Institut für Visualisierung und Interaktive Systeme

Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Masterarbeit

Visuelle Analyse von Netzwerkverkehr in Unternehmensnetzen

Tina Tremel

Studiengang:	Informatik
Prüfer/in:	Prof. Dr. Thomas Ertl
Betreuer/in:	Dr. Steffen Koch, Dr. Jochen Kögel, Sebastian Meier, Dr. Dennis Thom
Beginn am:	21. Februar 2018
Beendet am:	04. September 2018

Kurzfassung

Heutzutage sind schnelle, stabile und sichere Unternehmensnetze essentiell für den wirtschaftlichen Erfolg der meisten Unternehmen. Immer größer und komplexer werdende Unternehmensnetze erfordern jedoch auch den Einsatz umfangreicher Netzwerk-Monitoring Systeme, die die Netzwerkadministratoren bei der Betreuung, Überwachung und dem Management der Netze unterstützen. Viele Netzwerk-Monitoring Systeme arbeiten allerdings nicht mit fortgeschrittenen visuellen und automatisierten Mitteln um die enormen anfallenden Datenmengen im Netzwerk angemessen aufzubereiten und so für die Anwender/-innen analysierbar zu machen.

In dieser Arbeit wird ein Lösungsansatz vorgestellt, der den Anwender/-innen die visuelle Analyse von Netzwerkdaten ermöglicht. Die Arbeit befasst sich dabei konkret mit der Forschungsfrage, wie Darstellungen von aggregierten Zeitreihen ergänzt werden können, damit zeitabhängige multivariate Netzwerkdaten vollständig dargestellt und hinsichtlich Anomalien untersucht werden können. Für diesen Zweck wurde ein interaktiver, webbasierter Ansatz mit koordinierten Ansichten für die visuelle Analyse von Netzwerkdaten konzipiert und in Form eines Softwareprototypen umgesetzt. Dieser stellt den zeitlichen Bezug der Netzwerkdaten in den Fokus der Analyse und schafft durch mehrere hierarchisch organisierte Ansichten der Netzwerkdaten ein mentales Modell des Datenflusses. Zudem können interaktive Filter auf die Daten angewendet werden, die durch Selektion der Daten direkt in den Ansichten realisierbar sind.

Aus den Ergebnissen der Evaluation ging hervor, dass das Konzept den meisten in dieser Arbeit definierten konzeptionellen Anforderungen entspricht und speziell für das Erkennen und Klassifizieren auffälliger Verkehrsvolumina gut geeignet ist. Die Skalierbarkeit des erarbeiteten Lösungsansatzes ist dabei vielversprechend, sodass das Konzept auch für größere Datenmengen geeignet erscheint und Erweiterungen des Ansatzes auf weitere Gebiete des Netzwerkmanagements, wie die Netzwerksicherheit, aussichtsreich erscheinen.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Verzeichnis der Listings

1. Einleitung

Computernetzwerke sind aus der heutigen Welt nicht mehr wegzudenken. So wie das Internet längst Einzug in den Alltag der Menschen gehalten hat, gibt es kaum noch Unternehmen, die ohne eigenes Unternehmensnetz auskommen. Immer größer und komplexer werdende Unternehmensnetze machen es allerdings nahezu unmöglich, diese manuell und ohne die Unterstützung geeigneter Software zu betreuen. Gleichzeitig geht der Trend zum Einsatz von immer weniger Mitarbeitern für immer größere Netzbereiche, was zusätzlich die Notwendigkeit geeigneter Lösungen für die Betreuung, Überwachung und das Management von Netzwerken erhöht.

Aus Unternehmenssicht ist ein schnelles, stabiles und sicheres Unternehmensnetz in vielen Fällen essentiell für den Erfolg des Unternehmens. Von einer guten Infrastruktur profitieren nicht nur die Mitarbeiter, sondern auch Unternehmensprozesse. Treten Probleme in diesem Netz auf, ist es die Aufgabe der Netzwerkbetreuer oder Administratoren diese möglichst schnell zu beheben. Fachwissen alleine reicht nicht aus, wenn der Fehler im zugrundeliegenden Netz und einer Flut von Netzwerkdaten nicht gefunden werden kann. Auch finanziell stellt die Wartung von Unternehmensnetzen einen großen Kostenfaktor für die Unternehmen dar, der durch vorausschauende Netzwerk- und Kapazitätsplanung, sinnvoll genutzte Ressourcen, und eine angemessene Dimensionierung des Netzwerks den Anforderungen des Unternehmens angepasst und niedrig gehalten werden kann. Beispielweise muss bei wiederkehrenden Engpässen von Standortanbindungen nicht immer ein teures Upgrade der WAN-Leitungen erfolgen, wenn durch gezielte Provisionierung des Netzwerks und geeigneter QoS-Strategien bestehende Leitungen effizienter genutzt werden können.

Aber auch unabhängig vom unternehmensinternen Nutzen sind Unternehmen gesetzlich dazu verpflichtet, sich gegen Angriffe über das Netz zu schützen und so nicht zuletzt vertrauliche Kundendaten zu schützen. Die Einhaltung von IT-Compliance-Anforderungen schließt dabei die Informationssicherheit, den Datenschutz und die Datenaufbewahrung mit ein, die nicht selten durch Netzwerkangriffe bedroht werden. Zu einem umfassenden IT-Sicherheitskonzept gehört das Erkennen verdächtiger Aktivität im Unternehmensnetz. Bieten Unternehmen netzbasierte Anwendungen oder Dienstleitungen an, sehen sie sich zudem in der Pflicht den Kunden gegenüber die vereinbarten Leistungen zu erbringen. Um Verfügbarkeitsgarantien, Latenzzeiten oder Service-Leistungen zu gewährleisten, muss das Verhalten und die Auslastung des Netzwerkverkehrs beobachtet und analysiert werden. Schwachstellen werden auf diese Weise frühzeitig erkannt und geeignete Maßnahmen können eingeleitet werden. Letztendlich kann bei Nichteinhaltung der Verpflichtungen nicht nur kurzfristig finanzieller Schaden entstehen, sondern auch die Reputation des Unternehmens nachhaltig geschädigt werden.

Damit Unternehmensnetze diesen Anforderungen gerecht werden, müssen Unternehmen wissen, wie das eigene Netzwerk aufgebaut ist, wie es sich verhält und Probleme im Netz müssen erkennbar sein. Netzwerk-Monitoring und Netzwerkmanagement bilden zusammen ein breites Feld mit großem kommerziellem Potenzial, welches sich mit diesem Thema befasst. Entsprechende Software erlaubt es den Netzwerkadministratoren das Netzwerk zu überwachen, zu steuern und zu analysieren. Eine

große Herausforderung sind dabei die enormen anfallenden Datenmengen im Unternehmensnetz. Selbst für kleinere Unternehmen müssen die Daten oft bereits mit Hilfe von verteilten Systemen verwaltet und analysiert werden. Abhängig von den verwendeten Technologien, die zur Überwachung des Unternehmensnetzes eingesetzt werden, müssen Daten zudem aus unterschiedlichen Quellen in ein System überführt und den Anwender/-innen sinnvoll präsentiert werden. Erschwerend kommt hinzu, dass kontinuierlich neue Daten hinzustoßen, die es zu verarbeiten und zu verwalten gilt. Eine weitere Herausforderung ergibt sich aus der Komplexität der Daten. Netzwerkdaten sind oft multivariat, hochdimensional und haben einen zeitlichen Zusammenhang, was die Konzeption geeigneter Analysesysteme zusätzlich erschwert. Viele traditionelle Netzwerk-Monitoring Systeme arbeiten nur selten mit fortgeschrittenen visuellen Mitteln um die Daten zu analysieren, sondern beschränken sich bei der visuellen Analyse oft auf einfache Graphiken oder die tabellarische Form. Um mit diesen Datenmengen zu arbeiten bedarf es allerdings visueller Analysesysteme, die nicht nur in der Lage sind mehrere Dimension der Daten gleichzeitig darzustellen, sondern speziell auf die Fragestellungen und Anwendungsfälle der Domäne angepasst sind.

An diesem Punkt setzt das Forschungsinteresse dieser Masterarbeit an. Die Informationsvisualisierung, im speziellen die visuelle Analytik, befasst sich mit der Frage, wie visuelle Analysesysteme konzipiert werden müssen um domänenspezifischen Anforderungen gerecht zu werden. Die visuelle Analytik findet ihren Einsatz vor allem bei komplexen Problemstellungen, bei denen einfache Visualisierungen oder nicht-visuelle Lösungsansätze an ihre Grenzen stoßen. Nichtsdestotrotz stellt die Datenmenge und die Komplexität der Netzwerkdaten auch hier eine Herausforderung dar. Durch sinnvolle Aggregation der Daten kann die Datenmenge jedoch reduziert werden. Die Forschungsfrage mit der sich diese Arbeit befasst lautet entsprechend: *Wie können Darstellungen von aggregierten Zeitreihen ergänzt werden, damit zeitabhängige multivariate Netzwerkdaten vollständig dargestellt und hinsichtlich Anomalien untersucht werden können.* Anomalien in den Daten sind Anzeichen auf ungewöhnliches oder verändertes Verhalten im Netz, auf das angemessen reagiert werden muss. Welche Arten von Anomalien in Netzwerkdaten für die Analyse in einem Netzwerk-Monitoring Kontext interessant sind und wie diese visuell sichtbar gemacht werden können sind Fragen, die es in diesem Zusammenhang zu beantworten gilt. Welche Eigenschaften der Daten dabei von besonderer Bedeutung sind und in den Fokus gestellt werden müssen, sind entscheidende Punkte die in dieser Arbeit thematisiert werden.

Zur Beantwortung der Forschungsfrage ist im Rahmen dieser Arbeit ein interaktiver, webbasierter Ansatz mit koordinierten Ansichten für die Analyse von Netzwerkdaten konzipiert und in Form eines Softwareprototypen umgesetzt worden. Darin wurden die Erkenntnisse einer Anforderungsanalyse eingearbeitet, bei der domänenspezifische Anforderungen an das Visualisierungskonzept erarbeitet wurden. Das Konzept wurde mit Hilfe von Experteninterviews evaluiert. Die Ergebnisse der Konzeption, der Anforderungsanalyse und der Evaluation dienen schlussendlich alle der Beantwortung der Forschungsfrage. Durch die Kooperation mit der IsarNet Software Solution GmbH standen über den gesamten Bearbeitungszeitraum hinweg Domänenexperten als Ansprechpartner zur Verfügung.

1.1. Gliederung

Die Arbeit ist in folgender Weise gegliedert:

Kapitel 2 – Grundlagen: In den Grundlagen werden relevante Themengebiete beleuchtet, die für das Verständnis nachfolgender Kapitel wichtig sind. Dazu gehört eine Einführung in das Netzwerkmanagement mit Blick auf die verwendeten Netzwerkdaten, ein Überblick über das Forschungsgebiet der Visualisierung, sowie eine kurze Einführung in die Clusteranalyse.

Kapitel 3 – Verwandte Arbeiten: In den verwandten Arbeiten werden wissenschaftliche Arbeiten vorgestellt, die sich sowohl mit der Visualisierung multivariater zeitabhängiger Daten im Allgemeinen als auch mit der Netzwerkverkehrsvisualisierung im Speziellen befassen.

Kapitel 4 – Anforderungsanalyse: Um Herauszufinden welche Erwartungen an Visualisierungen von Netzwerkdaten gestellt werden, wurde zu Beginn der Arbeit eine Anforderungsanalyse mit Hilfe von Experteninterviews durchgeführt. In diesem Kapitel werden die Vorgehensweise und Ergebnisse der Experteninterviews sowie die daraus abgeleiteten Anforderungen vorgestellt.

Kapitel 5 – Konzept: In diesem Kapitel wird der erarbeitete visuelle Ansatz der Arbeit vorgestellt. Im Einzelnen wird dazu das entstandene System mit den koordinierten Ansichten beschrieben, die Designentscheidungen erklärt sowie das Zusammenspiel zwischen den Komponenten thematisiert.

Kapitel 6 – Implementierung: In der Implementierung werden die technischen Details der Umsetzung wie die Architektur des Prototypen, das Zusammenspiel zwischen Frontend, Backend und Datenbank, sowie das verwendete Datenmanagement im Hintergrund beschrieben.

Kapitel 7 – Evaluation: In diesem Kapitel wird das vorgestellte Konzept evaluiert. Dazu werden typische Anwendungsfälle vorgestellt und Feedback von Experten eingeholt um Vor- und Nachteile des visuellen Ansatzes diskutieren zu können.

Kapitel 8 – Diskussion: In einer abschließenden Diskussion werden die Ergebnisse der Arbeit, im Speziellen die der Evaluation, zusammengefasst und diskutiert. Dabei werden Stärken und Schwächen des Ansatzes herausgearbeitet und die Arbeit hinsichtlich Skalierbarkeit untersucht.

Kapitel 9 – Zusammenfassung und Ausblick Das letzte Kapitel enthält eine Zusammenfassung der Arbeit und einen Ausblick auf mögliche Weiterentwicklungen der Konzepte dieser Arbeit.

2. Grundlagen

Die folgenden Abschnitte geben einen Überblick über die wesentlichen Grundlagen dieser Arbeit, die sowohl zum Verständnis der nachfolgenden Kapitel beitragen, als auch die in dieser Arbeit thematisierten Netzwerkdaten und deren Kontext genauer vorstellen. Dazu gehören eine Einführung in das Netzwerkmanagement, ein Überblick über die Prinzipien und Grundlagen der visuellen Analyse und der Informationsvisualisierung, sowie der in dieser Arbeit verwendeten Clusteranalyse.

2.1. Netzwerkmanagement

Das Netzwerkmanagement ist zuständig für die Verwaltung, Steuerung und Überwachung von Netzwerken, mit dem obersten Ziel den ordnungsgemäßen Betrieb des Netzwerkes zu gewährleisten. Netzwerke werden durch den Einsatz von immer mehr Geräten und die zunehmende Vielfalt unterschiedlicher Geräte immer komplexer, was den Einsatz von visuellen und automatisierten Netzwerk Management Systemen erforderlich macht. Zudem befinden sich Netzwerke im stetigen Wandel. Komponenten müssen ausgetauscht oder hinzugefügt werden, sodass sich das Netzwerk im Laufe der Zeit immer mehr aus Komponenten unterschiedlicher Hersteller und Technologien zusammensetzt. Fehlerhafte Komponenten und falsche Konfigurationen müssen dabei erkennbar bleiben [Eck05]. Zusammengefasst werden die wesentlichen Aufgaben des Netzwerkmanagement durch das FCAPS-Modell. Diese lauten wie folgt:

Fehlermanagement (Fault Management) Das Ziel des Fehlermanagements ist die Gewährleistung eines ununterbrochenen Netzbetriebs. Dazu müssen Fehler erkannt, diagnostiziert und behoben werden. Dabei ist man bestrebt Fehler zu vermeiden, indem man Zeit in einen strukturieren Netzwerkaufbau investiert und gezielte Trend-Analysen im Netzwerk durchführt [Eck05].

Konfigurationsmanagement (Configuration Management) Das Konfigurationsmanagement umfasst alle Funktionen, die sich mit der Planung und Änderung von Konfigurationen im Netz beschäftigen. Dabei muss nicht nur der Überblick über das Netzwerk mit seinen Komponenten und Konfigurationen gewahrt werden, sondern das Konfigurationsmanagement ist auch für die Prozesse verantwortlich, die mit Software- oder Hardware-Änderungen einhergehen [Eck05].

Abrechnungsmanagement (Accounting Management) Das Abrechnungsmanagement protokolliert die in Anspruch genommenen Dienstleistungen im Netzwerk. Die übertragenen Datenmengen oder die Nutzungsdauer von Anwendungen sind hier interessant. Die Informationen dienen einer möglichst gerechten Aufteilung der anfallenden Kosten unter den Anwender/-innen [Eck05].

Leistungsmanagement (Performance Management) Das Ziel des Leistungsmanagement ist es, die Leistungsfähigkeit des Netzes zu verbessern. Um die Leistungsfähigkeit des Netzes zu untersuchen, müssen Daten ermittelt werden, die Aufschluss über den Zustand des Netzes geben, auf dessen Basis Veränderungen geplant, durchgeführt und überprüft werden können [Eck05].

Sicherheitsmanagement (Security Management) Beim Sicherheitsmanagement geht es darum, das Netz vor Bedrohungen von Außen und von Innen zu schützen. Die Absicherung des Netzes nach Außen mit Firewalls gehört zu den Aufgaben, ebenso wie das Absichern von Innen durch den konsequenten Einsatz von Zugriffskontrollen, Passwörtern und Verschlüsselungen [Eck05].

Netzwerk-Monitoring Systeme befassen sich im speziellen mit der Überwachung des Netzwerkes und dessen Komponenten und übernehmen damit eine wichtige Aufgabe im Netzwerkmanagement. Der Fokus dieser Arbeit liegt auf Netzwerk-Monitoring Systemen, die sich primär auf die Bereiche Fehlermanagement, Leistungsmanagement und Sicherheitsmanagement beziehen.

2.1.1. Netzwerk-Monitoring

Das Netzwerk-Monitoring befasst sich mit der Überwachung der Komponenten und des Verkehrs im Netzwerk. Es stellt einen Teilbereich des Netzwerkmanagements dar, welcher essentiell zur Erfüllung der oben genannten Aufgaben beiträgt. Das Monitoring lässt sich dabei nach verschiedenen Ansätzen unterteilen. Eine Unterscheidung zwischen externem und internem, aktivem und passivem, als auch zwischen historischem und Real-Time-Monitoring ist möglich. Darüber hinaus gibt es weitere Unterscheidungskriterien, wie Cloud-Monitoring und On-Premises-Monitoring, die hier in dieser Arbeit nicht genauer betrachtet werden [Don17]. Der Lösungsansatz dieser Arbeit lässt sich dem passiven und historischen Monitoring zuordnen.

Externes vs. internes Monitoring Externes Monitoring setzt zusätzliche Komponenten im Netzwerk ein, die zur Überwachung des Netzwerkes dienen. Im Gegensatz dazu wird die Überwachung beim internen Monitoring direkt auf den Komponenten im Netzwerk durchgeführt [Don17].

Passives vs. aktives Monitoring Beim passiven Netzwerk-Monitoring werden die Pakete im Netz mitgehört und darüber Analysen geführt. Beim aktiven Netzwerk-Monitoring werden hingegen aktiv Pakete in das Netzwerk gesendet, die bestimmte Eigenschaften des Netzes testen [Don17].

Historisches vs. Real-Time-Monitoring Zum historischen Monitoring zählen Systeme, die Messwerte des Netzwerks über einen längeren Zeitraum sammeln und für die Anwender/-innen zur Verfügung stellen. Langzeitbeobachtungen der Daten sind ebenso möglich wie das Erkennen von Trends oder das nachträgliche Finden von Netzwerkangriffen. Lang- und mittelfristige Netzwerkplanungen werden dadurch ermöglicht. Beim Real-time-Monitoring werden die Daten in Echtzeit aus dem Netzwerk erhoben, sodass der Administrator sofort auf Auffälligkeiten im Netz reagieren kann. Diese Art des Monitoring findet somit ihren Einsatz bei der Identifikation akuter Störungen oder Probleme im Netzwerk [Don17].

Für die Überwachung, Steuerung und Analyse von Netzwerkelementen oder Netzverkehr existiert eine Vielzahl an unterschiedlichen Netzwerkprotokollen, die zur Datenerfassung des Netzwerkverkehrs verwendet werden können. Zu den bekanntesten gehören SNMP, IPFIX, sFlow und NetFlow. Da es sich bei den für die Entwicklung verwendeten Daten um weiterverarbeitete und stark aggregierte NetFlow handelt, wird im folgenden Abschnitt die Gewinnung und Zusammensetzung dieser Daten genauer betrachtet.

2.1.2. NetFlow

NetFlow ist ein von Cisco Systems, Inc.[1] entwickeltes Netzwerkprotokoll zur Erfassung des Netzwerkverkehrs in Form von IP-Flow-Daten. Die Datenerfassung erfolgt auf NetFlow-enabled Layer-3-Switches und Routern, indem aus den vom Gerät weitergeleiteten Paketen IP-Flows generiert werden, die im NetFlow Cache auf dem Gerät zwischengespeichert werden. IP-Flows sind definiert als eine Menge von Paketen, die in ihren Attributen (Protokoll, Destination IP-Adresse, Source IP-Adresse, Destination Port, u.v.m) übereinstimmen und zeitlich zusammengehören. Neben den IP Paket Attributen werden weitere Informationen wie Subnetzmasken, TCP Flags oder Zeitstempel hinzugefügt. IPFIX ist die Nachfolger-Technologie.

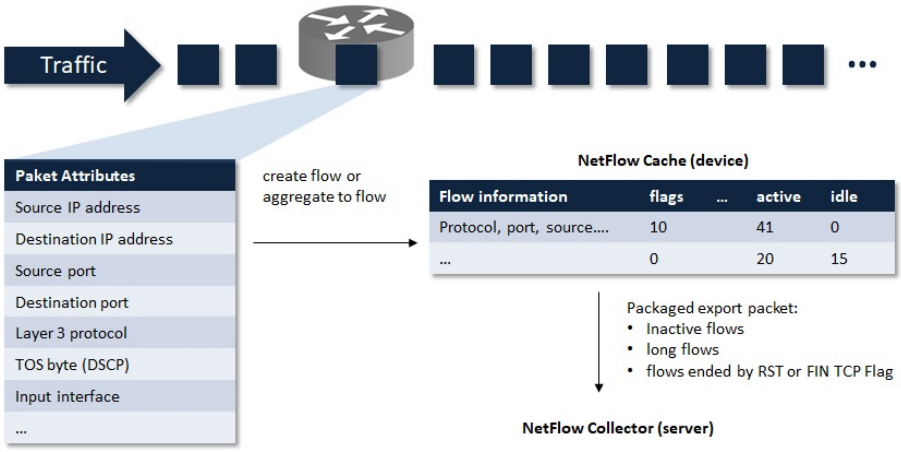


Abbildung 2.1.: Datenerfassung und Export von NetFlow-Daten. Bildquelle: In Anlehnung an [Cis12]

Neben dem direkten Zugriff auf die Daten über die NetFlow CLI können die Daten exportiert und so für die Weiterverarbeitung und Analyse bereitsgestellt werden. Diese Aufgabe wird vom NetFlow Collector übernommen, der die Daten periodisch von den Exportgeräten übermittelt bekommt (Push-Protokoll). Exportiert werden dabei immer eine begrenzte Anzahl an Flows, die bereits abgeschlossen sind und somit keine Veränderungen mehr im Cache erfahren würden. Als abgeschlossen gilt ein Flow-Eintrag, wenn er entweder bereits eine längere Zeit aktiv war (active timer – active Feld), schon ein Weile inaktiv ist (inactive timer – idle Feld) oder ein Paket mit einem RST oder FIN TCP Flag (flags) signalisiert, dass die Netzwerkkommunikation beendet ist. Insgesamt macht der Export der Daten zum NetFlow Collector auf diese Weise nur 1-5% des gesamten Netzverkehrs des Exportgerätes aus [Cis12].

[1] Cisco Systems, Inc. – https://www.cisco.com/

Die erarbeiteten Konzepte dieser Arbeit haben den Anspruch, nicht nur auf eine Datenquelle zugeschnitten, sondern möglichst flexibel den Daten anpassbar zu sein. Sie sollen so auf andere Datenquellen anwendbar sein, dass das Visualisierungskonzept auch auf multivariaten zeitbasierten Daten anderer Datenquellen zur Netzwerk-Analyse verwendet werden kann.

2.1.3. IsarFlow

IsarFlow[2] ist eine Software der IsarNet Software Solution GmbH zum Monitoring von Unternehmensnetzwerken. Basierend auf den im Netz erfassten Netflow/sFlow/IPFIX-Daten erstellt IsarFlow individuelle Analysen, Auswertungen und Reports über den Netzwerkverkehr. Die gesammelten Daten werden zu diesem Zweck aufbereitet und in einer zentralen Datenbank gespeichert. Mit Hilfe verschiedenster Filter und Ansichten können die aufbereiteten Daten hinsichtlich unterschiedlicher Aspekte in einem Webinterfacen dargestellt und untersucht werden [Isa18].

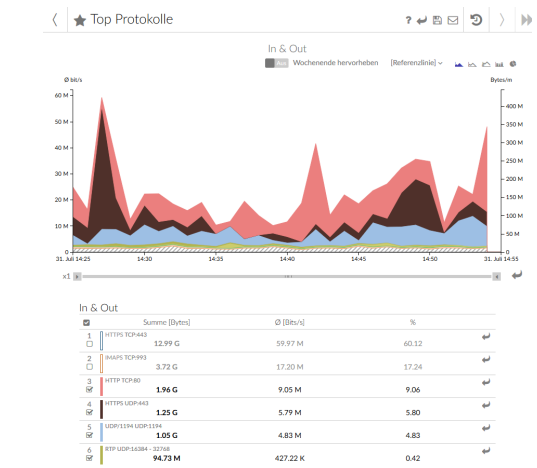
Da diese Arbeit in Kooperation mit IsarNet stattfindet, bildet IsarFlow die Basis für die Erhebung der Anforderungen an das entwickelnde Visualisierungskonzept und fungiert als Vorbild einiger Funktionalitäten. Für diese Arbeit sind dabei vor allem die visuellen Analyseansichten auf die Daten, die IsarFlow den Kunden bietet, von Interesse. IsarFlow verwendet für die visuelle Darstellung der Daten primär drei unterschiedliche Ansichten (siehe Abbildung 2.2). In Abbildung 2.2 (a) ist die Top-Ansicht, im speziellen die Top-Protokolle, abgebildet. In dieser Darstellung wird eine der vielen Dimensionen der Daten im zeitlichen Verlauf dargestellt. Dabei wird nach den Top-Kategorien, in Abbildung 2.2 (a) nach Protokollen, aufgeschlüsselt. Die zweite Darstellung ist die Top-Matrix-Ansicht (siehe Abbildung 2.2 (b)). Diese Ansicht für je Dimensionen, die zusammen eine Sender-Empfänger-Beziehung besitzen. Ein Beispiel dafür kann die Sender- und Empfänger-IP-Adresse sein. In dieser Ansicht werden in einem Ringdiagramm, welches die Anteil der Empfänger und Sender am Gesamtvolumen widerspiegelt, die Beziehungen zweier IP-Adressen durch Kanten visualisiert, deren Dicke die Menge des Verkehrsflusses repräsentiert. Dort ist zu erkennen wer mit wem im Netzwerk kommuniziert. Abschließend existiert eine weitere Darstellung der Daten in Form von geclusterten Zeitreihen, zusehen in Abbildung 2.2 (c). Das *Baselining* ermöglicht das Erkennen von Abweichungen von der Norm im Bezug auf das Gesamtverkehrsvolumen.

2.2. Visualisierung

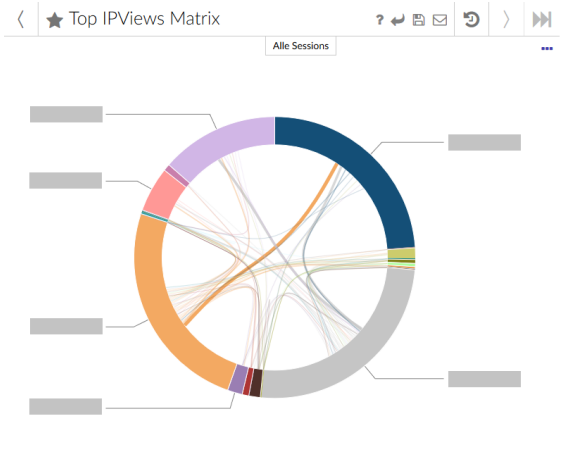
Das Ziel der Visualisierung ist es, Daten durch visuelle Repräsentationen für den Menschen greifbar zu machen und so den Erkenntnisgewinn aus den Daten zu fördern. Geeignete Visualisierungen von Daten helfen diese zu verstehen und Informationen daraus zu gewinnen. Die menschliche Wahrnehmung spielt dabei eine tragende Rolle. Üblicherweise wird die Visualisierung in 3 Teildisziplinen unterteilt:

- Informationsvisualisierung (Information Visualization)
- Wissenschaftliche Visualisierung (Scientific Visualization)
- visuelle Analytik (Visual Analytics)

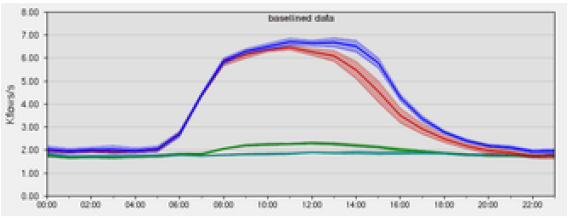
[2] IsarFlow – https://isarflow.de/



(a) Top Protokolle

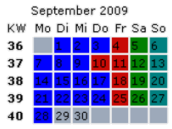


(b) Top IPView Matrix



Nr.	cluster	Flows	occurrences
1	cluster #24	327'502'281.00	15
2	cluster #19	303'418'925.00	5
3	cluster #16	165'377'265.00	4
4	cluster #17	155'312'770.00	4

CSV Export



(c) Baselining (Clusteranalyse)

Abbildung 2.2.: IsarFlow verwendet für die visuelle Darstellung der Daten primär drei unterschiedliche Ansichten. Bildquelle: [Isa18]

Der Unterschied zwischen Informationsvisualisierung und Scientific Visualization liegt nicht, wie der Name vermuten lässt, an einer wissenschaftlichen beziehungsweise nicht wissenschaftlichen Herkunft der Daten. Vielmehr ist die Art der Daten entscheidend. Scientific Visualiziation beschäftigt sich mit Daten, die meist aus Messungen oder Modellen einer realen Welt stammen und damit sowohl einen intrinsisch räumlichen als auch häufig einen zeitlichen Bezug besitzen. Die Aufgabe der Visualisierung ist, die räumlichen Strukturen der Daten durch angemessenes, meist realistisches, Rendering hervorzuheben. Informationsvisualisierung hingegen beschäftigt sich mit der Darstellung meist abstrakter Daten ohne natürliche Repräsentation. Die Definition geeigneter Abbildungen auf visuelle Variablen ist Aufgabe der Informationsvisualisierung. Aus der Informationsvisualisierung entstanden, kann das verhältnismäßig junge Forschungsfeld Visual Analytics als eigenständige dritte Disziplin der Visualisierung angesehen werden. Hier liegt der Fokus auf einer starken Zusammenarbeit visueller und automatischer Analysemethoden, die durch hohe Interaktivität miteinander verbunden werden. Die Ergebnisse dieser Arbeit lassen sich sowohl in die Informationsvisualisierung, als auch in die visuelle Analytik einordnen.

2.2.1. Informationsvisualisierung

Wie eingangs erwähnt, befasst sich die Informationsvisualisierung mit der Visualisierung meist abstrakter Daten ohne natürliche Repräsentation. Den Prozess von den abstrakten Daten hin zu den fertigen Ansichten wird von Card [Car99] in der *Information Visualization Pipeline* beschrieben

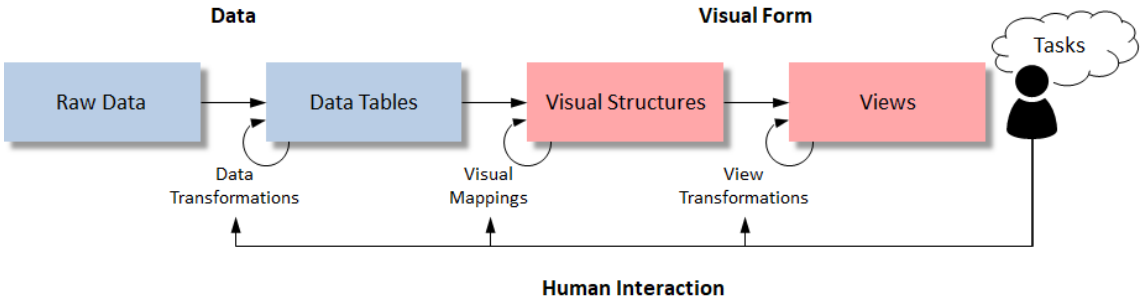


Abbildung 2.3.: Information Visualization Pipeline. Bildquelle: In Anlehnung an Card [Car99].

(siehe Abbildung 2.3). Beginnend mit den rohen Daten werden geeignete Transformationen auf die Daten angewandt, sodass die Daten in eine für die Visualisierung geeignete Form gebracht werden. Anschließend werden Abbildungen der Daten auf visuelle Variablen gesucht, die die gewünschten Eigenschaften dieser herausstellen. Durch *View Transformations*, wie das Skalieren oder das Auswählen eines Bildausschnittes, werden die visuellen Strukturen schließlich auf die benötigte Ansicht zugeschnitten. Die Anwender/-innen entscheiden nach Betrachtung der Ansichten und mit der Fragestellung im Kopf, ob Anpassungen an den Transformationen und Abbildungen vorgenommen werden müssen.

Bei der Konzeption der Visualisierungen gibt es einige allgemein anerkannte Prinzipien, Richtlinien und Regeln, die bei der Auswahl geeigneter Transformationen und Abbildungen unterstützen. Gestaltgesetze oder Richtlinien für Visual Encodings [Mac86] helfen angemessene Abbildungen zu finden, die die psychologische Wahrnehmung der Repräsentation mit der Bedeutung der Daten in Einklang bringen. Zudem sollte bei der Konzeption darauf geachtet werden, Darstellungen zu verwenden, die nicht zu Fehlinterpretationen führen können. Der Lie-Factor kann hierfür als Richtwert verwendet werden [TG83]. Eines der wohl bekanntesten Zitate in diesem Zusammenhang ist das *Information-Seeking-Mantra* von Shneiderman: ”overview first, zoom and filter, then details-on-demand”[Shn96]. Dieses verlangt, den Anwender/-innen zunächst immer einen groben Überblick über die Daten zu geben, sodass auf Basis dessen entschieden werden kann welche Daten im aktuellen Interesse liegen. Die Möglichkeit sich darauf zu fokussieren (zoom and filter) und sich erst dann weitere Details der Daten anzeigen zu lassen, ermöglichen dann eine effiziente und übersichtliche Analyse der Daten.

2.2.2. Visual Analytics

Visual Analytics geht einen Schritt weiter als die Informationsvisualisierung. Durch eine Kombination aus visuellen und automatisierten Methoden werden unterschiedliche Herangehensweisen kombiniert. Hohe Interaktion bindet die menschliche Wahrnehmung in den Prozess der Informationsgewinnung ein und ermöglicht die Steuerung und Kopplung der unterschiedlichen Methoden. Das oberste Ziel ist dabei den Menschen bei dem Verständnis, der Argumentation und der Entscheidungsfindung im Umgang mit großen und komplexen Datenmengen zu unterstützen [KKEM10].

In dem oft zitierten *Visual Analytics Process* nach Keim et al. [KKEM10] wird dieser Prozess der Wissensgenerierung beschrieben (siehe Abbildung 2.4). Durch Transformationen der Daten wie Datenbereinigungen, Normalisierungen oder Projektionen auf Unterräume werden die Daten für

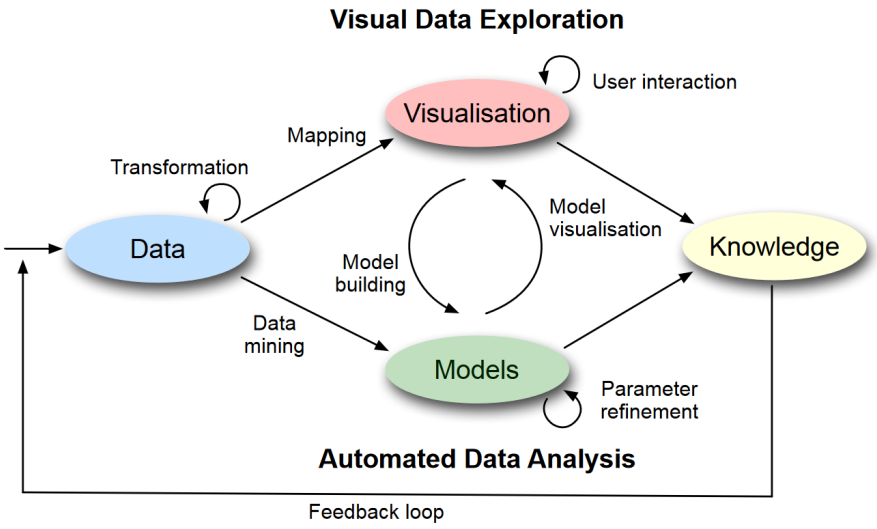


Abbildung 2.4.: Im Visual Analytics Process wird der Prozess der Wissensgenerierung aus Daten durch den Einsatz geeigneter Datentransformation, Visualisierungen und automatischer Analysemethoden beschrieben. Der Anwender wird durch Interaktion mit den Daten in den Prozess integriert und sorgt für eine Koppelung zwischen der visuellen und der automatischen Analyse der Daten [KKEM10]. Bildquelle: [KKEM10]

die visuelle und automatische Analyse vorbereitet. Durch Abbildungen der vorbereiteten Daten auf visuelle Repräsentationen können diese exploriert werden. Gleichzeitig werden statistische Methoden, Machine Learning, Data Mining oder andere Methoden in den Prozess miteinbezogen, um weitere Informationen aus den Daten zu gewinnen. Dabei finden sowohl die Ergebnisse der automatischen Analysen ihren Weg in die Visualisierung, wie umgekehrt die Erkenntnisse aus den Visualsisierung dazu verwendet werden, die Modelle der automatischen Analyse zu verfeinern. Durch Interaktion mit den Visualisierungen von Seiten der Anwender/-innen entsteht eine enge Koppelung zwischen visueller und automatischer Analyse. Am Ende des Prozesses wird aus den Visualisierungen und Modellen der Daten Wissen generiert, welches wieder in den Prozess einfließen kann. Es entsteht ein Feedback-Loop auf dessen Basis Modelle verfeinert und Transformationen oder die visuelle Repräsentation angepasst werden, um schließlich neues Wissen zu generieren [KKEM10].

