Identifizieren und Eliminieren von groben Messwertefehlern. Die Bad Data Detection wird dabei auf Grundlage der Messwerteresiduen, d.h. der Diskrepanz zwischen den gemessenen Werten und den theoretischen Werten auf Basis des geschätzten Systemzustands, durchgeführt [AE04]. In diesem Unterabschnitt wird

das grundlegende Vorgehen vorgestellt. Für eine detailliertere Betrachtung, z.B. der verschiedenen Verfahren, sei auf [AE04] verwiesen.

Bad Data kann nicht bei allen Messwerten identifiziert werden. Grundsätzlich werden Messwerte im Kontext der Bad Data Detection wie folgt klassifiziert [AE04]:

- Kritischer Messwert: Die Eliminierung eines kritischen Messwertes aus der Messwertemenge führt zu einem (teilweise) unüberwachbaren System. Das bedeutet, dass eine State Estimation auf Grundlage einer solchen reduzierten Messwertemenge nicht alle Zustandsvariablen bestimmen kann.
- Redundante Messwerte: Ein redundanter Messwert ist ein Messwert, der nicht kritisch ist.
- Kritische Messwertepaare: Die Eliminierung beider Messwerte eines kritischen Messwertepaares aus der Messwertemenge führt zu einem (teilweise) unüberwachbaren System.
- Kritische k-Messwertetupel: Die Eliminierung aller k Messwerte eines kritischen k-Messwertetupels aus der Messwertemenge führt zu einem (teilweise) unüberwachbaren System.

Man kann die Klassen folglich zu redundant und k-kritisch zusammenfassen für $k \geq 1$. Im ersten Schritt der Bad Data Detection geht es um das Erkennen, ob die Messwertemenge Bad Data enthält, ohne zu spezifizieren, um welche Messwerte es sich konkret handelt [AE04]. Da das System nach dem Entfernen der Bad Data noch überwachbar sein muss, kann Bad Data nur für redundante Messwerte erkannt werden. Die Verfahren zur Erkennung von Bad Data sind i.d.R. Schwellwertverfahren, wie z.B. der Chi-Quadrat-Test [AE04]. Der nächste Schritt ist die Identifizierung, bei welchen spezifischen Messwerten es sich um Bad Data handelt. In diesem Schritt kommen ebenfalls Schwellwertverfahren, wie z.B. der Größtes-Normalisiertes-Residuum-Test, zum Einsatz [AE04]. Ist das größte normalisierte Residuum aus der Messwertemenge größer als ein bestimmter Schwellwert, wird dieser Messwert aus der Messwertemenge eliminiert (dritter Schritt) und die State Estimation mit der reduzierten Messwertemenge erneut durchgeführt [AE04].

Das hier beschriebene Vorgehen, insbesondere für die Identifizierung und Eliminierung, eignet sich für einzelne Bad Data. Darüber hinaus können noch multiple Bad Data auftreten, die entweder nicht oder inkonsistent oder konsistent miteinander interagieren. Für die ersten beiden Fälle können ebenfalls Verfahren wie der Größtes-Normalisiertes-Residuum-Test angewandt werden, da die einzelnen inkonsistenten

Bad Data iterativ entfernt werden können. Für konsistent miteinander interagierende Bad Data müssen andere Verfahren angewandt werden [AE04].

2.2.4 State Estimation in unterbestimmten Systemen: Adaptive State Estimation

Das in Unterabschnitt 2.2.2 vorgestellte State-Estimation-Verfahren ist, wie die meisten anderen Verfahren auch, limitiert auf exakt oder überbestimmte Stromsysteme. In einem exakt bestimmten Stromsystem gibt genau so viele Messwerte wie nötig um die Zustandsvariablen eindeutig zu bestimmen. Sind mehr Messwerte vorhanden als benötigt, so spricht man von einem überbestimmten Stromsystem. Entsprechend gibt es bei einem unterbestimmten Stromsystem weniger Messwerte als benötigt um alle Zustandsvariablen zu bestimmen [KML15].

Insbesondere Verteilnetze sind in der Regel unterbestimmte Stromsysteme mit den folgenden, daraus resultierenden Herausforderungen [Hua+12]:

- Die Anzahl an verfügbaren Echtzeitmesswerten ist stark limitiert und nicht genug um das System im Ganzen überwachen zu können.
- Die zusätzlich verwendeten Pseudomesswerte, die hauptsächlich aus historischen Lastprofilen gewonnen werden, sind ungenau.
- Statt Leistungsdaten (P_i und Q_i) stehen oftmals Strommesswerte an den Sammelschienen zur Verfügung, was einen Einfluss auf die verwendeten Messwertegleichungen (siehe Unterabschnitt 2.2.1 hat.

Entsprechend werden Algorithmen speziell für unterbestimmte Stromsysteme entwickelt [Hua+12]. Ein in dieser Arbeit verwendeter Algorithmus aus dieser Kategorie ist der Adaptive State Estimator (ASE) [KL12; KML15], der einige für diese Arbeit nützliche Eigenschaften besitzt.

Anstelle einer Cholesky-Zerlegung (vgl. Unterabschnitt 2.2.2), verwendet der ASE eine Singulärwertzerlegung um eine Pseudoinverse der Verstärkungsmatrix $\boldsymbol{G}(\boldsymbol{x}^k)$ zu erstellen. Diesen Ansatz verfolgen zwar auch andere Arbeiten, diese zielen allerdings in der Regel darauf ab, den Teil eines unterbestimmten Stromsystems zu identifizieren, der vollständig beobachtbar ist, und sich anschließend auf die vollständige Beobachtbarkeit einzelner Zustandsvariablen zu konzentrieren [KML15]. Mit dem ASE ist es allerdings möglich, eine State Estimation für alle Bereiche eines unterbestimmten Stromsystems durchzuführen. Zwar können die Zustandsvariablen nur in den exakt oder überbestimmten Systemteilen mit einer gewissen

Genauigkeit geschätzt werden, allerdings ist eine explizite Beschränkung auf diese Bereiche vor der State Estimation nicht notwendig. Der ASE berechnet zusätzlich eine Modellabdeckung für eine Zustandsvariable. Dies ist ein Winkel zwischen $0°$ und $90°$. $0°$ bedeuten dabei, dass die Zustandsvariable durch die Messwerte vollständig bestimmt werden konnte, während $90°$ bedeuten, dass die Zustandsvariable unbestimmt ist [KML15]. Diese Modellabdeckung kann kann auch für eine Trust-Erhebung verwendet werden. Je größer der Winkel ist, desto weniger konnte die Zustandsvariable durch die Messwerte bestimmt werden und als desto geringer kann der Trust in die Zustandsvariable eingeschätzt werden.

Zwei weitere, für diese Arbeit nützliche Eigenschaften sind die folgenden. Erstens berechnet der ASE neben den Zustandsvariablen noch Unsicherheiten in die Zustandsvariablen auf Basis der Standardabweichungen der Messwerte. Zweitens wird die Jacobimatrix $\boldsymbol{H}$ nicht zentral aufgestellt, sondern reihenweise durch die entsprechenden Messwerte. Das macht den ASE flexibel ggü. Änderungen bei den zur Verfügung stehenden Messwerten.

2.3 Multivariater Trust

Ein anerkanntes multivariates Trust-Modell aus dem Bereich des Organic Computing (OC) stellt OC-Trust [Ste+10; SR12] dar, das in diesem Abschnitt vorgestellt wird. Auf OC-Trust basiert auch die Definition 1 von Trust in Abschnitt 1.1. Im OC werden zumeist Multiagentensysteme betrachtet, in denen Agenten mit anderen Agenten aber auch mit Menschen kommunizieren und interagieren. Im Rahmen von OC-Trust wird Trust verstanden als „ein subjektives Konzept, das alle an einem System teilnehmenden Komponenten und Benutzer mit einbezieht und Kooperation zwischen den Elementen verteilter Systeme ermöglicht. Es erlaubt den Elementen das Vertrauen, das sie in ihre Interaktionspartner in einem bestimmten Kontext haben, zu bemessen und entwickelt sich mit den Erfahrungen der Elemente im Laufe der Zeit fort“ [Ste+10; SR12]. Dabei tragen zum Trust mehrere Trust-Facetten bei, die im Folgenden definiert werden.

Definition 4 (Funktionale Korrektheit)**.** *„Die Eigenschaft eines Systems, seiner funktionalen Spezifikation zu entsprechen, unter der Bedingung, dass keine unvorhergesehenen Störungen in der Umgebung des Systems auftreten“ [Ste+10; SR12].*

Definition 5 (Betriebssicherheit)**.** *„Die Eigenschaft eines Systems, zu keiner Zeit in einen Zustand einzutreten oder einen Output zu erzeugen, in dem oder durch den*

das System seine Benutzer, sich selbst oder Teile von sich oder seine Umwelt schädigt“ [Ste+10; SR12].

Definition 6 (Datensicherheit[3])**.** *„Die Abwesenheit von Möglichkeiten, das System in einer Weise zu verwenden, die dazu führt, dass private Informationen preisgegeben werden, Daten ohne Autorisierung gelöscht oder verändert werden, oder die es erlaubt, unberechtigterweise im Namen von anderen mit dem System zu interagieren“ [Ste+10; SR12].*

Definition 7 (Zuverlässigkeit)**.** *„Die Eigenschaft eines Systems, selbst bei auftretenden Störungen oder teilweisen Ausfällen für eine spezifizierte Zeit verfügbar zu bleiben“ [Ste+10; SR12].*

Definition 8 (Glaubwürdigkeit)**.** *„Der Glaube an die Fähigkeit und den Willen eines Kooperationspartners, an einer Interaktion in einer vorteilhaften Weise teilzunehmen. Außerdem die Fähigkeit eines Systems, mit einem Benutzer konsistent und transparent zu kommunizieren“ [Ste+10; SR12].*

Definition 9 (Gebrauchstauglichkeit[4])**.** *„Die Eigenschaft eines Systems, eine Benutzerschnittstelle anzubieten, die vom Benutzer effizient, effektiv und zu seiner Zufriedenheit bedient werden kann, insbesondere unter Berücksichtigung von Benutzerkontrolle und Privacy“ [Ste+10; SR12].*

Die sechs Trust-Facetten können nach den Autoren von [Ste+10; SR12] in solche unterteilt werden, die Trust a-priori bewerten können (funktionale Korrektheit, Betriebs- und Informationssicherheit), und solche, die dieses zur Laufzeit tun (Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit). Letzteres geschieht auf Basis von Erfahrungen durch die Interaktionen mit der entsprechenden Komponente.

Abbildung 2.5 zeigt, wie im Rahmen von OC-Trust Trust-Werte entstehen und von Agenten verwendet werden. „Agenten wählen anhand von Trust-Werten (oder zufällig) einen Interaktionspartner aus. In einer ersten Interaktion entsteht dann ein Kontrakt, der z. B. die Bedingungen einer Leistungserbringung oder eine Verfügbarkeitszusage enthält. Im Folgenden wird geprüft, ob der Kontrakt eingehalten wurde und daraus eine Erfahrung abgeleitet. Die gesammelten Erfahrungen dienen dann dazu, mithilfe einer Trust-Metrik einen Trust-Wert für eine der Trust-Facetten abzuleiten. Dieser wird dann wiederum zur Entscheidungsfindung in den Algorithmen der Applikationen verwendet“ [SR12].

[3] In dieser Arbeit Informationssicherheit genannt.
[4] In dieser Arbeit Bedienbarkeit genannt.

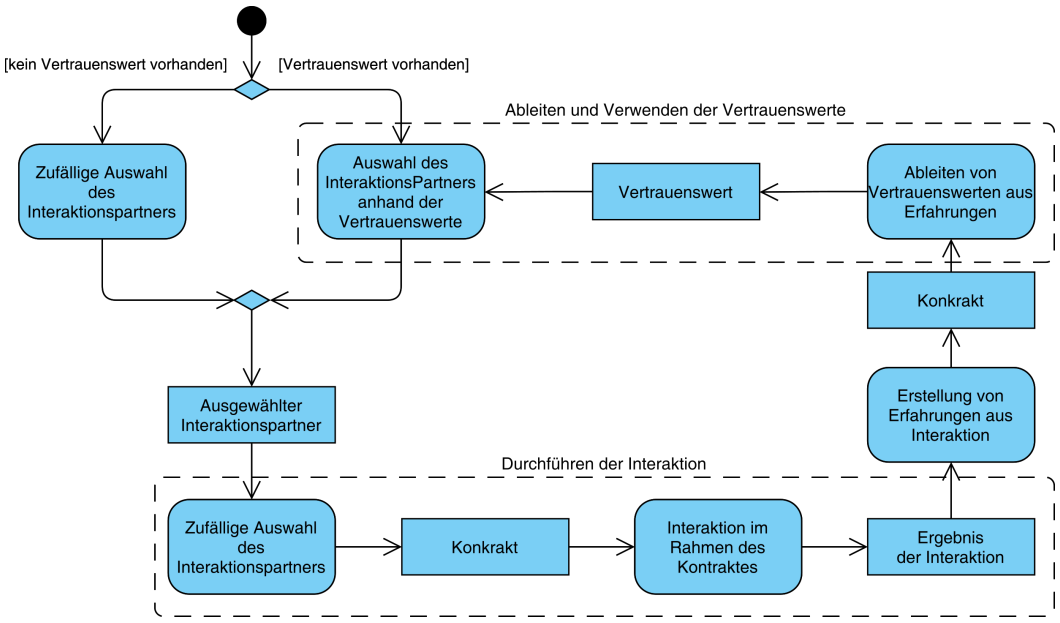


Abb. 2.5.: Entstehung und Verwendung von Trust bei OC-Trust [SR12].

Zusammenfassend betrachtet OC-Trust direkten Trust und Reputation in Netzwerken von technischen und menschlichen Akteuren (Agenten). Ein CPES, hierarchisch oder verteilt organisiert, kann ebenfalls als ein solches Netzwerk betrachtet werden, wodurch sich viele Konzepte aus OC-Trust übertragen lassen.

Da OC-Trust einen domäneninvarianten Ansatz zur Trust-Modellierung darstellt, abstrahiert OC-Trust von technischen Systemen, die die Akteure verbinden, und den ausgetauschten Prozessdaten. Für eine domänenspezifische Arbeit, wie die vorliegende, kann der Ansatz jedoch konkretisiert und domänenspezifisch weiterentwickelt werden, um die genannten technischen Systeme und Prozessdaten zu berücksichtigen. Dies ist insofern relevant, als dass jedes technische System, das z.B. für den Datenaustausch benötigt wird, zusätzliche Schwachstellen für Manipulationen oder ein Rauschen einspielen kann, die den Trust in die übermittelten Prozessdaten negativ beeinflussen. Beispiele dafür wurden u.a. mit Man-in-the-Middle-Angriffen, nicht zielgerichteter Malware oder Überlastung bereits in Abschnitt 1.1 und insbesondere Abbildung 1.1 aufgezeigt.

2.4 Datenstrommanagementsysteme

Nach den Definitionen 2 und 3 aus Abschnitt 1.3.1 ist ein Datenstrommanagementsystem (DSMS) ein System, das Daten ereignisbasiert und im Hauptspeicher verarbei-

tet. Dabei werden die Daten nur so lange vorgehalten, wie unbedingt notwendig, und nur einmal verarbeitet [SÇZ05]. Außerdem verfügt ein DSMS über Eigenschaften eines Datenbankmanagementsystems (DBMS), wie z.B. eine Anfrageverwaltung, vordefinierte Operatoren, Anfrageoptimierung und Zugriffskontrolle [CM12; Gei13].

Auf dieser Grundlage beschreibt dieser Abschnitt die für diese Arbeit wichtigsten Eigenschaften und Merkmale der Datenstromverarbeitung. Dabei wird zunächst in Unterabschnitt 2.4.1 auf Datenströme und Anfragen an ein DSMS eingegangen. Unterabschnitt 2.4.2 widmet sich im Anschluss sogenannter Fensteransätze.

2.4.1 Datenströme und kontinuierliche Anfragen

Für die Datenquellen von DSMSs wird angenommen, dass sie nicht unter der Kontrolle des DSMS sind. Die Daten, die eine aktive Datenquelle sendet, werden als Datenstrom bezeichnet. Ein Datenstrom ist dabei eine kontinuierliche, geordnete und potentiell unendliche Folge von flüchtigen Datenstromelementen [GÖ03]. Die Ordnung ist entweder explizit durch Zeitstempel in den eingehenden Datenstromelementen gegeben oder implizit, indem das DSMS die Datenstromelemente beim Eintreffen mit Zeitstempeln versieht. Ausgaben von DSMSs sind ebenfalls Datenströme [Krä07]. Das Paradigma der Datenstromverarbeitung kann mit dem von Datenbanken verglichen werden. In Datenbanken können die Daten im Vergleich zu den ad hoc gestellten Anfragen als statisch angesehen werden. In DSMSs sind die Daten vielmehr flüchtig und Anfragen langlaufend. Ein Beispiel dafür ist die kontinuierliche Berechnung der Durchschnittstemperatur auf einem Datenstrom von einem Temperatursensor [GÖ03]. Im Gegensatz zu kurzlebigen Anfragen an ein DBMS sind Anfragen an ein DSMS langlebig. Einmal installiert, werden dabei so lange kontinuierlich Ergebnisse beim Eintreffen neuer Datenstromelemente generiert, bis die kontinuierliche Anfrage entfernt wird.

Da nicht alle Datenstromelemente persistent gespeichert werden können, wird ihre Verweildauer in einem DSMS beschränkt. Dies hat zur Folge, dass die Ergebnisse von kontinuierlichen Anfragen meistens nicht exakt sind, sondern approximativ. Dies genügt allerdings den meisten Anwendungen, wenn es Approximationen von hoher Güte sind. Außerdem werden ältere Daten im Kontext von Datenströmen als nicht so wichtig erachtet wie neuere. Ursachen für die fehlende Exaktheit sind u.a. die folgenden [Krä07]:

- Viele kontinuierliche Anfragen sind nicht mit einem endlichen Hauptspeicher zu beantworten (z.B. ein kartesisches Produkt zweier unendlicher Datenströme).

- Einige Operationen aus der relationalen Algebra sind blockierend, da sie die gesamte Eingabe einsehen müssen, bevor sie ein Ergebnis produzieren (z.B. eine Aggregation).
- Es kann vorkommen, dass die aktiven Datenquellen schneller Daten an das DSMS senden, als das DSMS diese verarbeiten kann.

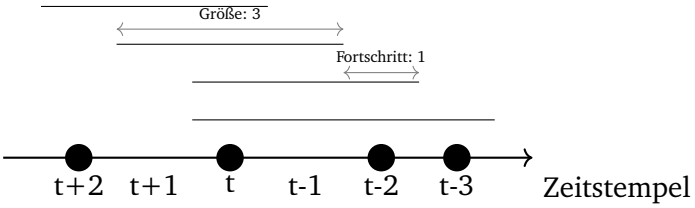
Eine weit verbreitete Technik, um Informationen über Datenstromelemente, vor allem (Teil-) Ergebnisse, im DSMS zu behalten, stellen Fensteransätze dar [ABW02], die im folgenden Unterabschnitt beschrieben werden.

2.4.2 Fensteransätze

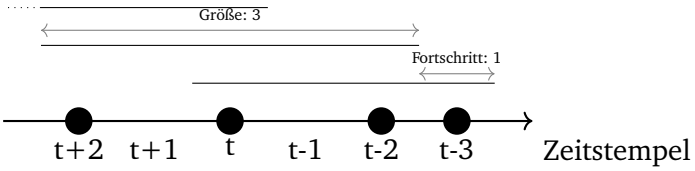
Aufgrund der potentiellen Unendlichkeit von Datenströmen können nicht alle eingehenden Datenstromelemente persistent gespeichert werden. Aus diesem Grund werden oft Fensteransätze verwendet, um einen endlichen Ausschnitt aus einem Datenstrom zu betrachten. Dabei wird zunächst darin unterschieden, ob und welche Randpunkte eines Fensters fixiert werden. Fixiert man nur den Startpunkt eines Fensters, so wächst dieses unendlich an. Fixiert man nur den Endpunkt eines Fensters, so werden alle Elemente vom Starten der Anfrage bis zu dem gewünschten Fensterende betrachtet. Werden beide Randpunkte eines Fensters fixiert, so befinden sich immer dieselben Elemente in einem Fenster [ABW02].

Am häufigsten werden aber so genannte gleitende Fenster (engl. sliding windows) verwendet, bei denen weder der Start- noch der Endpunkt fixiert sind und sich der Datenstrom durch das Fenster bewegt. Der Abstand zwischen den beiden Randpunkten wird dabei als Fensterbreite w bezeichnet. Die Anzahl an Datenstromelementen oder Zeiteinheiten, die pro Schritt neu in das Fenster aufgenommen werden, wird Fortschritt a genannt (analog verlassen bei einem gleitenden Fenster pro Schritt ebenfalls a Datenstromelemente bzw. Zeiteinheiten das Fenster). Breite und Fortschritt können sowohl in Datenstromelementen als auch in Zeiteinheiten gemessen werden. Typischerweise gilt für gleitende Fenster die Regel $1 \leq a < w$, so dass einzelne Datenstromelemente in mehreren Fenstern verarbeitet werden. Es gibt allerdings auch sogenannte taumelnde Fenster (engl. tumbling windows), bei denen $a \geq w$ gilt. Dies bedeutet, dass jedes Datenstromelement maximal in einem Fenster verarbeitet wird (für $a > w$ gibt es Elemente, die überhaupt nicht verarbeitet werden) [GÖ03].

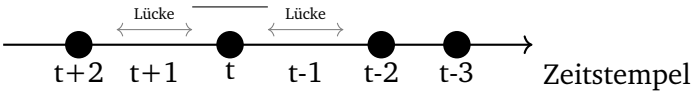
Beispiele für übliche Fenster sind in Abbildung 2.6 dargestellt.In Abbildung 2.6a wird der sogenannte Zeitfensteransatz verfolgt. Dabei bemessen sich die Breite und der



(a) Beispielhafte Zeitfenster mit einer Länge von drei Zeiteinheiten und einem Fortschritt von einer Zeiteinheit.



(b) Beispielhafte Elementfenster mit einer Länge von drei Elementen und einem Fortschritt von einem Element.



(c) Beispielhafte Sessionfenster mit einer erlaubten Inaktivitätsdauer von weniger als einer Zeiteinheit.

Abb. 2.6.: Beispiele für übliche Fenster in der Datenstromverarbeitung (eigene Darstellung). Der Datenstrom fließt dabei von links nach rechts und der aktuelle Verarbeitungspunkt ist t. Somit handelt es sich bei allen Elementen rechts von t um bereits verarbeitete und bei allen links von t um noch zu verarbeitende Elemente. Die Striche über der Zeitachse repräsentieren alle Fenster im jeweiligen Ansatz, zu denen das Element mit dem Zeitstempel t gehört.

Fortschritt eines Fensters nach Zeiteinheiten. Der Fortschritt eines Fensters ist dabei das Maß, wann ein neues Fenster startet. Es ist zu sehen, dass es in diesem Beispiel vier Zeitfenster gibt, zu denen das Element zum Zeitpunkt t gehört. Der sogenannte Elementfensteransatz ist in Abbildung 2.6b dargestellt. Die Breite und der Fortschritt eines Fensters bemessen sich analog zu Zeitfenstern nach einer bestimmten Anzahl von Elementen. Es ist zu sehen, dass es in diesem Beispiel drei Elementfenster gibt, zu denen das Element zum Zeitpunkt t gehört. Ein weiterer Ansatz, der hier vorgestellt werden soll, ist der der sogenannten Sessionfenster (Abbildung 2.6c). Er unterscheidet sich von den beiden vorigen Ansätzen insofern, als dass die Fenster keine feste Größe und keinen festen Fortschritt haben. Vielmehr wird ein neues Fenster für ein Element erstellt, wenn zurzeit keines offen ist. Geschlossen wird das Fenster bei anhaltender Inaktivität, d.h. es kommen für einen bestimmten Zeitraum keine neuen Datenstromelemente hinzu. Wichtig ist dabei, dass im Gegensatz zu Zeitfenstern die Systemzeit ausschlaggebend ist und nicht die Zeit, die durch die

Zeitstempel der Datenstromelemente vorgegeben wird. In dem Beispiel in Abbildung 2.6c ist von einer sehr kurzen erlaubten Inaktivitätsdauer auszugehen, so dass die kleinen Lücken im Datenstrom dazu führen, dass das Element mit dem Zeitstempel t ein einelementiges Fenster bildet.

Prädikatfenster und partitionierende Fenster sind weitere Ansätze. Bei Prädikatfenstern gibt ein komplexes Prädikat vor, welche Datenstromelemente sich in dem Fenster befinden sollen [GAE06]. Als ein Beispiel sei hier ein Datenstrom angedacht, der von einem Kfz-Sensor gesendet wird. Möchte man z.B. die Durchschnittsgeschwindigkeit der letzten $100\ km$ wissen, eignen sich weder zeit- noch elementbasierte Fenster, da die Position ein Attribut ist und aus ihr keine Zeitpunkte oder Elementanzahlen abgeleitet werden können. Hier eignet sich ein Prädikatfenster. Partitionierende Fenster hingegen bauen auf elementbasierten Fenstern auf. Dabei werden die Datenstromelemente zunächst anhand eines Gruppierungsattributs in Gruppen unterteilt (analog zu der Gruppierung bei Aggregationen). Für jede Gruppe kommt dann ein elementbasiertes Fenster zum Einsatz [Krä07].

2.5 Zusammenfassung

In diesem Kapitel wurden die wesentlichen Grundlagen der vorliegenden Arbeit vorgestellt: SCADA-Systeme, State Estimation und Bad Data Detection, multivariater Trust sowie DSMSs. SCADA-Systeme (Abschnitt 2.1) sind „eine Sammlung von Gerätschaften, die einen Operateur mit genügend Informationen versorgt, um den Status eines bestimmten Gerätes oder Prozesses in einer Fernwirkstation zu bestimmen und Aktionen bzgl. dieses Gerätes oder Prozesses zu veranlassen, ohne körperlich anwesend zu sein “ [TM15] (aus dem Englischen übersetzt). Ein relevanter Prozess in SCADA-Systemen ist der der Datenakquise und Überwachung. Dabei werden Daten im Feld gesammelt, in ein Übertragungsformat konvertiert, in Datenpakete zusammengefasst und an das Kontrollzentrum übertragen. Dort werden die Datenpakete entsprechend dekodiert und adäquat visualisiert [TM15]. Die wesentlichen Komponenten eines SCADA-Systems (Teilabschnitt 2.1.1) sind RTUs als Sammelpunkt im Feld für zu übertragende Messwerte oder auszuführende Steuerbefehle, die Masterstation, mit deren Mensch-Maschine-Schnittstelle die Operateure in der Netzleitwarte das System überwachen und steuern, und ein IKT-System zur Übertragung der Datenpakate [TM15]. SCADA-Systeme haben neben der State Estimation noch weitere Funktionen, die in Teilabschnitt 2.1.2 vorgestellt wurden. Eine Übersicht der zur Anwendung kommenden Protokolle ist Tabelle 2.1

in Teilabschnitt 2.1.3 zu entnehmen. Besondere Relevanz für diese Arbeit hat dabei der 104er-Standard (Teilabschnitt 2.1.4). Er basiert auf TCP/IP und im 104er übertragene Telegramme enthalten immer eine APDU. Eine APDU setzt sich aus einer APCI und je nach Steuerungsinformationen einer ASDU zusammen. Bezüglich der Steuerungsinformationen gibt es drei verschiedene Formate: das I-Format, das S-Format und das U-Format. Im I-Format werden Informationen in Form einer ASDU transferiert. Das S-Format enthält keine Informationen über Prozessdaten und wird lediglich als Antwort auf Telegramme im I-Format verwendet. Das U-Format dient zur Initiierung oder Beendigung des Datentransfers oder zum Versenden von Testnachrichten [Int16; Int15].

Die State Estimation (Abschnitt 2.2) ist eine Schätzung der Zustandsvariablen (der komplexen Spannungen an den Knotenpunkten im Stromnetz) auf Basis des Netzwerkmodells und der zur Verfügung stehenden Messwerte [AE04]. Dabei spielt der Zusammenhang zwischen den Zustandsvariablen und den Messwerten eine große Rolle (Gleichung 2.1 in Teilabschnitt 2.2.1). Es handelt sich bei der State Estimation um ein Minimierungsproblem bezüglich der Messwertefehler. Ein häufig angewandtes Verfahren ist dabei die State Estimation mit gewichteten kleinsten Quadraten (Teilabschnitt 2.2.2). Bei dem Verfahren kommen u.a. eine Maximum-Likelihood-Schätzung sowie das Gauß-Newton-Verfahren zum Einsatz [AE04]. Das Ziel einer Bad Data Detection (Teilabschnitt 2.2.3) im Anschluss an eine State Estimation ist das Erkennen, Identifizieren und Eliminieren von groben Messwertefehlern auf Grundlage der Messwerteresiduen. Da das System nach dem Entfernen der Bad Data noch überwachbar sein muss, kann Bad Data nur für redundante Messwerte erkannt werden [AE04].

Als Grundlage für den Trust-Begriff in dieser Arbeit (Abschnitt 2.3) wird mit OC-Trust ein anerkanntes multivariates Trust-Modell aus dem Bereich des OC verwendet. Im Rahmen von OC-Trust wird Trust verstanden als „ein subjektives Konzept, das alle an einem System teilnehmenden Komponenten und Benutzer miteinbezieht und Kooperation zwischen den Elementen verteilter Systeme ermöglicht. Es erlaubt den Elementen das Vertrauen, das sie in ihre Interaktionspartner in einem bestimmten Kontext haben, zu bemessen und entwickelt sich mit den Erfahrungen der Elemente im Laufe der Zeit fort“ [Ste+10; SR12]. Dabei tragen zum Trust die folgenden sechs Trust-Facetten bei: funktionale Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit [Ste+10; SR12]. OC-Trust ist allerdings domäneninvariant, so dass es weder die technischen Systeme, die die Akteure verbinden, noch die ausgetauschten Prozessdaten miteinbezieht.

DSMSs (Abschnitt 2.4) sind Systeme, die Daten ereignisbasiert und im Hauptspeicher verarbeiten. Dabei werden die Daten nur so lange vorgehalten, wie unbedingt notwendig, und nur einmal verarbeitet [SÇZ05]. Außerdem verfügt ein DSMS über Eigenschaften eines DBMS, wie z.B. eine Anfrageverwaltung, vordefinierte Operatoren, Anfrageoptimierung und Zugriffskontrolle [CM12; Gei13]. Im Gegensatz zu kurzlebigen Anfragen an ein DBMS sind Anfragen an ein DSMS (Teilabschnitt 2.4.1) langlebig. Sie werden einmal installiert und verarbeiten die flüchtigen Datenstromelemente kontinuierlich und i.d.R. über einen langen Zeitraum hinweg [Krä07]. Es werden oft Fensteransätze (Teilabschnitt 2.4.2) verwendet, um einen endlichen Ausschnitt aus einem Datenstrom zu betrachten [ABW02]. Fenster lassen sich dabei anhand der Fixierung von Start- und Endpunkt und deren Grundlage (Elemente, Zeit oder Prädikate) unterscheiden [GÖ03].

3 Trust-Modell

> *Vertrauen ist selbst ein Begriff für eine Anhäufung von Bedeutungen.*
>
> — **Harrison White**

Dieses Kapitel widmet sich dem ersten Forschungsziel, d.h. einem kontextsensitiven, multivariaten Trust-Modell auf Grundlage von OC-Trust, das in Abschnitt 2.3 vorgestellt wurde. Dabei werden zunächst in Abschnitt 3.1 verwandte Arbeiten zu dem Thema vorgestellt, bevor in Abschnitt 3.2 ein kontextsensitives und multivariates Trust-Modell für CPESs eingeführt wird. Das Kapitel schließt mit einer Zusammenfassung in Abschnitt 3.3.

3.1 Verwandte Arbeiten

In der Literatur wird der Trust-Begriff mit sehr unterschiedlicher Bedeutung belegt bzw. in der Regel nicht definiert. Alle gefundenen Arbeiten haben allerdings gemein, dass sie Trust im Rahmen von Multiagentensystemen betrachten.

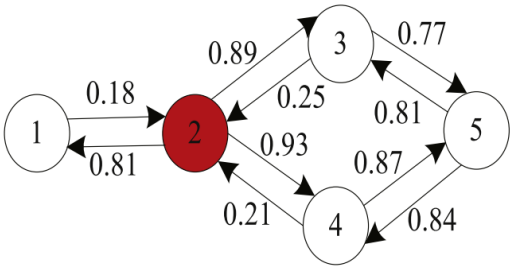


Abb. 3.1.: Ein Beispiel für Trust von Agenten an Sammelschienen in Nachbaragenten nach [Xie+19].

Oft wird ein univariates Trust-Modell zugrunde gelegt, bei dem sich Trust auf Abweichungen der Prozessdaten von benachbarten Agenten zu den eigenen Prozessdaten bezieht (u.a. [LL18; MX20; Mus+20; Xie+19]). Prozessdaten können dabei sowohl Messwerte als auch bei einer verteilten State Estimation geschätzte Zustandsvariablen sein. Abbildung 3.1 zeigt ein Beispiel für Trust von Agenten in

Nachbaragenten [Xie+19]. Die Graphstruktur entspricht dabei der Topologie des physischen Systems mit Agenten (Knoten) an den Sammelschienen. Der Wertebereich $[0, 1]$ für einen Trust-Wert ist typisch für die Anwendung von Trust in dieser Domäne. Ein univariates Trust-Modell hat folgende Nachteile. Entweder wird Trust nur an einer Information festgemacht, hier die Abweichung der Prozessdaten, oder unterschiedliche Informationen werden aggregiert, wodurch nur ein einzelner Trust-Wert entsteht. Im ersten Fall fehlt es an Flexibilität und Erweiterbarkeit. Auch bleibt die Frage unbeantwortet, wie kritische Messwerte als Prozessdaten von anderen Agenten falsifiziert werden können. Im zweiten Fall gibt es keinerlei Möglichkeit der Nachvollziehbarkeit bzw. Rückverfolgung von Trust-Werten. Auch ist nicht jede Trust-Facette in jeder Situation gleichbedeutend. Die Möglichkeit einer Gewichtung z.B. durch den Endnutzer wäre hier nicht möglich.

Bei OC-Trust, vorgestellt in Abschnitt 2.3, handelt es sich um ein Trust-Modell, das bereits vieles enthält, das für diese Arbeit relevant ist, vor allem Kontextsensitivität und Multivarietät. Allerdings basieren bei OC-Trust die Trust-Werte oft auf Erfahrungen mit Agenten. Im Rahmen der vorliegenden Arbeit geht es aber vielmehr um eine Trust-Einschätzung auf Basis aktueller Informationen über den Agenten bzw. die Komponente oder das Prozessdatum. Eine aktuelle Fehlfunktion oder ein aktueller Angriff lassen sich nicht mit Erfahrungswerten in Trust widerspiegeln. In diesem Zusammenhang erscheint auch die Unterteilung der Trust-Facetten in „a-priori“ und „zur Laufzeit“ nicht sinnvoll. In hochgradig komplexen und dynamischen Systemen sollten nach Möglichkeit alle Trust-Facetten zur Laufzeit bewertet werden. Eine zusätzliche Bewertung a-priori ist dadurch nicht ausgeschlossen.

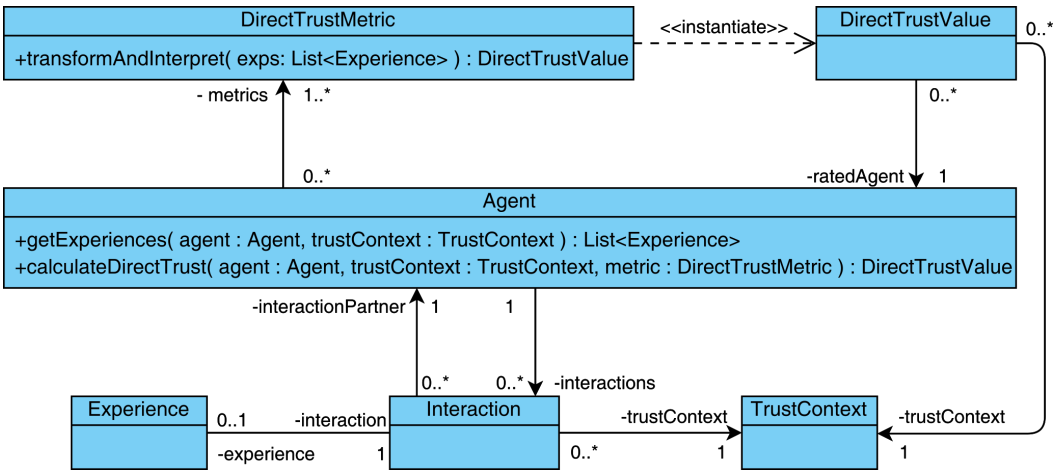


Abb. 3.2.: Das Entwurfsmuster direkten Trusts in OC-Trust nach [And+11] als UML-Klassendiagramm.

Abbildung 3.2 zeigt ein Entwurfsmuster als Unified Modeling Language (UML)-Klassendiagramm, das basierend auf OC-Trust von Anders et al. [And+11] vorgeschlagen wird, um direkten Trust eines Agenten in einen anderen zu erheben. Ein Agent kann dabei verschiedene Metriken verwenden, um jeweils einen Trust-Wert aus den Erfahrungen, die er mit dem entsprechenden anderen Agenten gemacht hat, abzuleiten. Eine Erfahrung basiert dabei auf einer vergangenen Interaktion mit dem anderen Agenten. Sowohl der Trust-Wert als auch die Interaktionen und daraus gewonnenen Erfahrungen sind dabei immer in einem Trust-Kontext zu betrachten [And+11].

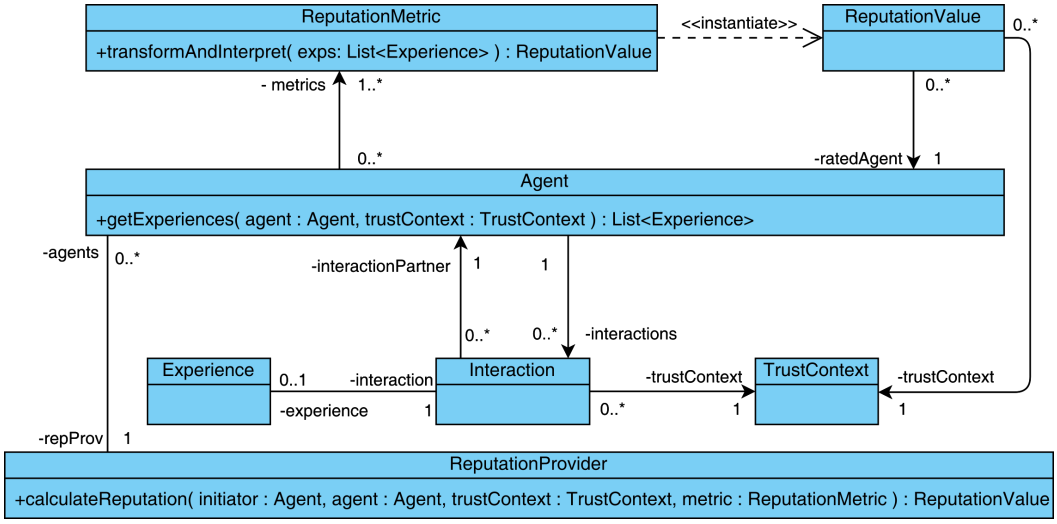


Abb. 3.3.: Das Entwurfsmuster für Reputation in OC-Trust nach [And+11] als UML-Klassendiagramm.

Ein weiteres Entwurfsmuster als UML-Klassendiagramm von Anders et al. [And+11] ist in Abbildung 3.3 zu sehen. In diesem Fall wird ein Trust-Wert durch eine Reputationsmetrik berechnet. Eine Reputation ist eine Aggregation der Erfahrungen anderer Agenten mit dem entsprechenden Agenten und wird von einem Reputationsanbieter berechnet, der zu diesem Zwecke die Erfahrungen aller ihm bekannten Agenten konsultiert [And+11]. Die beiden vorgeschlagenen Entwurfsmuster verdeutlichen noch einmal die zentrale Rolle der Erfahrungen von Agenten mit anderen. Auch wenn die Klasse der Reputationsanbieter als eine Entität, die Trust in einen Agenten auf Basis von Informationen anderer berechnet bzw. schätzt, der in dieser Arbeit verfolgten Idee bereits näher kommt, so stützen sich Reputationsanbieter nach Anders et al. [And+11] doch lediglich auf Erfahrungswerte anderer Agenten und nicht auf externe Informationen wie Angriffsalarme.

Arbeiten von Rosinger et al. [RUS13; RUS14; RB15] basieren ebenfalls auf OC-Trust. Der Anwendungsfall bei Rosinger et al. ist die Bildung dynamischer Wirkleistungs-

verbünde, zu denen sich Erzeuger, Verbraucher und Speicher zusammenschließen. Dabei werden die Koalitionspartner durch Agenten repräsentiert, die sich nicht zwangsläufig kooperativ und gutartig verhalten. Rosinger et al. schlagen daher vor den Trust in die Agenten bei der Koalitionsbildung zu berücksichtigen, um die Wirkleistungsverbünde robuster zu gestalten [RUS13].

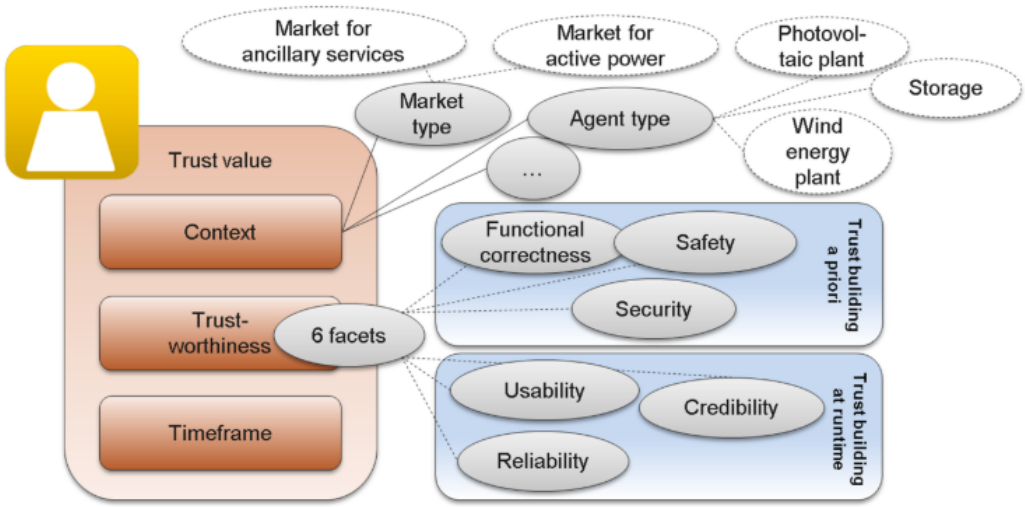


Abb. 3.4.: Das Modell eines Trust-Wertes nach [RUS13; RUS14].

Abbildung 3.4 zeigt das Modell eines Trust-Wertes nach [RUS13; RUS14]. Ein Trust-Wert besteht dabei aus den sechs Trust-Facetten, bekannt aus OC-Trust, sowie einem Kontext und einer Zeitdauer, für die der Trust-Wert gültig ist [RUS13]. Die Zeitdauer kann also als eine Art besondere Kontextinformation angesehen werden.

$$tv = [A_i, A_j, c, tw, t] \tag{3.1}$$

$$tw = [fc, saf, sec, u, cre, rel] \tag{3.2}$$

Gleichung 3.1 [RUS14] stellt den Trust-Wert als ein 5-Tupel dar, wobei A_i den vertrauenden Agenten, A_j den Agenten, dem Vertrauen entgegengebracht wird, c den Kontext, tw den Trust und t die Zeitdauer der Gültigkeit angeben. tw ist dabei nach Gleichung 3.2 [RUS14] ein 6-Tupel mit den sechs Trust-Facetten funktionale Korrektheit (fc), Betriebssicherheit (saf), Informationssicherheit (sec), Bedienbarkeit (u), Glaubwürdigkeit (cre) und Zuverlässigkeit (rel). Rosinger et al. geben ebenfalls an, dass einige oder alle Trust-Facetten zu einem Wert aggregiert werden können, ggf. mit unterschiedlichen Gewichten [RUS13]. Ein besonderes Augenmerk von Rosinger et al. liegt auf der Informationssicherheit [RUS14] und Glaubwürdigkeit [RB15],

wobei im Folgenden die Modellierung der Erfassung der Informationssicherheit vorgestellt wird.

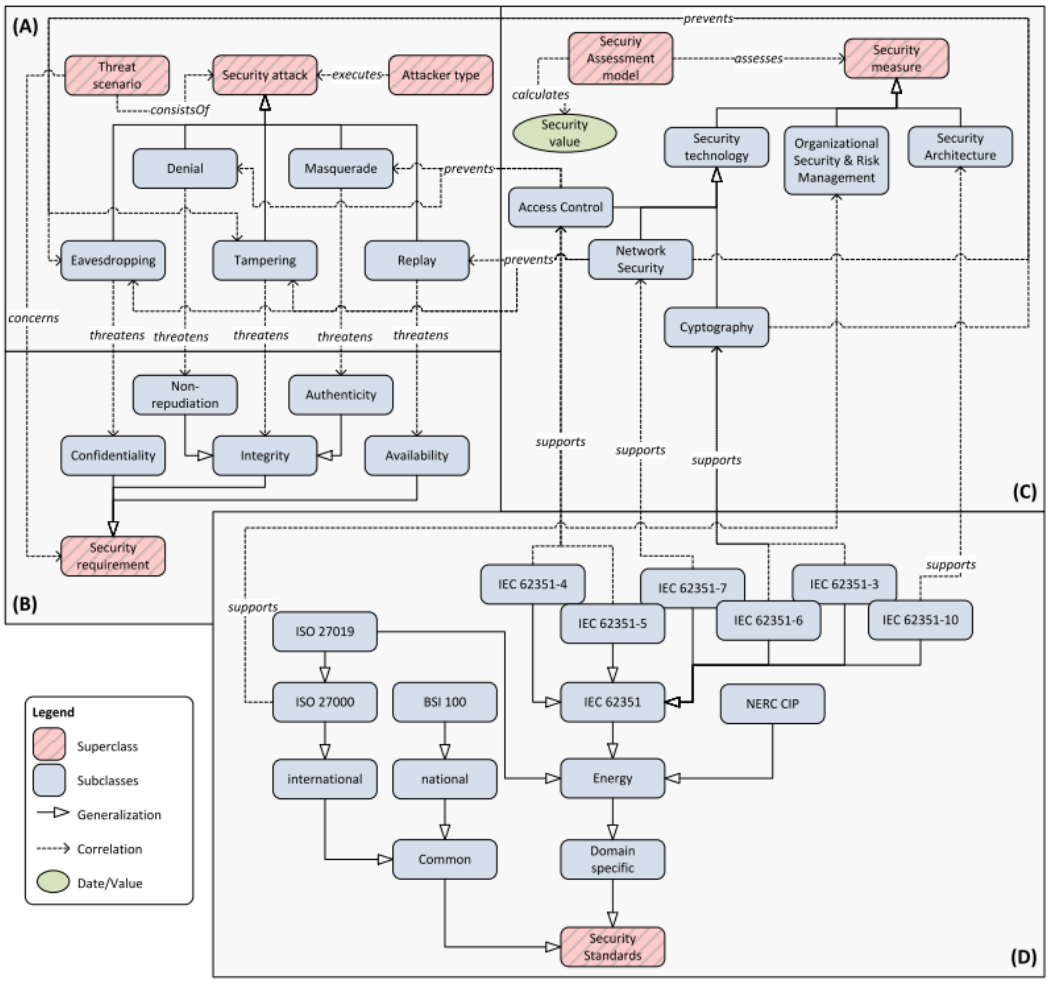


Abb. 3.5.: Das Modell zur Erfassung der a-priori Informationssicherheit von Agenten nach [RUS14].

Das Modell zur Erfassung der Informationssicherheit von Agenten von Rosinger et al. [RUS14] ist Abbildung 3.5 zu entnehmen. Das Modell kann in vier Teile unterteilt werden. In Teil (A) werden Cyberangriffe modelliert. Die von den Angriffen bedrohten Sicherheitsanforderungen befinden sich in Teil (B), während die Sicherheitsmaßnahmen zur Vereitlung der Angriffe in Teil (C) dargestellt werden. Auf Basis dieser Sicherheitsmaßnahmen wird der Trust-Wert für die Informationssicherheit berechnet. Teil (D) zeigt die Sicherheitsstandards auf, die bei der Umsetzung der Sicherheitsmaßnahmen unterstützen. Das in Abbildung 3.5 dargestellte Modell wird entsprechend für jeden Agenten instantiiert, da jeder Agent im Zweifel unterschiedliche Sicherheitsmaßnahmen umsetzt [RUS14]. Die Methode zur Berechnung der Informationssicherheit hat die Überlegung zur Grundlage, dass ein Agent um-

so vertrauenswürdiger ist, je mehr Sicherheitsanforderungen von ihm abgedeckt werden. Entsprechend fließen in die Berechnung der Informationssicherheit ein, welche Sicherheitsanforderungen durch -maßnahmen abgedeckt werden, ob die Umsetzung auf einem Standard basiert und welche Priorität die Sicherheitsanforderung hat [RUS14]. Es handelt sich demnach um eine reine A-Priori-Bewertung der Informationssicherheit und nicht um eine Metrik, die sich zur Laufzeit, z.B. aufgrund von Sicherheitsalarmen, verändert.

3.2 Trust-Modell für cyber-physische Energiesysteme

Anmerkung: Große Teile dieses Abschnitts wurden bereits in [Bra+19b; Bra+20; BBL21] veröffentlicht.

Basierend auf der Definition von OC-Trust (vgl. Abschnitt 2.3) wird Trust für diese Arbeit wie folgt definiert:

Definition 10 (Trust)**.** *Trust ist ein subjektives, kontextabhängiges und multivariates Empfinden gegenüber einer Entität bezüglich ihrer funktionalen Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit (bereits in [Bra+20] veröffentlicht)*[1].

Die Definitionen der sechs Trust-Facetten, auf die die Definition Bezug nimmt, sind dabei den Definitionen 4 bis 9 aus Abschnitt 2.3 oder dem Glossar zu entnehmen. Allerdings gibt es zum einen kontextuelle Unterschiede zwischen Trust in netzdienlichen Operationen in einem CPES und Trust im OC (vgl. Abschnitte 2.3 und 3.1). Zum anderen ist OC-Trust, wie bereits erläutert, domäneninvariant, wodurch von domänenspezifischen Aspekten wie den technischen Systemen, die die Akteure verbinden, oder den ausgetauschten Prozessdaten abstrahiert wird. Daher wird in diesem Abschnitt ein Trust-Modell im Kontext von CPESs, das PSNA-Trust heißt, vorgestellt, das entsprechend eine domänenspezifische Weiterentwicklung von OC-Trust darstellt, indem die technischen Systeme, die die Akteure verbinden, und die ausgetauschten Prozessdaten miteinbezogen werden. Dabei fokussiert PSNA-Trust auf die Erfassung von Trust auf Basis von Echtzeitinformationen zur Laufzeit, um z.B. Angriffe oder Überlastungen erkennen und miteinfließen lassen zu können, in Kombination mit Erfahrungswerten anstatt rein auf die Erfahrung mit Entitäten zu

[1] Es handelt sich dabei um eine Wiederholung von Definition 1 aus Abschnitt 1.1.

fokussieren. Beispiele für solche Echtzeitinformationen sind Angriffsalarme oder Ressourcenauslastungen. Somit stellt PSNA-Trust eine Weiterentwicklung von OC-Trust dar.

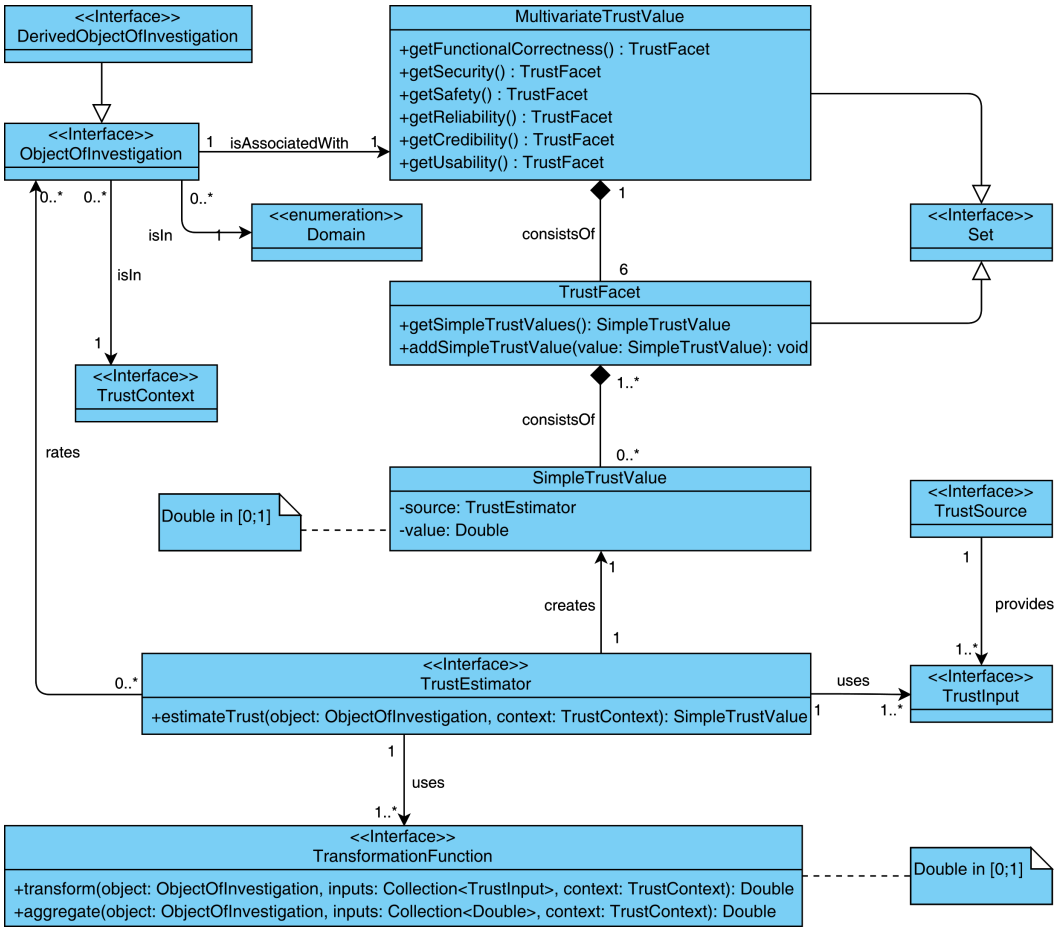


Abb. 3.6.: Das Modell für PSNA-Trust als UML-Klassendiagramm.

Abbildung 3.6 zeigt das Entwurfsmuster für PSNA-Trust als UML-Klassendiagramm, das an die Entwurfsmuster von Anders et al. [And+11] angelehnt ist (vgl. Abbildungen 3.2 und 3.3 in Abschnitt 3.1). Anstelle von Agenten sind es Untersuchungsgegenstände (engl. objects of investigation), deren Vertrauenswürdigkeit erhoben wird und die Inhalt von Unterabschnitt 3.2.1 sind. Die Trust-Erhebung geschieht durch Trust-Schätzer (engl. trust estimators) unter Berücksichtigung des Kontextes (engl. trust context). Die Trust-Schätzer nutzen dabei zum einen Trust-Inputs. Bei den Trust-Inputs handelt es sich um die bereits erwähnten Informationen, wie z.B. Angriffsalarme oder Ressourcenauslastung. Trust-Inputs werden dabei von Trust-Quellen (engl. trust sources), wie beispielsweise einem Intrusion Detection System (IDS) zur Verfügung gestellt. Zum anderen verwenden Trust-Schätzer sogenannte Transformationsfunktionen (engl. transformation functions), die auf die Trust-Inputs

zurückgreifen können, um diese in eine Wahrscheinlichkeit umzuwandeln, mit der der Untersuchungsgegenstand vertrauenswürdig ist. Das Konzept der Trust-Erhebung durch Trust-Schätzer mithilfe von Trust-Inputs, -quellen und Transformationsfunktionen wird in Teilabschnitt 3.2.2 erläutert. Die Ausgaben von Trust-Schätzern stellen einfache Trust-Werte (engl. simple trust values) dar, die Trust-Facetten zugeordnet werden. Die aus Abschnitt 2.3 bekannten Trust-Facetten bilden dann einen multivariaten Trust-Wert (engl. multivariate trust value). Unterabschnitt 3.2.3 widmet sich den Konzepten der einfachen Trust-Werte, Trust-Facetten und multivariaten Trust-Werte.

Anmerkung: Die Teile des Modells, die die Untersuchungsgegenstände, Trust-Inputs, -quellen sowie Transformationsfunktionen umfassen, stammen aus einer gemeinsamen Arbeit mit Kollegen:innen im Forschungsprojekt Smart Grid Cyber-Resilienz Labor[2].

3.2.1 Untersuchungsgegenstände

Anstelle von Agenten, wie sie Gegenstände der Trust-Modelle in der Literatur sind (vgl. Abschnitt 3.1), wird bei PSNA-Trust der Trust in sogenannte Untersuchungsgegenstände oder Entitäten erhoben (oben links in Abbildung 3.6). Die beiden Begriffe werden in dieser Arbeit synonym verwendet. Dabei wird zwischen Untersuchungsgegenständen und abgeleiteten Untersuchungsgegenständen unterschieden. Mit der Schnittstelle für abgeleitete Untersuchungsgegenstände ist es möglich, Untersuchungsgegenstände zu modellieren, für die keine direkten Trust-Informationen vorliegen und deren Vertrauenswürdigkeit über den geschätzten Trust in andere Untersuchungsgegenstände geschätzt wird.

Dies ist insbesondere für die Anwendung der Trust-sensitiven State Estimation von Bedeutung, da für diese der Trust in Messwerte im Fokus steht. Bei der Erhebung des Trusts in Messwerte muss demnach der Datenakquiseprozess untersucht werden. In diesem Prozess werden die Messwerte im Feld gesammelt, konvertiert, zusammengefasst und an das Kontrollzentrum übertragen (vgl. Abschnitt 2.1). Es sind entsprechend Untersuchungsgegenstände wie RTUs, Router, Switches und die Kommunikationskanäle von Interesse. Die Messwerte stellen abgeleitete Untersuchungsgegenstände dar, deren Vertrauenswürdigkeit aus dem Trust in die zuvor genannten Komponenten abgeleitet werden kann.

Die Untersuchungsgegenstände können in Domänen kategorisiert werden, wie in Abbildung 3.6 dargestellt. Domänen haben keinen direkten Einfluss auf die Trust-

[2] Smart Grid Cyber-Resilienz Labor: `https://www.offis.de/offis/projekt/cybreslab.html`

Erhebung, ermöglichen aber verschiedene Perspektiven auf diese. Konkret ermöglichen sie die Auswirkungen von Trust-Einbußen in einer Domäne auf den Trust im Gesamtsystem zu untersuchen. Beispiele für Domänen sind Strom (physisches System), IKT und Markt.

3.2.2 Trust-Erhebung durch Trust-Schätzer

Der Trust in die Untersuchungsgegenstände wird bei PSNA-Trust durch sogenannte Trust-Schätzer unter Berücksichtigung des Kontextes erhoben (siehe unterer Teil in Abbildung 3.6). Die Trust-Schätzer nutzen dabei zum einen Trust-Inputs von -quellen, wie z.B. Angriffsalarme von einem IDS. Zum anderen verwenden Trust-Schätzer sogenannte Transformationsfunktionen, die auf die Trust-Inputs zurückgreifen können, um diese in eine Wahrscheinlichkeit umzuwandeln, mit der der Untersuchungsgegenstand bzgl. einer bestimmten Trust-Facette vertrauenswürdig ist. Es ist auch möglich, dass mehrere Transformationsfunktionen verwendet und deren Ergebnisse im Anschluss durch eine weitere Transformationsfunktion aggregiert werden. Dies findet vor allem bei abgeleiteten Untersuchungsgegenständen Anwendung. Die Untersuchungsgegenstände sowie Trust-Informationen und -quellen werden an dieser Stelle bewusst nicht näher spezifiziert, da hier sehr unterschiedliche Komponenten und Informationen vorstellbar sind. Die Transformationsfunktionen und Trust-Schätzer stellen die Schnittstellen dar, um solch heterogene Daten in das Modell zu integrieren.

Die Trust-Erhebung in einem CPES hat im Wesentlichen die folgenden vier Ziele. Erstens soll der multivariate Trust in einem CPES erhoben werden. Ferner soll eine automatisierte Prüfung auf, zweitens, Vollständigkeit bzgl. der Fragen, ob Trust-Inputs über alle relevanten Untersuchungsgegenstände erfasst und alle Trust-Facetten angesprochen werden, und, drittens, Umsetzbarkeit unter Berücksichtigung der vorhandenen Erhebungsmöglichkeiten (Trust-Schätzer) möglich sein. Viertens soll ein Trust-Wert bis zu den Trust-Inputs und -quellen, die zu der Trust-Einschätzung geführt haben, zurückverfolgt werden können.

Ähnliche Ziele, wenn auch nicht für Trust, wurden bereits in der Literatur verfolgt. Schwarz et al. [Sch+19] entwickelten ein ontologiebasiertes Informationsmodell, um zukünftige Energieszenarien mit Nachhaltigkeitsanalysen zu verbinden. Das Modell besteht aus Simulationsmodellen, Domänen mit Domänenobjekten und -attributen, Nachhaltigkeitskriterien, welche in Facetten gruppiert sind, und Transformationsfunktionen. Letztere stellen die Abhängigkeiten und mathematischen

Beschreibungen für Beziehungen zwischen Domänenattributen und Nachhaltigkeitskriterien bereit [Sch+19]. Der Unterschied zu dem Kontext bei der Trust-Erhebung ist wie folgt. Bei der Trust-Erhebung geht es darum, ein CPES mit Trust-Analysen zu verbinden. Statt Simulationsmodellen sind, im Idealfall, reale Komponenten und Prozesse vorhanden, oder ebenfalls Simulationsmodelle dieser. Die Trust-Facetten entsprechen den Nachhaltigkeitsfacetten und es bedarf ebenfalls Transformationsfunktionen um Eigenschaften des Systems auf Trust-Facetten abzubilden. Auf Grund der Übertragbarkeit der wesentlichen Aspekte des Modells aus [Sch+19] wird für die Modellierung der Trust-Erhebung in CPESs ebenfalls ein ontologiebasiertes Informationsmodell angestrebt.