Anwendung findet Visual Analytics deshalb in vielen Bereichen, in denen die Daten besondere Herausforderungen für die Visualisierung bieten und reine automatisierte Methoden oder Visualisierungen an ihre Grenzen stoßen. Im speziellen große, komplexe, heterogene, multidimensionale oder multivariate Datenmengen können auf diese Weise untersucht werden. Anwendungsgebiete sind somit beispielsweise in vielen wissenschaftlichen Forschungsgebieten wie Astronomie oder Biologie, aber auch im Finanzwesen, Journalismus oder im Notfallmanagement zu finden.

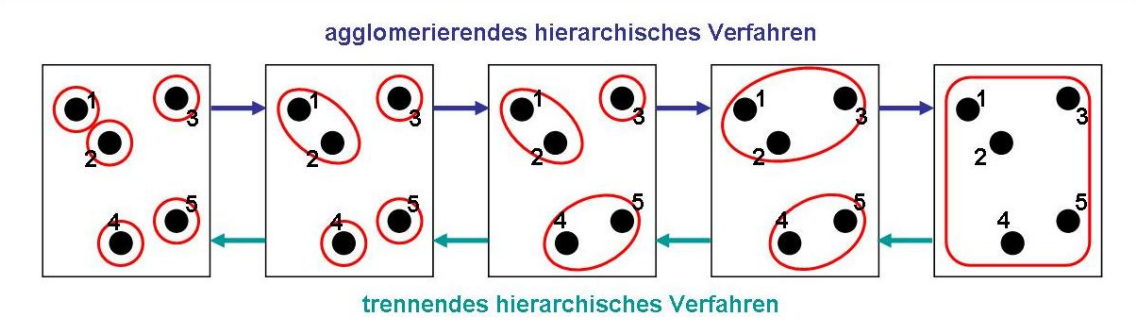


Abbildung 2.5.: Veranschaulichung der Hierarchischen Clusterverfahren. Bildquelle: [Kle09]

2.3. Clusteranalyse

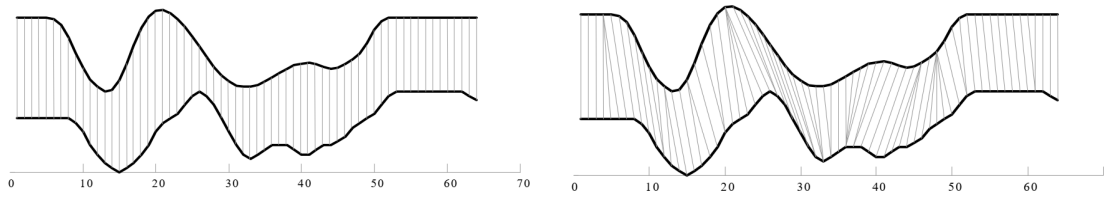
Machine-Learning und *Data-Mining* sind Teilbereiche der künstlichen Intelligenz, zu denen auch viele Methoden zählen, die sich mit der Mustererkennung in Daten befassen. Das *Clustering* ist eine davon. Bei der *Clusteranalyse* wird eine Menge von Objekten oder Daten in Gruppen, den sogenannten *Clustern*, eingeteilt. Dabei wird das Ziel verfolgt, alle ähnlichen Objekte einem Cluster zuzuweisen. Objekte unterschiedlicher Cluster sollen wiederum möglichst unähnlich sein. Im Gegensatz zur Klassifizierung, bei der eine Zuordnung der Objekte in bereits festgelegte Klassen erfolgt, befasst sich die Clusteranalyse mit der Erzeugung neuer Gruppierungen, die aus den Strukturen der Daten abgeleitet werden können. Die zwei wesentlichen Bestandteile der Clusteranalyse sind dabei das verwendete Verfahren und das Abstands- oder Ähnlichkeitsmaß.

2.3.1. Verfahren

Zwei grundlegende Verfahren, die in dieser Arbeit Verwendung finden, sind der k-Means-Algorithmus und die hierarchischen Clusterverfahren:

k-Means Der k-Means-Algorithmus ist eines der bekanntesten Verfahren. Es zählt zu den probabilistischen Verfahren, da die entstehenden Cluster von k oft beliebig gewählten Clusterzentren im Datenraum abhängen. Das Verfahren beginnt mit dem Platzieren der Clusterzentren im Datenraum. Anschließend wird in einem ersten Schritt zunächst allen Datenpunkten durch Bestimmung des nächstgelegenen Clusterzentrums ein Cluster zugeordnet. Daraufhin werden in einem zweiten Schritt die Clusterzentren der Cluster neu berechnet. Die letzten beiden Schritte werden solange wiederholt, bis das Verfahren konvergiert.

Hierarchische Clusterverfahren Hierarchische Clusterverfahren können entweder nach dem Top-Down (trennendes Clusterverfahren) oder Bottom-Up (agglomerierendes Clusterverfahren) Prinzip funktionieren [Kle09]. Beim Bottom-Up Prinzip wird zunächst für jeden Datenpunkt ein Cluster erstellt. Anschließend werden nacheinander immer die beiden ähnlichsten Cluster miteinander vereint bis die gewünschte Anzahl an Clustern erreicht oder eine andere Abbruchbedienung erfüllt ist (siehe Abbildung 2.5). Beim Top-Down-Prinzip beginnt die Clusteranalyse mit einem Cluster, welches im Laufe des Verfahrens immer weiter unterteilt wird.



(a) Euklidische Distanz

(b) Dynamic Time Warping Distance

Abbildung 2.6.: Vergleich der Euklidischen Distanz mit der Dynamic Time Warping Distance. Bildquelle: [SC07]

2.3.2. Distanzmaße

Das verwendete Distanzmaß zur Berechnung der Ähnlichkeit von Daten ist entscheidend für jedes Clusterverfahren. Bei der Wahl des Distanzmaßes sollte darauf geachtet werden, welche Eigenschaften der Daten dazu führen Objekte als ähnlich zu empfinden. Bei räumlichen Daten ist beispielsweise die Distanz der Objekte im Raum entscheidend. Die Euklidische Distanz $d_E(\vec{x}, \vec{y})$, welche auch im Beispiel in Abbildung 2.6 abgebildet ist, ist hier oft die beste Wahl:

$$d_E(\vec{x}, \vec{y}) = \|\vec{x} - \vec{y}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{2.1}$$

mit den Vektoren $\vec{x} = \{x_i\}_{i=1}^n$ und $\vec{y} = \{y_i\}_{i=1}^n$.

Sollen auf diese Weise univariate Zeitreihen miteinander verglichen werden, ist die euklidische Distanz zwar schnell zu berechnen, kann aber nicht immer die erwarteten Ergebnisse liefern. Sind beispielsweise Ausreißer in den Daten vorhanden oder die Zeitreihen leicht zueinander verschoben, können ansonsten ähnliche Zeitreihen schnell als unähnlich eingestuft werden. Eine Verbesserung bieten hier Distanzmaße wie die *Dynamic Time Warping Distance* [Kru83] (siehe Abbildung 2.6 (b)). Allerdings ist die Berechnung sehr aufwendig:

$$d_{DTW}(\vec{x}, \vec{y}) = \min \sum_{l=1}^{k} D(w_l) \quad \text{mit} \quad w_l = (i, j),\ i, j \in [n] \quad \text{und} \quad D(w_l) = D_{i,j} = \sqrt{(x_i - y_i)^2} \tag{2.2}$$

mit den Zeitreihen $\vec{x} = \{x_i\}_{i=1}^n$ und $\vec{y} = \{y_i\}_{i=1}^n$, der $n \times n$-Matrix D und dem *Warping Path* w_l. Dieser stellt eine zusammenhängende Menge von Matrix Elementen dar, die einen Pfad durch die Matrix D und dadurch eine Abbildung von $\vec{x}$ nach $\vec{y}$ definieren. Die Distanz $\mathrm{d}_{DTW}(\vec{x}, \vec{y})$ gibt somit die minimale Distanz zwischen den Zeitreihen $\vec{x}$ und $\vec{y}$ an, die durch eine Abbildung von $\vec{x}$ nach $\vec{y}$ berechnet werden kann. Da dieses Verfahren mit einer Komplexitätsklasse von $O(n^2)$ vor allem für lange Zeitreihen sehr aufwendig werden kann, gibt es einige Erweiterungen, die das Verfahren beschleunigen sollen. Die Definition von Zeitfenstern, die eine maximale zeitliche Abweichung angeben, ist eine Möglichkeit davon [SC07]. Auch eine Normalisierung der Werte kann in diesem Zusammenhang sinnvoll sein, damit Zeitreihen mit ähnlicher Silhouette aber unterschiedlicher Skalierung als ähnlich eingestuft werden.

Ähnlichkeitsmaße für multivariate Daten zu finden stellt eine besondere Herausforderung dar. Dabei stellt sich die Frage, wie die Ähnlichkeiten der unterschiedlichen Variablen miteinander verrechnet werden können. Gewichtungen der Variablen sind ein generischer Lösungsansatz für dieses Problem.

In den meisten Fällen macht es jedoch Sinn sehr spezifische, individuelle und auf das Problem abgestimmte Ähnlichkeitsfunktionen der Objekte zu definieren. Eine Möglichkeit besteht darin nur bezüglich einer Variable zu clustern und das Ergebnis auf die multivariaten Daten zu übertragen. In dieser Arbeit wird diese Variante für das spätere Clustering bevorzugt.

3. Verwandte Arbeiten

Multivariate zeitabhängige Daten stellen bei der Visualisierung in vielerlei Hinsicht eine Herausforderung dar. Speziell auf Netzwerkprobleme zugeschnittene Visualisierungen haben den Vorteil sich nach den Anforderungen der Anwender/innen richten zu können, müssen diese aber gleichzeitig auch erfüllen. In diesem Kapitel werden zunächst Arbeiten vorgestellt, die sich mit Lösungen zu Teilproblemen der multivariaten und zeitabhängigen Datenvisualisierung beschäftigen. Anschließend werden Kombinationen thematisiert und konkrete visuelle Ansätze in den Bereichen Netzwerkvisualisierung, Netzwerkanalyse und Netzwerk-Monitoring vorgestellt. Ein umfassender und allgemeiner Überblick über bestehende Arbeiten zu den einzelnen Themenblöcken ist in dieser Arbeit nicht vorgesehen. Deshalb beschränken sich die kommenden Abschnitte auf Arbeiten, die für die Konzeption der Arbeit von Bedeutung sind.

3.1. Visualisierungen multivariater Daten

Visualisierungstechniken, wie Chernoff Faces [Che73] oder Netzdiagramme, die Variablen durch den Einsatz unterschiedlicher visueller Variablen, wie die Position, Farbe, Länge oder Form, in die Visualisierung einbetten, werden häufig für multivariate Daten verwendet, wenn es darum geht viele unterschiedliche Aspekte der Daten kompakt darzustellen. Für Netzwerkdaten, die häufig kategorialer Natur sind und zudem einen großen Wertebereich besitzen, sind diese Ansätze häufig nicht geeignet, da zum einen zu viele Kategorien existieren und sich diese zudem schlecht oder überhaupt nicht auf visuelle Variablen abbilden lassen. Zudem sind viele dieser Methoden nicht geeignet um große unterschiedliche Datenmengen darzustellen. Ähnliches gilt auch für den Einsatz von Scatterplot-Matrizen oder pixelbasierten Techniken.

Parallele Koordinaten [ID87; Ins85] hingegen werden universell für multivariate Daten eingesetzt. Die Wertebereiche der Dimensionen werden auf Achsen abgebildet, die parallel zueinander angeordnet sind. Die einzelnen Datenelemente werden durch Polygonzüge dargestellt, die ihre Eckpunkte auf den dazugehörigen Werten der Achsen haben. Auf diese Weise lassen sich Daten über mehrere Dimensionen hinweg betrachten und Korrelationen der Variablen sind erkennbar. Für die Analyse der Daten wurden speziell bei den Parallelen Koordinaten zahlreiche Interaktionsmöglichkeiten vorgestellt, die bei der Exploration der Daten unterstützen. Das Umsortieren der Achsen, Brushing der Achsen, Clustern der Datenelemente [FWR99] oder andere Methoden [AOL04; SRJ+17] sollen vor allem bei großen Datenmengen die übersichtlich verbessern. Für kategoriale Daten ergeben sich allerdings wiederum Schwierigkeiten. Kategorien können zwar als Werte auf den Achsen eingezeichnet werden, dadurch kommt es allerdings vermehrt zu Bündelungen der Polygonzüge auf den Achsen, was das Verfolgen einzelner Polygonzüge über mehrere Achsen hinweg deutlich erschwert. Zudem stellt die Festlegung einer Ordnung der Werte auf den Achsen bei nominalen Daten ein Problem dar. Im speziellen werden das Brushing und die Clusterbildungen für kategoriale und nominale Daten schwierig, da durch die fehlende Ordnung der Werte auf den Achsen das

Darstellen von Clustern oder die Definition von Wertebereichen auf den Achsen oft zu sinnlosen Ergebnissen führt. Diese Verfahren sind jedoch essentiell wenn große Datenmengen betrachtet werden sollen. Folglich sind auch diese Techniken für den Einsatz von kategorialen Daten nur bedingt geeignet.

Parallel Sets [BKH05; KBH06; Kos10] gehören zu den wenigen Visualisierungstechniken, die sich speziell für kategoriale Daten eignen. Wie bei den Parallelen Koordinaten basiert dieser Ansatz auch auf parallel angeordneten Achsen. Statt einzelne Datenelemente werden hier jedoch ganze Datenmengen eingezeichnet, die sich an jeder Achse entsprechend der Kategorien der Dimension aufteilen. Die Dicke der Pfade zwischen den Achsen gibt dabei die Kardinalität der dazugehörigen Datenmenge an. Durch die aggregierte Darstellung können viele Datenelemente kompakt durch wenige Datenmengen dargestellt werden. Sie sind somit sowohl für große Datenmengen, als auch für kategoriale Daten geeignet. Parallel Sets erinnern in der Darstellung an die wesentlich bekannteren und älteren Sankey-Diagramme. Der Unterschied zwischen den beiden Visualisierungstechniken besteht im wesentlichen darin, dass Sankey-Diagramme Darstellungen von Mengenflüssen sind, die ihren Ursprung bereits im 19. Jahrhundert haben und heutzutage vor allem für Energie- und Materialflüsse eingesetzt werden [RHF05; TG83]. Kategorien sind dort nicht oder nur selten als Achsen angeordnet. Alluvial Diagramme sehen den Parallel Sets optisch ebenfalls sehr ähnlich. Im Gegensatz zu Parallel Sets werden bei Alluvial Diagramme oft nur die Verbindungen zwischen benachbarten Achsen angezeigt. Korrelationen zwischen mehr als zwei Achsen sind dann nicht mehr erkennbar.

3.2. Zeitreihenvisualisierungen

Linien-, Flächen- oder Balkendiagramme sind die wohl bekanntesten und einfachsten Methoden univariate zeitabhängige Daten darzustellen. Handelt es sich zudem um kategoriale Daten sind *Stacked Area Charts* oder *Stacked Bar Charts* oft die erste Wahl. Darüber hinaus gibt es eine Vielzahl unterschiedlicher visueller Ansätze, die diese grundlegenden Diagrammtypen schlicht erweitern oder komplett neue Wege gehen. Aigner et al. [AMST11] geben hierfür in ihrem Buch *Visualization of time-oriented data* einen umfassenden Überblick über Methoden und Ansätze zur Visualisierung zeitabhängiger Daten. Im Rahmen dieser Arbeit wird nur eine kleine Auswahl unterschiedlicher Ansätze vorgestellt, die bei der Visualisierung unterschiedliche Eigenschaften der Daten in den Fokus stellen und so für unterschiedliche Fragestellungen geeignet sind.

Radiale Layouts eigenen sich beispielsweise besonders gut um periodisches Verhalten in Zeitreihen darzustellen, welches aufgrund der Bürozeiten in Unternehmen auch in Netzwerkdaten vorkommen kann. Spiral Graph [WAM01] ist ein frühes Beispiel für die spiralförmige Visualisierung von Zeitreihen. Dieser Ansatz eignet sich besonders gut für große Datenmengen, da durch die spiralförmige Anordnung der Zeitreihen verhältnismäßig wenig Platz benötigt wird. Ein weiter Vorteil liegt in der guten Visualisierung des periodischen Verhaltens, da dieses in der Spirale immer auf der gleichen Höhe angezeigt wird und so gut sichtbar und vergleichbar ist.

Van Wijk et al. [VV99] präsentieren eine Kombination aus Kalender- und Clusterdarstellung um univariate Zeitreihen zu visualisieren. Um das periodische Verhalten der Zeitreihe darzustellen, wird diese in mehrere Tageszeitreihen unterteilt, geclustert und in Form der Clusterzentren in ein Liniendiagramm eingetragen. Im Diagramm lässt sich das periodische Verhalten der Zeitreihen über

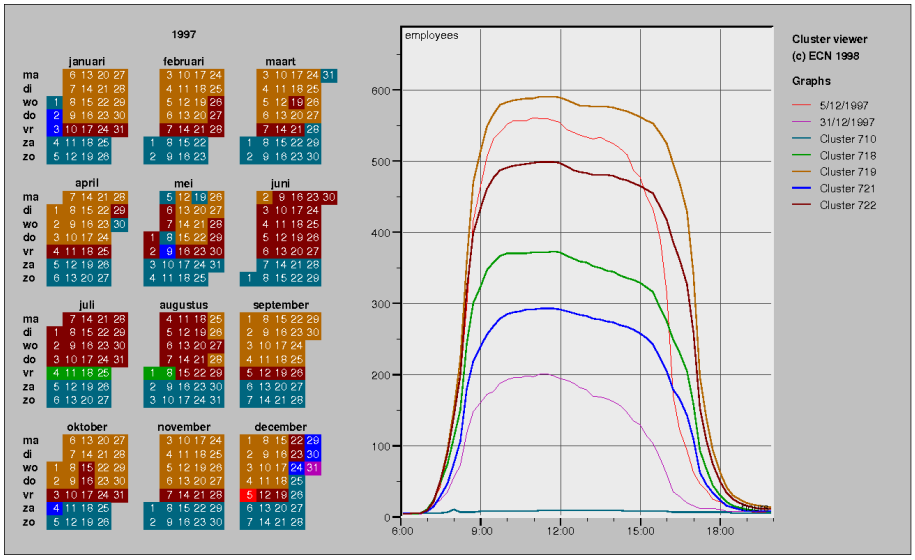


Abbildung 3.1.: Ohne die Eingabe von Feiertagen oder Berücksichtigung der Wochenenden bei der Clusteranalyse werden diese automatisch im Kalender sichtbar [VV99]. Bildquelle: [VV99]

den Tag betrachten, in der Kalenderdarstellung sieht man wiederkehrende Muster an bestimmten Wochentagen. Auch im Produkt IsarFlow werden mit der Funktion *Baselining* Zeitreihen nach Vorbild dieses Ansatzes dargestellt.

Sind die Daten darüber hinaus multivariat, sind viele der bereits vorgestellten Ansätze nicht mehr ohne weiteres anwendbar. Viele wissenschaftliche Arbeiten beschäftigen sich mit der Erweiterung multivariater Darstellungen für zeitabhängige Daten oder umgekehrt mit der Erweiterungen von Zeitreihenvisualisierungen für multivariate Daten. Aber auch komplett neue visuelle Ansätze beschäftigen sich mit der Frage, wie multivariate zeitabhängige Daten dargestellt werden können. Der nächste Abschnitt befasst sich mit diesen Arbeiten.

3.3. Visualisierungen multivariater zeitabhängiger Daten

Multivariate Daten, die einen zeitlichen Bezug besitzen, sind eine besondere Herausforderung für die Visualisierung. Alle Aspekte in einer Darstellung unterzubringen ist dabei eine große Herausforderung. Häufig wird deshalb auf *Multiple Coordinated Views* zurückgegriffen, die mehrere separate Ansichten auf die Daten kombinieren. Eine Zusammenstellung zeitabhängiger und zeitloser Ansichten findet dabei häufig Verwendung. *Small Multiples* werden oft eingesetzt, wenn sich multivariate Zeitreihen sinnvoll in univariate Zeitreihen verlegen lassen. Durch eine geeignete Anordnung vieler Einzeldarstellungen können Korrelationen in den Daten erkannt werden. Ausgehend von multivariaten Ansätzen besteht eine Möglichkeit darin, die Zeit wie jede andere Variable in die Visualisierung zu integrieren. So ist auch die Integration der Zeit in Parallele Koordinaten durch eine zusätzlich Achse möglich. Ein andere Möglichkeit ist durch Animationen multivariater Darstellungen den zeitlichen Bezug in die Darstellung einzubringen.

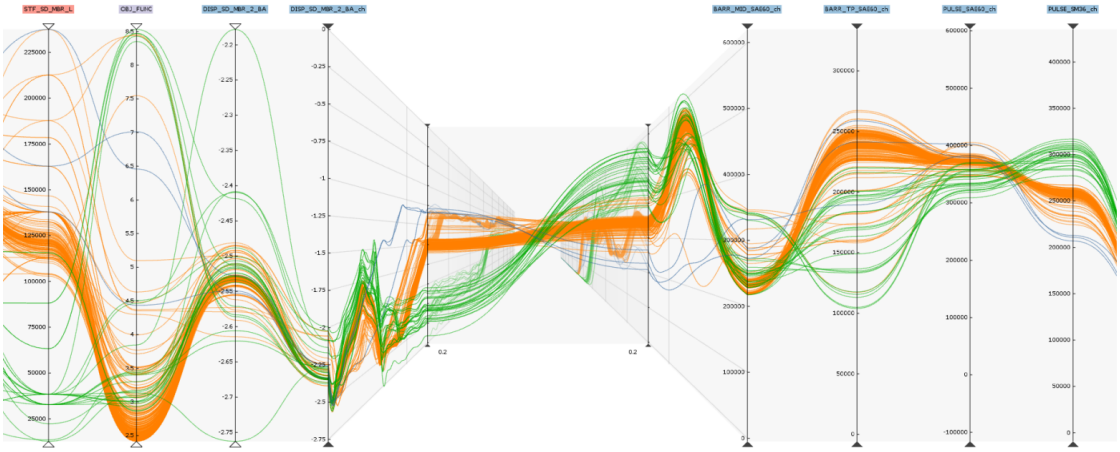


Abbildung 3.2.: Zwei Zeitreihendiagramme werden in einer pseudo-perspektivischen Ansicht zwischen benachbarte Achsen paralleler Koordinaten integriert und geben dort den zeitlichen Verlauf der Datenelement der dazugehörigen Achsen wieder [GRPF16]. Bildquelle: [GRPF16].

Tominski et al. präsentieren eine Reihe von Ansätzen, die die Zeit auf eine besondere Art in multivariate Darstellungen integrieren [TAS04; TAS05]. MultiComb, TimeWheel und Kiviat Tube tragen, wie bei den Parallelen Koordinaten, die Werte auf Achsen auf. Hierbei nimmt allerdings die Zeitachse eine besondere Position im Graphen ein und hebt sich dadurch von den anderen Achsen hervor. Darin enthalten sind auch dreidimensionale Varianten, die alle anderen Achsen fächerartig um eine zentrale Zeitachse aufspannen.

Gruendl et al. [GRPF16] integrieren komplette Zeitreihendiagramme in Parallelen Koordinaten, die die Daten zu einem bestimmten Zeitpunkt anzeigen. Zwei Zeitreihendiagramme werden in einer pseudo-perspektivischen Ansicht zwischen benachbarten Achsen paralleler Koordinaten integriert und geben dort den zeitlichen Verlauf der Datenelemente der dazugehörigen Achsen wieder (siehe Abbildung 3.2). Die Zeit wird durch die Tiefe der Ansicht abgebildet, die durch Zeitschieberegler individuell eingestellt werden kann. Bei diesem Ansatz liegt der Fokus auf den Parallelen Koordinaten. Die Zeit spielt hier nur eine Nebenrolle.

3.4. Netzwerkvisualisierungen

Viele wissenschaftliche Arbeiten beschäftigen sich beim Entwurf neuer Visualisierungskonzepte nicht nur mit der Art der Daten, sondern auch damit, welche Fragestellungen in der jeweiligen Domäne – aus der die Daten stammen – von Interesse sind und somit welche Eigenschaften und Beziehungen in den Daten durch die visuellen Ansätze hervorgehoben werden müssen.

Eine Reihe von Ansätzen versuchen die vielen Aspekte von Netzwerkdaten durch Multiple Coordinated Views darzustellen [CGY+14; CMS+13; FK14; GS09; LYL04]. Chen et al. [CGY+14; CMS+13] bieten mit *OCEANS* und *AnNetTe* einen Ansatz für die interaktive visuelle Untersuchung von Netzwerkdaten im Bereich der Netzwerksicherheit. Hierbei wird eine Kombination aus Timeline, Ring Graph und Paralleler Koordinaten verwendet um unterschiedliche Aspekte

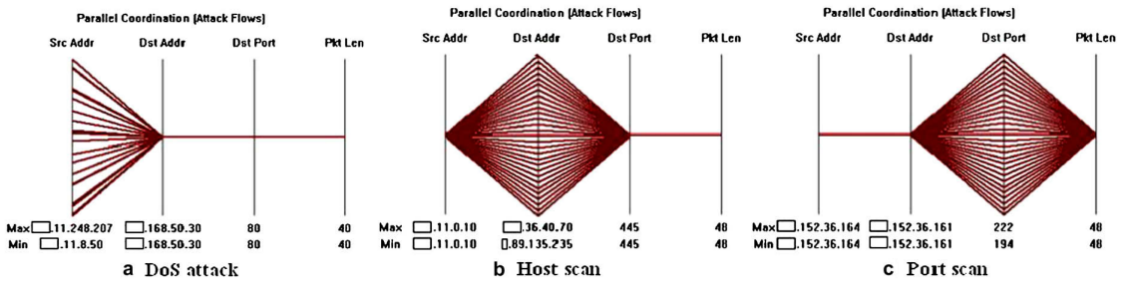


Abbildung 3.3.: Unterschiedliche Attack Signatures, die von Choi et al. identifiziert wurden. (a) zeigt das Muster von DoS-Attacken an. In (b) ist eindeutig zu sehen, dass von einer IP-Adresse ein Host Scan durchgeführt wird. In (c) ist der Port Scan auf die selbe Weise zu erkennen [CLK09]. Bildquelle: [CLK09]

darzustellen. Speziell DDoS-Attacken können damit erkannt werden. VIAssist [GS09] kombiniert ebenfalls unterschiedliche Ansichten für den Einsatz in der Netzwerksicherheit. Hier werden Parallele Koordinaten mit Streudiagrammen kombiniert. NStreamAware [FK14] unterscheidet sich durch einen anderen Fokus von den bereits vorgestellten Arbeiten. Durch Real-Time-Sliding werden die Daten in kleinen Zeitfenstern zusammengefasst, die die wichtigsten Eigenschaften des Netzwerkverkehrs durch kleine Multiple Views anzeigen. Abhängig davon wie interessant die selektierten Features sind, wird die Fenstergröße automatisch angepasst. Hierbei handelt es sich um einen Echtzeitansatz.

Die bereits vorgestellten Parallelen Koordinaten sind nicht nur im Zusammenhang mit Multiple Coordinated Views, sondern auch allein ein häufig genutztes Mittel um multivariate Netzwerkdaten darzustellen. Choi et al. [CLK09] befassen sich beispielsweise mit dem schnellen Erkennen von Netzwerkattacken mit Hilfe Paralleler Koordinaten. Für diese Fälle eignen sich Parallele Koordinaten hervorragend, da sich für bestimmte Netzwerkattacken charakteristische Muster, *Attack Signatures* bezeichnet, bilden. Die Existenz einer Attacke ist somit gut sichtbar, falls diese vorhanden ist. In Abbildung 3.3 ist eine Auswahl der im Paper identifizierten Signaturen zu sehen. Der Nachteil dieses Ansatzes liegt bei der Skalierbarkeit, da bei zunehmender Datenmenge auch die Anzahl der Polygonzüge wächst, die im Diagramm gezeichnet werden müssen. Zudem ist die Menge der einzelnen Kombinationen schwer ersichtlich. Parallele Koordinaten werden speziell für das Erkennen von Mustern im Netzwerkverkehr eingesetzt [BCH+10; CGY+14; CMS+13; GS09; SMW00; YYS05]. Hierfür existieren auch dreidimensionale Ansätze [ECN+15; NCA+13].

Mannsmann et al. [MFKN09; MKN+07; MMK08; MV06] verwenden einen hierarchischen Ansatz in Form einer TreeMap um Netzwerkdaten zu untersuchen. Keim et al. [KMSS06] kombinieren diese Treemap mit einem Sunburst-Diagramm (Radial Traffic Analyzer), welches Source- und Destination-IP-Adressen und Ports auf den einzelnen Ringen anzeigt. Durch Interaktion können Länder aus der TreeMap ausgewählt werden um die Zusammensetzung des Netzwerkverkehrs im Radial Traffic Analyzer genauer zu untersuchen.

Graphbasierte Ansätze versuchen die Empfänger und Sender als Graph darzustellen um Kommunikationspartner im Netzwerk identifizieren zu können. Die Schwierigkeit hierbei ist trotz der vielen Knoten Darstellungen zu konzipieren, die wenig Visual Clutter besitzen. Shi et al. [SLS+13]

reduzieren den Visual Clutter durch Kompression des Graphen. Beim resultierende Graph bleiben dabei die wichtigen Verbindungen und Informationen erhalten, sodass beispielsweise DoS-Attacken oder Botnets erkennbar bleiben.

Einen ganz anderen Ansatz verfolgen Bertini et al. mit *SpiralView* [BHL07]. Hier werden Netzwerk-Alarme spiralförmig aufgetragen um die Entwicklung von Events im Netz beobachten zu können. Periodisches Verhalten ist dadurch besonders gut zu erkennen. Eingebettet in ein System bestehend aus ergänzenden interaktiven Balkendiagrammen und Parallelen Koordinaten können Daten im SpiralView eingegrenzt werden.

4. Anforderungsanalyse

Um Herauszufinden welche Erwartungen an Visualisierungen von Netzwerkdaten gestellt werden, wurde zu Beginn der Arbeit eine Anforderungsanalyse mit Hilfe von Experteninterviews durchgeführt. In den folgenden Abschnitten werden zunächst der Ausgangspunkt der Analyse, sowie die Vorgehensweise und die Ergebnisse der Experteninterviews thematisiert. Die daraus abgeleiteten Anforderungen an das Visualisierungskonzept und den Prototypen bilden den Abschluss des Kapitels und sind das vorrangige Ziel der Anforderungsanalyse.

4.1. Ausgangspunkt der Analyse

Als Ausgangspunkt der Anforderungsanalyse dient IsarFlow, eine Monitoring Software von Unternehmensnetzwerken der IsarNet Software Solutions GmbH, welche bereits in den Grundlagen in Abschnitt 2.1.3 thematisiert wurde. Durch die Kooperation mit IsarNet kann nicht nur auf das Domänenwissen der Entwickler zurückgegriffen werden, sondern auch der Kontakt zu echten Kunden in diesem Bereich und das Arbeiten mit realistischen Datensätzen wird dadurch ermöglicht.

Kunden als Quelle für die Anforderungsermittlung helfen, den Kontext in dem das System eingebettet werden soll zu verstehen und die Funktionalität und Ausrichtung des Konzepts nach den Bedürfnissen, Anwendungsfällen und Anforderungen späterer Anwender/-innen auszurichten. Dank IsarFlow stehen für die Anforderungsermittlung Mitarbeiter und IsarFlow Kunden als Interviewpartner bereit, die perfekt in die Zielgruppe des späteren Visualsierungskonzepts passen. Aus den Anforderungen der Kunden an IsarFlow werden dann im weiteren Verlauf die Anforderungen an das Visualisierungskonzept dieser Arbeit abgeleitet. Sie dienen als Basis für die weitere Entwicklung und Festlegung der Ausrichtung dieser Arbeit.

Auch die für diese Arbeit verwendeten Datensätze stammen von Unternehmenskunden und ermöglichen somit realistische Bedingungen für die Entwicklung, Umsetzung und Evaluation des neuen Visualisierungskonzepts. Neben dem Vorteil, auf bereits bestehende Konzepte der Datenvorverarbeitung zurückgreifen zu können, ergibt sich darüber hinaus ein weiterer Vorteil für die spätere Evaluation, die mit dem selben Personenkreis durchgeführt wird. Da den Nutzern von IsarFlow die prinzipielle Datenstruktur bereits bekannt ist, können diese die Tauglichkeit des neuen Konzepts später besser beurteilen. Einen Überblick über die verwendeten Datensätze, die auch in allen folgenden Abbildungen und Beispielen dieser Arbeit verwendet werden, wird im nächsten Abschnitt gegeben.

4.1.1. Beispieldatensätze

Als Basis für die Entwicklung des neuen Ansatzes stellt IsarNet zwei Datensätze zur Verfügung. Diese beinhalten bereits voraggregierte Netflow-Daten zweier Unternehmenskunden unterschiedlicher Größe und ermöglichen dadurch einen realistischen Rahmen für die Entwicklung. Der erste Datensatz gehört zur einer deutschen Hochschule und ist repräsentativ für einen verhältnismäßig kleinen Unternehmenskunden. Der zweite Datensatz stammt von der Deutschen Bahn und zählt zu den größeren Unternehmenskunden der IsarNet GmbH. Genaue Angaben zur Größe der Datensätze befinden sich im Kapitel Implementierung in Tabelle 6.1.

Wie in der Einleitung und den Grundlagen beschrieben, fallen bereits bei kleinen Unternehmen riesige Mengen an Netzwerkdaten – in diesem Fall NetFlow-Daten – an. Um die Arbeit mit den darin enthaltenen Informationen zu beschleunigen werden diese Daten von IsarFlow vorverarbeitet. Dazu wird zum einen die zeitliche Auflösung der NetFlow-Daten reduziert um unterschiedliche zeitliche Granularitätsstufen vorzuberechnen. Zum anderen fasst eine erneute Aggregation der Daten alle Datenelemente zusammen, die sich nur bei der Anzahl der dazugehörigen Bytes, Flows oder Packets unterscheiden. Damit enthalten die Datensätze dieser Arbeit im einzelnen die folgenden Informationen pro aggregiertem Datenelement:

Granularity Die Granularität der Daten gibt an, welche Intervalllänge für die Aggregation der Daten verwendet wird. In dieser Arbeit wird mit vier Abstufungen der Granularität, die aus IsarFlow direkt übernommen wurden, gearbeitet: 1-Minuten-Granularität, 5-Minuten-Granularität, 1-Stunden-Granularität und 4-Stunden-Granularität.

Timestamp Der Zeitstempel beinhaltet den frühsten Zeitstempel aller in diesem Datenelement aggregierten Rohdaten. Zusammen mit der Granularität kann der zum Datenelement gehörende zeitliche Intervall eindeutig zugeordnet werden.

Input- und Output-IPView IPViews sind von IsarFlow definierte Ansichten auf eine Menge von IP-Subnetzen. Statt die Source- und Destination IP-Adressen der NetFlow-Daten direkt zu verwenden, wird nur die Zugehörigkeit zu den IPViews in den Datenelementen vermerkt. Da eine beliebige Menge an Subnetzen definiert werden kann und diese somit nicht disjunkt sein müssen, kann ein Datenelement mehreren IPViews zugeordnet sein. IPViews setzen sich aus einer ID, einem Namen und einer Beschreibung zusammen.

Input- und Output-InterfaceView Analog zur Definition der IPViews sind InterfaceViews als eine Menge an Interfaces definiert. Statt das Interface direkt zu verwenden, wird wieder nur die Zugehörigkeit zu den InterfaceViews in den Datenelementen vermerkt. Auch hier kann ein Datenelement mehreren IPInterfaces zugeordnet sein. IPInterfaces setzen sich aus einer ID, einem Namen und einer Beschreibung zusammen.

ToS (DSCP, Precedence) Das Type-of-Service Byte wird direkt aus den NetFlow-Daten übernommen. Es gibt zwei mögliche Interpretationen des ToS-Bytes. Zum einen kann nach DSCP und zum anderen nach Precedence aufgeschlüsselt werden. Beide Intepretationsmöglichkeiten sind im Prototyp realisiert.

Bytes Summe der Bytes aller aggregierten Rohdaten.

Flows Summe der Flows aller aggregierten Rohdaten.

Packets Summe der Packets aller aggregierten Rohdaten.

Darüber hinaus können weitere Informationen aus den oben gelisteten Informationen abgeleitet und in die Analyse integriert werden. Dazu zählen beispielsweise die Uhrzeit, die Tagesstunde, das Datum oder zusammengesetzte Dimensionen aus 2 Variablen wie Kombinationen aus In- und Output der Views. Die obige Auflistung entspricht dabei einer logischen Sicht auf die Daten und nicht der tatsächlichen physikalischen Datenstruktur in der Datenbank. Das Datenmanagement im Hintergrund wird in Abschnitt 6.1 separat behandelt. Dort befindet sich auch eine Zusammenfassung der wichtigsten Kenngrößen der Datensätze in Tabelle 6.1.

4.2. Experteninterviews

Durch die Experteninterviews soll das Nutzungsverhalten und die Anforderungen der Nutzer an IsarFlow kennengelernt und vorhandene Probleme bei der Nutzung, sowie Wünsche der Nutzer identifiziert werden. Das Ziel der Experteninterviews ist es, aus den Ergebnissen der Interviews Anforderungen an das Visualisierungskonzept dieser Arbeit abzuleiten. Von den Ergebnissen dieses Kapitels hängen wichtige Designentscheidungen, die im Laufe der Konzeptionierungs- und Entwicklungsphase getroffen werden, ab. Zu diesem Zweck werden die Zielgruppen, das Nutzungsverhalten der Kunden und die Anwendungsfälle von IsarFlow Thema der Interviews sein.

Darüber hinaus dient dieses Experteninterview als Basis der Evaluation. Gegen Ende dieser Arbeit werden in Kapitel 7 erneut Interviews mit dem gleichen Personenkreis geführt, um die Ergebnisse der Konzept- und Entwicklungsphasen vorzustellen und Experten-Feedback zu dem aus dieser Arbeit entstandenen Visualisierungskonzept einzuholen.

4.2.1. Vorgehensweise

Für die Interviews wurden insgesamt drei Person aus dem Umfeld von IsarNet ausgewählt. Die Auswahl der Teilnehmer wurde dabei von IsarNet übernommen. Dabei fiel die Wahl auf einen Mitarbeiter der Deutschen Bahn, der IsarFlow täglich in den Bereichen Überwachung, Administration und Entwicklung des Unternehmensnetzes verwendet, den ProduktOwner von IsarFlow, sowie einen Mitarbeiter im Consultant-Bereich, der in seiner Rolle im stetigen Kontakt mit den Kunden, deren Anliegen und Problemen steht. Entscheidendes Kriterium für die Wahl war der tägliche Kontakt mit der Software als Nutzer oder Kontakt zu den Kunden.

Da die Durchführung der Interviews aus organisatorischen Gründen online über eine Konferenzsoftware realisiert wurde, wurde zu Beginn das Einverständnis auf Verwendung der Tonaufnahme für die Auswertung der Antworten eingeholt. Nach einer kurzen Begrüßungs- und Vorstellungsrunde wurde der Kontext, in dem die Interviews stattfinden, und der Ablauf der Interviews vorgestellt. Insbesondere wurde darauf hingewiesen, dass die Interviews hauptsächlich dazu dienen, die Anforderungen und Anwendungsfälle von IsarFlow kennenzulernen und mehr über die typische Benutzung der Software herauszufinden. Das direkt anschließende Interview gliederte sich thematisch in die folgenden drei Abschnitte:

Allgemeine Fragen zu den Einsatzgebieten: In diesem Abschnitt liegt der Fokus auf den Einsatzgebieten von IsarFlow. Es soll herausgefunden werden, für welche Zwecke IsarFlow in den Unternehmen verwendet wird, welchen Fokus sie dabei haben, sowie welche Funktionen von IsarFlow hauptsächlich dazu verwendet werden.

Detaillierte Fragen zu den Use Cases: Dieser Abschnitt beschäftigt sich mit den konkreten Einsatzszenarien und Use Cases von IsarFlow, die stellvertretend für die typische Verwendung der Software im Unternehmen ist. Hierbei lag der Fokus auch auf den konkreten Wegen, die Benutzer gehen um an das Ziel der Analyse zu gelangen.

Parallel Sets: Da bereits vor der Durchführung der Interviews erste Konzepte vorhanden waren, wurde im letzten Abschnitt bereits Parallel Sets als mögliche Darstellung der Daten vorgestellt. Die Teilnehmer wurden um ihre persönliche Einschätzung gebeten, was die Verwendbarkeit der Darstellung im Zusammenhang mit Netzwerkdaten angeht. Dieser Abschnitt wurde nur bei Mitarbeitern von IsarNet angesprochen.

Die dazu verwendeten Fragebögen befinden sich im Anhang A. Insgesamt wurde eine Stunde pro Interview mit je einem Teilnehmer angesetzt.

4.2.2. Ergebnisse

In den Fragebögen wurde eine Vielzahl an Fragen innerhalb der unterschiedlichen Themenblöcke gestellt. In den folgenden Abschnitten werden die Erkenntnisse aus den gegebenen Antworten in thematischen Blöcken zusammengefasst wiedergegeben. Das Fazit aus den Ergebnissen folgt in Form der abgeleiteten Anforderungen in Abschnitt 4.3.

Einsatzgebiete von IsarFlow

Neben dem Einsatz von IsarFlow in den Bereichen Reporting, Schwellwertüberwachung und Accounting, die für diese Arbeit nicht relevant sind, lassen sich aus den Interviews folgende Haupteinsatzgebiete zusammenfassen. Übereinstimmend wurde *Troubleshooting* als Haupteinsatzgebiet von IsarFlow genannt. Troubleshooting umfasst alle Szenarien, die sich mit der Lösung akuter Netzwerkprobleme beschäftigen. Auslöser ist dabei immer ein externes Ereignis, wie eine Störung im Netzwerk, die manuell durch Nutzer oder automatisiert durch einen Alarm gemeldet wird. Anschließend folgt die Analyse des Netzbereiches im betroffenen Zeitraum. Von besonderem Interesse sind dabei die Auslastungen der Netzwerkverbindungen, sowie welche IP-Adressbereiche, Applikationen oder Protokolle die Störung oder die übermäßige Nutzung verursachen. Mögliche Ursachen können Engpässe, Routingprobleme oder Netzwerkangriffe sein. Laut eines Teilnehmers wird der Anteil des Troubleshooting in der Gesamtnutzung auf 60-80% geschätzt.

Die Kapazitätsplanung und Netzwerkplanung ist das zweite genannte Einsatzgebiet. Hierbei wird tiefgehendes Wissen über das eigene Netzwerk, sowie Fachwissen im Netzwerkbereich vorausgesetzt. Die Aufgabe von IsarFlow in diesem Zusammenhang ist es, auf Basis der Analysen Wissen über das Netzwerk zu erlangen und Schwachstellen kennenzulernen.

Einsatzgebiete in den Bereichen IT-Sicherheit und Netzwerksicherheit, wie das Erkennen von DDoS-Attacken oder Würmern im Netzwerk, liegt nicht im Fokus des Produkts. Allerdings ist auf Entwicklerseite bekannt, dass bereits DDoS-Attacken mit Hilfe der Software erkannt werden

konnten. Die Fähigkeit von IsarFlow Netzwerkattacken erkennen zu können, wird von allen als willkommener Nebeneffekt angesehen und der Bedarf aus Kundensicht besteht. Die Priorität ist allerdings gering.

Anwendergruppen von IsarFlow

Dem Haupteinsatzgebiet entsprechend machen Netzwerkbetreuer, Netzwerkadministratoren und Mitarbeiter des 2nd- und 3rd-Level-Supports einen großen Teil der Nutzer von IsarFlow aus. Diese beschäftigen sich interaktiv mit der Software, arbeiten täglich damit und fallen somit in die Kategorie der *Power-User* der Software.

Netzwerkplaner, als zweite große Anwendergruppe, interessieren sich besonders für die Entwicklung des Netzwerks über die Zeit und nutzen IsarFlow um das Nutzungsverhalten zu beobachten und anhaltende Kapazitätsengpässe zu finden. In regelmäßigen Abständen kann IsarFlow dazu verwendet werden, herauszufinden, welche Anwendungen und Protokolle den meisten Verkehr verursachen und ob Probleme und Engpässe im Netz durch Upgrade der Leitungen, Veränderungen im QoS-Bereich oder Veränderungen im Zeitplan für Softwareverteilungen oder Backups gelöst werden können.

Darüber hinaus wird IsarFlow auch vom Management und Gelegenheitsnutzern (1st Level Support) genutzt. Hier liegt der Fokus allerdings in den Bereichen des Reportings, Accountings und vorgefertigter Analyseansichten, weniger auf der interaktiven Nutzung von IsarFlow. Sie sind somit ebenfalls in dieser Arbeit nicht von Interesse.

Anwendungsfälle und typische Benutzung von IsarFlow

Durch die Einsatzgebiete werden auch die Anwendungsfälle von IsarFlow definiert. Hauptsächlich wurden in den Interviews Szenarien beschrieben, die sich im Bereich Troubleshooting einordnen lassen. Der Ausgangspunkt dieser Szenarien bildet immer ein externes Ereignis in Form einer Störungsmeldung (Netzwerk langsam) von Kunden oder Mitarbeitern oder als Alarm, wie er zum Beispiel durch eine hohe Auslastung einzelner Netzwerkverbindungen durch eine Schwellwertüberwachung ausgelöst werden kann. Dabei können Alarme sowohl intern von IsarFlow als auch extern von anderen Programmen kommen. Ziel ist es stets, die Ursache inklusive der Verursacher der Störung herauszufinden um entsprechende Gegenmaßnahmen einleiten zu können.

Aber auch das freie Explorieren der Daten ohne externe Störungsmeldungen anhand der Gesamtverkehrsvolumen der letzten Tage wurde als gängiges Szenario genannt. Hier sind ungewöhnlicher Netzwerkverkehr – wie beispielweise Peaks oder Dips im Gesamtverkehrsvolumen – der Auslöser der Analyse.

Insgesamt hängt die Benutzung von IsarFlow stark von den Anwendergruppen und dem jeweiligen Anwendungsfall ab. Die Bandbreite erstreckt sich dabei von der kurzen Beobachtung und schnellen Kontrolle des Systems, bei der in möglichst kurzer Zeit ersichtlich werden soll ob es Probleme im Netzwerk gibt, bis hin zur permanenten Überwachung des Netzwerks bei Migrationen im Netz. Wichtige Ansichten, die für eine Vielzahl der genannten Anwendungsfälle für die Nutzer von Interesse sind, sind dabei die Top Protokolle, Top Sessions, Top Interfaces und Top IPViews. Speziell die Top Session Matrix, die bereits erste Ansätze multivariater Analysen zeigt, gefalle laut Aussage eines Teilnehmers den Anwender/-innen gut, weil zum einen sichtbar sei, wer viel Verkehr erzeuge

und gleichzeitig mit wem dabei kommuniziert werde. Gleichzeitig wird allerdings beispielsweise die Top Host Ansicht, die ebenfalls versucht viele Informationen in einer Ansicht zu vereinen, von den Nutzern nur schlecht angenommen, weil die Ansicht als sehr komplex wahrgenommen werde.

Parallel Sets

Die beiden Mitarbeiter von IsarNet als Beteiligte an der Entwicklung wurden am Ende der Interviews bereits zu dem geplanten ersten Konzept der Parallel Sets befragt. Dabei ging es um die prinzipielle Idee Daten ohne Zeit, aber dafür über mehrere Dimensionen hinweg zu betrachten. Die Idee wurde von den Teilnehmern prinzipiell als interessant empfunden, allerdings wurden Bedenken geäußert, die sich auf die visuelle Skalierbarkeit der Darstellung bei großen realen Datenmengen beziehen. Die Hürde für die Anwender/-innen der Software das Prinzip zu erlernen und die Daten richtig zu deuten wird dabei auch angemerkt. Zu viele Linien und Farben könnten schnell unübersichtlich werden und die Darstellung so unbrauchbar machen. Es wird von allen allerdings auch explizit geäußert, dass ein Einsatz als Detailansicht durchaus vorstellbar sei.

Wünsche der Teilnehmer

Im Laufe der Interviews wurden an mehreren Stellen konkrete Wünsche der Teilnehmer zur Funktionalität geäußert. Alle Teilnehmer äußerten Probleme mit dem Finden der richtigen Parameter. Viele Interfaces und keine Topologie-Informationen führen zu Problemen bei der Auswahl der Interfaces. Explizit wurde die Auswahl der Interfaces aus unübersichtlichen und langen Listen von einem Teilnehmer bemängelt. Zudem können bei den Analyseansichten bestimmte Kombinationen und Einstellungen von Parametern zu leeren Mengen als Analyseergebnis führen, was für die Anwender/-innen allerdings erst ersichtlich wird, sobald die Analyse geladen und kein Ergebnis angezeigt werden konnte. Entsprechende Verbesserungen in diesem Bereich, wie eine Vorschau der Auswirkungen der ausgewählten Parameter, wären aus Teilnehmersicht wünschenswert.

Wie im vorherigen Abschnitt bereits anklingt, könnten, laut Einschätzung eines Teilnehmers, Topologie-Informationen über das Netz für das Verständnis bei der Analyse an machen Stellen förderlich sein. Auch die anderen Teilnehmer äußerten sich diesbezüglich zustimmend. Obwohl es interessant wäre diesen Ansatz weiter zu verfolgen, ist das aufgrund der momentan fehlenden Möglichkeiten, Topologie-Informationen aus den bestehenden Daten zu gewinnen, nicht möglich.

Der Wunsch, dass alte Daten erhalten bleiben, wird von den Kunden zwar nicht explizit geäußert, allerdings bestätigen alle Teilnehmer auf Nachfrage, dass eine solche Funktion wünschenswert wäre. Insbesondere wurde von einem Teilnehmer angemerkt, dass die Möglichkeit alte Daten beizubehalten die Wahrscheinlichkeit minimiere, “sich im Laufe der Analyse zu verlaufen´´. Wiederum explizit wurde der Wunsch geäußert, Daten nachladen zu können und so die gleichen Daten aufgeschlüsselt nach anderen Attributen ansehen zu können. Es gäbe viel Bedarf Daten nachzuliefern, gleichzeitig anzuzeigen und komplexere Filterkombinationen einzubauen. Dabei müsse man eine Balance zwischen der Komplexität der Ansicht und dem resultierenden Nutzen finden.

4.3. Abgeleitete Anforderungen an das Konzept

Aus den Ergebnissen der Experteninterviews lassen sich sowohl konzeptionelle als auch technische Anforderungen an das Konzept und den Prototypen ableiten. Die Anforderungen haben nicht den Anspruch bereits genaue Beschreibungen der später realisierten Funktionalitäten zu sein, sondern dienen vielmehr als Grundlage für den Entwurf, die Ausrichtung und den Fokus des Visualisierungskonzepts und prägen fundamentale Designentscheidungen im späteren Entwicklungsprozess. Neben den Interviews haben dabei auch die Erkenntnisse aus den verwandten Arbeiten und den Grundlagen, wie die grundlegenden Prinzipien der visuellen Analyse, entscheidenden Einfluss auf die in den folgenden Abschnitten auf informelle Weise beschriebenen Anforderungen.

4.3.1. Konzeptionelle Anforderungen

Als Zielgruppe des Konzepts dieser Arbeit werden Netzwerkbetreuer, Netzwerkadministratoren und Mitarbeit des 2nd- und 3rd-Level-Supports festgelegt. Entsprechend kann bei der Konzeption ein gewisses Vorwissen im Netzwerkbereich und Wissen über das eigene Unternehmensnetz vorausgesetzt werden. Eine Verallgemeinerung des Konzepts auf alle Anwendergruppen von IsarFlow, inklusive Standortverantwortlicher und Manager, führt zu unnötigen Abstrichen in der Komplexität der Anwendung. Das Konzept soll jedoch umfangreiche Möglichkeiten der Analyse bieten und muss deshalb auf die Bedürfnisse von Experten zugeschnitten werden.

Das Erkennen von Anomalien in den Daten soll im Mittelpunkt des Konzepts stehen. Dies entspricht dem Anwendungsfall des freien Explorierens der Daten und bezieht sich sowohl auf ungewöhnliche *Peaks* und *Dips* im Gesamtverkehrsvolumen als auch auf das visuelle Erkennen von Clusterbildungen im mehrdimensionalen Datenraum. Wurde ein Peak, Dip oder eine andere Anomalie in den Daten entdeckt, sollen die Anwender/-innen mithilfe des Systems in der Lage sein zu erkennen, welche Ursachen dahinter stehen. Für die Anwender/-innen ist dabei von besonderem Interesse wer (IP-Adressbereich, IPView) und was (Protokoll, Anwendung) diesen Verkehr verursacht hat und ob es andere Tage gibt, die die gleichen Auffälligkeiten zeigen. Eine Unterscheidung zwischen allgemein erhöhtem Verkehrsfluss und einem bestimmten Verursacher soll deutlich erkennbar sein. Aber auch der *Troubleshooting*-Anwendungsfall soll durch den Ansatz adressiert werden.

Bei den Interviews hat sich herausgestellt, dass der zeitliche Bezug der Daten von großer Bedeutung für die Analyse ist. Die ursprüngliche Idee mit einer Darstellung der Parallel Sets zu beginnen und den zeitlichen Bezug in diese Darstellung zu integrieren, wird der Bedeutung der Zeit für die Analyse nicht gerecht. Deshalb soll der zeitliche Bezug der Daten in den Fokus gestellt werden und die Analyse mit einem Überblick auf die Daten im zeitlichen Verlauf beginnen. Auch die Auslastung von Leitungen und Interfaces im zeitlichen Verlauf sind interessante Punkte, die es zu adressieren gilt. Die Parallel Sets Ansicht soll im Konzept als Detailansicht integriert werden um Verbindungen und Zusammenhänge der Daten über mehrere Dimensionen hinweg darzustellen. Als wichtige Dimensionen wurden in den Interviews die Protokolle, Sessions, Interfaces und IPViews erkannt. Zusätzlich ist für die Troubleshooting-Szenarien eine Type-of-Service Dimension interessant. Da in den Interviews Bedenken zur Skalierbarkeit der Parallel Sets geäußert wurden, muss das Konzept geeignete Gegenmaßnahmen beinhalten, um die Übersichtlichkeit der Darstellung zu verbessern.

4.3.2. Technische Anforderungen

Die technischen Anforderungen an das System beziehen sich hauptsächlich auf die konkrete Umsetzung des Konzepts als Softwareprototyp. Von Seiten der Universität Stuttgart und der IsarNet GmbH existiert hier die Vorgabe, den Prototypen als webbasierten Ansatz bestehend aus Frontend, Backend und einem angemessenen Datenmanagementsystem umzusetzen. Als Ergebnis aus den Experteninterviews, aber auch aus den Prinzipien der visuellen Analyse, lassen sich folgende konkrete technische Anforderungen an den Prototypen ableiten.

Mit einem Blick auf die bereitgestellten Datensätze und die üblichen im Netzwerkverkehr anfallenden Datenmengen wird klar, dass für die Analyse auf große Datenmengen im Hintergrund zurückgegriffen werden muss. Wichtig ist deshalb, dass der Prototyp auch für große Datenmengen geeignet ist und diese in einer angemessenen Zeit für den Nutzer verfügbar machen kann. Als angemessene Antwortzeit auf die Datenanfragen wird deshalb ein grober Richtwert im einstelligen Minutenbereich für die hier verwendeten Datensätze und die geplante Architektur angestrebt.

Für interaktive Visualisierungen, zu denen das Konzept dieser Arbeit zählt, ist es zudem essentiell, dass die Anwendung abgesehen von den Datenanfragen interaktiv vom Nutzer bedient werden kann. Reaktionen des Systems auf direkte Interaktion und Manipulationen an den visuellen Primitiven sollen wenn möglich im Millisekundenbereich liegen.

5. Konzept

Aus den Anforderungen geht hervor, dass die Zeit bei der Analyse von Netzwerkdaten von großer Bedeutung ist. Eine als unauffällig eingestufte Momentaufnahme des Netzwerkverkehrs zu einer Uhrzeit kann zu einer anderen Uhrzeit bereits als höchst ungewöhnlich eingestuft werden und Anlass für eine tiefgehende Analyse des Netzwerkverkehrs sein. Ungewöhnlich viel oder wenig, sowie eine andere Zusammensetzung des Netzwerkverkehrs zu einer bestimmten Tageszeit sind dabei wichtige Merkmale für die Klassifizierung der Daten. Die Netzwerkdaten komplett losgelöst vom zeitlichen Bezug zu betrachten führt hingegen nur in wenigen und sehr zielgerichteten Szenarien zu aussagekräftigen Analyseergebnissen. Multivariate Visualisierungstechniken, die die zeitliche Dimension gleichberechtigt in die Darstellung integrieren, können das Erkennen zeitlicher Muster erschweren, wenn der Sonderstatus der Zeit nicht ausreichend hervorgehoben wird.

Im Visualisierungskonzept dieser Arbeit wird aus diesen Gründen der zeitliche Bezug der Daten in den Fokus gestellt. Dies wird erreicht indem sowohl in der Übersicht der Daten als auch in den Detailansichten die Zeit eine tragende Rolle spielt. Die Parallel Sets Darstellung, die bereits in den Interviews als mögliche Darstellungsform multivariater Daten vorgestellt wurde, nimmt dabei die Rolle des Bindeglieds zwischen der eher groben, aber vollständigen zeitlichen Übersicht über die Daten und den fein aufgelösten Detailansichten der Daten ein. Um eine klare Strukturierung der Analyse zu erreichen, werden verschiedene visuelle Ansichten der Daten kombiniert und in eine eindeutige Reihenfolge für die Analyse gebracht. Dieser geplante Arbeitsablauf wird im folgenden als *Workflow* bezeichnet.

5.1. Workflow und Datenfluss

Das Design der Anwendung orientiert sich an den Prinzipien der Multiple Coordinated Views, wie sie von Baldonado et al. beschrieben werden [WWK00]. Der entscheidende Unterschied liegt bei der Anordnung der Komponenten auf drei hierarchisch organisierten Ebenen, zu sehen in Abbildung 5.1. Mit dieser Anordnung werden zwei Ziele verfolgt: Zum einen wird eine Reihenfolge der Komponenten vorgegeben, in der sie in der Analyse betrachtet werden sollen und zum anderen wird dadurch ein Datenfluss zwischen den Ebenen modelliert. Beginnend bei den globalen Einstellungen wird die gesamte Datenmenge zunächst grob nach dem Zeitraum gefiltert. Die resultierende Menge wird dann in Ebene 1 dargestellt. Hier kann wiederum ein zeitlicher Filter, der in den folgenden Abschnitten genauer erklärt wird, auf die Daten angewandt werden, sodass die Datenmenge für die darunter liegenden Ebenen weiter reduziert wird. Auch die Parallel Sets Ansicht kann als Filter auf die Daten fungieren. Hierbei können Einschränkungen der Dimensionen und Kategorien vorgenommen werden. Auf diese Weise propagieren sich die Filtereinstellungen einer Ebenen auf alle darunterliegenden Ebenen. Dadurch wird erreicht, dass beispielsweise Änderungen in der dritten Ebene keine Auswirkungen auf die darüber liegenden Ebenen haben, die im Workflow

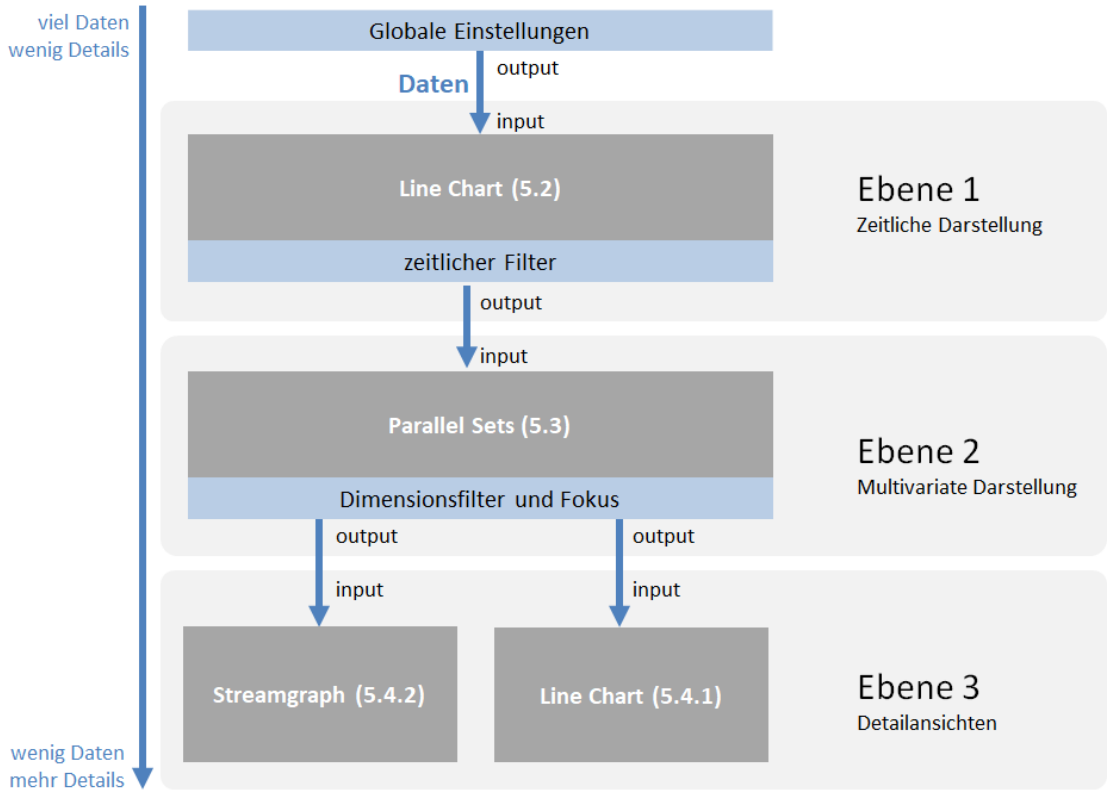


Abbildung 5.1.: Übersicht über die 3 Ebenen des Workflows. Die Filter sind blau, die Ebenen hellgrau und die Komponenten dunkelgrau gekennzeichnet. Der Datenfluss von oben nach unten wird durch die blauen Pfeile symbolisiert.

bereits betrachtet wurden. So kann bei der Analyse trotz der vielen unterschiedlichen Ansichten auf die Daten der Überblick bewahrt werden und mögliche Interaktionen zwischen den Ebenen sind klar strukturiert und vorhersehbar.

Die in Abbildung 5.1 definierten Ebenen spiegeln sich direkt in der Positionierung und Strukturierung der Komponenten in der Benutzeroberfläche wieder (siehe Abbildung 5.2). Die eingezeichneten Pfeile und Filterinformationen separieren die Ebenen optisch voneinander und verdeutlichen zudem den Datenfluss zwischen ihnen. Verstärkt wird dieser Effekt durch die Filterinformationen, die die Filter der darüber liegenden Ebene für den Nutzer noch einmal textuell zusammenfassen. Insgesamt wurde beim Design der Benutzeroberfläche darauf geachtet, möglichst klare Abgrenzungen zwischen den Komponenten und den Ebenen zu schaffen um so – trotz der vielen einzelnen Komponenten – eine klare und übersichtliche Struktur der Anwendung zu erreichen. Die klare Vorgabe des Workflows und die Beschränkung jeder Komponente auf eine Aufgabe sollen den Nutzer zudem unterstützen, die Analyse in einzelne Teilaufgaben herunterzubrechen und so die wahrgenommene Komplexität der Anwendungen verringern. Außerdem erleichtert die modulare Struktur, einzelne Komponenten durch andere zu ersetzen, falls es die Aufgabenstellung erfordert. In den nachfolgenden Abschnitten werden die einzelnen Komponenten nacheinander im Detail thematisiert und deren Funktionalität und Rolle im Workflow genauer vorgestellt.

Abbildung 5.2.: Übersicht über alle Komponenten (1) Ebene 1 – Clustering und Zeitfilter (2) Ebene 2 – Parallel Sets und Dimensionsfilter (3) Ebene 3 – Streamgraph (4) Ebene 3 – Filterzusammenfassung (5) Filterpfeile und Filterinformation

5.2. Ebene 1 – Clustering

In der ersten Ebene der Anwendung, und somit am Anfang der Analyse, befindet sich eine zeitliche Visualisierung der Netzwerkdaten in Form eines Zeitreihendiagramms. Dabei wird zunächst nur das Gesamtvolumen über die Zeit betrachtet und alle anderen Variablen außen vor gelassen. Diese Entscheidung beruht auf den Ergebnissen der Anforderungsanalyse, da das Erkennen auffälliger Kurvenverläufe im Gesamtvolumen der Daten ein gängiger Start für die Analyse im Anwendungsfall der freien Exploration ist. Anomalien in den Daten schlagen sich häufig im Verkehrsvolumen nieder, da diese zusätzlich zum normalen Verkehr auftauchen oder diesen stören. Peaks oder Dips im

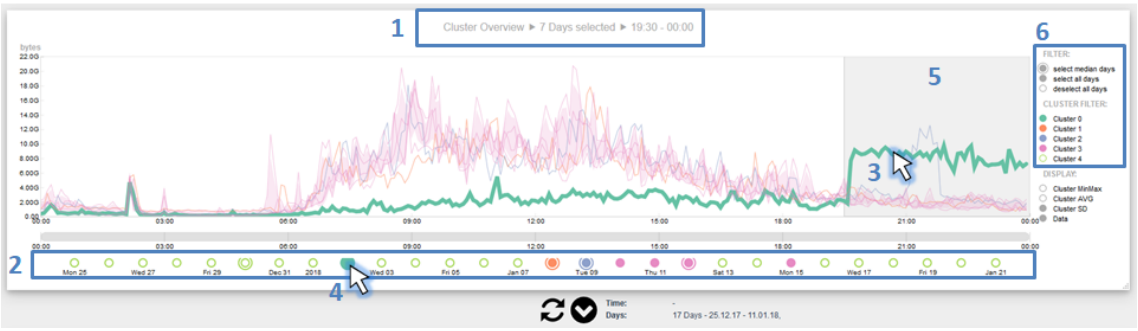
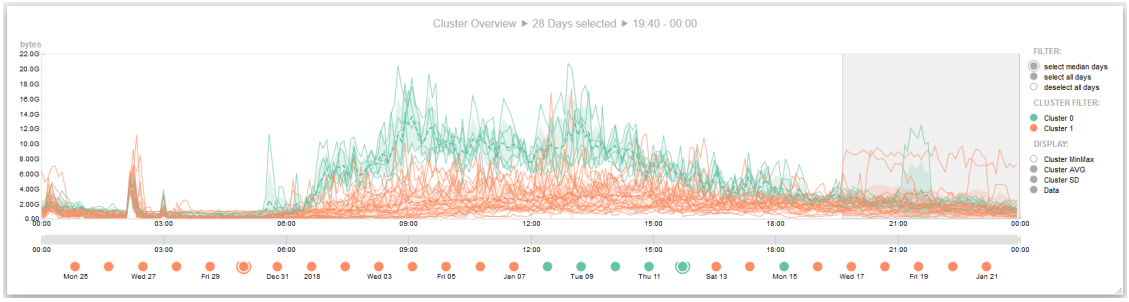


Abbildung 5.3.: Übersicht über alle Möglichkeiten des Zeitfilters. (1) Die Überschrift der Ansicht beinhaltet einen Überblick über die aktuell eingestellten Filter im Stil einer Breadcrumb-Navigation. (2) Tageszeitleiste. (3) Tage können direkt in der visuellen Ansicht der Zeitreihen ausgewählt werden. (4) Tage können auf Basis des Datums ausgewählt werden. (5) Tageszeiträume können durch Selektion des jeweiligen Bereichs ausgewählt werden. (6) Einzelne Cluster oder repräsentative Tage können ausgewählt werden.

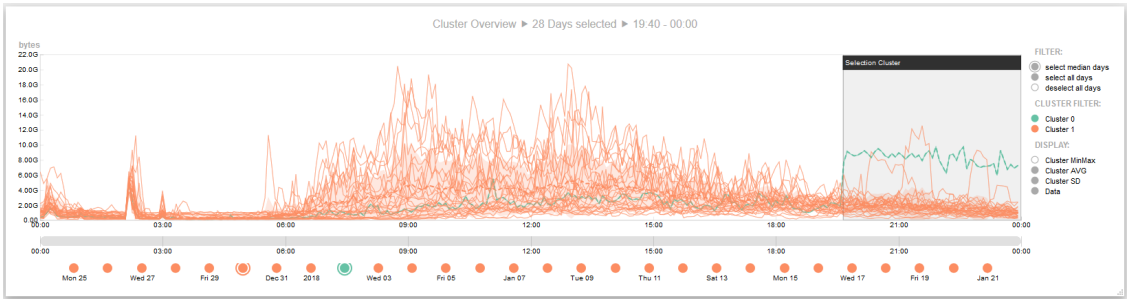
Kurvenverlauf sind die Folge. Das Clustern der Zeitreihen hilft, diese Anomalien zu erkennen, indem der Tagesverlauf einzelner Tage in einem größeren Zusammenhang dargestellt wird und Abweichungen von der Norm erkennbar sind.

Wie auch in IsarFlow werden die Zeitreihen nach Vorbild von Van Wijk et al. [VV99] geclustert. Dazu muss zunächst aus den multivariaten Daten das Gesamtvolumen gemessen an der Anzahl der *Flows*, *Packets* oder *Bytes* pro Zeitstempel durch Aggregation der Datenelemente berechnet werden. Die resultierende Zeitreihe wird in N Zeitreihen unterteilt, die jeweils den Zeitraum eines Tages abdecken. Die Clusteranalyse wird anschließend darauf ausgeführt. Dargestellt werden die geclusterten Zeitreihen in einem Tagesintervall von 0:00 Uhr - 24:00 Uhr in Form des Mittelwerts, der Standardabweichung, der minimalen und maximalen Ausdehnung sowie der Darstellung der Zeitreihen selbst als Linien im Diagramm. Nach Bedarf kann jede dieser Ansichten ein- oder ausgeblendet werden um sich auf unterschiedliche Aspekte der Daten konzentrieren zu können. Mit der Anzeige der Cluster durch Mittelwert und Standardabweichung können die Gemeinsamkeiten und Unterschiede der Tage dargestellt werden. Die Anzeige der Zeitreihen kann beispielsweise dazu verwendet werden um Ausreißer in den Daten zu erkennen und diese direkt in der Darstellung zu selektieren. Anstelle der Kalenderdarstellung von Van Wijk wird in dieser Arbeit ein linearer Zeitstrahl verwendet um die Tage im zeitlichen Verlauf darzustellen. Die Vorteile dieser Darstellung liegen in der kompakten Darstellung, die Nachteile ergeben sich durch die nicht periodische Darstellung der Wochen.

Die erste Ebene kann sowohl als eigenständige Visualisierung der Daten, als auch als reiner zeitlicher Filter, im Folgenden kurz *Zeitfilter* genannt, auf die Daten verstanden werden. Die Darstellung der geclusterten Zeitreihen dient hierbei als Hilfestellung für die erste Auswahl interessanter Zeiträume. Der Zeitfilter, der in dieser Ebene realisiert wird, bietet die Möglichkeit einzelne Tage, Cluster oder einen Tageszeitraum für die nächsten Ebenen auszuwählen. Auch die Auswahl eines repräsentativen Tages kann in Form des Medians des Clusters einfach ausgewählt werden. Die Auswahl kann dabei direkt auf den visuellen Primitiven der Darstellung durchgeführt werden. Abbildung Abbildung 5.3 gibt einen Überblick über die Interaktionsmöglichkeiten zur Filterung der Zeit.



(a)



(b)

Abbildung 5.4.: Beispiel für das adaptive Clustering. (a) Im grau markiertem Bereich passen die gefundenen Cluster nicht zu den dort zu sehenden Kurvenverläufen. (b) Eine Einschränkung des Verfahrens auf diesen Tageszeitraum führt zu einem besseren lokalen Ergebnis.