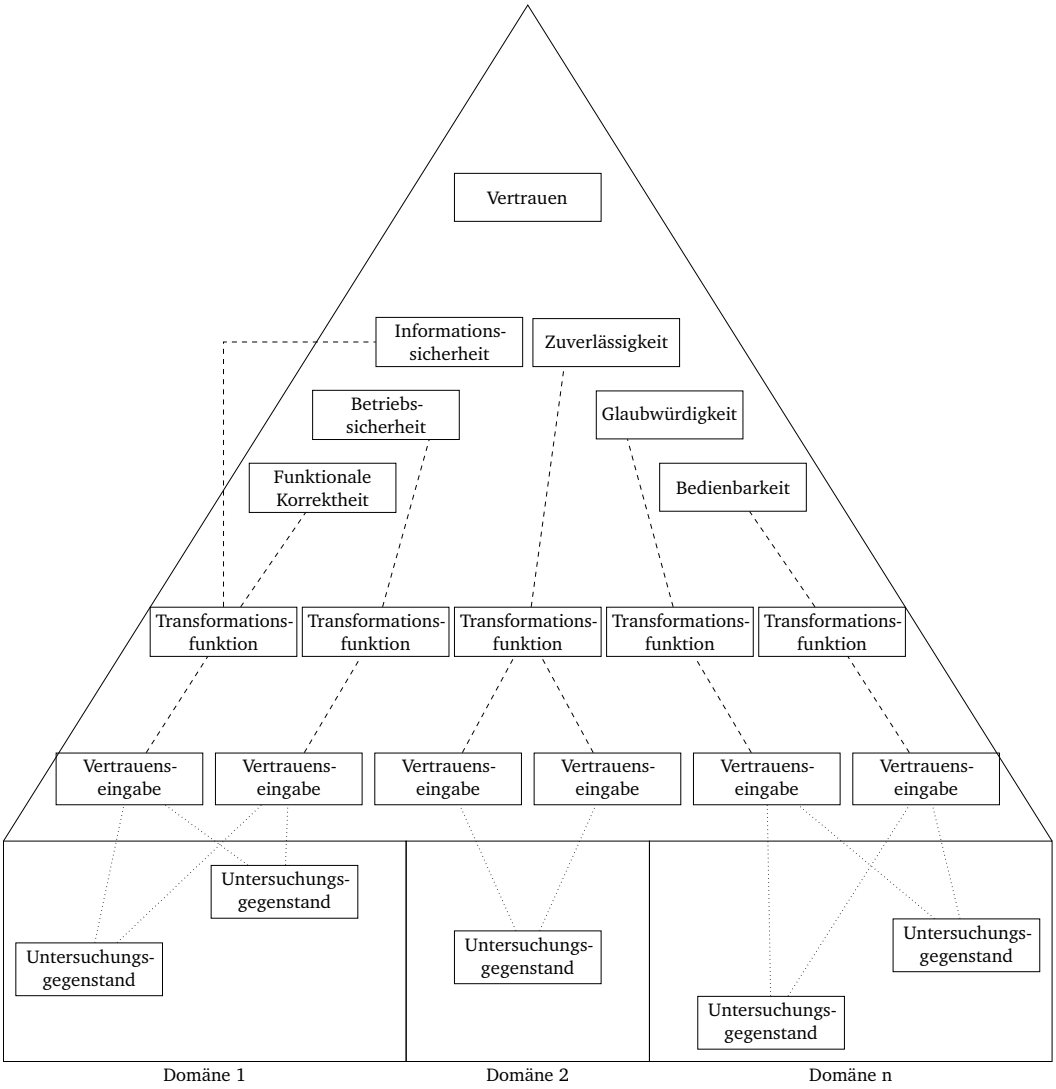


Abb. 3.7.: Die Trust-Erhebung in einem CPES als Trust Assessment Pyramid dargestellt [Bra+20].

Abbildung 3.7 stellt das Informationsmodell zur Trust-Erhebung in Form einer Pyramide, der sogenannten Trust Assessment Pyramid, dar [Bra+20]. Auf der untersten Ebene befinden sich die Untersuchungsgegenstände. Die Untersuchungsgegenstände können, wie in Teilabschnitt 3.2.1 erläutert, in Domänen kategorisiert werden. Auf der Ebene über den Untersuchungsgegenständen befinden sich die Trust-Inputs in Abbildung 3.7. Bei Trust-Inputs handelt es sich um Informationen, die im System gesammelt werden können und die zur Erhebung des Trusts in die Untersuchungsgegenstände beitragen [Bra+20]. Die Trust-Inputs können in ihrer Art, ihrem Umfang und ihrer Updaterate sehr stark variieren. Es handelt sich also um in einem hohen Maße heterogene Informationen. Beispiele für Trust-Inputs sind Ereignisse im Netzwerkverkehr, Ressourcenauslastungen von Komponenten und der Stresslevel von menschlichen Operateuren. Die Verbindungen zwischen Untersuchungsgegenständen und Trust-Inputs sind gepunktet in Abbildung 3.7, da es sich nicht um einen Informationsfluss, sondern um eine Beziehung handelt; entsprechend sind Informationsflüsse gestrichelt dargestellt.

Die nächste Ebene in der Trust Assessment Pyramid in Abbildung 3.7 bilden die Transformationsfunktionen. Wie bei [Sch+19] stellen die Transformationsfunktionen Abhängigkeiten und mathematische Beschreibungen bereit, um Eingangsgrößen auf Trust-Facetten abzubilden. Ausgangsgrößen von Transformationsfunktionen sind dabei Trust-Wahrscheinlichkeiten $p \in \{\mathbb{R}|0 \leq p \leq 1\}$ (siehe Abbildung 3.6). Eine Transformationsfunktion kann auch mehrere Trust-Facetten bedienen, wie beispielsweise die Transformationsfunktion, die sich in Abbildung 3.7 am weitesten links befindet. Die Eingangsgrößen für Transformationsfunktionen können Trust-Inputs, die Ausgaben anderer Transformationsfunktionen oder auch eine Kombination beider sein.

Ein einfaches Beispiel für eine Transformationsfunktion basierend auf Trust-Inputs könnte ein Schwellwertverfahren sein, um aus der Ressourcenauslastung einer

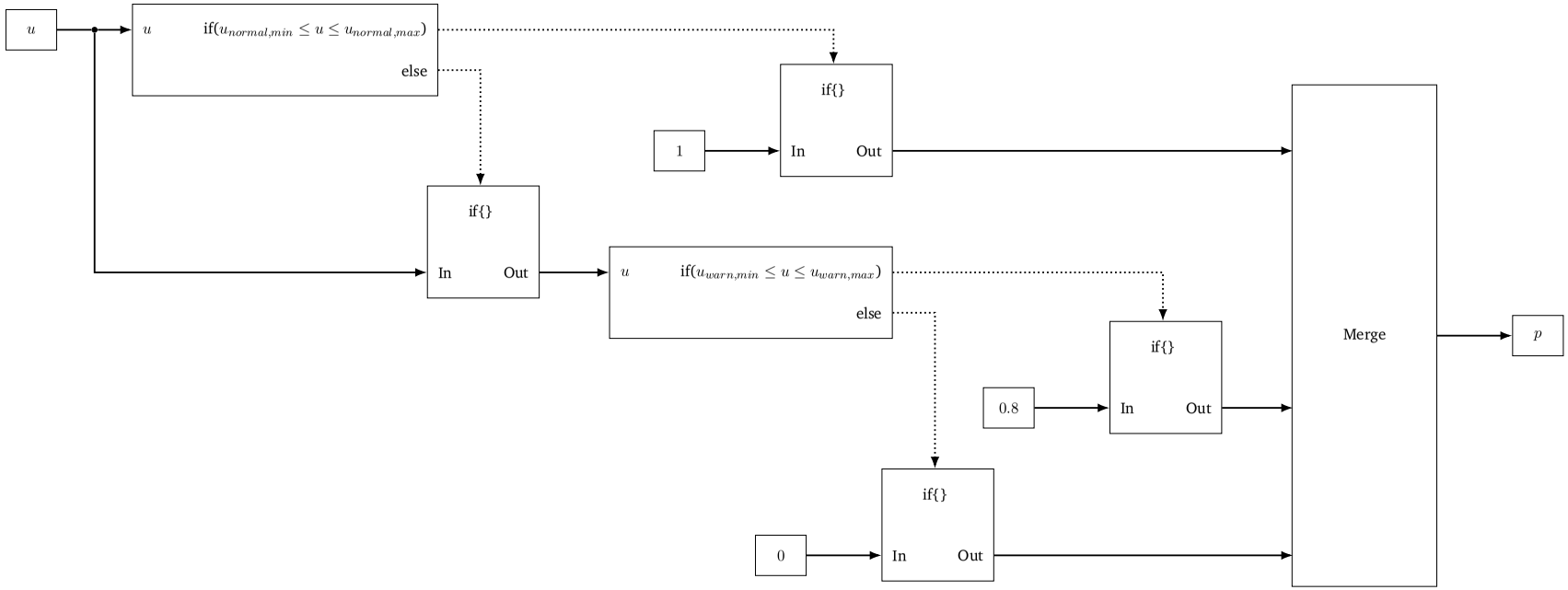


Abb. 3.8.: Eine beispielhafte Transformationsfunktion modelliert als Blockdiagramm. Ist eine Resourcenauslastung u innerhalb eines Intervalls $[u_{normal,min}, u_{normal_max}]$, so ist der geschätzte Trust 1. Für u innerhalb eines größeren Intervalls $[u_{warn,min}, u_{warn_max}]$ wird der Trust auf $0,8$ und ansonsten auf 0 geschätzt.

Komponente einen Aspekt der funktionalen Korrektheit dieser Komponente zu erheben. Eine solche beispielhafte Transformationsfunktion ist in Abbildung 3.8 dargestellt. Ist eine Resourcenauslastung (z.B. CPU oder RAM) u innerhalb eines Intervalls $[u_{normal,min}, u_{normal_max}]$, in dem die Resourcenauslastung als normal klassifiziert wird, so ist der geschätzte Trust 1. Für u innerhalb eines größeren, sogenannten Warn-, Intervalls $[u_{warn,min}, u_{warn,max}]$ wird der Trust auf $0,8$ und ansonsten auf 0 geschätzt. Ein komplexeres Beispiel für eine Transformationsfunktion wäre eine Bad Data Detection (vgl. Unterabschnitt 2.2.3).

Weitere Transformationsfunktionen sind Kapitel A des Anhangs zu entnehmen. Eine Transformationsfunktion basierend auf Ausgaben anderer Transformationsfunktionen könnte entsprechend eine Aggregationsfunktion sein, die die Ausgaben mehrerer der zuvor beispielsweise beschriebenen Transformationsfunktion aggregiert, um einen Aspekt der funktionalen Korrektheit für den Akquiseprozesses eines Messwertes zu erheben (eine Zusammenfassung der Aspekte aller Komponenten, die in die Akquise des Messwertes involviert sind) [Bra+20].

Auf der obersten Ebene der Trust Assessment Pyramid befindet sich der multivariate Trust mit den sechs Trust-Facetten (siehe Abbildung 3.7), auf das im nächsten Teilabschnitt detaillierter eingegangen wird. Aus Gründen der Übersicht wurde auf gepunktete Verbindungen zwischen den Trust-Facetten und dem multivariaten Trust verzichtet. Dieses Informationsmodell mit seiner darunterliegenden Ontologie ermöglicht es, die eingangs erwähnten vier Ziele für die Trust-Erhebung in CPESs zu erreichen.

3.2.3 Einfache Trust-Werte, Trust-Facetten und multivariate Trust-Werte

Die Ausgabe eines Trust-Schätzers ist ein sogenannter einfacher Trust-Wert (siehe Abbildung 3.6), der wie folgt definiert ist [BBL21]:

Definition 11 (Einfacher Trust-Wert)**.** *Ein einfacher Trust-Wert $t_{e,\gamma}$ einer Entität e, der von einem Schätzer γ erhoben wird, ist ein Tupel der Form $t_{e,\gamma} = (\gamma, p)$, wobei $\gamma \in \mathbf{\Gamma}$ ein Trust-Schätzer aus der Menge $\mathbf{\Gamma}$ aller Trust-Schätzer und $p \in [0,1]$ die Wahrscheinlichkeit ist, dass die Entität e vertrauenswürdig ist.*

Einfache Trust-Werte sind immer mindestens einer der Trust-Facetten zugeordnet (siehe Abbildung 3.6), welche wie folgt definiert sind [BBL21]:

Definition 12 (Trust-Facette)**.** *Eine Trust-Facette* $\boldsymbol{T}_{e,f}$ *mit* $f \in \{fc, saf, sec, r, c, u\}$ *einer Entität* e *ist eine Menge von einfachen Trust-Werten für diese Entität und die entsprechende Trust-Facette:*

$$\boldsymbol{T}_{e,f} = \{t_{e,\gamma} \mid t_{e,\gamma} \xrightarrow{\gamma} f\} = \{(\gamma, p_{e,\gamma}) \mid (\gamma, p_{e,\gamma}) \xrightarrow{\gamma} f\}$$

mit $t_{e,\gamma} \xrightarrow{\gamma} f := \gamma$ *ordnet* $t_{e,\gamma}$ f *zu.*

fc, saf, sec, r, c, u stehen dabei für funktionale Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit. Eine Trust-Facette ist demnach eine Menge von einfachen Trust-Werten oder aber eine leere Menge, wenn es keine einfachen Trust-Werte für diese Trust-Facette gibt. Für PSNA-Trust sind dabei alle sechs Trust-Facetten relevant und deren Relevanz wird im Folgenden kurz erläutert [Bra+19b]:

Funktionale Korrektheit Die Frage, ob sich eine Entität im Feld an ihre funktionalen Spezifikationen hält, ist für PSNA-Trust von hoher Relevanz. Ein Messgerät, das z.B. nicht so genau misst, wie es aufgrund seiner Spezifikationen sollte, hat keine hohe funktionale Korrektheit (und Messwerte von diesem Messgerät werden im Idealfall als Bad Data identifiziert; vgl. Bad Data Detection in Unterabschnitt 2.2.3). Dies ist insbesondere dann relevant, wenn die betriebliche Fehlermarge des aktuellen Zustands eine hohe Genauigkeit erfordert.

Betriebssicherheit Wenn eine Entität im Feld in einen Zustand geraten kann, der Schaden anrichtet, könnte dies zu einem Ausfall führen. Zum Beispiel könnte eine Übertragungsleitung überlastet werden oder ein Server, der eine kritische Funktion ausführt, ausfallen. Daher ist die Betriebssicherheit auch für PSNA von hoher Relevanz. Sie kann a priori durch Sicherheitsstandards und -methoden (z. B. Modellprüfung) oder während der Laufzeit bewertet werden, z.B. durch die Distanz von Prozessvariablen von ihren Grenzwerten oder, bei IKT-Geräten, mit IT-Monitoring-Tools.

Informationssicherheit Alle Prozessvariablen im Rahmen von PSNA sind zu sichern. Mit dem Wissen über die Daten könnte ein Unbefugter, d.h. ein Angreifer, das System kennenlernen und weitere schädliche Angriffe planen. Zusätzlich sind FDIAs ein Beispiel dafür, was Informationsveränderungen (mangelnde Integrität) bewirken können. Sie können zur Vortäuschung eines falschen Systemzustands und dementsprechend zu schädlichen Steuerungsaktionen führen. Die Sicherheit kann a priori

mit einem Informationssicherheitsmanagementsystem oder zur Laufzeit mit einem IDS bewertet werden.

Zuverlässigkeit Die Frage, ob eine Entität im Feld für einen bestimmten Zeitraum verfügbar bleibt, ist auch für PSNA von Bedeutung. Wenn die Entität in dem Zeitraum nicht verfügbar ist, darf sie für PSNA nicht berücksichtigt werden und ihre Funktionalität muss von anderen Entitäten übernommen werden. Die Verfügbarkeit kann z.B. im Fall von IKT-Geräten mit IT-Monitoring-Tools bewertet werden.

Glaubwürdigkeit Im Kontext von PSNA kann man zwischen internen und externen Entitäten unterscheiden. Interne Entitäten stehen unter der Kontrolle des Netzbetreibers und sind über ein IKT-System verbunden, das ebenfalls unter der Kontrolle des Netzbetreibers steht. Externe Entitäten stehen unter der Kontrolle Dritter und können über das Internet mit dem Netzbetreiber verbunden sein (z.B. ein Windpark). Die Erwartung an die Genauigkeit sowie den guten Willen dieser externen Entitäten muss Teil der Glaubwürdigkeit sein, auch unter Berücksichtigung früherer Erfahrungen mit (z.B. Daten von) einer Entität. Glaubwürdigkeit ist zudem von besonderer Bedeutung bei der Betrachtung dezentraler Energiesysteme und Märkte. Zusammenfassend lässt sich sagen, dass Glaubwürdigkeit wichtig ist und mit Hilfe von Kontextwissen über Entitäten bewertet werden kann.

Bedienbarkeit Menschliche Bediener spielen eine wichtige Rolle im Energiebereich, sowohl in SCADA-Leitwarten als auch bei der dezentralen Entscheidungsfindung. Daher ist die Bedienbarkeit auch für PSNA von Bedeutung, wo immer die Daten von einem Menschen interpretiert werden und es zu Stresssituationen oder Informationsüberflutung kommen kann.

Eine Menge von Trust-Facetten, genauer ein 6-Tupel, ist ein multivariater Trust-Wert, der oben in Abbildung 3.6 zu finden und in PSNA-Trust wie folgt definiert ist [BBL21]:

Definition 13 (Multivariater Trust-Wert)**.** *Ein multivariater Trust-Wert $\boldsymbol{T}_e$ einer Entität e ist ein Tupel bestehend aus sechs Trust-Facetten:*

$$\boldsymbol{T}_e = (\boldsymbol{T}_{e,fc}, \boldsymbol{T}_{e,saf}, \boldsymbol{T}_{e,sec}, \boldsymbol{T}_{e,r}, \boldsymbol{T}_{e,c}, \boldsymbol{T}_{e,u}).$$

In anderen Worten ist ein multivariater Trust-Wert in PSNA-Trust ein Tupel aus sechs Mengen einfacher Trust-Werte, die wiederum Trust-Wahrscheinlichkeiten von unterschiedlichen Trust-Schätzern darstellen. Dabei wird es im Rahmen dieser Arbeit nicht dazu kommen, dass multivariate Trust-Werte für alle sechs Trust-Facetten erhoben werden, da nicht für alle Trust-Facetten auch Trust-Informationen erhoben werden können bzw. die Erhebung nicht für alle Trust-Facetten bei Messwerten sinnvoll ist. Zum Beispiel spielt die Bedienbarkeit der Messwerte keine Rolle, da i.d.R. die Datenakquise ein automatisierter Prozess ohne Interaktion mit Menschen ist. Ein Mensch kommt lediglich über SCADA-Systeme mit dem Ergebnis des Datenakquiseprozesses in Berührung. Die Bedienbarkeit von SCADA-Systemen ist aber nicht Teil dieser Arbeit. Mit dem Datenmodell können allerdings multivariate Trust-Werte mit allen sechs Trust-Facetten und beliebig vielen einfachen Trust-Werten modelliert werden. Es ist aber ausdrücklich erlaubt, dass eine oder mehrere Trust-Facetten leere Mengen sind.

3.3 Zusammenfassung

In diesem Kapitel wurde mit PSNA-Trust ein Trust-Modell für CPESs vorgestellt. In der Literatur, die in Abschnitt 3.1 vorgestellt wurde, wird der Trust-Begriff mit sehr unterschiedlicher Bedeutung belegt oder nicht definiert. Alle gefundenen Arbeiten haben allerdings gemein, dass sie Trust im Rahmen von Multiagentensystemen betrachten. Bei den verwandten Arbeiten, die auf OC-Trust, bereits vorgestellt in Abschnitt 2.3, basieren, handelt es sich um Trust-Modelle, die in der Regel Erfahrungen mit Agenten in den Mittelpunkt stellen. Es bedarf daher einer Weiterentwicklung von OC-Trust, die es erlaubt, neben Erfahrungen mit Entitäten vor allem auch Echtzeitinformationen über die Entitäten und die technischen Systeme, die die Entitäten miteinander verbinden, einfließen zu lassen.

Das in Abschnitt 3.2 vorgestellte Modell PSNA-Trust beinhaltet Untersuchungsgegenstände anstelle von Agenten, für die der Trust erhoben wird. Die Trust-Erhebung geschieht durch Trust-Schätzer mithilfe von Trust-Inputs, die von Trust-Quellen zur Verfügung gestellt werden, und Transformationsfunktionen. Transformationsfunktionen überführen Trust-Inputs, Ausgaben anderer Transformationsfunktionen oder beides in eine Trust-Wahrscheinlichkeit. Die Ausgabe von Trust-Schätzern sind einfache Trust-Werte, die sich aus dem Schätzer und der Trust-Wahrscheinlichkeit zusammensetzen. Ein einfacher Trust-Wert wird von dem Trust-Schätzer mindestens einer Trust-Facette zugeordnet. Die sechs Trust-Facetten aus Abschnitt 2.3 bilden zusammen einen multivariaten Trust-Wert.

4 Integrationsplattform für Trust-Schätzer

> *Wir leben in einer Welt, die in Daten ertrinkt. Wir haben die Wahl, ob wir sie weiterhin ignorieren und als großes Rauschen abtun oder ob wir sie nutzen wollen.*
>
> **— Jorn Lyseggen**

Dieses Kapitel widmet sich dem zweiten Forschungsziel, d.h. einer Integrationsplattform für Trust-Schätzer auf Basis von PSNA-Trust, das in Abschnitt 3.2 als multivariates und kontextsensitives Trust-Modell vorgestellt wurde. Dabei werden zunächst in Abschnitt 4.1 verwandte Arbeiten zu dem Thema vorgestellt, bevor in Abschnitt 4.2 die Integrationsplattform mit ihren Schnittstellen beschrieben wird. Das Kapitel schließt mit einer Zusammenfassung in Abschnitt 4.3.

4.1 Verwandte Arbeiten

In der Literatur lassen sich viele Maßnahmen gegen die Bedrohungen (siehe Abbildung 1.1 in Abschnitt 1.1), insbesondere durch FDIAs, finden, die sich im Sinne der vorliegenden Arbeit grob in die folgenden Kategorien einteilen lassen:

- Maßnahmen im Feld,
- verbesserte Bad Data Detection, oder
- Trust-basiert.

Die meisten davon lassen sich nicht direkt mit dem Ziel einer Integrationsplattform vergleichen. Gegenüber Maßnahmen im Feld ist die vorliegende Arbeit als ein Zusatz zu betrachten, da eine hundertprozentige Absicherung vom Sensor bis zum Kontrollzentrum hinsichtlich aller denkbaren Bedrohungen wie Angriffe und Fehlfunktionen nicht gewährleistet werden kann. Arbeiten in der Kategorie der verbesserten Bad Data Detection haben den Nachteil, dass sie i.d.R. Lösungen oder

Erkennungsmöglichkeiten für nur eine Art von Bedrohungen darstellen, zum Beispiel Angriffe oder noch konkreter FDIAs. Die Arbeiten aus dieser Kategorie lassen sich daher eher als einzelne Trust-Schätzer in ein Framework, wie es die vorliegende Arbeit anbieten soll, einbetten. Die vorliegende Arbeit liefert hier also ein Rahmenwerk zur Integration von Verfahren zur Bad Data Detection oder Verfahren auf Basis von Monitoringsystemen. Gleiches gilt für einige Arbeiten, die auf Trust basieren und in Abschnitt 3.1 betrachtet wurden.

Arbeiten, die der vorliegenden Arbeit am nächsten bzgl. der Integration von Trust-Schätzern kommen, sind die folgenden. Constante et al. [CMW19] (vergleichbar auch [Dav+15; MHW18; Wan+20]) schlagen ein semantisches Analyseframework vor, mit dem IDSs erweitert werden können. Durch das Anreichern der Logik eines IDS mit Kontextinformationen aus dem Stromsystem wird genau der zu dieser Arbeit entgegengesetzte Integrationsansatz verfolgt.

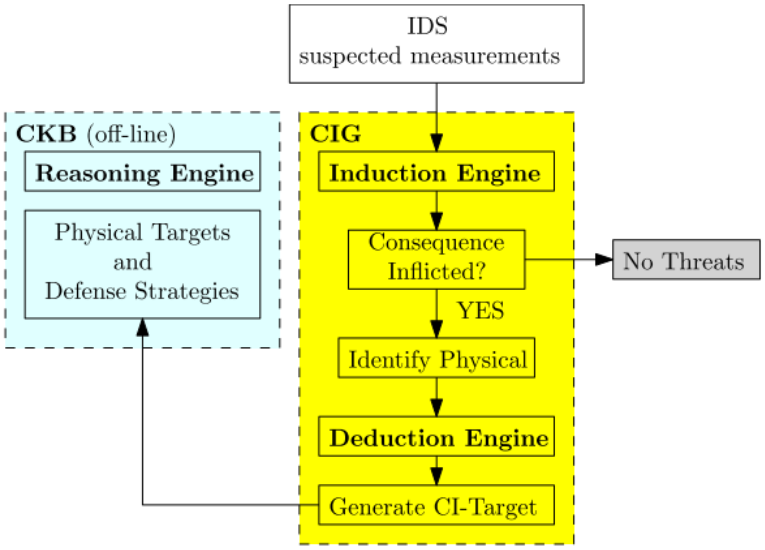


Abb. 4.1.: Das semantische Framework aus [CMW19].

Abbildung 4.1 zeigt das von Constante et al. [CMW19] vorgeschlagene semantische Framework bestehend aus einem Korrelationsindexgenerator (engl. correlation index generator (CIG)) und einer Korrelationswissensbasis (engl. correlation knowledge base (CKB)). Die Induktionsengine im Korrelationsindexgenerator schätzt die Intention des Angreifers durch die Verwendung von Kontextinformationen ab. Hier kann der betrachteten Arbeit ein Frameworkgedanke unterstellt werden, wenn beliebige Kontextinformationen hinzugefügt und verarbeitet werden können. Im Anschluss werden in der Deduktionsengine offline die durch den Angriff bedrohten Umspannwerke identifiziert. Die Korrelationswissensbasis wird immer dann aktualisiert, wenn sich Eigenschaften des Stromsystems, wie z.B. die Topologie, ändern [CMW19]. Im

Gegensatz zur vorliegenden Arbeit konzentrieren sich Constante et al. auf Angriffe, auf die durch ein IDS aufmerksam gemacht wird. Etwaig eingebundene Zusatzinformationen werden lediglich dazu verwendet, das Ziel des Angriffs zu identifizieren. Eine Anwendung auf andere Bedrohungen als Cyberangriffe erfolgt nicht.

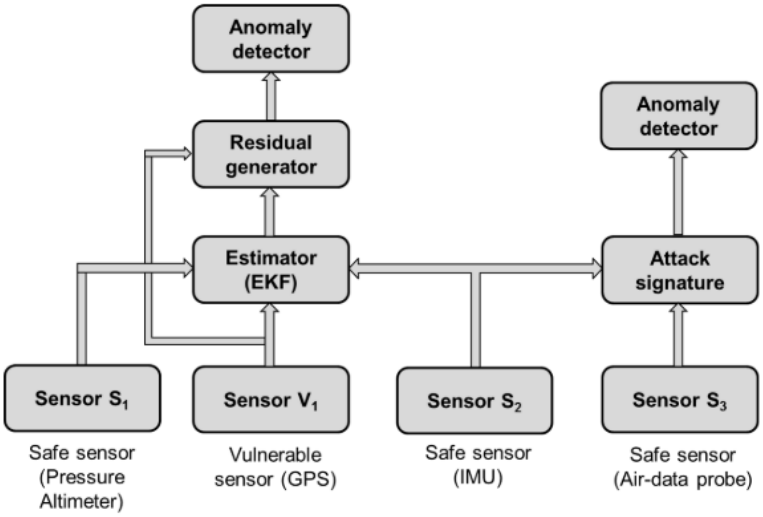


Abb. 4.2.: Das Framework aus [MF17] um Angriffe zu erkennen.

Ein weiteres Framework um Angriffe zu erkennen, wird von Muniraj und Farhood [MF17] vorgestellt und ist in Abbildung 4.2 dargestellt. Anders als in den anderen hier vorgestellten Arbeiten sind unbemannte Flugzeuge die Anwendungsdomäne von Muniraj und Farhood. Das vorgeschlagene Framework besteht aus mehreren Anomaliedetektoren, die entweder signaturbasiert sind und auf Basis von als sicher eingestuften Sensoren arbeiten oder sie basieren auf Residuen für die Ausgaben eines State Estimators. Als Ensemble für die verschiedenen Anomaliedeketoren wird ein Bayesisches Netzwerk eingesetzt [MF17]. „Ein Bayesisches Netzwerk für eine Menge von Variablen Z besteht aus einem gerichteten, azyklischen Graphen, der eine Menge bedingt unabhängiger Behauptungen über die Variablen in Z kodiert, und einer Menge P lokaler Wahrscheinlichkeitsverteilungen, die mit jeder Variable assoziiert ist“ [MF17] (aus dem Englischen übersetzt). Das Bayesische Netzwerk erhärtet den Verdacht auf einen Angriff indem die Ausgaben aller Anomaliedetektoren verwendet werden [MF17]. Der Ansatz ist sehr interessant, da beliebige Anomaliedetektoren eingebunden werden können. Der Ansatz geht allerdings davon aus, dass die Anomalien alle in einer Trust-Facette (Muniraj und Farhood verwenden den Begriff nicht) auftreten. Ansonsten würden Anomalien bzgl. der Informationssicherheit mit Anomalien bzgl. der Bedienbarkeit, als ein Beispiel, verwoben werden. Demnach ist es nicht möglich, anhand der Ausgabe des

Bayesischen Netzwerkes darauf rückzuschließen, was für eine Bedrohung vorliegt (welche Trust-Facetten betroffen sind).

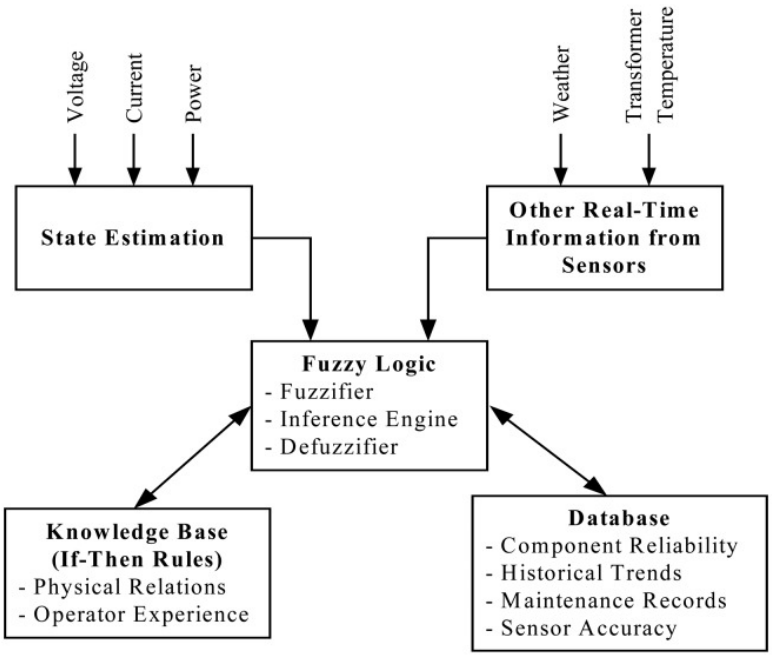


Abb. 4.3.: Das hybride Fuzzy-Logik-Klassifizierungssystem aus [HL04].

Holbert und Lin [HL04] schlagen eine Fuzzy-Logik-basierte State Estimation vor, die u.a. historische und statische Daten, wie z.B. Zuverlässigkeitsinformationen, mit berücksichtigt. Das System „zielt darauf ab, Informationen aus mehreren Domänen zu kombinieren um Bedrohungen für Energienetze zu detektieren, isolieren, identifizieren und abzumildern“ [HL04] (aus dem Englischen übersetzt). Damit verfolgen die Autoren ähnliche Ziele wie in der vorliegenden Arbeit. Das System besteht aus einem Fuzzifier, einer Wissensbasis, einer Inferenzengine und einem Defuzzifier, wie in Abbildung 4.3 dargestellt. Im Fuzzifier werden die Sensordaten in Fuzzyvariablen umgewandelt. Die Wissensbasis besteht typischerweise aus einer Menge von Wenn-Dann-Regeln, die neben gelernten, normalen operativen Zustände des CPES auch Informationen wie z.B. Komponentenzuverlässigkeit, Sensorgenauigkeit und Wartungsberichte berücksichtigen. In der Inferenzengine werden die Wenn-Dann-Regeln auf die Fuzzyvariablen angewandt. Der Defuzzifier konvertiert die Ausgaben der Wenn-Dann-Regeln abschließend wieder in entsprechende Formate [HL04]. Die angewandten Wenn-Dann-Regeln in [HL04] kombinieren die heterogenen Informationen und liefern eindeutige Aussagen bzgl. der Prozessvariablen, wie z.B. valide, fehlerhaft oder verdächtig [HL04]. Eine solche Klassifizierung ist allerdings insofern problematisch, als dass die vorliegenden heterogenen Information und deren Kombinationen nur selten eindeutig interpretiert werden können. Auch geht durch die

Anwendung von Wenn-Dann-Regeln die Multivarietät verloren, die notwendig ist um ursachenspezifisch handeln zu können. So erfordert ein Informationssicherheitsvorfall andere Handlungen als ein Betriebssicherheitsvorfall.

Auf Basis von OC-Trust (vgl. Abschnitte 2.3 und 3.1) schlagen Anders et al. [And+13] ein Trust-ermöglichendes Multiagentensystem (engl. trust-enabling multi-agent system (TEMAS)) für offene Umgebungen vor. TEMAS basiert dabei auf der sogenannten Trust-ermöglichenden Middleware (engl. trust-enabling middleware (TEM)), die u.a. die Aufgabe des Trust-Managements übernimmt. Dabei erlaubt die Integration anwendungsspezifischer Metriken sowohl TEMAS als auch den beteiligten Agenten, Trust-Werte für verschiedene Trust-Facetten auf Basis von Erfahrungen zu bestimmen [And+13].

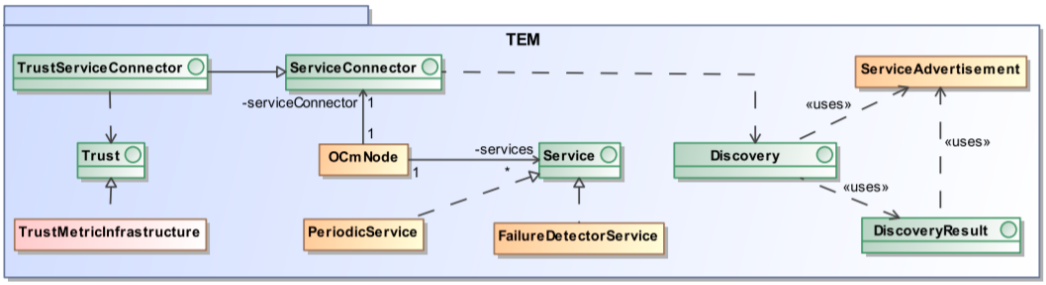


Abb. 4.4.: Die Konzepte der TEM aus [And+13].

Abbildung 4.4 [And+13] zeigt die relevanten Konzepte der TEM. Wie eine typische Middleware sind Services, Knoten auf denen die Services laufen sowie Prozesse zum Auffinden neuer Knoten und Services relevante Konzepte der TEM. Interessant für die vorliegende Arbeit ist die Integration von Trust in die TEM. Die TEM verfügt über eine Schnittstelle für Trust-Metriken, über die sowohl Metriken für direkten Trust als auch Reputationsmetriken eingebunden werden können. Die TEM stellt die Erfahrungen aus vorigen Interaktionen von Services mit Interaktionspartnern (z.B. Services oder Agenten) bereit und je nach gewünschter Trust-Facette und gewünschten Metriken kann die TEM Trust-Werte berechnen [And+13].

Um dies zu ermöglichen sammeln Services Rohdaten über ihre Interaktionen, die zur Berechnung der Trust-Werte herangezogen werden. Dazu werden die Rohdaten derart transformiert, dass nur die relevanten Daten für die Trust-Bestimmung verwendet werden. Dies können zum Beispiel nur die aktuellsten Rohdaten sein. Ein Interpret interpretiert im Anschluss die transformierten Daten und berechnet Trust-Daten, bei denen es sich entweder um eine einfache Gleitkommazahl oder ein komplexeres Objekt, wie z.B. eine Zeitreihe, handelt [And+13]. Vergleicht man die Konzepte mit denen von PSNA-Trust aus Abschnitt 3.2, so ähneln sich die Konzepte der Transformation und Transformationsfunktion sowie die der Interpretation und

Trust-Schätzer. Aus diesem Grund hätte die Arbeit von Anders et al. ebenfalls als verwandte Arbeit im vorigen Kapitel zum Trust-Modell behandelt werden können. Der Kern der Trust-Wertbestimmung besteht allerdings auch beim TEM aus den Entwurfsmustern von Anders et al. [And+11], die in Abschnitt 3.1 behandelt wurden. Insgesamt stellt das TEM eine gute Integrationsplattform für Trust-Schätzer dar. Der einzige Nachteil im Bezug auf die Ziele der vorliegenden Arbeit besteht darin, dass die Metriken, die eingebunden werden können, nicht auf Liveinformationen von anderen Services, wie Monitoringsystemen zurückgreifen können, sondern lediglich auf die Erfahrungen, und dass die Interpreter nach der Schnittstellenbeschreibung aus [And+13] lediglich einfache Gleitkommazahlen zurückgeben.

4.2 Integrationsplattform

In diesem Abschnitt wird das Konzept zur Umsetzung einer Integrationsplattform für Trust-Schätzer für ASSESS vorgestellt. Die Integrationsplattform besteht dabei aus Fix- und Variationspunkten und hat als technologische Grundlage, wie ASSESS im Allgemeinen, ein DSMS um die Trust-Schätzung ereignisgetrieben durchführen zu können (Aktualität). Die Fixpunkte der Integrationsplattform sind Schnittstellen zu den einzelnen Trust-Schätzern. Diese werden in Unterabschnitt 4.2.1 vorgestellt. In Unterabschnitt 4.2.2 wird ein beispielhafter Trust-Schätzer als Variationspunkt beschrieben.

4.2.1 Schnittstellen

Ein wesentlicher Baustein für Flexibilität und Skalierbarkeit der Integrationsplattform stellen wohldefinierte Schnittstellen für die Trust-Schätzer dar. Diese Schnittstellen verbinden innerhalb eines DSMS verschiedene kontinuierliche (Teil-) Anfragen (vgl. Unterabschnitt 2.4.1) miteinander. Demzufolge ist das Schema der Datenstromelemente an diesen Schnittstellen von zentraler Bedeutung um Trust-Schätzer einzufügen oder auszutauschen. Relevante Schemata sind dabei, erstens, die von Messwerten als wichtigste Eingangsgröße für die Trust-Schätzer, zweitens, die von Messwerten angereichert mit multivariaten Trust-Werten als Ausgangsgröße der Trust-Schätzer und, drittens, die von Topologien. Die Topologien sind für die Integrationsplattform insofern relevant, als dass es sich bei Messwerten um abgeleitete Untersuchungsgegenstände handelt (vgl. Unterabschnitt 3.2.1). Für die Schätzung des Trusts in diese abgeleiteten Untersuchungsgegenstände ist die Beziehung derer

zu Untersuchungsgegenständen wie RTUs, Routern u.a. essentiell und kann aus den entsprechenden Topologien abgeleitet werden.

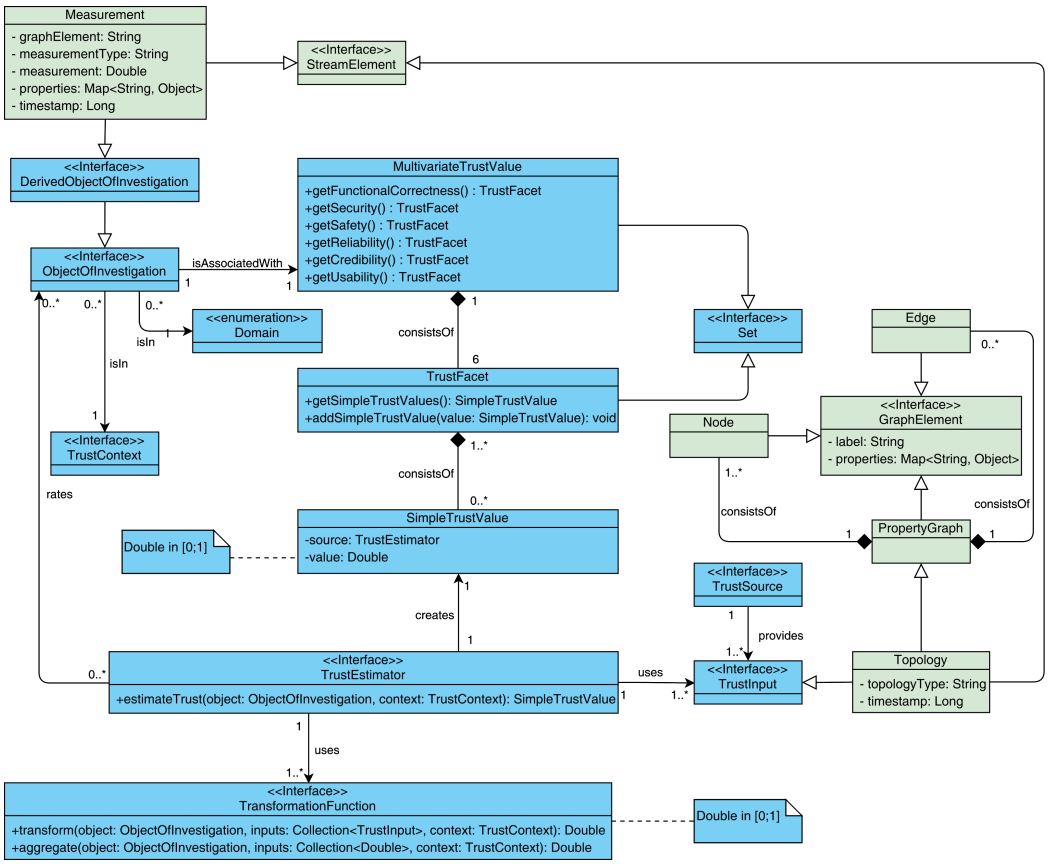


Abb. 4.5.: Das Modell für PSNA-Trust erweitert um Messwerte und Topologien als UML-Klassendiagramm. Bestandteile der Erweiterung sind grün hinterlegt (vgl. Abbildung 3.6).

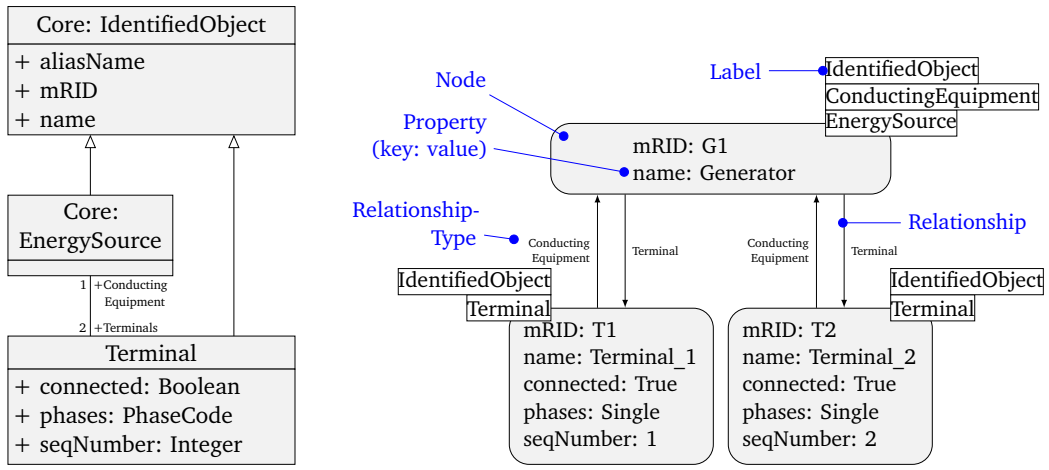
Abbildung 4.5 zeigt das aus Abschnitt 3.2 bekannte Entwurfsmuster für PSNA-Trust als UML-Klassendiagramm (vgl. Abbildung 3.6) erweitert um Messwerte als abgeleitete Untersuchungsgegenstände und gleichzeitig Datenstromelemente (oben links) sowie Topologien als Trust-Inputs und gleichzeitig Eigenschaftsgraphen (rechts). Auf die konkreten Datenmodelle wird im Folgenden eingegangen. Zunächst wird die Modellierung von Topologien als Eigenschaftsgraphen erläutert und begründet. Im Anschluss erfolgt die Erklärung für das Messwertedatenmodell. Abgeschlossen wird dieser Unterabschnitt mit dem Datenmodell, das Mess- und multivariate Trust-Werte verknüpft.

Datenmodell für Topologien (T-Modell)

Das Ziel ist es, ein einheitliches Datenmodell für Topologien für ASSESS zu erstellen, das es erlaubt beliebige Topologien (z.B. Strom und IKT) zu modellieren. Darüber hinaus muss es in einem DSMS anwendbar sein, d.h. beliebig komplexe Strukturen beeinträchtigen im Zweifel die ereignisgetriebene Verarbeitung im Hauptspeicher. Weitere Anforderungen an ein Datenmodell für Topologien sind die Folgenden. Erstens muss es Knoten und deren Verbindungen untereinander repräsentieren. In der Topologie eines Stromnetzes sind dies z.B. Sammelschienen, die über Leitungen miteinander verbunden sind. Analog sind Beispiele aus einer IKT-Netztopologie Router und Switches, die miteinander und mit RTUs und einer SCADA-Masterstation verbunden sind. Neben den Entitäten einer Topologie müssen zweitens auch deren Eigenschaften modelliert werden. Diese können je nach System (z.B. Strom oder IKT) stark variieren. Beispiele sind hier Leitungslängen, Impedanzen (Strom) und maximale Bandbreiten (IKT).

Die benannten Anforderungen legen eine Modellierung als Graphen nahe. Ein Graph ist dabei eine Struktur, die aus Knoten und Kanten besteht. Mathematisch ausgedrückt ist ein Graph G ein geordnetes Paar (V, E) mit einer Knotenmenge V und einer Kantenmenge E. E ist dabei bei ungerichteten Graphen, die sich bei ASSESS aufgrund bidirektionaler Strom- und Informationsflüsse anbieten, eine Teilmenge aller zweielementigen Teilmengen von V: $\forall e \in E : e = \{v_i, v_j\}; v_{i,j} \in V$. Graphen finden bereits bei der Speicherung stark vernetzter Informationen in sogenannten Graphdatenbanken Anwendung. Konkret wird in Graphdatenbanken i.d.R. das Modell des Eigenschaftsgraphen (engl. property graph) oder das Ressourcenbeschreibungsframework (engl. resource description framework) verwendet. In einem Eigenschaftsgraphen können Graphen, Knoten und Kanten mit Bezeichnern und einer beliebigen Anzahl von Eigenschaften versehen werden. Im Ressourcenbeschreibungsframework werden Graphen durch eine Menge von Tripeln modelliert. Ein Tripel hat dabei die Struktur Subjekt, Prädikat, Objekt. Der Vorteil des Ressourcenbeschreibungsframeworks liegt in der Möglichkeit automatisiert Schlussfolgerungen ableiten zu können. Dies ist zwar generell interessant, spielt allerdings im Rahmen dieser Arbeit keine Rolle, da die Topologien nicht die Untersuchungsgegenstände sind; es sind die Messwerte. Vielmehr ist es wichtig, alle möglichen Informationen über die verschiedenen Topologien und deren Komponenten modellieren zu können. Dafür scheint das Eigenschaftsgraphmodell besser geeignet, weshalb im Rahmen von ASSESS Topologien als Eigenschaftsgraphen modelliert werden.

Ravikumar und Khaparde stellen in ihrer Arbeit [RK17] bereits ein Common Information Model (CIM)-orientiertes Graphdatenbankenframework vor. Abbildung 4.6



(a) CIM-UML **(b)** Eigenschaftsgraph

Abb. 4.6.: Die Repräsentation eines Generators im CIM-UML (links) und als Eigenschaftsgraph (rechts) nach [RK17].

zeigt die Übersetzung von CIM-UML in einen Eigenschaftsgraphen anhand des Beispiels eines Generators. Ein Generator ist eine Energiequelle, und damit auch sowohl ein leitendes Gerät als auch ein identifiziertes Objekt, und verfügt über zwei Terminals. Die Hierarchie im CIM, in diesem Beispiel `IdentifiedObject` → `ConductingEquipment` → `EnergySource`, kann mittels Bezeichnern für das Graphelement dargestellt werden, so dass Informationen über die Klassenhierarchie aus dem CIM erhalten bleiben. Alle einfachen Attribute werden als Schlüssel-Wert-Paare dem Graphelement hinzugefügt, während alle Beziehungen aus dem CIM als Kanten modelliert werden. Im CIM haben Beziehungen nur selten Eigenschaften, aber Rollen und Kardinalitäten. Die Rollen werden dabei als Kantentypen übernommen. Kardinalitäten werden implizit übernommen, da in einem Eigenschaftsgraphen die einzelnen Objekte einer Topologie und nicht deren Klassen wie im CIM modelliert werden. Daher resultiert eine Kardinalität von n in n verschiedenen Knoten mit entsprechenden Kanten.

Mit dem in [RK17] vorgeschlagenen CIM-orientierten Datenmodell lassen sich alle Topologieinformationen in einem Eigenschaftsgraphen modellieren. Der Eigenschaftsgraph wird allerdings auch schnell sehr groß und komplex und somit ressourcenintensiv. Gerade wenn es nicht primär um die Verarbeitung der Topologie geht, sondern die Messwerte im Vordergrund stehen und die Topologie lediglich Kontextinformationen für die Verarbeitung liefert, wie es bei ASSESS der Fall ist, erscheint die in [RK17] vorgeschlagenen Übersetzung unzweckmäßig. Immerhin soll die Topologie im Sinne der ereignisgetriebenen Verarbeitung im DSMS im Hauptspeicher gehalten werden. Aus diesem Grund wird im Rahmen von ASSESS auf ein

so umfangreiches CIM-orientiertes Graphmodell verzichtet. Relevante Knotenpunkte aus den Topologien (z.B. Sammelschienen oder Router) werden im Rahmen von ASSESS als Knoten und deren Verbindungen als Kanten modelliert. Die Klasse bzw. Art eines Knotenpunktes oder einer Verbindung wird durch einen Bezeichner ausgedrückt (z.B. Sammelschiene oder Router). Das Hinzufügen weiterer Bezeichner um Vererbungshierarchien auszudrücken ist ausdrücklich möglich und zugelassen. Die CIM-Attribute werden in Form von Knoten- und Kanteneigenschaften modelliert. Dabei sind die Attribute so aufgebaut, dass es pro CIM-Klasse, in der das Attribut definiert ist, einen Schlüssel gibt und der Wert zu diesem Schlüssel wiederum eine Menge von Schlüssel-Wert-Paaren ist (CIM-Attributname und Wert).

```
{
  "IdentifiedObject": {
    "mRID": "G1",
    "name": "Generator"
  },
  "Terminals": [{
    "IdentifiedObject": {
      "mRID": "T1",
      "name": "Terminal_1"
    },
    "connected": true,
    "phases": "SinglePhase",
    "sequenceNumber": 1
  },
  {
    "IdentifiedObject": {
      "mRID": "T2",
      "name": "Terminal_2"
    },
    "connected": true,
    "phases": "SinglePhase",
    "sequenceNumber": 2
  }]
}
```

Skript 4.1: Modellierung von CIM-Attributen als Graphelementeigenschaften am Beispiel eines Generators aus [RK17] (siehe auch Abbildung 4.6).

Skript 4.1 demonstriert dies am bereits eingeführten Beispiel des Generators (Abbildung 4.6). Die Attribute `mrid` und `name` zu der Klasse `IdentifiedObject`, während `connected`, `phases` und `sequenceNumber` zu der Klasse `Terminal` gehören. Der Generator verfügt über zwei Terminals.

Tab. 4.1.: Das Datenmodell für Topologien als Datenstromtupel mit den Attributen und ihren Datentypen. Identifizierende Attribute sind kursiv dargestellt.

Attribut	**Datentyp**
Topology	Graph
Timestamp	Long

Zusammenfassend ist das Datenmodell für Topologien (T-Modell) ein Eigenschaftsgraph mit Knoten (z.B. Sammelschienen oder Router) und Kanten (z.B. Stromleitungen oder IKT-Verbindungen). Die Art eines Knotens oder einer Kante, also die CIM-Klasse, wird durch einen Bezeichner repräsentiert, während die Eigenschaften von Graphobjekten die Attribute eines CIM-Objekts hierarchisch abbilden. Der Aufbau ist auch Tabelle 4.1 zu entnehmen. Ein Topologietupel besteht dabei aus lediglich einem Attribut: `Topology` (Graph). Hinzu kommt als Kontextinformation der Zeitstempel der Topologie. Das T-Modell ist ebenfalls rechts in Abbildung 4.5 zu sehen.

Datenmodell für Messwerte (M-Modell)

Das Ziel ist es, ein einheitliches Datenmodell für Messwerte in ASSESS zu erstellen, um die Verarbeitung in ASSESS unabhängig von verwendeten Protokollen und Formaten zu gestalten. Relevant für eine Verarbeitung in ASSESS mit Hinblick auf höhere Anwendungen und insbesondere die State Estimation sind vor allem das Topologieelement, zu dem der Messwert gehört, der Typ des Messwertes (z.B. Wirkleistung) und der eigentliche Messwert.

Das zugehörige Topologieelement ist für alle folgenden Bereiche von ASSESS relevant. Für die Trust-Schätzung gilt, dass viele Informationen für Topologieelemente und nicht für Messwerte erhoben werden (vgl. Abschnitt 3.2). Bei der Trust-sensitiven State Estimation, wie bei einer herkömmlichen State Estimation auch, ist die Zuordnung von Messwerten zu Topologieelementen für die Bildung des Systemmodells vonnöten (vgl. Abschnitt 2.2). Letzteres gilt auch für den Messwerttyp.

Darüber hinaus können noch weitere Attribute, wie zum Beispiel die Phase, von Interesse sein. Der Unterschied zu den vorigen Attributen besteht darin, dass nicht

alle Messwerte einen Phasenbezug haben. Aus diesem Grund werden diese spezielleren Attribute in einem Eigenschaftenattribut zusammengefasst. Dabei wird dieses Attribut der Konsistenz halber CIM-orientiert aufgebaut (vgl. T-Modell oben im selben Abschnitt und Auflistung 4.1).

Tab. 4.2.: Das Datenmodell für Messwerte als Datenstromtupel mit den Attributen und ihren Datentypen. Identifizierende Attribute sind kursiv dargestellt.

Attribut	**Datentyp**
GraphElement	String
MeasurementType	String
Measurement	Double
Properties	KeyValue
Timestamp	Long

Zusammenfassend ist das Datenmodell für Messwerte (M-Modell) ein relationales Tupel um die Verarbeitung in einem DSMS zu erleichtern. Der Aufbau ist auch Tabelle 4.2 zu entnehmen. Ein Messwerttupel besteht dabei aus vier Attributen: `GraphElement` (Text), `MeasurementType` (Text), `Measurement` (Gleitkommazahl) und `Properties` (Schlüssel-Wert-Paare). Identifiziert wird ein Messwert über die Kombination aus `GraphElement`, also einem Knoten oder einer Kante aus der Topologie des Stromnetzes, und dem `MeasurementType`, z.B. einem einphasigen Leistungswert. Hinzu kommt als Kontextinformation der Zeitstempel des Messwertes. Das M-Modell ist ebenfalls oben links in Abbildung 4.5 zu sehen.

Datenmodell für Messwerte angereichert mit Trust-Werten (M-T-Modell)

Die Schnittstelle für die Ausgaben von Trust-Schätzern wird durch das Datenmodell für Messwerte mit Trust-Wert (M-T-Modell) definiert, das das M-Modell um multivariate Trust-Werte erweitert.

Der Aufbau ist auch Tabelle 4.3 zu entnehmen. Ein Tupel im M-T-Modell besteht neben den Attributen aus dem M-Modell aus je einem Schlüssel-Wert-Paar pro Trust-Facette als Kontextinformation. Genauer ist jede Trust-Facette nach der Definition von PSNA-Trust (vgl. Abschnitt 3.2) eine leere Menge oder eine Menge von einfachen Trust-Werten, die wiederum Schlüssel-Wert-Paare sind. Die Schlüssel sind dabei Identifizierer für die Trust-Schätzer und die Werte die geschätzte Trust-Wahrscheinlichkeit.

Tab. 4.3.: Das Datenmodell für Messwerte mit Trust-Wert als Datenstromtupel mit den Attributen und ihren Datentypen. Identifizierende Attribute sind kursiv dargestellt.

Attribut	**Datentyp**
GraphElement	String
MeasurementType	String
Measurement	Double
Properties	KeyValue
FunctionalCorrectness	KeyValue
Safety	KeyValue
Security	KeyValue
Reliability	KeyValue
Credibility	KeyValue
Usability	KeyValue
Timestamp	Long

4.2.2 Beispielhafter Trust-Schätzer

Abbildung 4.7 illustriert einen Trust-Schätzer als UML-Aktivitätsdiagramm, der die bereits in Abbildung 3.8 in Unterabschnitt 3.2.2 illustrierte Transformationsfunktion verwendet. Die Transformationsfunktion schätzt auf Basis einer aktuellen Ressourcenauslastung von Komponenten, wie z.B. CPU-Auslastung von RTUs, Router, etc., und definierten Schwellwerten die funktionale Korrektheit der entsprechenden Komponente. Dies ist der oberste Aktivitätszweig in Abbildung 4.7. Eine genauere Beschreibung der Transformationsfunktion ist Unterabschnitt 3.2.2 zu entnehmen. Auf Basis der IKT-Topologie im T-Modell werden für alle RTUs diejenigen Komponenten identifiziert, die höchstwahrscheinlich an der Datenakquise der Messwerte der entsprechenden RTU beteiligt sind. Für einen eingehenden Messwert im M-Modell werden dann die einfachen Trust-Werte für alle bei der Datenakquise dieses Messwertes beteiligten Komponenten aggregiert. Die Aggregation erfolgt durch das Bilden des Minimums. Weitere im Rahmen dieser Arbeit konzipierte Trust-Schätzer sind Kapitel A des Anhangs zu entnehmen.

4.3 Zusammenfassung

In diesem Kapitel wurde, hauptsächlich durch die Definition von Schnittstellen, eine Integrationsplattform für Trust-Schätzer vorgestellt. In der Literatur, die in Abschnitt 4.1 vorgestellt wurde, gibt es kaum Arbeiten, die sich mit einer Integrationsplattform für Trust-Schätzer auf Basis eines multivariaten Trust-Modells

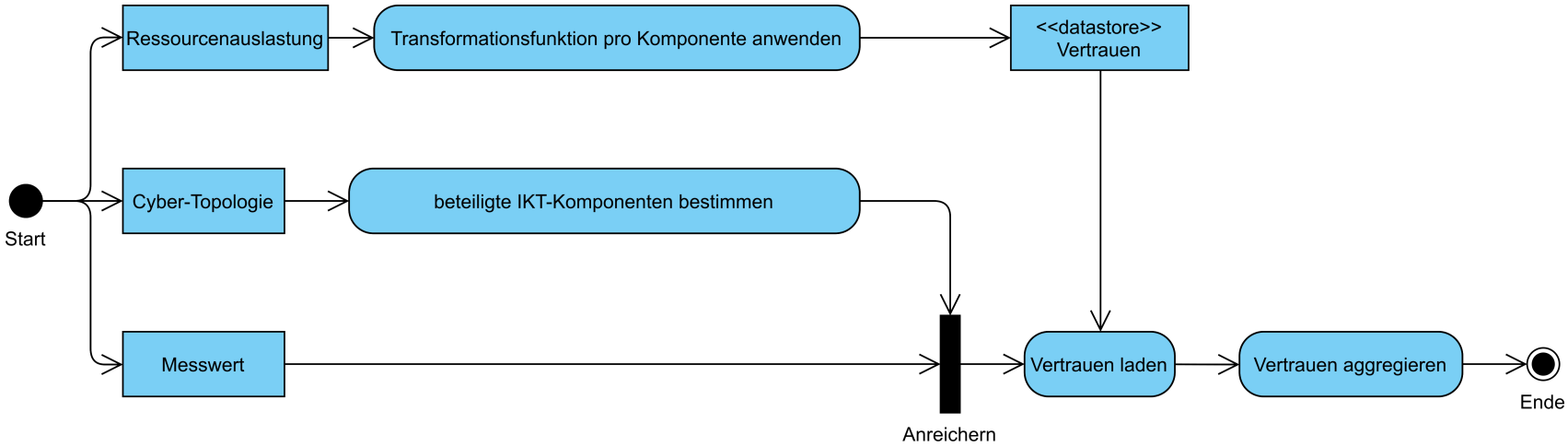


Abb. 4.7.: Das UML-Aktivitätsdiagramm für einen Trust-Schätzer auf Basis von Ressourcenauslastungsdaten. Es setzt dabei für jede Komponente, z.B. eine RTU, die Transformationsfunktion aus Abbildung 3.8 um. Für einen Messwert werden alle Komponenten bestimmt, die an der Datenakquise aller Wahrscheinlichkeit nach beteiligt waren. Die Ausgaben (Trust-Wahrscheinlichkeiten) der Transformationsfunktionen für diese Geräte werden entsprechend aggregiert.

vergleichen lassen. Entweder fokussieren sich die meisten Arbeiten auf lediglich eine Trust-Facette, die oft die Informationssicherheit ist, oder die Trust-Schätzung basiert, ganz im Sinne von OC-Trust, lediglich auf Erfahrungswerten und nicht auf Liveinformationen von z.B. Monitoringsystemen. Die in dieser Arbeit konzipierte Integrationsplattform wurde in Abschnitt 4.2 vorgestellt. Die Schnittstelle für Eingabegrößen besteht aus dem T-Modell für Topologien, das eine Topologie als einen Eigenschaftsgraphen darstellt, und dem M-Modell, das einen Messwert als ein relationales Tupel mit dem Graphelement (Text), dem Messwerttyp (Text), dem Messwert (Gleitkommazahl) und zusätzlichen Eigenschaften (Schlüssel-Wert-Paare) darstellt. Die Topologien gliedern sich dabei als Trust-Inputs und die Messwerte als abgeleitete Untersuchungsgegenstände in PSNA-Trust ein (vgl. Kapitel 3). Die Schnittstelle für Ausgangsgrößen besteht aus dem M-T-Modell, das das M-Modell um einen multivariaten Trust-Wert gemäß Kapitel 3 erweitert. Zudem wurde ein beispielhafter Trust-Schätzer beschrieben.

5 Trust-Sensitive Lagebilderkennung

> *Die Erweiterung des Vertrauens ist eine Artikulation unserer Einschätzung einer Situation, um ein erwartetes Ergebnis zu liefern.*
>
> — **David Amerland**

In diesem Kapitel werden für das dritte Forschungsziel einer Trust-sensitiven Lagebilderkennung, d.h. die multivariaten Trust-Werte der Messwerte in eine State Estimation einfließen zu lassen, zwei Alternativen diskutiert:

1. Über eine Unsicherheitsanalyse: Der multivariate Trust-Wert pro Messwert wird zu einer Trust-Wahrscheinlichkeit im Wertebereich $[0; 1]$ aggregiert und in eine Standardabweichung umgewandelt. Mit diesen Standardabweichungen für die Messwerte werden dann Unsicherheiten für die Zustandsvariablen berechnet [Bra+20]. Dieses Vorgehen wird in Abschnitt 5.2 beschrieben.
2. Über eine entkoppelte Schätzung einfacher Trust-Werte: Für jeden einfachen Trust-Wert der Messwerte wird auf Basis von Sensitivitäten der Zustandsvariablen bzgl. Messwertänderungen ein entsprechender einfacher Trust-Wert der Zustandsvariablen geschätzt [BBL21]. Dieses Vorgehen wird in Abschnitt 5.3 beschrieben.

Zuvor wird in Abschnitt 5.1 ein Blick auf verwandte Arbeiten zu dem Thema geworfen. Das Kapitel schließt mit einer Zusammenfassung in Abschnitt 5.4.