5.2.1. Adaptives Clustering

Selbst die besten Clusterverfahren bringen nicht immer die erwarteten Ergebnisse. Ausreißer stören die Verfahren, sind aber in manchen Fällen für die Analyse nicht interessant. Speziell beim Clustering von Zeitreihen kommt hinzu, dass das globale Optimum der Clusteranalyse nicht zwangsläufig auch gut lokal zu den Zeitreihen passt (siehe Abbildung 5.4 (a)). Andere, besser auf das Clustern von Zeitreihen abgestimmte Clustermethoden und Distanzmaße können zwar zu besseren Ergebnissen führen, sind allerdings häufig komplexer in der Berechnung, benötigen deutlich mehr Zeit und lösen nicht das Problem der lokalen Optimierung. Ist die Wahl bereits auf eine Teilmenge der Tage und einen interessanten Tageszeitraum gefallen, möchte man oft, dass die Clusteranalyse für diese Selektion die besten Ergebnisse liefert. Das Miteinbeziehen aller Daten ist für die Berechnung daher hinderlich. Aus diesen Gründen wurde eine Erweiterung integriert, welche es ermöglicht die zu berücksichtigenden Tage und den Tageszeitraum für das Clusterverfahren anzupassen. Statt das globale Optimum zu finden, kann das Clusterverfahren auf einen bestimmten Zeitraum beschränkt werden, sodass Cluster berechnet werden, die lokal gute Ergebnisse liefern (siehe Abbildung 5.4 (b)). Ein besonderer Vorteil liegt hierbei in der iterativen Anwendung der Clustermethode mit unterschiedlichen Tageszeiträumen. Das Verfahren kann beliebig oft mit unterschiedlichen Einstellungen wiederholt werden, bis das Ergebnis der Clusteranalyse den Erwartungen entspricht. Bei der Überprüfung der Clusterqualität wird die menschliche Wahrnehmung durch die Darstellung der Ergebnisse in der ersten Ebene unterstützt. Die Vorgehensweise beim adaptiven Clustering wird in Abbildung 5.5 erklärt. Im Gegensatz zu dem

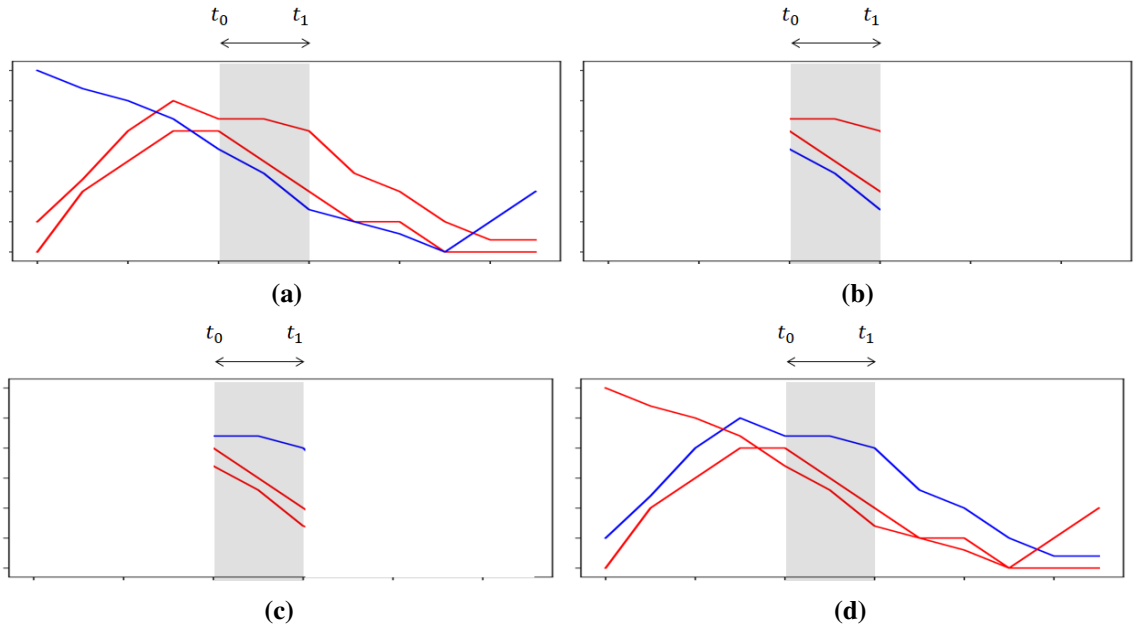


Abbildung 5.5.: Vorgehensweise beim adaptiven Clustering. (a) → (b) In einem ersten Schritt werden alle deselektierten Tage aus dem Verfahren ausgeschlossen und die Zeitreihen auf den eingestellten Tageszeitraum begrenzt. (c) Anschließend wird auf diesen Zeitreihen das Clusterverfahren angewandt. (d) In einem letzten Schritt werden die Ergebnisse der begrenzten Zeitreihen auf die Tageszeitreihen übertragen.

in der Literatur verwendeten Begriff des *adaptiven Clusterings* bezieht sich der hier vorgestellte Ansatz lediglich auf manuelle Anpassungen der Datenmenge und des Tageszeitraums auf Basis der Einschätzung der Anwender/-innen. Automatische Verfahren der adaptiven Clusteranalyse finden hier keine Anwendung.

Erweiterungen des Konzepts sind denkbar, so dass statt eines harten Box-Filters bei der Tageszeiteinschränkung auch andere, weichere Gewichtungsfunktionen denkbar wären, die beispielsweise auch direkt vom Nutzer definierbar sein könnten. Um die Interaktion zu verbessern, besteht zudem die Überlegung, die Auswirkungen einer Selektion auf das Clustering sofort sichtbar zu machen. Allerdings ist es nicht immer wünschenswert, dass sich das Clustering automatisch anpasst, wenn einzelne Tage deselektiert oder Tageszeiteinschränkungen vorgenommen werden. Eine Unterscheidung zwischen Selektion für den Zeitfilter und Selektion für das Clustering könnte eine mögliche Lösung sein, bringt allerdings Schwierigkeiten in der Darstellung und im Verständnis mit sich. Eine spätere Integration der Funktionalitäten, welche diese Problematik löst, könnte Thema der weiteren Arbeit am Konzept sein.

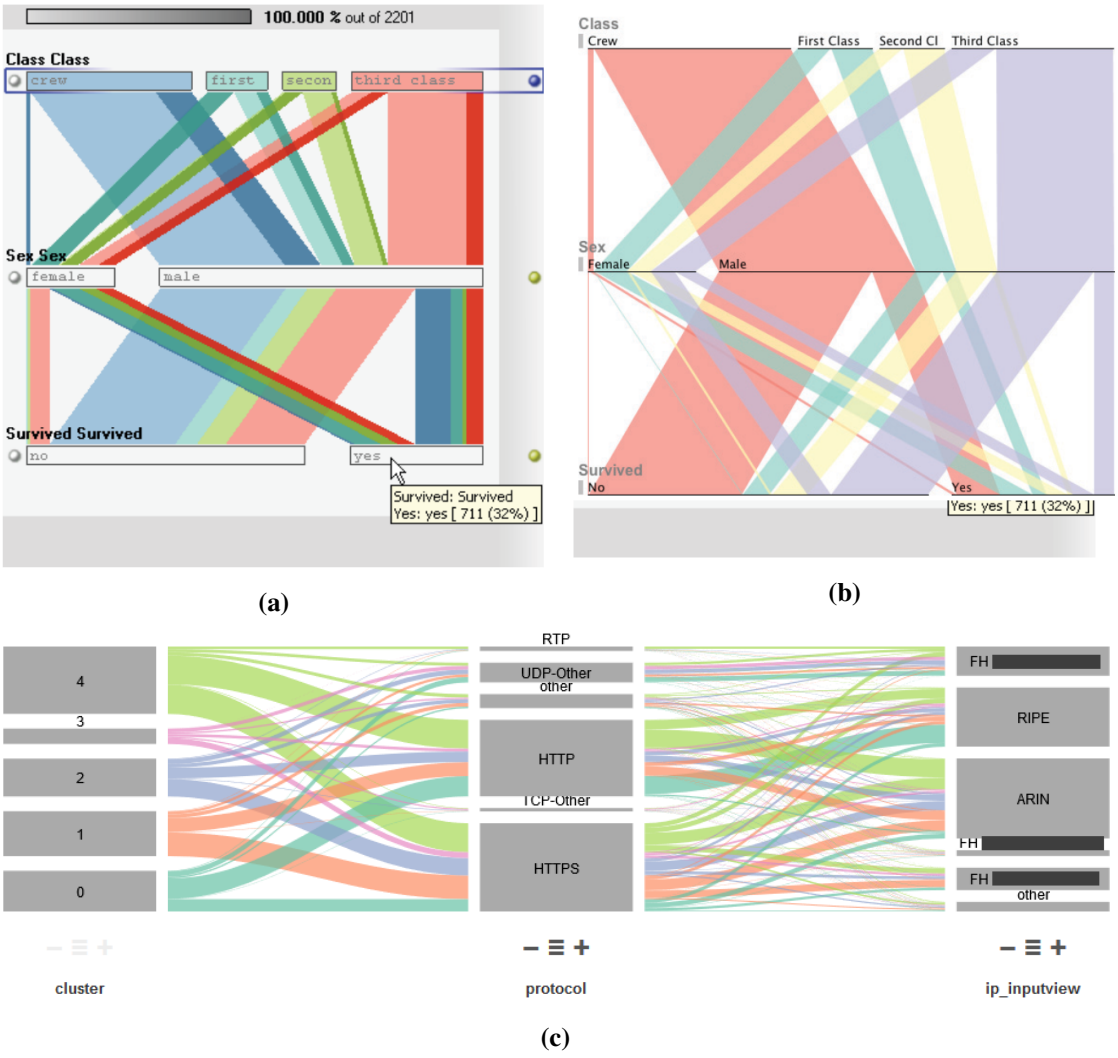


Abbildung 5.6.: Gegenüberstellung der Parallel Sets Designs. (a) Das ursprüngliche Design der Parallel Sets, (b) das überarbeitete Design nach Kosara [Kos10] und das in dieser Arbeit Verwendung findende Design (c). Bildquelle: [Kos10].

5.3. Ebene 2 – Parallel Sets und Dimensionsfilter

In der zweiten Ebene wird der Fokus auf die multivariaten Daten gelenkt. Die hier verwendete Parallel Sets Ansicht [BKH05; KBH06] ist eine der wenigen Visualisierungstechniken, die speziell für kategoriale Daten entwickelt wurden. In dieser Ansicht werden die Daten, die durch die erste Ebene ausgewählt wurden, nach mehreren Dimensionen aufgeschlüsselt dargestellt. Dabei wird ersichtlich, welche Kategorien in der Dimension am häufigsten vorkommen, ob und welche Korrelationen zwischen den Dimensionen bestehen und ob es Häufungen der Daten im mehrdimensionalen Datenraum gibt. Die Darstellung gibt somit einen Überblick darüber, welche Netzwerkdaten in dem gewählten Zeitraum vorhanden sind. Es macht Sinn, diese Eigenschaften zu nutzen um direkt auf Basis der dargestellten Kategorien Filter zu implementieren. Dadurch können die Anwender/-innen, im Gegensatz zu dem üblichen Einstellen der Parameter in einem Menü, bereits bei der Auswahl

sehen ob Daten zu den ausgewählten Einstellungen vorhanden sind. Die gefilterten Daten können sowohl in der Darstellung selbst betrachtet, als auch für die darunterliegenden Ebenen bereitgestellt werden. Zusammenfassend übernimmt die zweite Ebene zwei wesentliche Aufgaben:

- Überblick über die Zusammensetzung des Netzwerkverkehrs
- Dimensionsfilter für die dritte Ebene

Um diese beiden Aufgaben zu erfüllen bedarf es allerdings einiger Änderungen des ursprünglichen Designs. Sowohl das Design des Originalpapers, als auch das überarbeitet Designs nach Kosara [Kos10], sind für die Darstellung von Netzwerkdaten nicht geeignet. Durch die bei Netzwerkdaten übliche hohe Anzahl an Kategorien pro Dimension werden die Abschnitte auf den Achsen für einen Großteil der Kategorien so klein, dass sich die dazugehörigen Beschriftungen stark überschneiden. Zudem müssen dadurch viele kleine Pfade eingezeichnet werden, die kaum den richtigen Kategorien zugeordnet werden können. *Visual Clutter* ist die Folge. Da die Kategorien in dieser Arbeit zusätzlich als Filter fungieren, kommt erschwerend hinzu, dass die visuellen Primitiven zu klein sind um Interaktionen darauf auszuführen. Das in dieser Arbeit verwendete Design unterscheidet sich deshalb in einigen Punkten noch einmal deutlich von den beiden anderen Parallel Sets Designs. In Abbildung 5.6 sind alle 3 Varianten abgebildet. Die wichtigsten Änderungen und Designentscheidungen werden im Folgenden vorgestellt und begründet.

Horizontales Layout mit Boxen Das neue Design verwendet, wie im Originalpaper, Boxen für die Darstellung der Kategorien auf den Achsen. Allerdings wird die Box hier nicht dazu verwendet um Histogramme anzuzeigen, sondern um mehr Fläche für die Darstellung und das Handling des Filters und die Labels zu haben. Im Gegensatz zu den anderen Varianten verwendet diese Arbeit ein horizontales Layout. In Kombination mit den Boxen können so auch lange Bezeichnungen für sehr schmale Kategorien überschneidungsfrei dargestellt werden. Auch für die Einbettung in der zweiten Ebene ist diese Ausrichtung besser geeignet. Da auch in diesem Design in kleinen Kategorien kein Platz für Beschriftungen ist, wird die Beschriftung in diesen Fällen auf die Box aufgesetzt. Ein Minimalabstand zwischen den Kategorien, sowie eine Minimalhöhe der Boxen von einem Pixel sorgen dafür, dass alle Kategorien beschriftet werden können. Die Proportionalität geht nur im Subpixelbereich verloren.

Beschränkung auf Top 5 Eine weitere Erweiterung der Parallel Sets ist nötig, da speziell bei Netzwerkdaten viele Kategorien möglich sind und diese aus mehreren Gründen nicht alle in die Ansicht integriert werden können. Aus diesen Gründen werden nur die Top 5 Kategorien pro Dimension eingezeichnet und alle weiteren Kategorien als Restklasse mit der Bezeichnung *other* in die Ansicht integriert. Die Restklassen nehmen dabei eine wichtige Funktion ein. Sie sind ein Anhaltspunkt für die Anwender/-innen, wie viel Prozent der Daten nicht in die aktuelle Top 5 der Dimension fallen. Auf Basis dieser Anzeige können die Anwender/-innen entscheiden, ob nach weiteren Kategorien aufgeschlüsselt werden soll. Das Hinzufügen und Entfernen von Kategorien ist dabei sowohl aus einer Liste, als auch durch Shortcuts unter den Achsen möglich. Durch die Shortcuts kann beispielsweise schnell von der Top 5 auf die Top 6 gewechselt werden.

Dimensionsfilter Nach dem Vorbild des *Brushing* in Parallelen Koordinaten wird ein Dimensionsfilter eingeführt, mit dem die Anwender/-innen in der Lage sind einzelne Kategorien zu selektieren und diese so zu einem Filter hinzuzufügen. Dadurch kann sowohl der Fokus auf eine Teilmenge der Daten gelegt werden, als auch die Darstellung auf diese Teilmenge

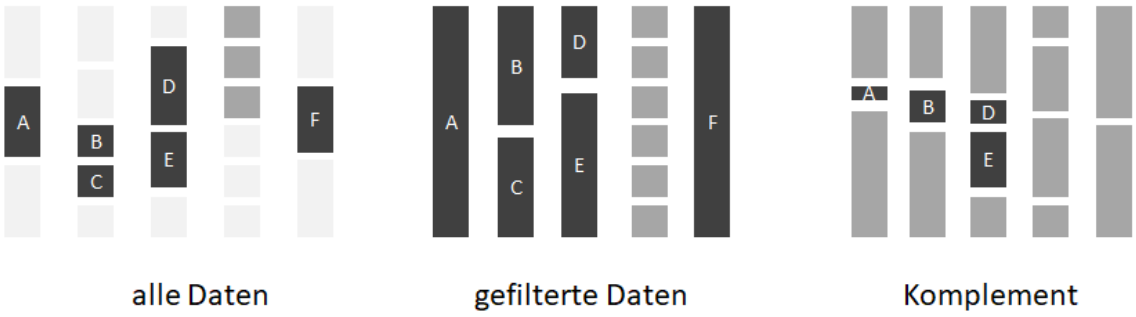


Abbildung 5.7.: Die Parallel Sets Darstellung bietet drei Ansichten der Daten: alle Daten, gefilterte Daten und das Komplement des Filters.

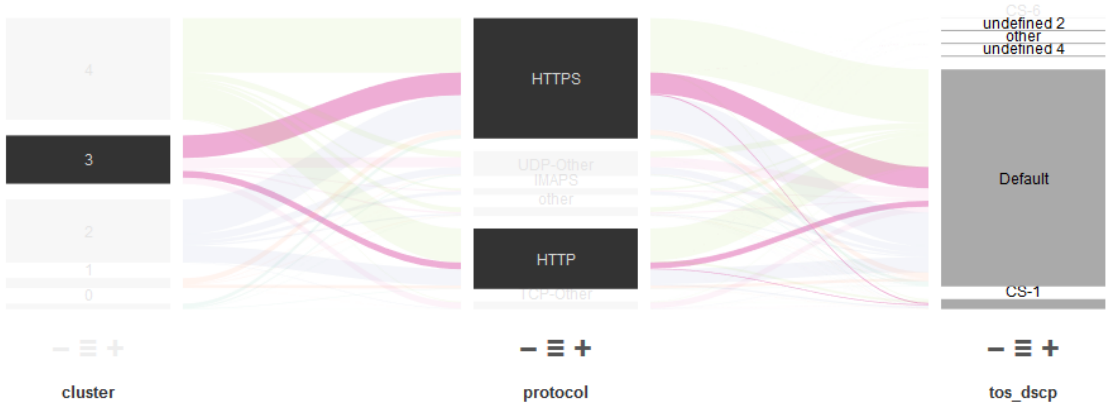


Abbildung 5.8.: Darstellung des Dimensionsfilters. Durch die Selektion der drei Kategorien (dunkelgrau) werden alle anderen Kategorien und Pfade die nicht zur Filtermenge gehören automatisch visuell in den Hintergrund gestellt.

beschränkt werden. Auch im ursprünglichen Design der Parallel Sets ist ein Filter vorgesehen. Dieser erlaubt es, einzelne Kategorien aus der Ansicht auszuschließen. In dieser Arbeit bleiben allerdings alle Daten in der Ansicht zunächst erhalten und verringern sich erst wenn die Anwender/-innen dies explizit fordern. So bleibt der Kontext erhalten und die Filterauswirkungen sind direkt sichtbar. Da es sich dabei um ein Kernstück der Arbeit handelt, wird dieser noch einmal ausführlich in Abschnitt 5.3.1 behandelt.

5.3.1. Dimensionsfilter

Die wohl wichtigste Erweiterung der Parallel Sets in dieser Arbeit ist die Integration des Dimensionsfilters. Dabei wird eines der im Experteninterview (vgl. Abschnitt 4.2) genannten Probleme direkt adressiert: Das Einstellen leerer Ergebnismengen bei der Parameterauswahl. Theoretisch ist die Auswahl eines Filters mit leerer Ergebnismenge zwar möglich, jedoch kann der Nutzer bereits während der Auswahl sehen, ob Netzwerkdaten für bestimmte Einzelwerte und Kombinationen existieren und ist sich seiner Auswahl deshalb bewusst.

Für den Einsatz in dieser Arbeit fiel die Wahl auf eine einfache Filter-Variante, die zwar nicht mächtig genug ist, um beliebige Teilmengen der Netzwerkdaten zu modellieren, dafür aber leicht verständlich und nachvollziehbar ist. Das Design orientiert sich dabei an der bereits bekannten Brushing-Interaktion Paralleler Koordinaten. Werden auf einer Achse Kategorien zum Filter hinzugefügt, beschränkt die die Datenmenge auf die ausgewählten Kategorien. Achsen für die keine Kategorien ausgewählt wurden beeinflussen den Filter nicht. Visuell werden die Kategorien, die sich im Filter befinden, dunkelgrau auf den Achsen dargestellt. Um die Auswirkungen der Filter zusätzlich sichtbar zu machen, werden Pfade, die sich auf nicht mehr mögliche Netzwerkdaten beziehen, sowie Kategorien, zu denen aufgrund der Filtereinstellungen keine Netzwerkdaten mehr vorhanden sind, optisch in den Hintergrund gerückt (siehe Abbildung 5.8).

Der Dimensionsfilters kann auf zwei Weisen bei der Exploration verwendet werden: Zum einen dient er, wie zuvor der Zeitfilter, als Filter für die darunter liegenden Ebenen. Über den Dimensionsfilter können so Teilmengen des Netzwerkverkehrs auf Basis der Werte ausgewählt werden und diese in einer der darunterliegenden Ebenen betrachtet werden. Zum anderen bietet der Dimensionsfilter einen Vorteil bei der Exploration der Daten in der Parallel Sets Ansicht selbst. Durch das Hinzufügen von Kategorien zum Filter werden alle nicht passenden Daten automatisch optisch in den Hintergrund gerückt. Dies ermöglicht die Fokussierung auf bestimmte Datenmengen, ohne den Kontext der restlichen Menge zu verlieren. Möchte man sich ausschließlich auf die gefilterten Daten beschränken, kann man in die Ansicht auf die gefilterten Daten wechseln. Hier geht zwar der Kontext verloren, dafür vergrößert sich aber die Ansicht auf die Daten. Dies ist besonders wichtig wenn der Fokus auf einer sehr kleinen Teilmenge der Daten liegt, die in der Darstellung aller Daten untergehen würde. Zudem kann auch das Komplement des Filters betrachtet werden. Abbildung 5.7 zeigt einen Überblick über die frei möglichen Parallel Sets Ansichten.

5.3.2. Nicht disjunkte Teilmengen

Aufgrund der Definition der Views als beliebige nicht disjunkte Teilmengen der IP-Subnetze oder Interfaces kann ein Datenelement mehreren Views zugewiesen werden. Sowohl in den Parallelen Koordinaten als auch in den Parallel Sets wird eine solche Mehrfachzuweisung allerdings nicht unterstützt. Um zu disjunkten Teilmengen zu gelangen, wird für diese Arbeit die Anzeige auf *Teilmengen-Kategorien* beschränkt, wie sie in Abbildung 5.9 definiert werden. Für eine Menge von ausgewählten Kategorien $N \subseteq \mathcal{K}(d)$ einer Achse (beispielsweise die Top $|N|$ der Dimension) müssen bei diesem Ansatz im Worst Case m Teilmengen-Kategorien daraus erzeugt werden:

$$m := |\mathcal{P}(N \subseteq \mathcal{K}(d))| = 2^{|N|} - 1 \tag{5.1}$$

Für $|N| = 5$, der Standardanzahl an Kategorien pro Achse, sind das bereits 32 Teilmengen-Kategorien. Beschränkt man sich hingegen auf eine hierarchische Ordnung der Teilmengen, d.h. $\forall A, B \in \mathcal{P}(\mathcal{K}) : A \cap B = \emptyset \vee A \subseteq B \vee A \supseteq B$, reduziert sich die Worst-Case-Anzahl an Teilmengen-Kategorien m_r auf:

$$m_r := |N| \tag{5.2}$$

Da es sowohl bei den IPViews als auch bei den InterfaceViews Sinn macht, weitestgehend hierarchisch geordnete Teilmengen zu definieren und beliebige Teilmengen eher selten sind, bleibt die Anzahl an Teilmengen-Kategorien weitestgehend proportional zu der Anzahl der Kategorien. Für die Bestimmung der Top X wird nicht die Kardinalität der Teilmengen-Kategorien, sondern die der atomaren Kategorien zugrunde gelegt.

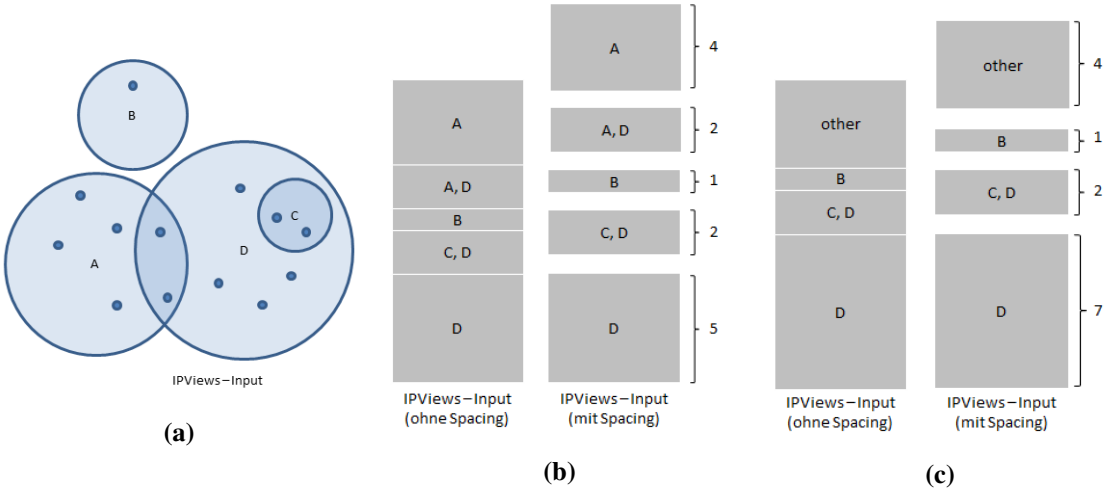


Abbildung 5.9.: Veranschaulichung der Konstruktion der Teilmengen-Kategorien auf den Achsen ohne Mehrfachzählungen der Datenpunkte anhand eines Beispiels mit insgesamt $N = 13$ Datenpunkten und den Teilmengen (Views) A, B, C und D. (a) Euler-Diagramm, welches die Zuordnung der Datenpunkte des Beispiels zu den Views aufzeigt. (b) Darstellung der resultierenden Achse, wenn die Daten nach allen Views aufgeschlüsselt werden sollen. Auf der linken Seite werden die Werte ohne Abstand zueinander angezeigt, auf der rechten Seite mit dem Abstand, der auch im Prototyp realisiert wurde. (c) Darstellung der resultierenden Achse, wenn nur die Views B, C und D angezeigt werden sollen. Man sieht, dass sich die komplette Höhe nicht verändert, da kein Datenpunkt mehrfach in der Darstellung aufgeführt wird. Daten die keiner ausgewähltem View zugewiesen werden konnten, sind nun der Restklasse *other* zugeordnet.

5.4. Ebene 3 – Detailansichten

Die dritte und letzte Ebene der Benutzeroberfläche bildet gleichzeitig den Abschluss des Workflows und dient vorrangig der Darstellung detaillierter Ansichten der Daten. Da auf dieser Ebene sowohl der zeitliche Filter der ersten Ebene, als auch der Dimensionsfilter der zweiten Ebene Anwendung finden, sind die hier dargestellten Daten bereits reduziert. Im Fall der Streamgraph Komponente besteht darüber hinaus eine starke Interaktivität mit der Parallel Sets Ansicht, die für die Exploration der Daten von besonderer Bedeutung ist. Im letzten Abschnitt dieses Kapitels wird deshalb noch einmal genauer auf die Interaktivität zwischen den Ebenen eingegangen. Insgesamt bringt die dritte Ebene nicht zuletzt dadurch wieder den zeitlichen Bezug der Daten in den Fokus, der in der Parallel Sets Ansicht nur eine untergeordnete Rolle gespielt hat.

Durch den modularen Aufbau der Benutzeroberfläche, besteht die Möglichkeit, auf dieser Ebene noch weitere Detailansichten als Option für die Analyse anzubieten. Abhängig von den Anforderungen an das System können entsprechende Module eingesetzt werden, die bestimmte Aspekte der Daten genauer beleuchten. Ebenfalls besteht die Möglichkeit, an Stelle der Clusteranalyse auf der ersten Ebene beispielsweise andere Machine-Learning Ansätze einzusetzen. Um dem Workflow des Konzepts zu entsprechen, sollte bei der Wahl der ersten Ebene allerdings darauf geachtet werden, die Daten vollständig und im zeitlichen Verlauf anzuzeigen.

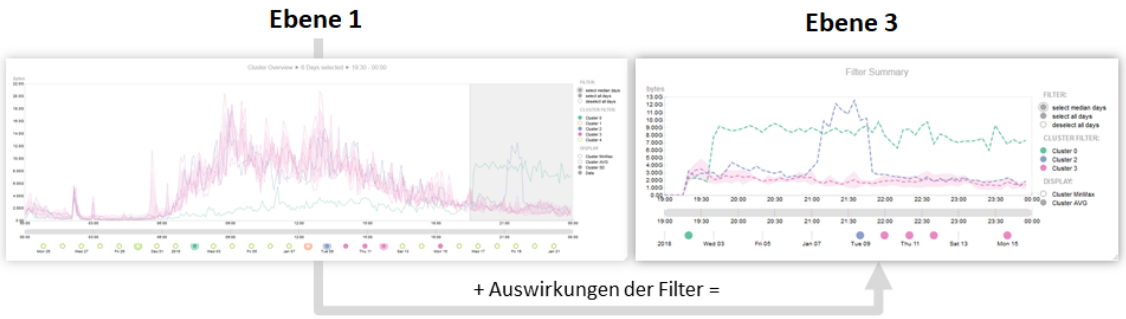


Abbildung 5.10.: Die Auswirkungen der auf Ebene 1 eingestellten Filtereinstellungen (linke Seite), sind auf Ebene 3 (rechte Seite) zu sehen. Wird zusätzlich auf Ebene 2 ein Dimensionsfilter definiert, sind dessen Auswirkungen ebenfalls auf Ebene 3 und somit in der rechten Seite sichtbar.

5.4.1. LineChart

Mit der LineChart Komponente werden die Daten auf der Detailebene wieder im zeitlichen Verlauf dargestellt. Es handelt sich dabei um die gleiche Darstellungsform wie bei der ersten Ebene, mit dem entscheidenden Unterschied, dass auf der dritten Ebene bereits die Filter der ersten beiden Ebenen auf die Daten angewendet wurden. Die Komponente erfüllt dabei eine vorrangige Aufgabe: In dieser Ansicht wird ersichtlich, welche Auswirkungen die verwendeten Filter auf die Daten haben. Um den Überblick über die Daten nicht zu verlieren, werden die Auswirkungen der Filter nicht direkt auf der ersten Ebene, sondern dem Workflow entsprechend ausschließlich auf den darunterliegenden Ebenen sichtbar. Da es dennoch sinnvoll ist zu sehen, welche Daten aus den Filtern resultieren, ist die selbe Ansicht nochmal auf Ebene 3 zu finden. Die LineChart Ansicht dient somit als Kontrollansicht und visuelle Zusammenfassung der Filterauswirkungen der darüberliegenden Ebenen (siehe Abbildung 5.10).

5.4.2. Streamgraph

Das Hauptaugenmerk der Ebene 3 liegt auf der Streamgraph Ansicht. Auch hier steht wieder der zeitliche Bezug der Daten im Vordergrund, allerdings geht es hierbei um die Darstellung der Details und nicht um die Darstellung der Filterauswirkungen. Im Gegensatz zur LineChart-Komponente der Ebene 3, besteht hier eine starke Interaktivität mit der Parallel Sets Ansicht, die für die Exploration der Daten von besonderer Bedeutung ist. Im letzten Abschnitt dieses Kapitels wird deshalb noch einmal genauer auf die Interaktivität zwischen den Ebenen eingegangen. Dieser Abschnitt befasst sich zunächst isoliert mit den Funktionen und Darstellungen der Streamgraph Komponente.

Die Streamgraph Ansicht besteht im wesentlichen aus den beiden in Abbildung 5.11 dargestellten Graphen. Der eigentliche Streamgraph im Vordergrund stellt die ausgewählten Daten aufgeschlüsselt nach den Werten einer Dimension dar. Im Hintergrund befindet sich ein weiteres Flächendiagramm, im Folgenden *Graph-Silhouette* genannt, welches alle Daten über den ausgewählten Zeitraum darstellt. Die Graph-Silhouette dient als Referenzlinie, um die Auswirkungen des Dimensionsfilters einordnen zu können. Beide Graphen werden nach Vorbild der Braided Graphs [JME10] ineinander verflochten, sodass der Streamgraph nur in den durch den Zeitfilter definierten Zeitintervallen

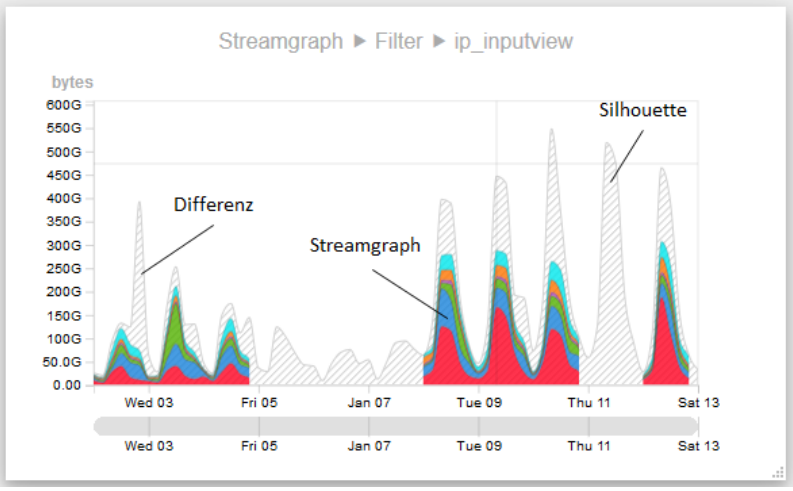
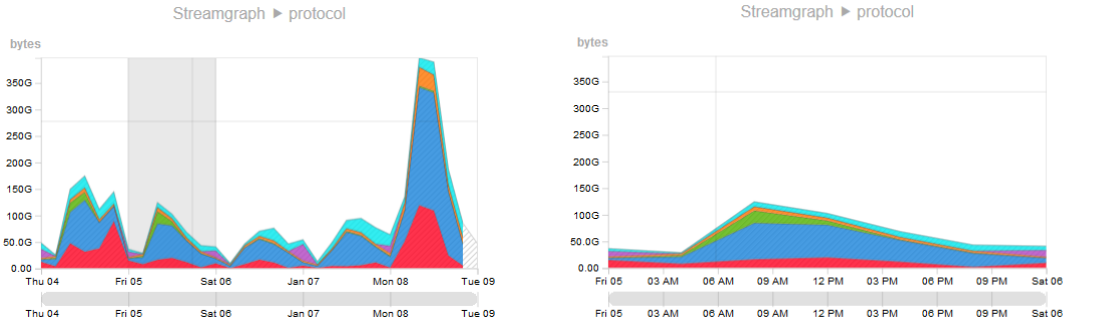


Abbildung 5.11.: Übersicht über die Komponenten der Streamgraph Ansicht.

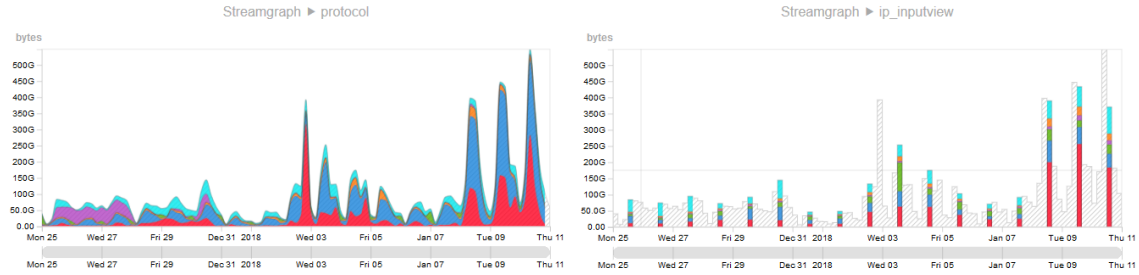


(a) **(b)**

Abbildung 5.12.: Streamgraph Tages-Zoom. Durch Anklicken des in (a) selektierten Tages, wird dieser auf das Sichtfenster vergrößert (b).

abgebildet wird. Die *Graph-Silhouette* im Hintergrund bleibt allerdings immer erhalten, um das Ergebnis der Filtereinstellungen im Kontext der restlichen Daten zu sehen. Dadurch wird ersichtlich, welche Daten durch den Filter in der zweiten Ebene verworfen wurden, indem die Differenz der Oberkanten von Streamgraph und Silhouette betrachtet wird.

Um das Sichtfenster anpassen zu können sind einige Zoom-Interaktionen in dieser Ansicht integriert worden. Neben einer horizontalen Scrollbar und der Möglichkeit mit dem Mausrad vertikal zu zoomen, existiert zudem die Möglichkeit die Ansicht auf einen Tag zu beschränken, um diesen genauer betrachten zu können (siehe Abbildung 5.12). Diese Designentscheidung wurde gewählt, da das Anzeigen aller Daten in der gewünschten Granularität im Streamgraph zu visuellen Problemen und Performanzschwierigkeiten geführt hat. Aus diesem Grund wird im Streamgraph in der globalen Sicht eine grobe Granularität gewählt und erst in der Tagesansicht werden die Daten in der gewünschten Granularität angezeigt.



(a) Lineare Interpolation **(b)** Step After

Abbildung 5.13.: Streamgraph mit unterschiedlichen Interpolationen. Wenn in der ersten Ebene nur ein Zeitpunkt ausgewählt wurde, wechselt die Ansicht auf die Interpolation in (b).

Eine weitere Besonderheit der Darstellung ist in Abbildung 5.13 zu sehen. Wird in der ersten Ebene keine Zeitspanne sondern ein Zeitpunk ausgewählt, hätte dies zur Folge, dass die einzelnen Abschnitte des Streamgraphs mit einer Breite von 0 Pixeln nicht angezeigt werden könnten. Um dieses Problem zu lösen, wird in diesem Fall zu einer anderen Interpolation gewechselt.

Welche Dimension aktuell zu sehen ist, nach welchen Werten diese Dimension aufgeschlüsselt werden soll und ob die Darstellung die ungefilterten, gefilterten Daten oder das Komplement dieser anzeigt, wird durch Interaktion mit der Parallel Sets Komponente bestimmt.

Interaktion und Highlighting mit der zweiten Ebene

Das Zusammenspiel aller Komponenten bildet das Herzstück des Prototypen. Jede Ebene stellt nur einen Aspekt der Daten dar und erst durch die Interaktionen zwischen den Ebenen werden die Verbindungen zwischen den Daten verdeutlicht. Die Parallel Sets Ansicht und der Streamgraph sind dabei besonders stark aneinander gekoppelt. Nicht nur der Dimensionsfilter entscheided über das Aussehen des Streamgraphs, sondern auch das Highlighting der Parallel Sets spiegelt sich im Streamgraph wider. Die aktuelle Position der Maus in den Parallel Sets entscheidet sowohl über die angezeigte Dimension als auch über eventuelle im Fokus stehende Kategorien im Streamgraph (siehe Abbildung 5.14). Zwischen anderen Ebenen existiert keine derartig enge Kopplung. Diese sind ausschließlich durch den Workflow und Datenfluss des Konzept miteinander verbunden.

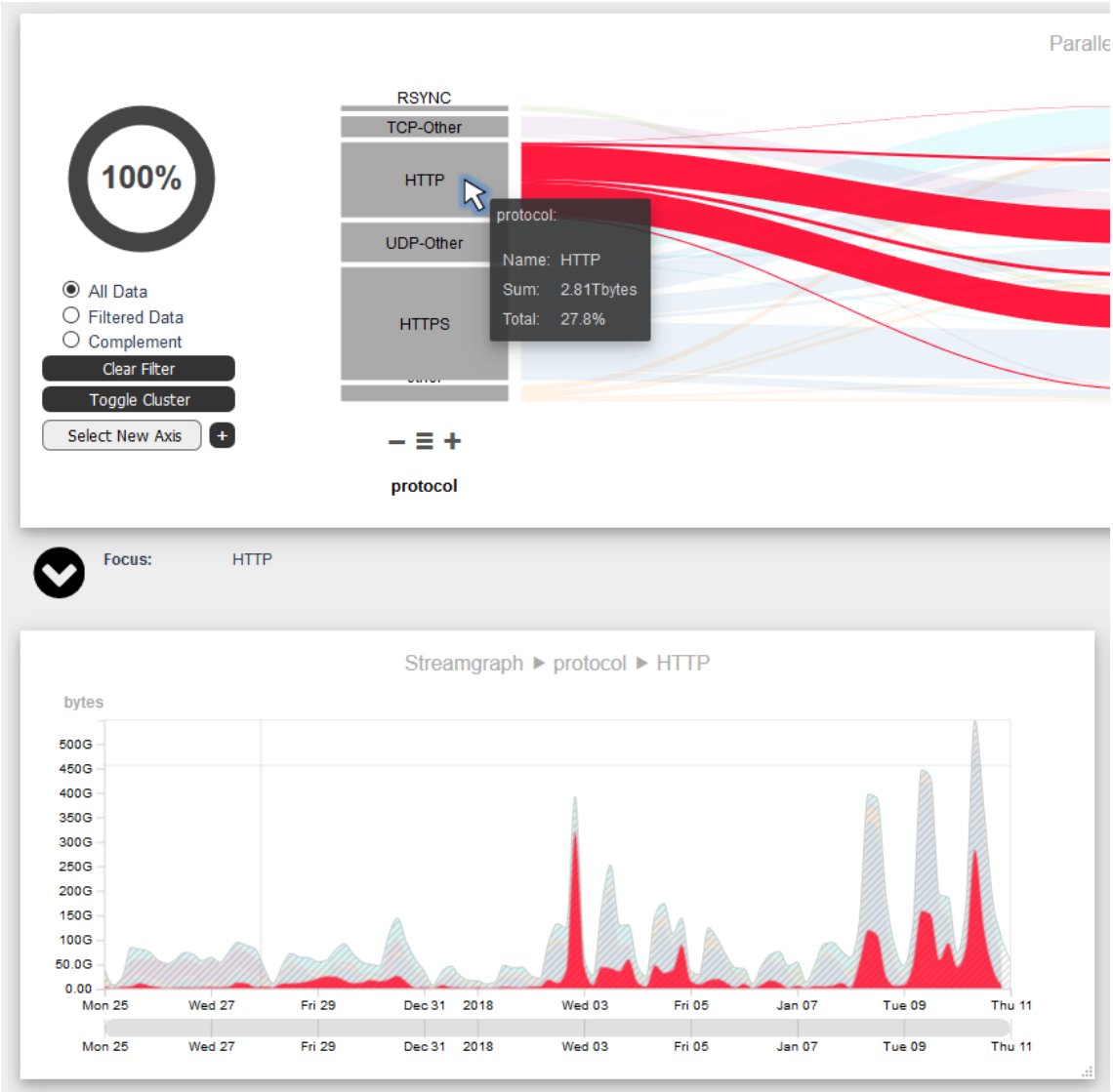


Abbildung 5.14.: Highlighting zwischen den Ebenen. Da die Maus in der zweiten Ebene auf HTTP zeigt, wird im Streamgraph HTTP gehighlightet und zusätzlich als unterste Kategorie angezeigt. Dadurch kann die Höhe der Peaks besser bestimmt werden.

6. Implementierung

Nach den Vorgaben der Aufgabenstellung wurde der entwickelte visuelle Ansatz in Form eines webbasierten Softwareprototypen umgesetzt. Dadurch wird ein hoher Grad an Plattformunabhängigkeit erreicht, da für die Nutzung der Webanwendung lediglich ein Webbrowser vorausgesetzt wird. Abbildung 6.1 gibt einen High-Level Überblick über die verwendeten Technologien und die Kommunikation zwischen den Komponenten der Webanwendung. Die Implementierung des Prototypen lässt sich in drei Teile untergliedern, denen in diesem Kapitel je ein Abschnitt gewidmet ist:

- Datenmanagement und Aufbereitung der Daten
- Backend
- Frontend

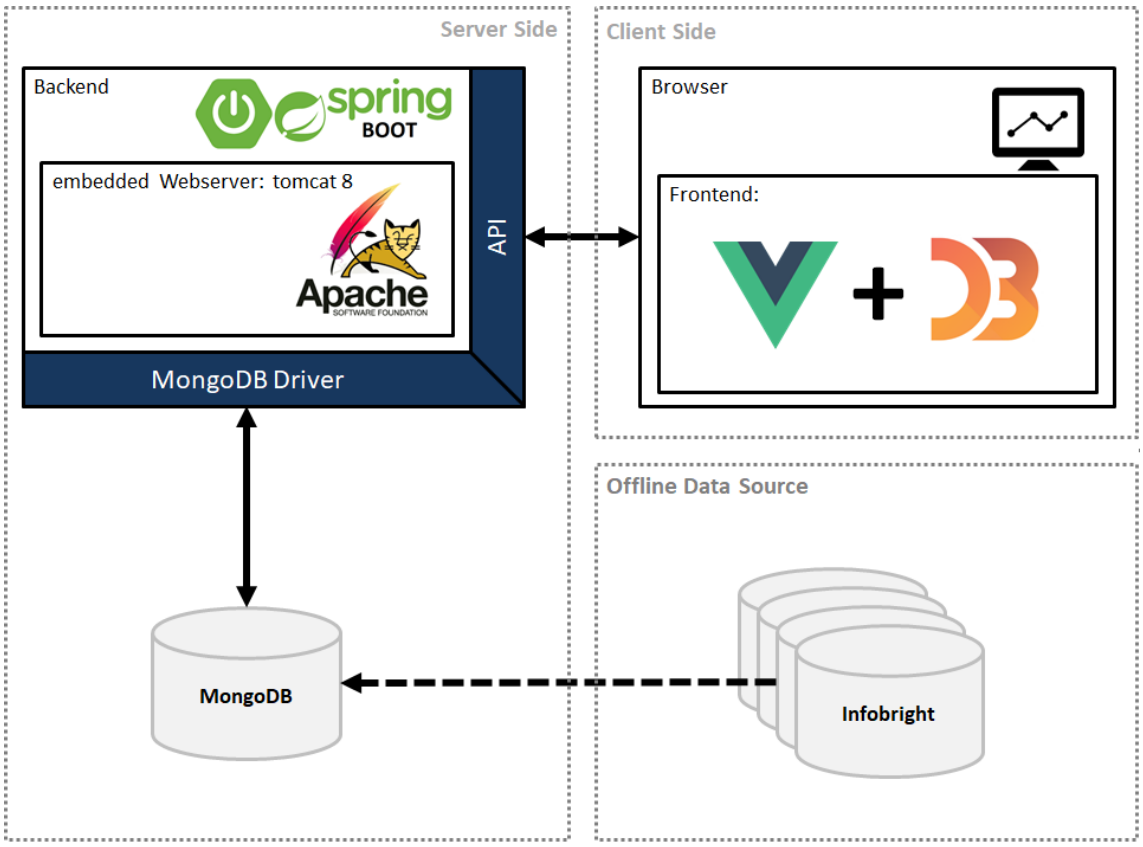


Abbildung 6.1.: High-Level Überblick der Webanwendung

Die Wahl für die Entwicklung des Frontends fiel auf Vue.js[1], ein leichtgewichtiges und progressives JavaScript Framework für die Entwicklung von Web-Benutzeroberflächen. Vue.js bietet einen leichten Einstieg in die JavaScript Frontend-Entwicklung und eignet sich dadurch vor allem für den Einsatz in kleinen Projekten wie diesem [You18]. Alle graphischen Ansichten wurden mit Hilfe von D3.js[2] realisiert. D3.js ist eine JavaScript-Bibliothek, welche direkte Manipulationen auf dem Document Object Model (DOM) erlaubt und Datenobjekte an dessen Elemente binden kann [**d3**]. Dies ermöglicht es, das Aussehen, die Positionierung und andere Attribute der Elemente auf Basis der Daten einfach zu modellieren, Event-Listener und Interaktionen einzubauen und erleichtert so die Arbeit mit den Elementen. Das Einbinden von JavaScript-Modulen – wie D3.js (`d3`) – wird in Vue.js mit npm[3] realisiert. Weitere Module, die im Prototyp Verwendung gefunden haben, sind: ColorBrewer (`colorbrewer`), D3 Context Menu (`d3-context-menu`), und Google Colour Palette Generator (`google-palette`).

Das Java-Backend wurde mit Hilfe von Spring Boot[4] umgesetzt. Spring Boot unterstützt die Entwicklung eigenständiger JVM-basierter Anwendungen durch einfache Konfiguration der benötigten Abhängigkeiten und Bibliotheken [Piv18]. Durch den eingebetteten Apache Tomcat 8.5 Webserver[5] kann die komplette Webanwendung, inklusive des Webservers und der von Vue.js erzeugten Frontend-Dateien, in eine JAR-Datei integriert werden.

Für das Datenmanagement im Hintergrund wurde MongoDB[6] ausgewählt. MongoDB zählt zu den dokumentenorientierten NoSQL-Datenbanken und verspricht große Flexibilität, hohe Geschwindigkeit und gute horizontale Skalierbarkeit, welche für die Arbeit mit großen Datenmengen von besonderem Vorteil ist. Zudem unterstützt MongoDB durch die integrierte Aggregation Pipeline eine flexibel einsetzbare und einfache Methode, um die Daten bereits in aggregierter Form anzufordern. Der nächste Abschnitt befasst sich mit dem ersten Teil der Implementierung: Datenmanagement und Aufbereitung der Daten.

6.1. Datenmanagement und Aufbereitung der Daten

IsarNet verwendet für die Datenhaltung von IsarFlow Infobright[7], eine kommerzielle spaltenorientierte Datenbank mit Fokus auf High Performance Data Analytics [Ign18]. Wie in Relationalen Datenbank Management Systemen (RDBMS) üblich, werden die Daten auch bei IsarFlow aufgrund der Normalisierung auf mehrere Relationen aufgeteilt, um möglichst wenig Redundanzen in den Daten zu besitzen. Dokumentbasierte Datenbanken wie MongoDB arbeiten hingegen überlicherweise mit Datenstrukturen, die möglichst bereits alle benötigten Informationen in einem Dokument vereinen. Die Migration der bereitgestellten Datensätze zu MongoDB erfordert somit, alle Informationen zu einem Eintrag der Hauptrelation mit den Informationen verknüpfter Relationen in einem Dokument zusammenzuführen. Dazu werden alle Attribute der Hauptrelation als Felder im Dokument übernommen und Informationen aus anderen Relationen, die über Fremdschlüssel

[1] Vue.js – The Progressive JavaScript Framework https://vuejs.org/
[2] D3.js V4– Data-Driven Documents https://d3js.org/
[3] npm – package manager for JavaScript https://www.npmjs.com/
[4] Spring Boot https://spring.io/
[5] Apache Tomcat8.5 https://tomcat.apache.org/
[6] MongoDB https://www.mongodb.com/
[7] Ignite InfobrightDB http://www.ignitetech.com/solutions/information-technology/infobrightdb

verknüpft sind, als Unterdokumente integriert. Die Aufteilung der Daten in Tabellen mit unterschiedlicher Granularität und die Partitionierung in Wochen- oder Tagestabellen (abhängig von der Granularität) wurden dabei für die Erzeugung der MongoDB Collections direkt von IsarFlow übernommen. Tabelle 6.1 fasst den Speicherbedarf und die Anzahl der Dokumente in MongoDB für die bereitgestellten Datensätze zusammen. In dieser Arbeit werden darüber hinaus weitere Collections verwendet, die die Anfragen an die Datenbank beschleunigen sollen. Sie werden im folgenden Abschnitt beschrieben.

	Hochschule		**Deutsche Bahn**	
Zeitraum	4 Wochen		8 Wochen	
Granularitäten	4h, 1h, 5min, 1min		1h	
Speicherbedarf je Granularität	4h:	4,8 GiB		
	1h:	10,1 GiB	1h:	290,2 GiB
	5min:	36,1 GiB		
	1min:	79.2 GiB		
	Summe:	**130,2 GiB**	Summe:	**290,2 GiB**
Anzahl der Dokumente je Granularität	4h:	19.750.265		
	1h:	41.183.249	1h:	1.157.917.327
	5min:	147.291.106		
	1min:	322.750.403		
	Summe:	**530.975.023**	Summe:	**1.157.917.327**
Anzahl der Exportserver	1		18	
Exportformat	CSV		CSV	

Tabelle 6.1.: Beispieldatensätze – eine Übersicht über die wichtigsten Kenngrößen der vollen Daten.

6.1.1. TopX-Collections und Dimension-Collections

Mit einem Blick auf die Größenordnungen in Tabelle 6.1 wird ersichtlich, dass für die Analyse sehr viele Daten im Hintergrund verarbeitet werden müssen, um die im Verhältnis recht kleinen und stark aggregierten Datenmengen für die Ansichten in der Webanwendung bereitzustellen. Betrachtet man beispielsweise die Darstellung auf der zweiten Ebene, sieht man, dass sich häufig die gleichen Werte in den Top 5 der Dimensionen befinden und sich oft nur die Verhältnisse verändern. Ein Großteil der Kategorien wird dabei fast immer der Restklassen-Kategorie jeder Dimension zugeordnet, muss allerdings trotzdem aufwändig aus den vollen Daten berechnet werden. Vor allem bei großen Datenmengen – bei denen sich die Ergebnisse erwartungsgemäß dem globalen Mittelwert annähern – ist die Berechnung besonders zeitaufwendig. Diese Beobachtungen motivieren die Einführung der *TopX-Collections*, die die Daten bereits hinsichtlich der am häufigsten vorkommenden Kategorien voraggregieren. Dazu werden zunächst die Top X Kategorien jeder Dimension der Daten berechnet. Besitzt ein Dokument in einer Dimension eine Kategorie, die nicht

Listing 6.1 Dokumentenstruktur anhand eines Beispiels aus der Full Collection `collection_full` und einem Beispiel aus der Dimension-Collection `collection_protocol`. Anhand der Werte der aggregierten Felder `bytes`, `packets` und `flows` ist klar zu erkennen, dass die Dokumente in `collection_protocol` bereits stark aggregiert werden konnten.

```
{
  "_id" : ObjectId("..."),
  "timestamp" : NumberLong(1514160000),
  "mod" : 0.0,
  "interface" : {
    "ed_ip" : NumberLong(2370963973),
    "input" : NumberInt(1),
    "output" : NumberInt(0)
  },
  "ip" : {
    "input" : NumberInt(19),
    "output" : NumberInt(146)
  },
  "protocol" : {
    "element_id" : NumberLong(377)
  },
  "tos" : {
    "tos" : NumberInt(0)
  }
  "ipversion" : NumberInt(4),
  "inputvrf" : NumberLong(0),
  "direction" : false,
  "bytes" : NumberLong(40),
  "packets" : NumberLong(1),
  "flows" : NumberLong(1),
}
```

Listing 6.1 collection_full

```
{
  "_id" : ObjectId("..."),
  "timestamp" : NumberLong(1514160000),
  "mod" : 0.0,
  "protocol" : {
    "element_id" : NumberLong(7),
    "id" : NumberLong(7),
    "name" : "MICROSOFT-DS",
    "des" : ""
  },
  "bytes" : NumberLong(821129),
  "packets" : NumberLong(17017),
  "flows" : NumberLong(16679),
}
```

.

Listing 6.2 collection_protocol

zur Top X gehört, wird der Wert an dieser Stelle im Dokument durch *other* ersetzt. Durch erneute Aggregation der resultierenden Dokumente können die Anzahl der Dokumente in der Collection und dadurch auch die Antwortzeiten auf Anfragen stark reduziert werden. Durch die Einführung der *TopX-Collections* wird die Genauigkeit nicht reduziert. Gibt es eine Differenz zwischen der zeitlich globalen und lokalen Top-Liste, werden die fehlenden Daten zusätzlich aus den vollständig aufgelösten Daten angefragt und mit den Ergebnissen der TopX-Anfrage zusammengeführt. Dies ist Aufgabe des Backends und wird in Abschnitt Abschnitt 6.2.2 noch einmal genauer behandelt. Die Wahl von X entscheidet darüber, wie stark die Daten komprimiert werden können. Wird X zu groß gewählt, müssen zwar in den meisten Fällen keine Daten nachgeholt werden, allerdings sind die Collections größer und Anfragen darauf langsamer. Auf der anderen Seite darf X nicht zu klein gewählt werden, da sonst zu viele Anfragen auf die teureren vollen Daten benötigt werden. Für diese Arbeit hat $X = 10$ zu guten Ergebnissen geführt. Eine Optimierung des Wertes wurde nicht durchgeführt.

Aus den gleichen Gründen wird eine weitere Gruppe an Collections eingeführt: die *Dimension-Collections*. Diese beinhalten nur die Felder, die für die Berechnungen der jeweiligen Dimension benötigt werden und Beschleunigen so zum Beispiel die Berechnung der lokalen Top-Listen für

die Parallel Sets Ansicht. Da alle Anfragen für einen bestimmten Zeitraum durchgeführt werden, sind die Felder *timestamp* und *mod* obligatorisch. Die Bedeutung des Feldes *mod* wird zu einem späteren Zeitpunkt in Abschnitt 6.1.2 erläutert.

Der Overhead der durch die zusätzlichen Collections entsteht ist für den Hochschul-Datensatz in Tabelle 6.2 zusammengefasst. Insgesamt wurden 7 Dimension-Collections angelegt und je eine TopX-Collection pro Full-Collection.

Anzahl der Dokument			
Dimension-Collections:	287.736	1,5%	1,2%
Top10 Collections:	4.845.174	24,5%	19,5%
volle Daten Collections:	19.750.264	100,0%	79,4%
Summe:	24.883.175	126,0%	100,0%

Tabelle 6.2.: Zusätzlicher Anzahl an Dokumente für die TopX- und Dimension-Collections in der 4h-Granularität.

6.1.2. Indexstrukturen

Beim Einsatz von MongoDB ist die Verwaltung der Indexstrukturen besonders wichtig. Standardmäßig erzeugt MongoDB bei der Erstellung einer Collection nur eine Indexstruktur auf dem `_id`-Feld der Dokumente. MongoDB überlässt es dem Nutzer, neue passende Indexstrukturen anzulegen, die auf die späteren Anfragen an die Datenbank abgestimmt sind und somit die Anfragen beschleunigen. In Tabelle 6.3 wird eine Übersicht über die wichtigsten Anfragen und die daraus resultierenden Indexstrukturen gegeben. Eine besondere Bedeutung nimmt in diesem Zusammenhang das Feld *mod* – nach der erforderlichen Modulo-Operation benannt – ein, welches die Anzahl der Sekunden seit 00:00 UTC des zum Dokument gehörenden Zeitstempels beinhaltet. Das Feld dient vorrangig als Hilfsmittel, um Indexstrukturen darauf aufzubauen, die Anfragen mit Tageszeitraumbeschränkung beschleunigen (siehe Tabelle 6.3).

Anfrage	**passende Indexstruktur**
Dokumente in einem bestimmten Zeitraum	timestamp
+ Filter auf einzelne Tage (Zeitfilter)	timestamp
+ Filter auf einen Tageszeitraum (Zeitfilter)	mod → timestamp
+ Filter auf Werte eines Felds (Dimensionsfilter)	mod → timestamp → $field$
+ Filter auf Werte mehrerer Felder (Dimensionsfilter)	mod → timestamp → $field_x$
Berechnung der Top-Kategorien eines Feldes	$field$
+ Filter auf einzelne Tage (Zeitfilter)	timestamp → $field$
+ Filter auf einen Tageszeitraum (Zeitfilter)	mod → timestamp → $field$

Tabelle 6.3.: Übersicht über Anfragen mit den passenden Indexstrukturen.

6.2. Backend

Das Backend ist das Verbindungsglied zwischen Frontend und Datenbank. Hier wird auf die Daten aus der Datenbank zugegriffen, diese wenn nötig zwischengespeichert und in die vom Frontend benötigte Form gebracht. Die einzelnen Aufgaben, die das Backend dabei übernimmt, werden im Folgenden beschrieben:

Datenbank-Anfragen Das Backend ist zuständig für die Anfragen an die Datenbank. Die über die API vom Frontend angeforderten Daten werden – falls nicht bereits im Zwischenspeicher des Backends vorhanden – aus der Datenbank geholt.

Datenvorverarbeitung Nach dem Zugriff auf die Datenbank müssen die übermittelten Dokumente in die benötigte Form gebracht werden. Weitere Verarbeitungsschritte sind oft nötig, um aus den Dokumenten die benötigten Informationen zu gewinnen. So müssen IDs auf Bezeichnungen abgebildet, oder Dokumente in JavaScript-Objekte konvertiert werden. Diese Aufgaben werden bereits im Backend erledigt, um möglichst kompakte Daten an das Frontend übermitteln zu können.

Clusteranalyse Zur Vorverarbeitung der Daten gehört auch die Anreicherung mit zusätzlichen Informationen, wie die Clusterzugehörigkeit der Zeitreihen für Ebene 1. Diese wird bereits im Backend durchgeführt und nur die Ergebnisse werden als kompakte JavaScript-Objekte in einer für die Visualisierung mit D3.js angepassten Form an das Frontend übermittelt.

API zum Frontend Nach vorne wird eine API für das Frontend bereitgestellt, mit der die vorbereiteten und angereicherten Daten aus dem Backend angefragt werden können.

Datenzwischenspeicher Um die Antwortzeiten der API-Anfragen gering zu halten, werden bereits angefragte Daten nicht mit der Termination der API-Anfrage verworfen. Das Backend dient somit auch als Zwischenspeicher der Daten, um diese – wenn möglich – für weitere Anfragen wiederzuverwenden.

6.2.1. Sessions vs RESTful-API

Bei der Implementierung der API fiel die Wahl bewusst gegen die REST-API. REST-APIs bieten durch ihre Architekturprinzipien viele Vorteile wie Flexibilität, Skalierbarkeit oder Unabhängigkeit der Anfragen zueinander. Für die Umsetzung des Konzepts ist eine REST-API allerdings nicht geeignet, da die Datenbankanfragen, die nötig sind um die Anfragen der API zu beantworten, zu groß sind um für jede API-Anfrage die Daten von Grund auf neu anzufragen. Aus diesem Grund wird im Backend ein Status in Form einer Session gehalten, in die die angefragten Daten der aktuellen Ansicht im Frontend zwischengespeichert werden. Da die Daten für die Darstellung im Vergleich zu den Datenmengen in der Datenbank recht klein sind, benötigen die fertig aggregierten Daten verhältnismäßig wenig Platz und können so im Backend gespeichert werden. Darauf aufbauende Anfragen, wie das Nachladen neuer Kategorien in der Parallel Sets Ansicht, müssen nur die fehlende Differenz anfordern, andere Anfragen, wie das Löschen einer Achse, kommen so sogar ohne Anfragen an die Datenbank aus. Zudem können so individuelle Hilfsstrukturen, die für die Auflösung der IDs, die Datenverarbeitung und die Auflösung der IDs verwendet werden können, zu Beginn der Analyse einmal aufgebaut werden, die nur nach Bedarf verändert werden müssen.

6.2.2. Datenanfragen

Bei dem in Abschnitt 5.1 beschriebenen Workflow und Datenfluss des Konzepts handelt es sich um ein mentales Modell, welches bei der Interpretation der Daten unterstützen soll. Aus technischer Sicht werden die Daten der Ebenen jedoch separat verwaltet und nur der Eindruck eines Datenflusses vermittelt, indem entsprechende Operationen simultan bei allen Datenanfragen durchgeführt werden. Die Vorteile hierfür sind offensichtlich, wenn man bedenkt, dass andernfalls auf Ebene 1 mit den kompletten Daten gearbeitet werden müsste nur um diese für die Darstellung stark zu aggregieren. Stattdessen wird beispielsweise für diese Ebene immer auf die stark aggregierten Daten der *Timestamp-Collection* zugegriffen, wodurch Datenanfragen für Ebene 1 im Millisekundenbereich liegen und nicht zwischengespeichert werden müssen.

Für die zweite und dritte Ebene hingegen ist ein Zugriff auf die *TopX-Collection* oder die vollen Daten nötig. Damit die Daten möglichst schnell in der benötigten Form im Frontend vorliegen, werden die Daten wenn möglich zwischengespeichert. Das Ergebnis der Anfrage wird im Backend zwischengespeichert um bei ergänzenden Anfragen, wie dem Erweitern auf die Top 6 einer Dimension nicht alle Daten erneut anfragen zu müssen. In diesem Fall ist es beispielsweise möglich, nur die fehlenden Daten in der Datenbank anzufragen und die Ergebnisse mit den gespeicherten Daten im Backend zusammenzuführen. Dieser Fall tritt auch ein, wenn bereits am Anfang mit der Anfrage an die *TopX-Collection* nicht alle benötigten Daten geholt werden konnten. Der Aufbau der MongoDB-Anfragen ist für alle Ebenen gleich und unterscheidet sich im wesentlichen nur durch den Inhalt der Pipeline-Stages.

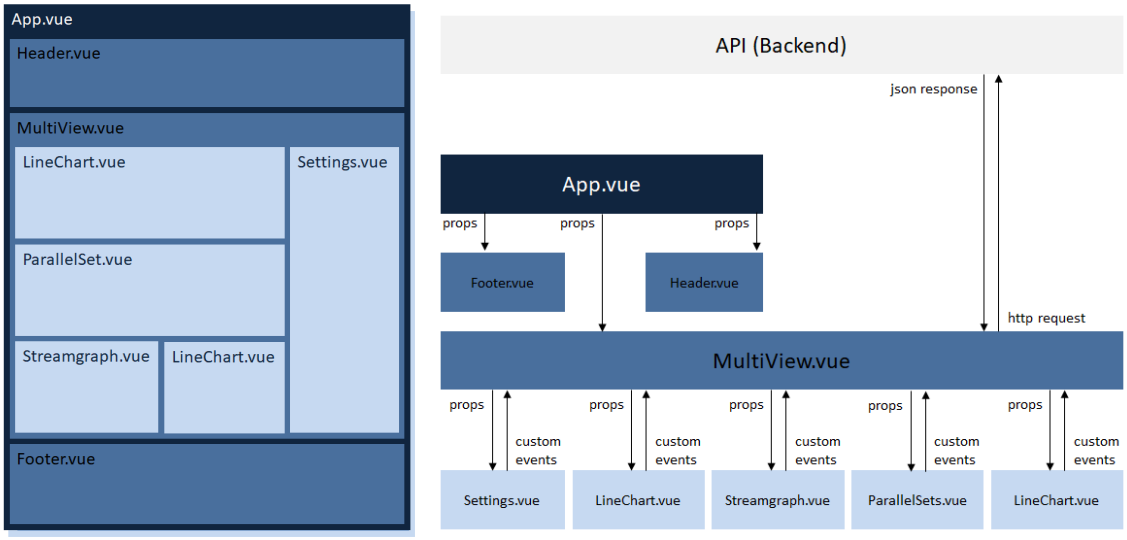
6.2.3. Clusteranalyse

Die Clusteranalyse, sowie die Berechnungen der Standardabweichungen, Mittelwert- und Medianzeitreihen wird bereits im Backend durchgeführt und nur die Ergebnisse in Form von JSON-Objekten werden an das Frontend übermittelt. Im Prototypen werden 2 Clusterverfahren unterstützt:

- k-Means
- hierarchisches Clustering

Für das Clustering nach k-Means und die statistischen Berechnungen wird die Java Machine Learning Library (Java-ML)[8] verwendet. Das hierarchische Clustering ist nach dem Vorbild von Van Wijk et al. [VV99] implementiert. Hierbei wurde auf die in JavaML definierten Datenobjektklassen (*Instance, Database*) und die breit aufgestellte JavaML-API zurückgegriffen. Das gewählte Distanzmaß entspricht der nicht normalisierten Euklidischen Distanz. Für das in Abschnitt 5.2 vorgestellte adaptive Clustering können die Verfahren ohne Modifikationen eingesetzt werden. Die Verwendung anderer Gewichtungsfunktionen, die nicht dem implementierten Box-Filter entsprechen, ist möglich, da beide Methoden als Eingabe gewichtete Instanzen unterstützen. Dabei gibt es zu beachten, dass es zu einer deutlichen Verlangsamung der Verfahren kommen kann, da nicht mehr mit gekürzten Zeitreihen gearbeitet werden kann.

[8] Java Machine Learning Library (Java-ML) http://java-ml.sourceforge.net/



(a) **(b)**

Abbildung 6.2.: Aufbau des Frontends. (a) Die Verschachtelung der Vue-Komponenten gibt an, welche Parent-Child-Beziehungen zwischen den Komponenten bestehen. Gleichzeitig wird dadurch auch die Verschachtelung der *HTML-Template-Tags* der Instanzen im DOM aufgezeigt. (b) Kommunikation und Data-Binding zwischen den Instanzen der Vue-Komponenten. Von den Parent-Komponenten werden Datenobjekte an die *props* (Properties) der Child-Komponenten gebunden. Müssen Child-Komponenten globale Änderungen an den Datenobjekten vornehmen, feuern diese Custom Events. Durch die Verwendung der *props* werden Änderungen an den Datenobjekten in MultiView.vue auf alle Child-Komponenten propagiert.

6.3. Frontend

Bei der Implementierung des Frontends wurde darauf geachtet, den Aufbau modular zu gestalten, sodass die angesprochene Erweiterbarkeit des Ansatzes mit anderen Komponenten einfach realisiert werden kann. Vue.js ist für diesen Zweck besonders gut geeignet, da Vue-Applikationen von Haus aus aus hierarchisch organisierten und wiederverwendbaren Vue-Instanzen bestehen. Diese bilden in sich geschlossene Einheiten, die aus einem Script-, Style-, und HTML-Template-Tag bestehen, und ermöglichen so eine klare Strukturierung und Abtrennung einzelner Komponenten. In Abbildung 6.2 (a) wird ein Überblick über die im Prototyp verwendeten Vue-Instanzen und deren Beziehungen gegeben.

In Abbildung 6.2 (b) wird die Kommunikation der Vue-Instanzen untereinander dargestellt. Alle Datenobjekte und globalen Einstellungen werden ausschließlich zentral über die Vue-Instanz MultiView verwaltet. Alle Komponenten, die Datenobjekte anzeigen oder diese verändern wollen, sind Child-Komponenten von MultiView und bekommen die benötigten Daten über die *props* (Vue-Properties) an die Komponente gebunden. Vue.js realisiert mit den *props One-Way-Data-Binding*, da Änderungen der Daten in MultiView an die Child-Komponenten propagiert werden, diese aber im Regelfall keine Änderungen an den Daten durchführen können. Müssen Datenänderungen – beispielsweise durch das Anfordern neuer Daten über das User Interface oder Änderungen an den

Listing 6.2 Rekursive Berechnung der *filter*-Property der Knoten im Baum. Die hier abgebildete Methode implementiert eine Tiefensuche im Baum. Die Methode *ninFilter* gibt zurück ob der Knoten gefiltert wird.

```
1   setFilterInNode: function(node, filterValue, filter){
2     var allChildrenFalse = true;
3     if(node.children == null)
4       return filterValue;
5     for(var child in node.children){
6       //calculate initial filter-property
7       node.children[child].filter = filterValue ? this.ninFilter(node, filter) : false;
8       //if filter-property still true, ...
9       if(this.setFilterInNode( node.children[child],  node.children[child].filter, filter))
10        allChildrenFalse = false;
11    }
12    if(allChildrenFalse)
13      node.filter = false;
14    return node.filter;
15  },
```

globalen Einstellungen – aus einer Child-Komponente heraus vorgenommen werden, erzeugt die Child-Komponente *Custom Events*, welche in MultiView aufgefangen und abgearbeitet werden. Die daraus entstehenden Änderungen an den Daten und Einstellungen werden über die *props* wiederum an alle Child-Komponenten propagiert. Watcher-Methoden auf die *props* in den Child-Komponenten passen daraufhin ihre Inhalte an. Auch das Highlighting und die Interaktionen zwischen den Komponenten werden über diesen Weg realisiert: In der auslösenden Child-Komponente wird ein entsprechendes Custom Event gefeuert, welches alle erforderlichen Informationen der Interaktion enthält. MultiView ändert auf Basis dieser Informationen die *props* aller Ziel-Komponenten, die daraufhin die Ansicht anpassen.

6.3.1. Parallel Sets

Die in diesem Prototyp verwendete Implementierung der Parallel Sets basiert auf einem Open Source Code von Davies[9], welcher als JavaScript-Modul in D3.js verwendet werden kann. Die nötigen Änderungen für diese Arbeit wurden direkt auf diesem Code durchgeführt. Davies bezieht sich bei seiner Implementierung auf das neue Design der Parallel Sets wie es von Kosara beschrieben wird [Kos10]. Kosara beschreibt darin auch das verwendete Datenmodell der Parallel Sets. Dieses besteht aus einer Baumstruktur, die aus einer Pivot-Tabelle der Daten berechnet werden kann. Abbildung 6.3 (a) zeigt die Zusammenhänge zwischen Kanten und Knoten des Baumes zu den Pfaden und Boxen der Parallel Sets an. Die Wurzel des Baumes repräsentiert dabei die kompletten Daten. Durch jede Achse der Parallel Sets wird die Datenmenge nach den dort vorhandenen Kategorien weiter unterteilt [Kos10]. Für die Integration des Dimensionsfilters wird direkt auf dieser Baumstruktur gearbeitet.