5.1 Verwandte Arbeiten

In der Literatur lassen sich zwar viele Maßnahmen gegen Bedrohungen, insbesondere durch FDIAs (siehe Abschnitt 1.1), finden, allerdings integrieren nur wenige Arbeiten zusätzliche Informationen direkt in eine State Estimation. In diesem Abschnitt

werden stellvertretend für die in der Literatur bekannten Verfahren zwei dieser Arbeiten beschrieben.

$$\Omega_i = \sqrt{1 + \sum_{k \in alert(device\_i)} m^{priority(k)}} \tag{5.1}$$

$$\boldsymbol{\Omega} = diag\{\Omega_1, \Omega_2, ...\Omega_n\} \tag{5.2}$$

Liu et al. [Liu+15] integrieren Informationen aus einem IDS in eine State Estimation. Zunächst werden alle Alarme den entsprechenden RTUs zugeordnet und unter Berücksichtigung der Alarmpriorität aufsummiert. Das Ergebnis ist ein Netzwerkauswirkungsfaktor (engl. network impact factor). Die genaue Berechnung ist in Gleichung 5.1 [Liu+15] gegeben. $alert(device\_i)$ ist dabei die Liste aller IDS-Alarme für RTU i, $priority(k)$ die dem Alarm vom IDS zugewiesene Priorität und $m > 1$ ein Gewichtungskoeffizient für die Priorität. Die Wurzel dient zur Glättung und das Addieren von eins sorgt für eine nichtlineare Steigung [Liu+15]. Die bei der Anwendung von Gleichung 5.1 für alle Messwerte entstehende Netwerkauswirkungsfaktormatrix ist in Gleichung 5.2 [Liu+15] dargestellt.

$$\text{minimiere } \boldsymbol{J}_{ATSE}(\boldsymbol{x}) = \sum_{j=1}^{m}\left(\frac{(z_j - h_j(\boldsymbol{x}))^2}{\Omega_{jj}, R_{jj}}\right) = [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})]^T \cdot (\boldsymbol{\Omega R})^{-1} \cdot [\boldsymbol{z} - \boldsymbol{h}(\boldsymbol{x})] \tag{5.3}$$

Die Netwerkauswirkungsfaktormatrix wird, wie in Gleichung 5.3 [Liu+15] gezeigt, in die Zielfunktion für die State Estimation als Anpassung der durch die Standardabweichungen $\boldsymbol{R}$ gegebenen Gewichte integriert (vgl. auch Gleichung 2.10 in Unterabschnitt 2.2.2). Mit diesem Vorgehen ist es Liu et al. zwar möglich, IDS-Alarme als Gewicht in eine State Estimation einzubringen, allerdings funktioniert dies auch nur für nicht zusammenhängende Alarme. Wie eigene Versuche [Bra+20] zeigen, beeinflussen von einander abhängige Gewichtungen das Verhalten des State Estimators bis hin zu Situationen, in denen er nicht mehr konvergiert. Des Weiteren müssten für eine Anwendung auf ein multivariates Trust-Modell der Trust zu einem einzelnen Wert aggregiert werden, wodurch die durch die Multivarietät gewonnenen Informationen wieder verloren gehen würden. In Conclusio funktioniert der Ansatz von Liu et al. leider nur sehr eingeschränkt und ist nicht für eine Trust-sensitive State Estimation geeignet.

Ein weiteres Vorgehen wird von Basciftci und Ozguner [BO12] vorgeschlagen. Anders als bei Liu et al. [Liu+15] und in der vorliegenden Arbeit wird kein State

Estimator mit kleinsten gewichteten Quadraten sondern ein Partikelfilter verwendet. Basciftci und Ozguner schlagen entsprechend einen neuen, Trust-sensitiven Partikelfilter vor, bei dem, basierend auf den Messwerten, ein Trust-Wert für jeden Sensor berechnet wird. Die Autoren definieren zwei Trust-Klassen für jeden Sensor, vertrauenswürdig ($\boldsymbol{J}_k^n = 1$) und nicht vertrauenswürdig ($\boldsymbol{J}_k^n = 0$), und arbeiten mit Wahrscheinlichkeiten für diese beiden Klassen [BO12]. $\boldsymbol{J}_k^n$ ist dabei eine Zufallsvariable, die den Zustand von Sensor k zum Zeitpunkt n repräsentiert. Die Wahrscheinlichkeiten für die Klassenzugehörigkeit sind entsprechend als $P[\boldsymbol{J}_k^n = 1]$ und $P[\boldsymbol{J}_k^n = 0]$ bezeichnet [BO12].

$$\boldsymbol{T}_{k,l}^n = \begin{bmatrix} P[\boldsymbol{J}_k^n = 0 | \boldsymbol{J}_l^n = 0] & P[\boldsymbol{J}_k^n = 0 | \boldsymbol{J}_l^n = 1] \\ P[\boldsymbol{J}_k^n = 1 | \boldsymbol{J}_l^n = 0] & P[\boldsymbol{J}_k^n = 1 | \boldsymbol{J}_l^n = 1] \end{bmatrix} \tag{5.4}$$

Auf dieser Grundlage wird eine Trust-Übergangsmatrix erstellt, die aus den Wahrscheinlichkeiten besteht, mit denen ein Sensor k vertrauenswürdig bzw. nicht vertrauenswürdig ist, unter der Bedingung, dass ein anderer Sensor l vertrauenswürdig bzw. nicht vertrauenswürdig ist. Die Trust-Übergangsmatrix ist in Gleichung 5.4 [BO12] dargestellt. Die Berechnung der bedingten Wahrscheinlichkeiten basiert dabei im Wesentlichen auf der Stichprobenvarianz der Sensorwerte der beiden Sensoren k und l [BO12]. Wie bei Liu et al. [Liu+15] werden die errechneten Trust-Wahrscheinlichkeiten als Gewichte in die Zielfunktion integriert, die allerdings beim Partikelfilter anders aussieht und an dieser Stelle nicht vorgestellt wird. Details sind [BO12] zu entnehmen. Die Eignung des Ansatzes von Basciftci und Ozguner [BO12] für multivariaten Trust ist ebenso wie bei Liu et al. nicht gegeben. So könnten zwar Trust-Übergangsmatrizen für alle Trust-Facetten aufgestellt werden, beim Einsatz in dem Partikelfilter fließen sie aber spätestens alle zu einer einzelnen Trust-Wahrscheinlichkeit zusammen.

Zusammenfassend bieten Liu et al. [Liu+15] und Basciftci und Ozguner [BO12] zwar interessante Ansätze für eine Trust-sensitive State Estimation, allerdings sind diese insofern nicht für ein multivariates Trust-Modell geeignet, als dass die Multivarietät in beiden Arbeiten durch die Integration in die State Estimation verloren geht.

5.2 Variante 1: Unsicherheitsanalyse

Anmerkung: Der Inhalt dieses Abschnitts wurde bereits in [Bra+20] veröffentlicht.

Bei der ersten Variante, die in diesem Kapitel für eine Trust-sensitive Lagebilderkennung diskutiert wird, wird der Einfachheit halber der multivariate Trust-Wert pro Messwert zu einer Trust-Wahrscheinlichkeit im Wertebereich $[0; 1]$ aggregiert und in eine Standardabweichung umgewandelt, wobei vereinfacht eine Gauß-Verteilung angenommen wird. Mit diesen Standardabweichungen für die Messwerte werden dann Unsicherheiten für die Zustandsvariablen berechnet [Bra+20]. Diese Alternative ist eine Lösung, die einfach umzusetzen ist, da es in der Literatur viele Verfahren für eine Unsicherheitsanalyse gibt. Auch ist eine solche bereits Bestandteil einiger State-Estimation-Programme. Eine erste Herausforderung ist dabei aber die Abbildung eines multivariaten Trust-Wertes auf eine Trust-Wahrscheinlichkeit. Zum einen erscheint es schwierig eine optimale Aggregationsfunktion zu finden, da diese von vielen Faktoren wie z.B. der Relevanz, die man bestimmten Trust-Facetten beimisst, abhängt. Zum anderen ist eine solche Aggregation keine bijektive Abbildung, so dass im Nachhinein nicht von der Trust-Wahrscheinlichkeit auf den multivariaten Trust-Wert geschlossen werden kann. Dies birgt Nachteile bei der Nachvollziehbarkeit des Verhaltens und Ergebnisses von ASSESS. Ein weiterer, gravierender Aspekt ist, dass ein State-Estimation-Programm die Standardabweichungen der Messwerte nicht nur für eine Unsicherheitsanalyse verwendet, sondern auch für die eigentliche State Estimation. Dabei werden, wie in Unterabschnitt 2.2.2 beschrieben, Messwertefehler als zufällig und ausdrücklich nicht zusammenhängend angenommen. Dies ist aber nicht mehr gegeben, wenn die multivariaten Trust-Werte in Standardabweichungen überführt werden, da sich Ereignisse, die mehrere Messwerte betreffen, wie z.B. koordinierte FDIAs (siehe Abschnitt 1.1), auf die Trust-Werte mehrerer Messwerte auswirken können. Im Folgenden wird zunächst die Methodik zur Unsicherheitsanalyse detailliert dargestellt. Im Anschluss werden Ergebnisse von Versuchen mit diesem Ansatz [Bra+20] präsentiert, die die Vermutung untermauern, dass sich dieser Ansatz nur bedingt eignet.

5.2.1 Methodik

Abbildung 5.1 zeigt das Vorgehen als UML-Aktivitätsdiagramm. Die Eingaben sind Messwerte im M-T-Modell (siehe Unterabschnitt 4.2.1). Für jede Trust-Facette des multivariaten Trust-Wertes eines jeden Messwertes werden zunächst die einfachen Trust-Werte zu einer einzelnen Trust-Wahrscheinlichkeit, einer Gleitkommazahl in $[0; 1]$, aggregiert. Als Aggregationsfunktionen sind hier beliebige denkbar, wie zum Beispiel das Bilden des Minimums, des Medians oder eines gewichteten arithmetischen Mittels. Außerdem können für unterschiedliche Trust-Facetten unterschiedliche Aggregationsfunktion angewandt werden.

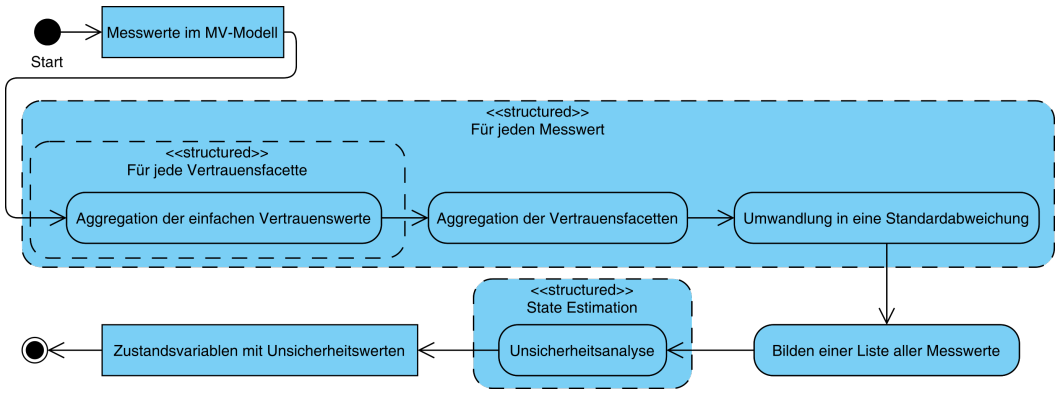


Abb. 5.1.: Die Unsicherheitsanalyse als Möglichkeit für eine Trust-sensitive Lagebilderkennung als UML-Aktivitätsdiagramm.

Im Anschluss werden die entstandenen Trust-Wahrscheinlichkeiten für jeden Messwert aggregiert und in eine Standardabweichung umgewandelt.

$$stdDev(z) = 1 - p_{agg}(z) \tag{5.5}$$

Gleichung 5.5 zeigt die Formel zur Berechnung der Standardabweichung $stdDev(z)$ für einen Messwert z. Die aggregierte Trust-Wahrscheinlichkeit $p_{agg}(z)$ für z wird dabei von 1 subtrahiert. Im Anschluss werden alle Messwerte, die für die State Estimation verwendet werden, in einer Liste zusammengefasst und die State Estimation wird durchgeführt. Der verwendete ASE [KL12; KML15] verfügt dabei über einen Algorithmus zur Berechnung der Unsicherheit jeder Zustandsvariablen auf Basis der Standardabweichungen der Messwerte (vgl. Unterabschnitt 2.2.4). Das Ergebnis der Unsicherheitsanalyse besteht entsprechend aus den geschätzten Zustandsvariablen und ihren Unsicherheiten.

5.2.2 Demonstration

Der Zweck der durchgeführten Demonstration ist die Untersuchung der Auswirkungen eines reduzierten Trusts in Messwerte auf die geschätzten Zustandsvariablen und deren Unsicherheiten.

Typischerweise ist die Unsicherheit von Spannungsmagnituden $u(V_i)$ vernachlässigbar, d.h. $u(V_i) \leq 0,001\ p.u.$ Die Unsicherheit von Phasenwinkeln $u(\theta_i)$ ist typischerweise höher, wenn keine Messungen von Phasenmessgeräten sondern nur von RTUs vorliegen. Daher wird die Unsicherheit eines Phasenwinkels als auffällig

betrachtet, wenn $u(\theta_i) > 0,03°$ gilt. Für Spannungsmagnituden wird die Standardabweichung der Messgeräte als Schwellenwert für eine Auffälligkeit verwendet, d.h. die geschätzten Spannungsmagnituden sollten nicht mehr als die maximale Standardabweichung der Messgeräte im Vergleich zu einem Szenario mit vollem Trust abweichen. Dieser Ansatz ist für Phasenwinkel nicht praktikabel, da in dem Versuchsaufbau weder Messgeräte noch Standardabweichungen für diese verfügbar sind. Es wird das Dreifache der maximalen Standardabweichung der Messgeräte für Spannungsmagnituden im Vergleich zu einem Szenario mit vollem Trust als Schwellenwert für eine Auffälligkeit verwendet.

Für die Demonstration werden die folgenden Hypothesen bezüglich der Auswirkungen eines reduzierten Trusts von Messwerten auf die geschätzten Zustandsvariablen und deren Unsicherheiten aufgestellt:

1. Der Trust in Messwerte einer einzelnen Sammelschiene beeinflusst den Trust in die Zustandsvariablen in den meisten Fällen nicht auffällig.
2. Der Trust in Messwerte mehrerer Sammelschienen beeinflusst den Trust in die Zustandsvariablen in den meisten Fällen auffällig.
3. Der Trust in Messwerte einer einzelnen Sammelschiene beeinflusst die Schätzung der Zustandsvariablen in den meisten Fällen nicht auffällig.
4. Der Trust in Messwerte mehrerer Sammelschienen beeinflusst die Schätzung der Zustandsvariablen in den meisten Fällen auffällig.

Der Grund für die Hypothesen ist, dass die State Estimation in der Lage sein sollte, andere Messwerte von Sammelschienen in der Nachbarschaft zu verwenden, wenn nur Messwerte von einer Sammelschiene betroffen sind. Wenn eine komplette Nachbarschaft betroffen ist, sind die Abweichungen (Fehler) der Messwerte nicht mehr, wie von der State Estimation erwartet, unabhängig (siehe Unterabschnitt 2.2.1).

Der Versuchsaufbau für die Demonstration gliedert sich in den Aufbau des CPES und den Aufbau der Trust-Erhebung. Der physikalische Teil des zu untersuchenden CPES ist das in Abbildung 5.2 gezeigte IEEE 39-Bus-System [Pai+89]. Es besteht aus 29 PQ-Sammelschienen, d.h. Sammelschienen, für die P_i und Q_i verfügbar sind, 9 PV-Sammelschienen, d.h. Sammelschienen, für die P_i und V_i verfügbar sind, und einer Referenzsammelschiene, d.h. einer Sammelschiene, für die V_i verfügbar ist und θ_i als $0°$ definiert ist. Der Versuchsaufbau sieht eine RTU pro Sammelschiene vor, die die Messwerte der jeweiligen Sammelschiene überträgt. Es wird angenommen, dass das IKT-System wie das Stromnetz aufgebaut ist, d.h. 39 Router, einer pro RTU, die entsprechend den Verbindungen des Stromnetzes verbunden sind. Es wird ferner

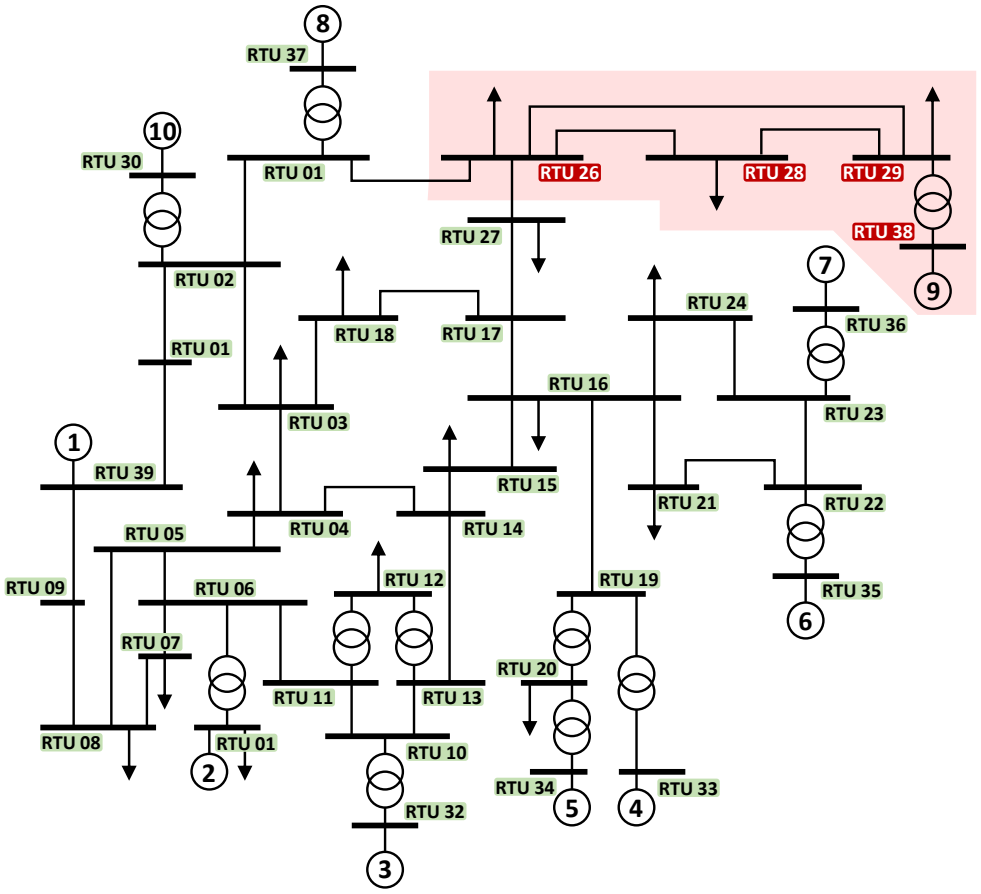


Abb. 5.2.: Das IEEE 39-Bus System [Bra+19a]. Relevante Sammelschienen sind hervorgehoben.

angenommen, dass der Router an Sammelschiene 16 mit dem Router der Leitwarte verbunden ist. Die Konfiguration des ASE ist die folgende. Er bricht den iterativen Prozess ab, wenn die Verbesserung gegenüber der letzten Iteration kleiner oder gleich $\epsilon = 0,001$ ist oder wenn er 50 Iterationen benötigt.

Abbildung 5.3 zeigt eine Instanziierung der Trust Assessment Pyramid (vgl. Abbildung 3.7 in Unterabschnitt 3.2.2) für die Demonstration. Abgeleitete Untersuchungsgegenstände sind die Messwerte. Sie leiten sich in dem Versuchsaufbau aus Messgeräten, RTUs und Routern als Untersuchungsgegenstände ab. Trust-Inputs sind die Standardabweichungen der Messgeräte, CPU-Auslastungsinformationen der RTUs und Router sowie Netzwerkverkehrsinformationen. Entsprechend werden drei Anomaliedetektoren verwendet.

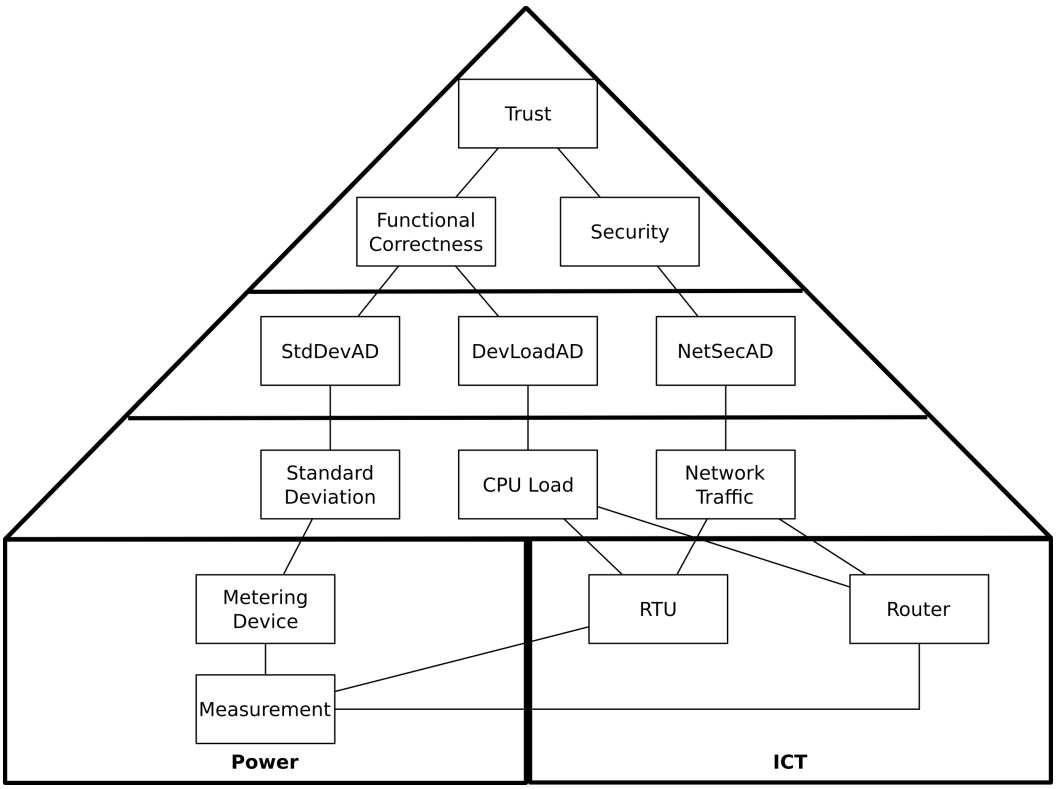


Abb. 5.3.: Eine Instantiierung der Trust Assessment Pyramid für die Demonstration aus Unterabschnitt 3.2.2 [Bra+20].

$$t_{z,StdDevAD} = (StdDevAD, p_{z,stdDevAD}) \text{ mit}$$
$$p_{z,stdDevAD} = 1 - stdDev_{m_z} \tag{5.6}$$

Der erste Anomaliedetektor transformiert die Standardabweichung der Messgeräte in einen einfachen Trust-Wert für die funktionale Korrektheit, wie in Gleichung 5.6 gezeigt. Dabei ist z ein Messwert, m_z das Gerät, das z misst, und $stdDev_{m_z}$ die Standardabweichung dieses Messgeräts. Der zweite Anomaliedetektor wird „Netzwerksicherheitsanomaliedetektor“ genannt und liefert einen einfachen Trust-Wert für die Informationssicherheit. Er basiert auf Alarmen von einem IDS. Die Berechnung eines einfachen Trust-Wertes auf Basis von Alarmen für potenziell mehrere Geräte, die an der Datenerfassung des Messwertes beteiligt sind, basiert auf dem Vorgehen von Liu et al. [Liu+15], das in Abschnitt 5.1 vorgestellt wurde.

$$
\begin{aligned}
t_{z,NetSecAD} &= (NetSecAD, p_{z,NetSecAD}) \text{ mit} \\
p_{z,NetSecAD} &= \frac{|D|}{\sum_{d \in D} \Omega_d}
\end{aligned}
\tag{5.7}
$$

Basierend auf dem Netzwerkauswirkungsfaktor aus Gleichung 5.1 wird von dem Netzwerksicherheitsanomaliedetektor ein einfacher Trust-Wert für ein z wie in Gleichung 5.7 definiert. D ist die Menge der an der Datenakquise von z beteiligten Komponenten. Für unendlich viele IDS-Alarme gilt $\exists d \in D : \Omega_d \to \infty \Rightarrow \lim_{(\sum_{d \in D} \Omega_d) \to \infty} p_{z,NetSecAD} = 0$ und für keine Alarme gilt $\Omega_d = 1 \; \forall d \in D \Rightarrow p_{z,NetSecAD} = 1$.

$$
\begin{aligned}
t_{d,DevLoadAD} &= (DevLoadAD, p_{d,DevLoadAD}) \text{ mit} \\
p_{d,DevLoadAD} &= \begin{cases}
0 & l_{d,CPU,5} > 5 \cdot c_d \\
0,25 & c_d < l_{d,CPU,5} \leq 5 \cdot c_d \\
0,5 & 0,7 \cdot c_d < l_{d,CPU,5} \leq c_d \\
1 & \text{else}
\end{cases}
\end{aligned}
\tag{5.8}
$$

Der dritte Anomaliedetektor heißt „Geräteauslastungsanomaliedetektor“ und liefert einen einfachen Trust-Wert für die funktionale Korrektheit. Er basiert auf CPU-Auslastungsinformationen aus einem IT-Monitoringsystem. Wie in Gleichung 5.8 gezeigt, wird die Trust-Wahrscheinlichkeit eines einzelnen Geräts d basierend auf der durchschnittlichen CPU-Auslastung der letzten fünf Minuten ($l_{d,CPU,5}$) berechnet. c_d ist die Anzahl der verfügbaren CPU-Kerne. Die Berechnung eines einfachen Trust-Wertes auf Basis der durchschnittlichen Lastinformationen basiert auf [Lew19]. Es wurde der Durchschnitt der letzten fünf Minuten gewählt, da der Durchschnitt der letzten Minute zu volatil ist und der Durchschnitt der letzten fünfzehn Minuten in dem vorliegenden Szenario, in dem alle fünfzig Millisekunden Messwerte erhalten werden, zu grob ist. Die Schwellenwerte von $5 \cdot c_d$, c_d und $0,7 \cdot c_d$ basieren auf den in [Lew19] beschriebenen Regeln.

$$
\begin{aligned}
t_{z,DevLoadAD} &= (DevLoadAD, p_{z,DevLoadAD}) \text{ mit} \\
p_{z,DevLoadAD} &= \frac{1}{|D|} \sum_{d \in D} p_{d,DevLoadAD}
\end{aligned}
\tag{5.9}
$$

Auf Grundlage der Geräteauslastungsmetrik für einzelne Geräte wird ein einfacher Trust-Wert für ein z wie in Gleichung 5.9 definiert. D ist die Menge der an der Datenerfassung von z beteiligten Geräte. Die Grenzen von $p_{z,DevLoadAD}$ sind 1 für $p_{d,DevLoadAD} = 1 \; \forall d \in D$ und 0 für $p_{d,DevLoadAD} = 0 \; \forall d \in D$.

Basierend auf dem Versuchsaufbau wurde in sieben verschiedenen Szenarien der Einfluss eines hoch priorisierten IDS-Alarms, einer hohen durchschnittlichen Geräteauslastung und von beidem untersucht. Weiterhin wurde das Verhalten bei einer reduzierten Trust-Wahrscheinlichkeit für eine RTU, konkret an Sammelschiene 26, oder für einen Router, der in die Datenakquise von Messwerten mehrerer RTUs involviert ist, konkret der Router an Sammelschiene 26, untersucht. Die Szenarien sind die folgenden:

1. keine Anomaliedetektoren außer der statischen Standardabweichung der Messgeräte (Baselinie),
2. ein IDS-Alarm für RTU 26,
3. eine hohe durchschnittliche CPU-Auslastung für RTU 26,
4. eine Kombination der Szenarien 2 und 3,
5. ein IDS-Alarm für Router 26,
6. eine hohe durchschnittliche CPU-Auslastung für Router 26 und
7. eine Kombination der Szenarien 5 und 6.

$p_{z,NetSecAD}$ wurde basierend auf den Gleichungen 5.1 und 5.7 mit $m = 2$ und $p(k) = 3$ für einen einzelnen Alarm und fünf beteiligten Routern berechnet. Die Trust-Wahrscheinlichkeit ist $p_{z,NetSecAD} = \frac{6}{5+\sqrt{9}} = 0,75$ für alle z, die von RTU 26 in Szenario 2 bereitgestellt werden. In Szenario 5 ist $|D|$ 6, 7, 8 bzw. 9 für Messwerte, die von RTU 26, 28, 29 bzw. 38 bereitgestellt werden. $p_{z,DevLoadAD}$ wurde basierend auf den Gleichungen 5.8 und 5.9 mit $c_d = 1$ und einer durchschnittlichen Last der letzten fünf Minuten von $c_d < l_{d,CPU,5} \leq 5 \cdot c_d$ berechnet. Die Trust-Wahrscheinlichkeit ist $p_{z,DevLoadAD} = \frac{5+0,25}{6} = 0,875$ für alle z, die von RTU 26 in Szenario 3 bereitgestellt werden. In Szenario 6 ist $|D|$ wie in Szenario 5. Für Szenario 4 und Szenario 7 wurde die Multiplikation aller einzelnen Trust-Wahrscheinlichkeiten ($p_{z,StdDevAD} \cdot p_{z,NetSecAD} \cdot p_{z,DevLoadAD}$) verwendet, um die Trust-Wahrscheinlichkeiten zu einer einzigen zusammenzufassen.

Tabelle 5.1 gibt einen Überblick über die wichtigsten Ergebnisse auf einer netzwerkweiten Skala. Die Anzahl der verwendeten Iterationen im State-Estimation-Prozess

Tab. 5.1.: Überblick über die Ergebenisse der Unsicherheitsanalyse für die entsprechenden Szenarien im Vergleich zu Szenario 1.

Szenario	1	2	3	4	5	6	7
# Iterationen	6	8	6	24	50	50	50
min. $\Delta V_i[p.u.]$	-	0	0	0	0	0	0
max. $\Delta V_i[p.u.]$	-	0,001	0	0,001	0,001	0,001	0,001
$\varnothing \Delta V_i[p.u.]$	-	≈ 0	0	≈ 0	≈ 0	≈ 0	≈ 0
# auffällige ΔV_i	-	0	0	0	0	0	0
min. $u(V_i)[p.u.]$	0	0,001	0,001	0,001	0,001	0,001	0,001
max. $u(V_i)[p.u.]$	0,001	0,003	0,002	0,004	0,11	0,051	0,146
$\varnothing u(V_i)[p.u.]$	≈ 0	0,001	0,001	0,001	0,013	0,006	0,0171
# auffällige $u(V_i)$	0	3	1	4	31	28	30
min. $\Delta \theta_i[^\circ]$	-	0	0	0	0	0	0
max. $\Delta \theta_i[^\circ]$	-	0,002	0,002	0,002	0,003	0,001	0,003
$\varnothing \Delta \theta_i[^\circ]$	-	0,001	0,001	0,001	0,002	≈ 0	0,002
# auffällige $\Delta \theta_i$	-	0	0	0	0	0	0
min. $u(\theta_i)[^\circ]$	0,001	0,001	0,001	0,001	0,001	0,001	0,001
max. $u(\theta_i)[^\circ]$	0,036	0,592	0,313	0,82	3,13	1,694	4,011
$\varnothing u(\theta_i)[^\circ]$	0,024	0,391	0,206	0,541	0,754	0,537	0,977
# auffällige $u(\theta_i)$	3	38	38	38	38	38	38

und sogar, ob er konvergiert oder nicht, unterscheidet sich für die einzelnen Szenarien. In Szenario 1 werden sechs Iterationen benötigt. Bei einer verringerten Trust-Wahrscheinlichkeit der Messwerte einer einzelnen Sammelschiene (Szenarien 2 bis 4) konvergiert der ASE und die verwendeten Iterationen steigen mit einer Abnahme der Trust-Wahrscheinlichkeit der Messwerte. Der ASE konvergiert nicht, wenn die Trust-Wahrscheinlichkeit der Messwerte von mehreren Sammelschienen verringert wird (Szenarien 5 bis 7). Dies zeigt den Einfluss der Änderung der Trust-Wahrscheinlichkeit in Bezug auf die Standardabweichungen der Eingangsmesswerte auf das Verhalten des State Estimators.

Die Werte V_i und θ_i werden immer mit den Werten aus Szenario 1 verglichen, während die Werte $u(V_i)$ und $u(\theta_i)$ absolut sind. Die Anzahl an Sammelschienen mit auffälligen Abweichungen oder Unsicherheiten werden auf der Grundlage der Annahmen über auffällige Werte berechnet. Die Ergebnisse stimmen nicht mit der ersten Hypothese überein (keine auffälligen Unsicherheiten in den Szenarien 2 bis 4). Es gibt auffällige Werte für bis zu vier V_i- und achtunddreißig θ_i-Werte. Die $u(V_i)$-Werte sind gering (max. $0,004\ p.u.$), die $u(\theta_i)$-Werte jedoch hoch (max. $0,82^\circ$). Für die zweite Hypothese (auffällige Unsicherheiten in den Szenarien 5 bis 7) sind die Ergebnisse wie erwartet. Es gibt bis zu einunddreißig auffällige $u(V_i)$-

und achtunddreißig auffällige $u(\theta_i)$-Werte. Die Werte können dabei hoch sein (max. $0,146\ p.u.$ bzw. $4,011°$). Die Ergebnisse stimmen auch mit der dritten und vierten Hypothese überein.

Die Ergebnisse stimmen mit drei von vier Hypothesen überein. Die Tatsache, dass die Daten nicht mit der ersten Hypothese übereinstimmen, ist ein gutes Ergebnis. Es zeigt, dass zumindest in diesem Aufbau auch die reduzierte Trust-Wahrscheinlichkeit der Messwerte von einer einzigen Sammelschiene die Unsicherheit der zugehörigen Zustandsvariablen beeinflusst.

Eine weitere wichtige Erkenntnis ist, dass der State Estimator ggf. nicht konvergiert, wenn die Trust-Wahrscheinlichkeit mehrerer Messwerte reduziert wird. Der Grund ist, wie bereits angeführt, dass bei einer typischen State Estimation die Messfehler als unabhängig angenommen werden. Daher ist die in diesem Abschnitt vorgestellte Unsicherheitsanalyse keine optimale Lösung. Die Anzahl benötigter Iterationen kann aber in eine Trust-Erhebung für die Zustandsvariablen einfließen.

5.3 Variante 2: Entkoppelte Schätzung einfacher Trust-Werte

Anmerkung: Der Inhalt dieses Abschnitts wurde bereits in [BBL21] veröffentlicht.

Die Grundlage der zweiten Variante ist die Tatsache, dass Zustandsvariablen durch die Funktion $\boldsymbol{h}(\boldsymbol{x})$ auf die Messwerte abgebildet werden (Gleichung 2.1 in Abschnitt 2.2.1). Im Zuge der State Estimation wird die Ableitung dieser Funktion und die Inverse der Ableitung gebildet. Die Ableitung $\boldsymbol{H}$ von $\boldsymbol{h}(\boldsymbol{x})$ kann als Sensitivitätsfunktion der Messwerte bzgl. Änderungen der Zustandsvariablen interpretiert werden. Entsprechend kann die Inverse von $\boldsymbol{H}$ als Sensitivitätsfunktion der Zustandsvariablen bzgl. Messwertänderungen betrachtet werden. Der Nachteil bei diesem Verfahren ist, dass $\boldsymbol{h}(\boldsymbol{x})$ nichtlinear und nicht stetig differenzierbar ist, wodurch die Sensitivitäten lediglich Abschätzungen und keine exakt berechenbaren Werte darstellen. Ein weiteres Argument für diesen Lösungsansatz ist, dass die Trust-Schätzung zwar auf Artefakten beruht, die während der State Estimation berechnet werden, aber losgelöst im Nachgang an die State Estimation durchgeführt werden kann. Im Folgenden wird zunächst die Methodik detailliert dargestellt. Im Anschluss werden Ergebnisse von Versuchen mit diesem Ansatz [BBL21] präsentiert, die die Vermutung untermauern, dass sich dieser Ansatz besser eignet als der der Unsicherheitsanalyse.

5.3.1 Methodik

Seien die Sensitivitäten $\boldsymbol{S}(\boldsymbol{x})$ der Zustandsvariablen $\boldsymbol{x}$ bzgl. Änderungen bei den Messwerten wie in Gleichung 5.10 definiert, wobei † das Bilden einer Pseudoinversen symbolisiert.

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} s_{1,1} & \dots & s_{1,m} \\ \vdots & \ddots & \vdots \\ s_{n,1} & \dots & s_{n,m} \end{bmatrix} = \begin{bmatrix} (\frac{\partial \boldsymbol{h}_1(\boldsymbol{x})}{\partial x_1})^\dagger & \dots & (\frac{\partial \boldsymbol{h}_m(\boldsymbol{x})}{\partial x_1})^\dagger \\ \vdots & \ddots & \vdots \\ (\frac{\partial \boldsymbol{h}_1(\boldsymbol{x})}{\partial x_n})^\dagger & \dots & (\frac{\partial \boldsymbol{h}_m(\boldsymbol{x})}{\partial x_n})^\dagger \end{bmatrix} = (\frac{\partial \boldsymbol{h}(\boldsymbol{x})}{\partial \boldsymbol{x}})^\dagger \tag{5.10}$$

Dann kann auf Grundlage von $\boldsymbol{S}(\boldsymbol{x})$ ein Gewichtsfaktor $w_{i,j,S}(\boldsymbol{x})$ für jede Kombination aus Zustandsvariable x und Messwert z definiert werden, der bestimmt, wie stark einfache Trust-Werte von z in die einfachen Trust-Werte von x einfließen.

$$w_{i,j,S}(\boldsymbol{x}) = \frac{|s_{i,j}(\boldsymbol{x})|}{\|s_{i,j}(\boldsymbol{x})\|} = \frac{|s_{i,j}(\boldsymbol{x})|}{\sqrt{\sum_{l=1}^{m} s_{i,l}^2}} \tag{5.11}$$

Die mathematische Beschreibung des Gewichtsfaktors ist Gleichung 5.11 zu entnehmen, wobei die Gewichtsfaktoren durch die Euklidische Norm normalisiert werden. Neben den Gewichtsfaktoren auf Basis der Sensitivitäten können ebenfalls die Messwertresiduen e, die nach der State Estimation zur Verfügung stehen, verwendet werden (siehe Abschnitt 2.2.1). Die Idee dabei ist, einfache Trust-Werte von Messwerten mit höherem Residuum geringer zu gewichten, da sie schlechter zum Modell passen und daher weniger Einfluss auf die Zustandsvariablen haben.

$$w_{i,j,e}(\boldsymbol{x}) = \frac{1 + \|\boldsymbol{e}\|_\infty - |e_j|}{\|1 + \|\boldsymbol{e}\|_\infty - |e_j|\|} = \frac{1 + \|\boldsymbol{e}\|_\infty - |e_j|}{\sqrt{\sum_{l=1}^{m}(1 + \|\boldsymbol{e}\|_\infty - |e_l|)^2}} \tag{5.12}$$

Eine Funktion um Messwerteresiduen in Gewichtsfaktoren umzurechnen kann wie in Gleichung 5.12 aussehen, wobei $\|\boldsymbol{e}\|_\infty = max_{0 \leq i < m}|e_i|$ die Supremumsnorm für die Messwerteresiduen ist. Durch die Subtraktion vom Maximum resultiert ein maximales Residuum in einem Gewichtsfaktor von 0 und ein minimales Residuum in dem höchsten Gewichtsfaktor. Messwerte mit dem höchsten Residuum würden demnach ignoriert werden, weshalb der Wert um 1 erhöht wird. Die Gewichtsfaktoren werden zusätzlich durch die Euklidische Norm normalisiert.

$$p_{x_i,\gamma} = \frac{\sum_{j=1}^{m} w_{i,j,S} \cdot w_{i,j,e} \cdot p_{z_j,\gamma}}{\sum_{j=1}^{m} w_{i,j,S} \cdot w_{i,j,e}} \tag{5.13}$$

Gleichung 5.13 zeigt für eine Zustandsvariable x_i die resultierende Berechnung einer geschätzten Trust-Wahrscheinlichkeit auf Basis der Trust-Wahrscheinlichkeiten der Messerte z und den Gewichtsfaktoren aus den Gleichungen 5.11 und 5.12. Gleichung 5.13 kann für jeden Trust-Schätzer angewandt werden, so dass die Zustandsvariablen für jeden einfachen Trust-Wert der Messwerte einen entsprechenden einfachen Trust-Wert besitzen.

5.3.2 Demonstration

Der Zweck der durchgeführten Demonstration ist die Validierung bestimmter Hypothesen über die Auswirkungen des reduzierten Trusts in Messwerte auf den geschätzten Trust in die geschätzten Zustandsvariablen. Die Hypothesen sind die folgenden:

1. Die Schätzung des Trusts in die Zustandsvariablen hat keinen Einfluss auf den State-Estimation-Prozess, d.h. auf die geschätzten Zustandsvariablen und die Konvergenz.
2. Ein reduzierter Trust in Messwerte einer einzelnen Sammelschiene reduziert den Trust in Zustandsvariablen nicht auffällig.
3. Ein reduzierter Trust in Messwerte mehrerer Sammelschienen reduziert zwar den Trust in die entsprechenden Zustandsvariablen auffällig, nicht aber den Trust in andere.

Der Begriff auffällig wird für diese Demonstration wie folgt definiert:

Definition 14 (Auffällig)**.** *Ein einfacher Trust-Wert $t_{x,\gamma} = (\gamma, p)$ einer geschätzten Zustandsvariable x ist auffällig reduziert, wenn p unter einem gegebenen Schwellwert ϵ_p liegt: auffällig $\Leftrightarrow p < \epsilon_p$.*

ϵ_p ist ein wichtiger und nicht einfach zu definierender Parameter. Er hängt vom System, vom Anwendungsfall und von der Erfahrung mit abnormalen Ereignissen und den daraus abgeleiteten einfachen Trust-Werten ab. Daher kann es kein ϵ_p geben, das für alle Szenarien geeignet ist. Eine Möglichkeit besteht darin, das Maximum der in der Vergangenheit berechneten Trust-Wahrscheinlichkeiten, von denen bekannt ist, dass sie auffällig sind, plus einen Puffer von 10% zu verwenden. Wenn es z.B. in der Vergangenheit eine Trust-Wahrscheinlichkeit von $0,8$ als höchste Trust-Wahrscheinlichkeit für ein Ereignis gab, von dem bekannt ist, dass es auffällig ist, sollte ϵ_p $0,88$ betragen.

Der Grund für die Hypothesen ist, dass der State Estimator in der Lage sein sollte, andere Messwerte von Sammelschienen in der Nachbarschaft zu verwenden, wenn nur Messwerte einer Sammelschiene betroffen sind. Wenn eine komplette Nachbarschaft betroffen ist, sind die Abweichungen (Fehler) der Messwerte nicht mehr, wie vom State Estimator erwartet, unabhängig (vgl. Unterabschnitt 2.2.1).

Tab. 5.2.: Überblick über die Ergebnisse der Schätzung einfacher Trust-Werte für die entsprechenden Szenarien im Vergleich zu Szenario 1.

Szenario	1	2	3	4	5	6	7
# Iterationen	6						
max. $\Delta V_i[p.u.]$	-	0					
max. $\Delta\theta_i[^\circ]$	-	0					
max. $u(V_i)[p.u.]$	-	0,001					
max. $u(V_i)[^\circ]$	-	0,036					
manipulierter Bereich							
min. $p_{x_i,StdDevAD}$	0,999						
max. $p_{x_i,StdDevAD}$	0,999						
$\varnothing p_{x_i,StdDevAD}$	0,999						
min. $p_{x_i,NetSecAD}$	-	0,994	-	0,994	0,874	-	0,874
max. $p_{x_i,NetSecAD}$	-	0,996	-	0,996	0,977	-	0,977
$\varnothing p_{x_i,NetSecAD}$	-	0,995	-	0,995	0,936	-	0,936
min. $p_{x_i,DevLoadAD}$	-	-	0,997	0,997	-	0,942	0,942
max. $p_{x_i,DevLoadAD}$	-	-	0,998	0,998	-	0,989	0,989
$\varnothing p_{x_i,DevLoadAD}$	-	-	0,998	0,998	-	0,971	0,971
# auffällig reduzierte p	0	0	0	0	4	3	4
nicht manipulierter Bereich							
min. $p_{x_i,StdDevAD}$	0,999						
max. $p_{x_i,StdDevAD}$	1						
$\varnothing p_{x_i,StdDevAD}$	0,999						
min. $p_{x_i,NetSecAD}$	-	0,995	-	0,995	0,968	-	0,968
max. $p_{x_i,NetSecAD}$	-	1	-	1	1	-	1
$\varnothing p_{x_i,NetSecAD}$	-	0,997	-	0,997	0,988	-	0,988
min. $p_{x_i,DevLoadAD}$	-	-	0,997	0,997	-	0,985	0,985
max. $p_{x_i,DevLoadAD}$	-	-	1	1	-	1	1
$\varnothing p_{x_i,DevLoadAD}$	-	-	0,999	0,999	-	0,994	0,994
# auffällig reduzierte p	0	0	0	0	0	0	0

Der Versuchsaufbau ist der gleiche wie in Teilabschnitt 5.2.2. Tabelle 5.2 gibt einen Überblick über die wichtigsten Ergebnisse der Experimente. Oben sind die Eigenschaften des State Estimators aufgeführt: Anzahl der verwendeten Iterationen, die maximale Abweichung der geschätzten Zustandsvariablen im Vergleich zu Szenario 1 und die maximalen Unsicherheiten der geschätzten Zustandsvariablen, die vom State Estimator auf der Grundlage der Standardabweichungen der Messwerte berechnet

werden. Der State Estimator verwendet in allen Szenarien sechs Iterationen, es gibt keine Abweichung in den geschätzten Zustandsvariablen und sie haben eine geringe Unsicherheit. Dies zeigt, dass die Schätzung des Trusts keinen Einfluss auf das Verhalten des State Estimators hat. Daher stützen die Ergebnisse Hypothese 1. Die Anzahl der Zustandsvariablen mit auffällig reduzierter Trust-Wahrscheinlichkeit wird auf Basis der Annahmen zu auffälligen Werten bestimmt (auffällig $\Leftrightarrow p_{x_i,\gamma} < 0,9625$). In den Szenarien 2, 3 und 4 werden nur Messwerte von einer einzelnen RTU mit einem reduzierten Trust annotiert. Alle geschätzten Trust-Wahrscheinlichkeiten für die Zustandsvariablen sind in diesen Szenarien nicht auffällig reduziert. Sie reichen von $0,994$ bis $0,998$ im manipulierten Teil (das sind die Werte von RTU 26 für diese Szenarien) und $0,995$ bis $1,0$ für den Rest des Netzes. Dies stimmt mit Hypothese 2 überein.

In den Szenarien 5, 6 und 7 werden die Messwerte von RTU 26, 28, 29 und 38 mit einem reduzierten Trust annotiert. Ein IDS-Alarm für Router 26 ($p_{x_i,NetSecAD}$) führt zu geschätzten Trust-Wahrscheinlichkeiten im manipulierten Teil, die zwischen $0,874$ und $0,977$ liegen. Vier Zustandsvariablen im manipulierten Teil haben einen auffällig reduzierten Trust. Diese Zustandsvariablen sind die Spannungsmagnituden. Der Grund dafür ist, dass die geschätzten Spannungsmagnituden stark von den gemessenen Spannungsmagnituden abhängen, während zur Schätzung der Spannungsphasenwinkel mehr Messwerte beitragen, da keine Phasenmessgeräte zur direkten Messung der Spannungsphasenwinkel eingesetzt werden. Für den nicht manipulierten Teil des Netzes gibt es keine auffällig reduzierten Trust-Wahrscheinlichkeiten ($0,968$ bis $1,0$). Eine hohe mittlere CPU-Auslastung auf dem Router 26 ($p_{x_i,DevLoadAD}$) führt im manipulierten Teil zu geschätzten Trust-Wahrscheinlichkeiten, die zwischen $0,942$ und $0,989$ liegen. Drei Zustandsvariablen im manipulierten Teil haben einen auffällig reduzierten Trust. Diese Zustandsvariablen sind alle Spannungsmagnituden mit Ausnahme von Sammelschiene 26. Der höhere Trust in die Zustandsvariablen, die sich auf Sammelschiene 26 beziehen, ist jedoch nachvollziehbar, da Sammelschiene 26 die Sammelschiene ist, die den manipulierten Teil mit dem Rest des Netzes verbindet (vgl. Abbildung 5.2). Für den nicht manipulierten Teil des Netzes gibt es keinen auffällig reduzierten Trust ($0,985$ bis $1,0$). Zusammenfassend lässt sich sagen, dass die Ergebnisse der Szenarien 5, 6 und 7 Hypothese 3 untermauern.

Der State-Estimation-Prozess wird durch die Trust-Schätzung nicht mehr beeinflusst, d.h. er konvergiert immer und die geschätzten Zustandsvariablen sind immer gleich (vgl. Anzahl der Iterationen, maximale V_i- und θ_i-Abweichungen in Tabelle 5.2). In den Ergebnissen in Tabelle 5.1 aus Unterabschnitt 5.2.2 ist eine große Menge von Zustandsvariablen, d.h. bis zu 68 von 78, in Form von hohen Unsicherheiten betroffen. Mit dem Ansatz, der in diesem Abschnitt vorgestellt wurde, sind nur Zustandsva-

riablen im manipulierten Teil betroffen. Darüber hinaus geben die multivariaten Trust-Werte der geschätzten Zustandsvariablen im Vergleich zu ihrer Unsicherheit einen deutlicheren Eindruck über das potenzielle Problem.

5.4 Zusammenfassung

In diesem Kapitel wurden zwei Alternativen vorgestellt um die multivariaten Trust-Werte von Messwerten in eine State Estimation einfließen zu lassen. Dazu wurden zunächst in Abschnitt 5.1 verwandte Arbeiten aus der Literatur vorgestellt. Die beiden als relevant für die vorliegende Arbeit betrachteten Arbeiten aus der Literatur lassen ebenfalls zusätzliche Informationen in die State Estimation einfließen und stellen daher interessante Ansätze für eine Trust-sensitive Lagebilderkennung dar. Allerdings sind sie insofern nicht für den Einsatz bei einem multivariaten Trust-Modell geeignet, als dass die Multivarietät in beiden Arbeiten durch die Integration in die State Estimation verloren gehen würde. Im Anschluss wurden die beiden in dieser Arbeit betrachteten Alternativen zur Einbringung multivariater Trust-Werte in eine State Estimation vorgestellt:

1. Über eine Unsicherheitsanalyse: Der multivariate Trust-Wert pro Messwert wird zu einer Trust-Wahrscheinlichkeit im Wertebereich $[0; 1]$ aggregiert und in eine Standardabweichung umgewandelt. Mit diesen Standardabweichungen für die Messwerte werden dann Unsicherheiten für die Zustandsvariablen berechnet [Bra+20]. Dieses Vorgehen wurde in Abschnitt 5.2 beschrieben.

2. Über eine entkoppelte Schätzung einfacher Trust-Werte: Für jeden einfachen Trust-Wert der Messwerte wird auf Basis von Sensitivitäten der Zustandsvariablen bzgl. Messwertänderungen ein entsprechender einfacher Trust-Wert der Zustandsvariable geschätzt [BBL21]. Dieses Vorgehen wurde in Unterabschnitt 5.3 beschrieben.

Für beide Varianten wurde neben der Methodik auch eine Demonstration mit gleichem Versuchsaufbau vorgestellt. Es zeigt sich, dass die erste Variante u.a. das Konvergenzverhalten des State Estimators insofern beeinflusst, als dass dieser nicht mehr in allen Szenarien konvergiert. Ferner geht auch, wie bei den verwandten Arbeiten, die Multivarietät der Trust-Werte verloren. Somit ist anhand des State-Estimation-Ergebnisses nicht mehr ersichtlich, welche Trust-Facetten von einem reduzierten Trust betroffen sind und welche Trust-Inputs zu diesem Ergebnis geführt haben. Die zweite Alternative weist diese Nachteile nicht auf, weshalb sie im folgenden Kapitel bei der Umsetzung von ASSESS verwendet wird.

6 Anomaliesensitive State Estimation mit Streaming Systemen

> *Der Wert einer Idee liegt in ihrer Umsetzung.*
>
> **— Thomas Alva Edison**

In diesem Kapitel wird das im Rahmen dieser Arbeit entwickelte System namens anomaliesensitive State Estimation mit Streaming Systemen (ASSESS) vorgestellt. ASSESS integriert dabei Umsetzungen der in den letzten drei Kapiteln vorgestellten Lösungen zur Erreichung der Forschungsziele (kontextsensitives, multivariates Trust-Modell, Integrationsplattform für Trust-Schätzer und Trust-sensitive State Estimation) und ermöglicht die Erfüllung der nichtfunktionalen Anforderungen (Aktualität, technische Interoperabilität, Prozessinteroperabilität, Flexibilität und Skalierbarkeit).

Zunächst wird in Abschnitt 6.1 mit Odysseus [App+12; Bol+09], einem Framework zur Erstellung maßgeschneiderter DSMSs, die konkrete technologische Grundlage für die Umsetzung von ASSESS vorgestellt. Abschnitt 6.2 widmet sich anschließend der Architektur von ASSESS. Dabei werden die Herausforderungen und Umsetzungsaspekte, die spezifisch für die nichtfunktionalen Anforderungen sind, in den folgenden Abschnitten 6.3 (Flexibilität und Skalierbarkeit), 6.4 (technische und Prozessinteroperabilität) und 6.5 (Aktualität) detaillierter erläutert. In Abschnitt 6.6 wird mithilfe einer durchgeführten Demonstration überprüft, ob die die Forschungsziele erreicht wurden, während eine Evaluation der nichtfunktionalen Anforderungen im Folgekapitel behandelt wird. Dieses Kapitel endet mit einer Zusammenfassung in Abschnitt 6.7.

Anmerkung: Große Teile des folgenden Kapitels sind ebenfalls in [BEL23] veröffentlicht.

6.1 Odysseus

Odysseus ist ein Framework für maßgeschneiderte DSMSs, mit dem es möglich ist, unterschiedliche, an gegebene Anwendungsszenarien angepasste DSMSs zu erstellen [App+12; Bol+09]. Es ist daher im Vergleich zu anderen DSMSs, wie beispielsweise Aurora [Hwa+05], Borealis [Ahm+05], OSIRIS-SE [BSS05] oder Stormy [Mer10], flexibler und leichter erweiterbar. Aus diesem Grund wurde Odysseus ausgewählt, um ASSESS umzusetzen.

Odysseus basiert auf der Open Services Gateway initiative (OSGi)-Service-Platform, einem dynamischen Modulsystem für Java. Mit der OSGi-Service-Platform ist es möglich, Anwendungen modular aufzubauen, so dass einzelne Module, die im Folgenden Plug-ins genannt werden, hinzugefügt, geändert oder entfernt werden können, ohne dass andere Plug-ins der Anwendung verändert werden müssen. Ein weiteres Merkmal der OSGi-Service-Platform stellen so genannte Services dar. Unter einem Service wird bei OSGi ein Java-Objekt verstanden, das in einem Plug-in erzeugt und dennoch systemweit zur Verfügung gestellt wird. Bei einer Serviceabfrage spielt es keine Rolle, in welchem Plug-in dieser erzeugt wurde. Es ist ausschließlich wichtig, dass es einen entsprechenden Service gibt und andere Plug-ins diesen nutzen können [Wüt08].

Der Flexibilität von Odysseus liegen viele sogenannte Variationspunkte zugrunde. Diese Variationspunkte sind OSGi-Services und durch die Einbindung unterschiedlicher Plug-ins können verschiedene Implementierungen verwendet werden. Die Anzahl sogenannter Fixpunkte, also Strukturen, für die eine feste Implementierung vorgesehen ist, ist auf ein Minimum beschränkt. In der Regel handelt es sich dabei um Odysseus-interne Verwaltungsstrukturen [App+12; Bol+09].

Abbildung 6.1 [Bol+09] gibt einen Überblick über die Architektur von Odysseus. Der linke Teil der Abbildung stellt ein Modell einer Datenstromverarbeitung von Datenquellen (engl. sources) bis zu Senken (engl. sinks) dar. Die sich dazwischen befindlichen Kanäle (engl. pipes) sind Datenstromoperatoren [App+12]. Damit unterschiedliche Datenquellen an Odysseus angebunden und neue Arten von Datenquellen verwendet werden können, ohne dass bestehende Komponenten angepasst werden müssen, verwendet Odysseus ein sogenanntes Access Framework. Dieses Framework sieht vor, Transport-, Protokoll- und Datenhandhabung für Datenquellen zu trennen und in separate Services auszulagern. So ist es möglich, neue Service-Implementierungen, z.B. TCP/IP für den Transport, hinzuzufügen [Ody]. Die Flexibilität des Access Frameworks stellt eine gute Grundlage dar, um die nichtfunktionalen Anforderungen der Flexibilität und technischen Interoperabilität zu erfüllen.

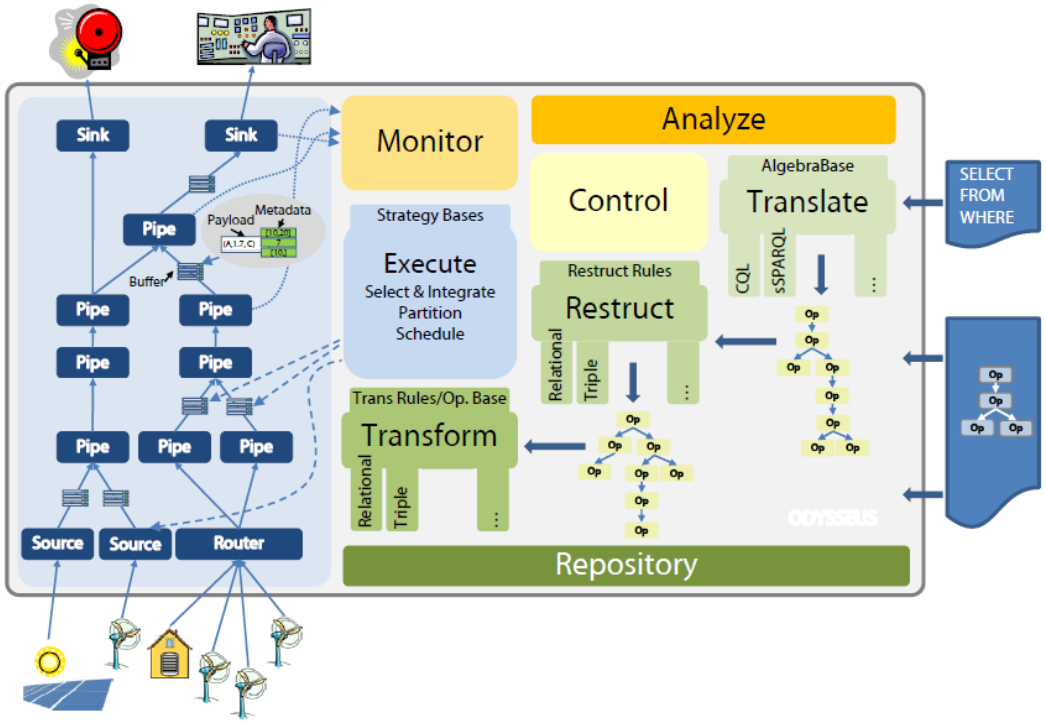


Abb. 6.1.: Die Architektur von Odysseus [Bol+09].

Der rechte Teil von Abbildung 6.1 stellt die verschiedenen Komponenten von Odysseus dar, die u.a. dazu dienen, eine an Odysseus gerichtete Anfrage in einen logischen Anfrageplan zu übersetzen (`Translate`), diesen zu optimieren (`Restruct`), ihn in einen physischen Anfrageplan zu übersetzen (`Transform`) und anschließend auszuführen (`Execute`) [Bol+09]. Neben den in Abbildung 6.1 dargestellten Komponenten verfügt Odysseus auch über eine rollenbasierte Nutzerverwaltung [Ody], die die für die Prozessinteroperabilität notwendige Zugriffskontrolle sicherstellen kann.

```
1  #PARSER CQL
2  #ADDQUERY
3  Select * FROM bid
4
5  #PARSER PQL
6  #ADDQUERY
7  result = PROJECT({ATTRIBUTES=['id','name']}, person)
```

Skript 6.1: Eine beispielhafte Verwendung von Odysseus Script.

Um einem Odysseus-Nutzer bei einigen Varationspunkten zur Laufzeit entscheiden lassen zu können, welche Implementierung verwendet werden soll, stellt Odys-

seus eine eigene regelbasierte Skriptsprache namens Odysseus Script zur Verfügung [Bol+10]. Dem Nutzer ist es dadurch möglich, neben dem eigentlichen Stellen einer Anfrage auf verschiedene Variationspunkte Einfluss zu nehmen. Dies ist allerdings nur möglich, wenn entsprechende Plug-ins mit unterschiedlichen Implementierungen des Services eingebunden sind.

Skript 6.1 zeigt eine beispielhafte Verwendung von Odysseus Script, bei der die Variation darin besteht, dass unterschiedliche Parser (`Translate`-Komponente) in einem Script verwendet werden können. In den ersten drei Zeilen wird eine einfache Selektionsanfrage in der Sprache „Continuous Query Language (CQL)“ erstellt. CQL ist dabei eine deklarative Sprache für das relationale Modell, die große Ähnlichkeiten zu SQL aufweist [ABW02]. In den Zeilen 5 bis 7 wird ebenfalls eine einfache Anfrage (dieses Mal eine Projektion) erstellt. Ein wesentlicher Unterschied zwischen den beiden Anfragetexten liegt darin, dass statt CQL die Odysseus-eigene Anfragesprache „Prozedural Query Language (PQL)“ verwendet wird. PQL ist eine prozedurale und damit leicht erweiterbare und wartbare Anfragesprache [App+12].

```
1  /// Erstes Beispiel:
2  output = OPERATOR({
3      parameter_key_1=parameter_value_1, ...,
           parameter_key_n=parameter_value_n
4    },
5    input_1, ..., input_n
6  )
7  /// Zweites Beispiel:
8  output = OPERATOR({
9      parameter_key_1=parameter_value_1, ...,
           parameter_key_n=parameter_value_n
10   },
11   OPERATOR({
12       parameter_key_1=parameter_value_1, ...,
             parameter_key_n=parameter_value_n
13     },
14     input_1, ..., input_n
15   ),
16   ..., input_n
17 )
```

Skript 6.2: Der Aufbau von PQL.

Die in Odysseus Script durch einen Hashtag gekennzeichneten Begriffe sind so genannte Schlüsselworte. In der Regel wird der Rest der jeweiligen Zeile dann dem Schlüsselwort als Argument zugewiesen und entsprechend ausgewertet (z.B. #PARSER CQL). Es gibt aber auch Schlüsselwörter, deren Argument sich über mehrere Zeilen erstreckt (z.B. #ADDQUERY, wobei das Argument die folgende Anfrage ist). Das in Skript 6.1 dargestellte Vorgehen ist das unter Odysseus-Nutzern am weitesten verbreitete: Es wird eine Datei in Odysseus Script verfasst, in der mehrere Anfragen in einer angegebenen Sprache definiert werden. Neben Odysseus Script, wo Nutzer für eine bestimmte Menge von Anfragen Einstellungen anpassen können, bietet Odysseus die Möglichkeit bestimmte Einstellungen anwendungsweit durch eine Konfigurationsdatei anzupassen [Ody].

Für die Implementierung von ASSESS innerhalb von Odysseus werden Odysseus Script und PQL verwendet. Skript 6.2 zeigt die Syntax von PQL, bei dem prozedural definiert wird, wie Datenströme verarbeitet werden. Ein Operator kann, je nach seiner Funktionsweise, mehrere Eingangsdatenströme verarbeiten und erzeugt i.d.R. einen Ausgabedatenstrom. Der Ausgabedatenstrom kann entsprechend wieder Eingabe für weitere Operatoren sein. Viele Operatoren müssen bzw. können dabei parametriert werden. Quelloperatoren, wie z.b. `ACCESS` oder `RECEIVER` haben keine Eingangsdatenströme, da sie ihre Daten von außerhalb beziehen. Die Anzahl der Eingangsdatenströme hängt, wie erwähnt, von dem Operator ab. Eine relationale Selektion arbeitet z.B. auf einem einzelnen Datenstrom, während eine relationale Verbundoperation auf zwei Datenströmen arbeitet. Es ist auch möglich, Operatoren in PQL derart zu verknüpfen, dass die Ausgabe eines Operators nicht extra in einer Variablen gespeichert wird. Dafür kann, wie ebenfalls in Skript 6.2 zu sehen, eine Eingabe durch einen Operatorblock ersetzt werden.