[9] An interactive parallel sets visualisation for D3.js. https://github.com/jasondavies/d3-parsets

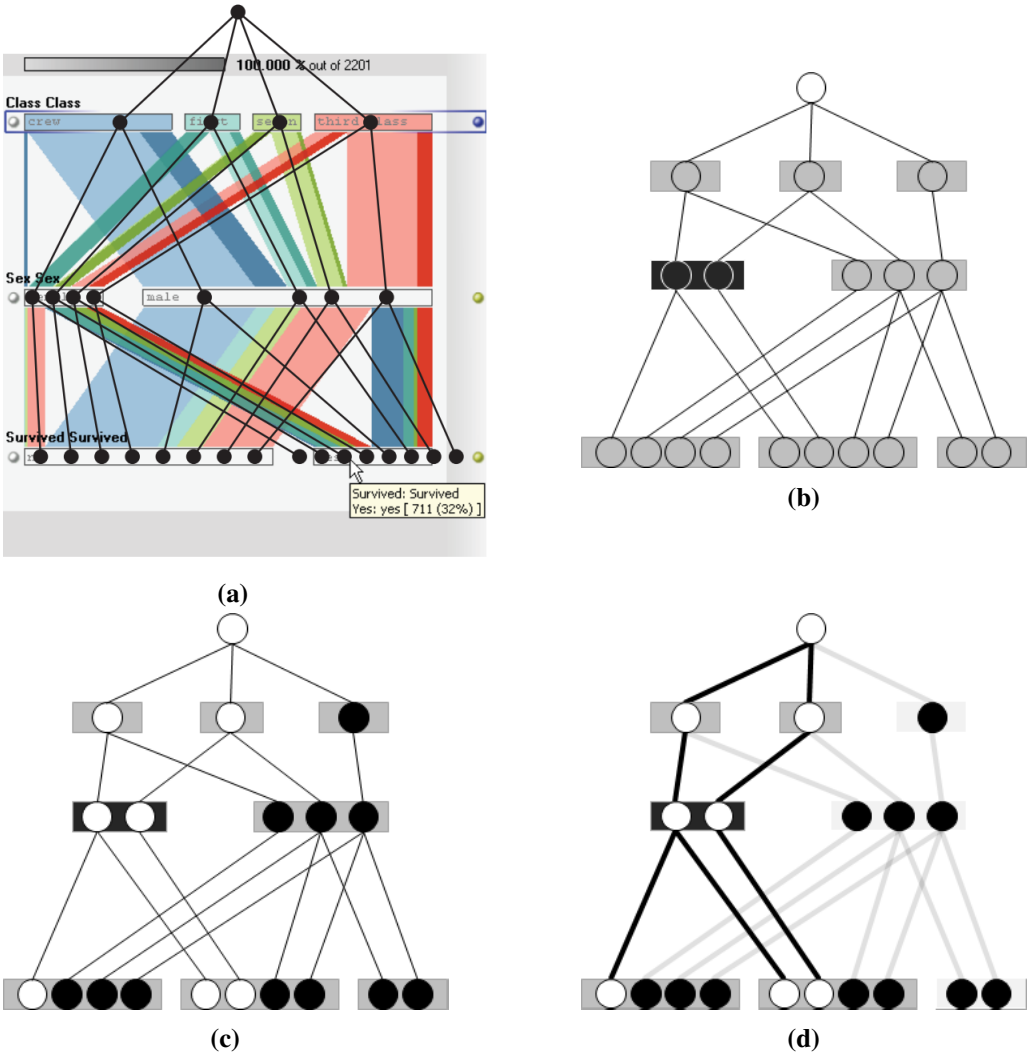


Abbildung 6.3.: Die zugrundeliegende Baumstruktur der Parallel Sets Ansicht. Abbildung (a) zeigt die Baumstruktur der Daten hinter der Parallel Sets Ansicht. Die Wurzel repräsentiert die komplette Datenmenge, die an jeder Verzweigung entsprechend den untergeordneten Kategorien in Teilmengen unterteilt wird. Bildquelle: [Kos10]. Die Abbildungen (b) - (d) zeigen das Vorgehen der Berechnung der Filter-Property der Knoten an: Beginnend bei der Wurzel werden die Knoten in der Reihenfolge der Tiefensuche traversiert und dabei abhängig von der Filtereinstellung (dunkelgrau) der dazugehörigen Kategorie (Box) mit einer vorläufigen *filter*-Property initialisiert. Ist $filter = false$ (schwarz), erhalten auch alle Kinder des Knotens diesen Wert. Ist $filter = true$ (weiß), werden die *filter*-Property für die Kinder unabhängig berechnet. Falls für alle Kinder $filter = false$ gilt, wird der initale Wert der *filter*-Property ebenfalls schwarz. Alle Boxen die keine weißen Knoten beinhalten verblassen. Die blassen Kanten und Boxen in Abbildung (d) geben an, dass die zugeordneten Pfade und Boxen der Parallel Sets Ansicht aufgrund des Dimensionsfilters optisch in den Hintergrund gestellt werden.

Für die Integration des Dimensionsfilters ist im wesentlichen nur ein zusätzliches JavaScript-Objekt nötig, welches für jede Dimension eine List der Kategorien speichert, die den Filter eindeutig definieren. Damit können anschließend die in Kapitel 5 vorgestellte visuelle Repräsentation des Filters in Form der Highlights auf den Boxen und Pfaden, sowie das optische Verblassen der anderen Daten modelliert werden. Um herauszufinden welche Pfade und Boxen aufgrund des Filters gehighlightet werden müssen, muss die im vorherigen Absatz beschriebene Baumstruktur von der Wurzel aus traversiert werden. Jeder Knoten bekommt dabei eine Boolesche Variable namens *filter* zugewiesen die angibt, ob die durch den Knoten repräsentierte Teilmenge aufgrund des Filters gehighlightet werden muss. Das Verfahren dazu wird in Abbildung 6.3 beschrieben und die konkrete Methode, die dieses Verfahren umsetzt, ist in Listing 6.2 zu sehen. Das vorgestellte Verfahren kann somit als Verallgemeinerung des Highlightings eines Knotens mit den dazugehörigen Pfaden im ursprünglichen Design der Parallel Sets angesehen werden.

7. Evaluation

Die Evaluation dient der Untersuchung und Bewertung des erarbeiteten Konzepts. Für diesen Zweck werden in diesem Kapitel zwei typische Anwendungsfälle zusammen mit deren Ablauf vorgestellt, um die Anwendbarkeit des Ansatzes auf reale Daten zu demonstrieren. Um fachspezifische Vor- und Nachteile des visuellen Ansatzes diskutieren zu können, wird – wie auch bereits in Kapitel 4 – ein Experteninterview durchgeführt. Die Zusammenfassung der Ergebnisse dieses Kapitels folgt zusammen mit allen anderen Erkenntnissen der Arbeit in Kapitel 8.

7.1. Anwendungsfälle

Aus der Anforderungsanalyse sind einige Anwendungsfälle hervorgegangen, die vom visuellen Ansatz adressiert werden sollen. In diesem Abschnitt werden zwei gegensätzliche Anwendungsfälle vorgestellt, die unterschiedliche übergeordnete Analyseziele verfolgen. Beim erste Anwendungsfall, *Peak-Detection*, steht die Exploration der Daten im Vordergrund. Hier startet die Analyse ohne konkrete Fragestellungen oder einschränkende Vorbedingungen und der Ausgang der Analyse wird durch die in den Daten sichtbaren Strukturen bestimmt. Beim zweiten Anwendungsfall, *Troubleshooting*, ist das Analyseziel hingegen klar definiert: Es existiert ein Problem im Netzwerk und dieses muss gefunden werden. Bestimmte Rahmenbedingungen, wie der Zeitpunkt an dem das Problem aufgetreten ist, sind bereits bekannt und die Analyse verfolgt das klare Ziel den Fehler zu lokalisieren, den Verursacher zu finden und weitere Informationen aus den Netzwerkdaten zu gewinnen, die für die Fehlerbehebung von Interesse sind. Für die Demonstrationen wurde der Datensatz einer Hochschule verwendet, der von IsarNet zur Verfügung gestellt wurde. Aus datenschutzrechtlichen Gründen werden die IP-Adressen der Screenshots im Folgenden zensiert.

7.1.1. Peak-Detection

Die Suche nach auffälligen Peaks oder Dips im Gesamtverkehrsvolumen ist einer der Anwendungsfälle, die aus der Anforderungsanalyse hervorgegangen sind. Das Analyseziel der Anwender/-innen ist, solche Muster in den Daten zu erkennen und herauszufinden, welche Ursachen diese haben. Für die Anwender/-innen ist dabei von besonderem Interesse, wer (IP-Adressbereich, IPView) und was (Protokoll, Anwendung) diesen Verkehr verursacht hat und ob es andere Tage gibt, die die gleichen Auffälligkeiten zeigen.

Die Analyse beginnt mit einer Übersicht der Daten in Form der geclusterten Zeitreihen (siehe Abbildung 7.1 oben). Wird beispielsweise ein Peak entdeckt (Abbildung 7.1 A) kann dieser mit Hilfe der Tageszeitraumbeschränkung selektiert werden. Da alle Cluster diesen Peak aufweisen, kann angenommen werden, dass auch alle Tage diesen Peak vorweisen. Diese Annahme kann überprüft werden indem entweder die Tageskurven angezeigt werden – wodurch Ausreißer visuell sichtbar

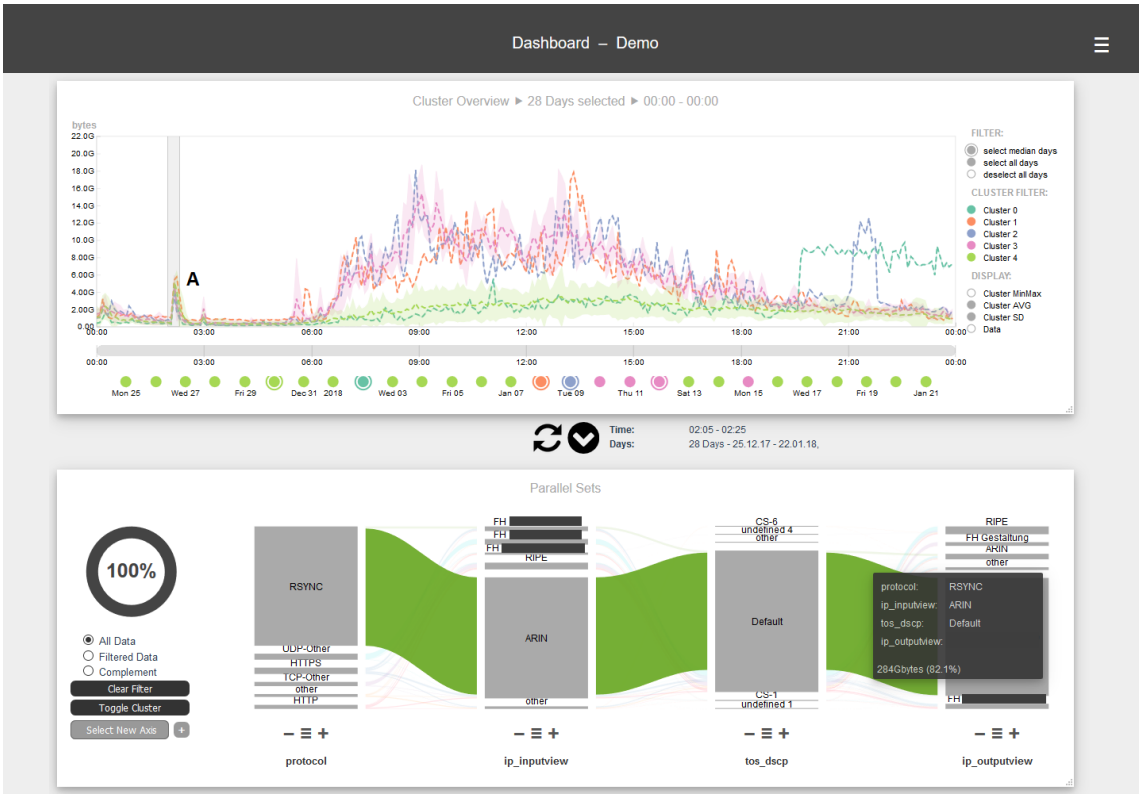


Abbildung 7.1.: Screenshot des Peak-Detection-Anwendungsfalls. Für den ausgewählten Peak in der ersten Ebene kann man in der zweiten Ebene bereits deutlich eine Häufung in der Parallel Sets Ansicht erkennen. Da das dazugehörige Protokoll *RSYNC* ist, handelt es sich dabei vermutlich um eine planmäßige Datensynchronisation der Hochschule.

werden – oder indem das adaptive Clusterverfahren eingesetzt wird, um die Tage erneut mit anderen Einstellungen zu clustern. In der zweiten Ebene kann der Peak anschließend untersucht werden. Es ist deutlich zu erkennen, dass 82% des Netzwerkverkehrs in dieser Zeitspanne gleichartig ist. Da der Peak bei allen Tagen vorhanden und das dazugehörige Protokoll *RSYNC* ist, ergibt sich hier die Vermutung, dass es sich dabei um eine planmäßige Datensynchronisation der Hochschule handelt. Die Ursache des Peaks ist somit erkannt und das Analyseziel erreicht.

7.1.2. Toubleshooting

Eine weitere Gruppe von Anwendungsfällen sind *Troubleshooting*-Szenarien. Diese haben gemein, dass netzbasierte Anwendungen in einer bestimmten Zeitspanne spürbar langsam werden oder die Qualität der Echtzeitdienste nachlässt. Dies hat zur Folge, dass Mitarbeiter oder Kunden beim Support eine Störung der Anwendung melden. Zum einen kann besonders viel Verkehr zu einer starken Auslastung der Leitung führen und so die Störung verursachen. Zum anderen können aber auch entsprechende Pakete nicht korrekt klassifiziert sein und dadurch nicht mit der nötigen Priorität behandelt werden. Im Folgenden wird der Ablauf eines Anwendungsfalls demonstriert, bei dem ein Voice-over-IP Telefonat schlechte Qualität ausweist. Das Analyseziel in diesem Anwendungsfall

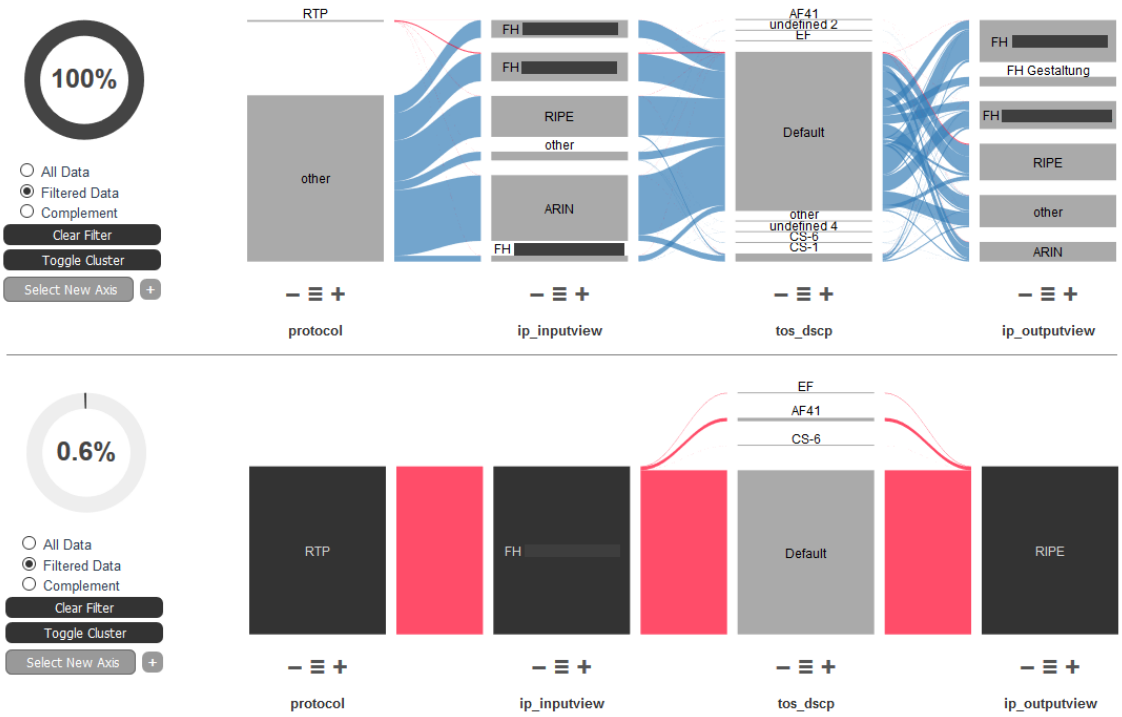


Abbildung 7.2.: Screenshot des Troubleshooting-Anwendungsfalls. Durch den Dimensionsfilter auf die 3 Kategorien wird der relevante Netzwerkverkehr in den Fokus gestellt und die benötigten Informationen können aus der Darstellung entnommen werden.

ist die Überprüfung der Klassifizierung der Netzwerkpakete. Für die Demonstration eines echten Troubleshooting-Anwendungsfalls muss ein Zeitpunkt in den Daten bekannt sein, bei dem dieses Problem im Netz aufgetreten ist. Diese Informationen liegen für den Datensatz allerdings nicht vor. Aus diesem Grund wird im folgenden nur der Ablauf der Analyse mit Hilfe des Prototypen skizziert.

Die Analyse startet wieder mit der Übersicht über die Daten in Form der geclusterten Zeitreihen. Da der Zeitraum des Problems bereits bekannt ist, wird dieser direkt eingestellt und ohne die Verwendung der Clusteranalyse mit der zweiten Ebene fortgefahren. Dort können alle bekannten Informationen des Netzwerkverkehrs als Dimensionsfilter eingestellt werden: IP-Adresse des Empfängers, RTP-Protokoll und IP-Adresse des Senders. In der Dimension *tos_dscp* kann anschließend die Klassifikation der Netzwerkpakete angezeigt werden. In Abbildung 7.2 ist deutlich zu sehen, dass der meiste Verkehr mit *Default* klassifiziert ist. Dieser wird nicht bevorzugt behandelt, was eine schlechte Qualität der Echtzeitanwendung nach sich ziehen kann. Werden die Interfaces zusätzlich als Dimensionen in die Parallel Sets Ansicht integriert, kann über diesen Weg herausgefunden werden, welche Ports konfiguriert werden müssen um das Problem zu beheben.

Zudem könnte anschließend überprüft werden, ob eine allgemein hohe Auslastung der Leitung für die schlechte Qualität des Telefonats auslösend war. Dazu kann zunächst in der ersten Ebene überprüft werden ob ein hoher Gesamtverkehr im Netz zu diesem Zeitpunkt vorliegt. Ist dies nicht der Fall, könnten anschließend die Leitungen überprüft werden. Dazu muss zunächst in der Parallel Sets Ansicht erneut ein Dimensionsfilter mit allen bekannten Informationen eingestellt werden.

Anschließend kann in der Dimension *Link* abgelesen werden, durch welche Leitungen im Netz dieser Verkehr fließt. Für diese Leitungen kann dann einzeln mit Hilfe der Streamgraph Ansicht in der dritten Ebene überprüft werden, wie hoch die Auslastung der Leitung in diesem Zeitraum ist. Die Referenzbandbreite der Leitungen muss den Anwender/-innen separat vorliegen. Diese Überprüfung ist im jetzigen Prototypen nicht möglich, da die Dimension *Link* nicht definiert ist. Das Hinzufügen beliebiger Dimensionen wird allerdings vom Konzept unterstützt.

7.2. Experten-Feedback

Bereits während der Entwicklungsphase wurde in enger Zusammenarbeit mit den betreuenden Mitarbeitern von IsarNet in regelmäßigen Abständen Feedback zum aktuellen Stand der Konzepte und des Prototypen eingeholt. Zudem wurden Teile der Arbeit auf internen Firmenkonferenzen und einem Meeting des Verbundprojekts AutoMon (Automatisiertes Performance-Monitoring) [Aut18] vorgestellt um Feedback von weiteren Experten einzuholen. Die daraus gewonnenen Erkenntnisse sind direkt in die Entwicklung der Konzepte eingeflossen.

Wie in der Anforderungsanalyse in Kapitel 4 werden wieder Interviews im selben Personenkreis geführt um abschließend Feedback für die gewählte Lösung einzuholen. Anhand der Ergebnisse können dann fachspezifische Vor- und Nachteile des Ansatzes diskutiert werden. Von Seiten der Deutschen Bahn ist ein Teilnehmer hinzugestoßen, der sich aufgrund der Vorstellung des Konzepts im Automon Projekt für den Fortgang der Arbeit interessierte.

7.2.1. Vorgehensweise der Interviews

Die Organisation der Experteninterviews ist identisch mit der der Interviews aus Kapitel 4. Da alle Personen sich bereits kannten, begann das Interview statt der Vorstellungsrunde mit einer kleinen Einführung des Prototypen, bei der der Workflow erklärt und die verschiedenen Funktionen der Benutzeroberfläche kurz vorgestellt wurden. Anschließend wurden die Teilnehmer darum gebeten, sich mit dem Prototypen auseinanderzusetzen. Als Hilfe gab es dafür konkrete Aufgabenstellungen, die es im Rahmen der Interviews zu lösen galt. Die Aufgaben dienten dazu, typische kleine Anwendungsfälle des Prototyps nachzustellen. Da beispielsweise für den Troubleshooting Anwendungsfall keine Informationen über tatsächliche Probleme im Netzwerk zu den Daten verfügbar waren, waren die Aufgaben abstrakter Natur und stellten einen ähnlichen Ablauf dar. Im einzelnen lauten die Aufgaben wie folgt:

- Peak-Detection auf allen Ebenen
- Peak-Detection mit Filter
- Häufungen im mehrdimensionalen Raum

Eine genau Beschreibung der Aufgaben befindet sich in Anhang B.

Der Prototyp wurde dabei zum Teil über eine Remote-Verbindung von IsarNet direkt an den PCs der Teilnehmer bereitgestellt. Leider konnte diese Verbindung aus technischen Gründen nicht bei allen Teilnehmern bereitgestellt werden. Als Alternative wurde eine Bildschirmübertragung mit ScreenControl gewählt. Der Fragebogen des abschließenden Interviews befindet sich ebenfalls im Anhang B.

Da der Prototyp zum Zeitpunkt der Interviews noch nicht in der in Kapitel 5 vorgestellten finalen Version war, wurden die Interviews auf einer Zwischenversion durchgeführt. Diese enthielt bereits alle wichtigen Komponenten und unterscheidet sich im wesentlichen durch zum Teil provisorische UI-Elemente. Die Version des Prototypen wurde über die Interviews hinweg konstant gehalten, um eine Vergleichbarkeit der gegebenen Antworten zu gewährleisten. In Anhang C ist ein Screenshot des Prototypen in dieser Version zu sehen.

7.2.2. Ergebnisse der Interviews

Die Ergebnisse der Interviews werden – wie in Kapitel 4 – in einzelne Themenblöcke zusammengefasst. Das Fazit der Ergebnisse bildet den Abschluss des Abschnitts.

Bewältigung der Aufgabenstellungen

Bei der Bewältigung der Aufgabenstellungen gab es vor allem Schwierigkeiten bei der Bedienung, was bei der Einführung neuer Visualisierungs- und Interaktionskonzepte in gewissem Umfang zu erwarten war. Auch die Aufgabenstellungen waren teilweise nicht klar, was zu Problemen bei der Ausführung der einzelnen Schritte geführt hat. Davon abgesehen konnten aber alle Teilnehmer alle Aufgaben lösen. Im einzelnen gab es bei der ersten Aufgabe am wenigsten Probleme. Die Bewertung dieser Aufgabe fiel von den Teilnehmern im Anschluss entsprechend positiv aus.

Bei der zweiten Aufgabe gab es bereits mehr Schwierigkeiten. Eine Steigerung der Komplexität der Aufgabenstellung, sowie die Tatsache, dass alle Teilnehmer die Aufgabenstellung zunächst falsch verstanden hatten, können als Gründe genannt werden. Nachdem die Teilnehmer jedoch wussten, was gefragt war, konnten alle die nötigen Schritte absolvieren, um das richtige Ergebnis in der zweiten Ebene zu erhalten. Die nötigen Filtereinstellungen auf dieser Ebene wurden von allen Teilnehmern richtig gewählt. Bewertet wurde diese Aufgabe als mittel gut bis gut lösbar. Die Teilnehmer die die Lösbarkeit mit mittelmäßig bewertet haben, gaben allerdings auch an, dass die Darstellung die nötigen Informationen enthalte, die Bedienung und der Weg dahin allerdings mühsam sei. Ein Teilnehmer nannte die Bedienung dabei "gewöhnungsbedürftig".

Bei der letzten Aufgabe gab es zwei sehr unterschiedliche Bewertungen. Zwei Teilnehmer konnten die Aufgabe gut bewältigen und bewerteten die Aufgabe entsprechend. Bei einem Teilnehmer kam es zu größeren Schwierigkeiten. Hier kann wieder die Aufgabenstellung als ein Grund genannt werden, da deutlich zu erkennen war, dass der Teilnehmer nicht genau wusste was bei der Aufgabe verlangt wurde. Zudem stellte sich heraus, dass der Teilnehmer eine andere Erwartungshaltung hatte, welche Auswirkungen die Parallel Sets Interaktionen auf die Streamgraph Ansicht haben. Dieses Problem wird im nächsten Abschnitt noch einmal genauer beschrieben. Auch von Seiten des Teilnehmers wurden diese Beobachtungen im Interview bestätigt. Zudem gab er allerdings auch an, dass man die nötigen Informationen aus der Darstellung erkennen könne, wenn man sich auskennen würde.

Insgesamt gab es Probleme vor allem dann, wenn die Teilnehmer die einzelnen Schritte der Aufgabenstellung aus dem Blick verloren. Der Hinweis auf die Aufgabenstellung zu achten, hat in den meisten Fällen das aktuelle Problem der Teilnehmer gelöst.

Verständnis des Konzepts

Die meisten Teilnehmer gaben an, den Workflow, die Ebenen und den Datenfluss mit den Filtern prinzipiell verstanden zu haben. Ein Teilnehmer kam jedoch zu einer anderen Einschätzung. Die Gliederung der Ebenen und des Workflows sei für ihn nicht zu erkennen gewesen, da alle Informationen bereits zu Beginn der Analyse angezeigt wurden. Die Analyse gebe den Weg dabei nicht vor. Er merkte allerdings an, dass wenn zuerst alle Komponenten ”ausgegraut” seien und erst nach dem Auswählen der Filter die Darstellungen angezeigt würden, das Konzept für ihn nachvollziehbar sei. Von allen Teilnehmern wurde dabei angemerkt, dass das explizite Anwenden des Filters in der ersten Ebene und das implizite Anwenden der Filter in der zweiten Ebene ein Bruch in der Bedienung sei und zu Verständnisproblemen führe. Auch das Umschalten auf gefilterte Daten in der Parallel Sets Ansicht war für die Teilnehmer nicht offensichtlich und wurde in den Interviews angesprochen.

Aus den Beobachtungen der Teilnehmer bei der Bewältigung der Aufgaben und den Antworten der Interviews kann das Verständnis der einzelnen Ansichten abgeleitet werden. Die erste Ebene wurde von den Teilnehmern prinzipiell gut verstanden, was damit zusammenhängen könnte, dass IsarFlow eine ähnliche Darstellung verwendet und Clusteranalysen allen Teilnehmern bekannt waren. Die Auswahl der Tage zählte hier zu den größten Schwierigkeiten.

Bei der Parallel Sets Ansicht gab es überwiegend positive Rückmeldungen. Alle Teilnehmer haben verstanden, wie die Darstellung zu interpretieren ist, konnten sich mit dem Dimensionsfilter zurechtfinden und die Dicke der Pfade korrekt mit Häufungen im Netzwerkverkehr in Verbindung bringen. Die Darstellung wurde entsprechend von allen gut bewertet. Zwar sei die Ansicht nicht selbsterklärend, allerdings verstehe man es, wenn man es sich mal angeschaut habe.

Größere Schwierigkeiten gab es bei den Detailansichten. Zur Streamgraph Ansicht kam explizit eine Anmerkung eines Teilnehmers, dass seine Erwartung der Auswirkungen der Interaktionen der Parallel Sets Ansicht mit der Implementierung nicht übereinstimme. Da die Farben der beiden Ansichten unterschiedlichen Bedeutungen zugeordnet sein können, führe dies zu Verwirrungen. Der Teilnehmer hätte erwartet, dass die Bedeutungen der Farben übereinstimmen und der Streamgraph – beim Fokussieren einer Kategorie in der Parallel Sets Ansicht – nach den Kategorien und Farben aufgeschlüsselt wird, die an dieser Box ankommen. Die Farbgestaltung sei entsprechend problematisch ("hätte ich nicht erwartet"). Die Filterzusammenfassung auf der rechten Seite wurde von allen Teilnehmern als wenig hilfreich eingestuft. Ein Teilnehmer merkte dabei an, dass es ausreiche, wenn die Zusammenfassung durch eine textuelle Beschreibung des Filters noch einmal aufgegriffen werde. Da dies weniger Platz beanspruche, sei mehr Platz für die linke Seite, die wichtiger sei (Streamgraph Ansicht).

Verbesserungsvorschläge und Anmerkungen

Während den Interviews wurden immer wieder Verbesserungsvorschläge und Anmerkungen zu bestimmten Funktionen des Prototypen eingebracht. Einige dieser sind bereits oben genannt worden. Alle weiteren werden hier zusammengefasst. Aufgrund der vielen Probleme bei der Handhabung des Prototypen kamen von den Teilnehmern viele Verbesserungsvorschläge zur Gestaltung der Bedienung. Zu den am häufigsten genannten Vorschlägen zählten eine deutliche Unterscheidung

zwischen impliziten und expliziten Filteranwendungen, größere visuelle Verdeutlichung wirkender Filter auf den Ebenen, bessere Beschriftungen für die Komponenten und eine Selektion des Dimensionsfilters ohne Kontextmenü.

Besonders positiv hervorgehoben wurde die Verbindung zwischen der ersten und zweiten Ebene. Optisch könne man in der ersten Ebene erkennen, wenn etwas auffällig ist, und dieses dann in der zweiten Ebene sehen. Ein Teilnehmer merkte an, man behalte im Blick was man ausgewählt habe und könne sich die Session in der Parallel Sets Ansicht aufbauen, die man brauche.

Speziell auf den Troubleshooting-Anwendungsfall bezogen wurde der Wunsch deutlich, die Analyse mit einer Einschränkung des Netzbereichs im speziellen auf eine einzelne IP-Adresse zu beginnen. Speziell beim Troubleshooting komme es häufig vor, dass Störungen von Kunden mit der Angabe eines Endgeräts kämen und man die Analyse damit beginnen wolle. Der Workflow und die Interaktionen zwischen den Ebenen sei ein klarer Vorteil der Anwendung, sagten zwei Teilnehmer.

Gesamteindruck

Abgesehen von den Schwierigkeiten bei den Aufgabenstellungen und der Bedienung, die als mühsam beschrieben wurde, fiel die Bewertung des Prototypen überwiegend positiv aus. Alle Teilnehmer könnten sich vorstellen das System auch häufiger zu benutzen und aus Sicht des Kundensupports "würde es den Kunden ans Herz legen", da man damit sehr gut Häufungen in den Daten erkennen könne. Zwar gaben die Teilnehmer mehrheitlich an, dass das System als komplex einzustufen sei, allerdings findet kein Teilnehmer das Konzept unnötig komplex, sondern für die Anforderungen angemessen ("Man kann auch komplexere Fragestellungen damit beantworten"). Die Einschätzungen der Teilnehmer was die Erlernbarkeit des Konzepts angeht fällt dabei positiv aus. Es müssten zwar noch Änderungen in der Bedienbarkeit umgesetzt werden, aber speziell für Experten sei das Konzept prinzipiell schnell erlernbar. Eine Einarbeitungszeit sei allerdings nötig.

Die Darstellung betreffend, wurde das System mit den unterschiedlichen Ansichten nicht als unübersichtlich oder überladen empfunden. Auch die Interaktionen seien nicht überladen, sondern hilfreich für die Bewältigung der Aufgaben. Lediglich bei der Parallel Sets Ansicht wurde angemerkt, dass die Übersichtlichkeit der Darstellung von den konkreten Einstellungen der Ansicht abhänge. Es liege allerdings in der Hand der Anwender/-innen, wie viele Informationen diese durch Erhöhen der Anzahl der Kategorien zusätzlich in die Darstellung integrieren.

Fazit

Das Fazit der Experteninterviews fällt insgesamt positiv aus. Das Visualisierungskonzept wurde von allen Teilnehmern prinzipiell verstanden und alle waren in der Lage damit die Aufgabenstellungen zu lösen. Besonders die Parallel Sets Ansicht – das Herzstück der Anwendung – und die Interaktionen zwischen den Ebenen wurden von den Teilnehmern positiv hervorgehoben. Schwierigkeiten gab es vor allem bei der Bedienung der Anwendung. Zum Teil können diese Schwierigkeiten auf die Remote-Verbindung zur Anwendung über IsarNet zurückgeführt werden, da die Teilnehmer mit den größten Schwierigkeiten den Prototypen nur über die Remote-Verbindung bedienen konnten. Dies trifft allerdings nicht auf alle Bedienungsschwierigkeiten zu. In den Experteninterviews wurde deutlich, dass einige Interaktionskonzepte überarbeitet werden müssen, um das Konzept intuitiver

zu gestalten und die Komplexität zu verringern. Nichtsdestotrotz war bei den Interviews auch zu erkennen, dass das Konzept für die Aufgabenstellungen und die Anwendungsfälle – für die es konzipiert wurde – geeignet ist und auch von den Teilnehmern so wahrgenommen wurde.

7.2.3. Änderungen basierend auf den Ergebnissen der Interviews

Nach den Experteninterviews wurden einige Kleinigkeiten am Prototypen verändert, die nicht mehr vor den Interviewterminen fertig gestellt werden konnten oder erst durch die Interviews in den Prototypen integriert wurden. Zu diesen Änderungen gehören vor allem optische Verbesserungen der UI-Elemente. So wurde die Positionierung der Pfeile verändert, die Filterinformationen überarbeitet und die provisorischen Überschriften der Komponenten durch angemessene Bezeichnungen mit Breadcrumb-Navigation ersetzt.

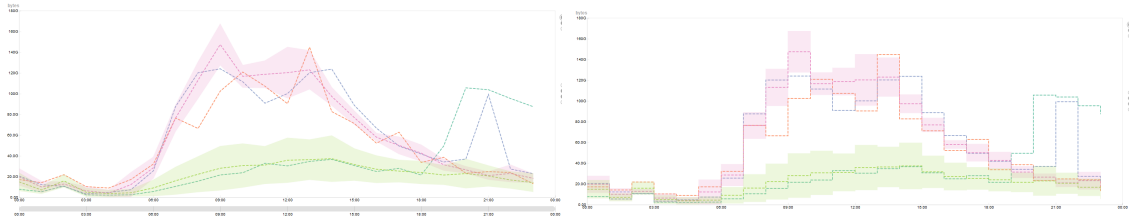
Alle Teilnehmer kritisierten, dass Dies wurde verbessert, indem ein neues Symbol daneben eingeführt wurde, welches die Funktion der Filteranwendung übernimmt. Um einen Punkt aller Teilnehmer zu adressieren wird in der neuen Version zudem der Pfeil zwischen Ebene 1 und 2 nicht zum Anwenden des Filters benutzt, sondern ein neues Symbol daneben übernimmt diese Funktion. Dadurch ist der Unterschied zwischen dem expliziten Anwenden des Filters in der ersten Ebene und dem impliziten Anwenden des Filters in der zweiten Ebene auch optisch zu sehen. Weitere kleine Änderungen bei der Funktionalität sind das Hinzufügen von *select-all*- und *deselect-all*-Optionen um die Auswahl der Tage zu erleichtern, einklappbare *Globale Einstellungen*, um sich besser auf die Daten fokusieren zu können und Änderungen im Backend, die nicht vollständige oder fehlende Funktionen des Konzepts ergänzen. Im Anhang C befindet sich eine Gegenüberstellung der beiden Versionen, bei der alle optischen Änderungen eingekreist sind.

8. Diskussion

Nachdem in der Evaluation typische Anwendungsfälle vorgestellt und Experten-Feedback zu den Ergebnissen der Arbeit eingeholt wurden, werden in diesem Kapitel die Stärken, Schwächen sowie die Grenzen des Konzepts herausgearbeitet. Die Skalierbarkeit des visuellen Ansatzes, mit seinen Stärken und Schwächen, wird dabei separat in einem Abschnitt der Diskussion behandelt.

8.1. Stärken und Schwächen

Während der Konzeption haben sich einige visuelle Schwächen des Konzepts herausgestellt, für die keine oder nur teilweise Lösungen gefunden wurden. Eines dieser Probleme bezieht sich auf die verwendete Interpolation zur Darstellung der aggregierten Zeitreihen in der ersten und dritten Ebene. Im Gegensatz zu Van Wijk et al. [VV99], bei dem sich die Zeitreihen aus Messpunkten zu bestimmten Zeitpunkten zusammensetzen, repräsentieren die Zeitstempel der Zeitreihen in dieser Arbeit einen zeitlichen Intervall. Die lineare Interpolation dieser Werte (Abbildung 8.1 (a)) kann – wie sich in den Experteninterviews gezeigt hat – zu Fehlinterpretationen der Daten führen. Andere Interpolationen, zu sehen in Abbildung 8.1 (b), lassen weniger Fehlinterpretationen zu, bilden den Verlauf der Kurven allerdings schlechter ab. Für die Darstellung der Zeitreihen in der ersten Ebene wurde aus diesem Grund die Variante (a) verwendet, obwohl diese den Sachverhalt der Aggregation nicht optimal wiedergibt. Eine weitere visuelle Schwäche des Systems ist die Darstellung der Streamgraph Ansicht bei der Auswahl einer starken Tageszeitbeschränkung in der ersten Ebene. Im Extremfall machen die selektierten Bereiche als schmale Streifen im Streamgraph nur einen kleinen Teil der Ansicht aus und ein Großteil der Ansicht nimmt das Komplement des Zeitfilters ein (siehe Abbildung 8.2). Dies stellt vor allem ein Problem für die visuelle Skalierbarkeit des Ansatzes dar, da der Effekt bei der Analyse größerer Zeiträume verstärkt wird (siehe Abschnitt 8.2). Ein weiteres sehr allgemeines Problem der Darstellung ist die mehrfache Verwendung der gleichen



(a) Lineare Interpolation

(b) Step After

Abbildung 8.1.: Vergleich verschiedener Interpolationen in der ersten Ebene. Obwohl Abbildung (b) den tatsächlichen korrekt ist, wird in Abbildung (a) der Verlauf der Werte deutlich. Abbildung (b) hingegen wirkt trotz der wenigen eingezeichneten Linien bereits unübersichtlich.

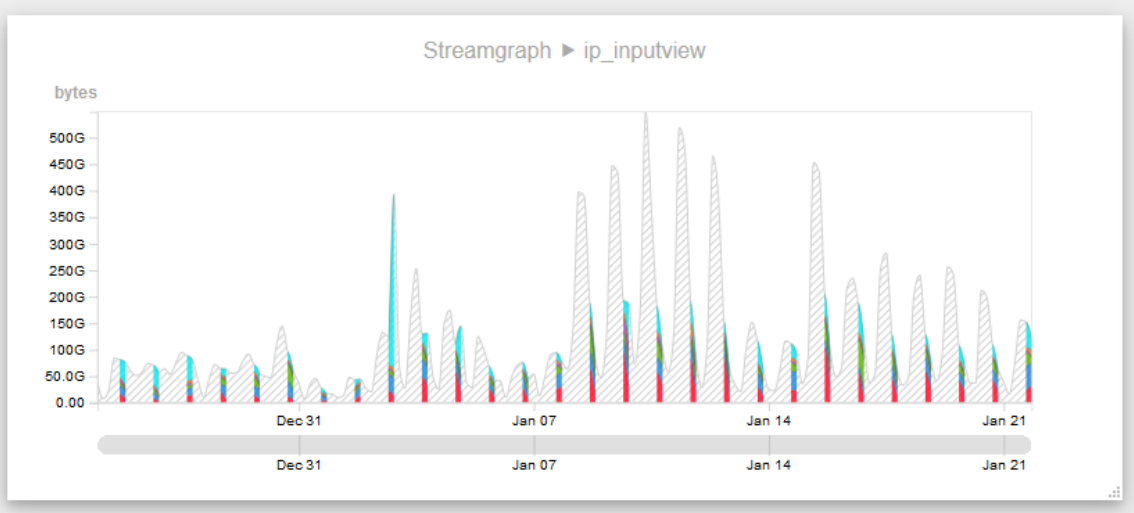


Abbildung 8.2.: Streamgraph Ansicht mit starker Tageszeiteinschränkung.

Farben in einem unterschiedlichen Kontext. So kann abhängig davon, welche Achse in der Parallel Sets Ansicht vorne steht, das Color Mapping im Vergleich zum Streamgraph oder der ersten Ebene identisch oder unterschiedlich sein. Dies führt zu Missverständnissen, da vom Kontext abhängt, ob die Farben den gleichen Sachverhalt darstellen. Dieser Konflikt kann allerdings auch nicht einfach aufgelöst werden, da mögliche strikte Kopplungen des Farbschemas der zweiten und dritten Ebene die Interaktionsmöglichkeiten einschränken würden.

Abgesehen von den oben genannten Schwächen des Konzepts liegen die Stärken des visuellen Ansatzes im Zusammenspiel der Komponenten untereinander. Auch der modulare Aufbau bietet den Vorteil, das System auf unterschiedliche Bedürfnisse anpassen zu können, indem nur die Komponenten ausgetauscht werden, aber nicht das Konzept des Workflows. Im Vergleich zu IsarFlow bekommen die Anwender/-innen durch die mehrstufige interaktive Filterauswahl eine sofortige visuelle Rückmeldung der Filterauswirkungen, was das Einstellen leerer Ergebnismengen verhindert. Zudem kann durch die enge Kopplung der Parallel Sets Ansicht und der Streamgraph Ansicht ein schnelles hin- und herspringen zwischen den Dimensionen ermöglicht werden, welches durch den Dimensionsfilter zusätzliche Möglichkeiten der Fokusierung birgt. Letztlich ist die Skalierbarkeit des Ansatzes vielversprechend und bietet damit eine gute Grundlage für eine mögliche Weiterentwicklung des Prototypen zu einer produktionsfähigen Anwendung (siehe Abschnitt 8.2).

Die konzeptionellen Anforderungen kann das Konzept zum größten Teil erfüllen, allerdings gibt es auch hier Punkte, die besser adressiert werden könnten. Der Fokus könnte stärker auf der Zeit liegen, da im jetzigen Konzept durch die Möglichkeit der Aggregation über große Zeiträume der Nutzen der Darstellung sinkt. Zudem hat sich in den Experteninterviews in Abschnitt 7.2.2 gezeigt, dass der Startpunkt der Analyse auf dem gesamten Netzwerkverkehr nicht immer sinnvoll ist. Besonders im *Troubleshooting*-Anwendungsfall ist neben dem Zeitraum auch eine Einschränkung auf einen Netzbereich, eine einzelne Leitung oder eine ganz bestimmte IP-Adresse eines Kunden nötig. Diese Einschränkung kann in diesem Konzept allerdings frühestens in der zweiten Ebene vorgenommen werden, und somit nicht in der Clusteranalyse berücksichtigt werden. Genau das ist aber aus Anwendersicht interessant. Besonders gut – und das wurde ebenfalls in den Experteninterviews

deutlich – adressiert der visuelle Ansatz den *Peak-Detection*-Anwendungsfall, da dieser durch die Clusteranalyse gut zu erkennen und durch die Filtermöglichkeiten der ersten Ebene einfach zu selektieren ist. Auf der zweiten Ebene ist dann deutlich zu sehen, welche Zusammensetzung der Netzwerkverkehr an dieser Stelle hat. Allerdings ist auch hier in manchen Fällen nicht immer interessant welche Peaks im Gesamtverkehrsvolumen vorkommen. Die Notwendigkeit einer Einschränkung auf Netzbereich wird dadurch noch einmal verstärkt.

8.2. Skalierbarkeit

Insbesondere im Netzwerk-Monitoring und Netzwerkmanagement muss mit großen, meist verteilt verwalteten Datenmengen gearbeitet werden. Eine gute Skalierbarkeit des visuellen Ansatzes ist somit von besonderem Interesse. Robertson et al. unterscheiden 5 wesentliche Arten der Skalierbarkeit im Bereich Visual Analytics [REE+09]: *Information Scalability, Visual Scalability, Display Scalability, Human Scalability* und *Computational Scalability*. Im Folgenden wird die Skalierbarkeit des visuellen Ansatzes hinsichtlich zweier dieser Kriterien untersucht:

- Visual Scalability
- Computational Scalability

8.2.1. Visual Scalability

Visuelle Skalierbarkeit ist die Fähigkeit des visuellen Ansatzes auch große Datenmengen, bezogen auf die Anzahl der Datenelemente und die Anzahl der Dimensionen, darstellen zu können [REE+09]. Die erste Ebene ermöglicht durch die Darstellung der Zeitreihencluster prinzipiell eine gute Skalierbarkeit mit der Anzahl an Tagen die untersucht werden sollen. *Visual Clutter* tritt hier nur auf, wenn zusätzlich die einzelnen Zeitreihen eingezeichnet werden, die Anzahl der Cluster – aufgrund sehr unterschiedlicher Zeitreihen – hoch gewählt werden muss oder die Cluster sich stark überlappen. Somit hängt die Komplexität der Ansicht nur teilweise von der tatsächlichen Datenmenge ab, sondern viel mehr von der Homogenität der Cluster. Die Auswahl einzelner Tage in der Zeitleiste unter der Zeitreihendarstellung ist hier allerdings problematisch. In einer Reihe können nur eine begrenzte Anzahl an Kreisen nebeneinander eingezeichnet werden, ohne dass diese sich überlappen und so die Auswahl der Tage erschweren oder ganz unmöglich machen. Um mit dem visuellen Ansatz auch größere Zeiträume zu betrachten, müsste das Konzept diesbezüglich erweitert werden. Eine hierarchische Organisation der Tage oder die Rückkehr zur Kalenderdarstellung nach Van Wijk et al. [VV99] könnten mögliche Lösungsansätze sein.

Die Parallel Sets Ansicht ist weitestgehend unabhängig von der Anzahl der zugrundeliegenden Datenelemente. Hier sind die Anzahl der Dimensionen und Kategorien die entscheidenden Faktoren, die für die Übersichtlichkeit der Darstellung sorgen. Diese werden anfänglich durch die Top-5-Beschränkung der Kategorien und die auf vier beschränkte Anzahl der Dimensionen begrenzt und sorgen so unabhängig von der Datenmenge für eine gleichbleibende Qualität der Darstellung. Dies stellt einen entscheidenden Vorteil gegenüber den verwandten parallelen Koordinaten dar, die ohne geeignete Gegenmaßnahmen ab einer gewissen Anzahl an Datenelementen *Visual Clutter* produzieren. Inwieweit neue Kategorien zur Darstellung hinzugefügt werden, bleibt den Anwender/-innen überlassen, die so die Übersichtlichkeit in der Darstellung aktiv steuern können. Ein weiteres

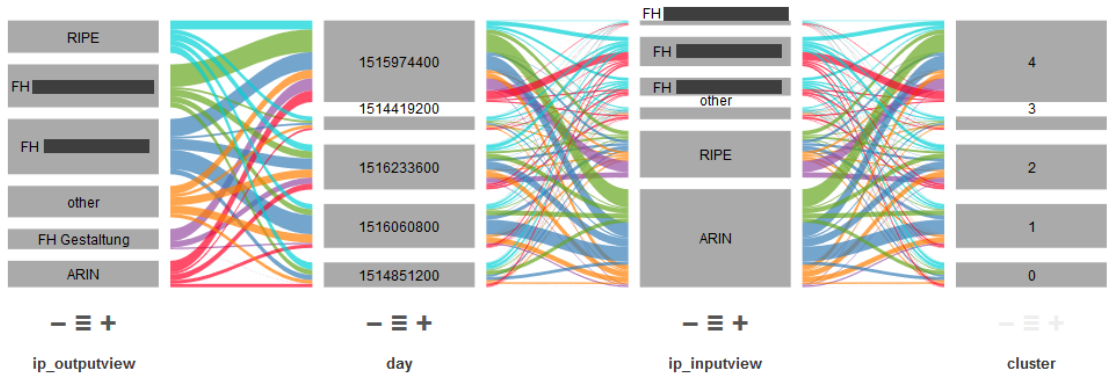


Abbildung 8.3.: Herrscht annähernd eine gleichmäßige Verteilung der Kategorien in der Parallel Sets Ansicht, wirkt die Darstellung schnell unübersichtlich. Durch die Interaktionsmöglichkeiten und die Definition eines Dimensionsfilters können einzelne Bereiche fokussiert und die Übersichtlichkeit in den Daten wiederhergestellt werden.

Kriterium für die Übersichtlichkeit der Darstellung ist die Verteilung der Datenelemente. Sind die Daten annähernd gleichverteilt, stößt die Ansicht an ihre visuellen Grenzen, da die Muster in den Daten nur schlecht zu erkennen sind und die Darstellung unübersichtlich wirkt. Abbildung 8.3 zeigt hierfür ein Beispiel. Durch Interaktion und Highlighting kann dieses Problem allerdings gelöst werden. Zudem kann durch setzen eines Dimensionsfilters der Fokus auf einen bestimmten Bereich der Daten gelenkt werden, der dann separat in der Filteransicht der zweiten Ebene betrachtet werden kann. Das größte Skalierungsproblem der Darstellung ergibt sich dann, wenn sehr große Datenmengen betrachtet werden, um darin Auffälligkeiten zu finden. Kleine Ausreißer in den Daten sind in der Summe der anderen Daten kaum zu erkennen und das Auffinden der Daten somit nahezu unmöglich. Hier liegt wiederum der Vorteil der Parallelen Koordinaten, da dort auch kleine Ausreißer gut sichtbar sind.

Bei der Streamgraph Ansicht gibt es besonders gravierende Schwierigkeiten bei der visuellen Skalierbarkeit bezogen auf den Analysezeitraum. Wie in Abbildung 8.2 bereits gezeigt wurde, nehmen bei starken Tageszeitbegrenzungen und gleichzeitig großen Zeitspannen die eigentlichen Daten nur wenig Platz in der Darstellung ein. Selbst wenn eine relativ kleine Datenmenge aufgrund der Filter dargestellt werden muss, können dadurch große Lücken in der Darstellung entstehen, die überproportional viel Platz einnehmen. Das Heranzoomen an die Daten ist zwar möglich, allerdings ist beispielsweise der Vergleich zweier Tagesgraphen, zwischen denen mehrere Wochen liegen, durch das Raus- und Reinzoomen mühselig. Da beide Graphen nicht gleichzeitig angezeigt werden können, werden den Anwender/-innen an dieser Stelle zu große kognitive Belastungen abverlangt. Zudem ist die verwendete Scrollbar ab einer gewissen Zeitspanne nicht mehr praktikabel, da präzise Einstellungen von Zeiträumen nicht mehr möglich sind. Eine mehrstufige Scrollbar wäre ein Ansatz, um dieses Problem zu adressieren.

Insgesamt ist das Konzept gut skalierbar, da die Daten in den meisten Ansichten weniger von den Datenmengen im Hintergrund abhängen, sondern andere Faktoren entscheidend sind, die sich auch bei größeren Datenmengen ähnlich verhalten. In den meisten Fällen sind nur kleine Änderungen, wie das Design der Scrollbars oder der Tagesleiste nötig, um auch größere Datenmengen über größere Zeiträume hinweg im System betrachten zu können. Die gravierendsten Änderungen sind

bei der Streamgraph Ansicht nötig, da diese in der jetzigen Form schnell visuell an ihre Grenzen stößt. Auf die allgemeine Skalierbarkeit des visuellen Ansatzes bezogen liegt der Flaschenhals der Anwendung jedoch bei der Bereitstellung der Daten, die im Folgenden untersucht wird.

8.2.2. Computational Scalability

Obwohl der Fokus der Arbeit nicht auf der Performanz des Prototyps liegt, darf eine angemessene Datenhaltung nicht außer Acht gelassen werden. Bei der Konzeption muss bedacht werden, dass die verwendeten Verfahren zur Berechnung der visuellen Ansichten und Beschaffung der erforderlichen Daten nicht nur für kleine Datensätze geeignet sind, sondern auch für größere Datenmengen funktionieren.

Im Prototypen werden nur kleine bereits aggregierte und vorberechnete Datenmengen an das Frontend zur Anzeige übermittelt. Speziell bei der Parallel Sets Ansicht ist die Datenmenge im Frontend unabhängig von der Datenmenge in der Datenbank. Die Anzahl der Kategorien und vor allem die Anzahl der Dimensionen sind die beiden wesentlichen Faktoren, die die Datenmenge der Parallel Sets beeinflussen. Werden diese jedoch zu hoch gewählt, kann die Datenmenge schnell problematisch werden. Für ein realistisches Szenario mit einer maximalen Anzahl von $X_{max} = 15$ Kategorien pro Achse und insgesamt $|D| = 6$ Dimensionen ergibt sich bereits ein Worst Case von $X_{max}^{|D|} \approx 11.400.000$ Datenelementen, die clientseitig verwaltet werden müssen. Eine Beschränkung der Anzahl an Dimensionen ist aus diesem Grund nötig, um die Interaktivität der Anwendung zu gewährleisten. Auch bei der Darstellung der Zeitreihen in der ersten und dritten Ebene kann es bei größeren Datenmengen zu Problemen kommen, wenn die zu untersuchende Zeitspanne groß wird. Dies stellt für die Anwendung allerdings weniger ein Problem dar, da derart große Zeitspannen bei der Analyse an anderen Stellen zu massiven Problemen führen würden. Insgesamt ist das Frontend somit gut skalierbar, da die Datenmenge nur eine untergeordnete Rolle bei der Skalierbarkeit einnimmt. Entscheidend ist hier, wie schnell die benötigten Daten bereitgestellt werden können.

Im jetzigen Prototypen wird ein Großteil der Daten im Backend vorbereitet. Damit auch das Backend gut skaliert, müssten diesbezüglich einige Änderungen vorgenommen werden. Beispielsweise müssten von der Datenbank kommende Dokumente bereits stärker aggregiert sein, sodass eine erneute Aggregation der Daten – wie sie im Moment nötig ist – nicht mehr oder nur in einem geringeren Ausmaß vorgenommen werden muss. Eine höhere Parallelisierung der Datenbankanfragen und anschließenden Verarbeitung, sowie eine effiziente In-Memory-Datenhaltung der Zwischenergebnisse müssten realisiert werden. Im jetzigen Stand ist das Backend nur bedingt skalierbar. Wie gut die Clusteranalyse skaliert hängt stark von dem verwendeten Verfahren und dem Distanzmaß ab. Für größere Datenmengen müsste bei der Auswahl die Skalierbarkeit des verwendeten Verfahrens genauer untersucht werden. Für die getesteten Datensätze sind die verwendeten Methoden dieser Arbeit ausreichend performant, zumal der Flaschenhals der Anwendung bei der Anfrage der Daten und somit beim Datenmanagement in der Datenbank liegt.

Die Datenhaltung des Ansatzes insgesamt bietet noch viel Potential für Verbesserungen. Durch die Verwendung der TopX-Collections wurden im Prototypen erste Ansätze realisiert, die die Performanz der Anfragen verbessern. Dennoch ist die Performanz des Prototypen in der jetzigen Form nicht ausreichend um große Datenmengen analysieren zu können. Hier ist mit der Wahl von MongoDB als Datenbank in Kombination mit den benötigten Anfragen für die Umsetzung des visuellen Konzepts allerdings bereits eine gute Basis für die Bewältigung großer verteilter

Datenmengen vorhanden. MongoDB unterstützt horizontale Skalierbarkeit durch den Einsatz von Sharding. Ein Vorteil der Datenanfragen ist, dass die benötigten Aggregationen der Daten hoch parallelisierbar sind. Wird der *Shard Key* so gewählt, dass sich die Daten, die aggregiert werden müssen, auf einem *Shard* befinden, können die Ergebnisse der einzelnen Anfragen einfach vereinigt werden. Aber auch wenn das nicht der Fall ist, sind die zu erwartenden Ergebnismengen pro *Shard* begrenzt. Werden alle Aggregationsschritte auf die Shards ausgelagert, müssen beispielsweise für die Parallel Sets Ansicht maximal $X_{max}^{|D|}$ mit X_{max} maximale Anzahl der Kategorien je Dimension und $|D|$ Anzahl der Dimensionen Dokumente zurückgeliefert werden. Diese Anzahl der Dokumente ist dabei unabhängig von der Datenmenge. So könnten die Anfragen auf unterschiedliche Server verteilt werden und auch große Datenmengen auf effiziente Weise angefragt werden. Inwiefern die verteilte Datenhaltung in der Praxis funktioniert, ist ein spannendes Thema für weiterführende Arbeiten an dem Prototypen.

9. Zusammenfassung und Ausblick

Diese Arbeit beschäftigt sich mit der Forschungsfrage, *wie Darstellungen von aggregierten Zeitreihen ergänzt werden können, damit zeitabhängige multivariate Netzwerkdaten vollständig dargestellt und hinsichtlich Anomalien untersucht werden können.* Um domänenspezifische Anforderungen der Anwender/-innen an das Visualisierungskonzept kennenzulernen, wurde eine Anforderungsanalyse mit Mitarbeitern und Kunden der IsarNet GmbH durchgeführt. Diese ergab, dass der Fokus des Konzepts auf der Zeit liegen und das Erkennen von Anomalien in den Daten im Mittelpunkt der Anwendung stehen soll. Zu den interessanten Anomalien in Netzwerkdaten zählen Häufungen im mehrdimensionalen Datenraum und Peaks und Dips im Gesamtverkehrsvolumen.

Für die Beantwortung der Forschungsfrage wurde in dieser Arbeit ein interaktiver Ansatz mit koordinierten Ansichten für die visuelle Analyse von Netzwerkdaten vorgestellt. Dieser stellt den zeitlichen Bezug der Netzwerkdaten in den Fokus der Analyse und verbindet mehrere hierarchisch organisierte Ansichten der Netzwerkdaten durch ein mentales Modell des Datenflusses. Die koordinierten Ansichten können drei Ebenen zugeordnet werden, die von oben nach unten den Arbeitsablauf der Analyse widerspiegeln. Zudem können in den ersten beiden Ebenen interaktive Filter auf die Daten angewendet werden, die durch Selektion der Daten direkt in den Ansichten realisierbar sind und die auf die darunterliegenden Ebenen wirken. Durch die Interaktionen zwischen den Ebenen werden die verschiedenen Ansichten der Daten zu einem Gesamtbild der Netzwerkdaten zusammengefügt. Die Umsetzung des Konzepts in Form eines webbasierten Softwareprototypen erfolgte durch eine typische dreiteilige Architektur bestehend aus Frontend, Backend und Datenbanksystem.

Um die umgesetzten Konzepte zu evaluieren wurden zwei gegensätzliche Anwendungsfälle vorgestellt, die die Anwendbarkeit des visuellen Ansatzes auf Probleme der Domäne demonstrieren. Für den *Peak-Detection*-Anwendungsfall – der auch in den anschließenden Experteninterviews als Aufgabe vorkam – eignet sich der visuelle Ansatz besonders gut. Zusätzlich wurde in Interviews Feedback von Domänenexperten eingeholt, die das Konzept insgesamt als geeignet einstufen und die Anwendung persönlich verwenden würden. Nachteile des Ansatzes werden vor allem bei der konkreten Umsetzung des Konzepts und der allgemein hohen Komplexität der Anwendung gesehen. Die Skalierbarkeit des visuellen Ansatzes ist insgesamt vielversprechend. Zwar gibt es im Prototypen die Performance betreffend Optimierungsbedarf, allerdings sind die Grundlagen für eine gute Skalierbarkeit des Konzepts vorhanden und auch die meisten der verwendeten Darstellungen sind für die Analyse großer Datenmengen geeignet.

Die Ergebnisse der Arbeit haben gezeigt, dass eine Kombination primär bestehend aus zwei unterschiedlichen Zeitreihendarstellungen, die durch eine Parallel Sets Darstellung miteinander verbunden sind, für die Analyse bestimmter Anomalien in Netzwerkdaten gut geeignet ist. Durch die Interaktionen zwischen den Ebenen werden die unterschiedlichen Ansichten auf die Daten miteinander verknüpft und so zu einem Gesamtbild zusammengefügt. Das vollständige Darstellen aller Netzwerkdaten auf einen Blick ist in diesem Konzept nicht möglich. Vielmehr wird die

Darstellung auf die wesentlichen Informationen beschränkt, die für die Analyse der Daten benötigt werden. Durch Interaktion mit den Ansichten können weitere Informationen aus den Daten gewonnen werden, die das Bild der Netzwerkdaten Stück für Stück vervollständigen. Insgesamt kann somit gesagt werden, dass das in dieser Arbeit entwickelte Konzept einen möglichen Lösungsansatz für die Forschungsfrage bietet.

9.1. Ausblick

In der Evaluation des Konzepts hat sich gezeigt, dass manche Punkte der Anforderungen bei der Konzeption noch nicht ausreichend adressiert wurden. Eine Auswahl möglicher Weiterentwicklungen des Konzepts, die speziell die Benutzbarkeit des visuellen Ansatzes verbessern könnten, sind nachfolgend beschrieben.

Erweiterungen der Parallel Sets Wird im aktuellen Prototypen ein Dimensionsfilter eingestellt, werden die angezeigten Top-Kategorien nicht verändert. Im Extremfall kann dadurch ein Großteil der Daten in die Restklasse *other* fallen. Dieses Problem könnte gelöst werden, indem die neuen Top-Kategorien automatisch dazukommen, sobald die Ansichten wechseln. Um die Nachvollziehbarkeit einiger Interaktionen zu verbessern, könnte zudem über den konsequenten Einsatz von sanften Übergängen (Interpolationen) zwischen den Effekten und Änderungen der Ansichten nachgedacht werden.

Vereinfachung der Clusteranalyse Eine bereits geplante, aber zeitlich nicht umgesetzte, Erweiterung der Clusteranalyse ist der Einsatz einer *Elbow Method*, um die optimale Anzahl an Clustern zu finden. Die Anwender/-innen könnten dann wahlweise die vorgeschlagene Anzahl übernehmen oder nach Bedarf manuell verändern. Zudem könnte für andere Clustermethoden und Distanzmaße untersucht werden, ob diese zu besseren Ergebnissen bei der Analyse führen.

Performance Optimierung Nicht zuletzt ist es sinnvoll sich über weitere Performance-Optimierungen Gedanken zu machen, da sich erst durch die schnelle Verarbeitung der Daten im Hintergrund weitere Interaktionsmöglichkeiten und Ansichten der Daten ergeben. Dazu zählt auch die Einführung eines impliziten Zeitfilters zwischen der ersten und zweiten Ebene. Für den Einsatz in einem realen Umfeld müsste die Datenhaltung noch einmal überarbeitet werden.

Aber auch komplett neue Lösungsansätze könnten in das bestehende Konzept des Workflows integriert werden. Einige der nachfolgenden Erweiterungen stehen allerdings mit bestehenden Bestandteilen des Konzepts im Konflikt, wodurch eine Integration dieser Ansätze umfangreiche Änderungen nach sich ziehen würde.

Topologie-Ebene Eine Erkenntnis, die vor allem aus den Experteninterviews heraus entstanden ist, ist die fehlende Möglichkeit sich auf Netzbereiche in der ersten Ebene einzuschränken. Aus den Interviews wurde die Notwendigkeit dieser Funktion vor allem für den Troubleshooting-Anwendungsfall deutlich. Daher existiert die Überlegung, vor der ersten Ebene eine weitere Ebene einzuführen, die mithilfe der Netzwerktopologie des Unternehmens die Auswahl von Netzbereichen ermöglicht. Eine graphbasierte Darstellung der Netzwerkverbindungen, bei der die Detailstufen interaktiv angepasst werden können, ist dabei denkbar.

Feedback-Loop Auch die Einführung eines Feedback-Loops ist vorstellbar, sodass Ergebnismengen der unteren Ebenen als neuer Startpunkt in der ersten Ebene übernommen werden können. Es müssten Überlegungen angestellt werden, inwiefern eine solche Funktion in den Workflow integriert werden kann, ohne gegen das mentale Modell des Datenflusses und den Workflow zu verstoßen. Das Propagieren der Auswirkungen auf alle Ebenen ist dabei zwingend. Welche weiteren Änderungen am Konzept vorgenommen werden müssen bleibt offen.

Time Slider Um den zeitlichen Bezug noch stärker in den Fokus der Anwendung zu rücken, kann über die Einführung eines *Time Sliders* für die zweite Ebene nachgedacht werden. Nach einen Vorbild von Chen et al. [CGY+14] kann durch die Auswahl eines Zeitfenster, welches über einen Slider verschoben werden kann, eine verhältnismäßig kleine Zeitspanne für die Parallel Sets Ansicht verwendet werden. Durch die Veränderung des Zeitfensters kann der zeitliche Verlauf der Daten sichtbar gemacht werden. Dafür sind allerdings einige Anpassungen im Konzept der zeitlichen Filterauswahl vorzunehmen, da diese im Konflikt mit dem *Time-Slider-Ansatz* stehen. Für die dritte Ebene hätte dies allerdings einen entscheidenden Vorteil: In der Streamgraph Ansicht könnte das ausgewählte Zeitfenster der zweiten Ebene dargestellt werden und mit einer Veränderung des Zeitfensters automatisch mitangepasst werden. Die Skalierbarkeit der Streamgraph Ansicht würde sich dadurch deutlich verbessern.

Nicht zuletzt wäre es interessant, das Konzept auch auf Netzwerkdaten anderer Quellen zu testen und die Skalierbarkeit mit großen verteilten Datenmengen – wie in Abschnitt 8.2.2 beschrieben – zu untersuchen. Inwieweit das Konzept auch auf andere Teilgebiete des Netzwerkmanagements, wie die Netzwerksicherheit, übertragbar ist, kann ebenfalls Thema einer weiterführenden Arbeit sein.

A. Fragebogen Anforderungsanalyse

IsarNet Fragebogen A

Fragen zur Person

Falls noch nicht in der Vorstellungsrunde geklärt:
Welche Rolle haben sie im Unternehmen?

Welchen Ausbildungshintergrund bringen Sie mit?

Wie lange sind Sie schon dabei? Sowohl bei IsarNet als auch bei IsarFlow.

Haben Sie abgesehen von IsarFlow Erfahrungen im Bereich Visualisierung oder Visual Analytics gesammelt? Welche?

Allgemeine Fragen

Welche Einsatzgebiete hat IsarFlow?
Auf der IsarFlow-Website sind einige Punkte aufgelistet, für die IsarFlow Analysemöglichkeiten anbietet.

- Effiziente und zielgerichtete Kapazitätsplanung im Netzwerk ("Right-Sizing")
- Frühzeitige Erkennung von im Netz aktiven Viren und Würmern
- Entwicklung einer geeigneten QoS-Strategie
- Kontrolle der Auslastung der unterschiedlichen Verkehrsklassen
- Troubleshooting
- weitere?

Gibt es noch weitere Punkte/Einsatzgebiete?
Könnten Sie erklären, was in den Bereichen jeweils mit IsarFlow möglich ist?
In welchen Bereichen liegt der Fokus im Moment (Priorisierung)?
Soll sich zukünftig daran etwas ändern?
Wie sieht es aus Kundensicht aus: Wofür wird IsarFlow tatsächlich überwiegend verwendet? Abweichungen?
Sind die Kunden daran interessiert mit interaktiven Visualisierungen zu arbeiten oder finden andere Funktionen (wie z.B. Reports) häufiger Anwendung? Wissen sie warum?

Falls Sicherheitsaspekte (Intrusion Detection) wichtig sind, welche Arten von Angriffen auf das Netzwerk sollen erkennbar sein?
Welche sind es schon?

- DDoS, DoS
- Network Scan (Port Scan, IP Scan)
- Würmer / Viren
- bestimmte Anomalien
- weitere?

Welche Zielgruppen werden adressiert? Wie sehen ihre typischen Kunden aus, und was machen diese mit IsarNet?

- Netzwerker
- Gelegenheitsnutzer (Für welche Zwecke?)
- IT-Sicherheitsexperten
- weitere?

Aus Nutzersicht

Wie kann ich mir die typische Benutzung abhängig von der Anwendergruppe vorstellen.

- kurze Interaktionen in regelmäßigen Abständen
- bei Bedarf tiefgehende Analysen
- insgesamt eher kurze/lange Benutzung

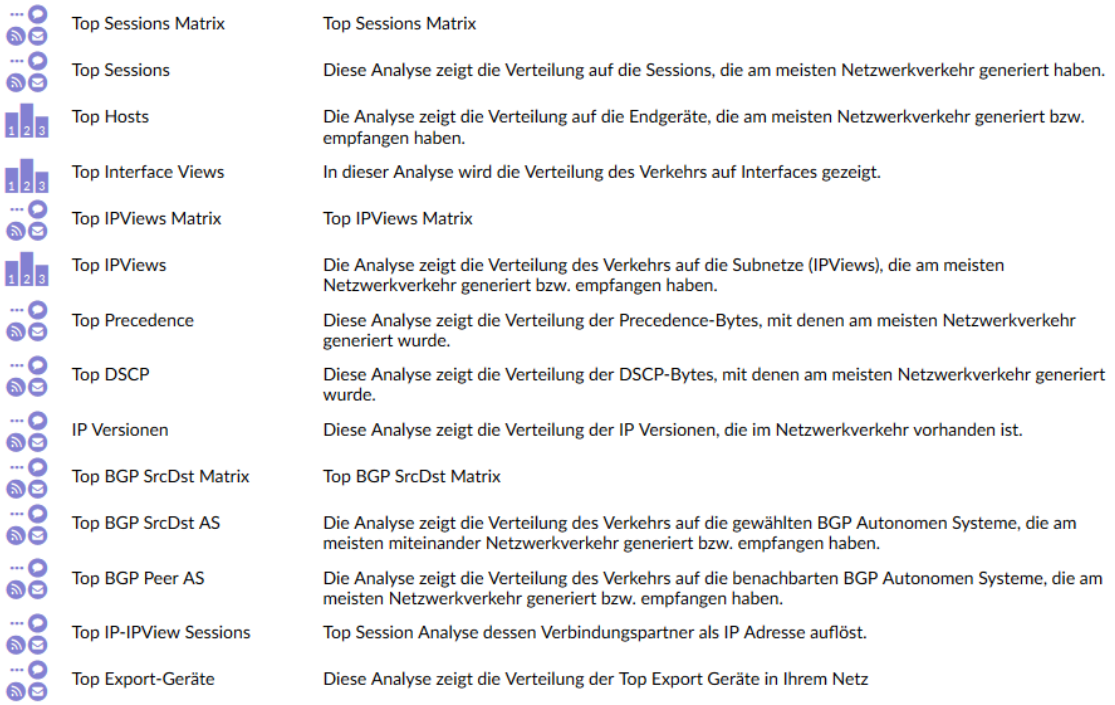
Sind Ihnen andere Tools oder Software bekannt, die in Kombination genutzt werden? Falls ja, wofür?

Beim Aufruf der Web-GUI, welche Ansichten sind für den Nutzer zu Beginn interessant?
Gibt es bestimmte Fragen unabhängig von der konkreten Aufgabenstellung, die man sich immer am Anfang der Analyse stellt?

Sind es immer die gleichen Ansichten, mit denen die Nutzer starten, oder gibt es je nach Einsatzgebiet unterschiedliche Startpunkte?

Welche Analysen werden typischerweise mit IsarFlow gemacht?
Gibt es Standard-Analysen die häufig verwendet werden? Warum?
Gibt es Standard-Analysen die im Moment selten oder überhaupt nicht verwendet werden? Warum?
Falls bestimmte Standard-Analysen nicht genannt wurden: Was ist mit [...]?

Matrix Analysen oder Timeline Analysen, welche werden da häufiger verwendet?



Top Sessions Matrix	Top Sessions Matrix
Top Sessions	Diese Analyse zeigt die Verteilung auf die Sessions, die am meisten Netzwerkverkehr generiert haben.
Top Hosts	Die Analyse zeigt die Verteilung auf die Endgeräte, die am meisten Netzwerkverkehr generiert bzw. empfangen haben.
Top Interface Views	In dieser Analyse wird die Verteilung des Verkehrs auf Interfaces gezeigt.
Top IPViews Matrix	Top IPViews Matrix
Top IPViews	Die Analyse zeigt die Verteilung des Verkehrs auf die Subnetze (IPViews), die am meisten Netzwerkverkehr generiert bzw. empfangen haben.
Top Precedence	Diese Analyse zeigt die Verteilung der Precedence-Bytes, mit denen am meisten Netzwerkverkehr generiert wurde.
Top DSCP	Diese Analyse zeigt die Verteilung der DSCP-Bytes, mit denen am meisten Netzwerkverkehr generiert wurde.
IP Versionen	Diese Analyse zeigt die Verteilung der IP Versionen, die im Netzwerkverkehr vorhanden ist.
Top BGP SrcDst Matrix	Top BGP SrcDst Matrix
Top BGP SrcDst AS	Die Analyse zeigt die Verteilung des Verkehrs auf die gewählten BGP Autonomen Systeme, die am meisten miteinander Netzwerkverkehr generiert bzw. empfangen haben.
Top BGP Peer AS	Die Analyse zeigt die Verteilung des Verkehrs auf die benachbarten BGP Autonomen Systeme, die am meisten Netzwerkverkehr generiert bzw. empfangen haben.
Top IP-IPView Sessions	Top Session Analyse dessen Verbindungspartner als IP Adresse auflöst.
Top Export-Geräte	Diese Analyse zeigt die Verteilung der Top Export Geräte in Ihrem Netz

Wie sieht die Interaktion mit den unterschiedlichen Ansichten aus?
Wird häufig zwischen den verschiedenen Ansichten gewechselt? Falls ja, warum?

- um auch die anderen Attribute zu sehen
  ⇒ Würden Sie sich wünschen, dass mehr Attribute in einem Diagramm vereint wären?
  ⇒ Denken Sie es wäre besser, wenn die zuvor dargestellten Informationen in irgendeiner Form sichtbar bleiben?
- um einen anderen Sachverhalt zu verfolgen

Finden dabei häufig die Drilldown Operationen Anwendung?
Bleiben die Nutzer häufig bei den Default Einstellungen oder werden viele der Analyse-Parameter verwendet?
Welche werden häufig gebraucht, welche selten?

- Netzbereich Parameter
  - InterfaceView
  - IPView
  - IPView B
  - IP Adresse
- IP Version
- Einheiten
- Protokoll
- IP CoS

Wird der Analyse-Vergleich häufig verwendet? Für welche Zwecke wird der Vergleich verwendet?

Insgesamt: Welche Aussage trifft für Sie am ehesten zu?

- Die verschiedenen Analysen werden verwendet um die Daten Stück für Stück zu explorieren und neue Erkenntnisse zu gewinnen.
- Die verschiedenen Analysen werden nacheinander betrachtet um eine Übersicht über den Netzwerkverkehr zu bekommen.

Nutzung und Use Cases

Wir hatten bereits über die Einsatzgebiete gesprochen.
Jetzt würde ich gerne ein paar Fragen zu den typischen Use Cases stellen.

Pro Use Case:

Beschreiben Sie was das Ziel des Use Cases ist.

Wie sieht der Ablauf mit IsarFlow optimalerweise aus?
Gibt es überhaupt einen typischen Ablauf oder ist das Vorgehen von Fall zu Fall sehr unterschiedlich?

Welche Probleme und Sonderfälle können dabei auftreten?
Wie werden diese Probleme gelöst?
Können diese Probleme mit den Funktionen von IsarFlow gelöst werden oder müssen sich die Nutzer anderweitig helfen? Falls ja, was wird stattdessen gemacht?
Gibt es zusätzliche Software, die verwendet wird um bestimmte fehlende Informationen zu ergänzen?

Gibt es weitere Fragestellungen in diesem Zusammenhang, die im Moment nur umständlich oder nicht gelöst werden können, die aber für die Analyse interessant oder hilfreich wären?