6.2 Architektur

In diesem Abschnitt wird die Architektur von ASSESS vorgestellt. Sie verbindet PSNA-Trust, eine Integrationsplattform für Trust-Schätzer, die mathematische Vorschrift zur Schätzung des multivariaten Trusts in Zustandsvariablen basierend auf dem multivariaten Trust in Messwerte und weitere Eigenschaften zur Erfüllung der Anforderungen in einem DSMS.

Abbildung 6.2 zeigt das Komponentendiagramm für ASSESS auf höchster Abstraktionsebene. Das Eingangstransformationsframework (engl. input transform. framework) (ITF) links ist der Eingangspunkt für Messwerte und Topologien.

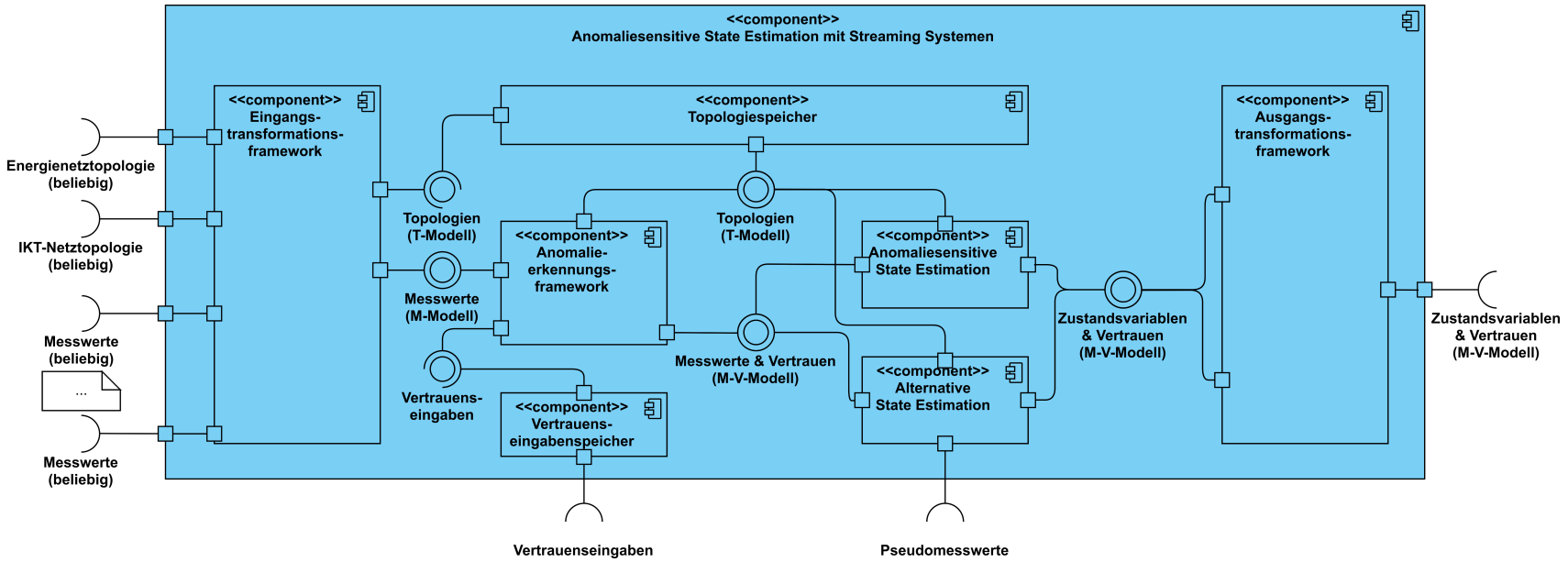


Abb. 6.2.: Die Architektur von ASSESS auf höchster Abstraktionsebene als UML-Komponentendiagramm.

Es ist ein Framework, da die Anzahl an eingehenden Datenströmen sowie die zur Übertragung verwendeten Protokolle von Anwendungsfall zu Anwendungsfall variieren können. Die Aufgabe des ITF ist es, die eingehenden Daten in das in Unterabschnitt 4.2.1 beschriebene T-Modell bzw. das im selben Unterabschnitt beschriebene M-Modell zu transformieren. Das ITF trägt damit zur Flexibilität und technischen Interoperabilität von ASSESS bei. Transformierte Topologien werden an einen Topologiespeicher weitergeleitet, dessen Aufgabe es ist, vorhandene Topologien in der jeweils aktuellen Version als Kontextinformation vorzuhalten. Zum einen wird die Energienetztopologie für die State Estimation benötigt, zum anderen können beliebige Topologien im ADF Verwendung finden.

An letzteres werden auch die transformierten Messwerte aus dem ITF weitergeleitet. Die Aufgabe des ADF ist es, durch die Integration verschiedener Anomaliedetektoren und auf Basis verschiedener externer Informationsquellen Anomalien im Trust in Messwerte zu finden und zu bewerten. Somit erfüllt das ADF die Aufgabe der Integrationsplattform für Trust-Schätzer und trägt durch seinen Framework-Charakter auch zur Flexibilität von ASSESS bei. Das ADF kann aus beliebig vielen Anomaliedetektoren bestehen, wobei jeder Anomaliedetektor Zugriff auf den Topologie- sowie einen Trust-Input-Speicher hat. Letzterer hält, analog zum Topologiespeicher, Trust-Inputs von potentiell beliebig vielen Trust-Quellen als Kontextinformationen vor. Ein Anomaliedetektor verwendet entsprechend die ihm zur Verfügung stehenden Informationen, um einfache Trust-Werte für einen Messwert zu erheben und sie Trust-Facetten zuzuordnen. Die Messwerte werden entsprechend mit dem in Abschnitt 3.2 beschriebenen Trust-Modell angereichert und im M-T-Modell weitergegeben. Eine Vereinigungskomponente überführt die einzelnen einfachen Trust-Werte in einen gemeinsamen multivariaten Trust-Wert pro Messwert.

Die nächste Komponente in dem Datenfluss ist eine anomaliesensitive State Estimation (ASSE). Das Ziel der ASSE ist es zum einen, eine State Estimation durchzuführen, d.h. die Zustandsvariablen auf Basis der Messwerte zu schätzen, und zum anderen eine Trust Estimation, d.h. den multivariaten Trust in die Zustandsvariablen auf Basis des multivariaten Trusts in die Messwerte zu schätzen. Dabei sind beide Schätzungen nicht zyklisch, sondern ereignisgetrieben durchzuführen. Dadurch erfüllt die ASSE die Aufgabe der Trust-sensitiven Lagebilderkennung und die nichtfunktionale Anforderung der Aktualität. Im Rahmen der vorliegenden Arbeit wurde der ASE [KL12; KML15] (vgl. Unterabschnitt 2.2.4) als State-Estimation-Komponente durch einen speziellen Operator umgesetzt. Generell abstrahiert die ASSE-Komponente von dem verwendeten State Estimator, stellt allerdings Anforderungen an die von diesem bereitgestellten Informationen. Neben den Zustandsvariablen und etwaigen zusätzlichen Informationen, wie z.B. die Modellabdeckung einer Zustandsvariable durch

die Messwerte oder die Unsicherheit einer Zustandsvariablen basierend auf den Standardabweichungen der Messwerte, ist es erforderlich, dass die State-Estimation-Komponente die in Abschnitt 5.3 beschriebenen Sensitivitäten bereitstellt. Darüber hinaus können noch weitere Informationen, wie zum Beispiel die Anzahl benötigter Iterationen, weitergegeben werden. Die Ausgabe der ASSE ist ein Datenstrom mit Zustandsvariablen angereichert mit Trust-Informationen im M-T-Modell.

Eine weitere, alternative State Estimation nutzt darüber hinaus alternative Messwerte (sogenannte Pseudomesswerte, z.B. historische), um ein alternatives, vertrauenswürdiges Lagebild zu erstellen, falls das anomaliesensitive ein Risiko für das System darstellt. Die alternative State Estimation dient daher der Prozessinteroperabilität und ist dabei im Wesentlichen wie die ASSE aufgebaut. Auf Grundlage des Trusts in die Originalmesswerte entscheidet eine Ersatzwertbildungskomponente darüber, welche Messwerte durch Pseudomesswerte ersetzt werden. Hier sei anzumerken, dass der Einsatz von Pseudomesswerten ebenfalls eine Trust-Einbuße darstellt. Dies wird in der Ersatzwertbildungskomponente berücksichtigt, indem die Aktualität bei historischen Messwerten oder die Unsicherheit bei simulierten Messwerten in eine reduzierte Glaubwürdigkeit überführt wird. Somit ist das Ergebnis einer alternativen State Estimation keineswegs zu einhundert Prozent vertrauenswürdig, aber im Zweifel vertrauenswürdiger als das Ergebnis der ASSE. Zwar wird die alternative State Estimation lediglich benötigt, falls das Ergebnis der ASSE nicht vertrauenswürdig genug ist (was zum Beispiel durch einen Schwellwert definiert werden kann), es bietet sich aber an, die alternative State Estimation immer parallel zur ASSE in einem Ensemble durchzuführen. Dadurch wird ein zeitlicher Verzug durch das Hintereinanderausführen der beiden State-Estimation-Prozesse vermieden. Der ausgehende Datenstrom der alternativen State Estimation folgt dabei dem gleichen Schema wie der der ASSE.

Die Ausgangsdatenströme beider State-Estimation-Komponenten werden vom Ausgangstransformationsframework (engl. output transf. framework) (OTF) verarbeitet. Das OTF hat dabei drei Funktionen. Erstens hat es die Aufgabe, aus dem Ensemble bestehend aus der ASSE und der alternativen State Estimation ein Ergebnis zu generieren. Dies kann z.B. durch ein Schwellwertverfahren geschehen; ist der Trust in den von der ASSE gelieferten Systemzustand unter einem bestimmten Wert, wird das Ergebnis der alternativen State Estimation verwendet. Zweitens sorgt es für adäquate Ausgabedatenraten. Aufgrund des ereignisgetriebenen Vorgehens kann es dazu kommen, dass alle paar Sekunden ein neues Lagebild zur Verfügung steht. Das OTF soll Lagebilder entsprechend nur bei Änderungen der Zustandsvariablen oder dem Trust in diese weiterleiten. Drittens hat das OTF analog zum ITF die Aufgabe, das Ergebnis in ein geeignetes Format zu überführen. Dies hängt wiederum von

dem Szenario und der Systemlandschaft ab, weshalb das OTF ein Framework ist. Somit trägt es ebenfalls zur Flexibilität, technischen und Prozessinteroperabilität von ASSESS bei.

6.3 Flexibilität und Skalierbarkeit

Flexibilität bezieht sich bei ASSESS auf die mögliche Unterstützung verschiedener Protokolle zur Datenübertragung (siehe Abschnitt 2.1.3) sowie die mögliche Verwendung sowohl diverser Trust-Schätzer und -eingaben (siehe Kapitel 4) als auch unterschiedlicher State-Estimation- und anderer Algorithmen, wie z.B. im OTF zur Bestimmung, welches State-Estimation-Ergebnis ASSESS ausgeben soll. Neben einer Flexibilität ist aber auch eine Skalierbarkeit vonnöten, da ASSESS mit unterschiedlichen Stromnetzgrößen und Anzahlen von Trust-Schätzern umgehen können muss. Die Herausforderung besteht demnach darin, entsprechende Systemkomponenten möglichst einfach anpassen, austauschen und/oder skalieren zu können.

In diesem Zusammenhang erlaubt es Odysseus, (Teil-) Anfragen, die sich in unterschiedlichen Skriptdateien befinden, zu verbinden [Ody]. Insbesondere eignet sich dafür der sogenannte Unteranfrageoperator (engl. subquery operator). Unteranfrageoperatoren sind spezielle Operatoren in Odysseus, die verschachtelte Anfragepläne ausblenden. Die ausgeblendeten Unteranfragen sind normale Anfragen. Sie werden also zusammen mit der Hauptanfrage installiert und gestartet. Die Vorteile von Unteranfragen sind zum einen eine verbesserte Übersicht durch mehrere Abstraktionsebenen und zum anderen eine bessere Austauschbarkeit. Letzteres wird dadurch erreicht, dass verschiedene Anfragen für dasselbe (Teil-) Problem definiert und einfach durch den Aufruf einer anderen Unteranfrage ausgetauscht werden können.

```
1  #PARSER PQL
2  #ADDQUERY
3  measurement = CONNECTOR({
4                    port = 0,
5                    source = '-> Measurement',
6                    schema = ${measurement_schema},
7                    metaattribute = ${std_meta_schema}
8                  }
9                )
10
```

```
measurement_with_trust = METADATA({METAATTRIBUTE = ${
    trust_schema}}, measurement)

#IF toInteger(num_ads) == toInteger(0)
    ad_result = measurement_with_trust
#ELSE
    ad_0 = measurement_with_trust
    #LOOP i 1 UPTO ${num_ads}
    ad_${i} = SUBQUERY({
                queryfile = '${ad_query_${i}}',
                schema = ${measurement_schema},
                metaattribute = ${trust_schema},
                name = '${ad_name_${i}}',
                options = ${ad_options_${i}}
            },
            ad_${i-1}
        )
    #ENDLOOP
    ad_result = ad_${num_ads}
#ENDIF

output = OUTPUTCONNECTOR({
              port = 0,
              name = 'Measurement ->'
            },
            ad_result
          )
```

Skript 6.3: Das Odysseus Script für das ADF.

Unteranfrageoperatoren stellen damit ein geeignetes Werkzeug dar, um die benötigte Flexibilität zu erreichen. Bezüglich der Skalierbarkeit bietet Odysseus Script Schleifenkonstrukte [Ody], mit deren Hilfe es möglich ist, eine Operation oder Unteranfrage zu skalieren. Die benötigte Anzahl an z.B. Datenquellen für Messwerte oder Trust-Schätzer kann dabei unter Verwendung von Variablen in Konfigurationsdateien ausgelagert werden.

Skript 6.3 zeigt die Verwendung von Unteranfragen in PQL und Schleifenkonstrukten in Odysseus Script anhand der Anfrage, die das ADF definiert, welche selbst eine Unteranfrage darstellt. Eine Unteranfrage benötigt Konnektoren für die Datenströme

aus der übergeordneten Anfrage. In diesem Fall gibt es einen eingehenden Datenstrom, den der Messwerte im M-Modell (Zeilen 3 bis 9). Dabei ist das Datenmodell in Odysseus generell aufgeteilt in ein Schema für den Payload (hier M-Modell; ausgelagert in die Variable `measurement_schema`) und ein Metaschema (hier das Standardmetaschema, das z.B. den Zeitstempel des entsprechenden Messwertes enthält; ausgelagert in die Variable `std_meta_schema`). Im nächsten Schritt wird das Metaschema um die multivariaten Trust-Werte erweitert (Zeile 11). Ab Zeile 13 kommen die Kontrollstrukturen von Odysseus Script [Ody] zum Einsatz. Zunächst wird die extern gesetzte Variable `num_ads` überprüft, über die die Anzahl an verwendeten Trust-Schätzern bzw. Anomaliedetektoren definiert wird. Kommt kein Trust-Schätzer zum Einsatz, entspricht das Ergebnis des ADF der Eingabe im M-T-Modell. Kommen allerdings Trust-Schätzer zum Einsatz, werden diese mithilfe einer Schleife und eines Unteranfrageoperators hintereinander ausgeführt. Dabei ist die Konfiguration der jeweiligen Unteranfrage wieder in Variablen ausgelagert, so dass das gesamte ADF über die Definition von Variablen konfiguriert werden kann. Hervorzuheben ist an dieser Stelle der Parameter `queryfile` des Unteranfrageoperators in Zeile 19. Durch diesen Parameter wird vorgegeben, welche Skriptdatei die Unteranfrage für den entsprechenden Trust-Schätzer definiert. Analog zu den Konnektoren zu Beginn in Skript 6.3 müssen für Unteranfragen, die Ergebnisdatenströme an die übergeordnete Anfrage liefern sollen, Konnektoren für diese Ergebnisdatenströme definiert werden. Dies ist in den Zeilen 31 bis 36 beschrieben. In diesem Fall wird das Ergebnis der Trust-Schätzung im M-T-Modell an die übergeordnete Anfrage weitergereicht.

Analog zu dem Beispiel in Skript 6.3 werden in ASSESS an allen für die Flexibilität und Skalierbarkeit relevanten Stellen Unteranfragen bzw. Schleifenkonstrukte verwendet. Unteranfragen werden z.B. im ITF bei der Übersetzung der Messwerte aus dem verwendeten Übertragungsprotokoll in das M-Modell oder in der ASSE für die State-Estimation-Komponente mit voriger Übersetzung der Messwerte aus dem M-T-Modell in ein Datenmodell spezifisch für den verwendeten State Estimator eingesetzt. Schleifenkonstrukte finden z.B. im ITF zur Skalierung der Datenquellen für Messwerte Verwendung.

6.4 Technische und Prozessinteroperabilität

Damit ASSESS auch perspektivisch von Stromnetzbetreibern eingesetzt werden kann, ist sowohl eine technische als auch eine Prozessinteroperabilität erforderlich. Bei der technischen Interoperabilität stehen vor allem in SCADA-Systemen verwendete

Kommunikations- und Datenprotokolle im Fokus, da der Prozess der Lagebilderkennung und auch jener der State Estimation fester Bestandteil von SCADA-Systemen sind. Für die prototypische Implementierung von ASSESS liegt dabei der Fokus auf dem 104er-Protokoll, da es in europäischen Energiesystemen zum Einsatz kommt. In Unterabschnitt 6.4.1 wird entsprechend eine offene Java-Implementierung des Standards vorgestellt, die im Rahmen dieser Arbeit vorgenommen wurde, um mit ASSESS Daten im 104er-Protokoll lesen und schreiben zu können.

Eine Herausforderung für die technische Interoperabilität ist es, eine gewisse Flexibilität gegenüber verwendeter Kommunikations- und Datenprotokolle zu gewährleisten. Diese Flexibilität ist allerdings durch das Access Framework von Odysseus (siehe Abschnitt 6.1) gegeben und wird durch die Erweiterung des Access Frameworks um die Fähigkeit, Nachrichten im 104er-Protokoll lesen und schreiben zu können, demonstriert.

Die Umsetzung der Prozessinteroperabilität umfasst zum einen die Herausforderung, ein möglichst vertrauenswürdiges Lagebild zu gewährleisten, um den Operateuren nicht nur ein Lagebild zur Verfügung zu stellen, das zwar Trust-sensitiv, aber im Zweifel nicht vertrauenswürdig ist. Zum anderen beinhaltet sie die Herausforderung, ein Lagebild nur dann an Operateure oder weiterverarbeitende Systeme weiterzuleiten, wenn es als wichtig eingestufte Änderungen am Systemzustand oder Trust gibt. Die Lösungen dieser Herausforderungen sind in den Unterabschnitten 6.4.2 und 6.4.3 beschrieben.

6.4.1 Open Java 104

Im Rahmen der vorliegenden Arbeit wurde eine Open-Source-Java-Bibliothek zum Senden, Empfangen und zur Weiterverarbeitung von Daten im 104er-Standard entwickelt, die Open Java 104 (OJ104) heißt[1].

Mit OJ104 ist es möglich, sowohl einen Server als auch einen Client zu implementieren. Wichtige Schnittstellen dabei sind die für APDU-, ASDU- sowie Kommunikationshandler. Ein APDU-Handler nimmt zum einen empfangene APDUs entgegen, führt das im 104er-Standard definierte Handshaking durch und kann empfangene ASDUs an einen ASDU-Handler weiterreichen. Zum anderen übernimmt der APDU-Handler noch die Aufgabe APDUs für zu sendende ASDUs zu erstellen und diese an einen Kommunikationshandler weiterzureichen.

[1] OJ104 ist frei verfügbar unter `https://git.swl.informatik.uni-oldenburg.de/projects/OJ`.

Die Interpretation von APDUs und ASDUs basiert auf Informationen aus XML-Dateien, wodurch eine größtmögliche Flexibilität und Erweiterbarkeit sichergestellt wird. Skript 6.4 zeigt einen Auszug aus der XML-Datei zur Interpretation von ASDUs. Für jeden im Standard definierten ASDU-Typen gibt es ein entsprechendes XML-Objekt mit der ID und Beschreibung aus dem Standard sowie Unterobjekten für die Kodierung, die enthaltenen Informations- und Zeitstempelklassen, ob Informationsobjekte in diesem ASDU-Typen als Sequenz oder einzeln übermittelt werden sowie die erlaubten Übertragungsgründe (engl. cause of transmission (COT)).

```
173   ...
174   <asdu id="34" description="measured value, normalized
          value with time tag CP56Time2a">
175     <code>M_ME_TD_1</code>
176   <ieclasses>de.uniol.inf.ei.oj104.model.
        informationelement.NormalizedValue de.uniol.inf.ei
        .oj104.model.informationelement.
        QualityDescriptorWithOV</ieclasses>
177   <ttclass>de.uniol.inf.ei.oj104.model.timetag.
        SevenOctetBinaryTime</ttclass>
178   <sq>0</sq>
179   <cot>3 5</cot>
180   </asdu>
181   ...
```

Skript 6.4: Ein Auszug aus der XML-Datei zur Interpretation von ASDUs.

Die interpretierten APDUs, APCIs und ASDUs werden in einer Java-Datenstruktur abgelegt, die JSON-serialisierbar ist. Skript 6.5 zeigt die JSON-Serialisierung einer beispielhaften ASDU. In den Zeilen 5-15 sind die typidentifizierenden Merkmale aus Skript 6.4 wiederzufinden. Die ASDU enthält ein Informationsobjekt, ein normalisierter Wert von 2036 mit einem Zeitstempel (Zeilen 21-48). Der ASDU-Handler ist die Schnittstelle, die in der Regel von der Applikation implementiert werden muss, die OJ104 einsetzt. Im ASDU-Handler wird bestimmt, was mit einer empfangenen ASDU geschehen soll. Kommunikationshandler werden von Server und Client implementiert und deren Aufgaben sind das Aufbauen einer Verbindung, das Senden und Empfangen von APDUs und das Schließen der Verbindung.

Im Rahmen der vorliegenden Arbeit wurde entsprechend das Access-Framework von Odysseus um einen Handler für Nachrichten im 104er-Standard erweitert. Der Handler nutzt dafür OJ104 und die JSON-Serialisierung der 104er-Elemente erweist sich insofern als gewinnbringend, als dass sie in Odysseus in Schlüssel-Wert-Paare

transformiert werden können, für die es ausreichend Funktionen zur Extraktion und Manipulation in Odysseus gibt.

```
{
  "dataUnitIdentifier": {
    "dataUnitType:" {
      "typeIdentification:" {
        "id": 34,
        "description": "measured value, normalized
           value with time tag CP56Time2a",
        "code": "M_ME_TD_1"
      },
      "structureQualifier": "SINGLE",
      "numberOfInformationElements": 1
    },
    "causeOfTransmission": {
      "id": 3,
      "description": "spontaneous"
    },
    "confirm": "POSITIVE",
    "test": false,
    "originatorAddress": 0,
    "commonAddressOfASDU": 20795
  },
  "informationObjects": [{
    "informationElements": [{
      "type": "NormalizedValue",
      "value": 2036
    },
    {
      "type": "QualityDescriptorWithOV",
      "blocked": false,
      "substituted": false,
      "notTopical": false,
      "invalid": false,
      "overflow": false
    }],
    "timeTag": {
      "type": "SevenOctetBinaryTime",
```

```
        "milliseconds": 230,
        "seconds": 7,
        "minutes": 17,
        "substituted": false,
        "invalid": false,
        "hours": 10,
        "daysOfMonth": 10,
        "daysOfWeek": 5,
        "months": 1,
        "years": 20
      },
      "informationObjectAddress": 39,
    }]
}
```

Skript 6.5: Eine beispielhafte, nicht standardisierte JSON-Serialisierung einer beispielhaften ASDU in OJ104.

6.4.2 Gewährleistung eines vertrauenswürdigen Lagebildes

Wie bereits in Abschnitt 6.2 beschrieben wurde, bildet eine parallele alternative State Estimation die Grundlage, um vertrauenswürdige Lagebilder zu gewährleisten.

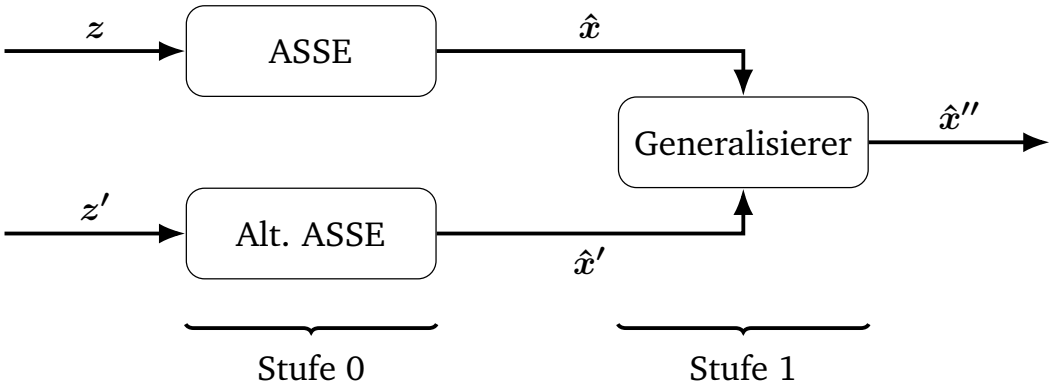


Abb. 6.3.: Zweistufiges „Stacked Generalization"-Ensemble nach [Wol92] bestehend aus der ASSE, der alternative State Estimation und einem Generalisierer. Die ASSE arbeitet auf den Messwerten z und die alternative State Estimation auf mit Pseudomesswerten veränderten Messwerten z', beide im M-T-Modell.

Die Kombination aus ASSE und alternativer State Estimation stellt ein spezielles zweistufiges „Stacked Generalization"-Ensemble [Wol92] dar, wie auch in Abbil-

dung 6.3 dargestellt. Die Stufe 0 bilden dabei die ASSE und die alternative State Estimation, bei denen es es sich um die gleichen Verfahren handelt (ASE und Trust-Schätzung), die allerdings auf zum Teil unterschiedlichen Messwertemengen $\boldsymbol{z}$ und $\boldsymbol{z'}$ arbeiten. Dabei stellt $\boldsymbol{z}$ die Grundlage für $\boldsymbol{z'}$ dar, allerdings werden besonders unvertrauenswürdige Messwerte durch Pseudomesswerte ersetzt. Sowohl $\boldsymbol{z}$ als auch $\boldsymbol{z'}$ liegen im M-T-Modell vor. Entsprechend weicht im Zweifel das Ergebnis $\hat{\boldsymbol{x}}'$ der alternativen State Estimation vom Ergebnis $\hat{\boldsymbol{x}}$ der ASSE ab. Beide Ergebnisse liegen ebenfalls im M-T-Modell vor und sind die Eingaben für einen Stufe-1-Generalisierer, der die Aufgabe hat, auf Basis von $\hat{\boldsymbol{x}}$ und $\hat{\boldsymbol{x}}'$ ein Ergebnis $\hat{\boldsymbol{x}}''$ zu erzeugen. Theoretisch können in ASSESS auch mehr als zwei Stufe-0-Modelle zum Einsatz kommen. Es ist zum Beispiel denkbar, dass mehrere alternative State Estimatoren parallel zum Einsatz kommen, bei denen jeweils unterschiedliche Anzahlen an Messwerten durch Pseudomesswerte ersetzt werden.

Da die alternative State Estimation Messwerte mit besonders niedrigem Trust nicht berücksichtigen soll, ist eine Herausforderung, solche Messwerte zu identifizieren. Die Messwerte liegen im M-T-Modell, d.h. angereichert mit multivariaten Trust-Werten, vor. Zwei multivariate Trust-Werte zu vergleichen, um zunächst festzustellen, welcher Messwert aus einer Menge von zwei Messwerten vertrauenswürdiger ist als der andere, ist nicht trivial. Es stellt sich die Frage, wie zwei multivariate Trust-Werte mit unterschiedlichen Trust-Wahrscheinlichkeiten in den einfachen Trust-Werten verglichen werden können.

$$\boldsymbol{T}_{e_1} = (\{(\gamma_1, 0, 5), (\gamma_2, 0, 8)\}, \emptyset, \{(\gamma_3, 0, 65)\}, \emptyset, \emptyset, \emptyset) \tag{6.1}$$

$$\boldsymbol{T}_{e_2} = (\{(\gamma_1, 0, 8), (\gamma_2, 0, 5)\}, \emptyset, \{(\gamma_3, 0, 65)\}, \emptyset, \emptyset, \emptyset) \tag{6.2}$$

$$\boldsymbol{T}_{e_3} = (\{(\gamma_1, 0, 65), (\gamma_2, 0, 8)\}, \emptyset, \{(\gamma_3, 0, 5)\}, \emptyset, \emptyset, \emptyset) \tag{6.3}$$

Seien, zum Beispiel, drei Entitäten e_1, e_2 und e_3 mit multivariaten Trust-Werten $\boldsymbol{T}_{e_1}, \boldsymbol{T}_{e_2}$ und $\boldsymbol{T}_{e_3}$ gegeben wie in Gleichungen 6.1 - 6.3. Alle multivariaten Trust-Werte bestehen aus drei einfachen Trust-Werten von den gleichen Trust-Schätzern, davon zwei in der Facette „Funktionale Korrektheit“ und einer in der Facette „Informationssicherheit“. Die einfachen Trust-Werte unterscheiden sich aber in den Trust-Wahrscheinlichkeiten. Um nun zu bestimmen, welche Entität vertrauenswürdiger ist als eine andere, muss bestimmt werden, ob die Trust-Wahrscheinlichkeiten unterschiedlicher Trust-Schätzer gleichwertig sind oder nicht. Dies gilt sowohl innerhalb einer Facette (vgl. Gleichungen 6.1 und 6.2) als auch zwischen unterschiedlichen Facetten (vgl. Gleichungen 6.1 und 6.3). Es ist somit eine Interpretation

der multivariaten Trust-Werte nötig, auch um festzustellen, welche Trust-Werte als besonders niedrig einzustufen sind. Eine solche Interpretation ist allerdings nicht Bestandteil dieser Arbeit. Vielmehr wurde für die prototypische Implementierung ein einfaches Schwellwertverfahren mit dem vorigen Bilden des Minimums aller einfachen Trust-Werte umgesetzt. Es ist allerdings anzumerken, dass die Identifikation besonders unvertrauenswürdiger Messwerte in eine austauschbare Unteranfrage (vgl. Abschnitt 6.3) ausgelagert wurde, um den Einsatz komplexerer Algorithmen zu ermöglichen. Die durch den Algorithmus identifizierten Messwerte werden entsprechend für die alternative State Estimation mit historischen vertrauenswürdigen Messwerten der gleichen Messpunkte ausgetauscht.

Generell kann bei einem „Stacked Generalization“-Ensemble ein beliebiges Modell als Stufe-1-Generalisierer verwendet werden. Entsprechend kann auch in ASSESS ein beliebiger Algorithmus für diese Ausgabe im OTF eingesetzt werden. Zu bedenken ist aber die Rahmenbedingung, dass das Ergebnis $\hat{\boldsymbol{x}}''$ in sich kohärent sein muss, d.h. die einzelnen Zustandsvariablen in $\hat{\boldsymbol{x}}''$ müssen zusammen ein physikalisch plausibles Gesamtbild ergeben. Aus diesem Grund wurde für den Prototypen von ASSESS ein Verfahren als Generalisierer eingesetzt, das zwischen $\hat{\boldsymbol{x}}$ und $\hat{\boldsymbol{x}}'$ wählt. Da es wieder auf einen Vergleich multivariater Trust-Werte ankommt, um ein Ergebnis auszuwählen, wurde wie bei der Ersatzwertbildung zuvor verfahren. Der zu verwendende Algorithmus wurde in eine austauschbare Unteranfrage ausgelagert und für die prototypische Implementierung mit der Bildung des Minimums des Trusts in das Ergebnis der ASSE und einem einfachen Schwellwertverfahren umgesetzt. Somit erlaubt es ASSESS, als besonders unvertrauenswürdig eingestufte Lagebilder durch alternative, im Zweifel vertrauenswürdigere zu ersetzen.

6.4.3 Als wichtig eingestufte Lagebildänderungen

Um Operateure nicht mit hochfrequenten Lagebildern zu überfordern, ist es wichtig, nur dann Lagebilder weiterzuleiten, wenn es Änderungen am Systemzustand oder Trust gibt, die als relevant eingestuft werden. Analog zur Gewährleistung eines vertrauenswürdigen Lagebildes in Unterabschnitt 6.4.2 sind dafür Interpretationen vonnöten, bezüglich zum einen Lagebild- und zum anderen Trust-Änderungen. Wie bereits erwähnt, sind derartige Interpretationen nicht Teil dieser Arbeit. Aus diesem Grund wurde erneut durch die Verwendung einer Unteranfrage ein Rahmenwerk geschaffen, um Algorithmen zur Erkennung von relevanten Zustands- oder Trust-Änderungen einbinden zu können. Für die prototypische Implementierung von ASSESS wurden einfache Schwellwertverfahren umgesetzt, mit denen es möglich

ist, Lagebilder ohne signifikante Änderung herauszufiltern und entsprechend nicht an Operateure oder weiterverarbeitende Systeme weiterzuleiten.

6.5 Aktualität

Die Anforderung der Aktualität bezieht sich sowohl auf eine sofortige Berücksichtigung möglicherweise kompromittierender Ereignisse als auch und damit verbunden auf eine ereignisgetriebene State Estimation anstelle einer zyklisch durchgeführten (vgl. Abschnitt 1.2).

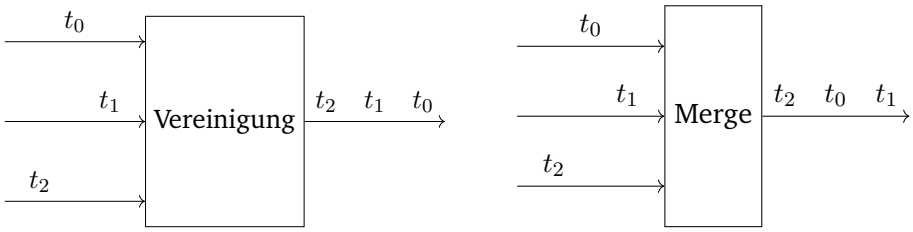
Bezogen auf die Aktualität gibt es auch in DSMSs Herausforderungen, von denen drei für diese Arbeit besonders relevant sind. Diese Herausforderungen und der Umgang mit ihnen werden in diesem Abschnitt behandelt: blockierende Operationen in Unterabschnitt 6.5.1, das effiziente Bilden von Messwertemengen für die State Estimation in Unterabschnitt 6.5.2 und die Einhaltung der Zeitsemantik bei Berücksichtigung von Kontextinformationen in Unterabschnitt 6.5.3.

6.5.1 Blockierungen

Die Idee einer Datenstromverarbeitung ist es, Elemente möglichst „on-the-fly“ ohne Blockierungen zu verarbeiten. Einige Datenstromoperationen sind allerdings blockierend, meistens um die zeitliche Ordnung in einem Datenstrom entweder herzustellen oder zu bewahren. Eine Verarbeitung nach diesem Prinzip wird In-Order-Verarbeitung genannt und entsprechend wird eine Verarbeitung nicht zwangsläufig zeitlich geordneter Datenströme Out-of-Order-Verarbeitung genannt. Ferner sind einige Fensteransätze, wie z.B. Element- und Sessionfenster, blockierend (vgl. Unterabschnitt 2.4.2). In ASSESS kann das Problem einer Blockierung an drei Stellen auftreten, wobei zwei sowie deren Auflösung im Folgenden und die dritte im anschließenden Unterabschnitt beschrieben werden.

ASSESS soll dazu in der Lage sein, Messwerte direkt von der Fernwirktechnik, z.B. von RTUs, entgegenzunehmen. Entsprechend gibt es in der Regel mehr als eine Datenquelle, z.B. eine pro RTU, und die unterschiedlichen Datenströme sollten, um die Komplexität zu begrenzen, möglichst frühzeitig zusammengeführt werden. Für eine In-Order-Verarbeitung eignet sich für das Zusammenführen ein Vereinigungsoperator (im Folgenden `Vereinigung` genannt), der allerdings blockierend ist. Um die zeitliche Ordnung des Ausgabedatenstroms zu gewährleisten, prüft die `Vereinigung` beim Eintreffen eines neuen Elementes in einem seiner Eingangsdatenströme, ob in

den anderen Eingangsdatenströmen noch Elemente eintreffen können, die älter sind. Erst wenn, unter der Annahme, dass die einzelnen Eingangsdatenströme in sich zeitlich geordnet sind, die genannte Bedingung erfüllt ist, kann das Element in den Ausgabedatenstrom aufgenommen werden. Die Blockierungsdauer hängt demnach von der niedrigsten Datenrate der Eingangsdatenströme ab, wird problematisch bei unerwartet fehlenden Messwerten und kann sogar unendlich sein, wenn von einer Datenquelle keine Elemente mehr empfangen werden. Letzterem kann in Odysseus durch sogenannte `Heartbeats` entgegengewirkt werden. `Heartbeats` sind Datenstromelemente ohne Nutzlast. Sie dienen lediglich der Anzeige von zeitlichem Fortschritt in einem Datenstrom und können im DSMS selbst generiert werden. Mit einem periodischen Erstellen von `Heartbeats` vor einer `Vereinigung` kann somit eine unendliche Blockierung vermieden werden. Allerdings ist das effiziente Konfigurieren von `Heartbeats`, d.h. wann in Systemzeit sollen sie mit welchem Zeitstempel (Applikationszeit) eingefügt werden, nicht einfach. Werden zu häufig `Heartbeats` erstellt, kann es passieren, dass spätere Datenstromelemente nicht mehr in zeitlicher Reihenfolge sind, da der erstellte `Heartbeat` einen jüngeren Zeitstempel trägt. Werden zu selten `Heartbeats` erstellt, blockieren sie die `Vereinigung`.



(a) Wahrung der Zeitsemantik durch eine `Vereinigung`.

(b) Ignorieren der Zeitsemantik durch einen `Merge`.

Abb. 6.4.: Die Unterschiede im Umgang mit der Zeitsemantik bei einer Vereinigung und einem Merge. Datenströme und deren Richtung werden durch Pfeile symbolisiert. t_i mit $i \in \{0, 1, 2\}$ ist der Zeitstempel eines Datenstromelementes, wobei für ein t_i und ein t_j mit $i < j$ gilt, dass t_i älter ist als t_j. Die Positionierung von Zeitstempeln auf einem Datenstrom stellt die systemzeitliche Ordnung der Datenstromelemente dar.

Um dieses Problem in ASSESS zu lösen, wird zunächst analysiert, an welchen Stellen eine In-Order-Verarbeitung erforderlich ist. Das ADF behandelt jeden Messwert individuell. Für die State Estimation in der ASSE muss, wie im folgenden Unterabschnitt erläutert, eine Messwertemenge definiert werden (vgl. Unterabschnitt 2.2.2). Diese Messwertemenge muss allerdings in sich nicht zeitlich geordnet sein. In conclusio kann in ASSESS auf eine In-Order-Verarbeitung der Messwerte verzichtet werden. Dies ermöglicht, statt einer `Vereinigung` einen sogenannten Merge-Operator (im Folgenden `Merge` genannt) zu verwenden, der nicht die zeitliche Ordnung garantiert, sondern Elemente aus verschiedenen Eingangsdatenströmen direkt in den

Ausgangsdatenstrom überführt. Somit kann eine Blockierung durch eine In-Order-Verarbeitung vermieden werden. Die Unterschiede im Umgang mit der Zeitsemantik bei einer `Vereinigung` und einem `Merge` sind ebenfalls in Abbildung 6.4 dargestellt.

Eine zweite mögliche Blockierung kann im ADF auftreten. Dort soll es möglich sein, beliebig viele Trust-Schätzer zu integrieren. Da die Trust-Schätzer von einander unabhängig arbeiten, bietet sich eine Parallelisierung an. Eine solche Parallelisierung kann in Odysseus zu unterschiedlichen Verarbeitungsreihenfolgen führen, abhängig vom verwendeten Prozessmanagement (engl. thread management). Dabei ist zu beachten, dass jede Datenquelle in ASSESS einen eigenen Prozess verwendet. Wird ein Datenstrom mit nur einer Datenquelle im Laufe der Verarbeitung parallelisiert, werden die einzelnen parallelen Verarbeitungszweige nacheinander abgearbeitet, wodurch es einer seriellen Verarbeitung entspricht. Vorteilhaft ist dabei, dass bei der Zusammenführung der parallelen Datenströme, z.B. durch einen `Merge`, die zeitliche Ordnung gegeben ist, da keine Teilergebnisse von unterschiedlichen Eingangselementen zeitlich durchmischt bei der Zusammenführung vorliegen können. Um eine nicht nur theoretische, sondern auch praktische parallele Verarbeitung umzusetzen, können in Odysseus zu Beginn der einzelnen parallelen Verarbeitungszweige sogenannte Threaded-Buffer-Operationen (im Folgenden `ThreadedBuffer` genannt) eingebaut werden. Neben der Funktionalität, Datenstromelemente zwischenspeichern zu können, nutzen `ThreadedBuffer` einen neuen Prozess für ihre Ausgaben. Dadurch ist allerdings bei der Zusammenführung der parallelen Datenströme die zeitliche Ordnung nicht gegeben und es muss auf eine `Vereinigung` mit den oben genannten Problemen zurückgegriffen werden.

In ASSESS könnten bei einer parallelen Verarbeitung die einzelnen Trust-Schätzer die Messwerte mit ihren einfachen Trust-Werten anreichern (M-T-Modell). Die somit entstehenden mehreren Teilergebnisse pro Messwert müssten bei einer Zusammenführung wieder zu einem Datenstromelement im M-T-Modell zusammengeführt werden. Dies kann durch eine `Vereinigung` mit einer anschließenden `Aggregation` oder durch eine `Verknüpfung` (engl. join) mit den bereits erläuterten Nachteilen geschehen. Aufgrund dessen und der Tatsache, dass die im Rahmen der prototypischen Umsetzung verwendeten Trust-Schätzer äußerst schnell arbeiten, wird in der prototypischen Umsetzung von ASSESS auf eine serielle Verarbeitung im ADF zurückgegriffen. Somit kann eine Blockierung auch im ADF vermieden werden. Dieses Vorgehen sollte allerdings überdacht werden, falls mehrere verwendete Trust-Schätzer deutlich mehr Zeit für ihre Verarbeitung benötigen. In einem solchen Fall kann im Zweifel auf eine parallele Verarbeitung zurückgegriffen werden.

6.5.2 State-Estimation-Messwertemengen

Klassischerweise wird eine State Estimation entweder manuell angestoßen oder in festen Zyklen durchgeführt. Aufgrund der Annahme, dass sich in zukünftigen Stromsystemen das Lagebild wesentlich dynamischer ändern wird, und unter Berücksichtigung der Prämisse, dass Ereignisse im System, die zu Trust-Einbußen führen, direkt verarbeitet werden sollten, wird im Rahmen dieser Arbeit der Ansatz einer ereignisgetriebenen State Estimation verfolgt. Im Folgenden wird der Fensteransatz näher betrachtet, der bei der ereignisgetriebenen State Estimation bestimmt, welche Messwerte zusammen die Eingabe der State-Estimation-Komponente bilden, und der eine dritte mögliche Blockierung darstellt.

Zunächst wird die Tauglichkeit der einzelnen in Unterabschnitt 2.4.2 beschriebenen Fensteransätze, Zeit-, Element-, Session- und Prädikatfenster, für eine ereignisgetriebene State Estimation erörtert. Zeitfenster eignen sich grundsätzlich; konkret taumelnde Zeitfenster mit einem Fortschritt, der der Fenstergröße entspricht, so dass jedes Element in genau einem Fenster verarbeitet wird. Ein Problem tritt aber dann auf, wenn es zu großen Zeitspannen ohne Elementen kommt, da der Zeitfortschritt anhand der Elementzeitstempel gemessen wird. Für solche Fälle haben sich jedoch `Heartbeats` etabliert (vgl. Unterabschnitt 6.5.1). Ein weiterer Nachteil des Zeitfensteransatzes für eine ereignisgetriebene State Estimation liegt aber in der Fixierung auf Zeitstempel. Wenn der Datenstrom nicht gleichmäßig fließt, die Datenquellen z.B. unterschiedliche Datenraten aufweisen oder Messwerte teilweise spontan übertragen werden, kann eine adäquate Fenstergröße nur schwierig bestimmt werden. Sie müsste so gewählt werden, dass in den meisten Fenstern Messwerte möglichst aller Messpunkte idealerweise einmal vorhanden sind. Wählt man die Fenstergröße zu klein, fehlen Messwerte in den Fenstern und müssen durch Pseudomesswerte (künstliche oder historische Messwerte) ersetzt werden. Wählt man die Fenstergröße hingegen zu groß, wird die State Estimation später als nötig ausgeführt und wichtige Informationen gelangen im Zweifel zu spät zum Operateur.

Betrachtet man Elementfenster, so eignen sich diese nur, wenn alle Datenquellen vergleichbare Datenraten haben und ihre Messwerte vergleichbare Zeitstempel aufweisen. Gibt es aber Datenquellen, die spontan oder häufiger senden als andere, ist keine adäquate Fenstergröße auf Basis einer Elementanzahl zu wählen. Die gleiche Argumentation gilt auch für die Sessionfenster. Diese sind nur dann sinnvoll, wenn die Messwerte aller Messpunkte zu vergleichbaren Zeiten gesendet werden. Da das Sessionfenster mit der Systemzeit arbeitet, kommt erschwerend dazu, dass die RTUs i.d.R. nicht zeitsynchronisiert sind. Als letzter Ansatz sei jener der Prädikatfenster betrachtet. Dieser bietet zwar eine hohe Flexibilität bei der Definition von Fenstern,

allerdings besteht hier das Problem, geeignete Bedingungen für das Schließen eines Fensters zu definieren. Daher seien, bevor der Einsatz von Prädikatfenstern bei einer ereignisgetriebenen State Estimation erörtert wird, zunächst sinnvolle Bedingungen für das Schließen von Fenstern diskutiert.

Die State Estimation sollte so zeitnah wie möglich durchgeführt werden. Allerdings ist es auch nicht sinnvoll, häufiger Elemente an die State-Estimation-Komponente weiterzuleiten, als diese sie verarbeiten kann. Sei Δt_{se} die Zeitspanne, die der State Estimator im Durchschnitt benötigt. Δt_{se} ist dabei im Wesentlichen von dem Systemmodell und der Anzahl an Messwerten abhängig. Es gibt also keinen Mehrwert, wenn die Systemzeitspanne zwischen dem Schließen zweier aufeinander folgender Fenster, als Δt_{win} bezeichnet, kleiner als Δt_{se} ist. Wählt man nun $\Delta t_{win} = \Delta t_{se}$, kann es aber sein, dass, je nach Datenquellen, nur Messwerte von wenigen Messpunkten im Fenster vorhanden sind. Daher empfiehlt sich, eine zusätzliche Bedingung zu wählen, und zwar, von wie vielen Messpunkten, in Relation zur Gesamtanzahl an Messpunkten, Messwerte in dem Fenster vorhanden sein sollen. Sollte diese kombinierte Bedingung nicht erfüllt werden, benötigt es noch eine weitere, harte Schließbedingung. Hier bietet es sich an, eine maximale Systemzeit Δt_{max} zu definieren, für die das Fenster offen sein sollte. Diese kann sich z.B. an den Datenraten der Datenquellen orientieren.

$$\Delta t_{win} \geq \Delta t_{max} \vee (\Delta t_{win} \geq \Delta t_{se} \wedge |\boldsymbol{mp}| >= p \cdot m) \tag{6.4}$$

Zusammenfassend sieht die Bedingung zum Schließen eines Fensters bei einer ereignisgetriebenen State Estimation wie in Gleichung 6.4 dargestellt aus. Dabei ist $|\boldsymbol{mp}|$ die aktuelle Anzahl an unterschiedlichen Messpunkten mit Elementen in dem Fenster. $p \in [0; 1]$ gibt an, wie viel Prozent der Gesamtanzahl an Messpunkten m mindestens im Fenster sein sollen, damit dieses bereits zum Zeitpunkt Δt_{se} geschlossen wird. An dieser Stelle sei angemerkt, dass Gleichung 6.4 keine Unterscheidung zwischen redundanten und kritischen Messwerten macht. Eine Bedingung, die den Unterschied in der Kritikalität berücksichtigt, ist sicherlich eine sinnvolle Weiterentwicklung, die allerdings nicht im Rahmen der vorliegenden Arbeit vorgenommen wird, sondern ein Aspekt für zukünftige Arbeiten darstellt.

Eine solche Bedingung lässt sich leider nicht mit einem herkömmlichen Prädikatfenster umsetzen, da bei solchen nur auf Elementattribute zugegriffen werden kann. Aus diesem Grund wird in dieser Arbeit die Umsetzung eines speziellen Fensters für die ereignisgetriebene State Estimation verfolgt. Skript 6.6 zeigt das Vorgehen für dieses spezielle Fenster.

```
1  z ← ∅
2  mp ← ∅
3  hist ← ∅
4  timer_1 ← null
5  timer_2 ← null
6  ---------------------------------------------------------------
7  if new element e
8      if |z| = 0
9          start timer_1
10         start timer_2
11     if mp_e ∈ mp
12         z ← z − {z_e}
13     else
14         mp ← mp + {mp_e}
15     z ← z + z_e
16     if |mp| = m
17         stop timer_1
18         stop timer_2
19         output *($\bm{z}$*)
20         hist ← z
21         z ← ∅
22         mp ← ∅
23 ---------------------------------------------------------------
24 if evaluate(timer_1) ∧ |mp| ≥ p
25     stop timer_2
26     ∀z ∈ {hist|z ∉ z} : z ← z + {z}
27     hist ← z
28     z ← ∅
29     mp ← ∅
30 ---------------------------------------------------------------
31 if evaluate(timer_2)
32     ∀z ∈ {hist|z ∉ z} : z ← z + {z}
33     hist ← z
34     z ← ∅
35     mp ← ∅
```

Skript 6.6: Der in ASSESS umgesetzte Fensteransatz zur Erstellung von Messwertemengen als Pseudocode. $\boldsymbol{z}$ und $\boldsymbol{mp}$ sind die Mengen an Messwerten bzw. unterschiedlichen Messpunkten im aktuellen Fenster, $\boldsymbol{hist}$ die Menge der jüngsten bekannten Messwerte eines jeden Messpunktes (Historie), $timer_1$ und $timer_2$ zwei Systemzeitschaltuhren und e ein Datenstromelement mit mp_e als Messpunkt und z_e als Messwert. p ist die vorgegebene minimale Anzahl an verschiedenen Messpunkten im Fenster und n die Gesamtanzahl an Messpunkten.

Zunächst werden die benötigten Mengen ($\boldsymbol{z}$ und $\boldsymbol{mp}$ sind die Mengen an Messwerten bzw. unterschiedlichen Messpunkten im aktuellen Fenster; $\boldsymbol{hist}$ die Menge

der jüngsten bekannten Messwerte eines jeden Messpunktes (Historie)) und zwei Systemzeitschaltuhren für Δt_{se} und Δt_{max} initiiert.

Durch den Einsatz der Systemzeitschaltuhren kommt es zu drei unterschiedlichen Verarbeitungspfaden mit unterschiedlichen Auslösern. In den Zeilen 7 bis 22 findet die Verarbeitung eines neuen Datenstromelementes e statt. Eröffnet e ein neues Fenster, so werden beide Systemzeitschaltuhren gestartet. Ist hingegen bereits ein Messwert vom gleichen Messpunkt im aktuellen Fenster, so wird dieser entfernt. Andernfalls wird der Messpunkt in $\boldsymbol{z}$ aufgenommen. Im Anschluss an diese Überprüfungen wird der Messwert $\boldsymbol{mp}$ hinzugefügt. Sollte bereits die maximale Anzahl an Messpunkten im System erreicht sein, wird $\boldsymbol{z}$ als Messwertemenge ausgegeben. Außerdem werden die Systemzeitschaltuhren gestoppt, die Historie aktualisiert und der Fensterzustand ($\boldsymbol{mp}$ und $\boldsymbol{z}$) zurückgesetzt.

Die Verarbeitung beim Ablauf einer Systemzeitschaltuhr ist in den Zeilen 24 bis 29 bzw. 31 bis 35 beschrieben. In beiden Fällen wird die Messwertemenge $\boldsymbol{z}$ mithilfe der Historie aufgefüllt, so dass nach Möglichkeit von jedem Messpunkt ein Messwert enthalten ist (Einsatz von Pseudomesswerten). Außerdem wird die Historie aktualisiert und der Fensterzustand zurückgesetzt. Handelt es sich bei der abgelaufenen Systemzeitschaltuhr um die für Δt_{se}, wird zusätzlich die andere noch laufende Systemzeitschaltuhr gestoppt.

Ein weiterer Aspekt, der bei der Erstellung bzw. Bereitstellung der Messwertemengen beachtet werden muss, ist die Dauer einer State Estimation. Es wurde bereits erwähnt, dass eine State Estimation im Zweifel schneller sein kann, als eine Messwertemenge bereitgestellt wird (siehe Gleichung 6.4). Analog kann es, gerade für komplexe Stromsysteme, vorkommen, dass die State Estimation deutlich mehr Zeit benötigt. Ohne besondere Handhabung würde die Verarbeitung in dem aktuellen Systemprozess in dieser Zeit blockiert werden. Um dies zu verhindern, wird ein bereits erwähnter `ThreadedBuffer` verwendet, der diese Blockierung auflöst. Zudem ist er derart konfiguriert, dass er nur eine Messwertemenge zwischenspeichert, wobei eine neu eintreffende Messwertemenge eine ggf. zwischengespeicherte ersetzt. Dadurch werden zwar Messwertemengen verworfen, aber eine Aktualität der Zustandsvariablen gewährleistet. Es wird immer die jüngste verfügbare Messwertemenge für eine State Estimation berücksichtigt.

6.5.3 Kontextinformationen & Zeitsemantik

Bei der Anreicherung von Messwerten mit Kontextinformationen ist die Zeitsemantik zu beachten, d.h. Messwerte sollten nur mit Kontextinformationen angereichert

werden, wenn beide zum gleichen Zeitpunkt gültig sind. Erschwerend ist, dass für weder Messwerte noch Kontextinformationen rechtzeitig definiert werden kann, wie lange sie gültig sind. Theoretisch endet die Gültigkeit bei einem neuen Messwert vom gleichen Messpunkt oder einer neuen Kontextinformation desselben Typs (z.B. eine Topologieveränderung). Für solche Szenarien gibt es in Odysseus einen sogenannten `ContextStore`, der einen begrenzten, partitionierbaren Zwischenspeicher für Datenstromelemente im Hauptspeicher darstellt. Die Besonderheit dieses Zwischenspeichers ist, dass beim Speichern eines neuen Elementes das zuletzt gespeicherte Element derart aktualisiert wird, dass das Ende seiner Gültigkeit auf den Zeitstempel des neues Elementes gesetzt wird. Dadurch ergibt sich insgesamt ein nach oben offenes Gültigkeitsintervall ohne Lücken im `ContextStore`, in dem zu jedem Zeitpunkt genau eine Kontextinformation gültig ist [Ody].

6.6 Demonstration

Anmerkung: Die im Folgenden beschriebene Demonstration ist eine gemeinsame Arbeit mit Kollegen:innen im Forschungsprojekt Smart Grid Cyber-Resilienz Labor[2] und ist ebenfalls in [Bra+21] veröffentlicht.

Im Rahmen des Forschungsprojektes „Smart Grid Cyber-Resilienz Labor“[2] wurde eine Plattform entwickelt, um die Funktionalität und den Mehrwert von ASSESS zu demonstrieren. Es soll gezeigt werden, dass der Trust in Messwerte eine wichtige Rolle bei der Verhinderung unerwünschter (manueller oder autonomer) Reaktionen des Stromsystems spielen kann. Die Demonstration besteht insbesondere aus einem FDIA, bei dem die Kontrolle über RTUs durch einen Eindringling erlangt wird. Der Angreifer verschafft sich Zugang zu den RTUs, indem er mehrstufige Exploits auf einem verwundbaren FTP-Server einsetzt. Nachdem er sich Zugang verschafft hat, ist der Eindringling in der Lage, die von einem Stromnetzsimulator empfangenen Messwerte abzufangen und durch manipulierte Messwerte auszutauschen. Ziel des koordinierten FDIA ist es, mit den Manipulationen nicht von einer Bad Data Detection erkannt zu werden (siehe Abschnitt 2.2.3), eine Unterspannungssituation vorzutäuschen und einen Stufenwechsel durch den Stufenregler eines Transformators auszulösen. Aufgrund der manipulierten Messwerte führt dieser Stufenwechsel dann in Wirklichkeit zu einer Überspannungssituation. Durch die Verwendung von ASSESS und der Berücksichtigung des Trusts in das Lagebild kann der Operateur den Stufenwechsel vermeiden und das System stabil halten. Das Trust-sensitive

[2] Smart Grid Cyber-Resilienz Labor: `https://www.offis.de/offis/projekt/cybreslab.html`

Lagebild speist sich dabei aus in einem IT-Monitoringtool und einem IDS festgestellten Anomalien. So geben beispielsweise Warnungen des IDS Hinweise auf einen illegalen Zugriff auf die RTUs [Bra+21]. Die Demonstration ist ebenfalls als Video veröffentlicht[3]. Im Folgenden wird die Plattform detaillierter beschrieben.

Zum Zwecke der Demonstration wurde ein Aufbau konzipiert, bestehend aus der Co-Simulation eines CPES, der Überwachung von RTUs sowie der Speicherung aller Messwerte aus dem CPES und Überwachungsdaten in einer Big-Data-Plattform. Der Trust in die Messwerte und Zustandsvariablen wird durch ASSESS aus diesen Informationen berechnet und beim Betrieb des Systems berücksichtigt [Bra+21]. Der Aufbau ist eine Erweiterung einer früheren Veröffentlichung über das theoretische Konzept [Bra+19a].

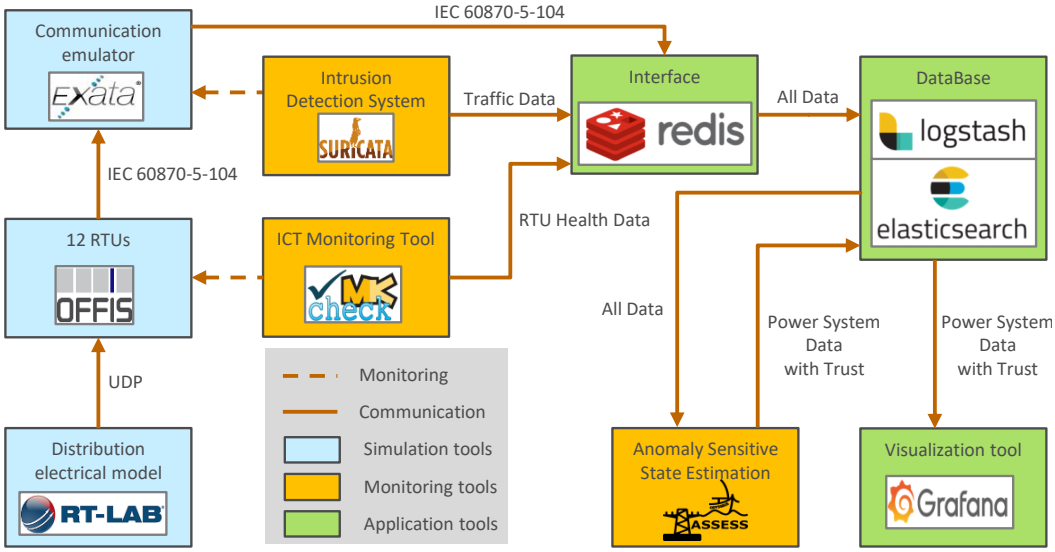


Abb. 6.5.: Echtzeit-Co-Simulationsplattform zur Trust-Erhebung in einem CPES [Bra+21].

Die Komponenten der Plattform können in drei Kategorien eingeteilt werden, namentlich in Simulations-, Überwachungs- und Anwendungswerkzeuge, wie in Abbildung 6.5 dargestellt. Das simulierte Stromsystem ist eine modifizierte Version des CIGRE-Mittelspannungsnetzes[4] mit 12 statt 15 Sammelschienen. Ein Transformator ist an ein Wohngebiet mit einer hohen installierten Kapazität an Photovoltaik- und Windkraftanlagen angeschlossen. Die Echtzeitsimulation des Stromnetzes ist durch virtuelle RTUs [Ans+19] (eine pro Sammelschiene) mit der Simulation des IKT-Systems verbunden. Das simulierte IKT-System besteht aus einem vermaschten Kernnetz mit drahtgebundenen Verbindungen zwischen allen Komponenten

[3] Video zur Demonstration: `https://youtu.be/3hwi49sfllQ`
[4] CIGRE-Mittelspannungsnetz: https://pandapower.readthedocs.io/en/v1.2.2/networks/cigre.html

und einem Edge-Router pro virtueller RTU. Das Verhalten der RTUs (zum Beispiel CPU- und Speicherauslastung) wird durch ein IT-Monitoringtool überwacht. Darüber hinaus untersucht ein IDS aktiv das Netzwerk, um bösartiges Verhalten wie Angriffe auf das Netzwerk oder schädliche Nutzdaten zu erkennen. Eine zentrale Datenplattform dient der Speicherung, Anreicherung und Analyse der Daten, die von ASSESS zur Erstellung eines Trust-sensitiven Lagebildes auf Grundlage der Daten des IT-Monitoringtools und des IDS genutzt werden. Die letzte Komponente im Setup ist eine Webanwendung, in der alle relevanten Informationen über den aktuellen Zustand des Systems visualisiert werden[Bra+21].

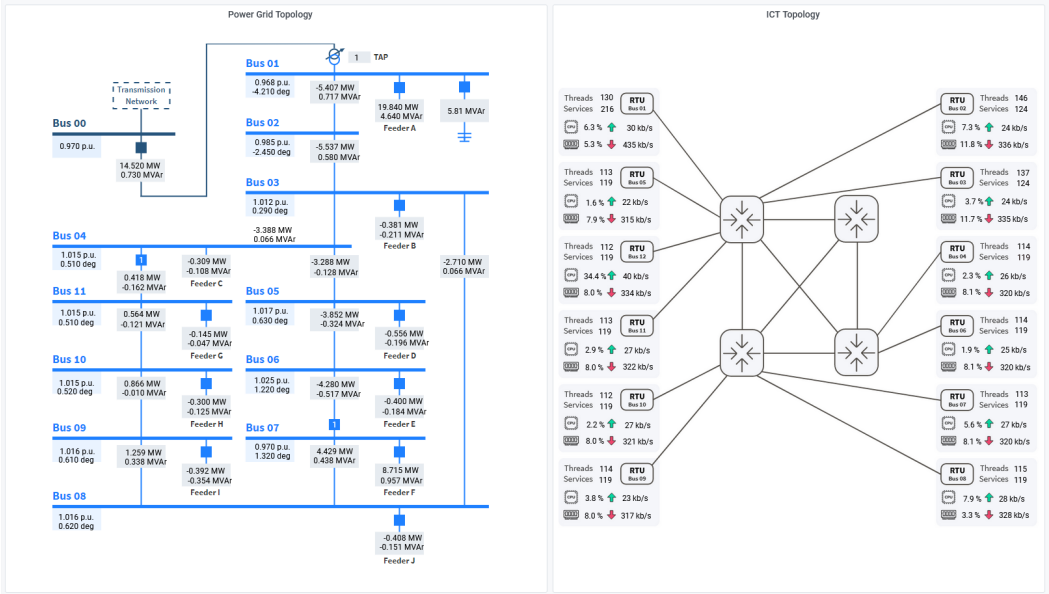


Abb. 6.6.: Kombinierte Topologiedarstellung, angereichert mit Livedaten [Bra+21].

Über die grafische Benutzeroberfläche kann der Operateur den Betriebszustand des Stromnetzes und die Hauptmerkmale des IKT-Systems, das alle im Feld abgerufenen Informationen an die Leitwarte sendet, einsehen. Dieser Teil der Benutzeroberfläche ist in Abbildung 6.6 dargestellt. Darüber hinaus kann der Betreiber auch die Ergebnisse von ASSESS mit detaillierten Informationen über den Trust in die Zustandsvariablen einsehen (dargestellt in Abbildung 6.7). Der aggregierte Wert jeder Trust-Facette wird in einem Netzdiagramm visualisiert. Die einfachen Trust-Werte, die zu den aggregierten Trust-Werten der Facetten beigetragen haben, werden in den Tabellen auf der rechten Seite dargestellt [Bra+21].

In der Demonstration[5] ist zu sehen, dass der FDIA beziehungsweise die durch ihn erzeugten und durch die Trust-Quellen (Monitoringsysteme) wahrgenommenen Anomalien von ASSESS in ein Trust-sensitives Lagebild überführt werden können,

[5]Video zur Demonstration: `https://youtu.be/3hwi49sfllQ`

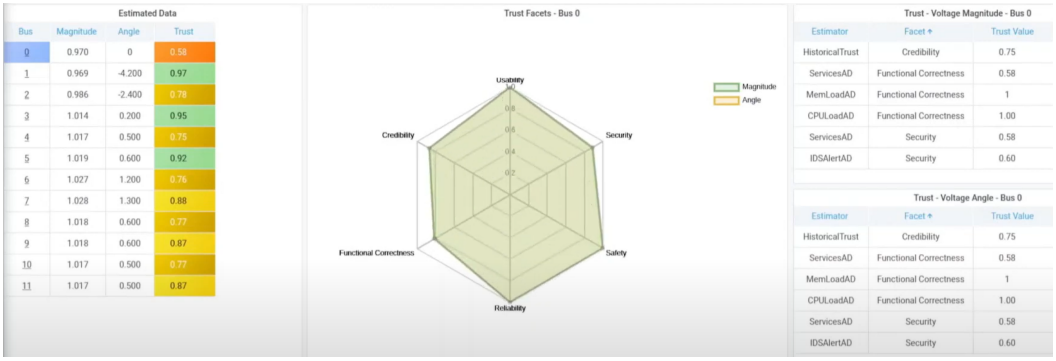


Abb. 6.7.: Visualisierung der Zustandsvariablen angereichert mit multivariaten Trust-Werten [Bra+21].

das den erwarteten geringeren Trust widerspiegelt. Des Weiteren wird der Mehrwert für Operateure bei der Entscheidungsfindung demonstriert, wobei die Interpretation der multivariaten Trust-Werte ausdrücklich nicht Teil dieser Arbeit ist und in der Demonstration angenommen wird, dass der Operateur auf Grundlage der multivariaten Trust-Werte vorsichtshalber keine Steuerungshandlung ausführt.

6.7 Zusammenfassung

In diesem Kapitel wurde mit ASSESS das im Rahmen dieser Arbeit prototypisch implementierte System vorgestellt, mit dem die Forschungsziele und nichtfunktionalen Anforderungen erfüllt werden sollen. ASSESS wurde in Odysseus, einem Framework für maßgeschneiderte DSMSs, umgesetzt, welches in Abschnitt 6.1 beschrieben ist. Odysseus ist modular aufgebaut und verfügt daher über eine große Flexibilität bei der Erstellung konkreter DSMSs. Um unterschiedliche Datenquellen anzubinden, verfügt Odysseus über ein sogenanntes Access Framework, das ebenfalls modular und erweiterbar gestaltet ist, so dass es stets um neue Kommunikations- und Datenprotokolle erweitert werden kann, ohne dass bestehende Komponenten angepasst werden müssen. Anfragen werden in Odysseus-Anwendungen in der Regel mittels einer regelbasierten Skriptsprache namens Odysseus Script formuliert. Dem Nutzer ist es, neben der Formulierung von Anfragen, dadurch möglich, auf verschiedene Variationspunkte von Odysseus Einfluss zu nehmen.

Die Architektur von ASSESS, beschrieben in Abschnitt 6.2, besteht auf höchster Abstraktionsebene aus den folgenden Komponenten:

- dem ITF zur Übersetzung von Messwerten und Topologien mithilfe des Access Frameworks in das M- bzw. T-Modell,

- je einem Topologie- und Trust-Input-Speicher zur zeitstempelsensitiven Zwischenspeicherung (Kontextspeicher) aktueller Topologien bzw. Trust-Inputs,
- dem ADF als Umsetzung der Integrationsplattform für Trust-Schätzer zur Anreicherung der Messwerte mit multivariaten Trust-Werten (M-T-Modell),
- der ASSE als Umsetzung der Trust-sensitiven Lagebilderkennung mit einer State-Estimation- und einer Trust-Estimation-Komponente,
- einer alternativen State Estimation zur Erstellung eines alternativen, im Zweifel vertrauenswürdigerem Lagebildes und
- dem OTF zur Gewährleistung möglichst vertrauenswürdiger Lagebilder sowie einer adäquaten Ausgabedatenrate und zur Übersetzung von Zustandsvariablen im M-T-Modell in beliebige Datenprotokolle und zum Versenden selbiger mittels beliebiger Kommunikationsprotokolle (beides mithilfe des Access Frameworks).

Im weiteren Verlauf dieses Kapitels wurden spezielle Herausforderungen behandelt, die es zu lösen galt, um die nichtfunktionalen Anforderungen zu erfüllen. Flexibilität und Skalierbarkeit werden im Wesentlichen durch den Einsatz von Unteranfragen, Schleifenkonstrukten und der Auslagerung von Variablen gewährleistet (siehe Abschnitt 6.3). Für die technische Interoperabilität, die zusammen mit der Prozessinteroperabilität in Abschnitt 6.4 beschrieben ist, wurde mit OJ104 eine offene Java-Implementierung des 104er-Protokolls vorgenommen, die es ASSESS ermöglicht, Daten nach diesem Standard zu lesen und zu schreiben. Die Möglichkeit, andere Kommunikations- oder Datenformate zu verwenden, wird durch das Access Framework von Odysseus gewährleistet. Für die Gewährleistung eines möglichst vertrauenswürdigen Lagebildes und der Möglichkeit, Lagebilder nur an Operateure oder weiterverarbeitende Systeme weiterzuleiten, wenn es signifikante Änderungen im Lagebild oder Trust gibt, sind die identifizierten Herausforderungen die folgenden beiden: Es muss definiert werden, wann ein Lagebild möglichst vertrauenswürdig ist und wann eine Änderung am Systemzustand oder Trust relevant ist. Eine elaborierte Lösung dieser beiden Herausforderungen bedarf einer Interpretation multivariaten Trusts, was nicht Gegenstand dieser Arbeit ist. Somit wurde für die prototypische Implementierung von ASSESS auf einfache Aggregationen und Schwellwertverfahren zurückgegriffen. Durch den Einsatz von austauschbaren Unteranfragen wurde allerdings die Möglichkeit geschaffen, komplexere Algorithmen einfach zu integrieren.

Die Herausforderungen zur Einhaltung der Anforderung der Aktualität wurden in Abschnitt 6.5 behandelt. Dabei wurden mit Blockierungen, der Bildung von

State-Estimation-Messwertemengen und der Einhaltung der Zeitsemantik bei der Anreicherung mit Kontextinformationen drei wesentliche Herausforderungen identifiziert. Blockierungen werden in ASSESS weitesgehend vermieden, indem auf eine Out-of-Order-Verarbeitung der Datenstromelemente bis zur Bildung der State-Estimation-Messwertemengen gesetzt wird. Für diese Bildung der Messwertemengen wurde ein eigens für diesen Zweck konzipierter Fensteransatz umgesetzt, der mit Systemzeitschaltuhren arbeitet und sowohl eine durchschnittliche State-Estimation-Dauer als auch die Datenraten der Fernwirktechnik berücksichtigt. Um Messwerte zeitsemantisch korrekt mit Kontextinformationen, wie z.B. Topologien, anzureichern, wird in ASSESS auf Odysseus-eigene Kontextspeicher zurückgegriffen, die garantieren, dass zu jedem Zeitpunkt genau eine Kontextinformation gültig ist.

In der in Abschnitt 6.6 beschriebenen Demonstration wurde gezeigt, dass durch einen FDIA erzeugte Anomalien von ASSESS in ein Trust-sensitives Lagebild überführt werden können, das den erwarteten geringeren Trust widerspiegelt. Des Weiteren wurde der Mehrwert für Operateure bei der Entscheidungsfindung demonstriert.

7 Evaluation

> *Eine Theorie ist nur dann wissenschaftlich, wenn sie überprüfbar ist.*
>
> **— Karl Raimund Popper**

Dieses Kapitel widmet sich der Evaluation von ASSESS, dem im vorigen Kapitel beschriebenen System, bezüglich der in Abschnitt 1.2 eingeführten nichtfunktionalen Anforderungen Aktualität, technische und Prozessinteroperabilität, Flexibilität sowie Skalierbarkeit. Dazu werden in Abschnitt 7.1 zunächst die Evaluationsziele definiert, bevor in den Abschnitten 7.2 und 7.3 die für die Evaluation herangezogenen Szenarien und das verwendete Evaluationssetup vorgestellt werden. Die Evaluationsergebnisse werden in Abschnitt 7.4 beschrieben und diskutiert. Das Kapitel endet mit einer Zusammenfassung in Abschnitt 7.5.

Anmerkung: Große Teile des folgenden Kapitels sind ebenfalls in [BEL23] veröffentlicht.

7.1 Ziele und Metriken

Neben den nichtfunktionalen Anforderungen Aktualität, technische und Prozessinteroperabilität, Flexibilität sowie Skalierbarkeit soll ebenfalls die Funktionalität von ASSESS, die in Abschnitt 6.6 demonstiert wurde, evaluiert werden. Konkret geht es bei der Funktionalität von ASSESS darum, „korrekte“ (im Folgenden erläutert) und aussagekräftige Ergebnisse zu liefern.

Tabelle 7.1 listet alle in der Evaluation zu überprüfenden Anforderungen auf und ob die Evaluation qualitativ oder quantitativ erfolgt. „Korrektheit“ meint in diesem Zusammenhang, dass die von ASSESS ausgegebenen Zustandsvariablen mit denen übereinstimmen, die der ASE als verwendeter State-Estimation-Algorithmus als eigenständiges Programm ausgibt. Für eine Aussagekraft der multivariaten Trust-Werte der Zustandsvariablen gelten zwei Kriterien. Zum einen sollten Abweichungen der Zustandsvariablen vom tatsächlichen Zustand des Stromsystems widergespiegelt

werden. Zum anderen sollte es sich dabei um Szenarien handeln, in denen eine Bad Data Detection nach Stand der Technik keine Bad Data erkennt, um den Mehrwert aufzuzeigen.

Tab. 7.1.: Eine tabellarische Übersicht über die Anforderungen, deren Erfüllung evaluiert wird.

Kürzel	Anforderung	Art
Korrektheit	Die Ergebnisse von ASSESS stimmen mit denen des ASE überein.	qualitativ
Aussagekraft	Der multivariate Trust über die Zeit korreliert mit der Abweichung der Zustandsvariablen von ihrem normalen Wert ohne Kompromittierung.	qualitativ
Aktualität	Die Latenz von ASSESS ist in der gleichen Größenordnung wie die Latenz des integrierten ASE.	quantitativ
technische Interoperabilität	ASSESS unterstützt das 104er-Protokoll und ist um die Unterstützung weiterer Protokolle erweiterbar.	qualitativ
Prozess-interoperabilität	ASSESS liefert zum einen unter Verwendung der alternativen State Estimation vertrauenswürdigere Lagebilder als ohne und zum anderen unter Verwendung einer Änderungserkennung nur Ergebnisse, wenn sie sich in den Zustandsvariablen oder dem Trust in diese von der vorigen Ergebnismenge unterscheiden.	qualitativ
Flexibilität	ASSESS ist für unterschiedliche Stromsysteme und Trust-Schätzer anpassbar.	qualitativ
	ASSESS ist für unterschiedliche verwendete Protokolle anpassbar.	argumentativ
Skalierbarkeit	ASSESS skaliert für unterschiedliche Anzahlen an Fernwirkverbindungen und Trust-Schätzer.	qualitativ

Letzteres wird durch das Setup, beschrieben in Abschnitt 7.3, sichergestellt. Die Evaluation, ob die Abweichungen der Zustandsvariablen vom tatsächlichen Zustand des Stromsystems durch die multivariaten Trust-Werte widergespiegelt werden, erfolgt durch eine Korrelationsanalyse.

Bezüglich der Aktualität sei nochmals erwähnt, dass nach aktuellem Stand der Technik die State Estimation als ein zyklischer Prozess durchgeführt wird, bei dem eine State Estimation z.B. alle fünf Minuten durchgeführt wird. Ein, wie in dieser Arbeit verfolgter, ereignisgetriebener Prozess ist zum einen schwerlich mit solch

einem zyklischen Prozess bezüglich der Aktualität zu vergleichen. Zum anderen sind Effizienz und Performanz nicht Gegenstand dieser Arbeit. Daher wird das Ziel verfolgt, dass die durch ASSESS benötigte Zeit in der gleichen Größenordnung liegen sollte wie die einer vergleichbaren State Estimation ohne Trust-Schätzung. Die Aktualität wird quantitativ evaluiert mit dem Ziel zu zeigen, dass die Latenz von ASSESS in der gleichen Größenordnung ist wie die des integrierten ASE. Latenz wird in diesem Zusammenhang als die Zeitdauer zwischen dem Eintreffen eines Datenstromelements und dem Bereitstellen eines Ergebnisses definiert. Die Latenz von ASSESS ergibt sich aus dem Eintreffen des letzten für eine State Estimation verwendeten Messwertes im ITF und der Bereitstellung der Ergebnisse im OTF, während die Latenz des integrierten ASE durch zwei Messpunkte definiert wird, einen direkt vor und einen direkt nach der State-Estimation-Komponente.

Für die Prozessinteroperabilität kommen das „Stacked-Generalization“-Ensemble mit der alternativen State Estimation sowie der Algorithmus zur Änderungserkennung zum Tragen (vgl. Abschnitt 6.4). Um sie zu evaluieren, können die Ergebnisse mit gleichen Setups ohne Prozessinteroperabilität verglichen werden. Durch den Einsatz des „Stacked-Generalization“-Ensembles sollte der aggregierte (minimale) Trust in die Zustandsvariablen größer sein als bei dem gleichen Setup ohne Prozessinteroperabilität. Für den Einsatz der Änderungserkennung gilt, dass sich zwei aufeinander folgende Ergebnismengen in mindestens einer Zustandsvariablen oder im Trust in mindestens eine Zustandsvariable unterscheiden sollten. Ergebnismengen, auf die das nicht zutrifft, sollten aufgrund der Änderungserkennung nicht ausgegeben werden.

Flexibilität und Skalierbarkeit von ASSESS werden evaluiert, indem unterschiedliche Stromsysteme mit unterschiedlichen Anzahlen an Fernwirkverbindungen sowie unterschiedliche Trust-Schätzer und Anzahlen dieser verwendet werden. Eine Flexibilität gegenüber verwendbarer Fernwirkprotokolle wird nicht explizit evaluiert, da in allen Evaluationsszenarien der in Europa weitverbreitete 104er-Standard verwendet wird. Für diese Arbeit wurde allerdings das Access Framework von Odysseus erweitert, um Nachrichten in dem 104er-Standard verarbeiten zu können (vgl. Abschnitt 6.4.1). Dies zeigt, dass Odysseus und damit auch ASSESS um Protokollimplementierungen erweiterbar sind.

7.2 Szenarien

In diesem Abschnitt werden die für die Evaluation umgesetzten und entsprechend verwendeten Szenarien sowie deren Auswahl beschrieben. Die Motivation, verschiedene Szenarien zu betrachten, liegt darin, zu zeigen, dass ASSESS die nichtfunktionalen Anforderungen der Prozessinteroperabilität, Flexibilität und Skalierbarkeit erfüllt. Als Methodologie zur Identifikation möglicher Szenarien wird ein morphologischer Kasten verwendet, dessen Variationspunkte, die die einzelnen Szenarien definieren, im Folgenden erläutert werden.

Um ASSESS bezüglich der Skalierbarkeit zu evaluieren, werden Stromnetze unterschiedlicher Größe (und damit auch eine unterschiedliche Anzahl an Datenquellen für das ITF) betrachtet: ein reduziertes CIGRE Mittelspannungsnetz mit 12 Sammelschienen (CIGRE12MV), das IEEE Hochspannungsnetz mit 39 Sammelschienen (IEEE39HV) und das IEEE Hochspannungsnetz mit 118 Sammelschienen (IEEE118HV). Außerdem wird die Skalierbarkeit bezüglich der Anzahl an Trust-Quellen evaluiert, wobei drei verschiedene Trust-Quellen zur Verfügung stehen: ein IDS, ein IT-Monitoringtool und eine Quelle für historische Trust-Werte. Diese beiden Variationspunkte (Stromnetze und Trust-Quellen) werden außerdem herangezogen, um die Flexibilität von ASSESS zu evaluieren. Eine Flexibilität bezüglich verwendbarer Protokolle für die Messwerte wird, wie in Abschnitt 7.1 bereits erwähnt, nicht evaluiert. In allen Szenarien kommen Protokolle nach dem 104er-Standard zum Einsatz, wodurch die Anforderung an die technische Interoperabilität erfüllt wird. Die Argumentation, warum ASSESS dennoch über eine Flexibilität bezüglich verwendeter Protokolle verfügt, ist Abschnitt 7.1 zu entnehmen.

Ein weiterer Variationspunkt des morphologischen Kastens dient der Evaluation von ASSESS bezüglich der Prozessinteroperabilität. Es werden dabei unterschiedliche sogenannte Interoperabilitätsstrategien verfolgt. Erstens gibt es die Möglichkeit, durch die alternative State Estimation möglichst vertrauenswürdige Lagebilder zu erzeugen (vgl. Abschnitt 6.4.2). Kommt die alternative State Estimation nicht zum Einsatz, wird entsprechend ein Trust-sensitives Lagebild ohne vorige Ersetzung besonders unvertrauenswürdiger Messwerte ausgegeben. Außerdem kann durch den Einsatz einer Änderungserkennung bzw. dem Verzicht des Einsatzes gesteuert werden, ob Lagebilder nur dann ausgegeben werden sollen, wenn es signifikante Änderungen gibt oder immer (vgl. Abschnitt 6.4.3). Dieser Variationspunkt für die Prozessinteroperabilität zeigt ferner die Flexibilität von ASSESS bezüglich des Einsatzes unterschiedlichster Komponenten, wie zum Beispiel der ASSE oder der Änderungserkennung.

Tab. 7.2.: Die Variationspunkte, die den morphologischen Kasten zur Auswahl der Evaluationsszenarien (Tabelle 7.3) definieren.

Variationspunkt	Anforderung	Variationen
Stromnetz	• Flexibilität • Skalierbarkeit	• CIGRE12MV • IEEE39HV • IEEE118HV
Trust-Quellen		• IDS • IT-Monitoring • Historie
Interoperabilitäts-strategie	• Prozess-interoperabilität • Flexibilität	• alternative State Estimation • Änderungserkennung

Tabelle 7.2 gibt einen Überblick über die oben beschriebenen Variationspunkte für den morphologischen Kasten. Insgesamt sind mit diesen Variationspunkten 96 Szenarien möglich. Für die Evaluation von ASSESS wurden sechs Szenarien ausgewählt, die sich zum einen durch unterschiedliche Stromnetze (und deren Größen) und Anzahlen an Trust-Quellen auszeichnen. Zum anderen umfassen sie zwei der vier Interoperabilitätsstrategien: alternative State Estimation und Änderungserkennung zusammen sowie keine besondere Interoperabilitätsmaßnahme. Die sechs Szenarien mit den unterschiedlichen Ausprägungen der Variationspunkte sind auch Tabelle 7.3 zu entnehmen.

7.3 Setup

In diesem Abschnitt wird das Setup beschrieben, in dem die Evaluation durchgeführt wurde. In dem im Folgenden beschriebenen Setup wurde jedes der in Abschnitt 7.2 beschriebenen Szenarien zehn Mal mit einer jeweiligen Dauer von fünf Minuten durchgeführt. Die Anzahl und Dauer ergibt sich aus Anforderungen an die Latenzmessungen. Die Latenzen haben in einem wie in der Evaluation verwendeten relativ statischen Setup zwar keine große Varianz, allerdings kommt es durch das Prozessmanagement des Betriebssystems zu kleineren Schwankungen.

Tab. 7.3.: Ein Ausschnitt aus der Übersicht aller möglichen Szenarien, der die im Rahmen der Evaluation umgesetzten Szenarien enthält. Die Liste aller möglichen Szenarien wurde auf Basis einer Auswahl von Stromnetzen, Trust-Quellen und Interoperabilitätsstrategien und mittels eines morphologischen Kastens definiert. SE steht für State Estimation.

Eigenschaft	**Szenario 1**	**Szenario 2**	**Szenario 3**	**Szenario 4**	**Szenario 5**	**Szenario 6**
Stromnetz	CIGRE12MV	CIGRE12MV	IEEE39HV	IEEE39HV	IEEE118HV	IEEE118HV
Trust-quellen	• IDS • IT-Monitoring • Historie	• IDS • IT-Monitoring • Historie	• IT-Monitoring	• IT-Monitoring	• IDS	• IDS
Inter-operabilitäts-strategie	keine	• Änderungs-erkennung • alternat. SE	keine	• Änderungs-erkennung • alternat. SE	keine	• Änderungs-erkennung • alternat. SE

Aus diesem Grund wurde eine Anzahl an Durchläufen von zehn pro Szenario als genügend erachtet. Da es innerhalb eines Szenarios vor allem darauf ankommt, ASSESS sowohl ohne als auch mit kompromittierten Messwerten zu speisen, sich das Systemverhalten allerdings nicht mit mehr oder weniger State-Estimation-Durchläufen mit den gleichen Messwerten ändert, ist eine Evaluationsdauer von fünf Minuten ausreichend (die State Estimation für das IEEE118HVer-Netz als komplexestes dauert gute 20 Sekunden).

Die Eingangsgrößen für ASSESS (Messwerte und Trust-Inputs) werden für die Evaluation durch eine Co-Simulation bereitgestellt, die in Unterabschnitt 7.3.1 beschrieben wird. ASSESS und die Co-Simulation wurden für die Evaluation jeweils in Containern auf unterschiedlichen virtuellen Maschinen installiert. Durch die Containerisierung wird eine Plattformunabhängigkeit und durch die Verwendung unterschiedlicher virtueller Maschinen eine exklusive Nutzung der Ressourcen der entsprechenden Maschinen erreicht.

Für die zu evaluierenden Szenarien ist es ferner wichtig, Kompromittierungen, konkret FDIAs zu erzeugen, die nicht von einer traditionellen Bad Data Detection erkannt werden können. Dies wird in Unterabschnitt 7.3.2, erläutert. Unterabschnitt 7.3.3 stellt die verwendeten Trust-Schätzer vor, die die von den in Abschnitt 7.2 vorgestellten Trust-Quellen bereitgestellten Trust-Inputs verwenden.

7.3.1 Co-Simulation

Die für die Evaluation umgesetzte Co-Simulation stellt zum einen elektrotechnische Messwerte für die State Estimation und zum anderen Trust-Inputs für die Trust-Schätzer bereit. Es sei angemerkt, dass im Kontext der Trust-Inputs für die Evaluation lediglich die Ausgaben von Trust-Schätzern simuliert werden. Beispielsweise werden in einem Simulator Alarme eines IDS simuliert, statt das IDS an sich zu simulieren oder gar ein richtiges IDS zu verwenden. Die Gründe dafür sind die folgenden. Erstens dient der Aufbau einer Komplexitätsreduktion in der Co-Simulation und der Sicherstellung der Reproduzierbarkeit. Zweitens steht die Konfiguration von Trust-Quellen und Trust-Schätzern nicht im Fokus dieser Arbeit. Drittens wurde die Machbarkeit in der in Abschnitt 6.6 beschriebenen Demonstration gezeigt, in der echte Trust-Quellen die Co-Simulation überwacht haben.

Alle Simulationen orientieren sich an einem gemeinsamen Taktgeber, der einen sekündlichen Takt vorgibt. Allerdings obliegt es den Simulatoren, in welchen Abständen sie Werte erzeugen. Die elektrotechnischen Messwerte werden alle fünf

Sekunden und die Trust-Inputs jede Sekunde simuliert. Ferner ist die Co-Simulation eines Evaluationsszenarios stets in zwei Phasen unterteilt: einer sogenannten normalen Phase, in der weder Kompromittierungen der elektrotechnischen Messwerte noch Anomalien in den Trust-Inputs vorliegen, und einer sogenannten anormalen Phase, in der sich entsprechend die Auswirkungen eines FDIA in den elektrotechnischen Messwerten widerspiegelt und Anomalien in den Trust-Inputs vorliegen. Das Stromnetz wird dabei in einem sogenannten „steady state“ simuliert, was bedeutet, dass jede simulierte Messwertemenge zu jedem Zeitpunkt innerhalb einer Phase die gleiche ist. Gleiches gilt für die Trust-Inputs, von denen konkret die folgenden vier je nach Evaluationsszenario simuliert werden können: Alarme eines IDS, die aktuelle CPU- und RAM-Auslastung der simulierten RTUs sowie die Anzahl an laufenden Systemprozessen auf den simulierten RTUs. In einer normalen Phase werden keine IDS-Alarme, normale CPU- und RAM-Auslastungen und eine normale Anzahl an Systemprozessen simuliert. Für simulierte RTUs, deren elektrotechnischen Messwerte durch einen FDIA manipuliert werden, werden in der anormalen Phase hingegen IDS-Alarme, eine erhöhte CPU- und RAM-Auslastung sowie eine um eins erhöhte Anzahl an Systemprozessen simuliert. Es sei nochmals erwähnt, dass es das Ziel ist, zu evaluieren, wie sich ASSESS bei solchen Eingaben verhält. Ein elaborierterer Co-Simulationsaufbau war hingegen Teil der Demonstration (siehe Abschnitt 6.6).

Die beschriebene Co-Simulation wurde in Odysseus umgesetzt. Die elektrotechnischen Messwerte werden von Odysseus als Nachrichten im 104er-Format pro RTU ausgegeben, wobei eine RTU pro Sammelschiene angenommen wurde. So ergibt sich zum Beispiel für das CIGRE12MV eine Anzahl von zwölf Datenquellen elektrotechnischer Messwerte für ASSESS. Die Trust-Inputs werden im CSV-Format über TCP versendet, wobei eine Kommunikationsverbindung pro Trust-Quelle vorgesehen ist (d.h. eine für das IDS und eine für das IT-Monitoringtool). Durch das Access Framework von Odysseus können sowohl Daten- als auch Übertragungsprotokoll jederzeit angepasst werden. Die konkreten in der Simulation verwendeten elektrotechnischen Messwerte und Trust-Inputs sind Kapitel B des Anhangs zu entnehmen.

7.3.2 Erstellung von Angriffen mit Einspeisung falscher Daten

Um Kompromittierungen von Messwerten zu erhalten, die von einer traditionellen Bad Data Detection nicht erkannt werden können, wird für die Evaluation auf FDIAs zurückgegriffen. Daher ist es notwendig, für alle in der Evaluation verwendeten Stromnetze ein potentielles Ergebnis eines FDIA zu erzeugen.

Skript 7.1 zeigt den als Java-Programm umgesetzten Algorithmus zur Erstellung von FDIAs. Als wesentlichste Eingabe erhält der Algorithmus vom Benutzer den gewünschten Effekt des FDIA, d.h. die angestrebten Änderungen an den Zustandsvariablen $\boldsymbol{x}$, bezeichnet als $\boldsymbol{\Delta_x}$. Zunächst wird dann mit dem ASE als verwendeten State Estimator und den normalen „steady-state“ Messwerten eine State Estimation durchgeführt, um die normalen Zustandsvariablen zu erhalten. Durch eine Addition dieser mit den gewünschten Änderungen ergeben sich die anormalen Zustandsvariablen $\boldsymbol{x}_a$. Ziel des Algorithmus ist es, daraus neue Messwerte $\boldsymbol{z}_a$ zu errechnen, mit denen die anormalen Zustandsvariablen durch eine State Estimation zustandekommen. Dazu wird für jeden Messpunkt aus $\boldsymbol{z}$ mittels der in Abschnitt 2.2.1 beschriebenen Zusammenhänge zwischen Messwerten und Zustandsvariablen der Messwert berechnet, der zu den anormalen Zustandsvariablen passt.

```
1  Δx ← user input
2  x ← ASE(z)
3  xa ← x + Δx
4  za ← ∅
5  ∀j ∈ {1, ..., M}
6        z_aj ← calc_measurement(z_j, xa)
7  return za
```

Skript 7.1: Der für die Erstellung der FDIAs verwendete Algorithmus als Pseudocode.

Für die betrachteten Stromnetze (CIGRE12MV, IEEE39HV und IEEE118HV) wurden die Sammelschienen mit gewünschten Effekten durch eine manuelle Analyse der Topologie bestimmt. Im Anschluss an den in Skript 7.1 beschriebenen Algorithmus wurde dann überprüft, ob der erzeugte Angriff tatsächlich nicht von einer Bad Data Detection erkannt werden kann. Dazu wurde der ASE mit Messwerten $\boldsymbol{z}'$ gespeist, wobei $\boldsymbol{z}'$ den ursprünglichen Messwertevektor $\boldsymbol{z}$ ohne diejenigen Messwerte, die manipuliert werden sollen, darstellt. Der ASE kann für die geschätzten Zustandsvariablen eine Modellabdeckung berechnen, die angibt, ob und zu welchem Grade sich jede Zustandsvariable bestimmen ließ. Demnach eignet sich $\boldsymbol{z}_a$ für einen FDIA, wenn es unter Verwendung von $\boldsymbol{z}'$ zu der Situation kommt, dass sich mindestens eine Zustandsvariable nicht bestimmen lässt. Dies bedeutet, dass es sich bei den weggelassenen Messwerten um eine kritische Messwertemenge handelt und eine Bad Data Detection keine Bad Data darin erkennen kann (vgl. Abschnitt 2.2.3).

7.3.3 Trust-Schätzer

Wie bereits in Abschnitt 7.3.1 erwähnt, stehen für die Evaluation vier unterschiedliche Trust-Inputs zur Verfügung, die entsprechend von Trust-Schätzern verwendet

werden können: Alarme eines IDS, CPU- und RAM-Auslastungen sowie die Anzahl an laufenden Systemprozessen. Im Folgenden werden die für die Evaluation umgesetzten Trust-Schätzer beschrieben, die diese Trust-Inputs verwenden. Außerdem wird ein weiterer Trust-Schätzer beschrieben, dessen Eingabe historische Trust-Werte darstellen.

Trust-Schätzer auf Basis von Intrusion-Detection-System-Alarmen

Der Trust-Schätzer auf Basis von IDS-Alarmen basiert auf einer Metrik von Liu et al. [Liu+15] (vgl. Abschnitt 5.1). In [Liu+15] werden alle für den Untersuchungsgegenstand relevanten Alarme unter Berücksichtigung ihrer Schwere aufsummiert, wodurch sich ein Wertebereich von $[0, \infty]$ ergibt.

$$\begin{aligned} t_{z,\text{IDSAlerts}} &= (\text{IDSAlerts}, p_{\text{IDSAlerts}}) \\ \text{mit } p_{\text{IDSAlerts}} &= \frac{1}{\sqrt{1 + n \cdot \sum_{a \in A} m^{p_a}}} \end{aligned} \tag{7.1}$$

Formel 7.1 beschreibt die Transformationsfunktion für den Trust-Schätzer auf Basis von IDS-Alarmen, basierend auf [Liu+15], jedoch normiert, um einen Wertebereich von $[0, 1]$ zu erhalten. Im Nenner der Gleichung für die Trust-Wahrscheinlichkeit $p_{\text{IDSAlerts}}$ wird für alle betrachteten Alarme $a \in A$ ein in [Liu+15] beschriebener Faktor m mit der Priorität des Alarms p (Priorität ist aufsteigend definiert, während die Schwere oft absteigend definiert wird) potenziert. Für die Evaluation wurde, auf Grundlage einer Sensitivitätsanalyse, $m = 4$ verwendet. Die betrachteten Alarme A werden durch einen Fensteransatz in ASSESS definiert, wobei im Rahmen der Evaluation immer alle in einem Szenario auftretenden Alarme berücksichtigt werden (Fenstergröße ist mindestens so lang wie die Evaluationdauer). Der Faktor n, mit dem die Summe multipliziert wird, ist ein für die Transformationsfunktion integrierter Faktor, um die Abbildung der Summenformel aus [Liu+15] auf eine Trust-Wahrscheinlichkeit konfigurieren zu können. Je kleiner n, desto größer ist $p_{\text{IDSAlerts}}$ für gleiche Alarme. Für die Evaluation wurde, auf Grundlage einer Sensitivitätsanalyse, $n = 0,01$ verwendet. Die Wurzel im Nenner dient der Glättung [Liu+15]. Treten keinerlei Alarme auf, ergibt sich die Trust-Wahrscheinlichkeit zu $p_{\text{IDSAlerts}} = 1$. Andererseits gilt $\lim_{|A| \to \infty} p_{\text{IDSAlerts}} = 0$. Ein Blockdiagramm der Transformationsfunktion und ein UML-Aktivitätsdiagramm des Trust-Schätzers sind Abschnitt A.3 des Anhangs zu entnehmen.

Trust-Schätzer auf Basis von Ressourcenauslastungen

Die beiden Trust-Schätzer auf Basis von CPU- bzw. RAM-Auslastungen werden im Folgenden zusammen beschrieben, da ihre Transformationsfunktionen durch die gleiche simple und nur zur Demonstration der Machbarkeit entwickelten Formel beschrieben wird.

$$
\begin{aligned}
& t_{z,\gamma} = (\gamma, p_\gamma) \quad \forall \gamma \in \{\text{CPULoad}, \text{MemLoad}\} \\
& \text{mit } p_\gamma = \begin{cases} 1 & u \in [u_{\text{normal,min}}, u_{\text{normal,max}}] \\ 0,8 & u \notin [u_{\text{normal,min}}, u_{\text{normal,max}}] \wedge u \in [u_{\text{warn,min}}, u_{\text{warn,max}}] \\ 0 & \text{sonst} \end{cases}
\end{aligned} \tag{7.2}
$$

Die Transformation einer Ressourcenauslastung u (CPU oder RAM) in eine Trust-Wahrscheinlichkeit für eine RTU als Untersuchungsgegenstand geschieht anhand eines Schwellwertverfahrens, das in Formel 7.2 beschrieben ist. Die Variablen $u_{\text{normal,min}}$ und $u_{\text{normal,max}}$ definieren dabei ein Intervall, in dem die Ressourcenauslastung als normal angesehen wird, was zu einer Trust-Wahrscheinlichkeit von 1 führt. In einem weiteren Intervall $[u_{\text{warn,min}}, u_{\text{warn,max}}]$, das $[u_{\text{normal,min}}, u_{\text{normal,max}}]$ komplett umschließt, gilt die Ressourcenauslastung als nicht mehr normal, aber noch nicht alarmierend. Die für Ressourcenauslastungen in diesem Intervall vorgesehene Trust-Wahrscheinlichkeit ist $0,8$. Jegliche Ressourcenauslastung außerhalb dieses Intervalls wird als alarmierend eingestuft (Trust-Wahrscheinlichkeit von 0). Für die Transformationsfunktion auf Basis von CPU-Auslastungen wurde im Rahmen der Evaluation nach Beobachtungen ein Normalbereich von $[1,5; 6,5]$ und ein Warnbereich von $[1; 7]$ gewählt, während die Intervalle für die Transformationsfunktion auf Basis von RAM-Auslastungen $[7,5; 11,5]$ und $[7; 12]$ sind. Ein Blockdiagramm der Transformationsfunktion und ein UML-Aktivitätsdiagramm des Trust-Schätzers sind Abschnitt A.1 des Anhangs zu entnehmen.

Trust-Schätzer auf Basis von laufenden Systemprozessen

Formel 7.3 zeigt die mathematische Beschreibung der Transformation einer Anzahl von laufenden Systemprozessen $\#_s$ in eine Trust-Wahrscheinlichkeit.

$$
\begin{aligned}
t_{z,\text{RunningServices}} &= (\text{RunningServices}, p_{\text{RunningServices}}) \\
\text{mit } p_{\text{RunningServices}} &= \begin{cases} 1 & \#_s = \#_{s,\text{normal}} \\ 0 & \text{sonst} \end{cases}
\end{aligned} \tag{7.3}
$$

Aufgrund der Tatsache, dass auf RTUs in der Regel eine fixe Menge an Prozessen $\#_{s,\text{normal}}$ instanziiert sind, wird die ermittelte Anzahl mit der Größe dieser Menge verglichen. Nur, wenn die angenommene Anzahl an Systemprozessen beobachtet wird, ergibt sich eine Trust-Wahrscheinlichkeit von 1. Andernfalls wird sie auf 0 gesetzt. Im Rahmen der Evaluation wurde eine normale Anzahl von 100 Systemprozessen beobachtet und entsprechend für $\#_{s,\text{normal}}$ verwendet. Ein Blockdiagramm der Transformationsfunktion und ein UML-Aktivitätsdiagramm des Trust-Schätzers sind Abschnitt A.2 des Anhangs zu entnehmen.

Trust-Schätzer auf Basis von historischen Trust-Werten

Ein weiterer Trust-Schätzer basiert auf Trust-Inputs, die in ASSESS selbst generiert werden: historische Trust-Werte für einen Messwert.

$$
\begin{aligned}
t_{z,\text{HistoricalTrust}} &= (\text{HistoricalTrust}, p_{\text{HistoricalTrust}}) \\
\text{mit } p_{\text{HistoricalTrust}} &= \frac{\sum_{z_i \in H} \sum_{\gamma \in \Gamma} w_{z_i} \cdot p_{z_i,\gamma}}{\sum_{z_i \in H} w_{z_i}} \\
\text{und } w_{z_i} &= \max(0, 1 - d \cdot (ts_z - ts_{z_i}))
\end{aligned} \tag{7.4}
$$

Gleichung 7.4 beschreibt, wie dieser Trust-Schätzer definiert ist. Er betrachtet eine Historie H des entsprechenden Messwertes. Die Trust-Wahrscheinlichkeit $p_{\text{HistoricalTrust}}$ ergibt sich dabei aus dem arithmetischen Mittel der historischen Trust-Wahrscheinlichkeiten aller Trust-Schätzer. Das Gewicht w_{z_i} für einen historischen Messwert z_i errechnet sich aus der gewichteten zeitlichen Differenz zwischen dem aktuellen und dem historischen Messwert in Millisekunden. Somit werden jüngere Erfahrungen stärker gewichtet als ältere. Ein Blockdiagramm der Transformationsfunktion und ein UML-Aktivitätsdiagramm des Trust-Schätzers sind Abschnitt A.4 des Anhangs zu entnehmen.

7.4 Ergebnisse

Durch den Einsatz von ASSESS in den in Abschnitt 7.2 beschriebenen Szenarien konnte die Erfüllung der Anforderungen an die technische Interoperabilität, Flexibilität und Skalierbarkeit gezeigt werden. Die „Korrektheit“, also die Übereinstimmung der von ASSESS ausgegebenen Zustandsvariablen mit denen, die der ASE als eigenständiges Programm ausgibt, wurde für alle im Rahmen der Evaluation durchgeführten Szenarien überprüft und ist gegeben.

Im Folgenden werden die Ergebnisse zur Aussagekraft der Trust-sensensitiven Lagebilderkennung in Unterabschnitt 7.4.1, Aktualität in Unterabschnitt 7.4.2 und Prozessinteroperabilität in Unterabschnitt 7.4.3 vorgestellt.

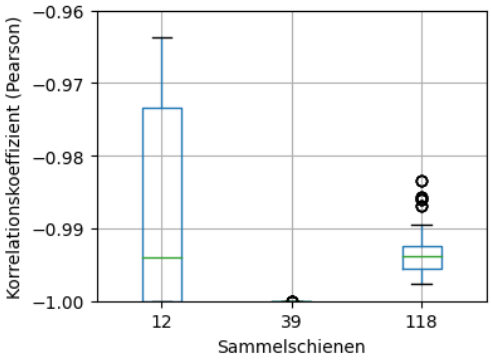
7.4.1 Aussagekraft der Trust-sensensitiven Lagebilderkennung

Die Fragestellung, ob die Trust-sensitive Lagebilderkennung eine relevante Aussagekraft hat, umfasst zwei Anforderungen. Zum einen sollten Abweichungen der Zustandsvariablen vom tatsächlichen Zustand des Stromsystems im Trust widergespiegelt werden. Zum anderen sollte es sich dabei um Kompromittierungen handeln, in denen eine Bad Data Detection nach Stand der Technik keine Bad Data erkennt.

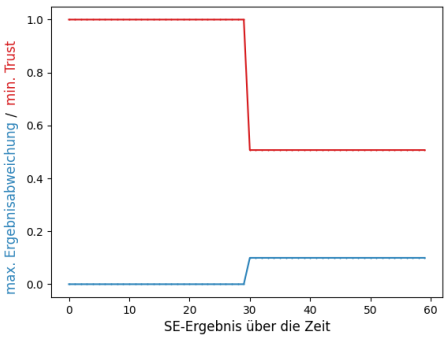
In Unterabschnitt 7.3.2 wurde bereits das Vorgehen zur Erstellung der in der Evaluation verwendeten FDIAs vorgestellt. Da die FDIAs so konstruiert werden, dass die Menge an kompromittierten Messwerten eine kritische Messwertemenge darstellt, ist gegeben, dass eine Bad Data Detection nach Stand der Technik diese Kompromittiertheit nicht erkennen kann (siehe Unterabschnitt 2.2.3).

Die Evaluation, ob die Abweichungen der Zustandsvariablen vom tatsächlichen Zustand des Stromsystems durch die multivariaten Trust-Werte widergespiegelt werden, erfolgt durch eine Korrelationsanalyse nach Pearson. Analysiert wird dabei die Korrelation der folgenden beiden Funktionen. Die erste Funktion stellt den Betrag der Abweichung einer Zustandsvariablen von ihrem erwarteten Wert über die Zeit dar, wobei der erwartete Wert jener ist, der sich aus einer State Estimation mit normalen, nicht kompromittierten Messwerten ergibt. Die zweite Funktion ist der von einem Trust-Schätzer geschätzte Trust in die Zustandsvariable über die Zeit.

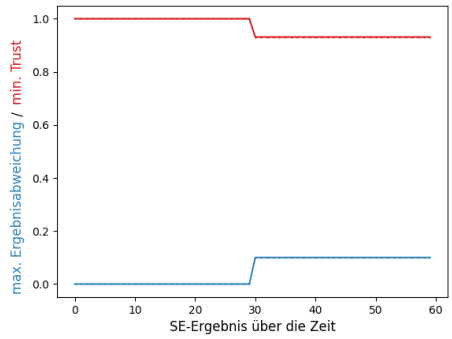
Exemplarisch ist dies in den Abbildungen 7.1b bis 7.1d für die Szenarien 1, 3 und 5 dargestellt. Die x-Achse wird durch die State-Estimation-Ergebnismengen im zeitlichen Verlauf einer Szenariodurchführung definiert. In blau ist jeweils der maximale Abstand einer Zustandsvariablen von ihrem erwarteten Wert dargestellt, aggregiert



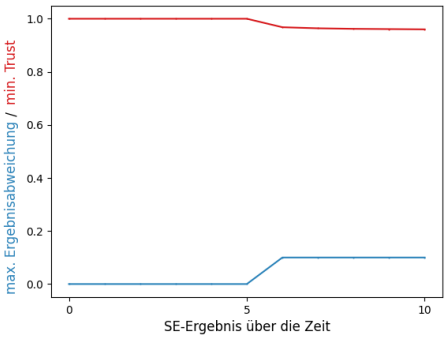
(a) Die Korrelationskoeffizienten nach Pearson als Boxplot pro Szenario (12 Sammelschienen: CIGRE12MV, 39: IEEE39HV, 118: IEEE118HV). In die Boxplots fließen die Korrelationskoeffizienten aller Kombinationen aus Zustandsvariable und Trust-Schätzer ein.



(b) Szenario 1 (CIGRE12MV): Exemplarisch der maximale Abstand einer Zustandsvariablen von ihrem erwarteten Normalwert, aggregiert über alle Zustandsvariablen, dargestellt über die Zeit unten in blau. Der minimale Trust in eine Zustandsvariable, aggregiert über alle Trust-Schätzer und Zustandsvariablen, dargestellt über die Zeit oben in rot.



(c) Szenario 3 (IEEE39HV): Exemplarisch der maximale Abstand einer Zustandsvariablen von ihrem erwarteten Normalwert, aggregiert über alle Zustandsvariablen, dargestellt über die Zeit unten in blau. Der minimale Trust in eine Zustandsvariable, aggregiert über alle Trust-Schätzer und Zustandsvariablen, dargestellt über die Zeit oben in rot.



(d) Szenario 5 (IEEE118HV): Exemplarisch der maximale Abstand einer Zustandsvariablen von ihrem erwarteten Normalwert, aggregiert über alle Zustandsvariablen, dargestellt über die Zeit in blau. Der minimale Trust in eine Zustandsvariable, aggregiert über alle Trust-Schätzer und Zustandsvariablen, dargestellt über die Zeit in rot.

Abb. 7.1.: Auswertung der Aussagekraft der Trust-sensensitiven Lagebilderkennung. Die Abbildungen 7.1b - 7.1d zeigen exemplarische Verläufe und dienen der Illustration.

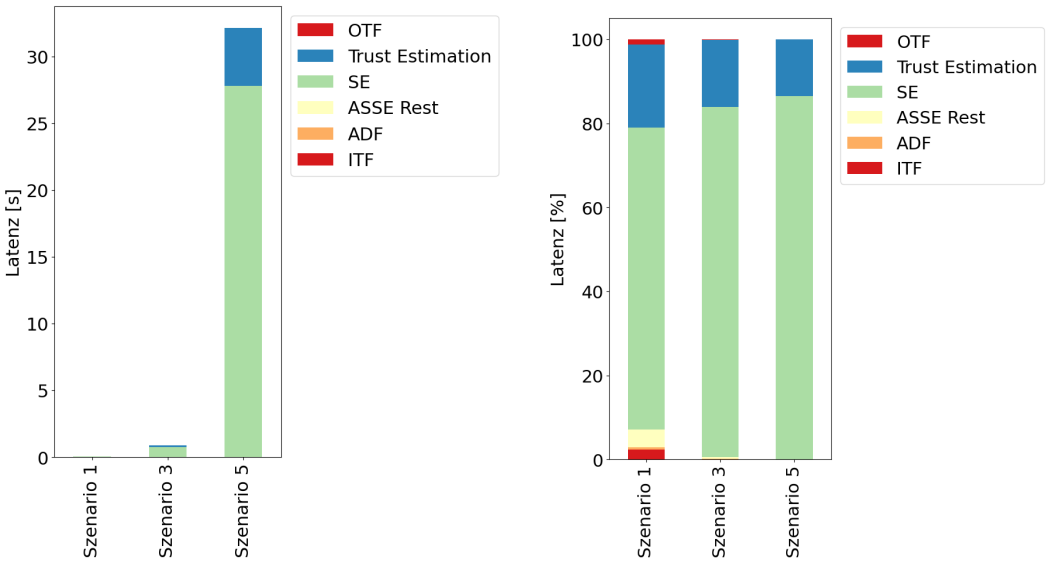
über alle Zustandsvariablen. Die rote Kurve zeigt den minimalen Trust in eine Zustandsvariable, aggregiert über alle Trust-Schätzer und Zustandsvariablen. Diese Abbildungen zeigen bereits, dass die beiden Funktionen miteinander korrelieren, was gleichbedeutend ist mit der Tatsache, dass der Trust in die Zustandsvariablen eine qualitative Aussage über die Kompromittiertheit selbiger zulässt. Untermauert wird dies durch die errechneten Korrelationskoeffizienten nach Pearson, die als Boxplots in Abbildung 7.1a dargestellt sind. Dabei fließen die Korrelationskoeffizienten

aller Kombinationen aus Zustandsvariable und Trust-Schätzer in die Boxplots ein. Korrelationskoeffizienten nach Pearson sind für das Intervall $[-1, 1]$ definiert, wobei 0 einer Unkorreliertheit und ± 1 einer starken positiven bzw. negativen Korrelation entspricht. Abbildung 7.1a zeigt eine starke negative Korrelation zwischen dem Abstand einer Zustandsvariablen von ihrem erwarteten Wert und dem Trust in sie für die durchgeführten Experimente. Das bedeutet, dass eine Kompromittiertheit einer Zustandsvariablen mit einem verringerten Trust in sie einhergeht. Dadurch konnte gezeigt werden, dass die Trust-sensitive Lagebilderkennung eine Aussagekraft hinsichtlich der Kompromittiertheit von Zustandsvariablen hat und dadurch einen Mehrwert darstellt. Die einzelnen Korrelationskoeffizienten nach Pearson für jede Kombination aus Zustandsvariable und Trust-Schätzer und für jedes Szenario sind Abschnitt C.1 des Anhangs zu entnehmen. Der genannte Abschnitt im Anhang enthält darüber hinaus für jedes Szenario Netzdiagramme, die den multivariaten Trust jeweils zu Beginn und Ende der normalen und abnormalen Phase darstellen.

Idealerweise wären die beiden Kurven in den Abbildungen 7.1b bis 7.1d jeweils deckungsgleich, wodurch eine Kompromittiertheit direkt am Trust abgelesen werden könnte. Dies erscheint allerdings nur theoretisch möglich, da es neben einer Interpretation der multivariaten Trust-Werte auch eine exakte Bestimmung des Trusts durch die Trust-Schätzer voraussetzt. Ferner wurden weder Fehleinschätzungen der Trust-Schätzer bzw. der Trust-Quellen berücksichtigt noch wurde ein Übergang von einer anormalen in eine normale Phase analysiert. Dies sind potentielle Aspekte weiterführender Untersuchungen im Anschluss an diese Arbeit.

7.4.2 Aktualität

Die Anforderung an die Aktualität ist, dass die Latenz von ASSESS in der gleichen Größenordnung wie die des integrierten ASE sein muss. Die Latenzen werden dabei in ASSESS wie folgt gemessen. Beim Eintreffen oder Erstellen eines neuen Datenstromelements wird diesem die aktuelle Systemzeit als Startpunkt für die Latenzmessung angehangen. Außerdem werden spezielle Odysseus-Operatoren in die Anfragen integriert, die, wenn ein Datenstromelement sie passiert, diesem ebenfalls die aktuelle Systemzeit anfügen. Dabei können die Operatoren mit unterschiedlichen Messpunkten versehen werden, so dass die gesetzten Systemzeiten jeweils einem Messpunkt zugeordnet werden. Ein Beispiel für einen solchen Messpunkt in ASSESS ist „ITF“, unter dem die Systemzeit abgelegt wird, zu der ein Datenstromelement das ITF verlassen hat. Aus der Differenz zwischen Systemzeiten unterschiedlicher Messpunkte (z.B. ADF und ITF) berechnet sich entsprechend die Latenz für die dazwischenliegende Verarbeitung.



(a) Absolute Latenzen in Sekunden.

(b) Relative Latenzen in Prozenten.

Abb. 7.2.: Die Latenzen (absolut und relativ) für die Szenarien 1, 3 und 5 in Histogrammen, aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

Abbildung 7.2 zeigt die Latenzen von ASSESS für die Szenarien 1, 3 und 5 als gestapelte Histogramme, aufgeschlüsselt nach den Komponenten von ASSESS. Dabei sind Abbildung 7.2a die absoluten Latenzen in Sekunden und Abbildung 7.2b die relativen Latenzen in Prozenten zu entnehmen. In Abbildung 7.2a ist eine deutliche Latenzsteigerung über die Szenarien zu sehen, was in der Größe der jeweils betrachteten Stromnetze begründet ist. Das Stromnetz in Szenario 1 umfasst zwölf Sammelschienen (CIGRE12MV), das in Szenario 3 39 (IEEE39HV) und das in Szenario 5 118 (IEEE118HV). Eine Vergrößerung des Stromnetzes geht einher mit einer deutlich größeren Latenz des ASE (grüne Balken), einer größeren Latenz der Trust-Schätzung für die Zustandsvariablen (blaue Balken) und keiner sichtbaren Latenzänderung für die übrigen Komponenten. Bezüglich der Anforderung, dass die Latenz von ASSESS in der gleichen Größenordnung wie die des integrierten ASE ist, ist in Abbildung 7.2b ersichtlich, dass die Latenz des ASE in allen betrachteten Szenarien deutlich über 50% der Gesamtlatenz ausmacht.

Die genauen absoluten und relativen Latenzen sind Tabelle 7.4 zu entnehmen. Die absolute und relative Latenz für die State Estimation steigt mit der Stromnetzgröße an, wobei die geringste gemessene relative Latenz von ca. 72% bedeutet, dass die Anforderung an die Aktualität erfüllt ist. Für die Trust-Schätzung innerhalb der ASSE („Trust Estimation“) steigt zwar die absolute Latenz mit der Stromnetzgröße an, ihr relativer Anteil an der Gesamtlatenz nimmt allerdings ab. Daraus lässt sich ableiten, dass ihre Latenz nicht so stark mit der Stromnetzgröße wächst wie die der

Tab. 7.4.: Die Latenzen (absolut und relativ) für die Szenarien 1, 3 und 5 aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

	Szenario 1		**Szenario 3**		**Szenario 5**	
	[s]	**[%]**	**[s]**	**[%]**	**[s]**	**[%]**
ITF	0,0	2,384	0,001	0,105	0,001	0,003
ADF	0,0	0,568	0,0	0,018	0,0	0,0
ASSE Rest	0,005	4,116	0,004	0,403	0,008	0,024
State Estimation	0,037	71,99	0,741	83,387	27,808	86,431
Trust Estimation	0,011	19,671	0,142	15,95	4,321	13,528
OTF	0,001	1,256	0,001	0,137	0,004	0,014
total	0,054	100,0	0,888	100,0	32,142	100,0

State Estimation. Bei allen anderen Komponenten (ITF, ADF, „ASSE Rest“, d.h. ASSE ohne State Estimation und Trust-Schätzung, und OTF) gibt es keinen bedeutenden Anstieg der absoluten Latenzen mit der Stromnetzgröße. Ihr prozentualer Anteil an der Gesamtlatenz nimmt analog zu dem Anteil der Trust-Schätzung ab.

Die Ergebnisse legen zum einen nahe, dass die Latenzen aller Komponenten von ASSESS außer der State Estimation und der Trust-Schätzung relativ konstant sind. Zum anderen verdeutlichen sie, dass die Latenz von ASSESS im Wesentlichen durch die Latenz der State Estimation definiert wird, wobei dies für steigende Stromnetzgrößen noch weiter zunimmt.

Abbildung 7.3 zeigt eine parabolische Approximation der Latenzen für ASSESS in rot, für die State Estimation in orange und für die übrigen Komponenten von ASSESS in blau. Die blaue Kurve kann dabei als die durch ASSESS erzeugte zusätzliche Latenz im Vergleich zur alleinigen State Estimation gedeutet werden. Weitere Vergleiche von Latenzen unterschiedlicher Szenarien als Histogramme und in tabellarischer Form sind Abschnitt C.2 des Anhangs zu entnehmen.

7.4.3 Prozessinteroperabilität

Die Evaluation der Prozessinteroperabilität bezieht sich auf Szenarienkombinationen aus dem morphologischen Kasten, deren Szenarien sich lediglich in der Interoperabi-

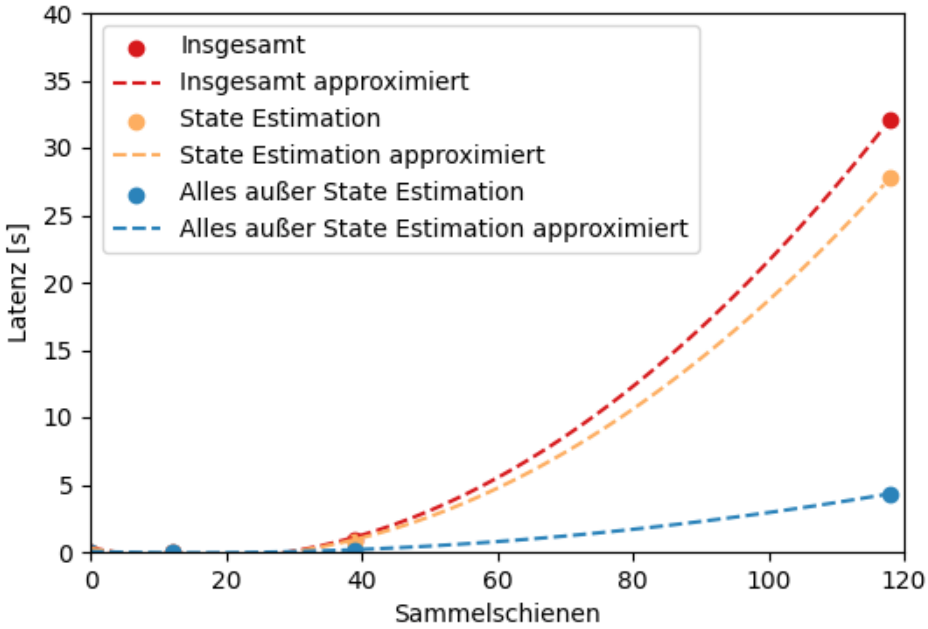


Abb. 7.3.: Die Latenzen in Sekunden in Abhängigkeit der Anzahl an Sammelschienen jeweils als Punkte und parabolisch approximiert.

litätsstrategie unterscheiden. Die Unterschiede in der Interoperabilitätsstrategie sind dabei, dass entweder keine angewandt wird oder sowohl eine Änderungserkennung als auch eine alternative State Estimation zum Einsatz kommen (vgl. Tabelle 7.3 in Abschnitt 7.2). Die konkrete zu evaluierende Anforderung ist, dass ASSESS vertrauenswürdigere Lagebilder liefert und auch nur Ergebnisse liefert, wenn sich ein Ergebnis in den Zustandsvariablen oder dem Trust in diese vom vorigen Ergebnis unterscheidet.

In allen Szenarien, bei denen die Interoperabilitätsstrategie zum Einsatz kam, gab es im Ergebnis lediglich eine State-Estimation-Ergebnismenge, deren Zustandsvariablen nicht von ihren erwarteten Werten abwichen und die als vertrauenswürdig eingestuft wurden. Die Begründung für dieses Ergebnis ist wie folgt. Da jedes Szenario mit einer Phase mit normalen, nicht kompromittierten Messwerten beginnt, sind die Ergebnisse von ASSESS in dieser Phase immer gleich und die Änderungserkennung greift, wodurch das Ergebnis nur einmal weitergeleitet wird. In der anschließenden Phase mit einem FDIA werden die kompromittierten Messwerte für die alternative State Estimation durch historische Messwerte aus der vorigen Phase ersetzt. Da dieses alternative Lagebild als deutlich vertrauenswürdiger eingeschätzt wird als das der ASSE, ist es das Ergebnis des „Stacked-Generalization“-Ensembles (vgl. Unterabschnitt 6.4.2). Da es sich aber auch nicht von den bisherigen Ergebnissen aus

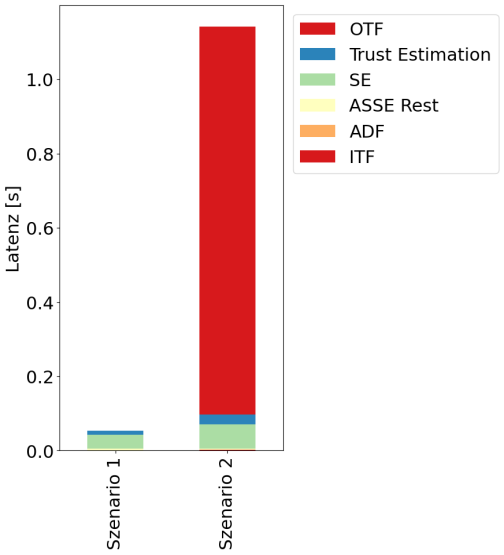
der vorigen Phase unterscheidet, wird auch dieses durch die Änderungserkennung verworfen. Dadurch ist die Anforderung an die Prozessinteroperabilität erfüllt.

Tab. 7.5.: Die absoluten Latenzen für alle Szenarien, jeweils für die Szenarien mit und ohne Interoperabilitätsstrategie gegenübergestellt, aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

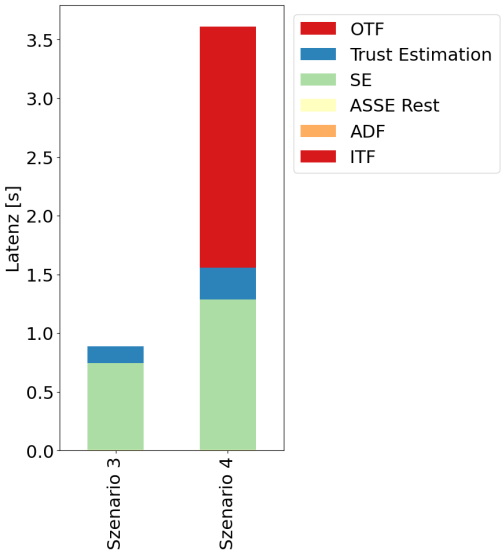
	CIGRE12MV		**IEEE39HV**		**IEEE118HV**	
	Szen. 1	**Szen. 2**	**Szen. 3**	**Szen. 4**	**Szen. 5**	**Szen. 6**
	[s]	**[s]**	**[s]**	**[s]**	**[s]**	**[s]**
ITF	0,0	0,002	0,001	0,001	0,001	0,001
ADF	0,0	0,0	0,0	0,0	0,0	0,0
ASSE Rest	0,005	0,004	0,004	0,005	0,008	0,019
State Estimation	0,037	0,065	0,741	1,283	27,808	53,862
Trust Estimation	0,011	0,026	0,142	0,267	4,321	9,951
OTF	0,001	1,045	0,001	2,055	0,004	2,714
total	0,054	1,142	0,888	3,611	32,142	66,549

Neben der qualitativen Erfüllung der Anforderung ist allerdings noch die Fragestellung interessant, welche Kosten in Form von Latenz die Interoperabilitätsstrategie verursacht. Diese Kosten sind in absoluten Zahlen (Sekunden) Tabelle 7.5 zu entnehmen. Dabei werden jeweils die beiden Szenarien gegenübergestellt, die sich lediglich darin unterscheiden, ob die Interoperabilitätsstrategie zum Einsatz kommt oder nicht. Es ist den Zahlen zu entnehmen, dass die Interoperabilitätsstrategie zu vergleichsweise hohen Kosten bzw. Latenzen bei der State Estimation, der Trust-Estimation und dem OTF führt. Dafür gibt es unterschiedliche Ursachen. Erstens müssen für den Level-1-Generalisierer des „Stacked-Generalization"-Ensembles die beiden Ergebnisse (das der ASSE und das der alternativen State Estimation) jeweils zu einem Datenstromelement zusammengefasst werden. Das bedeutet, dass ein Fenster zum Einsatz kommt, was zu einer Blockierung führt (siehe Unterabschnitt 6.5.1). Gleiches gilt, zweitens, für die anschließende Änderungserkennung. Zwar könnte man argumentieren, dass es bei der Änderungserkennung nicht mehr notwendig ist, die einzelnen Zustandsvariablen mit einem Fenster zusammenzuführen, da dies bereits für den Level-1-Generalisierer geschieht. Allerdings würde ein solches Vorgehen die Flexibilität von ASSESS im OTF stark einschränken. Das durchgängige Datenmodell im OTF ist das M-T-Modell, d.h. ein Messwert pro Datenstromelement.

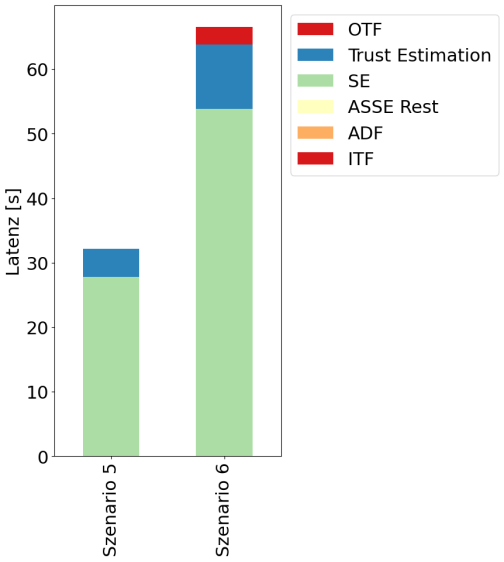
Daher müssen alle integrierbaren Komponenten mit diesem Schema als Ein- und Ausgabe arbeiten.



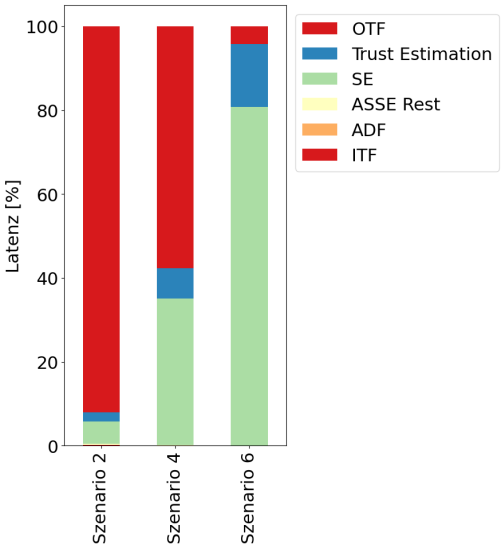
(a) Absolute Latenzen in Sekunden für die Szenarien 1 und 2.



(b) Absolute Latenzen in Sekunden für die Szenarien 3 und 4.



(c) Absolute Latenzen in Sekunden für die Szenarien 5 und 6.



(d) Relative Latenzen in Prozenten für die Szenarien 2, 4 und 6.

Abb. 7.4.: Vergleiche der Latenzen zwischen Szenarien, die sich nur in der Interoperabilitätsstrategie unterscheiden, (a-c) und zwischen Szenarien, die alle die gleiche Interoperabilitätsstrategie umsetzen, (d).

Zwar deuten die Latenzen auf ein erhebliches Optimierungspotential bei der Implementierung der Interoperabilitätsstrategie hin, allerdings wird der Effekt mit steigender Stromnetzgröße deutlich geringer. Dies ist auch in den Histogrammen in Abbildung 7.4 zu sehen. Abbildung 7.4d ist zu entnehmen, wie der relative La-

tenzanteil des OTF mit der Stromnetzgröße abnimmt. Es bleibt demnach bei der Beobachtung, dass für größere Stromnetze die State Estimation die ausschlaggebende Komponente für die Latenz ist.

7.5 Zusammenfassung

In diesem Kapitel wurde die Evaluation von ASSESS beschrieben. Dazu wurden zunächst in Abschnitt 7.1 die Evaluationsziele und Metriken zu deren Überprüfung wie folgt definiert:

- „Korrektheit“: Die von ASSESS geschätzten Zustandsvariablen stimmen mit denen vom ASE geschätzten überein.
- Aussagekraft: Der multivariate Trust in die Zustandsvariablen korreliert mit der Abweichung der Zustandsvariablen von ihrem normalen Wert (ohne Kompromittierung).
- Aktualität: Die Latenz von ASSESS ist in der gleichen Größenordnung wie die des integrierten ASE.
- Technische Interoperabilität: ASSESS unterstützt das 104er-Protokoll und ist um die Unterstützung weiterer Protokolle erweiterbar.
- Prozessinteroperabilität: Erstens liefert ASSESS unter Verwendung der alternativen State Estimation vertrauenswürdigere Lagebilder als ohne. Zweitens produziert ASSESS unter Verwendung der Änderungserkennung nur dann Ergebnisse, wenn sie sich in den Zustandsvariablen oder dem Trust in diese von dem vorigen Ergebnis unterscheiden.
- Flexibilität: ASSESS ist für unterschiedliche Stromsysteme, Trust-Schätzer und verwendete Protokolle anpassbar.
- Skalierbarkeit: ASSESS skaliert für unterschiedliche Anzahlen an Fernwirkverbindungen und Trust-Schätzer.

Im Anschluss wurden in Abschnitt 7.2 die für die Evaluation berücksichtigten Szenarien vorgestellt. Dabei kommen drei unterschiedliche Stromsysteme mit unterschiedlicher Anzahl an Fernwirkverbindungen zum Einsatz, konkret 12, 39 und 118. Außerdem variieren die verwendeten Trust-Schätzer in den Szenarien. Für jede berücksichtigte Kombination aus den beiden genannten Variationspunkten wurde jeweils ein Szenario ohne spezielle Prozessinteroperabilität umgesetzt und eins, in

dem sowohl die alternative State Estimation als auch die Änderungserkennung zum Einsatz kommen.

Das Evaluationssetup, beschrieben in Abschnitt 7.3, besteht aus einer Co-Simulation von RTUs und Trust-Quellen. Die Co-Simulation eines Szenarios ist dabei stets in eine normale Phase, ausgezeichnet durch normale nicht kompromittierte Messwerte und ohne Anomalien in den Trust-Inputs, und eine anormale mit kompromittierten Messwerten und Anomalien unterteilt. Jedes Experiment wurde zehn Mal durchgeführt, da die Latenzen in einem wie in der Evaluation verwendeten relativ statischen Setup zwar keine große Varianz haben, es allerdings durch das Prozessmanagement des Betriebssystems zu kleineren Schwankungen kommt. Die kompromittierten Messwerte wurden dabei durch einen eigens entwickelten Algorithmus zur Erstellung von FDIAs erzeugt, der sicherstellt, dass eine Bad Data Detection nach heutigem Stand der Technik eine solche Kompromittierung nicht erkennen kann.

Die Erfüllung der technischen Interoperabilität, Flexibilität und Skalierbarkeit wurde durch die Ausführung der unterschiedlichen Szenarien gezeigt. Die Auswertung bezüglich der „Korrektheit“, Aussagekraft, Aktualität und Prozessinteroperabilität erfolgte in Abschnitt 7.4. ASSESS liefert stets die gleichen Zustandsvariablen wie der ASE und die einfachen Trust-Werte korrelieren jeweils sehr stark negativ mit den Abweichungen der Zustandsvariablen von ihren Erwartungswerten. Der prozentuale Anteil der Latenz der State-Estimation-Komponente an der Gesamtlatenz liegt, je nach Stromnetzgröße, zwischen 72% und $86,5\%$, wobei er mit der Stromnetzgröße ansteigt. Bei der Verwendung der alternativen State Estimation und der Änderungserkennung zur Prozessinteroperabilität lieferte ASSESS in allen Szenarien jeweils nur ein Ergebnis, bei dem die Zustandsvariablen nicht von ihren Erwartungswerten abweichen und als vertrauenswürdig eingestuft werden. Somit konnte ASSESS bei den in der Evaluation berücksichtigten Szenarien alle Anforderungen erfüllen.

8 Schlussbetrachtung

> *Alle Wissenschaftler versuchen, an der Pyramide menschlichen Wissens weiter zu bauen. Ich hoffe, dass ich einen kleinen Stein dazutun konnte.*
>
> **— Stephen Hawking**

In diesem Kapitel folgt eine abschließende Betrachtung der vorliegenden Arbeit, indem zunächst in Abschnitt 8.1 auf bereits existierende Verwertungen hingewiesen wird. Im Anschluss wird in Abschnitt 8.2 auf identifizierte Limitierungen und offene Aspekte für weiterführende Arbeiten eingegangen. Abschnitt 8.3 schließt diese Arbeit mit einem Fazit ab.

8.1 Verwertung

Die Inhalte der vorliegenden Arbeit wurden bereits vor Veröffentlichung dieser Arbeit auf internationalen Fachtagungen veröffentlicht und diskutiert, was die Relevanz der behandelten Fragestellung und der erzielten Forschungsergebnisse widerspiegelt. Einige ausgewählte Arbeiten werden im Folgenden beschrieben.

In [BNL23][1] wird ein Vorgehen vorgestellt, um die Korrektheit zweier auf einander aufbauender Services zu bestimmen. Dies geschieht auf Basis des multivariaten Trusts in Messwerte, Zustandsvariablen und Fernwirkstationen. Bei den Services handelt es sich um eine State Estimation und eine koordinierte Spannungshaltung. Die Arbeit zeigt, dass sich PSNA-Trust auch auf weitere Services als die State Estimation anwenden lässt. Außerdem behandelt [BNL23] ausdrücklich die Fortpflanzung multivariater Trust-Werte über auf einander aufbauende Services, da die koordinierte Spannungshaltung als eine wesentliche Eingabe die von der State Estimation geschätzten Zustandsvariablen erhält. Zu dem Thema der Fortpflanzung multivariater Trust-Werte in Digitalen Zwillingen (als Verallgemeinerung von SCADA-Systemen) ist ebenfalls eine Dissertation geplant. Mit [Has+21] beschäftigt sich zudem eine

[1] Die Arbeit ist noch unter Begutachtung und daher noch nicht veröffentlicht.

weitere Arbeit mit der Verwendung von PSNA-Trust, um die Performanz einer State Estimation zu erfassen.

Um alle Informationen, die bezüglich der Sicherheit von CPESs relevant sind, zu korrelieren und für menschliche Operateure derart aufzubereiten, dass sie diese interpretieren können und somit einen Überblick über die aktuelle Sicherheit des CPES erhalten, wird in [SFL22] ebenfalls auf PSNA-Trust zurückgegriffen. Ein weiterer Aspekt, durch den PSNA-Trust in der Arbeit erweitert werden soll, ist die Rückverfolgbarkeit von Trust-Einbußen zu den konkreten Ursachen und deren Lokalisation. Neben der (Informations-) Sicherheit spielt die Resilienz von CPESs eine immer entscheidendere Rolle. Die Autoren von [Bab+22][2] identifizieren eine cyber-physische Lagebilderkennung (nicht nur auf elektrotechnischen Prozessdaten basierend) als einen wesentlichen Baustein für resiliente CPESs und ASSESS als einen vielversprechenden Ansatz für eine solche. Damit wird durch ASSESS ebenfalls ein Beitrag zur Resilienzforschung geleistet. Eine weitere Veröffentlichung zu diesem Thema ist in Planung.

8.2 Limitierungen und offene Aspekte für weiterführende Arbeiten

PSNA-Trust basiert sehr stark auf OC-Trust und definiert sich entsprechend über die gleichen Trust-Facetten. Offen ist die Frage, ob Trust und seine Facetten für die generelle Anwendung in CPESs anders definiert sein sollten. Zwar ist das Datenmodell von PSNA-Trust weitgehend unabhängig von den Trust-Facetten und deren Anzahl, allerdings sind die konkreten Trust-Facetten relevant für die Trust-Erhebung und Interpretation der multivariaten Trust-Werte. In einem geplanten Forschungsprojekt soll u.a. diese Fragestellung behandelt werden. Ein weiterer Schwerpunkt dieses Forschungsprojektes soll eine hierarchische oder sogar verteilte Trust-Erhebung sein. Eine solche, z.B. hierarchische, Trust-Erhebung verlangt nach einer Aggregation multivariater Trust-Werte vor der Weitergabe, was nicht in der vorliegenden Arbeit betrachtet wurde. Ferner wird entsprechend eine Disaggegregation notwendig, um beispielsweise Trust-Einbußen zu deren Quellen zurückverfolgen zu können. In diesem Zusammenhang sei auch auf eine Limitierung der vorliegenden Arbeit hingewiesen. Im Rahmen der Trust-Erhebung für die elektrotechnischen Messwerte als abgeleitete Untersuchungsgegenstände werden die für Komponenten, wie zum Beispiel RTUs, erhobenen einfachen Trust-Werte aggregriert. Somit sind Trust-Einbußen

[2] Der Autor der vorliegenden Arbeit ist einer der Autoren von [Bab+22].

in ASSESS lediglich bis zu Messwerten zurückzuverfolgen, allerdings nicht bis zu den eigentlichen Untersuchungsgegenständen. Wie in Abschnitt 8.1 bereits erwähnt, ist diese Rückverfolgbarkeit auch Gegenstand einer weiteren Doktorarbeit [SFL22].

Eine weitere Limitierung dieser Arbeit besteht in der Nutzung weitgehend sehr einfacher Trust-Schätzer. Die Erforschung und Entwicklung elaborierterer Trust-Schätzer bzw. Transformationsfunktionen ist entsprechend ein Aspekt für weiterführende Arbeiten. Die Aussagekraft multivariater Trust-Werte (vgl. Kapitel 7) hängt zu einem gewissen Grad von der Güte der verwendeten Trust-Schätzer ab. Außerdem wurden in der durchgeführten Evaluation Fehlalarme nicht berücksichtigt bzw. gar nicht erst simuliert. Der Umgang mit Fehlalarmen bei den Trust-Inputs kann aber durchaus ein Gütekriterium für Trust-Schätzer sein. Auch tragen die im Rahmen dieser Arbeit umgesetzten Trust-Schätzer lediglich zur funktionalen Korrektheit, Informationssicherheit und Glaubwürdigkeit bei. Als Ursache für eine Kompromittierung wird sowohl in der Demonstration als auch in der Evaluation nur ein FDIA herangezogen. Zwar stellt dies keine Limitierung dar, da ASSESS von den Ursachen für Kompromittierungen abstrahiert, allerdings kann die Betrachtung vieler unterschiedlicher Kompromitierungsszenarien den Mehrwert von ASSESS unterstreichen.

Zwar ist die Informationssicherheit nur eine der Trust-Facetten, allerdings stellen Cyberangriffe und insbesondere FDIAs eine große Bedrohung dar, wie bereits in Abschnitt 1.1 dargestellt. Um speziell für die Informationssicherheit den Mehrwert von ASSESS genauer zu evaluieren, bedarf es einer umfassenden Sicherheitsanalyse, die die folgenden Aspekte umfasst:

- Angreifer: eine Definition der in Frage kommenden Angreiferklassen (z.B. staatlich finanzierte Angreifer) sowie deren verfügbare Ressourcen und Möglichkeiten (wie z.B. für den Cyberangriff auf das Ukrainische Stromversorgungssystem 2015 [Ele16]),
- Angriff: eine detaillierte Modellierung der potentiellen Angriffsvektoren (wie z.B. ebenfalls für den oben genannten Cyberangriff [Ele16]) und
- Sicherheitsannahmen: eine umfassende Übersicht über die Annahmen bzgl. der Informationssicherheit (z.B., dass Trust-Quellen selbst vertrauenswürdig sind).

Basierend auf diesen Informationen können dann, je nach Angreifer, Angriff und Sicherheitsannahmen, die potentiellen Auswirkungen mit und ohne dem Einsatz von ASSESS analysiert werden. Allerdings hängen die potentiellen Angriffsvektoren bzw. das Wirken von ASSESS gegen diese, und die Sicherheitsannahmen stark von den verwendeten Trust-Quellen und -schätzern ab, so dass eine Sicherheitsanalyse

nicht allgemein für ASSESS durchgeführt werden kann. Sollten aber Trust-Quellen und -schätzer eingesetzt werden, für die bereits entsprechende Sicherheitsanalysen durchgeführt wurden, lassen sich diese wiederverwenden.

Eine weitere relevante Fragestellung bzgl. der Sicherheitsanalyse ist, ob und unter welchen Umständen Angreifer ASSESS gezielt umgehen können, wie es koordinierte FDIAs mit der Bad Data Detection tun können. Ohne einer ausführlichen Sicherheitsanalyse, die nicht Bestandteil dieser Arbeit ist, vorgreifen zu wollen, erscheint es aber so, dass eine Absicherung von ASSESS gegen gezielte Angriffe mit der Anzahl an unterschiedlichen Trust-Quellen und -schätzern zunimmt, da sowohl die Komplexität eines erfolgreichen Angriffsvektors als auch die dafür benötigten Informationen zunehmen.

Bezüglich der Integrationsplattform für Trust-Schätzer kann es für die Zukunft sinnvoll sein, sowohl für das T-Modell als auch für das M-Modell auf CIM zurückzugreifen, um die Interoperabilität weiter zu erhöhen. Ferner könnte eine Abbildung der Trust-Erhebung auf das in der Domäne weit verbreitete IEC 62559-2-Anwendungsfalltemplate [GUD17] und das Smart Grid Architecture Model [GUD17] zu einer standardisierteren und interoperableren Modellierung von Trust und dessen Erhebung führen.

In der vorliegenden Arbeit wurde die Trust-sensitive Lagebilderkennung auf Basis des verwendeten State-Estimation-Verfahrens durchgeführt, da das benötigte Domänenwissen in dem State Estimation-Verfahren enthalten ist und die Trust-Schätzung auf diese Informationen zurückgreifen kann. Es sind allerdings auch andere Verfahren zur Trust-Schätzung denkbar, wie zum Beispiel Verfahren des maschinellen Lernens oder Korrelationsverfahren. Bei beiden Verfahrensarten wird allerdings eine Herausforderung sein, dass Trust-Einbußen statistisch selten auftreten und daher bei der Gütebewertung eines Modells unterrepräsentiert wären. Es bedarf demnach Verfahren, die berücksichtigen, dass im Kontext der ASSE Randfälle interessanter sind als der Regelfall.

Ein weiterer Aspekt für weiterführende Arbeiten ist die Fortpflanzung multivariater Trust-Werte über Services hinweg. Zwar wurde dieses Thema auch in der vorliegenden Arbeit behandelt, da zunächst der Trust im Datenakquiseservice erhoben, anschließend an die ASSE weitergereicht und zuletzt durch die ASSE zur Einschätzung des Trusts in die Zustandsvariablen verwendet wurde. Die vorliegende Arbeit ist allerdings auf diese beiden Services limitiert. Ein abstrakteres Konzept und eine Umsetzung, die die wichtigsten Services in einem CPES umfasst, wären wünschenswert. Allerdings ist anzunehmen, dass die Fortpflanzung multivariater Trust-Werte

von Ein- zu Ausgaben eines Servies sehr servicespezifisch und daher für jeden Service neu zu entwickeln ist, falls nicht auf Verfahren des maschinellen Lernens o.ä. zurückgegriffen wird. Ein solches umfangreiches Trust-Propagation-Modell würde auch neue Schwachstellenanalysen erlauben. Zum Beispiel könnte mit Monte-Carlo-Simulationen der Einfluss von Trust-Einbußen in der Datenakquise auf beliebige Services und das CPES als Ganzes analysiert werden. Für die Fortpflanzung von Trust in digitalen Zwillingen zur Überwachung und Steuerung von CPESs ist bereits eine weitere Doktorarbeit geplant (siehe Abschnitt 8.1).

Wie bereits angedeutet, kann PSNA-Trust und die Trust-Erhebung, die in dieser Arbeit für die Domäne eingeführt, aber nur auf die State Estimation angewandt wurden, derart erweitert werden, dass der Trust in das gesamte CPES erhoben wird. Dadurch ergibt sich auch die Forschungsfrage, ob eine solche Trust-Erhebung mit den unterschiedlichen, vielleicht in Zukunft neu definierten Trust-Facetten die bisherige Grundlage zur Zustandsbestimmung in einem CPES ablösen kann. Neben elektrotechnischen Prozessdaten (Zustandsvariablen, Engpassanalysen, etc.) könnten weichere Kriterien wie die Informationssicherheit integriert werden, um ein holistischeres Lagebild zu erlangen. Ein solches Vorgehen erfordert allerdings auch eine Interpretation multivariater Trust-Werte, was, wie bereits mehrfach angesprochen, nicht Bestandteil dieser Arbeit ist, aber ein Untersuchungsgegenstand zukünftiger Arbeiten sein sollte. Außerdem bedarf es neben einer Interpretation auch einer adäquaten Darstellung multivariater Trust-Werte, entweder in Form eines Assistenzsystems oder integriert in existierende SCADA-Systeme.

Bei der Umsetzung von ASSESS in Odysseus wurde versucht, Blockierungen weitestgehend zu vermeiden und somit eine daten- oder ereignisgetriebene State Estimation umzusetzen. Dies ist auch bis auf die Erstellung der Messwertemenge für die State Estimation gelungen. In einer weiterführenden Arbeit kann allerdings untersucht werden, ob nicht auch diese Blockierung aufgelöst oder gemildert werden könnte. Dafür wäre allerdings ein State-Estimation-Verfahren notwendig, das inkrementell arbeitet und nicht bei jedem Durchlauf eine vollständige Messwertemenge benötigt.

8.3 Fazit

In dieser Arbeit wurde mit ASSESS ein System vorgestellt, mit dem es möglich ist, den multivariaten Trust in physische Messwerte in einem CPES zu modellieren, zu schätzen und in eine Lagebilderkennung einfließen zu lassen. Die Demonstration

(Abschnitt 6.6) zeigt die Machbarkeit und den Mehrwert des Ansatzes, da eine schädliche Steuerungshandlung aufgrund einer Einspeisung falscher Daten (engl. false data injection attack) (FDIA) vermieden werden konnte. Dies ist allerdings auch durch andere bereits existierende Lösungen, z.B. im Bereich verbesserter Bad Data Detection, möglich. Der in dieser Arbeit verwendete holistische Ansatz durch ein Trust-Modell eröffnet allerdings, im Gegensatz zu anderen Lösungen, sowohl eine Integration unterschiedlicher Erkennungsverfahren als auch eine Ausweitung auf andere Kompromittierungsursachen. Ferner ist, wie in Abschnitt 8.2 beschrieben, denkbar, Trust und dessen Erhebung zu verwenden, um die Definition des Zustands eines CPES neu zu denken. Aufgrund dieser Überlegungen und der Tatsache, dass bis dato keine anderen vergleichbaren Ansätze bekannt sind, erscheint die in Abschnitt 1.1 aufgestellte Hypothese, dass es für eine vertrauenswürdige Lagebilderkennung notwendig ist, den kontextabhängigen und multivariaten Trust in Prozessvariablen zu erfassen, noch immer gültig zu sein.

Die durchgeführte Evaluation (Kapitel 7) zeigt, dass sich die Verwendung eines Datenstrommanagementsystems (DSMS) und konkret von Odysseus als technologische Grundlage von ASSESS eignet, um die nichtfunktionalen Anforderungen der Aktualität, technischen und Prozessinteroperabilität, Flexibilität sowie Skalierbarkeit umzusetzen. Die datengetriebene Verarbeitung sorgt für deutlich aktuellere Lagebilder bezüglich Änderungen an sowohl den Zustandsvariablen als auch dem Trust in diese. Dies wird in immer komplexer und dynamisch werdenden Systemen, wie es CPESs sind, immer wichtiger. Diese Arbeit zeigt die Einsatzmöglichkeit von DSMSs zu diesem Zwecke und auch deren Mehrwert. Ein weiterer Mehrwert begründet sich aus der gezeigten Flexibilität und Skalierbarkeit des Ansatzes, wobei die Skalierbarkeit sicherlich noch eingehender untersucht werden muss, da zwei der drei getesteten Stromnetzgrößen deutlich unter realen Stromnetzgrößen liegen. Für ansteigende Stromnetzgrößen wurde jedoch die durch ASSESS zusätzlich verursachte Latenz im Verhältnis deutlich geringer, was als sehr positiv zu bewerten ist.

Die neben dem Gesamtsystem ASSESS in dieser Arbeit erforschten Lösungen zur Beantwortung der Forschungsfrage, konkret ein kontextsensitives, multivariates Trust-Modell (Kapitel 3), eine Integrationsplattform für Trust-Schätzer (Kapitel 4) und eine Trust-sensitive State Estimation (Kapitel 5), sind allerdings unabhängig von deren Umsetzung in einem DSMS. Das Trust-Modell PSNA-Trust ist sicherlich das allgemeingültigste Artefakt, welches auch von dem Anwendungsfall einer Trust-sensitiven State Estimation abstrahiert. Zwar basiert es auf OC-Trust und hat dessen Trust-Facetten übernommen, bei denen fraglich ist, ob sie die für die Anwendung in einem CPES geeignet sind, allerdings können sowohl Anzahl als auch Definition der Trust-Facetten einfach verändert werden.

Das Anomalieerkennungsframework (engl. anomaly detection framework) (ADF) als Integrationsplattform für Trust-Schätzer ist vielleicht das schwächste Artefakt, allerdings notwendig, um eine flexible und multivariate Trust-Erhebung zu ermöglichen. Ein wesentlicher Forschungsbereich dürfte im Anschluss an diese Arbeit die Auswahl, Umsetzung und Konfiguration geeigneter Trust-Schätzer sein. ASSESS liefert allerdings mit dem ADF und wohl definierten Schnittstellen eine einfache Integration dieser Trust-Schätzer.

Bei der Trust-sensitiven Lagebilderkennung werden für einen konkreten Service, den der State Estimation, multivariate Trust-Werte eingehender in selbige ausgehender Prozessdaten transformiert. Diese Transformation ist zum einen sehr servicespezifisch, da sie auf den gleichen mathematischen Berechnungen wie in der State Estimation beruht, zeigt allerdings auf, wie dies auch für andere Services umgesetzt werden kann. Dabei ist allerdings zu beachten, dass andere Services, wie zum Beispiel eine koordinierte Spannungshaltung, komplexer sind, da sie ausführende Komponenten im Feld beinhalten.

Zusammenfassend lässt sich sagen, dass die vorliegende Arbeit eine Antwort auf die Forschungsfrage gibt, wie der multivariate Trust in physische Messwerte in einem CPES modelliert, geschätzt und in eine Lagebilderkennung integriert werden kann. Auch werden die identifizierten nichtfunktionalen Anforderungen an die Aktualität, technische und Prozessinteroperabilität, Flexibilität sowie Skalierbarkeit erfüllt.

Anhang

A Weitere Abbildungen zu den verwendeten Trust-Schätzern

A.1 Trust-Schätzer auf Basis von Ressourcenauslastungsdaten

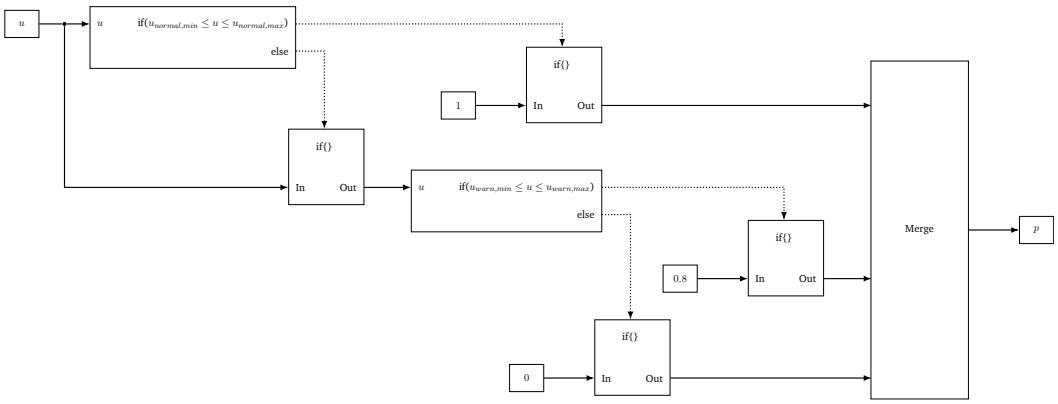


Abb. A.1.: Blockdiagramm einer Transformationsfunktion für Ressourcenauslastungen. Ist eine Resourcenauslastung u innerhalb eines Intervalls $[u_{normal,min}, u_{normal_max}]$, so ist der geschätzte Trust 1. Für u innerhalb eines größeren Intervalls $[u_{warn,min}, u_{warn,max}]$ wird der Trust auf $0,8$ und ansonsten auf 0 geschätzt. Diese Abbildung ist identisch mit Abbildung 3.8.

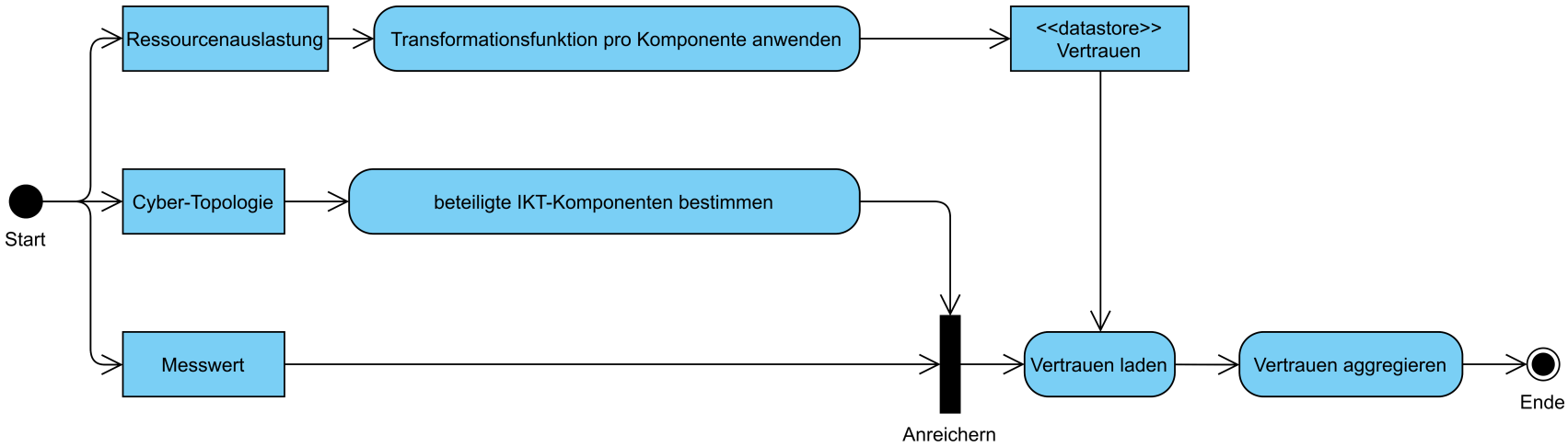


Abb. A.2.: Das UML-Aktivitätsdiagramm für einen Trust-Schätzer auf Basis von Ressourcenauslastungsdaten. Es setzt dabei für jede Komponente, z.B. eine RTU, die Transformationsfunktion aus Abbildung A.1 um. Für einen Messwert werden alle Komponenten bestimmt, die an der Datenakquise aller Wahrscheinlichkeit nach beteiligt waren. Die Ausgaben (Trust-Wahrscheinlichkeiten) der Transformationsfunktionen für diese Geräte werden entsprechend aggregiert. Diese Abbildung ist identisch mit Abbildung 4.7.

A.2 Trust-Schätzer auf Basis von Prozessinformationen

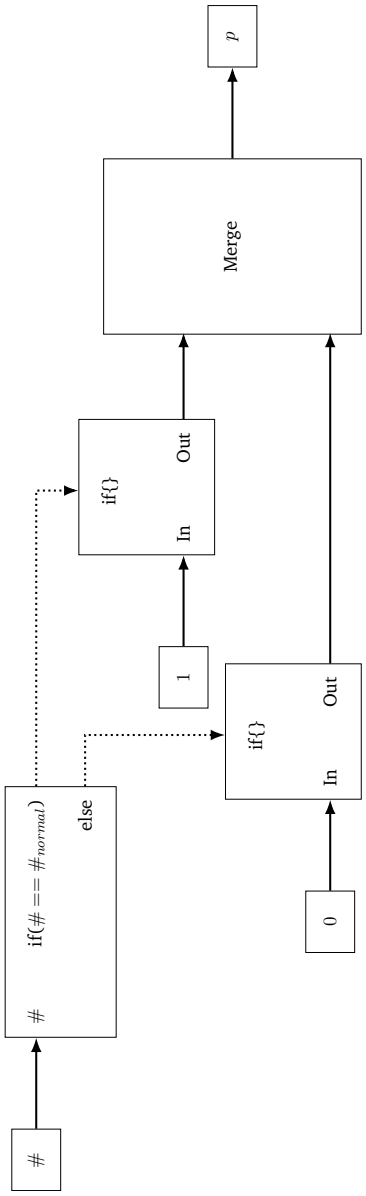


Abb. A.3.: Eine Transformationsfunktion für Prozessinformationen modelliert als Blockdiagramm. Entspricht eine Anzahl an laufenden Prozessen $\#$ der erwarteten Anzahl $\#_{normal}$, so ist der geschätzte Trust 1 und ansonsten 0.

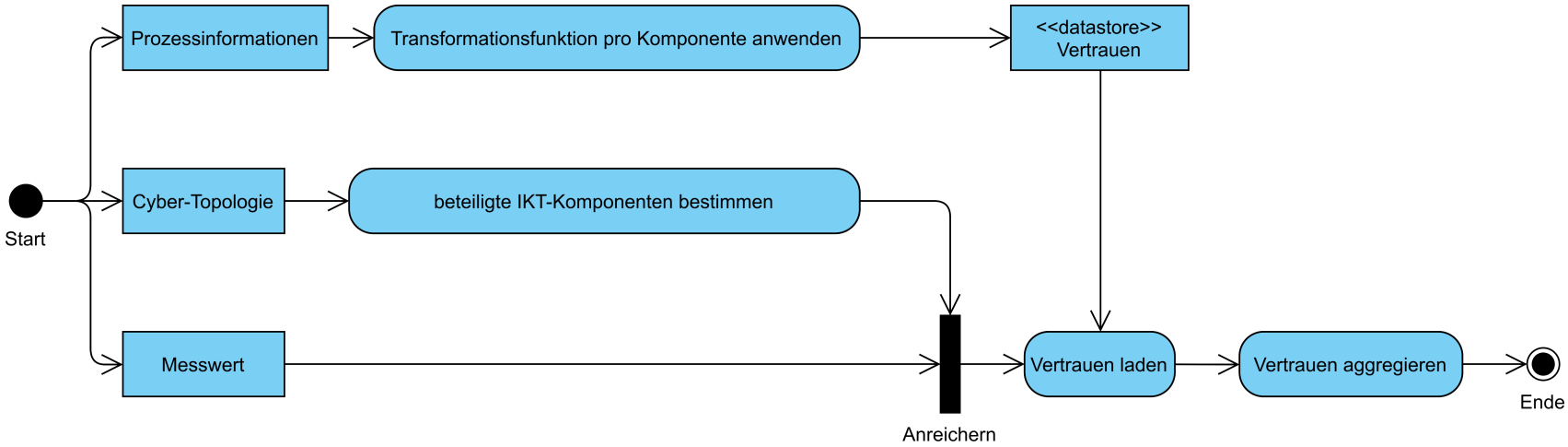


Abb. A.4.: Das UML-Aktivitätsdiagramm für einen Trust-Schätzer auf Basis von Prozessinformationen. Es setzt dabei für jede Komponente, z.B. eine RTU, die Transformationsfunktion aus Abbildung A.3 um. Für einen Messwert werden alle Komponenten bestimmt, die an der Datenakquise aller Wahrscheinlichkeit nach beteiligt waren. Die Ausgaben (Trust-Wahrscheinlichkeiten) der Transformationsfunktionen für diese Geräte werden entsprechend aggregiert.

A.3 Trust-Schätzer auf Basis von Intrusion-Detection-System-Alarmen

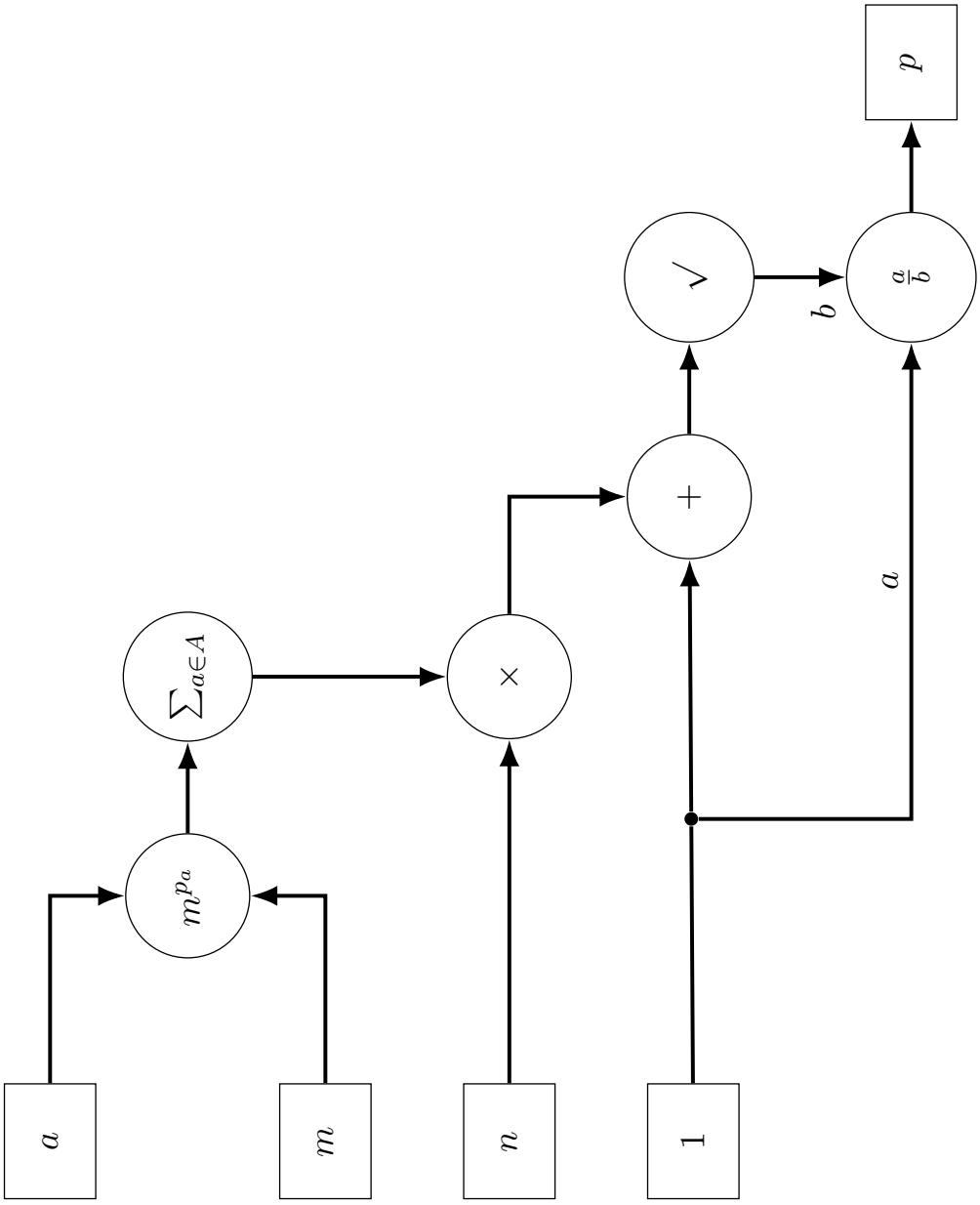


Abb. A.5.: Eine Transformationsfunktion für Alarme eines IDS modelliert als Blockdiagramm. Das Vorgehen entspricht Formel 7.1.

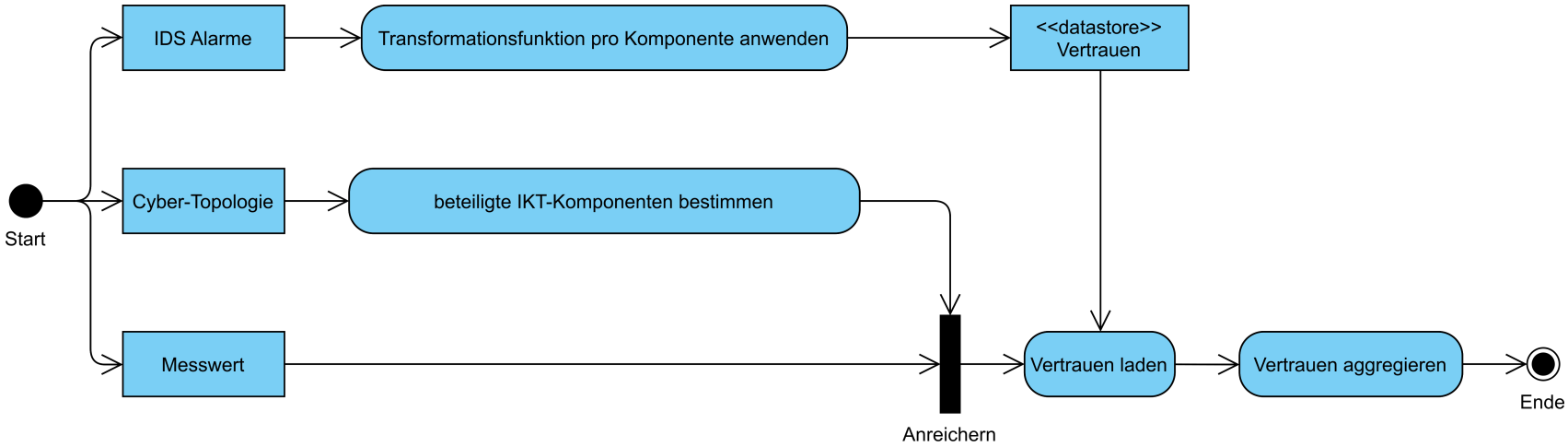


Abb. A.6.: Das UML-Aktivitätsdiagramm für einen Trust-Schätzer auf Basis von Alarmen eines IDS. Es setzt dabei für jede Komponente, z.B. eine RTU, die Transformationsfunktion aus Abbildung A.5 um. Für einen Messwert werden alle Komponenten bestimmt, die an der Datenakquise aller Wahrscheinlichkeit nach beteiligt waren. Die Ausgaben (Trust-Wahrscheinlichkeiten) der Transformationsfunktionen für diese Geräte werden entsprechend aggregiert.

A.4 Trust-Schätzer auf Basis historischen Trusts

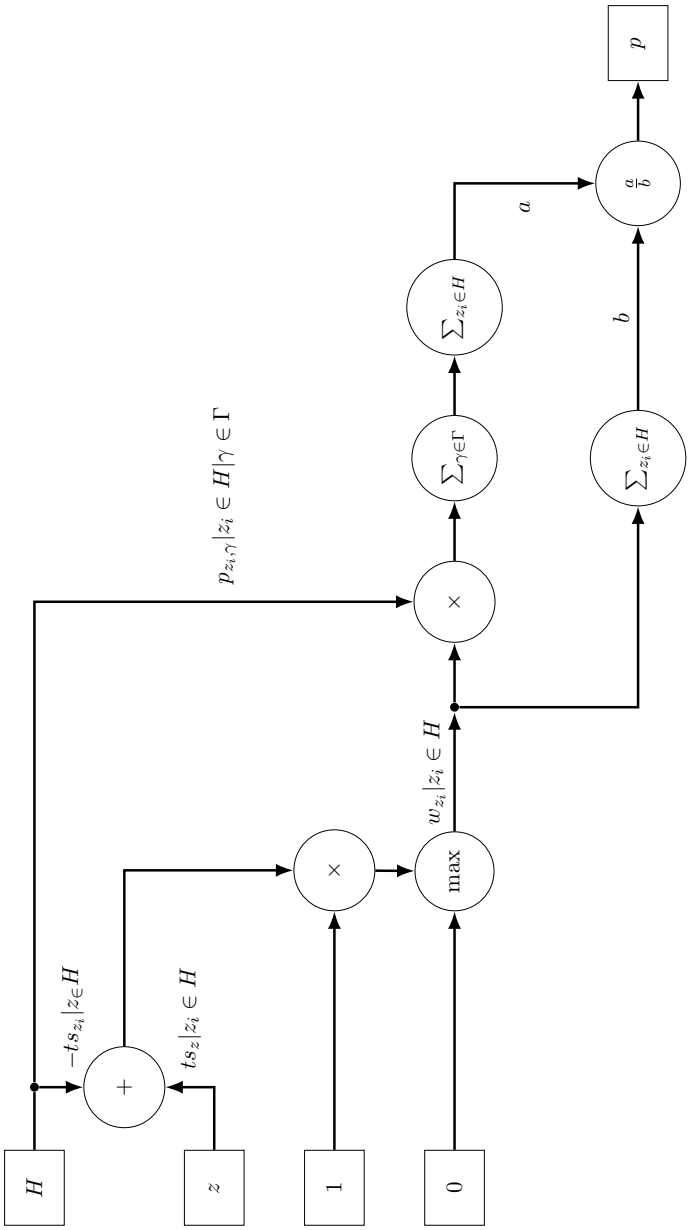


Abb. A.7.: Eine Transformationsfunktion für historische Trust-Werte modelliert als Blockdiagramm. Das Vorgehen entspricht Formel 7.4.

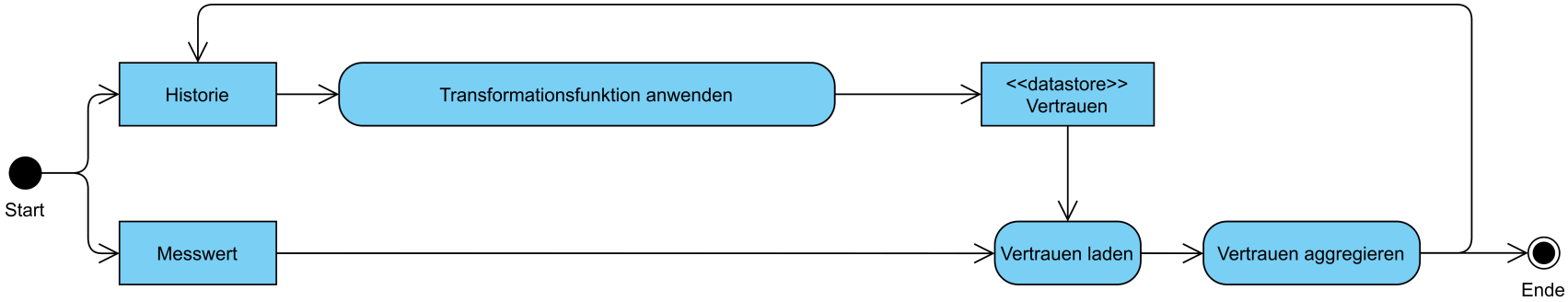


Abb. A.8.: Das UML-Aktivitätsdiagramm für einen Trust-Schätzer auf Basis historischer Trust-Werte. Es setzt dabei für die Historie eines Messwertes die Transformationsfunktion aus Abbildung A.7 um. Der neu bestimmte Trust wird im Anschluss der Historie hinzugefügt.

B Weitere Daten zum Evaluationssetup

B.1 Reduziertes CIGRE Mittelspannungsnetz mit 12 Sammelschienen

Tab. B.1.: Daten der Sammelschienen im CIGRE12MV.

Nummer	Name	G'	B'
1	Bus 00	0.0	0.0
2	Bus 01	0.0	0.0
3	Bus 02	0.0	0.0
4	Bus 03	0.0	0.0
5	Bus 04	0.0	0.0
6	Bus 05	0.0	0.0
7	Bus 06	0.0	0.0
8	Bus 07	0.0	0.0
9	Bus 08	0.0	0.0
10	Bus 09	0.0	0.0
11	Bus 10	0.0	0.0
12	Bus 11	0.0	0.0

Tab. B.2.: Daten der Verbindungen zwischen Sammelschienen im CIGRE12MV.

Quelle	Senke	Typ	circuit	R	X	C	tap node	nom.node	turnsratio
Bus 00	Bus 01	1	1	0.0064	0.4800002	0.0	Bus 00	Bus 01	1.0
Bus 01	Bus 02	0	1	0.353205	0.50478	0.00053572	Bus 01	Bus 02	0.0
Bus 02	Bus 03	0	1	0.553605	0.79118	0.00083968	Bus 02	Bus 03	0.0
Bus 03	Bus 04	0	1	0.0764025	0.10919	0.00011588	Bus 03	Bus 04	0.0
Bus 03	Bus 08	0	1	0.162825	0.2327	0.00024696	Bus 03	Bus 08	0.0
Bus 04	Bus 05	0	1	0.07014	0.10024	0.00010638	Bus 04	Bus 05	0.0
Bus 05	Bus 06	0	1	0.192885	0.27566	0.00029256	Bus 05	Bus 06	0.0
Bus 06	Bus 07	0	1	0.03006	0.04296	4.559e-05	Bus 06	Bus 07	0.0
Bus 07	Bus 08	0	1	0.2091675	0.29893	0.00031725	Bus 07	Bus 08	0.0
Bus 08	Bus 09	0	1	0.04008	0.05728	6.079e-05	Bus 08	Bus 09	0.0
Bus 09	Bus 10	0	1	0.0964425	0.13783	0.00014628	Bus 09	Bus 10	0.0
Bus 10	Bus 11	0	1	0.0413325	0.05907	6.269e-05	Bus 10	Bus 11	0.0
Bus 11	Bus 04	0	1	0.0613725	0.08771	9.309e-05	Bus 11	Bus 04	0.0

Tab. B.3.: Normale Messwerte für das CIGRE12MV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	0.99	9999	9999	9999	9999
2	9999	19.839	4.637	0.0	5.8084
3	9999	0.0	0.0	0.0	0.0
4	9999	0.502	0.209	0.1176	0.0
5	9999	0.4317	0.1082	0.1176	0.0
6	9999	0.7275	0.1823	0.1764	0.0
7	9999	0.548	0.1374	0.1764	0.0
8	9999	0.0765	0.0474	8.82	0.0
9	9999	0.5868	0.1471	0.1764	0.0
10	9999	0.5738	0.3556	0.1764	0.0
11	9999	0.543	0.161	0.2352	0.0
12	9999	0.3298	0.0827	0.0588	0.0

Tab. B.4.: Kompromittierte Messwerte für das CIGRE12MV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	0.99	9999	9999	9999	9999
2	9999	19.839	4.637	0.0	5.8084
3	9999	0.0	0.0	0.0	0.0
4	9999	0.502	0.209	0.1176	0.0
5	9999	24.2964	33.9797	-23.7471	-33.8715
6	9999	-25.2227	-37.2734	26.1266	37.4557
7	9999	0.5913	0.137	0.1331	0.0004
8	9999	-9.3129	-12.4643	18.2094	12.5117
9	9999	8.7047	11.4454	-7.9415	-11.2983
10	9999	0.5738	0.3556	0.1764	0.0
11	9999	0.543	0.161	0.2352	0.0
12	9999	0.3298	0.0827	0.0588	0.0

Tab. B.5.: Ressourcenauslastung und Prozessinformationen in der normalen Phase für das CIGRE12MV.

IP	CPU	RAM	# Prozesse
10.10.118.40	5	10	100
10.10.118.41	5	10	100
10.10.118.42	5	10	100
10.10.118.43	5	10	100
10.10.118.44	5	10	100
10.10.118.45	5	10	100
10.10.118.46	5	10	100
10.10.118.47	5	10	100
10.10.118.48	5	10	100
10.10.118.49	5	10	100
10.10.118.50	5	10	100
10.10.118.51	5	10	100

Tab. B.6.: Ressourcenauslastung und Prozessinformationen in der abnormalen Phase für das CIGRE12MV.

IP	CPU	RAM	# Prozesse
10.10.118.40	5	10	100
10.10.118.41	5	10	100
10.10.118.42	5	10	100
10.10.118.43	5	10	100
10.10.118.44	7.5	12.7	101
10.10.118.45	7.5	12.7	101
10.10.118.46	7.5	12.7	101
10.10.118.47	7.5	12.7	101
10.10.118.48	7.5	12.7	101
10.10.118.49	5	10	100
10.10.118.50	5	10	100
10.10.118.51	5	10	100

Tab. B.7.: IDS-Alarme in der abnormalen Phase für das CIGRE12MV.

IP	Schwere (-1 := kein Alarm)
10.10.118.40	-1
10.10.118.41	-1
10.10.118.42	-1
10.10.118.43	-1
10.10.118.44	0
10.10.118.45	0
10.10.118.46	0
10.10.118.47	0
10.10.118.48	0
10.10.118.49	-1
10.10.118.50	-1
10.10.118.51	-1

Tab. B.8.: Normale State-Estimation-Ergebnisse für das CIGRE12MV.

Nummer	Name	Typ	Schätzung
1	Bus 00	PhaseToPhaseVoltage	0.9899
1	Bus 00	PMU	0.0
2	Bus 01	PhaseToPhaseVoltage	0.9837
2	Bus 01	PMU	-4.099
3	Bus 02	PhaseToPhaseVoltage	0.9945
3	Bus 02	PMU	-2.1614
4	Bus 03	PhaseToPhaseVoltage	1.0131
4	Bus 03	PMU	0.8102
5	Bus 04	PhaseToPhaseVoltage	1.0148
5	Bus 04	PMU	1.0434
6	Bus 05	PhaseToPhaseVoltage	1.0168
6	Bus 05	PMU	1.2404
7	Bus 06	PhaseToPhaseVoltage	1.0238
7	Bus 06	PMU	1.844
8	Bus 07	PhaseToPhaseVoltage	1.0251
8	Bus 07	PMU	1.944
9	Bus 08	PhaseToPhaseVoltage	1.0162
9	Bus 08	PMU	1.2046
10	Bus 09	PhaseToPhaseVoltage	1.0156
10	Bus 09	PMU	1.1688
11	Bus 10	PhaseToPhaseVoltage	1.0148
11	Bus 10	PMU	1.0948
12	Bus 11	PhaseToPhaseVoltage	1.0147
12	Bus 11	PMU	1.0698

Tab. B.9.: Kompromittierte State-Estimation-Ergebnisse für das CIGRE12MV.

Nummer	Name	Typ	Schätzung
1	Bus 00	PhaseToPhaseVoltage	0.9899
1	Bus 00	PMU	0.0
2	Bus 01	PhaseToPhaseVoltage	0.9839
2	Bus 01	PMU	-4.1225
3	Bus 02	PhaseToPhaseVoltage	0.9946
3	Bus 02	PMU	-2.2197
4	Bus 03	PhaseToPhaseVoltage	1.0128
4	Bus 03	PMU	0.7
5	Bus 04	PhaseToPhaseVoltage	1.0145
5	Bus 04	PMU	0.9288
6	Bus 05	PhaseToPhaseVoltage	1.1164
6	Bus 05	PMU	1.1217
7	Bus 06	PhaseToPhaseVoltage	1.1234
7	Bus 06	PMU	1.7211
8	Bus 07	PhaseToPhaseVoltage	1.1247
8	Bus 07	PMU	1.8212
9	Bus 08	PhaseToPhaseVoltage	1.016
9	Bus 08	PMU	1.0893
10	Bus 09	PhaseToPhaseVoltage	1.0153
10	Bus 09	PMU	1.0537
11	Bus 10	PhaseToPhaseVoltage	1.0146
11	Bus 10	PMU	0.9799
12	Bus 11	PhaseToPhaseVoltage	1.0145
12	Bus 11	PMU	0.955

B.2 IEEE Hochspannungsnetz mit 39 Sammelschienen

Topologische Daten stammen von dem Illinois Center for a Smarter Electric Grid[1].

[1] `https://icseg.iti.illinois.edu/ieee-39-bus-system/`

Tab. B.10.: Normale Messwerte für das IEEE39HV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	9999	0.0	0.0	0.0	0.0
2	9999	0.0	0.0	0.0	0.0
3	9999	322.0	2.4	0.0	0.0
4	9999	500.0	184.0	0.0	0.0
5	9999	0.0	0.0	0.0	0.0
6	9999	0.0	0.0	0.0	0.0
7	9999	233.8	84.0	0.0	0.0
8	9999	522.0	176.0	0.0	0.0
9	9999	0.0	0.0	0.0	0.0
10	9999	0.0	0.0	0.0	0.0
11	9999	0.0	0.0	0.0	0.0
12	9999	7.5	88.0	0.0	0.0
13	9999	0.0	0.0	0.0	0.0
14	9999	0.0	0.0	0.0	0.0
15	9999	320.0	153.0	0.0	0.0
16	9999	329.0	32.3	0.0	0.0
17	9999	0.0	0.0	0.0	0.0
18	9999	158.0	30.0	0.0	0.0
19	9999	0.0	0.0	0.0	0.0
20	9999	628.0	103.0	0.0	0.0
21	9999	274.0	115.0	0.0	0.0
22	9999	0.0	0.0	0.0	0.0
23	9999	247.5	84.6	0.0	0.0
24	9999	308.6	-92.2	0.0	0.0
25	9999	224.0	47.2	0.0	0.0
26	9999	139.0	17.0	0.0	0.0
27	9999	281.0	75.5	0.0	0.0
28	9999	206.0	27.6	0.0	0.0
29	9999	283.5	26.9	0.0	0.0
30	1.0475	0.0	9999	250.0	9999
31	0.982	9999	9999	9999	9999
32	0.9831	0.0	9999	650.0	9999
33	0.9972	0.0	9999	632.0	9999
34	1.0123	0.0	9999	508.0	9999
35	1.0493	0.0	9999	650.0	9999
36	1.0635	0.0	9999	560.0	9999
37	1.0278	0.0	9999	540.0	9999
38	1.0265	0.0	9999	830.0	9999
39	1.03	1104.0	9999	1000.0	9999

Tab. B.11.: Kompromittierte Messwerte für das IEEE39HV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	9999	0.0	0.0	0.0	0.0
2	9999	0.0	0.0	0.0	0.0
3	9999	322.0	2.4	0.0	0.0
4	9999	500.0	184.0	0.0	0.0
5	9999	0.0	0.0	0.0	0.0
6	9999	0.0	0.0	0.0	0.0
7	9999	233.8	84.0	0.0	0.0
8	9999	522.0	176.0	0.0	0.0
9	9999	0.0	0.0	0.0	0.0
10	9999	0.0	0.0	0.0	0.0
11	9999	0.0	0.0	0.0	0.0
12	9999	7.5	88.0	0.0	0.0
13	9999	0.0	0.0	0.0	0.0
14	9999	0.0	0.0	0.0	0.0
15	9999	320.0	153.0	0.0	0.0
16	9999	329.0	32.3	0.0	0.0
17	9999	0.0	0.0	0.0	0.0
18	9999	158.0	30.0	0.0	0.0
19	9999	0.0	0.0	0.0	0.0
20	9999	628.0	103.0	0.0	0.0
21	9999	274.0	115.0	0.0	0.0
22	9999	0.0	0.0	0.0	0.0
23	9999	247.5	84.6	0.0	0.0
24	9999	308.6	-92.2	0.0	0.0
25	9999	224.0	47.2	0.0	0.0
26	9999	172.3725	208.2122	-33.3725	-191.2122
27	9999	281.0	75.5	0.0	0.0
28	9999	223.0402	-90.5272	-17.0402	118.1272
29	9999	314.8734	-54.3444	-31.3734	81.2444
30	1.0475	0.0	9999	250.0	9999
31	0.982	9999	9999	9999	9999
32	0.9831	0.0	9999	650.0	9999
33	0.9972	0.0	9999	632.0	9999
34	1.0123	0.0	9999	508.0	9999
35	1.0493	0.0	9999	650.0	9999
36	1.0635	0.0	9999	560.0	9999
37	1.0278	0.0	9999	540.0	9999
38	1.1265	-84.226	9999	914.226	9999
39	1.03	1104.0	9999	1000.0	9999

Tab. B.12.: Ressourcenauslastung und Prozessinformationen in der normalen Phase für das IEEE39HV.

IP	CPU	RAM	# Prozesse
10.10.118.40	5	10	100
10.10.118.41	5	10	100
10.10.118.42	5	10	100
10.10.118.43	5	10	100
10.10.118.44	5	10	100
10.10.118.45	5	10	100
10.10.118.46	5	10	100
10.10.118.47	5	10	100
10.10.118.48	5	10	100
10.10.118.49	5	10	100
10.10.118.50	5	10	100
10.10.118.51	5	10	100
10.10.118.52	5	10	100
10.10.118.53	5	10	100
10.10.118.54	5	10	100
10.10.118.55	5	10	100
10.10.118.56	5	10	100
10.10.118.57	5	10	100
10.10.118.58	5	10	100
10.10.118.59	5	10	100
10.10.118.60	5	10	100
10.10.118.61	5	10	100
10.10.118.62	5	10	100
10.10.118.63	5	10	100
10.10.118.64	5	10	100
10.10.118.65	5	10	100
10.10.118.66	5	10	100
10.10.118.67	5	10	100
10.10.118.68	5	10	100
10.10.118.69	5	10	100
10.10.118.70	5	10	100
10.10.118.71	5	10	100
10.10.118.72	5	10	100
10.10.118.73	5	10	100
10.10.118.74	5	10	100
10.10.118.75	5	10	100
10.10.118.76	5	10	100
10.10.118.77	5	10	100
10.10.118.78	5	10	100

Tab. B.13.: Ressourcenauslastung und Prozessinformationen in der abnormalen Phase für das IEEE39HV.

IP	CPU	RAM	# Prozesse
10.10.118.40	5	10	100
10.10.118.41	5	10	100
10.10.118.42	5	10	100
10.10.118.43	5	10	100
10.10.118.44	5	10	100
10.10.118.45	5	10	100
10.10.118.46	5	10	100
10.10.118.47	5	10	100
10.10.118.48	5	10	100
10.10.118.49	5	10	100
10.10.118.50	5	10	100
10.10.118.51	5	10	100
10.10.118.52	5	10	100
10.10.118.53	5	10	100
10.10.118.54	5	10	100
10.10.118.55	5	10	100
10.10.118.56	5	10	100
10.10.118.57	5	10	100
10.10.118.58	5	10	100
10.10.118.59	5	10	100
10.10.118.60	5	10	100
10.10.118.61	5	10	100
10.10.118.62	5	10	100
10.10.118.63	5	10	100
10.10.118.64	5	10	100
10.10.118.65	7.5	12.7	101
10.10.118.66	5	10	100
10.10.118.67	7.5	12.7	101
10.10.118.68	7.5	12.7	101
10.10.118.69	5	10	100
10.10.118.70	5	10	100
10.10.118.71	5	10	100
10.10.118.72	5	10	100
10.10.118.73	5	10	100
10.10.118.74	5	10	100
10.10.118.75	5	10	100
10.10.118.76	5	10	100
10.10.118.77	7.5	12.7	101
10.10.118.78	5	10	100

Tab. B.14.: Normale State-Estimation-Ergebnisse für das IEEE39HV.

Nummer	Name	Typ	Schätzung
1	Bus_01	PhaseToPhaseVoltage	1.0472
1	Bus_01	PMU	-8.5035
2	Bus_02	PhaseToPhaseVoltage	1.0482
2	Bus_02	PMU	-5.8204
3	Bus_03	PhaseToPhaseVoltage	1.0291
3	Bus_03	PMU	-8.6675
4	Bus_04	PhaseToPhaseVoltage	1.0029
4	Bus_04	PMU	-9.6671
5	Bus_05	PhaseToPhaseVoltage	1.0046
5	Bus_05	PMU	-8.6613
6	Bus_06	PhaseToPhaseVoltage	1.007
6	Bus_06	PMU	-7.9963
7	Bus_07	PhaseToPhaseVoltage	0.9963
7	Bus_07	PMU	-10.1748
8	Bus_08	PhaseToPhaseVoltage	0.9954
8	Bus_08	PMU	-10.6678
9	Bus_09	PhaseToPhaseVoltage	1.028
9	Bus_09	PMU	-10.3805
10	Bus_10	PhaseToPhaseVoltage	1.0164
10	Bus_10	PMU	-5.4767
11	Bus_11	PhaseToPhaseVoltage	1.012
11	Bus_11	PMU	-6.333
12	Bus_12	PhaseToPhaseVoltage	0.9993
12	Bus_12	PMU	-6.2941
13	Bus_13	PhaseToPhaseVoltage	1.0134
13	Bus_13	PMU	-6.1497
14	Bus_14	PhaseToPhaseVoltage	1.0105
14	Bus_14	PMU	-7.7153
15	Bus_15	PhaseToPhaseVoltage	1.0131
15	Bus_15	PMU	-7.8077
16	Bus_16	PhaseToPhaseVoltage	1.0291
16	Bus_16	PMU	-6.2566
17	Bus_17	PhaseToPhaseVoltage	1.0315
17	Bus_17	PMU	-7.3729
18	Bus_18	PhaseToPhaseVoltage	1.0292
18	Bus_18	PMU	-8.2962
19	Bus_19	PhaseToPhaseVoltage	1.043
19	Bus_19	PMU	-1.0514
20	Bus_20	PhaseToPhaseVoltage	0.9755
20	Bus_20	PMU	-2.0587

Tab. B.15.: Normale State-Estimation-Ergebnisse für das IEEE39HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
21	Bus_21	PhaseToPhaseVoltage	1.0299
21	Bus_21	PMU	-3.8418
22	Bus_22	PhaseToPhaseVoltage	1.0488
22	Bus_22	PMU	0.6168
23	Bus_23	PhaseToPhaseVoltage	1.0437
23	Bus_23	PMU	0.4182
24	Bus_24	PhaseToPhaseVoltage	1.0348
24	Bus_24	PMU	-6.1371
25	Bus_25	PhaseToPhaseVoltage	1.0571
25	Bus_25	PMU	-4.4313
26	Bus_26	PhaseToPhaseVoltage	1.0511
26	Bus_26	PMU	-5.5965
27	Bus_27	PhaseToPhaseVoltage	1.0363
27	Bus_27	PMU	-7.5686
28	Bus_28	PhaseToPhaseVoltage	1.0496
28	Bus_28	PMU	-2.0819
29	Bus_29	PhaseToPhaseVoltage	1.0496
29	Bus_29	PMU	0.6789
30	Bus_30	PhaseToPhaseVoltage	1.0475
30	Bus_30	PMU	-3.3995
31	Bus_31	PhaseToPhaseVoltage	0.982
31	Bus_31	PMU	0.0
32	Bus_32	PhaseToPhaseVoltage	0.9831
32	Bus_32	PMU	2.525
33	Bus_33	PhaseToPhaseVoltage	0.9972
33	Bus_33	PMU	4.1818
34	Bus_34	PhaseToPhaseVoltage	1.0123
34	Bus_34	PMU	8.5512
35	Bus_35	PhaseToPhaseVoltage	1.0493
35	Bus_35	PMU	5.5834
36	Bus_36	PhaseToPhaseVoltage	1.0635
36	Bus_36	PMU	8.278
37	Bus_37	PhaseToPhaseVoltage	1.0278
37	Bus_37	PMU	2.3557
38	Bus_38	PhaseToPhaseVoltage	1.0265
38	Bus_38	PMU	7.7436
39	Bus_39	PhaseToPhaseVoltage	1.03
39	Bus_39	PMU	-10.1153

Tab. B.16.: Kompromittierte State-Estimation-Ergebnisse für das IEEE39HV.

Nummer	Name	Typ	Schätzung
1	Bus_01	PhaseToPhaseVoltage	1.0472
1	Bus_01	PMU	-8.5037
2	Bus_02	PhaseToPhaseVoltage	1.0482
2	Bus_02	PMU	-5.8207
3	Bus_03	PhaseToPhaseVoltage	1.0291
3	Bus_03	PMU	-8.6677
4	Bus_04	PhaseToPhaseVoltage	1.0029
4	Bus_04	PMU	-9.6673
5	Bus_05	PhaseToPhaseVoltage	1.0046
5	Bus_05	PMU	-8.6615
6	Bus_06	PhaseToPhaseVoltage	1.007
6	Bus_06	PMU	-7.9965
7	Bus_07	PhaseToPhaseVoltage	0.9963
7	Bus_07	PMU	-10.175
8	Bus_08	PhaseToPhaseVoltage	0.9954
8	Bus_08	PMU	-10.6679
9	Bus_09	PhaseToPhaseVoltage	1.028
9	Bus_09	PMU	-10.3806
10	Bus_10	PhaseToPhaseVoltage	1.0164
10	Bus_10	PMU	-5.4768
11	Bus_11	PhaseToPhaseVoltage	1.012
11	Bus_11	PMU	-6.3331
12	Bus_12	PhaseToPhaseVoltage	0.9994
12	Bus_12	PMU	-6.2942
13	Bus_13	PhaseToPhaseVoltage	1.0134
13	Bus_13	PMU	-6.1499
14	Bus_14	PhaseToPhaseVoltage	1.0105
14	Bus_14	PMU	-7.7155
15	Bus_15	PhaseToPhaseVoltage	1.0131
15	Bus_15	PMU	-7.8079
16	Bus_16	PhaseToPhaseVoltage	1.0291
16	Bus_16	PMU	-6.2568
17	Bus_17	PhaseToPhaseVoltage	1.0315
17	Bus_17	PMU	-7.3732
18	Bus_18	PhaseToPhaseVoltage	1.0292
18	Bus_18	PMU	-8.2964
19	Bus_19	PhaseToPhaseVoltage	1.043
19	Bus_19	PMU	-1.0516
20	Bus_20	PhaseToPhaseVoltage	0.9755
20	Bus_20	PMU	-2.059

Tab. B.17.: Kompromittierte State-Estimation-Ergebnisse für das IEEE39HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
21	Bus_21	PhaseToPhaseVoltage	1.0299
21	Bus_21	PMU	-3.842
22	Bus_22	PhaseToPhaseVoltage	1.0488
22	Bus_22	PMU	0.6166
23	Bus_23	PhaseToPhaseVoltage	1.0437
23	Bus_23	PMU	0.418
24	Bus_24	PhaseToPhaseVoltage	1.0348
24	Bus_24	PMU	-6.1373
25	Bus_25	PhaseToPhaseVoltage	1.0571
25	Bus_25	PMU	-4.4315
26	Bus_26	PhaseToPhaseVoltage	1.0511
26	Bus_26	PMU	-5.5969
27	Bus_27	PhaseToPhaseVoltage	1.0363
27	Bus_27	PMU	-7.5689
28	Bus_28	PhaseToPhaseVoltage	1.1496
28	Bus_28	PMU	-2.0823
29	Bus_29	PhaseToPhaseVoltage	1.1496
29	Bus_29	PMU	0.6784
30	Bus_30	PhaseToPhaseVoltage	1.0475
30	Bus_30	PMU	-3.3998
31	Bus_31	PhaseToPhaseVoltage	0.982
31	Bus_31	PMU	0.0
32	Bus_32	PhaseToPhaseVoltage	0.9831
32	Bus_32	PMU	2.5249
33	Bus_33	PhaseToPhaseVoltage	0.9972
33	Bus_33	PMU	4.1816
34	Bus_34	PhaseToPhaseVoltage	1.0123
34	Bus_34	PMU	8.551
35	Bus_35	PhaseToPhaseVoltage	1.0493
35	Bus_35	PMU	5.5831
36	Bus_36	PhaseToPhaseVoltage	1.0635
36	Bus_36	PMU	8.2777
37	Bus_37	PhaseToPhaseVoltage	1.0278
37	Bus_37	PMU	2.3554
38	Bus_38	PhaseToPhaseVoltage	1.1265
38	Bus_38	PMU	7.7432
39	Bus_39	PhaseToPhaseVoltage	1.03
39	Bus_39	PMU	-10.1155

B.3 IEEE Hochspannungsnetz mit 118 Sammelschienen

Topologische Daten stammen von dem Illinois Center for a Smarter Electric Grid[2].

Tab. B.18.: Normale Messwerte für das IEEE118HV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	0.955	51.0	9999	0.0	9999
2	9999	20.0	9.0	0.0	0.0
3	9999	39.0	10.0	0.0	0.0
4	0.998	30.0	9999	-9.0	9999
5	9999	0.0	0.0	0.0	0.0
6	0.99	52.0	9999	0.0	9999
7	9999	19.0	2.0	0.0	0.0
8	1.015	0.0	9999	-28.0	9999
9	9999	0.0	0.0	0.0	0.0
10	1.05	0.0	9999	450.0	9999
11	9999	70.0	23.0	0.0	0.0
12	0.99	47.0	9999	85.0	9999
13	9999	34.0	16.0	0.0	0.0
14	9999	14.0	1.0	0.0	0.0
15	0.97	90.0	9999	0.0	9999
16	9999	25.0	10.0	0.0	0.0
17	9999	11.0	3.0	0.0	0.0
18	0.973	60.0	9999	0.0	9999
19	0.963	45.0	9999	0.0	9999
20	9999	18.0	3.0	0.0	0.0
21	9999	14.0	8.0	0.0	0.0
22	9999	10.0	5.0	0.0	0.0
23	9999	7.0	3.0	0.0	0.0
24	0.992	0.0	9999	-13.0	9999
25	1.05	0.0	9999	220.0	9999
26	1.015	0.0	9999	314.0	9999
27	0.968	62.0	9999	-9.0	9999
28	9999	17.0	7.0	0.0	0.0
29	9999	24.0	4.0	0.0	0.0
30	9999	0.0	0.0	0.0	0.0
31	0.967	43.0	9999	7.0	9999
32	0.964	59.0	9999	0.0	9999
33	9999	23.0	9.0	0.0	0.0
34	0.986	59.0	9999	0.0	9999

[2] `https://icseg.iti.illinois.edu/ieee-118-bus-system/`

Tab. B.19.: Normale Messwerte für das IEEE118HV (Fortsetzung).

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
35	9999	33.0	9.0	0.0	0.0
36	0.98	31.0	9999	0.0	9999
37	9999	0.0	0.0	0.0	0.0
38	9999	0.0	0.0	0.0	0.0
39	9999	27.0	11.0	0.0	0.0
40	0.97	20.0	9999	-46.0	9999
41	9999	37.0	10.0	0.0	0.0
42	0.985	37.0	9999	-59.0	9999
43	9999	18.0	7.0	0.0	0.0
44	9999	16.0	8.0	0.0	0.0
45	9999	53.0	22.0	0.0	0.0
46	1.005	28.0	9999	19.0	9999
47	9999	34.0	0.0	0.0	0.0
48	9999	20.0	11.0	0.0	0.0
49	1.025	87.0	9999	204.0	9999
50	9999	17.0	4.0	0.0	0.0
51	9999	17.0	8.0	0.0	0.0
52	9999	18.0	5.0	0.0	0.0
53	9999	23.0	11.0	0.0	0.0
54	0.955	113.0	9999	48.0	9999
55	0.952	63.0	9999	0.0	9999
56	0.954	84.0	9999	0.0	9999
57	9999	12.0	3.0	0.0	0.0
58	9999	12.0	3.0	0.0	0.0
59	0.985	277.0	9999	155.0	9999
60	9999	78.0	3.0	0.0	0.0
61	0.995	0.0	9999	160.0	9999
62	0.998	77.0	9999	0.0	9999
63	9999	0.0	0.0	0.0	0.0
64	9999	0.0	0.0	0.0	0.0
65	1.005	0.0	9999	391.0	9999
66	1.05	39.0	9999	392.0	9999
67	9999	28.0	7.0	0.0	0.0
68	9999	0.0	0.0	0.0	0.0
69	1.035	9999	9999	9999	9999
70	0.984	66.0	9999	0.0	9999

Tab. B.20.: Normale Messwerte für das IEEE118HV (Fortsetzung).

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
71	9999	0.0	0.0	0.0	0.0
72	0.98	0.0	9999	-12.0	9999
73	0.991	0.0	9999	-6.0	9999
74	0.958	68.0	9999	0.0	9999
75	9999	47.0	11.0	0.0	0.0
76	0.943	68.0	9999	0.0	9999
77	1.006	61.0	9999	0.0	9999
78	9999	71.0	26.0	0.0	0.0
79	9999	39.0	32.0	0.0	0.0
80	1.04	130.0	9999	477.0	9999
81	9999	0.0	0.0	0.0	0.0
82	9999	54.0	27.0	0.0	0.0
83	9999	20.0	10.0	0.0	0.0
84	9999	11.0	7.0	0.0	0.0
85	0.985	24.0	9999	0.0	9999
86	9999	21.0	10.0	0.0	0.0
87	1.015	0.0	9999	4.0	9999
88	9999	48.0	10.0	0.0	0.0
89	1.005	0.0	9999	607.0	9999
90	0.985	78.0	9999	-85.0	9999
91	0.98	0.0	9999	-10.0	9999
92	0.993	65.0	9999	0.0	9999
93	9999	12.0	7.0	0.0	0.0
94	9999	30.0	16.0	0.0	0.0
95	9999	42.0	31.0	0.0	0.0
96	9999	38.0	15.0	0.0	0.0
97	9999	15.0	9.0	0.0	0.0
98	9999	34.0	8.0	0.0	0.0
99	1.01	0.0	9999	-42.0	9999
100	1.017	37.0	9999	252.0	9999
101	9999	22.0	15.0	0.0	0.0
102	9999	5.0	3.0	0.0	0.0
103	1.001	23.0	9999	40.0	9999
104	0.971	38.0	9999	0.0	9999
105	0.965	31.0	9999	0.0	9999
106	9999	43.0	16.0	0.0	0.0
107	0.952	28.0	9999	-22.0	9999
108	9999	2.0	1.0	0.0	0.0
109	9999	8.0	3.0	0.0	0.0
110	0.973	39.0	9999	0.0	9999
111	0.98	0.0	9999	36.0	9999

Tab. B.21.: Normale Messwerte für das IEEE118HV (Fortsetzung).

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
112	0.975	25.0	9999	-43.0	9999
113	0.993	0.0	9999	-6.0	9999
114	9999	8.0	3.0	0.0	0.0
115	9999	22.0	7.0	0.0	0.0
116	1.005	0.0	9999	-184.0	9999
117	9999	20.0	8.0	0.0	0.0
118	9999	33.0	15.0	0.0	0.0

Tab. B.22.: Kompromittierte Messwerte für das IEEE118HV.

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
1	0.955	51.0	9999	0.0	9999
2	9999	20.0	9.0	0.0	0.0
3	9999	39.0	10.0	0.0	0.0
4	0.998	30.0	9999	-9.0	9999
5	9999	0.0	0.0	0.0	0.0
6	0.99	52.0	9999	0.0	9999
7	9999	19.0	2.0	0.0	0.0
8	1.015	0.0	9999	-28.0	9999
9	9999	0.0	0.0	0.0	0.0
10	1.05	0.0	9999	450.0	9999
11	9999	70.0	23.0	0.0	0.0
12	0.99	47.0	9999	85.0	9999
13	9999	34.0	16.0	0.0	0.0
14	9999	14.0	1.0	0.0	0.0
15	0.97	90.0	9999	0.0	9999
16	9999	25.0	10.0	0.0	0.0
17	9999	11.0	3.0	0.0	0.0
18	0.973	60.0	9999	0.0	9999
19	0.963	45.0	9999	0.0	9999
20	9999	18.0	3.0	0.0	0.0
21	9999	14.0	8.0	0.0	0.0
22	9999	10.0	5.0	0.0	0.0
23	9999	7.0	3.0	0.0	0.0
24	0.992	4.4	9999	-17.4	9999
25	1.05	0.0	9999	220.0	9999
26	1.015	0.0	9999	314.0	9999
27	0.968	62.0	9999	-9.0	9999
28	9999	17.0	7.0	0.0	0.0
29	9999	24.0	4.0	0.0	0.0
30	9999	0.0	0.0	0.0	0.0
31	0.967	43.0	9999	7.0	9999
32	0.964	59.0	9999	0.0	9999
33	9999	23.0	9.0	0.0	0.0
34	0.986	59.0	9999	0.0	9999
35	9999	33.0	9.0	0.0	0.0
36	0.98	31.0	9999	0.0	9999
37	9999	0.0	0.0	0.0	0.0
38	9999	0.0	0.0	0.0	0.0
39	9999	27.0	11.0	0.0	0.0
40	0.97	20.0	9999	-46.0	9999

Tab. B.23.: Kompromittierte Messwerte für das IEEE118HV (Fortsetzung).

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
41	9999	37.0	10.0	0.0	0.0
42	0.985	37.0	9999	-59.0	9999
43	9999	18.0	7.0	0.0	0.0
44	9999	16.0	8.0	0.0	0.0
45	9999	53.0	22.0	0.0	0.0
46	1.005	28.0	9999	19.0	9999
47	9999	34.0	0.0	0.0	0.0
48	9999	20.0	11.0	0.0	0.0
49	1.025	87.0	9999	204.0	9999
50	9999	17.0	4.0	0.0	0.0
51	9999	17.0	8.0	0.0	0.0
52	9999	18.0	5.0	0.0	0.0
53	9999	23.0	11.0	0.0	0.0
54	0.955	113.0	9999	48.0	9999
55	0.952	63.0	9999	0.0	9999
56	0.954	84.0	9999	0.0	9999
57	9999	12.0	3.0	0.0	0.0
58	9999	12.0	3.0	0.0	0.0
59	0.985	277.0	9999	155.0	9999
60	9999	78.0	3.0	0.0	0.0
61	0.995	0.0	9999	160.0	9999
62	0.998	77.0	9999	0.0	9999
63	9999	0.0	0.0	0.0	0.0
64	9999	0.0	0.0	0.0	0.0
65	1.005	0.0	9999	391.0	9999
66	1.05	39.0	9999	392.0	9999
67	9999	28.0	7.0	0.0	0.0
68	9999	0.0	0.0	0.0	0.0
69	1.035	9999	9999	9999	9999
70	0.984	99.4	9999	-33.4	9999
71	9999	-36.8	-143.8	36.8	143.8
72	1.08	-6.6	9999	-5.4	9999
73	1.09	3.2	9999	-9.2	9999
74	0.958	68.0	9999	0.0	9999
75	9999	47.0	11.0	0.0	0.0
76	0.943	68.0	9999	0.0	9999
77	1.006	61.0	9999	0.0	9999
78	9999	71.0	26.0	0.0	0.0
79	9999	39.0	32.0	0.0	0.0
80	1.04	130.0	9999	477.0	9999

Tab. B.24.: Kompromittierte Messwerte für das IEEE118HV (Fortsetzung).

Nummer	V	P_{load}	Q_{load}	P_{gen}	Q_{gen}
81	9999	0.0	0.0	0.0	0.0
82	9999	54.0	27.0	0.0	0.0
83	9999	20.0	10.0	0.0	0.0
84	9999	11.0	7.0	0.0	0.0
85	0.985	24.0	9999	0.0	9999
86	9999	21.0	10.0	0.0	0.0
87	1.015	0.0	9999	4.0	9999
88	9999	48.0	10.0	0.0	0.0
89	1.005	0.0	9999	607.0	9999
90	0.985	78.0	9999	-85.0	9999
91	0.98	0.0	9999	-10.0	9999
92	0.993	65.0	9999	0.0	9999
93	9999	12.0	7.0	0.0	0.0
94	9999	30.0	16.0	0.0	0.0
95	9999	42.0	31.0	0.0	0.0
96	9999	38.0	15.0	0.0	0.0
97	9999	15.0	9.0	0.0	0.0
98	9999	34.0	8.0	0.0	0.0
99	1.01	0.0	9999	-42.0	9999
100	1.017	37.0	9999	252.0	9999
101	9999	22.0	15.0	0.0	0.0
102	9999	5.0	3.0	0.0	0.0
103	1.001	23.0	9999	40.0	9999
104	0.971	38.0	9999	0.0	9999
105	0.965	31.0	9999	0.0	9999
106	9999	43.0	16.0	0.0	0.0
107	0.952	28.0	9999	-22.0	9999
108	9999	2.0	1.0	0.0	0.0
109	9999	8.0	3.0	0.0	0.0
110	0.973	39.0	9999	0.0	9999
111	0.98	0.0	9999	36.0	9999
112	0.975	25.0	9999	-43.0	9999
113	0.993	0.0	9999	-6.0	9999
114	9999	8.0	3.0	0.0	0.0
115	9999	22.0	7.0	0.0	0.0
116	1.005	0.0	9999	-184.0	9999
117	9999	20.0	8.0	0.0	0.0
118	9999	33.0	15.0	0.0	0.0

Tab. B.25.: IDS-Alarme in der abnormalen Phase für das IEEE118HV.

IP	Schwere (-1 := kein Alarm)
10.10.118.40	-1
10.10.118.41	-1
10.10.118.42	-1
10.10.118.43	-1
10.10.118.44	-1
10.10.118.45	-1
10.10.118.46	-1
10.10.118.47	-1
10.10.118.48	-1
10.10.118.49	-1
10.10.118.50	-1
10.10.118.51	-1
10.10.118.52	-1
10.10.118.53	-1
10.10.118.54	-1
10.10.118.55	-1
10.10.118.56	-1
10.10.118.57	-1
10.10.118.58	-1
10.10.118.59	-1
10.10.118.60	-1
10.10.118.61	-1
10.10.118.62	-1
10.10.118.63	-1
10.10.118.64	-1
10.10.118.65	-1
10.10.118.66	-1
10.10.118.67	-1
10.10.118.68	-1
10.10.118.69	-1
10.10.118.70	-1
10.10.118.71	-1
10.10.118.72	-1
10.10.118.73	-1
10.10.118.74	-1
10.10.118.75	-1
10.10.118.76	-1
10.10.118.77	-1
10.10.118.78	-1
10.10.118.79	-1
10.10.118.80	-1

Tab. B.26.: IDS-Alarme in der abnormalen Phase für das IEEE118HV (Fortsetzung).

IP	Schwere (-1 := kein Alarm)
10.10.118.81	-1
10.10.118.82	-1
10.10.118.83	-1
10.10.118.84	-1
10.10.118.85	-1
10.10.118.86	-1
10.10.118.87	-1
10.10.118.88	-1
10.10.118.89	-1
10.10.118.90	-1
10.10.118.91	-1
10.10.118.92	-1
10.10.118.93	-1
10.10.118.94	-1
10.10.118.95	-1
10.10.118.96	-1
10.10.118.97	-1
10.10.118.98	-1
10.10.118.99	-1
10.10.118.10	-1
10.10.118.101	-1
10.10.118.102	-1
10.10.118.103	-1
10.10.118.104	-1
10.10.118.105	-1
10.10.118.106	-1
10.10.118.107	-1
10.10.118.108	-1
10.10.118.109	-1
10.10.118.110	0
10.10.118.111	0
10.10.118.112	0
10.10.118.113	-1
10.10.118.114	-1
10.10.118.115	-1
10.10.118.116	-1
10.10.118.117	-1
10.10.118.118	-1
10.10.118.119	-1
10.10.118.120	-1

Tab. B.27.: IDS-Alarme in der abnormalen Phase für das IEEE118HV (Fortsetzung).

IP	Schwere (-1 := kein Alarm)
10.10.118.121	-1
10.10.118.122	-1
10.10.118.123	-1
10.10.118.124	-1
10.10.118.125	-1
10.10.118.126	-1
10.10.118.127	-1
10.10.118.128	-1
10.10.118.129	-1
10.10.118.130	-1
10.10.118.131	-1
10.10.118.132	-1
10.10.118.133	-1
10.10.118.134	-1
10.10.118.135	-1
10.10.118.136	-1
10.10.118.137	-1
10.10.118.138	-1
10.10.118.139	-1
10.10.118.140	-1
10.10.118.141	-1
10.10.118.142	-1
10.10.118.143	-1
10.10.118.144	-1
10.10.118.145	-1
10.10.118.146	-1
10.10.118.147	-1
10.10.118.148	-1
10.10.118.149	-1
10.10.118.150	-1
10.10.118.151	-1
10.10.118.152	-1
10.10.118.153	-1
10.10.118.154	-1
10.10.118.155	-1
10.10.118.156	-1
10.10.118.157	-1

Tab. B.28.: Normale State-Estimation-Ergebnisse für das IEEE118HV.

Nummer	Name	Typ	Schätzung
1	Riversde V2	PhaseToPhaseVoltage	0.9551
1	Riversde V2	PMU	-19.0268
2	Pokagon V2	PhaseToPhaseVoltage	0.9714
2	Pokagon V2	PMU	-18.486
3	HickryCk V2	PhaseToPhaseVoltage	0.9677
3	HickryCk V2	PMU	-18.1433
4	NwCarlsl V2	PhaseToPhaseVoltage	0.9981
4	NwCarlsl V2	PMU	-14.4264
5	Olive V2	PhaseToPhaseVoltage	1.002
5	Olive V2	PMU	-13.9813
6	Kankakee V2	PhaseToPhaseVoltage	0.9901
6	Kankakee V2	PMU	-16.7081
7	JacksnRd V2	PhaseToPhaseVoltage	0.9894
7	JacksnRd V2	PMU	-17.1518
8	Olive V1	PhaseToPhaseVoltage	1.015
8	Olive V1	PMU	-8.9611
9	Bequine V1	PhaseToPhaseVoltage	1.0429
9	Bequine V1	PMU	-1.7073
10	Breed V1	PhaseToPhaseVoltage	1.0499
10	Breed V1	PMU	5.8739
11	SouthBnd V2	PhaseToPhaseVoltage	0.9851
11	SouthBnd V2	PMU	-16.9928
12	TwinBrch V2	PhaseToPhaseVoltage	0.99
12	TwinBrch V2	PMU	-17.5089
13	Concord V2	PhaseToPhaseVoltage	0.9683
13	Concord V2	PMU	-18.3678
14	GoshenJt V2	PhaseToPhaseVoltage	0.9836
14	GoshenJt V2	PMU	-18.2253
15	FtWayne V2	PhaseToPhaseVoltage	0.9701
15	FtWayne V2	PMU	-18.5197
16	N. E. V2	PhaseToPhaseVoltage	0.9839
16	N. E. V2	PMU	-17.8105
17	Sorenson V2	PhaseToPhaseVoltage	0.9952
17	Sorenson V2	PMU	-16.0029
18	McKinley V2	PhaseToPhaseVoltage	0.9731
18	McKinley V2	PMU	-18.2162
19	Lincoln V2	PhaseToPhaseVoltage	0.963
19	Lincoln V2	PMU	-18.694
20	Adams V2	PhaseToPhaseVoltage	0.9578
20	Adams V2	PMU	-17.8161

Tab. B.29.: Normale State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
21	Jay V2	PhaseToPhaseVoltage	0.9584
21	Jay V2	PMU	-16.2293
22	Randolph V2	PhaseToPhaseVoltage	0.9696
22	Randolph V2	PMU	-13.6762
23	CollCrnr V2	PhaseToPhaseVoltage	0.9997
23	CollCrnr V2	PMU	-8.7587
24	Trenton V2	PhaseToPhaseVoltage	0.992
24	Trenton V2	PMU	-8.8893
25	TannrsCk V2	PhaseToPhaseVoltage	1.05
25	TannrsCk V2	PMU	-1.8242
26	TannrsCk V1	PhaseToPhaseVoltage	1.0149
26	TannrsCk V1	PMU	-0.0437
27	Madison V2	PhaseToPhaseVoltage	0.968
27	Madison V2	PMU	-14.3954
28	Mullin V2	PhaseToPhaseVoltage	0.9616
28	Mullin V2	PMU	-16.1196
29	Grant V2	PhaseToPhaseVoltage	0.9632
29	Grant V2	PMU	-17.1115
30	Sorenson V1	PhaseToPhaseVoltage	0.9855
30	Sorenson V1	PMU	-10.9689
31	DeerCrk V2	PhaseToPhaseVoltage	0.967
31	DeerCrk V2	PMU	-16.9945
32	Delaware V2	PhaseToPhaseVoltage	0.964
32	Delaware V2	PMU	-14.9541
33	Haviland V2	PhaseToPhaseVoltage	0.9716
33	Haviland V2	PMU	-19.1438
34	Rockhill V2	PhaseToPhaseVoltage	0.986
34	Rockhill V2	PMU	-18.5015
35	WestLima V2	PhaseToPhaseVoltage	0.9807
35	WestLima V2	PMU	-18.9276
36	Sterling V2	PhaseToPhaseVoltage	0.9801
36	Sterling V2	PMU	-18.9231
37	EastLima V2	PhaseToPhaseVoltage	0.9921
37	EastLima V2	PMU	-18.0373
38	EastLima V1	PhaseToPhaseVoltage	0.962
38	EastLima V1	PMU	-12.9006
39	NwLibrty V2	PhaseToPhaseVoltage	0.9705
39	NwLibrty V2	PMU	-21.4084
40	West End V2	PhaseToPhaseVoltage	0.97
40	West End V2	PMU	-22.4816

Tab. B.30.: Normale State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
41	S.Tiffin V2	PhaseToPhaseVoltage	0.9669
41	S.Tiffin V2	PMU	-22.9276
42	Howard V2	PhaseToPhaseVoltage	0.985
42	Howard V2	PMU	-21.3318
43	S.Kenton V2	PhaseToPhaseVoltage	0.9786
43	S.Kenton V2	PMU	-18.5465
44	WMVernon V2	PhaseToPhaseVoltage	0.9851
44	WMVernon V2	PMU	-16.0596
45	N.Newark V2	PhaseToPhaseVoltage	0.9867
45	N.Newark V2	PMU	-14.2284
46	W.Lancst V2	PhaseToPhaseVoltage	1.005
46	W.Lancst V2	PMU	-11.4222
47	Crooksvl V2	PhaseToPhaseVoltage	1.0171
47	Crooksvl V2	PMU	-9.1989
48	Zanesvll V2	PhaseToPhaseVoltage	1.0206
48	Zanesvll V2	PMU	-9.979
49	Philo V2	PhaseToPhaseVoltage	1.025
49	Philo V2	PMU	-8.9758
50	WCambrdg V2	PhaseToPhaseVoltage	1.0011
50	WCambrdg V2	PMU	-11.0149
51	Newcmrst V2	PhaseToPhaseVoltage	0.9669
51	Newcmrst V2	PMU	-13.6341
52	SCoshoct V2	PhaseToPhaseVoltage	0.9568
52	SCoshoct V2	PMU	-14.5876
53	Wooster V2	PhaseToPhaseVoltage	0.946
53	Wooster V2	PMU	-15.5626
54	Torrey V2	PhaseToPhaseVoltage	0.955
54	Torrey V2	PMU	-14.651
55	Wagenhls V2	PhaseToPhaseVoltage	0.952
55	Wagenhls V2	PMU	-14.9413
56	Sunnysde V2	PhaseToPhaseVoltage	0.954
56	Sunnysde V2	PMU	-14.7539
57	WNwPhil1 V2	PhaseToPhaseVoltage	0.9706
57	WNwPhil1 V2	PMU	-13.5491
58	WNwPhil2 V2	PhaseToPhaseVoltage	0.959
58	WNwPhil2 V2	PMU	-14.406
59	Tidd V2	PhaseToPhaseVoltage	0.985
59	Tidd V2	PMU	-10.5514
60	SWKammer V2	PhaseToPhaseVoltage	0.9932
60	SWKammer V2	PMU	-6.77

Tab. B.31.: Normale State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
61	W.Kammer V2	PhaseToPhaseVoltage	0.995
61	W.Kammer V2	PMU	-5.8785
62	Natrium V2	PhaseToPhaseVoltage	0.998
62	Natrium V2	PMU	-6.4955
63	Tidd V1	PhaseToPhaseVoltage	0.9687
63	Tidd V1	PMU	-7.1727
64	Kammer V1	PhaseToPhaseVoltage	0.9837
64	Kammer V1	PMU	-5.4069
65	Muskngum V1	PhaseToPhaseVoltage	1.005
65	Muskngum V1	PMU	-2.2814
66	Muskngum V2	PhaseToPhaseVoltage	1.05
66	Muskngum V2	PMU	-2.4405
67	Summerfl V2	PhaseToPhaseVoltage	1.0197
67	Summerfl V2	PMU	-5.0808
68	Sporn V1	PhaseToPhaseVoltage	1.0033
68	Sporn V1	PMU	-2.4018
69	Sporn V2	PhaseToPhaseVoltage	1.0349
69	Sporn V2	PMU	0.0
70	Portsmth V2	PhaseToPhaseVoltage	0.984
70	Portsmth V2	PMU	-7.3835
71	NPortsmt V2	PhaseToPhaseVoltage	0.9869
71	NPortsmt V2	PMU	-7.7946
72	Hillsbro V2	PhaseToPhaseVoltage	0.98
72	Hillsbro V2	PMU	-8.8937
73	Sargents V2	PhaseToPhaseVoltage	0.991
73	Sargents V2	PMU	-8.0061
74	Bellefnt V2	PhaseToPhaseVoltage	0.958
74	Bellefnt V2	PMU	-8.3319
75	SthPoint V2	PhaseToPhaseVoltage	0.9673
75	SthPoint V2	PMU	-7.0698
76	Darrah V2	PhaseToPhaseVoltage	0.943
76	Darrah V2	PMU	-8.2
77	Turner V2	PhaseToPhaseVoltage	1.006
77	Turner V2	PMU	-3.246
78	Chemical V2	PhaseToPhaseVoltage	1.0034
78	Chemical V2	PMU	-3.5499
79	CapitlHl V2	PhaseToPhaseVoltage	1.0092
79	CapitlHl V2	PMU	-3.2508
80	CabinCrk V2	PhaseToPhaseVoltage	1.04
80	CabinCrk V2	PMU	-1.0052

Tab. B.32.: Normale State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
81	Kanawha V1	PhaseToPhaseVoltage	0.9968
81	Kanawha V1	PMU	-1.8531
82	Logan V2	PhaseToPhaseVoltage	0.9888
82	Logan V2	PMU	-2.728
83	Sprigg V2	PhaseToPhaseVoltage	0.9846
83	Sprigg V2	PMU	-1.5391
84	BetsyLne V2	PhaseToPhaseVoltage	0.9798
84	BetsyLne V2	PMU	0.9925
85	BeaverCk V2	PhaseToPhaseVoltage	0.985
85	BeaverCk V2	PMU	2.5457
86	Hazard V2	PhaseToPhaseVoltage	0.9867
86	Hazard V2	PMU	1.1763
87	Pinevlle V3	PhaseToPhaseVoltage	1.015
87	Pinevlle V3	PMU	1.4354
88	Fremont V2	PhaseToPhaseVoltage	0.9875
88	Fremont V2	PMU	5.6751
89	ClinchRv V2	PhaseToPhaseVoltage	1.005
89	ClinchRv V2	PMU	9.7296
90	Holston V2	PhaseToPhaseVoltage	0.985
90	Holston V2	PMU	3.3259
91	HolstonT V2	PhaseToPhaseVoltage	0.98
91	HolstonT V2	PMU	3.3471
92	Saltvlle V2	PhaseToPhaseVoltage	0.993
92	Saltvlle V2	PMU	3.8363
93	Tazewell V2	PhaseToPhaseVoltage	0.9874
93	Tazewell V2	PMU	0.8322
94	Switchbk V2	PhaseToPhaseVoltage	0.9908
94	Switchbk V2	PMU	-1.3173
95	Caldwell V2	PhaseToPhaseVoltage	0.9811
95	Caldwell V2	PMU	-2.2881
96	Baileysv V2	PhaseToPhaseVoltage	0.9928
96	Baileysv V2	PMU	-2.4546
97	Sundial V2	PhaseToPhaseVoltage	1.0114
97	Sundial V2	PMU	-2.0804
98	Bradley V2	PhaseToPhaseVoltage	1.0235
98	Bradley V2	PMU	-2.557
99	Hinton V2	PhaseToPhaseVoltage	1.01
99	Hinton V2	PMU	-2.9192
100	Glen Lyn V2	PhaseToPhaseVoltage	1.017
100	Glen Lyn V2	PMU	-1.9232

Tab. B.33.: Normale State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
101	Wythe V2	PhaseToPhaseVoltage	0.9928
101	Wythe V2	PMU	-0.355
102	Smythe V2	PhaseToPhaseVoltage	0.9916
102	Smythe V2	PMU	2.3367
103	Claytor V2	PhaseToPhaseVoltage	1.001
103	Claytor V2	PMU	-5.5238
104	Hancock V2	PhaseToPhaseVoltage	0.971
104	Hancock V2	PMU	-8.2631
105	Roanoke V2	PhaseToPhaseVoltage	0.965
105	Roanoke V2	PMU	-9.3703
106	Cloverdl V2	PhaseToPhaseVoltage	0.9611
106	Cloverdl V2	PMU	-9.6249
107	Reusens V2	PhaseToPhaseVoltage	0.952
107	Reusens V2	PMU	-12.4289
108	Blaine V2	PhaseToPhaseVoltage	0.9662
108	Blaine V2	PMU	-10.5703
109	Franklin V2	PhaseToPhaseVoltage	0.967
109	Franklin V2	PMU	-11.0229
110	Fieldale V2	PhaseToPhaseVoltage	0.973
110	Fieldale V2	PMU	-11.8696
111	DanRiver V2	PhaseToPhaseVoltage	0.98
111	DanRiver V2	PMU	-10.2244
112	Danville V2	PhaseToPhaseVoltage	0.975
112	Danville V2	PMU	-14.9689
113	Deer Crk V2	PhaseToPhaseVoltage	0.993
113	Deer Crk V2	PMU	-16.0044
114	WMedford V2	PhaseToPhaseVoltage	0.9607
114	WMedford V2	PMU	-15.2816
115	Medford V2	PhaseToPhaseVoltage	0.9606
115	Medford V2	PMU	-15.2888
116	KygerCrk V2	PhaseToPhaseVoltage	1.005
116	KygerCrk V2	PMU	-2.8369
117	Corey V2	PhaseToPhaseVoltage	0.9738
117	Corey V2	PMU	-19.05
118	WHuntngd V2	PhaseToPhaseVoltage	0.9494
118	WHuntngd V2	PMU	-8.0576

Tab. B.34.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV.

Nummer	Name	Typ	Schätzung
1	Riversde V2	PhaseToPhaseVoltage	0.9551
1	Riversde V2	PMU	-18.947
2	Pokagon V2	PhaseToPhaseVoltage	0.9714
2	Pokagon V2	PMU	-18.406
3	HickryCk V2	PhaseToPhaseVoltage	0.9677
3	HickryCk V2	PMU	-18.0635
4	NwCarlsl V2	PhaseToPhaseVoltage	0.9981
4	NwCarlsl V2	PMU	-14.3468
5	Olive V2	PhaseToPhaseVoltage	1.002
5	Olive V2	PMU	-13.9017
6	Kankakee V2	PhaseToPhaseVoltage	0.9901
6	Kankakee V2	PMU	-16.6283
7	JacksnRd V2	PhaseToPhaseVoltage	0.9894
7	JacksnRd V2	PMU	-17.0719
8	Olive V1	PhaseToPhaseVoltage	1.015
8	Olive V1	PMU	-8.8818
9	Bequine V1	PhaseToPhaseVoltage	1.0429
9	Bequine V1	PMU	-1.628
10	Breed V1	PhaseToPhaseVoltage	1.0499
10	Breed V1	PMU	5.9532
11	SouthBnd V2	PhaseToPhaseVoltage	0.9851
11	SouthBnd V2	PMU	-16.9129
12	TwinBrch V2	PhaseToPhaseVoltage	0.99
12	TwinBrch V2	PMU	-17.4289
13	Concord V2	PhaseToPhaseVoltage	0.9683
13	Concord V2	PMU	-18.2878
14	GoshenJt V2	PhaseToPhaseVoltage	0.9836
14	GoshenJt V2	PMU	-18.1453
15	FtWayne V2	PhaseToPhaseVoltage	0.9701
15	FtWayne V2	PMU	-18.4395
16	N. E. V2	PhaseToPhaseVoltage	0.9839
16	N. E. V2	PMU	-17.7295
17	Sorenson V2	PhaseToPhaseVoltage	0.9952
17	Sorenson V2	PMU	-15.9197
18	McKinley V2	PhaseToPhaseVoltage	0.9731
18	McKinley V2	PMU	-18.134
19	Lincoln V2	PhaseToPhaseVoltage	0.963
19	Lincoln V2	PMU	-18.6125
20	Adams V2	PhaseToPhaseVoltage	0.9577
20	Adams V2	PMU	-17.7211

Tab. B.35.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
21	Jay V2	PhaseToPhaseVoltage	0.9584
21	Jay V2	PMU	-16.1243
22	Randolph V2	PhaseToPhaseVoltage	0.9695
22	Randolph V2	PMU	-13.5597
23	CollCrnr V2	PhaseToPhaseVoltage	0.9997
23	CollCrnr V2	PMU	-8.6242
24	Trenton V2	PhaseToPhaseVoltage	0.992
24	Trenton V2	PMU	-8.7241
25	TannrsCk V2	PhaseToPhaseVoltage	1.05
25	TannrsCk V2	PMU	-1.7125
26	TannrsCk V1	PhaseToPhaseVoltage	1.0149
26	TannrsCk V1	PMU	0.0586
27	Madison V2	PhaseToPhaseVoltage	0.968
27	Madison V2	PMU	-14.2868
28	Mullin V2	PhaseToPhaseVoltage	0.9616
28	Mullin V2	PMU	-16.014
29	Grant V2	PhaseToPhaseVoltage	0.9632
29	Grant V2	PMU	-17.0092
30	Sorenson V1	PhaseToPhaseVoltage	0.9855
30	Sorenson V1	PMU	-10.8903
31	DeerCrk V2	PhaseToPhaseVoltage	0.967
31	DeerCrk V2	PMU	-16.8934
32	Delaware V2	PhaseToPhaseVoltage	0.964
32	Delaware V2	PMU	-14.8448
33	Haviland V2	PhaseToPhaseVoltage	0.9716
33	Haviland V2	PMU	-19.075
34	Rockhill V2	PhaseToPhaseVoltage	0.986
34	Rockhill V2	PMU	-18.445
35	WestLima V2	PhaseToPhaseVoltage	0.9807
35	WestLima V2	PMU	-18.8713
36	Sterling V2	PhaseToPhaseVoltage	0.9801
36	Sterling V2	PMU	-18.8667
37	EastLima V2	PhaseToPhaseVoltage	0.9921
37	EastLima V2	PMU	-17.9811
38	EastLima V1	PhaseToPhaseVoltage	0.962
38	EastLima V1	PMU	-12.8447
39	NwLibrty V2	PhaseToPhaseVoltage	0.9705
39	NwLibrty V2	PMU	-21.3585
40	West End V2	PhaseToPhaseVoltage	0.97
40	West End V2	PMU	-22.4353

Tab. B.36.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
41	S.Tiffin V2	PhaseToPhaseVoltage	0.9669
41	S.Tiffin V2	PMU	-22.8841
42	Howard V2	PhaseToPhaseVoltage	0.985
42	Howard V2	PMU	-21.296
43	S.Kenton V2	PhaseToPhaseVoltage	0.9786
43	S.Kenton V2	PMU	-18.5009
44	WMVernon V2	PhaseToPhaseVoltage	0.9851
44	WMVernon V2	PMU	-16.0301
45	N.Newark V2	PhaseToPhaseVoltage	0.9867
45	N.Newark V2	PMU	-14.2047
46	W.Lancst V2	PhaseToPhaseVoltage	1.005
46	W.Lancst V2	PMU	-11.4028
47	Crooksvl V2	PhaseToPhaseVoltage	1.0171
47	Crooksvl V2	PMU	-9.1831
48	Zanesvll V2	PhaseToPhaseVoltage	1.0206
48	Zanesvll V2	PMU	-9.9611
49	Philo V2	PhaseToPhaseVoltage	1.025
49	Philo V2	PMU	-8.9581
50	WCambrdg V2	PhaseToPhaseVoltage	1.0011
50	WCambrdg V2	PMU	-10.9976
51	Newcmrst V2	PhaseToPhaseVoltage	0.9669
51	Newcmrst V2	PMU	-13.6172
52	SCoshoct V2	PhaseToPhaseVoltage	0.9568
52	SCoshoct V2	PMU	-14.5709
53	Wooster V2	PhaseToPhaseVoltage	0.946
53	Wooster V2	PMU	-15.5462
54	Torrey V2	PhaseToPhaseVoltage	0.955
54	Torrey V2	PMU	-14.6348
55	Wagenhls V2	PhaseToPhaseVoltage	0.952
55	Wagenhls V2	PMU	-14.9253
56	Sunnysde V2	PhaseToPhaseVoltage	0.954
56	Sunnysde V2	PMU	-14.7378
57	WNwPhil1 V2	PhaseToPhaseVoltage	0.9706
57	WNwPhil1 V2	PMU	-13.5325
58	WNwPhil2 V2	PhaseToPhaseVoltage	0.959
58	WNwPhil2 V2	PMU	-14.3895
59	Tidd V2	PhaseToPhaseVoltage	0.985
59	Tidd V2	PMU	-10.5365
60	SWKammer V2	PhaseToPhaseVoltage	0.9932
60	SWKammer V2	PMU	-6.7555

Tab. B.37.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
61	W.Kammer V2	PhaseToPhaseVoltage	0.995
61	W.Kammer V2	PMU	-5.8641
62	Natrium V2	PhaseToPhaseVoltage	0.998
62	Natrium V2	PMU	-6.481
63	Tidd V1	PhaseToPhaseVoltage	0.9687
63	Tidd V1	PMU	-7.1584
64	Kammer V1	PhaseToPhaseVoltage	0.9837
64	Kammer V1	PMU	-5.3928
65	Muskngum V1	PhaseToPhaseVoltage	1.005
65	Muskngum V1	PMU	-2.268
66	Muskngum V2	PhaseToPhaseVoltage	1.05
66	Muskngum V2	PMU	-2.4254
67	Summerfl V2	PhaseToPhaseVoltage	1.0197
67	Summerfl V2	PMU	-5.066
68	Sporn V1	PhaseToPhaseVoltage	1.0033
68	Sporn V1	PMU	-2.3963
69	Sporn V2	PhaseToPhaseVoltage	1.0349
69	Sporn V2	PMU	0.0
70	Portsmth V2	PhaseToPhaseVoltage	0.984
70	Portsmth V2	PMU	-7.5583
71	NPortsmt V2	PhaseToPhaseVoltage	1.0864
71	NPortsmt V2	PMU	-7.9971
72	Hillsbro V2	PhaseToPhaseVoltage	1.08
72	Hillsbro V2	PMU	-8.7997
73	Sargents V2	PhaseToPhaseVoltage	1.09
73	Sargents V2	PMU	-8.3155
74	Bellefnt V2	PhaseToPhaseVoltage	0.958
74	Bellefnt V2	PMU	-8.4355
75	SthPoint V2	PhaseToPhaseVoltage	0.9673
75	SthPoint V2	PMU	-7.1503
76	Darrah V2	PhaseToPhaseVoltage	0.943
76	Darrah V2	PMU	-8.2557
77	Turner V2	PhaseToPhaseVoltage	1.006
77	Turner V2	PMU	-3.2654
78	Chemical V2	PhaseToPhaseVoltage	1.0034
78	Chemical V2	PMU	-3.5684
79	CapitlHl V2	PhaseToPhaseVoltage	1.0092
79	CapitlHl V2	PMU	-3.2675
80	CabinCrk V2	PhaseToPhaseVoltage	1.04
80	CabinCrk V2	PMU	-1.0171

Tab. B.38.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

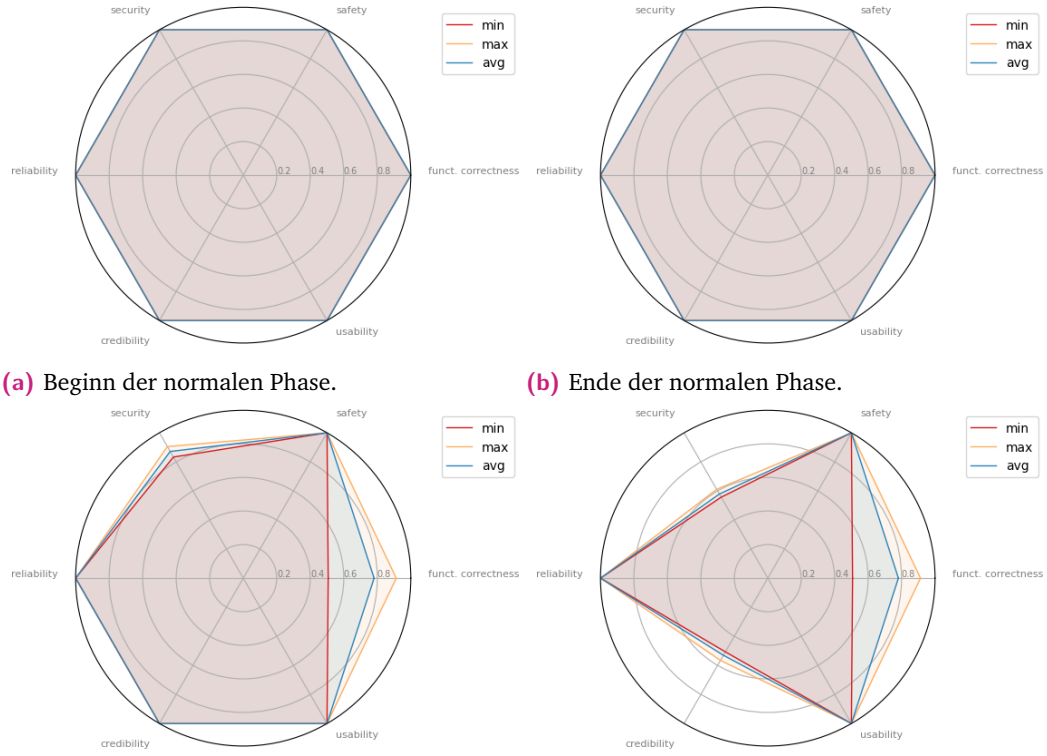
Nummer	Name	Typ	Schätzung
81	Kanawha V1	PhaseToPhaseVoltage	0.9968
81	Kanawha V1	PMU	-1.8541
82	Logan V2	PhaseToPhaseVoltage	0.9888
82	Logan V2	PMU	-2.7439
83	Sprigg V2	PhaseToPhaseVoltage	0.9846
83	Sprigg V2	PMU	-1.5548
84	BetsyLne V2	PhaseToPhaseVoltage	0.9798
84	BetsyLne V2	PMU	0.9772
85	BeaverCk V2	PhaseToPhaseVoltage	0.985
85	BeaverCk V2	PMU	2.5307
86	Hazard V2	PhaseToPhaseVoltage	0.9867
86	Hazard V2	PMU	1.1612
87	Pinevlle V3	PhaseToPhaseVoltage	1.015
87	Pinevlle V3	PMU	1.4204
88	Fremont V2	PhaseToPhaseVoltage	0.9875
88	Fremont V2	PMU	5.6605
89	ClinchRv V2	PhaseToPhaseVoltage	1.005
89	ClinchRv V2	PMU	9.7152
90	Holston V2	PhaseToPhaseVoltage	0.985
90	Holston V2	PMU	3.3115
91	HolstonT V2	PhaseToPhaseVoltage	0.98
91	HolstonT V2	PMU	3.3328
92	Saltvlle V2	PhaseToPhaseVoltage	0.993
92	Saltvlle V2	PMU	3.8221
93	Tazewell V2	PhaseToPhaseVoltage	0.9874
93	Tazewell V2	PMU	0.8182
94	Switchbk V2	PhaseToPhaseVoltage	0.9908
94	Switchbk V2	PMU	-1.3312
95	Caldwell V2	PhaseToPhaseVoltage	0.9811
95	Caldwell V2	PMU	-2.3022
96	Baileysv V2	PhaseToPhaseVoltage	0.9928
96	Baileysv V2	PMU	-2.4688
97	Sundial V2	PhaseToPhaseVoltage	1.0114
97	Sundial V2	PMU	-2.0935
98	Bradley V2	PhaseToPhaseVoltage	1.0235
98	Bradley V2	PMU	-2.5695
99	Hinton V2	PhaseToPhaseVoltage	1.01
99	Hinton V2	PMU	-2.9322
100	Glen Lyn V2	PhaseToPhaseVoltage	1.017
100	Glen Lyn V2	PMU	-1.9367

Tab. B.39.: Kompromittierte State-Estimation-Ergebnisse für das IEEE118HV (Fortsetzung).

Nummer	Name	Typ	Schätzung
101	Wythe V2	PhaseToPhaseVoltage	0.9928
101	Wythe V2	PMU	-0.3688
102	Smythe V2	PhaseToPhaseVoltage	0.9916
102	Smythe V2	PMU	2.3227
103	Claytor V2	PhaseToPhaseVoltage	1.001
103	Claytor V2	PMU	-5.5373
104	Hancock V2	PhaseToPhaseVoltage	0.971
104	Hancock V2	PMU	-8.2766
105	Roanoke V2	PhaseToPhaseVoltage	0.965
105	Roanoke V2	PMU	-9.3838
106	Cloverdl V2	PhaseToPhaseVoltage	0.9611
106	Cloverdl V2	PMU	-9.6384
107	Reusens V2	PhaseToPhaseVoltage	0.952
107	Reusens V2	PMU	-12.4424
108	Blaine V2	PhaseToPhaseVoltage	0.9662
108	Blaine V2	PMU	-10.5838
109	Franklin V2	PhaseToPhaseVoltage	0.967
109	Franklin V2	PMU	-11.0364
110	Fieldale V2	PhaseToPhaseVoltage	0.973
110	Fieldale V2	PMU	-11.8831
111	DanRiver V2	PhaseToPhaseVoltage	0.98
111	DanRiver V2	PMU	-10.2379
112	Danville V2	PhaseToPhaseVoltage	0.975
112	Danville V2	PMU	-14.9824
113	Deer Crk V2	PhaseToPhaseVoltage	0.993
113	Deer Crk V2	PMU	-15.9182
114	WMedford V2	PhaseToPhaseVoltage	0.9607
114	WMedford V2	PMU	-15.1725
115	Medford V2	PhaseToPhaseVoltage	0.9606
115	Medford V2	PMU	-15.1798
116	KygerCrk V2	PhaseToPhaseVoltage	1.005
116	KygerCrk V2	PMU	-2.8313
117	Corey V2	PhaseToPhaseVoltage	0.9738
117	Corey V2	PMU	-18.97
118	WHuntngd V2	PhaseToPhaseVoltage	0.9494
118	WHuntngd V2	PMU	-8.1266

C Weitere Evaluationsergebnisse

C.1 Aussagekraft der Trust-sensensitiven Lagebilderkennung



(a) Beginn der normalen Phase.

(b) Ende der normalen Phase.

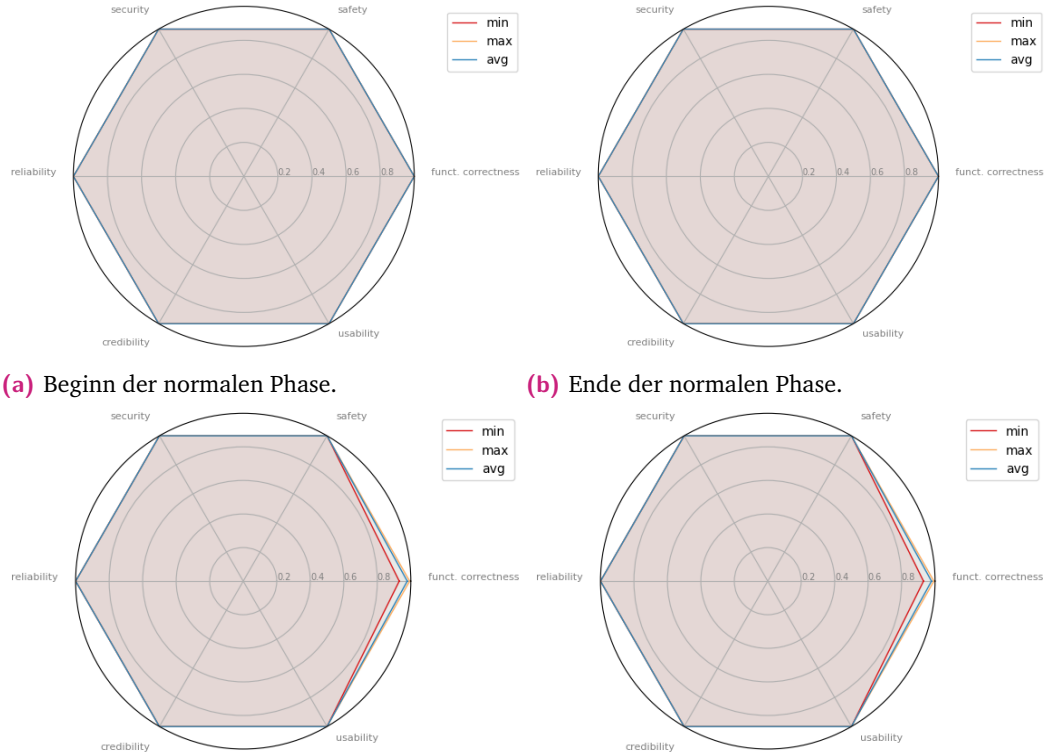
(c) Beginn der abnormalen Phase.

(d) Ende der abnormalen Phase.

Abb. C.1.: Der multivariate Trust in die Zustandsvariablen für Szenario 1 (CIGRE12MV) zu unterschiedlichen Zeitpunkten aggregiert pro Trust-Facette dargestellt als Netzdiagramme.

Tab. C.1.: Korrelationskoeffizienten nach Pearson für Szenario 1 (CIGRE12MV).

Zustandsvariable		Trust							
Sammel-schiene	Var.	Running Services AD	Mem Load AD	CPU Load AD	IDS Alerts AD	Historical Trust AD	min	max	avg
1	Vm	nan	nan	nan	nan	nan	nan	nan	nan
	Va	nan	nan	nan	nan	nan	nan	nan	nan
2	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
3	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
4	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
5	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
6	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
7	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
8	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
9	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
10	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
11	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
12	Vm	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
	Va	-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
min		-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
max		-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988
mavg		-1.0	-1.0	-1.0	-0.977	-0.964	-1.0	-0.964	-0.988



(a) Beginn der normalen Phase.

(b) Ende der normalen Phase.

(c) Beginn der abnormalen Phase.

(d) Ende der abnormalen Phase.

Abb. C.2.: Der multivariate Trust in die Zustandsvariablen für Szenario 3 (IEEE39HV) zu unterschiedlichen Zeitpunkten aggregiert pro Trust-Facette dargestellt als Netzdiagramme.

Tab. C.2.: Korrelationskoeffizienten nach Pearson für Szenario 3 (IEEE39HV).

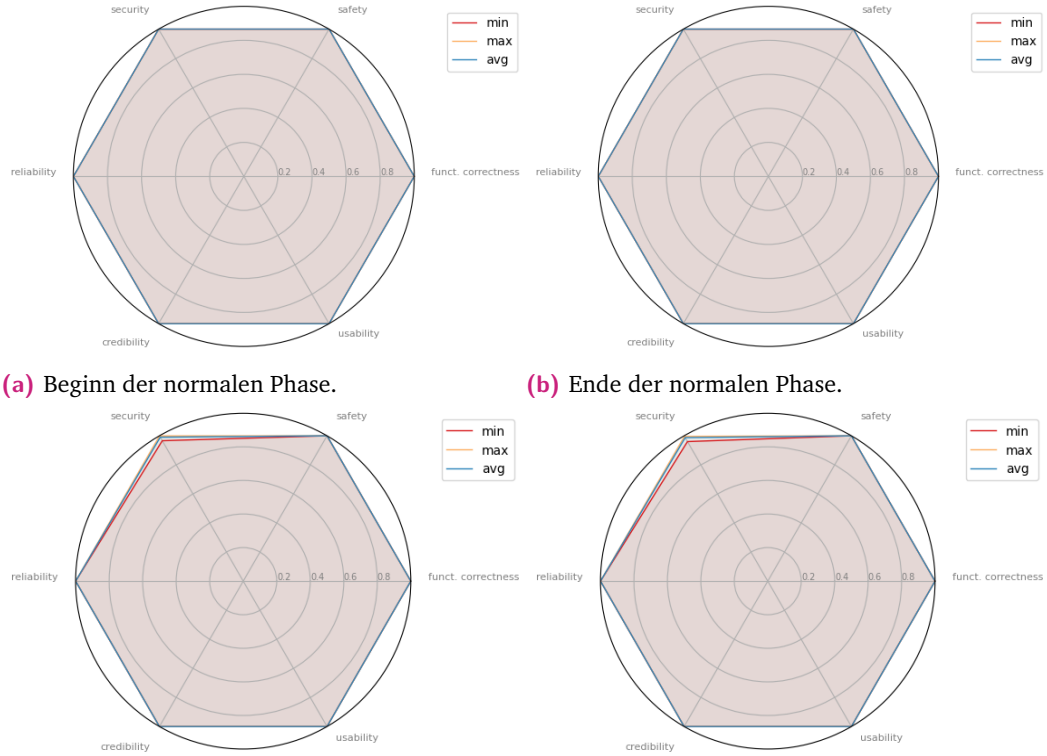
Zustandsvariable		Trust					
Sammel-schiene	Var.	Running Services AD	Mem Load AD	CPU Load AD	min	max	avg
1	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
2	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
3	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
4	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
5	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
6	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
7	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
8	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
9	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
10	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
11	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
12	Vm	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
13	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

Tab. C.3.: Korrelationskoeffizienten nach Pearson für Szenario 3 (IEEE39HV) (Fortsetzung).

Zustandsvariable		Trust					
Sammel-schiene	Var.	Running Services AD	Mem Load AD	CPU Load AD	min	max	avg
14	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
15	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
16	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
17	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
18	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
19	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
20	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
21	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
22	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
23	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
24	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
25	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
26	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0

Tab. C.4.: Korrelationskoeffizienten nach Pearson für Szenario 3 (IEEE39HV) (Fortsetzung).

Zustandsvariable		Trust					
Sammel-schiene	Var.	Running Services AD	Mem Load AD	CPU Load AD	min	max	avg
27	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
28	Vm	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
29	Vm	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
30	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
31	Vm	nan	nan	nan	nan	nan	nan
	Va	nan	nan	nan	nan	nan	nan
32	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
33	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
34	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
35	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
36	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
37	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
38	Vm	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
39	Vm	nan	nan	nan	nan	nan	nan
	Va	-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
min		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
max		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0
mavg		-1.0	-1.0	-1.0	-1.0	-1.0	-1.0



(a) Beginn der normalen Phase.

(b) Ende der normalen Phase.

(c) Beginn der abnormalen Phase.

(d) Ende der abnormalen Phase.

Abb. C.3.: Der multivariate Trust in die Zustandsvariablen für Szenario 5 (IEEE118HV) zu unterschiedlichen Zeitpunkten aggregiert pro Trust-Facette dargestellt als Netzdiagramme.

Tab. C.5.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
1	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
2	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
3	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
4	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
5	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
6	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
7	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
8	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
9	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
10	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
11	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
12	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
13	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992

Tab. C.6.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
14	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
15	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
16	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
17	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
18	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
19	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
20	**Vm**	-0.996	-0.996	-0.996	-0.996
	Va	-0.996	-0.996	-0.996	-0.996
21	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
22	**Vm**	-0.992	-0.992	-0.992	-0.992
	Va	-0.992	-0.992	-0.992	-0.992
23	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
24	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
25	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
26	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992

Tab. C.7.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
27	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
28	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
29	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
30	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
31	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
32	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
33	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
34	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
35	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
36	**Vm**	nan	nan	nan	nan
	Va	-0.997	-0.997	-0.997	-0.997
37	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
38	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
39	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996

Tab. C.8.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
40	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
41	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
42	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
43	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
44	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
45	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
46	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
47	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
48	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
49	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
50	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
51	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
52	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996

Tab. C.9.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
53	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
54	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
55	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
56	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
57	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
58	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
59	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
60	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
61	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
62	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
63	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
64	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
65	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996

Tab. C.10.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
66	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
67	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
68	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
69	**Vm**	nan	nan	nan	nan
	Va	nan	nan	nan	nan
70	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
71	**Vm**	-0.996	-0.996	-0.996	-0.996
	Va	-0.996	-0.996	-0.996	-0.996
72	**Vm**	-0.996	-0.996	-0.996	-0.996
	Va	-0.996	-0.996	-0.996	-0.996
73	**Vm**	-0.991	-0.991	-0.991	-0.991
	Va	-0.991	-0.991	-0.991	-0.991
74	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
75	**Vm**	nan	nan	nan	nan
	Va	-0.991	-0.991	-0.991	-0.991
76	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
77	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995
78	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995

Tab. C.11.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
79	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
80	**Vm**	nan	nan	nan	nan
	Va	-0.991	-0.991	-0.991	-0.991
81	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
82	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
83	**Vm**	nan	nan	nan	nan
	Va	-0.991	-0.991	-0.991	-0.991
84	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
85	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
86	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
87	**Vm**	nan	nan	nan	nan
	Va	-0.989	-0.989	-0.989	-0.989
88	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
89	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
90	**Vm**	nan	nan	nan	nan
	Va	-0.998	-0.998	-0.998	-0.998
91	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995

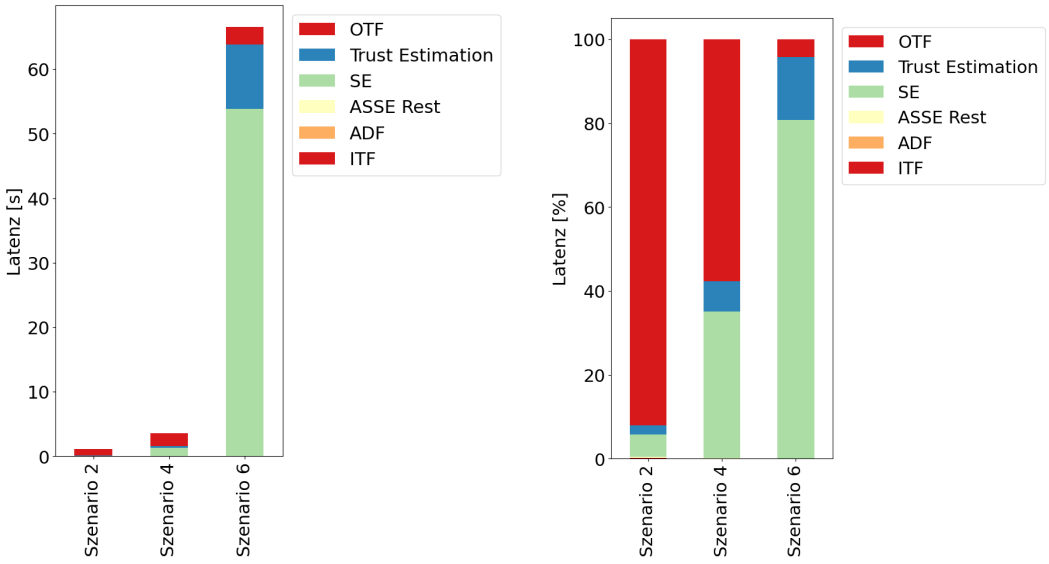
Tab. C.12.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
92	**Vm**	nan	nan	nan	nan
	Va	-0.998	-0.998	-0.998	-0.998
93	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
94	**Vm**	nan	nan	nan	nan
	Va	-0.989	-0.989	-0.989	-0.989
95	**Vm**	nan	nan	nan	nan
	Va	-0.99	-0.99	-0.99	-0.99
96	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
97	**Vm**	nan	nan	nan	nan
	Va	-0.99	-0.99	-0.99	-0.99
98	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
99	**Vm**	nan	nan	nan	nan
	Va	-0.986	-0.986	-0.986	-0.986
100	**Vm**	nan	nan	nan	nan
	Va	-0.989	-0.989	-0.989	-0.989
101	**Vm**	nan	nan	nan	nan
	Va	-0.996	-0.996	-0.996	-0.996
102	**Vm**	nan	nan	nan	nan
	Va	-0.997	-0.997	-0.997	-0.997
103	**Vm**	nan	nan	nan	nan
	Va	-0.986	-0.986	-0.986	-0.986
104	**Vm**	nan	nan	nan	nan
	Va	-0.995	-0.995	-0.995	-0.995

Tab. C.13.: Korrelationskoeffizienten nach Pearson für Szenario 5 (IEEE118HV) (Fortsetzung).

Zustandsvariable		Trust			
Sammel-schiene	**Var.**	**IDS Alerts AD**	**min**	**max**	**avg**
105	**Vm**	nan	nan	nan	nan
	Va	-0.997	-0.997	-0.997	-0.997
106	**Vm**	nan	nan	nan	nan
	Va	-0.987	-0.987	-0.987	-0.987
107	**Vm**	nan	nan	nan	nan
	Va	-0.997	-0.997	-0.997	-0.997
108	**Vm**	nan	nan	nan	nan
	Va	-0.99	-0.99	-0.99	-0.99
109	**Vm**	nan	nan	nan	nan
	Va	-0.994	-0.994	-0.994	-0.994
110	**Vm**	nan	nan	nan	nan
	Va	-0.993	-0.993	-0.993	-0.993
111	**Vm**	nan	nan	nan	nan
	Va	-0.992	-0.992	-0.992	-0.992
112	**Vm**	nan	nan	nan	nan
	Va	-0.998	-0.998	-0.998	-0.998
113	**Vm**	nan	nan	nan	nan
	Va	-0.998	-0.998	-0.998	-0.998
114	**Vm**	nan	nan	nan	nan
	Va	-0.989	-0.989	-0.989	-0.989
115	**Vm**	nan	nan	nan	nan
	Va	-0.99	-0.99	-0.99	-0.99
116	**Vm**	nan	nan	nan	nan
	Va	-0.99	-0.99	-0.99	-0.99
117	**Vm**	nan	nan	nan	nan
	Va	-0.983	-0.983	-0.983	-0.983
118	**Vm**	nan	nan	nan	nan
	Va	-0.998	-0.998	-0.998	-0.998
min		-0.998	-0.998	-0.998	-0.998
max		-0.983	-0.983	-0.983	-0.983
mavg		-0.994	-0.994	-0.994	-0.994

C.2 Aktualität



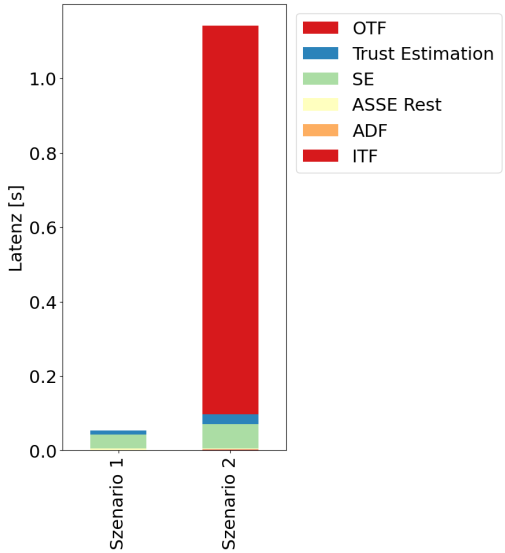
(a) Absolute Latenzen in Sekunden.

(b) Relative Latenzen in Prozenten.

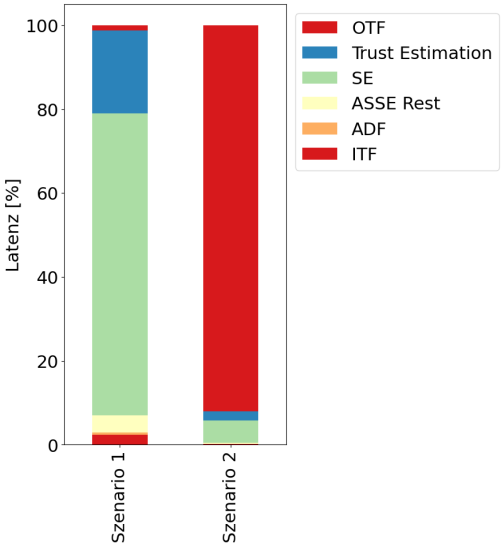
Abb. C.4.: Die Latenzen (absolut und relativ) für die Szenarien 2, 4 und 6 in Histogrammen aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

Tab. C.14.: Die Latenzen (absolut und relativ) für die Szenarien 2, 4 und 6 aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

	Szenario2		**Szenario4**		**Szenario6**	
	[s]	**[%]**	**[s]**	**[%]**	**[s]**	**[%]**
ITF	0.002	0.138	0.001	0.022	0.001	0.002
ADF	0.0	0.029	0.0	0.004	0.0	0.0
ASSE Rest	0.004	0.36	0.005	0.124	0.019	0.028
State Estimation	0.065	5.297	1.283	34.913	53.862	80.783
Trust Estimation	0.026	2.084	0.267	7.236	9.951	14.914
OTF	1.045	92.091	2.055	57.7	2.714	4.273
total	1.142	100.0	3.611	100.0	66.549	100.0

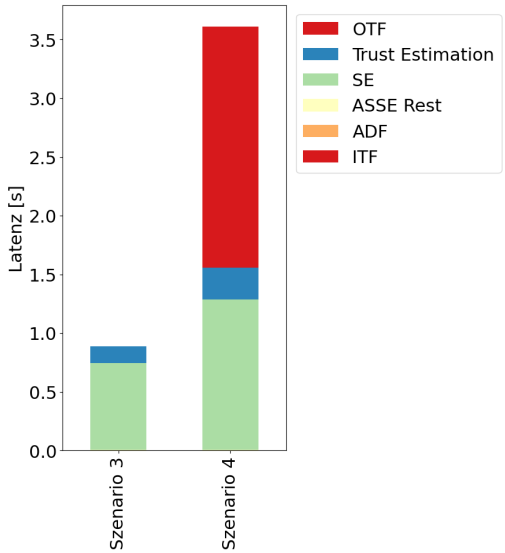


(a) Absolute Latenzen in Sekunden.

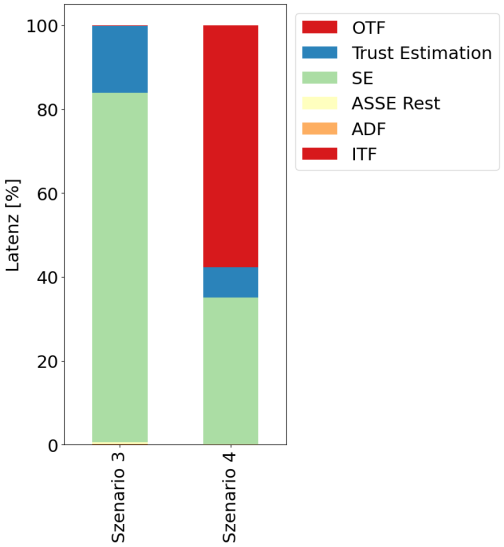


(b) Relative Latenzen in Prozenten.

Abb. C.5.: Die Latenzen (absolut und relativ) für die Szenarien 1 und 2 in Histogrammen aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

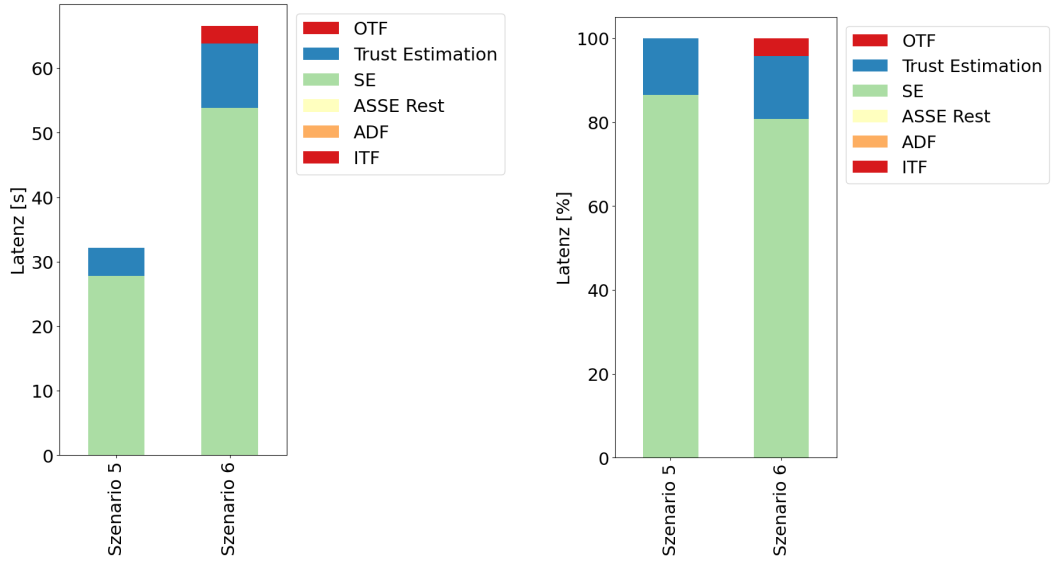


(a) Absolute Latenzen in Sekunden.



(b) Relative Latenzen in Prozenten.

Abb. C.6.: Die Latenzen (absolut und relativ) für die Szenarien 3 und 4 in Histogrammen aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.



(a) Absolute Latenzen in Sekunden.

(b) Relative Latenzen in Prozenten.

Abb. C.7.: Die Latenzen (absolut und relativ) für die Szenarien 5 und 6 in Histogrammen aufgeschlüsselt nach den wesentlichsten Komponenten von ASSESS.

Glossar

A

abgeleiteter Untersuchungsgegenstand Ein abgeleiteter Untersuchungsgegenstand ist ein Untersuchungsgegenstand, ohne direkte Trust-Informationen und für den Trust über den geschätzten Trust in andere Untersuchungsgegenstände geschätzt wird.

Aktualität Die Aktualität der Trust-sensitiven Lagebilderkennung beschreibt die Zeitspanne zwischen dem Eintreffen neuer Informationen (elektrotechnische Messwerte oder Trust-Inputs) und dem Bereitstellen eines neuen Lagebildes (siehe auch Latenz).

alternative State Estimation Die alternative State Estimation nutzt alternative Messwerte (sogenannte Pseudomesswerte, z.B. historische), um ein alternatives, vertrauenswürdiges Lagebild zu erstellen, falls das anomaliesensitive ein Risiko für das System darstellt.

Anomalieerkennungsframework (engl. anomaly detection framework) Die Aufgabe des Anomalieerkennungsframeworks als eine Komponente der anomaliesensitive State Estimation mit Streaming Systemen ist es, durch die Integration verschiedener Anomaliedetektoren und auf Basis verschiedener externer Informationsquellen Anomalien in dem Trust in die Messwerte zu finden und zu bewerten.

anomaliesensitive State Estimation Durch eine anomaliesensitive State Estimation werden Zustandsvariablen und der Trust in diese auf Basis von Messwerten und dem Trust in diese geschätzt.

anomaliesensitive State Estimation mit Streaming Systemen Das in dieser Arbeit umgesetzte System zur Erreichung der Forschungsziele und nichtfunktionalen Anforderungen heißt anomaliesensitive State Estimation mit Streaming Systemen. Es ist ein in einem Datenstrommanagementsystem umgesetztes System, das ein Eingangstransformationsframework (engl. input transform. framework), ein Anomalieerkennungsframework (engl. anomaly detection framework), eine anomaliesensitive State Estimation, eine alternative State

Estimation und ein Ausgangstransformationsframework (engl. output transf. framework) beinhaltet.

Application Protocol Control Information Eine Anwendungssteuerungsinformation (engl. application protocol control information) ist im IEC 60870-5-104-Standard Bestandteil einer Application Protocol Data Unit. Eine Anwendungssteuerungsinformation fängt stets mit demselben Byte an, 64 hexadezimal, gefolgt von der Restlänge des Telegramms (Länge abzüglich Start- und Längenbyte). Es folgen vier Bytes, die je nach Steuerungsinformationsformat variieren [Int16].

Application Protocol Data Unit Ein nach dem IEC 60870-5-104-Standard übertragenes Telegramm besteht aus einer Anwendungsprotokolldateneinheit (engl. application protocol data unit). Eine Anwendungsprotokolldateneinheit setzt sich aus einer Application Protocol Control Information und je nach Steuerungsinformationen einer Application Service Data Unit zusammen [Int16].

Application Service Data Unit Eine Anwendungsservicedateneinheit (engl. application service data unit) kann Bestandteil einer Application Protocol Data Unit sein. Sie ist der Bestandteil einer Application Protocol Data Unit, der Informationen überträgt [Int16].

Ausgangstransformationsframework (engl. output transf. framework) In der anomaliesensitiven State Estimation mit Streaming Systemen ist das Ausgangstransformationsframework eine Komponente mit drei Funktionen. Erstens hat es die Aufgabe auszuwählen, welches State-Estimation-Ergebnis ausgegeben werden soll, zweitens sorgt es für adäquate Ausgabedatenraten und drittens hat es analog zum Eingangstransformationsframework (engl. input transform. framework) die Aufgabe das Ergebnis in ein geeignetes Format zu überführen.

B

Bad Data Detection Die Erkennung falscher Messwerte (engl. Bad Data Detection) ist der Prozess, zufallsverteilte, fehlerhafte Messwerte zu erkennen, zu identifizieren und herauszurechnen [AE04].

Bedienbarkeit „Die Eigenschaft eines Systems, eine Benutzerschnittstelle anzubieten, die vom Benutzer effizient, effektiv und zu seiner Zufriedenheit bedient werden kann, insbesondere unter Berücksichtigung von Benutzerkontrolle und Privacy“ [Ste+10; SR12].

Betriebssicherheit „Die Eigenschaft eines Systems, zu keiner Zeit in einen Zustand einzutreten oder einen Output zu erzeugen, in dem oder durch den das System seine Benutzer, sich selbst oder Teile von sich oder seine Umwelt schädigt“ [Ste+10; SR12].

C

cyber-physisches Energiesystem Cyber-physische Energiesysteme sind Energiesysteme, die sich im Vergleich zu traditionellen Energiesystemen durch eine deutlich stärkere Verwobenheit mit Informations- und Kommunikationstechnik auszeichnen.

D

Datenmodell für Messwerte Das Datenmodell für Messwerte ist ein relationales Tupel bestehend aus den folgenden vier Attributen: `GraphElement` (Text), `MeasurementType` (Text), `Measurement` (Gleitkommazahl) und `Properties` (Schlüssel-Wert-Paare). Identifiziert wird ein Messwert über die Kombination aus `GraphElement`, also einem Knoten oder einer Kante aus der Topologie des Stromnetzes, und dem `MeasurementType`, z.B. einem einphasigen Leistungswert. Hinzu kommt als Kontextinformation der Zeitstempel des Messwertes.

Datenmodell für Messwerte mit Trust-Wert Die Schnittstelle für die Ausgaben von Trust-Schätzern wird durch das Datenmodell für Messwerte mit Trust-Wert definiert, das das Datenmodell für Messwerte um multivariate Trust-Werte erweitert.

Datenmodell für Topologien Das Datenmodell für Topologien ist ein Eigenschaftsgraph mit Knoten (z.B. Sammelschienen oder Router) und Kanten (z.B. Stromleitungen oder IKT-Verbindungen). Die Art eines Knotens oder einer Kante, also die CIM-Klasse, wird durch einen Bezeichner repräsentiert, während die Eigenschaften von Graphobjekten die Attribute eines CIM-Objekts hierarchisch abbilden.

Datenstrom Ein Datenstrom ist eine kontinuierliche, geordnete und potentiell unendliche Folge von flüchtigen Datenstromelementen [GÖ03].

Datenstrommanagementsystem Ein Datenstrommanagementsystem ist ein Streaming System erweitert um Eigenschaften von Datenbankmanagementsystemen wie Anfrageverwaltung, vordefinierte Operatoren, Anfrageoptimierung und Zugriffskontrolle [CM12; Gei13].

Domäne Untersuchungsgegenstände können in Domänen kategorisiert werden. Domänen haben keinen direkten Einfluss auf die Trust-Erhebung, ermöglichen aber verschiedene Perspektiven auf diese. Konkret ermöglichen sie die Auswirkungen von Trust-Einbußen in einer Domäne auf den Trust im Gesamtsystem zu untersuchen. Beispiele für Domänen sind Strom (physisches System), Informations- und Kommunikationstechnik sowie Markt.

E

einfacher Trust-Wert Ein einfacher Trust-Wert $t_{e,\gamma}$ einer Entität e, der von einem Schätzer γ erhoben wird, ist ein Tupel der Form $t_{e,\gamma} = (\gamma, p)$, wobei $\gamma \in \Gamma$ ein Trust-Schätzer aus der Menge Γ aller Trust-Schätzer und $p \in [0, 1]$ die Wahrscheinlichkeit ist, dass die Entität e vertrauenswürdig ist.

Eingangstransformationsframework (engl. input transform. framework) Den Eingangspunkt für Messwerte und Topologien bildet das Eingangstransformationsframework. Es ist ein Framework, da die Anzahl an eingehenden Datenströmen sowie die zur Übertragung verwendeten Protokolle von Anwendungsfall zu Anwendungsfall variieren können. Die Aufgabe des Eingangstransformationsframeworks ist es, die eingehenden Daten in das Datenmodell für Topologien bzw. Datenmodell für Messwerte zu transformieren.

Elementfenster In Elementfenstern bemessen sich die Breite und der Fortschritt des Fensters nach Anzahlen von Elementen. Der Fortschritt eines Fensters ist dabei das Maß, wann ein neues Fenster startet.

exakt bestimmtes Stromsystem In einem exakt bestimmten Stromsystem gibt es genau so viele Messwerte wie nötig um die Zustandsvariablen eindeutig zu bestimmen [KML15].

F

False Data Injection Attack Bei der Einspeisung falscher Daten (engl. false data injection attack) bringen Angreifer mehrere Messgeräte unter ihre Kontrolle und haben Kenntnis über das elektrotechnische Modell des Stromsystems (Netztopologie und Leitungsimpedanzen). Dadurch ist es ihnen möglich, den geschätzten Systemzustand unbemerkt zu beeinflussen und so den Operateuren einen falschen Systemzustand glauben zu machen. Etwaige Ziele können schädliche Kontrollaktionen oder das Verbergen schädlicher Manipulationen sein [LNR11].

Feld Als Feld wird in dieser Arbeit der Teil des verteilten Stromsystems bezeichnet, der mittels Fernwirktechnik überwacht und gesteuert wird.

Fenster Fensteransätze werden verwendet, um einen endlichen Ausschnitt aus einem potentiell unendlichen Datenstrom zu betrachten.

Flexibilität Die Flexibilität beschreibt die Fähigkeit des entwickelten Systems für unterschiedliche Stromsysteme, Trust-Schätzer und verwendete Protokolle anpassbar zu sein.

funktionale Korrektheit "Die Eigenschaft eines Systems, seiner funktionalen Spezifikation zu entsprechen, unter der Bedingung, dass keine unvorhergesehenen Störungen in der Umgebung des Systems auftreten" [Ste+10; SR12].

G

Glaubwürdigkeit „Der Glaube an die Fähigkeit und den Willen eines Kooperationspartners, an einer Interaktion in einer vorteilhaften Weise teilzunehmen. Außerdem die Fähigkeit eines Systems, mit einem Benutzer konsistent und transparent zu kommunizieren" [Ste+10; SR12].

I

IEC 60870-5-104 IEC 60870-5-104 stellt einen Standard zur Datenübertragung dar, in dem Datenstruktur, Handshaking und Übertragung per TCP/IP definiert sind [Int16]. Er ist im Energiesystembereich in Europa weit verbreitet [Chr19].

IKT-System Die Datenübertragung von Messwerten und Steuerungsbefehlen erfolgt durch ein IKT-System mit Sendern und Empfängern, zwischen denen Nachrichten (Messwerte oder Steuerbefehle verpackt in Datenpakete) übermittelt werden. Zu diesem Zweck können unterschiedliche Medien, wie z.B. Kabel, und Protokolle, wie z.B. das IEC 60870-5-104, zum Einsatz kommen.

Informationssicherheit „Die Abwesenheit von Möglichkeiten, das System in einer Weise zu verwenden, die dazu führt, dass private Informationen preisgegeben werden, Daten ohne Autorisierung gelöscht oder verändert werden, oder die es erlaubt, unberechtigterweise im Namen von anderen mit dem System zu interagieren" [Ste+10; SR12].

Integrität Ein System gewährleistet die Datenintegrität „[...], wenn es Subjekten nicht möglich ist, die zu schützenden Daten unautorisiert und unbemerkt zu manipulieren" [Eck14].

K

kritischer Messwert Die Eliminierung eines kritischen Messwertes aus der Messwertemenge führt zu einem (teilweise) unüberwachbaren System. Das bedeutet, dass eine State Estimation auf Grundlage einer solchen reduzierten Messwertemenge nicht alle Zustandsvariablen bestimmen kann.

L

Lagebilderkennung Den Prozess der Identifizierung des aktuellen Zustands wird auch Sicherheitsanalyse [AE04] genannt. Man spricht aber häufig auch vom Situationsbewusstsein (engl. situation(al) awareness) [PK15]. In dieser Arbeit werden diese unterschiedlichen Perspektiven und Aspekte unter dem Begriff der Lagebilderkennung zusammengefasst.

Latenz Latenz wird in diesem Zusammenhang als die Zeitdauer zwischen dem Eintreffen eines Datenstromelements und dem Bereitstellen eines Ergebnisses definiert.

M

Masterstation Im Kontrollzentrum, auch Netzleitwarte genannt, befindet sich die (häufig zentrale) Masterstation. Sie stellt im wesentlichen eine Infrastruktur dar, die Operateure bei der Überwachung und Steuerung des Stromsystems unterstützt. Dies umfasst vor allem auch eine Mensch-Maschine-Schnittstelle, die dem Operateur alle Mess- und Systemdaten grafisch aufbereitet zur Verfügung stellt und mittels derer der Operateur Steuerbefehle ausführen kann [TM15].

multivariater Trust-Wert Ein multivariater Trust-Wert T_e einer Entität e ist ein Tupel $T_e = (T_{e,fc}, T_{e,saf}, T_{e,sec}, T_{e,r}, T_{e,c}, T_{e,u})$ bestehend aus den sechs Trust-Facetten funktionale Korrektheit ($T_{e,fc}$), Betriebssicherheit ($T_{e,saf}$), Informationssicherheit ($T_{e,sec}$), Zuverlässigkeit ($T_{e,r}$), Glaubwürdigkeit ($T_{e,c}$) und Bedienbarkeit ($T_{e,u}$).

O

OC-Trust OC-Trust ist ein anerkanntes multivariates Trust-Modell aus dem Bereich des Organic Computing. Im OC werden zumeist Multiagentensysteme betrachtet, in denen Agenten mit anderen Agenten aber auch mit Menschen kommunizieren und interagieren. Im Rahmen von OC-Trust wird Trust verstanden als „ein subjektives Konzept, das alle an einem System teilnehmenden Komponenten und Benutzer mit einbezieht und Kooperation zwischen den

Elementen verteilter Systeme ermöglicht. Es erlaubt den Elementen das Vertrauen, das sie in ihre Interaktionspartner in einem bestimmten Kontext haben, zu bemessen und entwickelt sich mit den Erfahrungen der Elemente im Laufe der Zeit fort“ [Ste+10; SR12]. Dabei tragen zu Trust die Trust-Facetten funktionale Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit, Bedienbarkeit bei.

Odysseus Odysseus ist ein Framework für maßgeschneiderte Datenstrommanagementsysteme, mit dem es möglich ist unterschiedliche, an gegebene Anwendungsszenarien angepasste, Datenstrommanagementsysteme zu erstellen [App+12; Bol+09].

Odysseus Script Odysseus Script ist eine regelbasierte Skriptsprache mit der Odysseus-Nutzer bei einigen Varationspunkten zur Laufzeit entscheiden können, welche Implementierung verwendet werden soll [Bol+10].

Open Java 104 Open Java 104 ist eine im Rahmen der vorliegenden Arbeit entwickelte Open-Source-Java-Bibliothek zum Senden, Empfangen und zur Weiterverarbeitung von Daten im IEC 60870-5-104-Standard.

P

Prädikatfenster Bei Prädikatfenstern gibt ein komplexes Prädikat vor, welche Datenstromelemente sich in dem Fenster befinden sollen [GAE06].

Prozedural Query Language Die Prozedural Query Language ist eine Odysseus-eigene, prozedurale und damit leicht erweiterbare und wartbare Anfragesprache [App+12].

Prozessinteroperabilität Die Prozessinteroperabilität beschreibt die Eigenschaft der anomaliesensitiven State Estimation mit Streaming Systemen zum einen unter Verwendung der alternative State Estimation Lagebilder mit höherem Vertrauen als ohne und zum anderen unter Verwendung einer Änderungserkennung nur Ergebnisse zu liefern, wenn sie sich in den Zustandsvariablen oder dem Trust in diese von der vorigen Ergebnismenge unterscheiden.

PSNA-Trust PSNA-Trust ist ein Trust-Modell im Kontext von cyber-physischen Energiesystemen, das Trust im Kontext der Prozessdatenakquise definiert und auf OC-Trust basiert. Dabei wird der Trust in Untersuchungsgegenstände erhoben. Die Trust-Erhebung geschieht durch Trust-Schätzer unter Berücksichtigung des Kontextes. Die Trust-Schätzer nutzen dabei zum einen Trust-Inputs und zum anderen Transformationsfunktionen.

R

redundanter Messwert Ein redundanter Messwert ist ein Messwert, der kein kritischer Messwert ist.

Remote Terminal Unit Eine Remote Terminal Unit dient zum einen bei der Datenakquise und Überwachung als Sammelpunkt für Daten von Geräten im Feld. Sensoren erfassen physikalische Größen, wie etwa Spannung oder Strom, und eine Remote Terminal Unit versendet die von diesen Geräten gesammelten Daten gebündelt mithilfe des IKT-Systems. Zum anderen erfüllt eine Remote Terminal Unit bei der Steuerung den Zweck eines Verteilungspunktes für Steuerbefehle, die mithilfe des IKT-Systems übertragen wurden. Sie interpretiert die Befehle und setzt sie durch Aktoren, wie z.B. einen Leistungsschalter, um [TM15].

S

SCADA-System SCADA-Systeme sind „eine Sammlung von Gerätschaften, die einen Operateur mit genügend Informationen versorgt, um den Status eines bestimmten Gerätes oder Prozesses in einer Fernwirkstation zu bestimmen und Aktionen bzgl. dieses Gerätes oder Prozesses zu veranlassen, ohne körperlich anwesend zu sein“ [TM15] (aus dem Englischen übersetzt).

Sessionfenster Sessionfenster haben weder eine feste Größe noch einen festen Fortschritt. Vielmehr wird ein neues Fenster für ein Element erstellt, wenn zurzeit keines offen ist. Geschlossen wird das Fenster bei anhaltender Inaktivität, d.h. es kommen für einen bestimmten Zeitraum keine neuen Datenstromelemente hinzu.

Skalierbarkeit Die Skalierbarkeit beschreibt die Fähigkeit des entwickelten Systems für unterschiedliche Anzahlen an Fernwirkverbindungen und Trust-Schätzer zu skalieren.

State Estimation Auf Basis des Netzwerkmodells und der zur Verfügung stehenden Messwerte wird eine Schätzung der Zustandsvariablen vorgenommen [AE04].

Streaming System Ein Streaming System ist ein System, das Daten ereignisbasiert und im Hauptspeicher verarbeitet. Dabei werden die Daten nur so lange vorgehalten, wie unbedingt notwendig, und nur einmal verarbeitet [SÇZ05].

T

technische Interoperabilität Die technische Interoperabilität ist die Fähigkeit der Nachrichten im IEC 60870-5-104-Standard zu unterstützen und um die Unterstützung weiterer Protokolle erweiterbar zu sein.

Transformationsfunktion Transformationsfunktionen sind mathematische Funktionen um Trust-Inputs in eine Wahrscheinlichkeit umzuwandeln, mit der der Untersuchungsgegenstand bzgl. einer bestimmten Trust-Facette vertrauenswürdig ist.

Trust Trust ist ein subjektives, kontextabhängiges und multivariates Empfinden einer Entität bezüglich ihrer funktionalen Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit.

Trust Assessment Pyramid Die Trust Assessment Pyramid (deutsch: Vertrauenserhebungspyramide) ist eine Darstellungsform des Informationsmodells zur Erhebung von Trust in PSNA-Trust.

Trust-Facette Eine Trust-Facette $T_{e,f}$ mit $f \in \{fc, saf, sec, r, c, u\}$ einer Entität e ist eine Menge von einfachen Trust-Werten für diese Entität und die entsprechende Trust-Facette: $T_{e,f} = \{t_{e,\gamma} | t_{e,\gamma} \xrightarrow{\gamma} f\} = \{(\gamma, p_{e,\gamma}) | (\gamma, p_{e,\gamma}) \xrightarrow{\gamma} f\}$ mit $t_{e,\gamma} \xrightarrow{\gamma} f :=$ γ ordnet $t_{e,\gamma} f$ zu. fc, saf, sec, r, c, u stehen dabei für funktionale Korrektheit, Betriebssicherheit, Informationssicherheit, Zuverlässigkeit, Glaubwürdigkeit und Bedienbarkeit.

Trust-Input Trust-Inputs sind Informationen von Trust-Quellen, die Trust-Schätzer zur Erhebung des Trusts in Untersuchungsgegenstände verwenden.

Trust-Quelle Trust-Quellen stellen Trust-Inputs, die Trust-Schätzer zur Erhebung des Trusts in Untersuchungsgegenstände verwenden, bereit.

Trust-Schätzer Trust-Schätzer bestimmen den Trust in Untersuchungsgegenstände mithilfe von Trust-Inputs.

Trust-Wahrscheinlichkeit Eine Trust-Wahrscheinlichkeit $p \in \{R | 0 \leq p \leq 1\}$ ist eine Wahrscheinlichkeit mit der ein Untersuchungsgegenstand vertrauenswürdig ist.

U

überbestimmtes Stromsystem Sind mehr Messwerte vorhanden als benötigt, um die Zustandsvariablen zu bestimmen, so spricht man von einem überbestimmten Stromsystem [KML15].

Unsicherheitsanalyse Der multivariate Trust-Wert pro Messwert wird zu einer Trust-Wahrscheinlichkeit im Wertebereich $[0; 1]$ aggregiert und in eine Standardabweichung umgewandelt. Mit diesen Standardabweichungen für die Messwerte werden dann Unsicherheiten für die Zustandsvariablen berechnet.

Unteranfrage Unteranfragen werden in Odysseus durch spezielle Unteranfrageoperatoren erstellt, die verschachtelte Anfragepläne ausblenden. Die ausgeblendeten Unteranfragen sind normale Anfragen. Sie werden also zusammen mit der Hauptanfrage installiert und gestartet.

unterbestimmtes Stromsystem Sind weniger Messwerte vorhanden als benötigt, um die Zustandsvariablen zu bestimmen, so spricht man von einem unterbestimmten Stromsystem [KML15].

Untersuchungsgegenstand Untersuchungsgegenstände sind bei PSNA-Trust Entitäten, für die Trust erhoben werden soll.

V

Vertrauen Siehe Trust.

Vertrauenswürdigkeit Vertrauenswürdigkeit beschreibt eine Eigenschaft einer Entität, während Vertrauen einer Entität entgegengebracht wird.

Z

Zeitfenster In Zeitfenstern bemessen sich die Breite und der Fortschritt des Fensters nach Zeiteinheiten. Der Fortschritt eines Fensters ist dabei das Maß, wann ein neues Fenster startet.

Zustandsvariable Die Zustandsvariablen sind die komplexen Spannungen an den Netzknoten [AE04].

Zuverlässigkeit „Die Eigenschaft eines Systems, selbst bei auftretenden Störungen oder teilweisen Ausfällen für eine spezifizierte Zeit verfügbar zu bleiben“ [Ste+10; SR12].

Abkürzungsverzeichnis

Zahlen

104er IEC 60870-5-104

A

ADF Anomalieerkennungsframework (engl. anomaly detection framework)

APCI Application Protocol Control Information

APDU Application Protocol Data Unit

ASDU Application Service Data Unit

ASE Adaptive State Estimator

ASSE anomaliesensitive State Estimation

ASSESS anomaliesensitive State Estimation mit Streaming Systemen

C

CIGRE12MV reduziertes CIGRE Mittelspannungsnetz mit 12 Sammelschienen

CIM Common Information Model

CPES cyber-physisches Energiesystem

CQL Continuous Query Language

D

DBMS Datenbankmanagementsystem

DSMS Datenstrommanagementsystem

F

FDIA False Data Injection Attack

I

IDS Intrusion Detection System

IEEE118HV IEEE Hochspannungsnetz mit 118 Sammelschienen

IEEE39HV IEEE Hochspannungsnetz mit 39 Sammelschienen

IKT Informations- und Kommunikationstechnik

ITF Eingangstransformationsframework (engl. input transform. framework)

M

M-Modell Datenmodell für Messwerte

M-T-Modell Datenmodell für Messwerte mit Trust-Wert

O

OC Organic Computing

OJ104 Open Java 104

OSGi Open Services Gateway initiative

OTF Ausgangstransformationsframework (engl. output transf. framework)

P

PQL Prozedural Query Language

PSNA Power System Network Assessment

R

RTU Remote Terminal Unit

S

SCADA Supervision, Control, and Data Acquisition

T

T-Modell Datenmodell für Topologien

U

UML Unified Modeling Language

Nomenklatur

default

θ_i Spannungsphasenwinkel an Sammelschiene i

E

$\boldsymbol{e}$ m-elementiger Fehlervektor

G

$\boldsymbol{G}(\boldsymbol{x}^k)$ Verstärkungsmatrix (engl. gain matrix)

H

$\boldsymbol{H}$ Jacobi-Matrix von $\boldsymbol{h}(\boldsymbol{x})$

$\boldsymbol{h}(\boldsymbol{x})$ nichtlineare Funktion, die den komplexen Zustandsvektor auf den komplexen Messwertevektor abbildet

P

P_i Wirkleistungseinspeisung an Sammelschiene i

Q

Q_i Blindleistungseinspeisung an Sammelschiene i

T

$\boldsymbol{T}_e$ Multivariater Trust-Wert für die Entität e

$t_{e,\gamma}$ Einfacher Trust-Wert des Trust-Schätzers γ für die Entität e

$\boldsymbol{T}_{e,f}$ Trust-Facette f für die Entität e

U

$u(\theta_i)$ Unsicherheit des Spannungsphasenwinkels an Sammelschiene i

$u(V_i)$ Unsicherheit der Spannungsmagnitude an Sammelschiene i

V

V_i Spannungsmagnitude an Sammelschiene i

X

$\boldsymbol{x}$ 2n-elementiger Zustandsvektor

Y

$\boldsymbol{Y}$ Admittanzmatrix ($n \times n$)

Z

$\boldsymbol{z}$ m-elementiger Messwertevektor

Abbildungsverzeichnis

Tabellenverzeichnis

Skriptverzeichnis

Publikationsverzeichnis

[Bra+19a] Michael Brand, Shoaib Ansari, Felipe Castro et al. “A Framework for the Integration of ICT-relevant Data in Power System Applications”. In: *Proceedings of the 2019 IEEE Milan PowerTech*. Milan: IEEE, 2019, S. 1–6 (zitiert auf den Seiten 9, 12, 77, 114).

[Bra+20] Michael Brand, Davood Babazadeh, Carsten Krüger, Björn Siemers und Sebastian Lehnhoff. “Trust assessment of power system states”. In: *Energy Informatics* 3.1 (2020), S. 18 (zitiert auf den Seiten 1, 5, 9, 12, 44, 48, 49, 51, 71–74, 78, 87).

[BBL21] Michael Brand, Davood Babazadeh und Sebastian Lehnhoff. “Trust in Power System State Variables based on Trust in Measurements”. In: *Proceedings of the 2021 IEEE Madrid PowerTech*. Madrid: IEEE, 2021, S. 1–6 (zitiert auf den Seiten 1, 9, 12, 44, 51, 53, 71, 82, 87).

[Bra+19b] Michael Brand, Davood Babazadeh, Sebastian Lehnhoff und Dominik Engel. “Trust in control: a trust model for power system network assessment”. In: *EPJ Web of Conferences* 217 (2019), S. 01008 (zitiert auf den Seiten 1, 9, 12, 44, 52).

[Bra+21] Michael Brand, Felipe Castro, Batoul Hage Hassen et al. “Demo abstract: A Platform to Assess the Trust in Power System Components, Data, and Services”. In: *Abstracts of the 10th DACH+ Conference on Energy Informatics*. Springer, 2021, S. 26 (zitiert auf den Seiten 9, 12, 113–116).

[BEL23] Michael Brand, Dominik Engel und Sebastian Lehnhoff. “ASSESS – Anomaly Sensitive State Estimation with Streaming Systems”. In: Bd. 6. 19. Springer Nature, 2023, S. 1–20 (zitiert auf den Seiten 1, 9, 12, 89, 119).

[BNL23] Michael Brand, Anand Narayan und Sebastian Lehnhoff. “Applying Trust for Operational States of ICT-Enabled Power Grid Services (in Begutachtung)”. In: *ACM Trans. Auton. Adapt. Syst.* ?? (2023) (zitiert auf den Seiten 9, 13, 141).

Literaturverzeichnis

[AE04] Ali Abur und Antonio Gomez Exposito. *Power System State Estimation: Theory and Implementation*. CRC Press, 2004 (zitiert auf den Seiten 2, 23–29, 37, 216, 220, 222, 224, 241).

[Ahm+05] Yanif Ahmad, Bradley Berg, Uğur Cetintemel et al. “Distributed operation in the borealis stream processing engine”. In: *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 2005, S. 882–884 (zitiert auf Seite 90).

[AA20] Montdher Alabadi und Zafer Albayrak. “Q-Learning for Securing Cyber-Physical Systems: A survey”. In: *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*. Ankara: IEEE, 2020 (zitiert auf den Seiten 2–4).

[And+13] Gerrit Anders, Florian Siefert, Nizar Msadek et al. *TEMAS - A Trust-Enabling Multi-Agent System for Open Environments*. Techn. Ber. Augsburg: Universität Augsburg, 2013 (zitiert auf den Seiten 59, 60).

[And+11] Gerrit Anders, Jan-Philipp Steghöfer, Florian Siefert und Wolfgang Reif. “Patterns to Measure and Utilize Trust in Multi-Agent Systems”. In: *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*. Ann Arbor, MI, USA: IEEE, 2011, S. 35–40 (zitiert auf den Seiten 40, 41, 45, 60).

[Ans+19] Shoaib Ansari, Felipe Castro, Dennis Weller, Davood Babazadeh und Sebastian Lehnhoff. “Towards virtualization of operational technology to enable large-scale system testing”. In: *IEEE EUROCON 2019-18th International Conference on Smart Technologies*. IEEE. 2019, S. 1–5 (zitiert auf Seite 114).

[App+12] H.-Jürgen Appelrath, Dennis Geesen, Marco Grawunder, Timo Michelsen und Daniela Nicklas. “Odysseus: A Highly Customizable Framework for Creating Efficient Event Stream Management Systems”. In: *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems*. ACM, 2012, S. 367–368 (zitiert auf den Seiten 89, 90, 92, 221).

[ABW02] Arvind Arasu, Shivnath Babu und Jennifer Widom. *An abstract semantics and concrete language for continuous queries over streams and relations*. Techn. Ber. Stanford, 2002 (zitiert auf den Seiten 34, 38, 92).

[Bab+22] Davood Babazadeh, Payam Teimourzadeh Baboli, Michael Brand et al. “Resilience of Smart Integrated Energy Systems”. In: *Handbook of Smart Energy Systems*. Springer, 2022, S. 1–27 (zitiert auf den Seiten 9, 12, 142).

[BO12] Yuksel Ozan Basciftci und Fusun Ozguner. “Trust aware particle filters for autonomous vehicles”. In: *2012 IEEE International Conference on Vehicular Electronics and Safety, ICVES 2012* (2012), S. 50–54 (zitiert auf den Seiten 72, 73).

[Bol+10] Andre Bolles, Dennis Geesen, Marco Grawunder et al. “Sensordatenverarbeitung mit dem Open Source Datenstrommanagementframework Odysseus”. In: *GI Jahrestagung (2)*. 2010, S. 404–409 (zitiert auf den Seiten 92, 221).

[Bol+09] Andre Bolles, Marco Grawunder, Jonas Jacobi, Daniela Nicklas und Hans-Jürgen Appelrath. “Odysseus: Ein Framework für maßgeschneiderte Datenstrommanagementsysteme”. In: *GI Jahrestagung*. 2009, S. 2000–2014 (zitiert auf den Seiten 89–91, 221).

[Bra+19a] Michael Brand, Shoaib Ansari, Felipe Castro et al. “A Framework for the Integration of ICT-relevant Data in Power System Applications”. In: *Proceedings of the 2019 IEEE Milan PowerTech*. Milan: IEEE, 2019, S. 1–6 (zitiert auf den Seiten 9, 12, 77, 114).

[Bra+20] Michael Brand, Davood Babazadeh, Carsten Krüger, Björn Siemers und Sebastian Lehnhoff. “Trust assessment of power system states”. In: *Energy Informatics* 3.1 (2020), S. 18 (zitiert auf den Seiten 1, 5, 9, 12, 44, 48, 49, 51, 71–74, 78, 87).

[BBL21] Michael Brand, Davood Babazadeh und Sebastian Lehnhoff. “Trust in Power System State Variables based on Trust in Measurements”. In: *Proceedings of the 2021 IEEE Madrid PowerTech*. Madrid: IEEE, 2021, S. 1–6 (zitiert auf den Seiten 1, 9, 12, 44, 51, 53, 71, 82, 87).

[Bra+19b] Michael Brand, Davood Babazadeh, Sebastian Lehnhoff und Dominik Engel. “Trust in control: a trust model for power system network assessment”. In: *EPJ Web of Conferences* 217 (2019), S. 01008 (zitiert auf den Seiten 1, 9, 12, 44, 52).

[Bra+21] Michael Brand, Felipe Castro, Batoul Hage Hassen et al. “Demo abstract: A Platform to Assess the Trust in Power System Components, Data, and Services”. In: *Abstracts of the 10th DACH+ Conference on Energy Informatics*. Springer, 2021, S. 26 (zitiert auf den Seiten 9, 12, 113–116).

[BEL23] Michael Brand, Dominik Engel und Sebastian Lehnhoff. “ASSESS – Anomaly Sensitive State Estimation with Streaming Systems”. In: Bd. 6. 19. Springer Nature, 2023, S. 1–20 (zitiert auf den Seiten 1, 9, 12, 89, 119).

[BNL23] Michael Brand, Anand Narayan und Sebastian Lehnhoff. “Applying Trust for Operational States of ICT-Enabled Power Grid Services (in Begutachtung)”. In: *ACM Trans. Auton. Adapt. Syst.* ?? (2023) (zitiert auf den Seiten 9, 13, 141).

[BSS05] Gert Brettlecker, Heiko Schuldt und Hans-Jörg Schek. “Towards Reliable Data Stream Processing with OSIRIS-SE.” In: *BTW*. 2005, S. 405–414 (zitiert auf Seite 90).

[Cai+19] Xingpu Cai, Qi Wang, Yi Tang und Lei Zhu. “Review of Cyber-attacks and Defense Research on Cyber Physical Power System”. In: *2019 IEEE Sustainable Power and Energy Conference (iSPEC)*. Beijing: IEEE, 2019, S. 487–492 (zitiert auf den Seiten 2–4).

[Chr19] Justyna J. Chromik. *Process-aware SCADA Traffic Monitoring: A Local Approach*. 2019, S. 215 (zitiert auf den Seiten 19, 20, 219).

[CMW19] Gonzalo Constante, Christian Moya und Jiankang Wang. “Semantic-based detection architectures against monitoring-control attacks in power grids”. In: *2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. Beijing: IEEE, 2019 (zitiert auf Seite 56).

[CM12] Gianpaolo Cugola und Alessandro Margara. “Processing Flows of Information”. In: *ACM Computing Surveys* 44.3 (2012) (zitiert auf den Seiten 11, 33, 38, 217).

[Cui+12] Shuguang Cui, Z. Han, Soummya Kar et al. “Coordinated Data-Injection Attack and Detection in the Smart Grid: A detailed look at enriching detection solutions”. In: *IEEE Signal Processing Magazine* 29.5 (2012), S. 106–115 (zitiert auf Seite 4).

[Dav+15] Katherine R. Davis, Charles M. Davis, Saman A. Zonouz et al. “A Cyber-Physical Modeling and Assessment Framework for Power Grid Infrastructures”. In: *IEEE Transactions on Smart Grid* 6.5 (2015), S. 2464–2475 (zitiert auf Seite 56).

[DFD19] Hugo F.M. De Figueiredo, Matheus K. Ferst und Gustavo W. Denardin. “An Overview about Detection of Cyber-Attacks on Power SCADA Systems”. In: *2019 IEEE 15th Brazilian Power Electronics Conference and 5th IEEE Southern Power Electronics Conference (COBEP/SPEC)*. Santos: IEEE, 2019 (zitiert auf den Seiten 2–4).

[Deh+19] Kaveh Dehghanpour, Zhaoyu Wang, Jianhui Wang, Yuxuan Yuan und Fankun Bu. “A survey on state estimation techniques and challenges in smart distribution systems”. In: *IEEE Transactions on Smart Grid* 10.2 (2019), S. 2312–2322 (zitiert auf Seite 4).

[DLA15] Aline Dresch, Daniel Pacheco Lacerda und José Antônio Valle Antunes. “Design Science Research”. In: *Design Science Research: A Method for Science and Technology Advancement*. Cham: Springer International Publishing, 2015, S. 67–102 (zitiert auf Seite 9).

[Eck14] Claudia Eckert. *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. 9. Aufl. Walter de Gruyter GmbH & Co. KG, 2014, S. 1004 (zitiert auf den Seiten 5, 219).

[Ele16] Electricity Information Sharing and Analysis Center (E-ISAC). “Analysis of the cyber attack on the Ukrainian power grid – Defense Use Case”. In: 388 (2016), S. 1–29 (zitiert auf Seite 143).

[FBY20] Mohamed Amine Ferrag, Messaoud Babaghayou und Mehmet Akif Yazici. “Cyber security for fog-based smart grid SCADA systems: Solutions and challenges”. In: *Journal of Information Security and Applications* 52 (2020) (zitiert auf den Seiten 2–4).

[Gei13] Sandra Geisler. “Data Stream Management Systems”. In: *Proceedings of the 2010 Data Exchange, Integration, and Streams*. Hrsg. von Phokion G Kolaitis, Maurizio Lenzerini und Nicole Schweikardt. Leipzig, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013, S. 275–304 (zitiert auf den Seiten 11, 33, 38, 217).

[GAE06] Thanaa M Ghanem, Walid G Aref und Ahmed K Elmagarmid. “Exploiting Predicate-window Semantics over Data Streams”. In: *ACM SIGMOD Record* 35.1 (2006), S. 3–8 (zitiert auf den Seiten 36, 221).

[GH15] Sanjay Goel und Yuan Hong. “Security Challenges in Smart Grid Implementation”. In: *Smart Grid Security*. London: Springer, 2015, S. 1–39 (zitiert auf den Seiten 2–4).

[GÖ03] Lukasz Golab und M Tamer Özsu. “Issues in Data Stream Management”. In: *ACM Sigmod Record* 32.2 (2003), S. 5–14 (zitiert auf den Seiten 33, 34, 38, 217).

[GUD17] Marion Gottschalk, Mathias Uslar und Christina Delfs. *The use case and smart grid architecture model approach: the IEC 62559-2 use case template and the SGAM applied in various domains*. Springer, 2017 (zitiert auf Seite 144).

[GD20] Muhammed Zekeriya Gunduz und Resul Das. “Cyber-security on smart grid: Threats and potential solutions”. In: *Computer Networks* 169 (2020), S. 107094 (zitiert auf den Seiten 2–4).

[Has+21] Batoul Hage Hassan, Anand Narayan, Davood Babazadeh, Marcel Klaes und Sebastian Lehnhoff. “Performance assessment of state estimation in cyber-physical energy systems”. In: *2021 IEEE Madrid PowerTech*. IEEE. 2021, S. 1–6 (zitiert auf Seite 141).

[HL04] Keith E Holbert und Kang Lin. “Reducing State Estimation Uncertainty Through Fuzzy Logic Evaluation of Power System measurements”. In: *2004 International Conference on Probabilistic Methods Applied to Power Systems*. Ames, IA, USA: IEEE, 2004, S. 205–211 (zitiert auf Seite 58).

[Hua+12] Yih-Fang Huang, Stefan Werner, Jing Huang, Neelabh Kashyap und Vijay Gupta. “State Estimation in Electric Power Grids: Meeting new challenges presented by the requirements of the future grid”. In: *IEEE Signal Processing Magazine* 29.5 (2012), S. 33–43 (zitiert auf den Seiten 4, 29).

[Hum+17] Abdulmalik Humayed, Jingqiang Lin, Fengjun Li und Bo Luo. “Cyber-Physical Systems Security - A Survey”. In: *IEEE Internet of Things Journal* 4.6 (2017), S. 1802–1831 (zitiert auf den Seiten 2–4).

[Hwa+05] Jeong-Hyon Hwang, Magdalena Balazinska, Alex Rasin et al. “High-Availability Algorithms for Distributed Stream Processing”. In: *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*. IEEE, 2005, S. 779–790 (zitiert auf Seite 90).

[Int15] International Electrotechnical Commission. *IEC60870-5-101:2003+AMD1:2015 CSV Consolidated version*. 2015 (zitiert auf den Seiten 20–22, 37).

[Int16] International Electrotechnical Commission. *IEC60870-5-104:2006+AMD1:2016 CSV Consolidated version*. 2016 (zitiert auf den Seiten 20, 21, 37, 216, 219).

[Kar+20] Hadis Karimipour, Pirathayini Srikantha, Hany Farag und Jin Wei-Kocsis. *Security of Cyber-Physical Systems*. Hrsg. von Hadis Karimipour, Hany Farag, Pirathayini Srikantha und Jin Wei-Kocsis. Cham: Springer Nature Switzerland AG, 2020 (zitiert auf den Seiten 2–4).

[KSZ13] Andrew J Kornecki, Nary Subramanian und Janusz Zalewski. "Studying interrelationships of safety and security for software assurance in cyber-physical systems: Approach based on bayesian belief networks". In: *2013 Federated Conference on Computer Science and Information Systems*. IEEE. 2013, S. 1393–1399 (zitiert auf Seite 2).

[Krä07] Jürgen Krämer. "Continuous Queries over Data Streams - Semantics and Implementation". Diss. University of Marburg, 2007 (zitiert auf den Seiten 33, 36, 38).

[KML15] O. Krause, D. Martin und S. Lehnhoff. "Under-Determined WLMS State Estimation". In: *Proceedings of the Asia-Pacific Power and Energy Engineering Conference*. Brisbane, Australia: IEEE, 2015, S. 1–6 (zitiert auf den Seiten 29, 30, 75, 95, 218, 223, 224).

[KL12] Olav Krause und Sebastian Lehnhoff. "Generalized Static-State Estimation – Approach and Key Features". In: *Proceedings of the 22nd Australasian Universities Power Engineering Conference*. Denpasar, Indonesia: IEEE, 2012, S. 1–6 (zitiert auf den Seiten 29, 75, 95).

[Lew19] Andre Lewis. *Understanding Linux CPU Load - when should you be worried?* `https://scoutapm.com/blog/understanding-load-averages`. Accessed: 2020-05-28. 2019 (zitiert auf Seite 79).

[Liu+15] Ting Liu, Yanan Sun, Yang Liu et al. "Abnormal traffic-indexed state estimation: A cyber-physical fusion approach for Smart Grid attack detection". In: *Future Generation Computer Systems* 49 (2015), S. 94–103 (zitiert auf den Seiten 72, 73, 78, 128).

[LNR11] Yao Liu, Peng Ning und Michael K. Reiter. "False Data Injection Attacks against State Estimation in Electric Power Grids". In: *ACM Transactions on Information and System Security (TISSEC)* 14.1 (2011), S. 1–33 (zitiert auf den Seiten 4, 218).

[LL18] Ying Liu und Chunguang Li. "Secure Distributed Estimation Over Wireless Sensor Networks Under Attacks". In: *IEEE Transactions on Aerospace and Electronic Systems* 54.4 (2018), S. 1815–1831 (zitiert auf Seite 39).

[MX20] Liang Ma und Gang Xu. "Distributed resilient voltage and reactive power control for islanded microgrids under false data injection attacks". In: *Energies* 13.15 (2020) (zitiert auf Seite 39).

[MHB19] Magdi S. Mahmoud, Mutaz M. Hamdan und Uthman A. Baroudi. “Modeling and control of Cyber-Physical Systems subject to cyber attacks: A survey of recent advances and challenges”. In: *Neurocomputing* 338 (2019), S. 101–115 (zitiert auf den Seiten 2–4).

[Mer10] Stephan Merkli. “Streaming in the Cloud”. Diss. Master Thesis, Eidgenössische Technische Hochschule, 2009-2010, 2010 (zitiert auf Seite 90).

[MS15] Yilin Mo und Bruno Sinopoli. “Secure Estimation in the Presence of Integrity Attacks”. In: *IEEE Transactions on Automatic Control* 60.4 (2015), S. 1145–1151 (zitiert auf Seite 4).

[MHW18] Christian Moya, Junho Hong und Jiankang Wang. “Application of correlation indices on intrusion detection systems: Protecting the power grid against coordinated attacks”. In: *arXiv* (2018), S. 1–10 (zitiert auf Seite 56).

[MF17] Devaprakash Muniraj und Mazen Farhood. “A framework for detection of sensor attacks on small unmanned aircraft systems”. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. Miami: IEEE, 2017, S. 1189–1198 (zitiert auf Seite 57).

[Mus+20] Aquib Mustafa, Binod Poudel, Ali Bidram und Hamidreza Modares. “Detection and Mitigation of Data Manipulation Attacks in AC Microgrids”. In: *IEEE Transactions on Smart Grid* 11.3 (2020), S. 2588–2603 (zitiert auf Seite 39).

[Ody] Odysseus-Team. *Odysseus Dokumentation*. `odysseus.uni-oldenburg.de`; zuletzt besucht: 19.12.2020 (zitiert auf den Seiten 90, 91, 93, 97–99, 113).

[Pai+89] MA Pai, T Athay, R Podmore und S Virmani. “IEEE 39-Bus System”. In: (1989) (zitiert auf Seite 76).

[Pan13] Mathaios Panteli. “Impact of ICT Reliability and Situation Awareness on Power System Blackouts”. Doctor Thesis. The University of Manchester, 2013, S. 270 (zitiert auf Seite 2).

[PK15] Mathaios Panteli und Daniel S. Kirschen. “Situation awareness in power systems: Theory, challenges and applications”. In: *Electric Power Systems Research* 122 (2015), S. 140–151 (zitiert auf den Seiten 23, 220).

[Pef+06] Ken Peffers, Tuure Tuumanen, Charles E Gengler et al. “The Design Science Research Process: A Model for Producing and Presenting Information Systems Research”. In: *Proceedings of the first international conference on design science research in information systems and technology*. 2006, S. 83–106 (zitiert auf den Seiten 9, 10, 13).

[PB14] Y. Pillitteri, Victoria und Tanya L. Brewer. *Guidelines for smart grid cybersecurity*. Techn. Ber. 2014, S. 290 (zitiert auf den Seiten 2, 4).

[RK17] Gelli Ravikumar und Shrikrishna A. Khaparde. “A Common Information Model Oriented Graph Database Framework for Power Systems”. In: *IEEE Transactions on Power Systems* 32.4 (2017), S. 2560–2569 (zitiert auf den Seiten 62–64, 241).

[RB15] Christine Rosinger und Sebastian Beer. “Glaubwürdigkeit in dynamischen Wirkleistungsverbünden”. In: *Informatik-Spektrum* 38.2 (2015), S. 103–110 (zitiert auf den Seiten 41, 42).

[RUS13] Christine Rosinger, Mathias Uslar und Jürgen Sauer. “Threat Scenarios to evaluate Trustworthiness of Multi-agents in the Energy Data Management”. In: *7th International Conference on Environmental Informatics for Environmental Protection, Sustainable Development and Risk Management, EnviroInfo 2013, Hamburg, Germany, September 2-4, 2013. Proceedings*. Hamburg, Germany: Shaker, 2013, S. 258–264 (zitiert auf den Seiten 41, 42).

[RUS14] Christine Rosinger, Mathias Uslar und Jürgen Sauer. “Using Information Security as a Facet of Trustworthiness for Self-Organizing Agents in Energy Coalition Formation Processes”. In: *EnviroInfo*. Oldenburg, Germany: BIS-Verlag, 2014, S. 373–380 (zitiert auf den Seiten 41–44).

[SG16] Bashar Sabeeh und Chin Gan. “Power System Frequency Stability and Control: Survey”. In: 11 (Mai 2016), S. 5688–5695 (zitiert auf Seite 22).

[Sch+19] Jan Sören Schwarz, Tobias Witt, Astrid Nieße et al. “Towards an Integrated Development and Sustainability Evaluation of Energy Scenarios Assisted by Automated Information Exchange”. In: *Communications in Computer and Information Science* 921.Mcdm (2019), S. 3–26 (zitiert auf den Seiten 47–49).

[SFL22] Björn Siemers, Lars Fischer und Sebastian Lehnhoff. “A Trust Model in Control Systems to Enhance and Support Cybersecurity”. In: *2022 IEEE 7th International Energy Conference (ENERGYCON)*. IEEE. 2022, S. 1–6 (zitiert auf den Seiten 142, 143).

[Ste+10] Jan-Philipp Steghöfer, Rolf Kiefhaber, Karin Leichtenstern et al. “Trustworthy organic computing systems: Challenges and perspectives”. In: Springer Berlin Heidelberg, 2010, S. 62–76 (zitiert auf den Seiten 10, 30, 31, 37, 216, 217, 219, 221, 224).

[SR12] Jan-Philipp Steghöfer und Wolfgang Reif. “Die Guten, die Bösen und die Vertrauenswürdigen - Vertrauen im Organic Computing”. In: *Infor-matik-Spektrum* 35.2 (2012), S. 119–131 (zitiert auf den Seiten 5, 10, 30–32, 37, 216, 217, 219, 221, 224).

[SÇZ05] Michael Stonebraker, Uğur Çetintemel und Stan Zdonik. “The 8 Requirements of Real-Time Stream Processing”. In: *SIGMOD Record* 34.4 (2005), S. 42–47 (zitiert auf den Seiten 11, 33, 38, 222).

[TM15] Mini S Thomas und John D McDonald. *Power System SCADA and Smart Grids*. 1. Aufl. Boca Raton, FL: CRC Press, 2015, S. 313 (zitiert auf den Seiten 8, 15–18, 20, 36, 220, 222).

[Wan+20] Jiankang Wang, Gonzalo Constante, Christian Moya und Junho Hong. “Semantic analysis framework for protecting the power grid against monitoring-control attacks”. In: *IET Cyber-Physical Systems: Theory and Applications* 5.1 (2020), S. 119–126 (zitiert auf Seite 56).

[WS18] Jingyu Wang und Dongyuan Shi. “Cyber-Attacks Related to Intelligent Electronic Devices and Their Countermeasures: A Review”. In: *2018 53rd International Universities Power Engineering Conference (UPEC)*. Glasgow: IEEE, 2018 (zitiert auf den Seiten 2–4).

[Wol92] David H Wolpert. “Stacked generalization”. In: *Neural networks* 5.2 (1992), S. 241–259 (zitiert auf Seite 103).

[Wüt08] G. Wütherich. *Die OSGi Service Platform: Eine Einführung mit Eclipse Equinox*. Dpunkt.Verlag GmbH, 2008 (zitiert auf Seite 90).

[Xie+19] Bin Xie, Chen Peng, Minjing Yang, Xiaobing Kong und Tengfei Zhang. “A novel trust-based false data detection method for power systems under false data injection attacks”. In: *Journal of the Franklin Institute* (2019), S. 1–18 (zitiert auf den Seiten 39, 40).

[Xin20] Liudong Xing. “Cascading Failures in Internet of Things: Review and Perspectives on Reliability and Resilience”. In: *IEEE Internet of Things Journal* 4662.c (2020), S. 1–1 (zitiert auf den Seiten 2–4).

[XN15] Kaiqi Xiong und Peng Ning. “Cost-Efficient and Attack-Resilient Approaches for State Estimation in Power Grids”. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*. Salamanca, Spain: ACM, 2015, S. 2192–2197 (zitiert auf Seite 4).

Index

Erklärung

Hiermit versichere ich, dass ich diese Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Außerdem versichere ich, dass ich die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis an der Carl von Ossietzky Universität Oldenburg und den DFG-Richtlinien festgelegt sind, befolgt habe. Des Weiteren habe ich im Zusammenhang mit dem Promotionsvorhaben keine kommerziellen Vermittlungs- oder Beratungsdienste in Anspruch genommen.

Oldenburg, den 12. Oktober 2023

Michael Brand