Denken Sie dass die Aufgabe insgesamt gut lösbar ist oder gibt es Schwächen?
Falls es Schwächen gibt, welche?

- Ist das Konzept ungeeignet für diese Aufgabe? (schwer zu erkennen, unübersichtlich, ungeeignet)
- Ist die Interaktion ungeeignet oder umständlich?
- Fehlen Komponenten dafür?

Falls gut lösbar, warum?

Gibt es sonst noch Anmerkungen zu diesem Use Case?

Neue Use Cases:

Gibt es mögliche Use Cases die überhaupt nicht vorkommen, aber gewünscht wären? Wie würden die aussehen?

Stichwort zeitliche Dynamik: Würden Sie sich wünschen dass es mehr Möglichkeiten gibt, sich einen Überblick über den zeitlichen Verlauf des Netzverkehrs zu machen, sodass aktuelle Trends erkennbar sind?
Denken Sie, dass Trends bereits ausreichend gut erkennbar sind? Welche?
In welchen Einsatzgebieten sehen Sie Vorteile für die Analyse, wenn Trends besser erkennbar wären?

Möchte man auch verschiedene Zeitbereiche miteinander vergleichen können. Für was wäre das sinnvoll?

In der alten Version 4.x gab es noch Baselining, in der aktuellen Version zumindest im Moment noch nicht.
Gibt es Use Cases für die diese Funktion sinnvoll wäre? Welche?
Welche Zeitperiode ist dann am interessantesten? Tag/Woche/Monat/Jahr?

Anmerkungen und Wünsche

Gibt es darüber hinaus weitere Anmerkungen?

Gibt es darüber hinaus irgendwelche Probleme die Sie noch nicht genannt haben?
Irgendwelche Schwächen oder Stärken von IsarFlow?
Gibt es weitere Funktionen, die Sie sich wünschen würden? Oder gibt es noch irgendetwas von dem Sie glauben, dass es für mich noch interessant sein könnte.

Ausblick

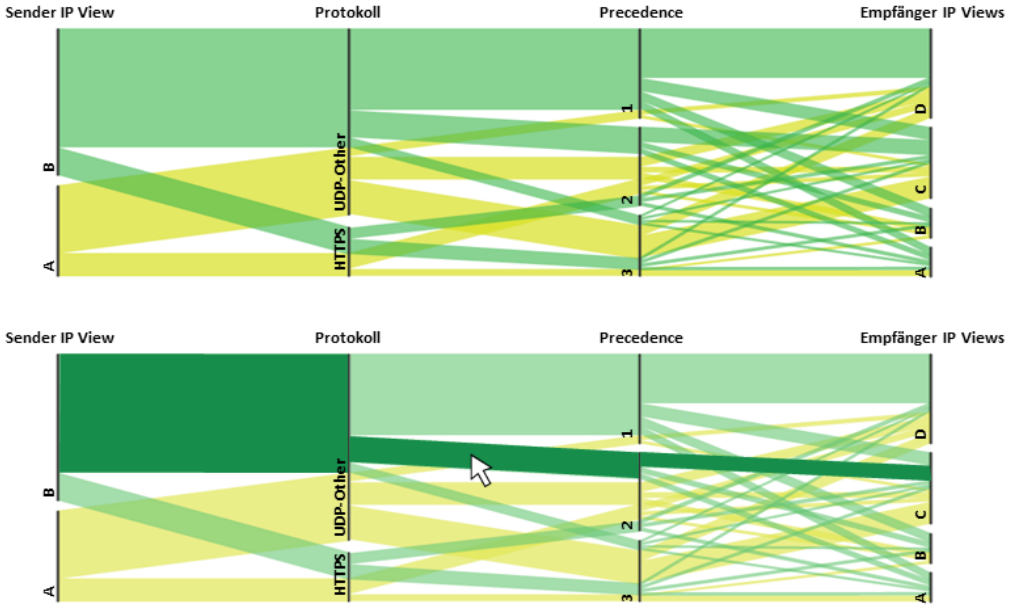


Figure 0.1: Netzwerkverkehr als Parallel Set

Minimalbeispiel um das Konzept von Parallel Sets zu verdeutlichen. Es wird der Verkehr zwischen IPView A und B als Sender und IPView A, B, C und D als Empfänger zusammen mit den Attributen Precedence und Protokoll dargestellt. Die Breite der Balken entspricht der Anzahl der übertragenen Bytes.

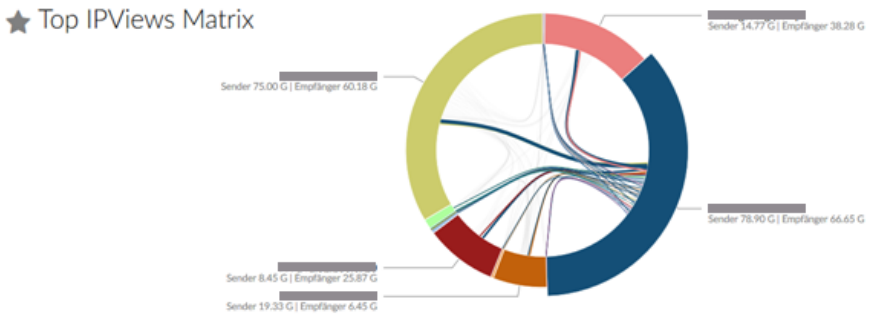


Figure 0.2: Vergleichbare Darstellung in IsarFlow

Das erste Diagramm enthält die gleichen Informationen wie die Top IPView Matrix mit folgenden Unterschieden:

- Einschränkung auf 2 Sender IPViews (einstellbar)
- Einschränkung auf 4 Empfänger IPViews (einstellbar)
- zusätzliche Anzeige von Precedence
- zusätzliche Anzeige der Protokolle

DB Fragebogen A

Fragen zur Person

Falls noch nicht in der Vorstellungsrunde geklärt:

Welche Rolle haben sie im Unternehmen?

Wie lange arbeiten Sie schon bei der Deutschen Bahn?

Wie lange nutzen Sie schon IsarFlow?

Welchen Ausbildungshintergrund bringen Sie mit?

Haben Sie abgesehen von IsarFlow Erfahrungen im Bereich Visualisierung oder Visual Analytics gesammelt? Welche?

Allgemeine Fragen

Für welche Einsatzgebiete verwenden Sie IsarFlow?

- Effiziente und zielgerichtete Kapazitätsplanung im Netzwerk ("Right-Sizing")
- Frühzeitige Erkennung von im Netz aktiven Viren und Würmern
- Entwicklung einer geeigneten QoS-Strategie
- Kontrolle der Auslastung der unterschiedlichen Verkehrsklassen
- Troubleshooting
- weitere?

Auf welchen Einsatzgebieten liegt dabei Ihr Fokus im Moment?
Gibt es darüber hinaus andere Einsatzgebiete für IsarFlow in Ihrem Unternehmen?
Für welche Zwecke wird dort IsarFlow verwendet?

Falls Sicherheitsaspekte (Intrusion Detection) wichtig sind, welche Arten von Angriffen auf das Netzwerk sollen erkennbar sein? Welche sind es schon?

- DDoS, DoS
- Network Scan (Port Scan, IP Scan)
- Würmer / Viren
- bestimmte Anomalien
- weitere?

Mir wurde gesagt, dass Sie im Moment Version 4 von IsarFlow verwenden.
Welche Gründe gibt es für Sie, bei dieser Version zu bleiben?
Gibt es daneben Gründe, die für einen Wechsel sprechen würden?

Wie kann ich mir die typische Benutzung vorstellen?

- kurze Interaktionen in regelmäßigen Abständen
- bei Bedarf tiefgehende Analysen
- insgesamt eher kurze/lange Benutzung

Wie häufig verwenden Sie somit IsarFlow?

Sind Sie dabei eher daran interessiert mit den interaktiven Visualisierungen zu arbeiten oder verwenden Sie IsarFlow um zum Beispiel Reports oder Schwellwerte zu definieren?

Bei ihrer Arbeit mit IsarFlow, verwenden Sie andere Tools oder Software in Kombination mit IsarFlow? Zum Beispiel um fehlende Informationen zu ergänzen? Falls ja, welche und warum?

Wenn Sie mit der Benutzung von IsarFlow beginnen, welche Analysen sind für Sie zuerst interessant?
Gibt es da bestimmte Fragen unabhängig von der konkreten Aufgabenstellung, die man sich immer am Anfang der Analyse stellt?

Sind es immer die gleichen Standard-Analysen mit denen Sie starten oder gibt es je nach Zweck unterschiedliche Startpunkte? Falls immer die gleichen, welche?

Gibt es Standard-Analysen die Sie häufig verwenden? Welche und warum?
Gibt es Standard-Analysen die Sie selten oder überhaupt nicht verwenden? Welche und warum?

. . . . . .

Wie sieht die Interaktion mit den unterschiedlichen Analysen aus.
Wechseln Sie häufig zwischen den verschiedenen Analysen? Falls ja, aus welchen Gründen?

- um auch die anderen Attribute zu sehen
  ⇒ Würden Sie sich wünschen, dass mehr Attribute in einem Diagramm vereint wären?
  ⇒ Denken Sie es wäre besser, wenn die zuvor dargestellten Informationen in irgendeiner Form sichtbar bleiben?
- um einen anderen Sachverhalt zu verfolgen

Verwenden Sie dabei die Drilldown Operationen von IsarFlow?
Bleiben Sie dabei häufig bei den Default Einstellungen oder verwenden sie oft auch die Analyse-Parameter?
Welche verwenden sie dabei häufig?

In der Version 5 von IsarFlow gibt es die Möglichkeit, 2 Analysen miteinander zu vergleichen. Würden Sie diese Funktion verwenden? Für welche Zwecke würden Sie diesen Vergleich verwenden?

IsarFlow stellt echtes Baselining für Netflow-Daten zur Verfügung.
Verwenden Sie diese Funktion? Warum?
Welche Zeitperiode ist für Sie am interessantesten?
Tag/Woche?

Baselining ist eine Methode, um langfristige Grunddaten zu erfassen und somit eine Referenz zu erstellen. Sie bilden also die Grundlage, um Abweichungen leichter zu erkennen.

Insgesamt: Welche Aussage trifft für Sie am ehesten zu?

- Die verschiedenen Analysen werden verwendet, um die Daten Stück für Stück zu explorieren und neue Erkenntnisse zu gewinnen.
- Die verschiedenen Analysen werden nacheinander betrachtet, um eine Übersicht über den Netzwerkverkehr zu bekommen.

. . . . . .

Use Cases

Wir hatten bereits über die Einsatzgebiete gesprochen.
Jetzt würde ich gerne ein paar Fragen zu den typischen Use Cases stellen.

Pro Use Case:

Beschreiben Sie was das Ziel des Use Cases ist.

Wie sieht der Ablauf mit IsarFlow optimalerweise aus?
Gibt es überhaupt einen typischen Ablauf oder ist das Vorgehen von Fall zu Fall sehr unterschiedlich?

Welche Probleme und Sonderfälle können dabei auftreten?
Wie lösen Sie diese Probleme?
Können diese Probleme mit den Funktionen von IsarFlow gelöst werden oder müssen Sie sich anderweitig helfen?
Falls ja, was machen Sie stattdessen?
Gibt es zusätzliche Software die Sie verwenden um bestimmte fehlende Informationen zu ergänzen?

Gibt es weitere Fragestellungen in diesem Zusammenhang, die im Moment nur umständlich oder nicht gelöst werden können, die Sie aber für die Analyse interessant oder hilfreich finden?

Denken Sie dass die Aufgabe insgesamt gut lösbar ist oder gibt es Schwächen?
Falls es Schwächen gibt, welche?

- Ist das Konzept ungeeignet für diese Aufgabe? (schwer zu erkennen, unübersichtlich, ungeeignet)
- Ist die Interaktion ungeeignet oder umständlich?
- Fehlen Komponenten dafür?

Falls gut lösbar, warum?

Gibt es sonst noch Anmerkungen zu diesem Use Case?

Neue Use Cases:

Gibt es mögliche Use Cases die überhaupt nicht vorkommen, aber gewünscht wären? Wie sehen diese aus?

Stichwort zeitliche Dynamik:
Würden Sie sich wünschen dass es mehr Möglichkeiten gibt, sich einen Überblick über den zeitlichen Verlauf des Netzverkehrs zu machen, sodass aktuelle Trends erkennbar sind?
Denken Sie, dass Trends bereits ausreichend gut erkennbar sind?
Welche Trends würden Sie interessieren?
In welchen Einsatzgebieten sehen Sie Vorteile für die Analyse, wenn Trends besser erkennbar wären?

Möchte man auch verschiedene Zeitbereiche miteinander vergleichen können?
Gibt es spontan etwas, das Sie interessieren könnte?

Anmerkungen und Wünsche

Gibt es darüber hinaus weitere Anmerkungen?

Gibt es darüber hinaus irgendwelche Probleme, die Sie noch nicht genannt haben?
Was denken Sie sind die Schwächen oder Stärken von IsarFlow?
Gibt es weitere Funktionen, die Sie sich wünschen würden?

B. Fragebogen Evaluation

1 Einführung

1.1 Workflow

① **Ebene 1:** geclusterte Zeitreihen (je 24h) als Line Chart + Zeitfilter

② **Ebene 2:** Parallel Sets + Dimensionsfilter

③ **Details:** Streamgraph (Zeit + eine Dimension)

④ **Details:** geclusterte Zeitreihen (je 24h) als Line Chart

⑤ **Filter:** Auf allen darunter liegenden Ebenen wird der hier angezeigte Filter angewandt

Workflow-Prinzip: von oben nach unten weniger Daten (durch Filter) und mehr Details

1.2 Ebene 1: Line Chart mit Clustering

- Display Einstellung → Legende
- Tag Filter (unten)
- Tag Filter (oben)
- Cluster Filter → Legende
- Median Filter → Legende
- Tageszeit Filter → Selection im Chart
- Tooltips mit Zusatzinfos → MouseHover (falls vorhanden)
- Scrollbar

1.3 Globale Einstellungen

- Clusteranzahl verändern
- Daten mit neuen Clustereinstellungen holen → refresh data
- Daten mit neuen Clustereinstellungen holen mit Filtereinstellungen → refresh data with selection

1.4 Ebene 2: Parallel Sets

- Achsen hinzufügen → Button "select new axis"
- Achsen entfernen → Kontextmenü auf dem jeweiligen Wert
- Achsen umsortieren → Drag
- Wert zum Filter hinzufügen/entfernen → Kontextmenü auf dem jeweiligen Wert
- mehr/weniger Werte anzeigen → + / -
- Werte aus einer Liste hinzufügen → ≡
- Filter zurücksetzen → Button "clear filter"
- durch Cluster toggeln → Button "toggle cluster"
- Tooltips mit Zusatzinfos → MouseHover (falls vorhanden)

1.5 Streamgraph

- vertical Zoom → Mousewheel
- Zoom auf einen Tag → Klick im Chart
- Zoom aufheben → erneut Klick im Chart
- Wert nach unten verschieben → Klick auf den Wert
- Tooltips mit Zusatzinfos → MouseHover (falls vorhanden)
- Scrollbar

1.6 Line Chart

- Display Einstellung
- Tooltips mit Zusatzinfos → MouseHover (falls vorhanden)
- Scrollbar

2 Studie

2.1 Peak Detection auf allen Ebenen

In der 4h Granularität gibt es einen Tag der abends deutlich mehr Verkehrsvolumen besitzt. Finden Sie diesen Tag und finden Sie heraus, welche Kombination aus Protokoll, IP-InputView und IP-OutputView hauptsächlich dafür verantwortlich ist. Verifizieren Sie mithilfe der Detailansichten ob diese Kombination korrekt ist.

Einzelne Schritte:

- Klicken Sie auf "Szenario 1" um den Startpunkt der Analyse einzustellen.
- Finden Sie den Tag in Ebene 1.
- Finden Sie mit Hilfe der Ebene 2 heraus, welches Protokoll, IP-InputView und IP-OutputView hauptsächlich verantwortlich sind.
- Erstellen Sie einen Filter mit dieser Kombination in der Ebene 2 und wenden Sie diesen an.
- Verifizieren Sie das Ergebnis in Ebene 3.

2.2 Peak Detection mit Filter

Suchen Sie in der 5min Granularität nach einem Tageszeitraum, bei dem ein oder mehrere Tage den gleichen auffälligen Kurvenverlauf zeigen.

a) Wählen Sie die betroffenen Tage aus und finden Sie heraus, welche Art von Verkehr dafür verantwortlich ist.

Einzelne Schritte:

- Klicken Sie auf "Szenario 2" um den Startpunkt der Analyse einzustellen.
- Wählen Sie nur die betroffenen Tage und den passenden Tageszeitraum aus.
- Wenden Sie diesen Zeitfilter an.
- Finden Sie mit Hilfe der Ebene 2 heraus, welches Protokoll, IP-InputView und IP-OutputView hauptsächlich verantwortlich sind.

b) Vergleichen Sie anschließend einen repräsentativen Tag eines "normalen" und eines auffälligen Tages miteinander. Welche Unterschiede sind sichtbar?

Einzelne Schritte:

- Verwenden Sie Clustering sodass alle "normalen" Tage in einem Cluster sind und alle auffälligen in einem anderen.
- Wählen Sie einen repräsentativen Tag eines normalen und eines auffälligen Tages aus.
- Stellen sie den passenden Tageszeitraum ein.
- Vergleichen Sie die beiden Tage miteinander in der Ebene 2 (Parallel Set).
- Verwenden Sie dafür auch die "Toggle Cluster"-Funktion.

2.3 Häufungen im mehrdimensionalen Raum

Suchen Sie in der Ebene 2 nach häufig vorkommenden Kombinationen aus Protokoll, IP-InputView und IP-OutputView. Überprüfen Sie in Ebene 3 ob die Daten auch einen zeitlichen Zusammenhang besitzen.
(Hoher zeitlicher Zusammenhang = Peak, geringer zeitlicher Zusammenhang = gleichmäßig verteilt)

Einzelne Schritte:

- Klicken Sie auf "Szenario 3" um den Startpunkt der Analyse einzustellen.
- Verwenden Sie den Dimensionsfilter in Ebene 2 und die Maus-Hover-Effekt in Ebene 2 und 3.

3 Fragebogen

Ist Clustering bereits bekannt?

Ist das Konzept der Parallel Sets (oder Parallel Coordinates) bereits bekannt?

3.1 Darstellung / Visualisierung der Daten + Interaktion

Ist der Workflow verständlich und nachvollziehbar?
Erinnerung:
Workflow: Ebene 1 (Clustering + Zeitfilter) → Ebene 2 (Parallel Sets + Dimensionsfilter) → Ebene3 (Details)

Ist klar, welche Filter in den jeweiligen Ebenen angewandt werden und welche Daten somit auf den darunter liegenden Ebenen zu sehen sind?

Wird dadurch ein nachvollziehbarer Drilldown realisiert?

Sind die Zusammenhänge zwischen den einzelnen Ebenen erkennbar, sodass klar ist welche Darstellungen sich auf die gleichen Daten beziehen?

Sind die Darstellungen überladen oder unübersichtlich?

Sind die Interaktionen überladen?
Helfen die Interaktionen/Effekte die Daten besser zu verstehen?
Gibt es Interaktionen/Effekte die unnötig sind oder sogar stören?

Können mehrere Dimensionen der Daten gleichzeitig mithilfe des Prototyps betrachtet werden?
Bleibt dabei der zeitliche Bezug im Fokus?
Können Anomalien in den Daten erkannt werden?

3.2 Think Aloud Studie – Anwendungsfälle

Kann Szenario 1 mit dem Prototypen gut gelöst werden? Begründung?

Kann Szenario 2 mit dem Prototypen gut gelöst werden? Begründung?

Kann Szenario 3 mit dem Prototypen gut gelöst werden? Begründung?

Insgesamt:

Ist das Konzept mit dem realisierten Workflow und den Filtern sinnvoll für die Aufgabenstellungen?

3.3 Gesamteindruck

Rückblickend auf das erste Interview:

- Welche Anmerkungen/Änderungswünsche/Vorschläge aus dem ersten Interview sind eingeflossen?
- Welche Anmerkungen/Änderungswünsche/Vorschläge aus dem ersten Interview sind nicht eingeflossen?
- Welche Vorteile sehen Sie bei diesem Ansatz gegenüber IsarFlow?
- Welche Nachteile sehen Sie bei diesem Ansatz gegenüber IsarFlow?

Fehlen grundlegende Darstellungen / Funktionen im System?
Falls ja, welche?

Ist das System insgesamt eher einfach zu verstehen oder eher komplex?
Falls komplex, unnötig komplex?

Ist die Bedienung insgesamt eher einfach oder eher mühsam?
Ist die Bedienung selbsterklärend?
Gibt es etwas, das bei der Benutzung nervig werden könnte?

Können Sie sich vorstellen, dass die meisten Leute schnell lernen würden, mit diesem System umzugehen?
Können Sie sich vorstellen, dass Sie schnell lernen würden, mit diesem System umzugehen?

Können Sie sich vorstellen, das System häufiger zu benutzen?

Sonstige Anmerkungen?

C. Prototypen Vorher-Nachher-Vergleich

Abbildung C.1.: Prototyp vor den Experteninterviews aus Kapitel 7

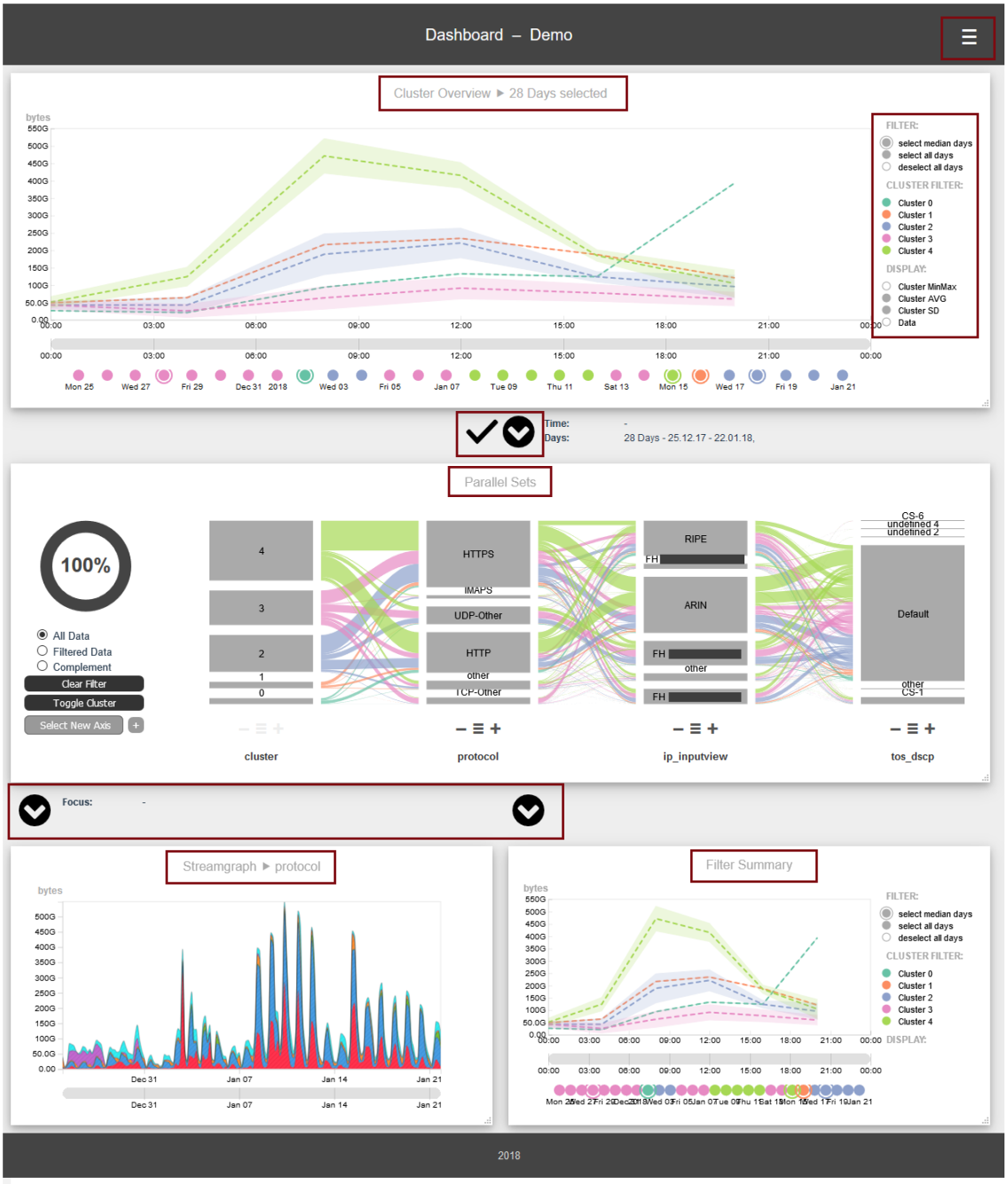


Abbildung C.2.: Prototyp nach den Experteninterviews aus Kapitel 7. Alle Änderungen wurden rot umrandet.

Literaturverzeichnis

[AMST11] W. Aigner, S. Miksch, H. Schumann, C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011 (zitiert auf S. 28).

[AOL04] A. O. Artero, M. C. F. de Oliveira, H. Levkowitz. „Uncovering clusters in crowded parallel coordinates visualizations“. In: IEEE. 2004, S. 81–88 (zitiert auf S. 27).

[Aut18] Automon. *Automon – Automated Performance Monitoring*. 2018. URL: https://automon-projekt.de/ (zitiert auf S. 72).

[BCH+10] R. Berthier, M. Cukier, M. Hiltunen, D. Kormann, G. Vesonder, D. Sheleheda. „Nfsight: netflow-based network awareness tool“. In: *Proceedings of LISA'10: 24th Large Installation System Administration Conference*. 2010, S. 119 (zitiert auf S. 31).

[BHL07] E. Bertini, P. Hertzog, D. Lalanne. „SpiralView: towards security policies assessment through visual correlation of network resources with evolution of alarms“. In: *Visual Analytics Science and Technology, 2007. VAST 2007. IEEE Symposium on*. IEEE. 2007, S. 139–146 (zitiert auf S. 32).

[BKH05] F. Bendix, R. Kosara, H. Hauser. „Parallel sets: visual analysis of categorical data“. In: *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. IEEE. 2005, S. 133–140 (zitiert auf S. 28, 47).

[Car99] M. Card. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999 (zitiert auf S. 21, 22).

[CGY+14] S. Chen, C. Guo, X. Yuan, F. Merkle, H. Schaefer, T. Ertl. „Oceans: Online collaborative explorative analysis on network security“. In: *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*. ACM. 2014, S. 1–8 (zitiert auf S. 30, 31, 85).

[Che73] H. Chernoff. „The use of faces to represent points in k-dimensional space graphically“. In: *Journal of the American statistical Association* 68.342 (1973), S. 361–368 (zitiert auf S. 27).

[Cis12] Cisco Systems, Inc. *Introduction to Cisco IOS NetFlow – A Technical Overview*. 1. Mai 2012. URL: https://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/ios-netflow/prod_white_paper0900aecd80406232.html (zitiert auf S. 19).

[CLK09] H. Choi, H. Lee, H. Kim. „Fast detection and visualization of network attacks on parallel coordinates“. In: *computers & security* 28.5 (2009), S. 276–288 (zitiert auf S. 31).

[CMS+13] S. Chen, F. Merkle, H. Schaefer, C. Guo, H. Ai, X. Yuan, T. Ertl. „VAST 2013 mini challenge 3: AnNetTe collaboration oriented visualization of network data“. In: *IEEE VIS*. 2013 (zitiert auf S. 30, 31).

[Don17] A. Donner. *Was ist Netzwerk-Monitoring?* 2017. URL: https://www.ip-insider.de/was-ist-netzwerk-monitoring-a-642648/ (zitiert auf S. 18).

[Eck05] L. Eckert. *Vorlesungsunterlagen – Netzwerkmanagement in der Automatisierungstechnik in der Automatisierungstechnik*. 2005. URL: http://www.w3service.net/vorlesungen/verteilte-systeme/0062-netzwerkmanagement/Netzwerk-Management-FCAPS-026.pdf (zitiert auf S. 17, 18).

[ECN+15] T. C. Eskridge, M. Carvalho, F. Nembhard, H. Thotempudi, P. J. Polack. „Interactive visualization of netflow traffic“. In: *Intelligence and Security Informatics Conference (EISIC), 2015 European*. IEEE. 2015, S. 188–188 (zitiert auf S. 31).

[FK14] F. Fischer, D. A. Keim. „NStreamAware: Real-time visual analytics for data streams to enhance situational awareness“. In: *Proceedings of the Eleventh Workshop on Visualization for Cyber Security*. ACM. 2014, S. 65–72 (zitiert auf S. 30, 31).

[FWR99] Y.-H. Fua, M. O. Ward, E. A. Rundensteiner. „Hierarchical parallel coordinates for exploration of large datasets“. In: *Proceedings of the conference on Visualization'99: celebrating ten years*. IEEE Computer Society Press. 1999, S. 43–50 (zitiert auf S. 27).

[GRPF16] H. Gruendl, P. Riehmann, Y. Pausch, B. Froehlich. „Time-Series Plots Integrated in Parallel-Coordinates Displays“. In: *Computer Graphics Forum*. Bd. 35. 3. Wiley Online Library. 2016, S. 321–330 (zitiert auf S. 30).

[GS09] J. R. Goodall, M. Sowul. „VIAssist: Visual analytics for cyber defense“. In: *Technologies for Homeland Security, 2009. HST'09. IEEE Conference on*. IEEE. 2009, S. 143–150 (zitiert auf S. 30, 31).

[ID87] A. Inselberg, B. Dimsdale. „Parallel coordinates for visualizing multi-dimensional geometry“. In: *Computer Graphics 1987*. Springer, 1987, S. 25–44 (zitiert auf S. 27).

[Ign18] Ignite Infobright DB. *Ignite InfobrightDB*. 2018. URL: http://www.ignitetech.com/ (zitiert auf S. 58).

[Ins85] A. Inselberg. „The plane with parallel coordinates“. In: *The visual computer* 1.2 (1985), S. 69–91 (zitiert auf S. 27).

[Isa18] IsarNet Software Solutions GmbH. *IsarFlow*. 2018. URL: https://isarflow.de/ (besucht am 03. 09. 2018) (zitiert auf S. 20, 21).

[JME10] W. Javed, B. McDonnel, N. Elmqvist. „Graphical perception of multiple time series“. In: *IEEE Transactions on Visualization & Computer Graphics* 6 (2010), S. 927–934 (zitiert auf S. 52).

[KBH06] R. Kosara, F. Bendix, H. Hauser. „Parallel sets: Interactive exploration and visual analysis of categorical data“. In: *IEEE transactions on visualization and computer graphics* 12.4 (2006), S. 558–568 (zitiert auf S. 28, 47).

[KKEM10] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann. „Mastering the information age: solving problems with visual analytics“. In: *Eurographics*. Bd. 2. 2010, S. 5 (zitiert auf S. 22, 23).

[Kle09] F. Klein. *Clusteranalyse*. 2009. URL: http://www-m9.ma.tum.de/material/felix-klein/clustering/ (zitiert auf S. 24).

[KMSS06] D. A. Keim, F. Mansmann, J. Schneidewind, T. Schreck. „Monitoring network traffic with radial traffic analyzer“. In: *Visual Analytics Science And Technology, 2006 IEEE Symposium On*. IEEE. 2006, S. 123–128 (zitiert auf S. 31).

[Kos10] R. Kosara. „Turning a table into a tree: Growing parallel sets into a purposeful project“. In: *Beautiful Visualization: Looking at Data through the Eyes of Experts, Steele J., Iliinsky N.,(Eds.). O'Reilly* (2010), S. 193–204 (zitiert auf S. 28, 47, 48, 65, 66).

[Kru83] J. B. Kruskall. „The symmetric time warping algorithm: From continuous to discrete“. In: *Time warps, string edits and macromolecules* (1983) (zitiert auf S. 25).

[LYL04] K. Lakkaraju, W. Yurcik, A. J. Lee. „NVisionIP: netflow visualizations of system state for security situational awareness“. In: *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM. 2004, S. 65–72 (zitiert auf S. 30).

[Mac86] J. Mackinlay. „Automating the design of graphical presentations of relational information“. In: *Acm Transactions On Graphics (Tog)* 5.2 (1986), S. 110–141 (zitiert auf S. 22).

[MFKN09] F. Mansmann, F. Fischer, D. A. Keim, S. C. North. „Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations“. In: *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*. ACM. 2009, S. 3 (zitiert auf S. 31).

[MKN+07] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, D. Sheleheda. „Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats“. In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), S. 1105–1112 (zitiert auf S. 31).

[MMK08] F. Mansman, L. Meier, D. A. Keim. „Visualization of host behavior for network security“. In: *VizSEC 2007*. Springer, 2008, S. 187–202 (zitiert auf S. 31).

[MV06] F. Mansmann, S. Vinnik. „Interactive exploration of data traffic with hierarchical network maps“. In: *IEEE transactions on visualization and computer graphics* 12.6 (2006), S. 1440–1449 (zitiert auf S. 31).

[NCA+13] T. Nunnally, P. Chi, K. Abdullah, A. S. Uluagac, J. A. Copeland, R. Beyah. „P3D: A parallel 3D coordinate visualization for advanced network scans“. In: *Communications (ICC), 2013 IEEE International Conference on*. IEEE. 2013, S. 2052–2057 (zitiert auf S. 31).

[Piv18] Pivotal Software, Inc. *Spring*. 2018. URL: https://spring.io/ (zitiert auf S. 58).

[REE+09] G. Robertson, D. Ebert, S. Eick, D. Keim, K. Joy. „Scale and complexity in visual analytics“. In: *Information Visualization* 8.4 (2009), S. 247–253 (zitiert auf S. 79).

[RHF05] P. Riehmann, M. Hanfler, B. Froehlich. „Interactive sankey diagrams“. In: *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*. IEEE. 2005, S. 233–240 (zitiert auf S. 28).

[SC07] S. Salvador, P. Chan. „Toward accurate dynamic time warping in linear time and space“. In: *Intelligent Data Analysis* 11.5 (2007), S. 561–580 (zitiert auf S. 25).

[Shn96] B. Shneiderman. „The eyes have it: A task by data type taxonomy for information visualizations“. In: *Visual Languages, 1996. Proceedings., IEEE Symposium on*. IEEE. 1996, S. 336–343 (zitiert auf S. 22).

[SLS+13] L. Shi, Q. Liao, X. Sun, Y. Chen, C. Lin. „Scalable network traffic visualization using compressed graphs“. In: *Big Data, 2013 IEEE International Conference on*. IEEE. 2013, S. 606–612 (zitiert auf S. 31).

[SMW00] J. Solka, D. Marchette, B. Wallet. „Statistical visualization methods in intrusion detection“. In: *Computing Science and Statistics* 32 (2000), S. 16–24 (zitiert auf S. 31).

[SRJ+17] J. Sansen, G. Richer, T. Jourde, F. Lalanne, D. Auber, R. Bourqui. „Visual Exploration of Large Multidimensional Data Using Parallel Coordinates on Big Data Infrastructure“. In: *Informatics*. Bd. 4. 3. Multidisciplinary Digital Publishing Institute. 2017, S. 21 (zitiert auf S. 27).

[TAS04] C. Tominski, J. Abello, H. Schumann. „Axes-based Visualizations with Radial Layouts“. In: *Proceedings of the 2004 ACM Symposium on Applied Computing*. SAC '04. Nicosia, Cyprus: ACM, 2004, S. 1242–1247. ISBN: 1-58113-812-1. DOI: 10.1145/967900.968153. URL: http://doi.acm.org/10.1145/967900.968153 (zitiert auf S. 30).

[TAS05] C. Tominski, J. Abello, H. Schumann. „3D Axes-Based Visualizations for Time Series Data, Poster Paper“. In: *IEEE Information Visualization 2005 (InfoVis 2005)* (2005) (zitiert auf S. 30).

[TG83] E. Tufte, P. Graves-Morris. *The visual display of quantitative information*. 1983 (zitiert auf S. 22, 28).

[VV99] J. J. Van Wijk, E. R. Van Selow. „Cluster and calendar based visualization of time series data“. In: *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*. IEEE. 1999, S. 4–9 (zitiert auf S. 28, 29, 44, 63, 77, 79).

[WAM01] M. Weber, M. Alexa, W. Müller. „Visualizing time-series on spirals.“ In: *Infovis*. Bd. 1. 2001, S. 7–14 (zitiert auf S. 28).

[WWK00] M. Q. Wang Baldonado, A. Woodruff, A. Kuchinsky. „Guidelines for using multiple views in information visualization“. In: *Proceedings of the working conference on Advanced visual interfaces*. ACM. 2000, S. 110–119 (zitiert auf S. 41).

[You18] E. You. *Vue.js*. 2018. URL: https://vuejs.org/ (zitiert auf S. 58).

[YYS05] X. Yin, W. Yurcik, A. Slagell. „VisFlowConnect-IP: an animated link analysis tool for visualizing netflows“. In: *FLOCON-Network Flow Analysis Workshop (Network Flow Analysis for Security Situational Awareness)*. 2005 (zitiert auf S. 31).

Alle URLs wurden zuletzt am 03. 09. 2018 geprüft.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift