HOCHSCHULE
MITTWEIDA
University of Applied Sciences

BACHELORARBEIT

Frau
Emilia Cora Weinhold

Entwurf und prototypische Implementierung einer Teilnehmeridentifikation basierend auf Digital Fingerprinting für LabCon

Mittweida, Dezember 2023

Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Entwurf und prototypische Implementierung einer Teilnehmeridentifikation basierend auf Digital Fingerprinting für LabCon

Autorin:
Emilia Cora Weinhold

Studiengang:
Medieninformatik und Interaktives Entertainment

Seminargruppe:
MI20w3-B

Erstprüfer:
Prof. Dr.-Ing. Alexander Lampe

Zweitprüfer:
Dr. rer. nat. Rico Beier-Grunwald

Einreichung:
Mittweida, 11.12.2023

Verteidigung/Bewertung:
Mittweida, 2023

Faculty of **Applied Computer Sciences and Biosciences**

BACHELOR THESIS

Design and prototypical implementation of user identification based on digital fingerprinting for LabCon

Author:
Emilia Cora Weinhold

Course of Study:
Media Informatics and Interactive Entertainment

Seminar Group:
MI20w3-B

First Examiner:
Prof. Dr.-Ing. Alexander Lampe

Second Examiner:
Dr. rer. nat. Rico Beier-Grunwald

Submission:
Mittweida, 11.12.2023

Defense/Evaluation:
Mittweida, 2023

Bibliografische Beschreibung

Weinhold, Emilia Cora:
Entwurf und prototypische Implementierung einer Teilnehmeridentifikation basierend auf Digital Fingerprinting für LabCon . – 2023. – 38 S.
Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Referat

Die vorliegende Bachelorarbeit widmet sich dem Entwurf und der prototypischen Implementierung einer Teilnehmeridentifikationsmethode für LabCon, ein Nutzerverwaltungssystem für Online-Praktika. Der Fokus liegt auf dem Einsatz von Digital Fingerprinting, einer Technologie, die durch die Extraktion charakteristischer Merkmale der Browser und der Geräte von Nutzern eine eindeutige Identifikation ermöglicht.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung und Motivation

Im Zuge der stetig fortschreitenden Digitalisierung der Lehre gibt es eine Vielzahl von Managementsystemen für digitale Kurse. Jedoch können diese Systeme schnell zu kostenintensiv werden, insbesondere wenn eine Remote-Interaktion mit realer Laborhardware erforderlich ist. In diesem Kontext entstand LabCon als webbasiertes Remote-Lab-System (RLS), das speziell für Online-Laborpraktika im Bereich der Ingenieurwissenschaften entwickelt wurde. Es dient der Organisation und Verwaltung dieser Online-Praktika für Studenten. In dem System können sich Studenten über ihre Hochschul-Accounts anmelden, Gruppen bilden und Zeitslots buchen, in welchen sie bestimmte Laborpraktika durchführen können. Während dieser Zeitslots haben die Studenten einen Online-Zugriff auf die physischen Laborgeräte in ihrer Hochschule über eine integrierte Schnittstelle im RLS.

LabCon wurde an der Hochschule Mittweida entwickelt und wird dort in der Fakultät Ingenieurwissenschaften in dem Studiengang ”Elektrotechnik - Automation” eingesetzt. Dort führen die Studenten unter anderem Laborpraktika im Modul ”Signale und Systeme” durch, in welchen sie verschiedene Messungen an Oszilloskopen durchführen. LabCon ermöglicht dabei eine Remote-Interaktion mit den Oszilloskopen sowie eine Nutzerverwaltung, um einen geordneten Online-Zugriff auf die Laborgeräte zu gewährleisten, damit die Studenten ihre Praktika zu jeder Zeit und von jedem Ort aus absolvieren können.

In dieser Arbeit soll ein Prototyp entworfen und implementiert werden, welcher die Teilnehmer an den Online-Laborpraktika, die als Prüfungsvorleistung für Studenten vorgesehen sind, identifizieren kann. Ein zentrales Anliegen ist dabei das Aufdecken von potenziellen Betrugsfällen, insbesondere die Anmeldung einer Person mit den Anmeldedaten einer anderen, um zu verhindern, dass eine Person die Praktika auch für andere durchführt. Um diesen Herausforderungen zu begegnen, wird die digitale Teilnehmeridentifikation mittels Digital Fingerprinting, genauer gesagt des Browser-Fingerprintings, als Technologie umgesetzt. Im Zuge der Anmeldung werden digitale Fingerabdrücke generiert und abgespeichert. Diese dienen dazu, später zu überprüfen und zu identifizieren, ob sich ein Nutzer als eine andere Person ausgibt. In dieser Arbeit werden keine Verfahren vorgestellt, um sich vor Browser-Fingerprinting zu schützen. Ebenso werden rechtliche und ethische Aspekte des Digital Fingerprintings nicht betrachtet.

In dem zweiten Kapitel, dem Grundlagenteil, wird zunächst das webbasierte Remote-Lab-System LabCon vorgestellt. Anschließend werden die Konzepte der digitalen Identifikation und des Digital Fingerprintings beleuchtet, gefolgt von der Ermittlung der Einzigartigkeit einzelner Merkmale, welche beim Digital Fingerprinting extrahiert werden und der Bedeutung von Cookies, Local Storage und IndexedDB. In Kapitel 3 wird darauf aufbauend das Konzept zur Umsetzung der Digital-Fingerprinting-Technologie vorgestellt und ergänzend die Architektur des Prototyps erläutert. Das vierte Kapitel widmet sich der prototypischen Implementierung, wobei die Sammlung, Speicherung und Auswertung der Fingerprint-Daten im Detail beschrieben werden. Die Evaluation der implementierten Technologie erfolgt im fünften Kapitel, in dem die Fingerprints und Attribute, die generierten IDs und die Performanz des Systems analysiert werden. Schließlich bietet das sechste Kapitel eine Zusammenfassung der gewonnenen Erkenntnisse, reflektiert die wichtigsten Aspekte der Arbeit und gibt einen Ausblick auf mögliche Weiterentwicklungen.

2 Grundlagen

Die grundlegende Herausforderung in digitalen Umgebungen besteht darin, individuelle Entitäten präzise zu identifizieren und zu unterscheiden. Dies ist entscheidend für viele Anwendungen, angefangen von Sicherheitssystemen bis hin zu personalisierten Diensten. In diesem Kapitel wird zunächst das webbasierte Remote-Lab-System (RLS) LabCon eingeführt, welches den Remote-Zugriff auf Laborgeräte und die Verwaltung von Online-Praktika ermöglicht. Anschließend wird die digitale Identifikation als Ergebnis der individuellen Zuordnung von Entitäten in digitalen Räumen und das Konzept des Digital Fingerprintings als Methode zur präzisen Identifikation von Entitäten beleuchtet. Darauf folgend wird die Entropie zur Ermittlung der Einzigartigkeit einzelner Merkmale betrachtet, um die Informationsmenge, die ein Browser-Fingerprinting-Algorithmus bietet, einschätzen zu können. Abschließend werden die Technologien Cookies, Local Storage und IndexedDB vorgestellt, die in der Webentwicklung für die Datenspeicherung und -verwaltung eine zentrale Rolle spielen.

2.1 LabCon

Remote web-basierter Unterricht bietet Flexibilität und kann durch Open-Source- oder kommerzielle Learning Management Systeme (LMS) sowie Tools wie Moodle, Adobe Captivate, MS Teams usw. für die Mehrheit der Vorlesungsformate erleichtert werden. Jedoch ist für Studenten der Ingenieurwissenschaften die Labortätigkeit mit physischen Geräten ein wesentlicher Bestandteil ihrer Ausbildung, der ebenfalls online verfügbar sein muss. Während auf einiger Labor-Hardware über Web-Benutzeroberflächen zugegriffen werden kann, ist es oft schwierig, sie in Standard-LMS zu integrieren und sicherzustellen, dass nur autorisierte Benutzer die Geräte bedienen. Es gibt kommerzielle Lösungen von den Geräteherstellern [1], aber diese können teuer sein und sich schwierig in die bestehende IT-Infrastruktur von Universitäten integrieren lassen. Um diesen Herausforderungen zu begegnen, wurde ein Open-Source webbasiertes Remote-Lab-System (RLS) mit wichtigen LMS-Funktionen und der Flexibilität zur späteren Integration von Web-Benutzeroberflächen verschiedener Geräte namens LabCon entwickelt. Das System wurde entwickelt, um eine kostengünstige Alternative zu bieten, welche die vorhandene IT-Infrastruktur der Hochschule nutzt, um die Effizienz zu maximieren. [2]

Die RLS-Softwarearchitektur wird in Abbildung 2.1 dargestellt. Das Managementmodul umfasst eine Benutzeroberfläche für die Benutzerinteraktion, die über eine API mit dem Backend kommuniziert, was die Logik des Systems und die Datenbankkommunikation kapselt. Das Modul für die Hardwarezugänglichkeit besteht aus Proxy-Servern für HTTP- und WebSocket-Anfragen, die Zugang zur Laborhardware bieten. Ein Reverse-Proxy-Server wird verwendet, um Sicherheitsfunktionen wie Verbindungskodierung und Autorisierung über das Single-Sign-On-System Shibboleth der Universität hinzuzufügen. Die Auswahl einer geeigneten Netzwerkkonfiguration ist entscheidend, um zu verhindern, dass Benutzer das System durch direkten Zugriff umgehen. [2]

LabCon bietet für Studenten die Funktion, Zeitslots zu buchen, um dann Zugriff auf die physischen Messgeräte der Hochschule zu erhalten. Dozenten können verschiedene Rollen, Gruppen, Messgeräte und Buchungen verwalten. Diese Online-Praktika stellen für die Studenten eine Prüfungsvorleistung dar. Um das Wissen der Studenten und ihre Ergebnisse aus dem Praktikum besser abzufragen,

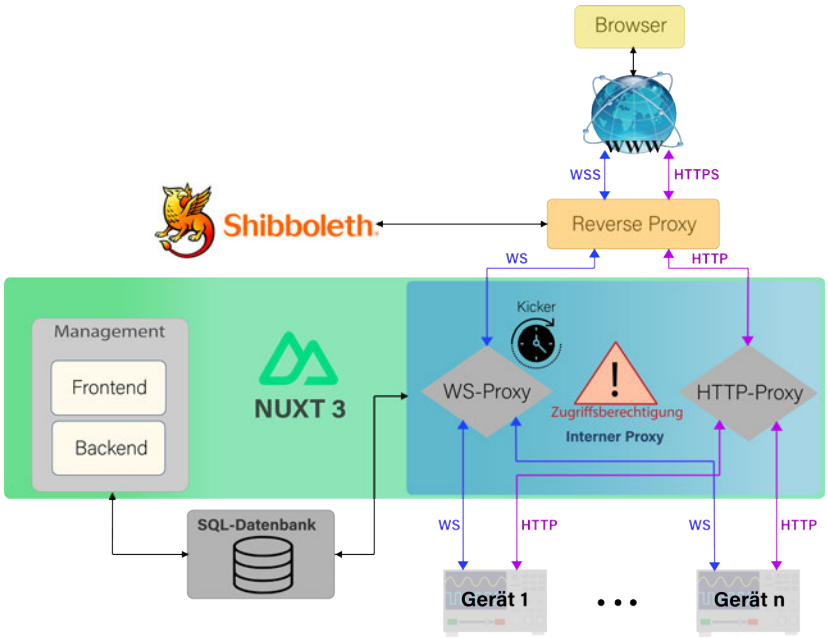

Abbildung 2.1: Systemarchitektur von LabCon [2]

ist für LabCon eine Erweiterung für die Abgabe von Dokumenten und die Beantwortung einiger Fragen im Anschluss an das Praktikum geplant. Daher ist eine Nutzeridentifikation erforderlich, um zu überprüfen, ob sich jemand als eine andere Person ausgibt und das Praktikum für diese andere Person absolviert, einschließlich der Testfragen im Anschluss.

2.2 Digitale Identifikation

Digitale Identifikation bezieht sich auf die eindeutige Zuordnung und Darstellung von individuellen Entitäten oder Benutzern in einer digitalen Umgebung. Dies geschieht durch die Verwendung von digitalen Merkmalen oder Eigenschaften, die einer bestimmten Person oder einem Gerät zugeordnet werden. Typischerweise verwendet man in einer digitalen Umgebung eine Kombination aus Benutzernamen oder E-Mail-Adresse und einem Passwort als individuelles Merkmal. Darüber hinaus können auch Apps zur Freigabe von Transaktionen, Chipkarten, SMS, Hardware- bzw. Software-Token oder biometrische Daten als zweiter Faktor für eine eindeutige Identifikation dienen. [3]

Im Kontext dieser Arbeit ist die digitale Identifikation von zentraler Bedeutung, da sie das Ergebnis für die Teilnehmeridentifikation mittels Digital Fingerprinting darstellt. Durch die Zuweisung einer oder mehrerer einzigartiger, digitaler Identitäten zu jedem Teilnehmer wird eine zuverlässige Identifikation und Unterscheidung ermöglicht, was wiederum essentiell ist für Anwendungen wie Sicherheitssysteme, Zugriffskontrollen, Echtzeitüberwachung und weitere.

Die digitale Identifikation setzt voraus, dass bestimmte digitale Merkmale oder Eigenschaften eines Teilnehmers eindeutig sind. Diese Merkmale sollten zudem möglichst schwer zu fälschen sein. Digital Fingerprinting nutzt diese eindeutigen Eigenschaften, um individuelle Profile zu erstellen und eine zuverlässige Identifikation zu gewährleisten. Ein Teilnehmer kann dabei mehrere digitale Identitäten besitzen, da ihm durch eine Anmeldung über verschiedene Geräte mit unterschiedlichen Eigen-

schaften mehrere Profile unter seinem Nutzernamen angelegt werden. Im weiteren Verlauf dieser Arbeit wird detailliert untersucht, wie Digital Fingerprinting als Werkzeug für die digitale Identifikation eingesetzt werden kann.

2.3 Digital Fingerprinting

Als wegweisende Technologie zur eindeutigen Identifikation von Benutzern bedient sich das Digital Fingerprinting verschiedener Methoden zur Extraktion charakteristischer Merkmale der digitalen Entitäten. Beispielsweise gibt es eine neuartige Technik, die die Verfolgungsdauer von fingerprintbasierten Tracking-Methoden signifikant verlängern kann, namens DRAWNAPART. Diese GPU-Fingerprinting-Technik identifiziert ein Gerät anhand der einzigartigen Eigenschaften seines GPU-Stacks. DRAWNAPART nutzt spezifische Variationen in der Geschwindigkeit der verschiedenen Ausführungseinheiten, die einen GPU-Stack bilden, als zuverlässige und robuste Gerätesignatur. Diese kann mithilfe von JavaScript gesammelt werden. [4]

Eine weitere Methode innerhalb des Digital Fingerprintings ist das Browser-Fingerprinting. Die Entwicklungen im modernen Web, insbesondere durch die Einführung von HTML5 und CSS3, haben zu einer reichhaltigeren und dynamischeren Webumgebung geführt, die eine Vielzahl von Geräten, von Laptops über Smartphones bis hin zu Tablets, unterstützt. Zudem können auch viele unterschiedliche Browser, mit unterschiedlichen Versionen und Einstellungen verwendet werden. Diese zunehmende Diversität bildet die Grundlage für das Browser-Fingerprinting, eine viel genutzte Identifikationstechnik, die darauf abzielt, eine umfassende Liste von Browser- und Geräteeigenschaften auf verschiedenen Ebenen des Systems zu sammeln. Die Wurzeln des Browser-Fingerprintings reichen bis zu den Anfängen des Webs zurück, wodurch es zu einer Technik geworden ist, die sich nicht mit einem einfachen Patch beheben lässt. Seit den Anfängen des Webs teilen Clients, also die Nutzer, die eine Webseite aufrufen, und Server, über den die Webseite gehostet wird, gerätespezifische Informationen, um die Benutzererfahrung zu verbessern. [5]

Beim Browser-Fingerprinting werden vielfältige Informationen über den Webbrowser eines Nutzers gesammelt, darunter beispielsweise das Betriebssystem, die Bildschirmauflösung, installierte Plugins und andere browserbezogene oder gerätespezifische Eigenschaften. Diese Informationen dienen als Grundlage für die Erstellung eines individuellen Fingerabdrucks, der als einzigartige Identifikation des Browsers/Gerätes und somit des Nutzers, der sich angemeldet hat, fungiert.

Im Kontext des Browser-Fingerprinting lassen sich zwei grundlegende Ansätze unterscheiden: aktives und passives Fingerprinting, wie in Abbildung 2.2 dargestellt ist. Diese Ansätze repräsentieren unterschiedliche Methoden der Datensammlung. Einen Überblick über die Daten, die bei beiden Ansätzen gesammelt werden könnten, gibt Abbildung 2.3.

Beim passiven Browser-Fingerprinting erfolgt die Datenerfassung, ohne dass auf der Client-Seite ein spezielles Programm oder Skript gestartet werden muss. Stattdessen werden bestimmte Steuerdaten vom Client zum Server übermittelt, und zwar standardmäßig mit den Kopfdaten der IP-Pakete. Diese Informationen, darunter z.B. die IP-Adresse, der genutzte Port oder der zu den HTTP-Kopfdaten gehörende User-Agent werden zwangsläufig an den Webserver bei jedem Aufruf einer Internetressource gesendet. [6]

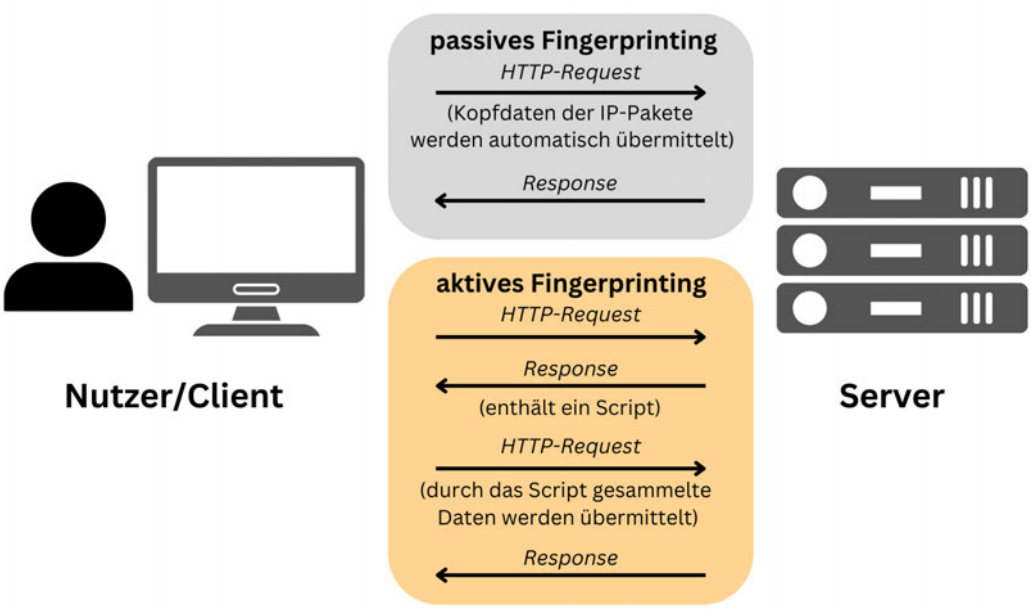

Abbildung 2.2: aktives/passives Browser-Fingerprinting

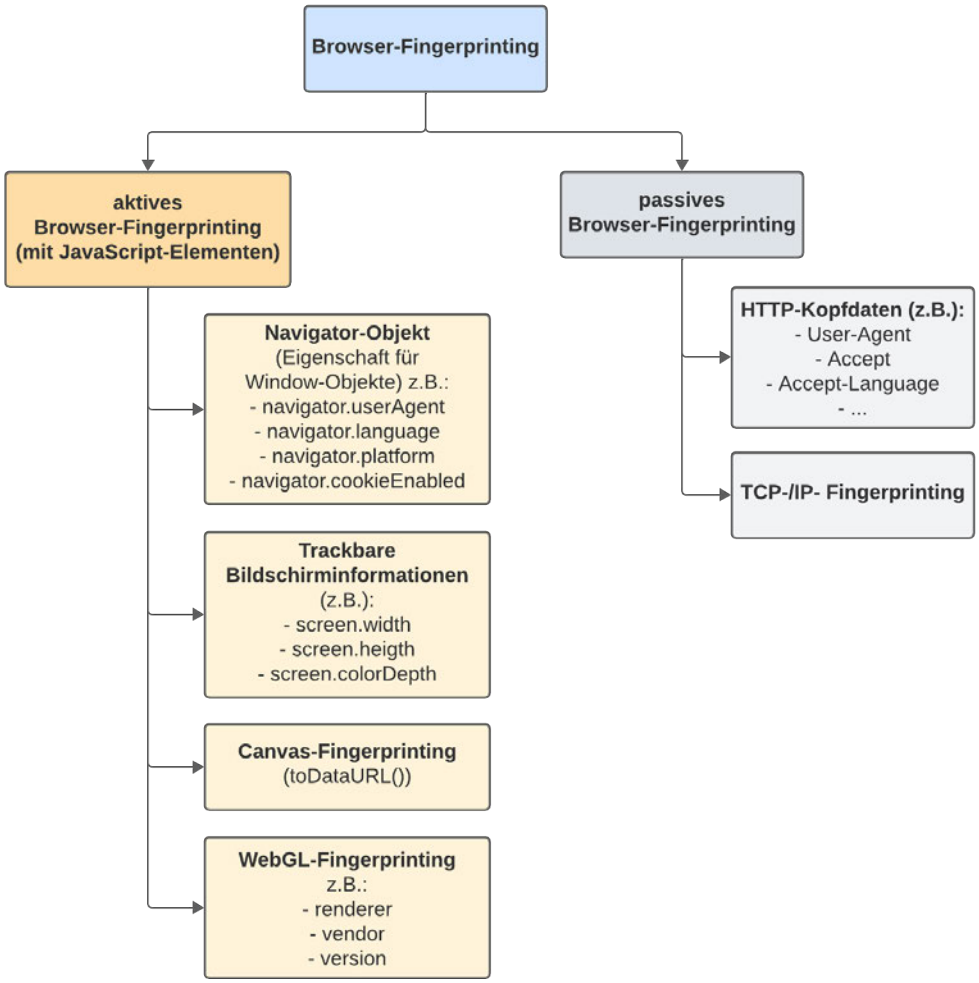

Abbildung 2.3: Browser-Fingerprinting Kategorien

Im Gegensatz dazu erfordert das aktive Fingerprinting eine gezielte Abfrage bestimmter Informationen, die nicht automatisch bei einem einfachen Aufruf einer Webressource übermittelt werden. Dieser Ansatz setzt eine aktive Anfrage auf der Client-Seite voraus, beispielsweise durch den Einsatz von JavaScript-Anwendungen. Beispielsweise können über das JavaScript Navigator-Objekt vertiefte Informationen über den Browser, Angaben zum Betriebssystem sowie installierte Plugins oder die im Browser eingestellte Sprache gewonnen werden. Zusätzliche Informationen betreffen beispielsweise detaillierte Daten über den Bildschirm des Nutzers, einschließlich Breite, Höhe und Auflösung, welche über das JavaScript Screen-Objekt gesammelt werden können. [6]

Ein spezifischer Ansatz des aktiven Browser-Fingerprinting ist das Canvas-Fingerprinting. Diese Methode nutzt die einzigartigen Eigenschaften der Canvas-API in HTML5, um Informationen über ein Benutzersystem zu sammeln und zu analysieren. Im Kern des Canvas-Fingerprintings steht die Erkenntnis, dass verschiedene Browser und Betriebssysteme Texte und Grafiken auf unterschiedliche Weise rendern. Diese Unterschiede, obwohl oft subtil, sind signifikant genug, um als Teil eines Digital Fingerprints verwendet zu werden. Interessanterweise zeigen sich die Unterschiede vor allem in der Art und Weise, wie Text und Grafiken dargestellt werden. Ein markantes Beispiel hierfür ist die Textkernung, die über verschiedene Plattformen hinweg variiert und zu unterschiedlichen Textlängen führt. Diese Variationen sind nicht nur auf die Kernung beschränkt, sondern können auch in anderen Aspekten wie Anti-Aliasing oder Subpixel-Hinting auftreten. Solche feinen Unterschiede in den Rendering-Engines können für präzisere und aussagekräftigere Fingerabdrücke genutzt werden. [7]

Die Praxis des Canvas-Fingerprintings umfasst das Rendern von Text auf einem Canvas-Element, gefolgt von einer detaillierten Untersuchung der erzeugten Pixel. Durch diese Analyse können einzigartige Muster identifiziert werden, die Rückschlüsse auf das verwendete System zulassen. In einigen Experimenten wurden beispielsweise kurze Sätze in der Schriftart Arial gerendert, um die Unterschiede in der Darstellung zu erfassen. Diese Methode ist nicht nur auf Text beschränkt, sondern kann auch andere grafische Elemente umfassen, wodurch die Vielfalt und Genauigkeit der gesammelten Fingerabdrücke erhöht wird. Die Stärke des Canvas-Fingerprintings liegt in seiner Fähigkeit, subtile Unterschiede zwischen verschiedenen Font-Handling-Stacks aufzudecken. Es hat sich gezeigt, dass Faktoren wie das Betriebssystem, die Browser-Version, die Grafikkarte, installierte Schriftarten, Subpixel-Hinting und Anti-Aliasing alle eine Rolle bei der Erzeugung des endgültigen Fingerprints spielen. Um diese Rendering-Informationen zu extrahieren wird die ToDataURL-Methode der Canvas-API verwendet, um die Pixelinformationen in einem DataURL-Format zu erhalten. [7]

Eine weitere Methode des aktiven Browser-Fingerprintings ist das WebGL-Fingerprinting, welches die WebGL-Grafikbibliothek nutzt, um interaktive 2D- und 3D-Grafiken zu rendern. Ähnlich wie beim Canvas-Fingerprinting werden Grafiken generiert und Informationen über die Grafikkarte extrahiert. [8]

Zudem gibt es auch eine erweiterte Browser-Fingerprinting-Technik, die es ermöglicht, Benutzer nicht nur innerhalb eines einzelnen Browsers, sondern auch über verschiedene Browser hinweg auf demselben Gerät zu verfolgen. Diese Methode nutzt eine Vielzahl von Merkmalen auf Betriebssystem- und Hardware-Ebene, darunter solche von Grafikkarten, CPUs und installierten Skripts. Die Extraktion dieser Merkmale erfolgt durch die Aufforderung an die Browser, Aufgaben auszuführen, die auf den entsprechenden Betriebssystem- und Hardware-Funktionalitäten basieren. [9]

2.4 Ermittlung der Einzigartigkeit einzelner Merkmale

Die Entstehung des Browser-Fingerprintings als Technik zur individuellen Identifikation von Benutzern ist eng mit wegweisenden Studien und Forschungen verbunden. Besonders bedeutend sind hierbei große Studien wie ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12], welche auf umfangreichen Datensätzen beruhen und tiefgehende Analysen durchführten. Diese und viele weitere Studien bildeten die Basis für die moderne Anwendung des Browser-Fingerprintings, indem sie aufzeigen, wie Browser anhand ihrer charakteristischen Eigenschaften eindeutig identifiziert werden können.

Eine wichtige Kennzahl, die bei der Analyse von Fingerprinting-Verfahren von großer Bedeutung ist, ist die Entropie, die unter anderem in diesen Studien ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] verwendet wurde. Die Entropie, ein Konzept aus der Informationstheorie, misst die Unsicherheit oder Überraschung angesichts der Ausgabe eines Algorithmus. Angewendet auf ein Browser-Fingerprinting-Algorithmus $F(x)$, der eine diskrete Wahrscheinlichkeitsdichte $P(f_n), n \in [0, 1, ..., N]$ generiert, kann die ”Selbstinformation” oder ”Überraschung” (I) für eine bestimmte Ausgabe f_n durch die Formel

$$I(F(x) = f_n) = -\log_2(P(f_n)) \tag{2.1}$$

berechnet werden. Hierbei wird die Unsicherheit in Bits gemessen, aufgrund der Wahl von 2 als Logarithmus-Basis. Die Entropie der Verteilung $P(f_n)$ wird durch den erwarteten Wert der Unsicherheit über alle Browser hinweg gegeben:

$$H(F) = -\sum_{n=0}^{N} P(f_n)\log_2(P(f_n)) \tag{2.2}$$

Diese Entropie ermöglicht eine Einschätzung der Informationsmenge, die ein Browser-Fingerprinting-Algorithmus bietet. Auf ein einzelnes Merkmal bzw. Attribut angewendet, beschreibt die Entropie dessen Einzigartigkeit. [10]

Um Datensätze unterschiedlicher Größe in Bezug auf die Entropien der einzelnen Fingerprint-Attribute miteinander zu vergleichen, kann man die ”Normalized Shannon’s entropy” eines Attributes berechnen und für diesen Vergleich verwenden. Diese normalisierte Entropie berechnet sich aus:

$$\frac{H(F)}{H_M} \tag{2.3}$$

H_M repräsentiert das ”Worst-Case-Szenario”, bei dem die Entropie maximal ist und alle Werte eines Merkmals einzigartig sind ($H_M = \log_2(N)$, wobei N die Anzahl der Fingerabdrücke in einem Datensatz ist). Der Vorteil dieser Methode liegt darin, dass sie nicht von der Größe des Datensatzes abhängt, sondern von der Verteilung der Wahrscheinlichkeiten. Die Qualität eines Datensatzes wird hinsichtlich der Einzigartigkeit eines Merkmals quantifiziert, unabhängig von der Anzahl der Fingerabdrücke in einer Datenbank. Auf diese Weise können Datensätze trotz ihrer unterschiedlichen Größen verglichen werden. [5]

Tabelle 2.1 zeigt eine Übersicht über die Attribute, welche die Studien ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] verwendet haben sowie deren Entropie und normalisierte Entropie.

Attribute	**Panopticlick (2010)**		**AmIUnique (2016)**		**Hiding (2018)**	
	Entropy	Norm. entropy	Entropy	Norm. entropy	Entropy	Norm. entropy
User Agent	10.000	0.531	9.779	0.580	7.150	0.341
Accept	-	-	1.383	0.082	0.729	0.035
Content encoding	-	-	1.534	0.091	0.382	0.018
Content language	-	-	5.918	0.351	2.716	0.129
List of plugins	15.400	0.817	11.060	0.656	9.485	0.452
Cookies enabled	0.353	0.019	0.253	0.015	0.000	0.000
Use of local/session storage	-	-	0.405	0.024	0.043	0.002
Timezone	3.040	0.019	3.338	0.198	0.164	0.008
Screen resolution and color depth	4.830	0.256	4.889	0.290	4.847	0.231
List of fonts	13.900	0.738	8.379	0.497	6.904	0.329
List of HTTP headers	-	-	4.198	0.249	1.783	0.085
Platform	-	-	2.310	0.137	1.200	0.057
Do Not Track	-	-	0.944	0.056	1.919	0.091
Canvas	-	-	8.278	0.491	8.546	0.407
WebGL Vendor	-	-	2.141	0.127	2.282	0.109
WebGL Renderer	-	-	3.406	0.202	5.541	0.264
Use of an ad blocker	-	-	0.995	0.059	0.045	0.002
H_M **(worst scenario)**	18.843		16.860		20.980	
Number of FPs	470,161		118,934		2,067,942	

Tabelle 2.1: Browser Attribute, ihre Entropie und ihre normalisierte Entropie von den Studien ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] aus [5]

2.5 Cookies, Local Storage und IndexedDB

Cookies sind kleine Datenelemente, die von einem Server an den Webbrowser eines Benutzers gesendet werden. Diese Datenelemente werden vom Browser gespeichert und bei nachfolgenden Anfragen an denselben Server zurückgesendet. Cookies werden typischerweise verwendet, um festzustellen, ob zwei Anfragen vom selben Browser stammen, beispielsweise um einen Benutzer eingeloggt zu halten. Sie speichern zustandsbezogene Informationen für das zustandslose HTTP-Protokoll. Cookies werden hauptsächlich für drei Zwecke verwendet: Sitzungsverwaltung (Logins, Einkaufswagen, Spielstände), Personalisierung (Benutzereinstellungen, Themen) und Tracking (Aufzeichnung und Analyse des Benutzerverhaltens). Obwohl Cookies einst für die allgemeine clientseitige Speicherung verwendet wurden, werden sie heute aufgrund ihrer Größenbeschränkungen und der Tatsache, dass sie bei jeder HTTP-Anfrage gesendet werden, zunehmend durch modernere Technologien wie die Web Storage API (localStorage und sessionStorage) und IndexedDB ersetzt. [13]

Local Storage, ein Teil der Web Storage API, ermöglicht es Webanwendungen, Daten in Form von Schlüssel-Wert-Paaren im Browser des Benutzers zu speichern. Im Gegensatz zu Cookies werden diese Daten bei keiner Serveranfrage gesendet, weshalb der Local Storage effizienter für die Speicherung größerer Datenmengen ist. Im Vergleich zum Session Storage, der zweite Teil der Web Storage API, dessen Daten nur für die Dauer einer Browsersitzung bestehen bleiben, bleiben die Daten des Local Storage auch erhalten, wenn der Browser geschlossen und wieder geöffnet wird. Die Daten werden ohne Ablaufdatum gespeichert und nur durch JavaScript oder das Löschen des Browser-Caches entfernt. [14]

IndexedDB ist eine Low-Level-API für die clientseitige Speicherung erheblicher Mengen strukturierter Daten, einschließlich Dateien/Blobs. Diese API verwendet Indizes, um leistungsstarke Suchvorgänge dieser Daten zu ermöglichen. Während Web Storage nützlich ist, um kleinere Datenmengen zu speichern, ist es weniger nützlich für die Speicherung größerer Mengen strukturierter Daten. IndexedDB ist ein transaktionales Datenbanksystem, ähnlich einem SQL-basierten Relationalen Datenbank Management System (RDBMS). Im Gegensatz zu SQL-basierten RDBMS, die feste Spaltentabellen verwenden, ist IndexedDB eine JavaScript-basierte objektorientierte Datenbank. IndexedDB ermöglicht es, Objekte zu speichern und abzurufen, die mit einem Schlüssel indiziert sind. [15]

3 Konzept und Design

In diesem Kapitel wird das Konzept der Digital-Fingerprinting-Technologie vorgestellt, das auf drei Hauptstrategien basiert: dem Vergleich von Fingerprints unterschiedlicher Personen, dem Vergleich generierter IDs und der Analyse von Mustern in Anmeldezeiten. Zusätzlich wird die Architektur des entwickelten Prototyps detailliert beschrieben, um das Verständnis der Funktionsweise zu ermöglichen.

3.1 Konzept der Digital-Fingerprinting-Technologie

Das Konzept der Digital-Fingerprinting-Technologie zielt darauf ab, spezifische Anforderungen zu erfüllen, die im Kontext des vorhandenen Remote-Lab-Systems (LabCon) und der Online-Praktika an einer Hochschule entstehen. Die Digital-Fingerprinting-Technologie muss in LabCon integriert werden. Die Studenten müssen sich über ihre Hochschul-Accounts anmelden, um ihr Online-Praktikum abzulegen. Da diese Praktika eine Prüfungsvorleistung darstellen, soll das Digital Fingerprinting dazu dienen, einen möglichen Betrug zu erkennen. Insbesondere soll erkannt werden, wenn sich eine Person mit dem Hochschul-Account einer anderen anmeldet, um das Praktikum für diese Person zu absolvieren. Daher soll bei der Anmeldung für jede Person der Fingerprint gespeichert werden. Zusätzlich sollen generierte IDs in den lokalen Speichermöglichkeiten des Nutzers (Cookies, Local Storage und IndexedDB) abgelegt werden und bei späteren Anmeldungen wieder ausgelesen werden. Ergänzend dazu sollen alle Anmeldezeitpunkte jeder Person erfasst werden. Diese drei Strategien dienen alle dazu, einen möglichen Betrug aufzudecken, wie in Abbildung 3.1 dargestellt ist.

Die erste Strategie, der Vergleich zweier Fingerprints, zielt darauf ab, dass Fingerprint-Daten der Nutzer gesammelt, gespeichert und ausgewertet werden. Ein Teil der für die Fingerprints zu sammelnden Daten wurde unter Berücksichtigung der Methoden von ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] ausgewählt und um weitere Attribute ergänzt. Eine Übersicht der Attribute und deren Entropie dieser drei Studien zeigt Tabelle 2.1. Bei den ausgewählten Attributen für die Fingerprint-Technologie für LabCon handelt es sich hauptsächlich um Attribute des aktiven Browser-Fingerprintings, genauer des Navigator-Objektes und trackbarer Bildschirminformationen (s. Abbildung 2.3). Dabei wurde eine sorgfältige Auswahl von Attributen getroffen, wobei einige Merkmale aus spezifischen Gründen nicht in die Implementierung aufgenommen wurden.

Zunächst wurden Attribute, die typischerweise im passiven Fingerprinting verwendet werden, wie ’Accept’, ’Content-Encoding’ und die ’List of HTTP Headers’, nicht berücksichtigt. Das passive Fingerprinting ist durch die VPN-Verbindung der Hochschule, die benötigt wird, um die Webseite zu öffnen, nur eingeschränkt möglich und wurde daher nicht verwendet. Weiterhin wurden die Verwendung von Local/Session Storage und die Verwendung eines Ad-Blockers ausgelassen, da diese Informationen nicht direkt über das JavaScript Navigator-Objekt abrufbar sind. Außerdem besitzen diese beiden Attribute nur eine niedrige Entropie, wie aus den Studien ”AmIUnique” [11] und ”Hiding in the Crowd” [12] hervorgeht und wurden daher nicht zusätzlich verwendet. Schließlich wurde auch die ’Liste der installierten Schriftarten’ nicht in LabCon integriert. Obwohl dies ein potenziell wertvolles Merkmal für das Fingerprinting darstellt, ist die Implementierung aufgrund des hohen Zeitaufwands

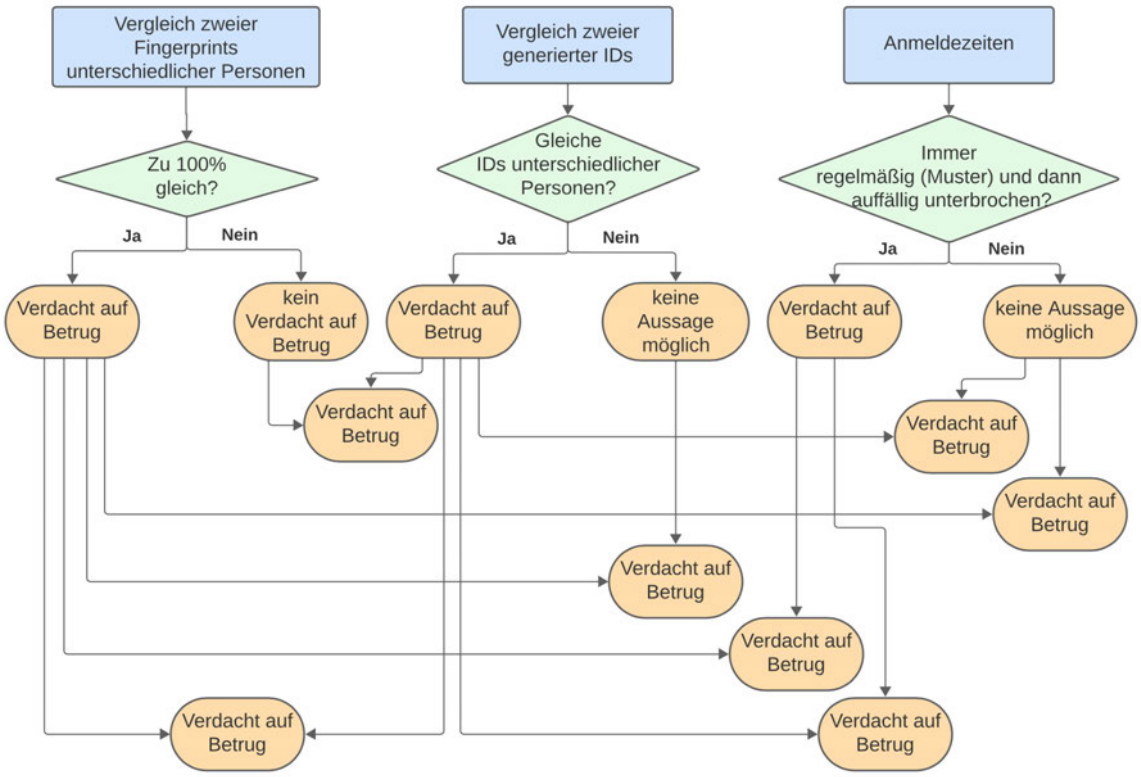

Abbildung 3.1: Übersicht über die drei Strategien zur Aufdeckung eines möglichen Betruges

und der technischen Herausforderungen im gegebenen Rahmen dieser Arbeit nicht praktikabel. Jede mögliche Schriftart müsste einzeln geprüft werden, um festzustellen, ob sie auf dem Gerät des Benutzers installiert ist, was besonders bei seltenen oder ungewöhnlichen Schriftarten schwierig abzudecken ist. Diese Entscheidungen wurden getroffen, um ein effizientes und zielgerichtetes Fingerprinting-System für LabCon zu entwickeln, das sowohl praktikabel als auch im Rahmen der gegebenen Ressourcen umsetzbar ist.

Zusätzlich zu den ausgewählten Attributen aus den drei Studien wurden weitere über das Navigator-Objekt integriert, darunter die CPU-Kerne, eine Schätzung des Arbeitsspeichers eines Gerätes in Gigabyte (memory) und der Online Status, der anzeigt, ob der Browser in einem Zustand ist, in dem er versucht, eine Netzwerkverbindung herzustellen. In einigen Fällen kann der Online Status ”true” (hat eine bestehende Netzwerkverbindung) zurückgeben, auch wenn der Computer tatsächlich offline ist, zum Beispiel wenn er mit einem lokalen Netzwerk verbunden ist, aber keinen Zugang zum Internet hat.

Des Weiteren soll ein Canvas-Fingerprint-Attribut und ein WebGL-Fingerprint-Attribut für jede Person erstellt werden. Das Canvas-Fingerprinting hat in der Studie ”Pixel Perfect” [7] vielversprechende Ergebnisse gezeigt. Ebenso hat die Untersuchung auf ”cover your tracks” [16] ergeben, dass der Informationsgehalt des WebGL-Hashs, welcher aus vielen verschiedenen Attributen der WebGL-Grafikbibliothek generiert wird, um ein Vielfaches höher ist, als der, der herkömmlichen Browser-Attribute. Tabelle 3.1 zeigt eine Übersicht über die ausgewählten Attribute für das Browser-Fingerprinting für LabCon.

Navigator-Objekt	Trackbare Bildschirm-informationen	Canvas	WebGL
user agent	width	toDataURL()	renderer
language	height		vendor
platform	color depth		version
cookie enabled	pixel depth		extensions
timezone			max texture size
do not track			max render buffer size
plugins			max viewport dims
cpu cores			alpha bits
memory			red bits
online status			green bits
			blue bits
			depth bits
			stencil bits
			aliased line width range
			aliased point size range
			samples
			shading language version
			unmasked renderer
			unmasked vendor

Tabelle 3.1: Gesammelte Fingerprint-Attribute

Diese Attribute sollen, nachdem sie gesammelt wurden, für die Speicherung zu Fingerprints zusammengefasst werden. Diese erstellten Fingerprints werden für jeden Nutzer in der Datenbank von LabCon gespeichert. Angesichts der Tatsache, dass ein Nutzer mehrere Geräte besitzen kann oder durch Updates neue Fingerprint-Daten erfasst werden können, kann ein Nutzer auch mehrere Fingerprints besitzen.

Nach der Speicherung der gesammelten Attribute sollen bei der Auswertung die jeweiligen Fingerprint-Daten unterschiedlicher Personen miteinander verglichen werden. Dafür wurden zwei unterschiedliche Auswertungsstrategien für die Berechnung der Ähnlichkeit zweier Fingerprints in Prozent konzipiert, welche in der Implementierung getestet werden sollen. Diese Auswertungsstrategien beinhalten einen Vergleich mit individueller Gewichtung der einzelnen Fingerprint-Attribute. Diese Gewichtung wird einmal mit der Selbstinformation eines einzelnen Attributwertes pro Fingerprint und einmal mit der Entropie eines Attributes berechnet. Da beide Berechnungen die Einzigartigkeit der Attribute bestimmen, sind diese geeignet, um die individuelle Gewichtung der Attribute zu berechnen, welche die Grundlage für die Bestimmung der Ähnlichkeit zweier Fingerprints in Prozent darstellt. Dieser Ansatz zur Berechnung der Gewichtung ist dynamisch und kann auf jeden Fingerprint-Datensatz angewendet werden, im Gegensatz zu einer festen Verteilung/vordefinierten Gewichtung. Beide Auswertungsstrategien und deren Berechnungen der Ähnlichkeit zweier Fingerprints sind anhand eines Beispiel-Datensatzes in Abbildung 3.2 dargestellt.

Bei der ersten Auswertungsstrategie wird für jeden Vergleich zweier Fingerprints für jedes Attribut (in diesem Beispiel-Datensatz userAgent und language) einzeln die Selbstinformation berechnet, wie in Formel 2.1 dargestellt ist. Anschließend wird aus den jeweiligen Selbstinformationen der Durch-

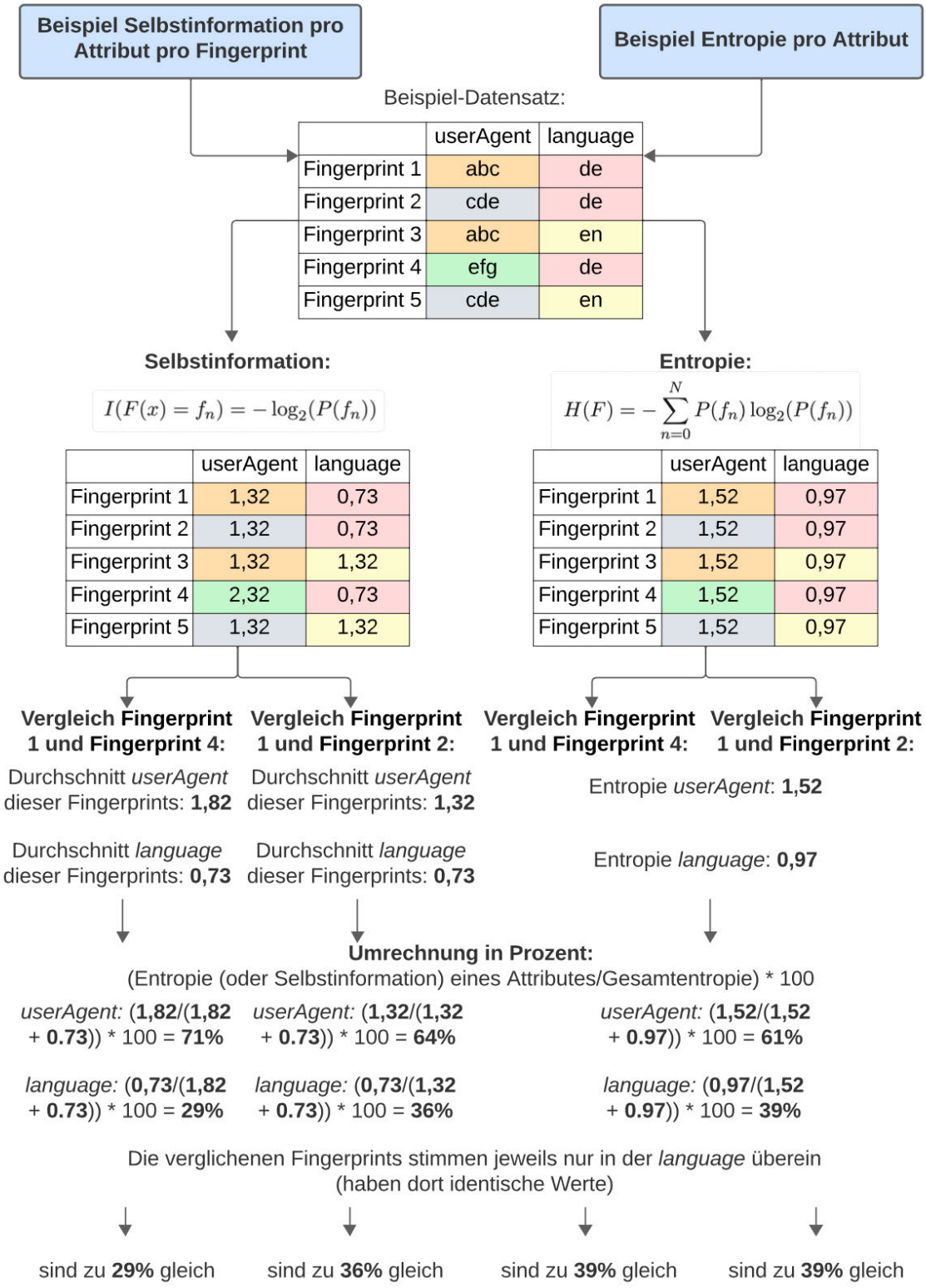

Abbildung 3.2: Beide Auswertungsstrategien zur Berechnung der Ähnlichkeit zweier Fingerprints anhand eines Beispiel-Datensatzes

schnitt für jedes Attribut beider Fingerprints berechnet. So erhält man für jedes Fingerprint-Paar von jedem Attribut eine Zahl, welche die Einzigartigkeit dieser Attribute von diesen beiden Fingerprints beschreibt. Im Beispiel für userAgent von Fingerprint 1 und Fingerprint 4 ist der Wert der Einzigartigkeit gleich 1,82 Bit. Wird nun der Inhalt der jeweiligen Attribute verglichen, dann wird der Durchschnitt als Gewichtung verwendet, in einen Prozentwert umgerechnet und bei Übereinstimmung mehrerer Attribute aufaddiert, so dass letztendlich ein Prozentwert resultiert, welche die Ähnlichkeit zweier Fingerprints angibt. Im Beispiel haben die Fingerprints 1 und 4 nur bei language den gleichen Wert und sind sich daher zu 29% ähnlich. Bei dieser Auswertungsstrategie fällt auf, dass Fingerprints, die in einem Attribut (language) den gleichen Wert besitzen und im anderen Attribut (userAgent) jeweils einen unterschiedlichen Wert aufweisen, unterschiedlich ähnlich sein können, je nachdem, wie einzigartig ihre Attributwerte in Bezug auf alle Attributwerte des Datensatzes sind. So hat Fingerprint 4 einen Wert bei userAgent, welcher einmalig im gesamten Datensatz ist. Daher unterscheidet sich Fingerprint 4 mehr von Fingerprint 1, als Fingerprint 2 von Fingerprint 1, obwohl beide den gleichen Wert bei language haben und einen unterschiedlichen bei userAgent.

Bei der zweiten Auswertungsstrategie wird nicht die Selbstinformation der Attribute einzelner Fingerprints berechnet, sondern die Entropie der Attribute für den gesamten Datensatz, mit der Formel 2.2. Anschließend werden die einzelnen Fingerprints miteinander verglichen. Die vorher berechneten Entropien werden auch hier für die Gewichtung verwendet und in Prozentwerte umgerechnet. Sind mehrere Attribute zweier Fingerprints gleich, dann wird dieser Prozentwert aufaddiert und ergibt am Ende ebenfalls die Ähnlichkeit zweier Fingerprints in Prozent. Im Beispiel aus Abbildung 3.2 stimmen sowohl Fingerprint 1 und 4 als auch Fingerprint 1 und 2 im Attribut language überein und im Attribut userAgent nicht überein. Im Vergleich zur ersten Auswertungsstrategie sind sich beide Fingerprint-Paare gleich ähnlich. Je mehr einzigartige Attributwerte ein Datensatz enthält, desto höher ist die Gewichtung dieses Attributes für alle Fingerprints, weshalb sich alle Fingerprints mehr voneinander unterscheiden, bis auf die, die in jedem Attribut übereinstimmen. Der Unterschied zwischen beiden Auswertungsstrategien besteht darin, dass einzigartige Attributwerte der ersten Strategie nur Auswirkungen auf den aktuellen Vergleich haben, wohingegen bei der zweiten Strategie alle Vergleiche von den einzigartigen Attributwerten beeinflusst werden.

Anhand der Ergebnisse beider Auswertungsstrategien, der Ähnlichkeit zweier Fingerprints in Prozent, lässt sich ableiten, ob es sich möglicherweise um das gleiche Gerät und daher die gleiche Person handelt (bei 100% Ähnlichkeit) oder nicht (bei allen andern Prozentwerten), wie es links in Abbildung 3.1 dargestellt ist. Jedoch lässt sich nicht sicher sagen, dass bei einer Ähnlichkeit von 100% ein Betrug vorliegen muss oder dass bei jeder anderen Ähnlichkeit kein Betrug vorliegt. Beispielsweise könnte ein Student seinen Laptop einem anderen geliehen haben. In diesem Fall wären zwei Fingerprints unterschiedlicher Personen gleich, aber es handelt sich nicht um die gleiche Person, weshalb es kein Betrug wäre. Andererseits könnte ein Student sein Praktikum an einem PC durchführen und gleichzeitig an einem zweiten Gerät dieses Praktikum für eine andere Person durchführen. In diesem Fall würde es sich um zwei unterschiedliche Fingerprints handeln, die zu zwei unterschiedlichen Personen zugeordnet werden, aber in Wahrheit handelt es sich um die gleiche Person, die sich einmal als eine andere ausgibt. Daher ist es das Ziel der Digital-Fingerprinting-Technologie, Dozenten auf Auffälligkeiten hinzuweisen, damit diese selbst entscheiden können, ob und wie sie eingreifen möchten.

Neben diesem Vergleich von Fingerprint-Daten soll in der zweiten Strategie, die in Abbildung 3.1 mittig dargestellt ist, eine einzigartige, generierte ID in Cookies, Local Storage und IndexedDB gespeichert werden. Diese IDs werden aus zwei Teilen generiert: dem Datum in Millisekunden als Hexadezimalzahl, kombiniert mit einer zufällig generierten Hexadezimalzahl, basierend auf den Prinzipien der UUID Versionen 1 und 4 [17]. Bei jeder Anmeldung eines Nutzers wird zunächst überprüft, ob bereits eine ID aus den Speichermöglichkeiten ausgelesen werden konnte. Falls keine ID vorhanden ist, wird eine neue generiert und in Cookies, Local Storage und IndexedDB gesichert. Ist bereits eine ID vorhanden, bleibt diese bestehen, und falls sie in einem dieser Speicher fehlt, weil dieser gelöscht wurde, wird sie dort wieder ergänzt.

Wenn der Nutzer nicht alle drei Speicher leert, bleibt die ID bestehen und es besteht nahezu eine Sicherheit von 100%, dass es sich bei gleichen IDs zweier unterschiedlicher Fingerprints um dasselbe Gerät handelt, außer im sehr seltenen und unwahrscheinlichen Fall, dass für zwei unterschiedliche Geräte exakt die gleiche ID generiert wurde. Gleiche IDs für unterschiedliche Fingerprints, die von unterschiedlichen Personen stammen, erzeugt daher einen Verdacht auf Betrug. Wenn keine gleichen IDs entstehen, lässt sich keine Aussage in Bezug auf einen Betrug tätigen, da die IDs regelmäßig aus allen Speichern gelöscht werden könnten. Beispielsweise würden zwei unterschiedliche IDs vorliegen, wenn sich eine Person mit ihrem Account anmeldet, die Speicher löscht und sich anschließend mit dem Account einer anderen Person anmeldet, um für diese das Praktikum durchzuführen, was als Betrug gilt. Wenn sich jedoch dieselbe Person nach dem Löschen der Speicher nicht für eine andere Person anmeldet, sondern sich wieder mit ihrem eigenen Account anmeldet, würden auch zwei unterschiedliche IDs vorliegen, obwohl es dasselbe Gerät und dieselbe Person ist, was kein Betrug ist.

Die Kombination dieser zweiten Strategie mit der ersten, dem Vergleich zweier Fingerprints, kann den Verdacht auf Betrug verstärken, wenn zwei Fingerprints zu 100% übereinstimmen, was ebenfalls in Abbildung 3.1 dargestellt wird. Wenn die IDs dieser beiden Fingerprints gleich sind, dann verstärken sie den Verdacht auf Betrug, da dann bestätigt ist, dass es sich um dasselbe Gerät handelt, welches zwei unterschiedlichen Personen zugeordnet wird. Dies passiert, wenn sich eine Person nicht nur als sie selbst anmeldet, sondern sich auch als eine andere ausgibt, um das Praktikum für diese andere Person durchzuführen. Es ist jedoch auch möglich, dass eine Person einer anderen ihr Gerät geliehen hat und daher sowohl der gleiche Fingerprint, als auch die gleiche ID für beide Personen gespeichert werden. Alternativ können zwei Fingerprints zu 100% gleich sein, aber deren generierte IDs ungleich, was die IDs in diesem Fall irrelevant macht, da sie sich permanent ändern können. Der Verdacht auf Betrug würde dennoch bestehen. Für den Fall, dass zwei Fingerprints nicht zu 100% gleich sind, aber die generierten IDs übereinstimmen, besteht ebenfalls ein Verdacht auf Betrug, da es sich um dasselbe Gerät handelt. Es könnte eine kleine Änderung im Fingerprint durch ein Browser-Update oder einen anderen Monitor mit einer anderen Bildschirmauflösung entstanden sein. In diesem Fall wäre es sinnvoll, den Prozentwert der Übereinstimmung der beiden Fingerprints zu betrachten. Ein Wert über 90% könnte in diesem Zusammenhang die Annahme einer kleinen Änderung im Fingerprint verstärken. Sind sowohl die Fingerprints als auch deren IDs nicht gleich, besteht kein Verdacht auf Betrug.

Als dritte Strategie zur Aufdeckung möglichen Betrugs werden die Anmeldezeiten betrachtet, was in Abbildung 3.1 rechts dargestellt wird. Diese Zeiten werden bei jeder Anmeldung eines Nutzers erfasst und zusammen mit dessen Fingerprint gespeichert. Durch die Analyse dieser Daten können Regelmäßigkeiten oder Muster in den Anmeldezeiten identifiziert werden. Ein Beispiel hierfür wäre,

wenn sich ein Student regelmäßig jede Woche Freitag um 21:00 Uhr anmeldet. Eine Unterbrechung dieses Musters, beispielsweise wenn sich dieser Student nach sechs Wochen einmal in einer Woche zu ungewöhnlichen Zeiten wie Dienstag um 8:00 Uhr, Mittwoch um 12:00 Uhr und dann wieder Freitag um 21:00 Uhr anmeldet, könnte einen Verdacht auf Betrug nahelegen. Diese unregelmäßigen Anmeldezeiten können darauf hindeuten, dass entweder eine andere Person sich für den Studenten angemeldet hat, um das Praktikum für ihn durchzuführen, oder dass der Student selbst das Praktikum für andere durchgeführt hat. Es könnte jedoch auch sein, dass der Student nur Schwierigkeiten oder Wiederholungsbedarf hatte und daher mehrfach ein Praktikum absolvierte. Liegt keine Regelmäßigkeit in den Anmeldezeiten vor, lässt sich keine konkrete Aussage darüber treffen, ob ein Verdacht auf Betrug vorliegt, da es möglich ist, dass sich die Studenten keine festen Termine für die Durchführung der Praktika einplanen und sich daher zu unregelmäßigen Zeiten anmelden. Die Anmeldezeiten allein betrachtet sind daher nicht besonders aussagekräftig. In Kombination mit den zuvor beschriebenen Strategien, dem Vergleich zweier Fingerprints und den generierten IDs, können sie jedoch den Verdacht auf Betrug verstärken. Dies ist insbesondere der Fall, wenn entweder zwei Fingerprints zu 100% übereinstimmen oder zwei Fingerprints dieselbe generierte ID besitzen, oder wenn beide Fälle eintreten. Unregelmäßigkeiten in sonst sehr regelmäßigen Anmeldezeiten können in solchen Fällen noch verdächtiger wirken. Ein Verdacht auf Betrug bleibt bestehen, selbst wenn bei den anderen Strategien Anzeichen für Betrug vorliegen, aber die Anmeldezeiten regelmäßig bleiben. Dies entsteht beispielsweise dann, wenn ein Student zu seiner üblichen Praktikumszeit das Praktikum für eine andere Person durchführt. Daher ist es wichtig, alle Strategien gemeinsam zu betrachten, um ein umfassendes Bild zu erhalten und möglichen Betrug aufzudecken.

3.2 Architektur des Prototyps

Das UML-Diagramm für das Datenmodell aus Abbildung 3.3 illustriert den Zusammenhang zwischen Personen, Fingerprints und Zeitstempel der Anmeldungen. Jede Person kann mehrere Fingerprints haben, aber ein Fingerprint gehört immer zu einer Person. Jeder Fingerprint besitzt eine eindeutige ID, die ihn identifiziert, selbst wenn inhaltlich identische Fingerprints existieren. Zu jedem Fingerprint können mehrere Zeitstempel zugeordnet sein, wobei jeder Zeitstempel nur zu einem Fingerprint gehört. Zu den Zeitstempeln wird auch die generierte ID aus Cookies, Local Storage und IndexedDB gespeichert.

Abbildung 3.4 zeigt eine Übersicht zur Sammlung, Speicherung und Auswertung der Fingerprint-Daten: Nach der Anmeldung über den Hochschul-Login werden die Fingerprint-Daten direkt im Frontend, auf der Startseite von LabCon, gesammelt. Diese gesammelten Daten werden dann an eine API übergeben, die sie in der Datenbank speichert, sofern der Fingerabdruck bei der angemeldeten Person noch nicht vorhanden ist. Parallel dazu wird die einzigartige, generierte ID in Cookies, Local Storage und IndexedDB abgelegt, falls keine vorherige ID ausgelesen werden konnte. Der Dozent kann dann auf einer gesonderten Seite, die für die Studenten nicht einsehbar ist, die Auswertung starten, indem er eine Prozentzahl angibt, die als Schwellwert dient, zu dem alle Fingerprint-Paare aufgelistet werden, die diesen Ähnlichkeitsschwellwert erreichen oder überschreiten. Außerdem kann er sich die Fingerprints anzeigen lassen, deren generierte IDs identisch sind. Ebenso können die Anmeldezeiten mit den dazugehörigen Fingerprints angezeigt werden, indem der Dozent einen bestimmten Zeitraum auswählt.

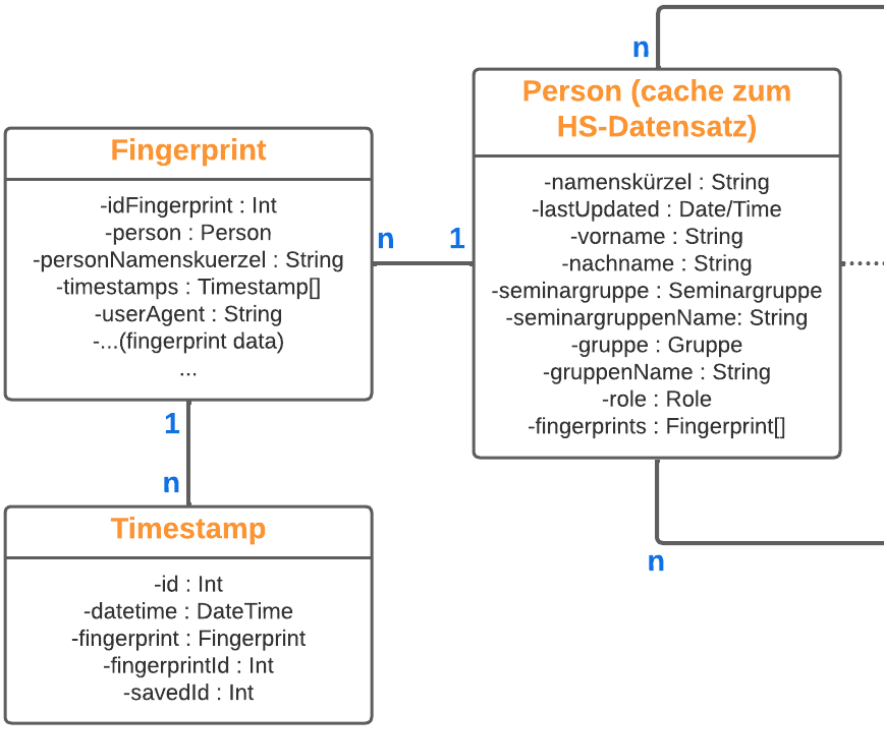

Abbildung 3.3: UML-Diagramm Datenmodell

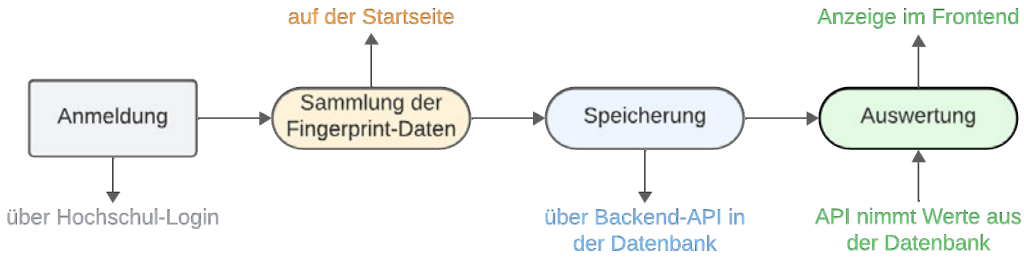

Abbildung 3.4: Übersicht über die Sammlung, Speicherung und Auswertung von Fingerprint-Daten

Der Prototyp für die Digital-Fingerprinting-Technologie wird in die bestehende LabCon-Architektur, die in Abbildung 2.1 dargestellt ist, integriert, als Teil des Managementsystems. Dort fügt er sich sowohl im Frontend, zum Sammeln der Fingerprint-Daten und Anzeigen der Ergebnisse der Auswertung, als auch im Backend, durch weitere API's zum Speichern und Auswerten der Daten, ein.

4 Implementierung

Die Implementierung der Digital-Fingerprint-Technologie innerhalb der bestehenden LabCon-Architektur wird detailliert betrachtet, wobei eine Vielzahl von Tools und Technologien zum Einsatz kommt:

Visual Studio Code dient als zentrale Entwicklungsumgebung, die eine effiziente Implementierung gewährleistet. Nuxt3, ein modernes Framework basierend auf Vue.js, bildet den Rahmen des Projektes. Es ermöglicht eine schnelle Entwicklung von benutzerfreundlichen Oberflächen. Das Frontend wird mithilfe von JavaScript entwickelt, wobei die verwendete PrimeVue-Bibliothek eine Vielzahl an vorgefertigten Komponenten bereitstellt und somit die Gestaltung der Benutzeroberfläche erleichtert. Das Backend wird mit TypeScript entwickelt, wodurch typsichere und wartbare Codebasen ermöglicht werden. Prisma, als leistungsfähiges ORM-Framework, unterstützt die Interaktion mit der Datenbank und erleichtert die Speicherung und Abfrage von Daten.

Eine erfolgreiche Umsetzung der Digital-Fingerprint-Technologie erfordert eine präzise Vorgehensweise in Bezug auf die Sammlung, Speicherung und Auswertung der Fingerprint-Daten. Im Folgenden werden diese Aspekte im Detail beschrieben, um einen umfassenden Einblick in die Implementierung zu geben.

4.1 Sammlung der Fingerprint-Daten

Die Erfassung der digitalen Fingerabdrücke erfolgt im Frontend der Anmeldeseite von LabCon. Hierbei werden verschiedene Methoden angewandt, um eine eindeutige Identifikation des Nutzers zu ermöglichen.

Zuerst wird eine ID verwendet, die generiert wird und in Cookies, Local Storage und IndexedDB im Browser des Nutzers gespeichert ist. Diese ID wird auf ihre Verfügbarkeit überprüft. Sollte diese ID bereits vorhanden sein, wird sie extrahiert und an das Backend zur Speicherung übermittelt. Falls diese ID nicht gefunden wird, generiert das System eine neue eindeutige ID, die anschließend in allen genannten Speicherorten abgelegt wird. Sollte die ID in nur einem oder in zwei der Speicherorte fehlen, weil der Nutzer diesen geleert hat, dann wird sie mit Hilfe der ID aus den anderen Speichermöglichkeiten wiederhergestellt. Dieser Ansatz ermöglicht eine konstante, eindeutige Identifikation des Nutzers, solange dieser seine Speicher nicht regelmäßig leert.

Zusätzlich zur ID-Sammlung werden auch WebGL-Daten erfasst. Hierzu wird ein Canvas-Objekt erstellt und WebGL aktiviert, um spezifische Informationen zur Grafikkarte zu sammeln, die mit WebGL in Verbindung stehen. Basierend auf diesen WebGL-Daten wird ein Hash generiert.

Ein weiteres Canvas-Objekt wird verwendet, um zusätzliche Daten als Canvas-Fingerprint abzurufen. Dabei wird unsichtbarer Content erstellt und die Methode "toDataURL()äufgerufen, um spezifische Informationen zu extrahieren.

Die Datenerfassung umfasst auch Informationen über den verwendeten Browser, die mithilfe des Navigator-Objektes abgerufen werden. Alle gesammelten Daten, einschließlich der ID aus den verschiedenen Speicherorten, WebGL-Daten, Canvas-Daten und Browser-Informationen, werden an

eine API übermittelt, um sie zu speichern. Dabei wird auch ein Zeitstempel erfasst, der die genaue Zeit der Anmeldung angibt. Tabelle 3.1 zeigt eine Übersicht über alle Fingerprint-Daten, die gesammelt werden.

4.2 Speicherung der Fingerprint-Daten

Die Speicherung der Fingerprint-Daten im Backend erfolgt in mehreren Schritten: Die zuvor im Frontend gesammelten Fingerprint-Daten, einschließlich der eindeutigen generierten ID des Nutzers, die in den lokalen Speichermöglichkeiten im Browser gespeichert wurde, WebGL-Daten, Canvas-Daten und Browserinformationen, werden an das Backend übertragen.

Im Backend wird zunächst überprüft, ob bereits Fingerprint-Daten für die angemeldete Person in der Datenbank vorhanden sind. Sind die dort gespeicherten Einträge für eine Person gleich, mit ihren neu gesammelten Fingerprint-Daten, dann erfolgt nur eine Aktualisierung und kein neuer Eintrag dieses Fingerprints. Dabei werden der neue Zeitstempel und die generierte ID aus den lokalen Speichermöglichkeiten des Browsers in der Datenbank abgespeichert. Falls keine bestehenden Daten für die angemeldete Person gefunden werden, die sich mit den neu gesammelten Fingerprint-Daten gleichen, werden alle gesammelten Fingerprint-Daten, der Zeitstempel und die generierte ID in die Datenbank eingefügt.

Die Fingerprint-Daten werden in der ”Fingerprint-Tabelle” in der Datenbank gespeichert (s. Abbildung 3.3). Dabei wird sichergestellt, dass jeder Datensatz in dieser Tabelle eindeutig einer bestimmten Person zugeordnet ist, um eine klare Identifikation zu gewährleisten. Zusätzlich dazu werden die Zeitstempel und die dazugehörigen IDs bei jeder Anmeldung in der ”Timestamp-Tabelle” in der Datenbank gespeichert. Jeder Zeitstempel wird exakt einer spezifischen Fingerprint-Identität (und damit einer bestimmten Person) zugeordnet. Diese Struktur ermöglicht eine präzise zeitliche Nachverfolgung.

4.3 Auswertung der Fingerprint-Daten

In der Implementierung der Fingerprinting-Technologie für LabCon spielen die Schritte zur Auswertung der Fingerprint-Daten eine entscheidende Rolle. Die Dozenten haben die Möglichkeit, im Frontend verschiedene Optionen zur Anpassung der Auswertung auszuwählen. Zunächst können sie über einen Slider eine gewünschte Prozentangabe auswählen, um die angezeigten Fingerprint-Paare zu filtern. Die Prozentangabe dient dazu, die Fingerprint-Paare anzuzeigen, die diesen Ähnlichkeitsschwellwert erreichen oder überschreiten. Dies ermöglicht den Dozenten, die Ähnlichkeitsrate individuell festzulegen. Ein entsprechender Button startet dann die Auswertung basierend auf dieser Einstellung. So können sie überprüfen, welche Fingerprints zu 100% gleich sind, was einen Verdacht auf Betrug erregen würde. Genauso können sie die Fingerprints, die zu über 90% ähnlich sind, beobachten, da bei diesen nur kleine Änderungen/Unterschiede vorliegen, was bedeutet, dass diese zu sehr ähnlichen Geräten gehören, wenn nicht sogar zum gleichen.

Darüber hinaus können die Dozenten die generierten IDs jeder Person auswerten, die jeweils in den lokalen Speichermöglichkeiten im Browser abgelegt wurden. Ein entsprechender Button startet diesen Auswertungsprozess. Dies ermöglicht die Anzeige von IDs, die gleich sind und von unterschiedlichen Fingerabdrücken stammen. Dieser Schritt ist besonders relevant, um gleiche Fingerprint-Paare

gegenzuprüfen. Sind zwei Fingerprints gleich und die dazugehörigen ID's ebenso, dann lässt sich sicher feststellen, dass es sich um das gleiche Gerät handeln muss, was einen Verdacht auf Betrug darstellt. Sind die Fingerprints nicht gleich, aber die IDs schon, so handelt es sich dennoch um das gleiche Gerät, was ebenfalls einen Verdacht auf Betrug erregt.

Ebenso haben die Dozenten die Möglichkeit, einen Zeitraum auszuwählen, um die Zeitstempel und die damit verbundenen Anmeldungen der Personen zu betrachten. Ein weiterer Button startet die Ausgabe innerhalb des gewählten Zeitraums. Dies bietet einen zeitlichen Kontext für die Auswertung der Fingerprint-Daten und ermöglicht es den Dozenten, Anmeldezeiten/-muster in Bezug auf die Fingerprint-Daten und deren IDs zu analysieren, um so einen möglichen Verdacht auf Betrug zu verstärken. Diese Einstellungen im Frontend sind von Bedeutung, da sie den Dozenten die Flexibilität bieten, die Auswertung der Fingerprint-Daten an ihre individuellen Anforderungen anzupassen.

Bei der Implementierung des Auswertungsprozesses der Fingerprint-Daten im Backend wurden mehrere entscheidende Schritte durchgeführt. Dieser Prozess beinhaltet umfassende Vergleiche zwischen jedem Attribut einer Person und jedem Attribut einer anderen Person. Diese Vergleiche bilden die Grundlage für die Berechnung der Ähnlichkeitswerte zwischen den Fingerprints. Um einen Ähnlichkeitswert in Prozent zu ermitteln, werden die Attribute gewichtet. Diese Gewichtung berücksichtigt die Relevanz bzw. die Einzigartigkeit jedes Attributs. Die gewichteten Attribute stellen die Anteile dar, aus denen die finale Prozentzahl für die Ähnlichkeit zwischen zwei Fingerprints berechnet wird. Wenn die Werte eines bestimmten Attributes in den Fingerprints zweier unterschiedlicher Personen übereinstimmen, wird die berechnete Prozentzahl für die Ähnlichkeit addiert. Wenn die Werte nicht übereinstimmen, erfolgt keine Addition. Auf diese Weise entsteht der endgültige Prozentsatz für die Ähnlichkeit.

Die Gewichtung der Attribute basiert auf der Berechnung der Entropie dieser Attribute. Die Entropie ist ein Maß für die Unsicherheit oder Vielfalt der Werte eines Attributs. Je höher die Entropie, desto mehr trägt das Attribut zur Unterscheidung zwischen den Fingerprints bei. In der Implementierung wurden zwei verschiedene Ansätze zur Berechnung der Entropie getestet. Der erste Ansatz zielt auf die Berechnung der Selbstinformation pro Fingerprint pro Attribut ab, wobei dieser Wert für jedes Attribut eines Fingerprints individuell berechnet wird, wie in Abbildung 3.2 links dargestellt ist. Der zweite Ansatz berechnet die Entropie pro Attribut. Diese ist somit für die Attribute aller Fingerprints gleich, wie in Abbildung 3.2 rechts dargestellt ist. In beiden Ansätzen werden die berechneten Werte für die verschiedenen Attribute genutzt, um die Gewichtung der Attribute festzulegen.

Das Ablaufdiagramm aus Abbildung 4.1 beschreibt den Algorithmus des ersten Ansatzes zur Auswertung der Fingerprint-Daten im Backend. Der Algorithmus beginnt damit, Fingerprints einzulesen. In einer äußeren Schleife wird überprüft, ob noch weitere Fingerprints vorhanden sind. Innerhalb dieser Schleife wird eine weitere Schleife gestartet, um jeden Fingerprint mit allen anderen zu vergleichen. Bevor der Vergleich beginnt, wird überprüft, ob die jeweiligen Personen der beiden Fingerprints identisch sind. Wenn ja, wird der Vergleich übersprungen, und die Schleife wird mit dem nächsten Fingerprint fortgesetzt, da nur die Fingerprints unterschiedlicher Personen verglichen werden sollen. Wenn die Fingerprints zu unterschiedlichen Personen gehören, erfolgt die Berechnung der Entropie für jedes Attribut der beiden Fingerprints individuell. Anschließend werden alle Attribute der beiden Fingerprints miteinander verglichen, um die Ähnlichkeit in Prozent zu erhalten. Sollten alle Entropien den Wert null haben, bedeutet dies, dass alle Attribute gleich sind und der Algorithmus gibt diese Information zurück. Wenn die Ähnlichkeit der Fingerprints den vorgegebenen Schwellenwert erreicht

oder überschreitet, werden die Fingerprints als Paar gespeichert. Der Algorithmus kehrt zu dem Punkt zurück, an dem die innere Schleife begonnen hat, um den nächsten Fingerprint mit allen verbleibenden Fingerprints zu vergleichen. Dies wird so lange wiederholt, bis alle Fingerprints miteinander verglichen wurden. Abschließend gibt der Algorithmus alle gespeicherten Fingerprint-Paare zurück.

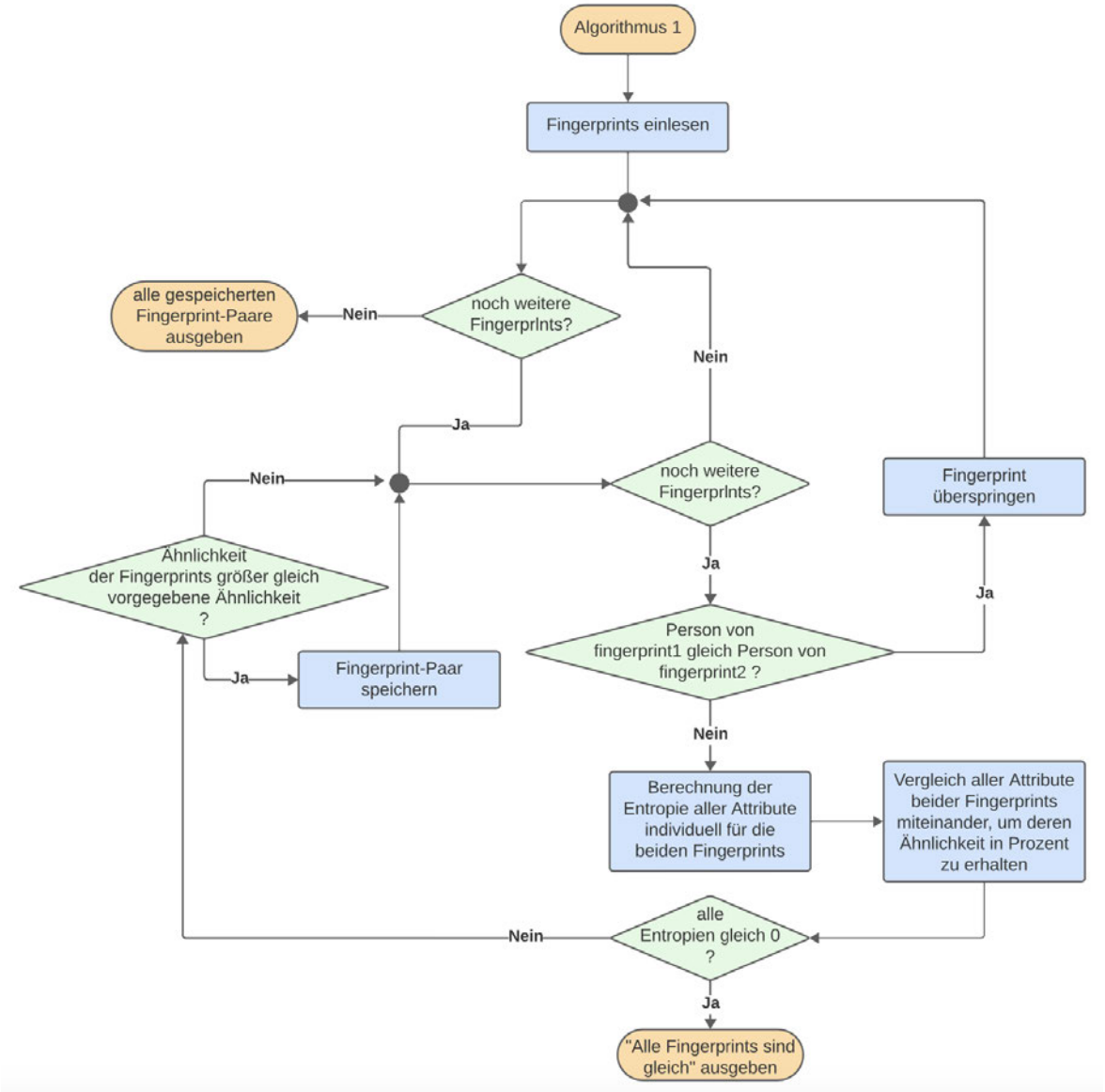

Abbildung 4.1: Ablaufdiagramm zum Algorithmus des ersten Ansatzes

Das Ablaufdiagramm aus Abbildung 4.2 beschreibt den zweiten Ansatz des Algorithmus zur Auswertung der Fingerprint-Daten. Der Algorithmus beginnt damit, Fingerprints einzulesen und die Entropie aller Attribute zu berechnen (nicht mehr individuell pro Fingerprint). In einer äußeren Schleife wird überprüft, ob noch weitere Fingerprints vorhanden sind. Innerhalb dieser Schleife wird eine weitere Schleife gestartet, um jeden Fingerprint mit allen anderen zu vergleichen. Bevor der Vergleich beginnt, wird ebenfalls überprüft, ob die beiden Fingerprints zur selben Person gehören. Wenn ja, wird der Vergleich übersprungen, und die Schleife wird mit dem nächsten Fingerprint fortgesetzt. Wenn die Fingerprints unterschiedlichen Benutzern zugeordnet sind, erfolgt der Vergleich aller Attribute der

beiden Fingerabdrücke miteinander, um die Ähnlichkeit in Prozent zu erhalten. Der restliche Ablauf verhält sich identisch zum Algorithmus des ersten Ansatzes. Wenn alle Fingerprints miteinander verglichen wurden, gibt der Algorithmus alle gespeicherten Fingerprint-Paare zurück.

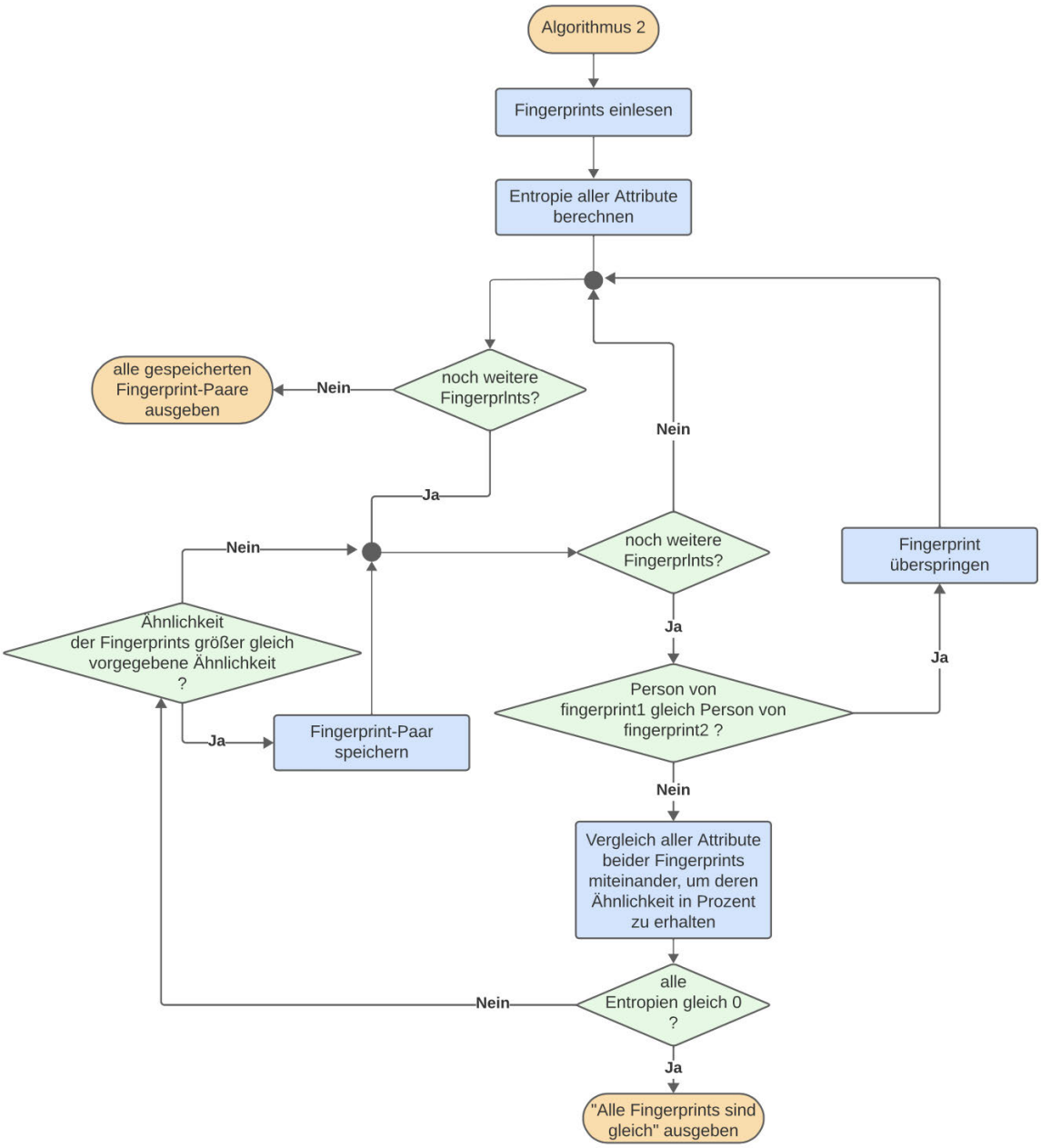

Abbildung 4.2: Ablaufdiagramm zum Algorithmus des zweiten Ansatzes

Beide Ansätze wurden mit einem Test-Datensatz, bestehend aus 20 Fingerprints, auf ihre Ergebnisse und ihre Performanz getestet. Abbildung 4.3 zeigt die Ergebnisse beider Algorithmen. In der unteren rechten Ecke sind die Ergebnisse von Ansatz 1, also die Berechnung der Selbstinformation pro Attribut pro Person, dargestellt und in der oberen linken Ecke sind die von Ansatz 2, also die Berechnung der Entropie über alle Attribute, dargestellt. Das ”x” steht für zwei Fingerprints, die zur gleichen Person gehören und daher nicht miteinander verglichen werden. Die Ergebnisse zeigen, dass beide Algorithmen bei der wichtigsten Zahl übereinstimmen, die 100 Prozent. Die Unterschiede liegen darin, dass Ansatz 1 (untere rechte Ecke) im Schnitt deutlich mehr niedrige Ähnlichkeitspro-

zente aufweist, was bedeutet, dass sich viele Fingerprint-Paare mehr voneinander unterscheiden, als bei Ansatz 2. Dafür sind die einzelnen Werte bei Ansatz 2 breiter gestreut, was auch in Abbildung 4.4 zu sehen ist, welche die Ergebnisse beider Algorithmen in einem Liniendiagramm darstellt.

Fingerprints	A	B	C	D	Q	F	G	S	T	R	K	E	H	J	I	L	N	M	O	P
P	7	7	4	7	4	18	3	12	18	21	21	7	12	21	12	4	22	13	100	x
O	7	7	4	7	4	18	3	12	18	21	21	7	12	21	12	4	22	13	x	100
M	9	9	12	9	12	10	5	4	16	8	8	27	4	8	10	32	11	x	14	14
N	28	28	24	18	24	16	36	13	7	10	10	13	23	10	16	15	x	10	18	18
L	12	12	15	12	15	7	8	7	13	45	45	23	13	45	20	x	12	28	2	2
I	26	26	23	17	23	18	13	22	28	12	12	10	38	12	x	17	6	7	6	6
J	27	27	24	37	24	32	17	32	26	100	100	48	26	x	7	34	10	7	20	20
H	x	47	43	x	43	39	19	42	36	26	26	18	x	13	24	12	10	2	7	7
E	32	32	28	41	28	32	21	32	x	48	48	x	6	34	8	21	15	29	8	8
K	27	27	24	37	24	32	17	32	26	100	x	34	13	100	7	34	10	7	20	20
R	27	27	24	37	x	32	17	32	26	x	100	34	13	100	7	34	10	7	20	20
T	28	38	35	38	35	42	20	87	x	14	14	x	19	14	18	11	3	16	16	16
S	51	51	48	51	48	54	33	x	86	19	19	14	24	19	12	4	7	2	7	7
G	60	60	57	60	57	20	x	20	12	14	14	15	11	14	6	5	38	4	1	1
F	60	60	63	50	63	x	12	40	30	23	23	20	21	23	10	9	8	15	20	20
Q	97	97	100	87	x	52	42	33	22	x	13	14	26	13	13	17	16	13	2	2
D	x	90	87	x	81	35	45	36	25	28	28	26	x	28	9	9	12	6	4	4
C	97	x	x	81	100	52	42	33	22	13	13	14	26	13	13	17	16	13	2	2
B	100	x	x	88	92	43	46	37	25	17	17	16	29	17	15	9	18	6	4	4
A	x	100	92	x	92	43	46	37	25	17	17	16	x	17	15	9	18	6	4	4

Fingerprints

Legende:
0% - 9%
10% - 19%
20% - 29%
30% - 39%
40% - 49%
50% - 59%
60% - 69%
70% - 79%
80% - 89%
90% - 99%
100%

Abbildung 4.3: Ergebnisse beider Verfahren zur Auswertung von 20 Fingerprints

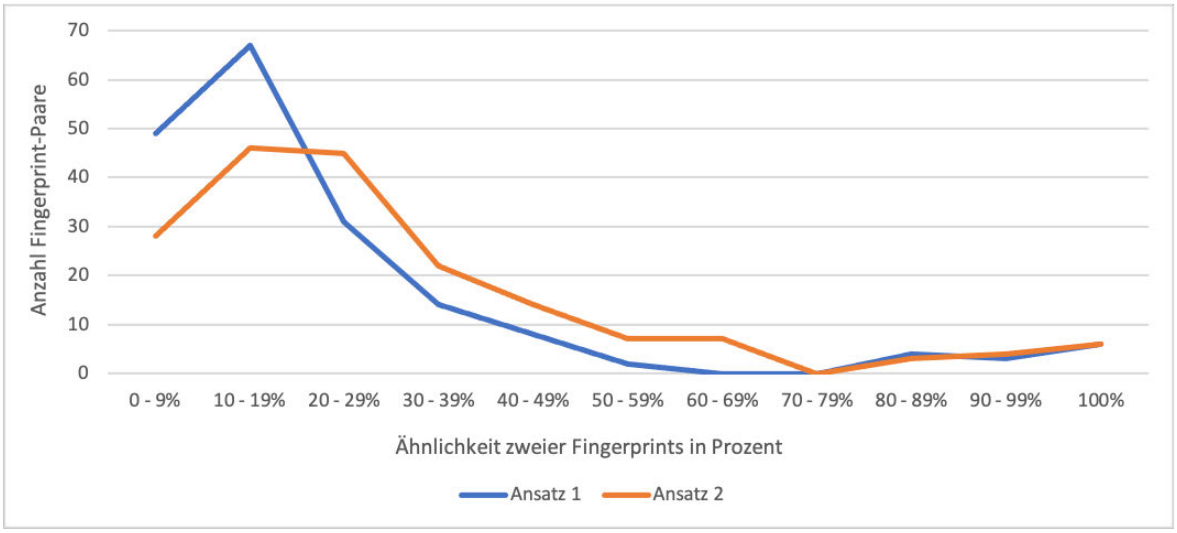

Abbildung 4.4: Gebündelte Ergebnisse beider Verfahren zur Auswertung von 20 Fingerprints

Ein erster lokaler Performanz-Test (nicht über den Server) hat deutliche Unterschiede gezeigt. Die Auswertung von 20 Fingerprints dauerte bei einer vollständigen Berechnung und Auflistung aller Ergebnisse bei Ansatz 1 ca. 2 Minuten, wohingegen Ansatz 2 nach 16 Sekunden abgeschlossen war. Das bedeutet, dass der Ansatz 2, wo die Entropie einmal zu Beginn des Algorithmus für alle Attribute berechnet wird, wesentlich performanter ist, als Ansatz 1. Daher wurde Ansatz 2 für die finale Implementierung gewählt und auf den minimal genaueren Vergleich zweier Fingerprints unterschiedlicher Personen aus Ansatz 1 verzichtet.

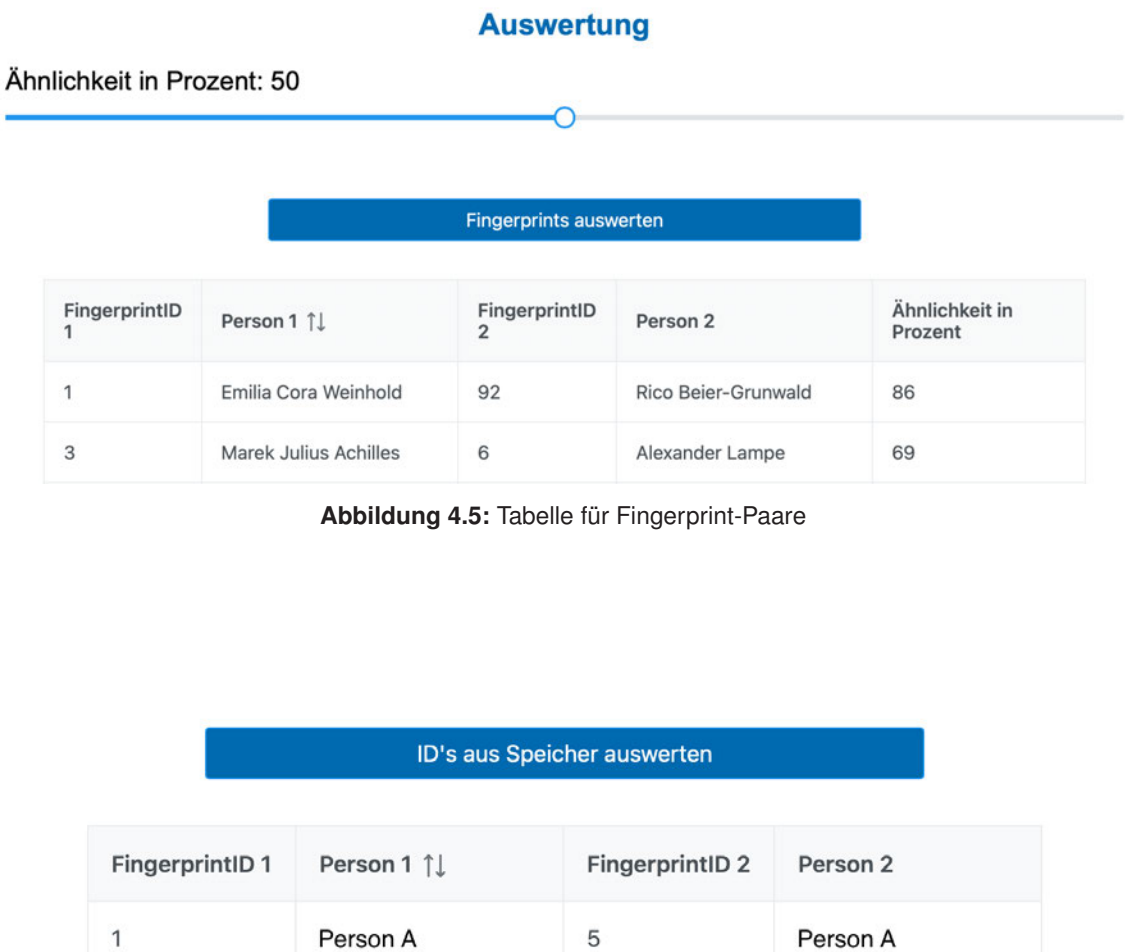

Abbildung 4.5: Tabelle für Fingerprint-Paare

Abbildung 4.6: Tabelle für gleiche ID's der Fingerprints

Nachdem die Auswertung der Fingerprint-Daten im Backend abgeschlossen ist, erfolgt die Präsentation der Ergebnisse im Frontend. Hierbei werden die berechneten Daten aus den APIs abgerufen und in Tabellen übersichtlich dargestellt. Die Benutzeroberfläche im Frontend bietet drei verschiedene Tabellen: In der ersten Tabelle (s. Abbildung 4.5) werden die erfassten FingerprintIDs aufgeführt, zusammen mit den zugehörigen Namen von Person 1 und Person 2. Zusätzlich wird der ermittelte Prozentsatz der Ähnlichkeit zwischen den Fingerprint-Paaren angezeigt. Diese Übersicht ermöglicht es den Dozenten, schnell zu erkennen, wie ähnlich bestimmte Fingerabdrücke zueinander sind. Die zweite Tabelle (s. Abbildung 4.6) dient der Darstellung von Fällen, in denen den Fingerprints einzelner Personen dieselbe generierte ID aus Cookies, Local Storage und IndexedDB zugeordnet wird. Solche Duplikate werden übersichtlich aufgelistet, um zu überprüfen, ob zwei Fingerprints entweder zur gleichen Person gehören oder ob zwei unterschiedliche Personen das gleiche Gerät verwenden, was den Verdacht auf einen Betrug erregt. Die dritte Tabelle (s. Abbildung 4.7) zeigt die Anmeldezeiten in Verbindung mit den jeweiligen Fingerprint-ID's und Personennamen. Dies bietet Einblicke in das Anmeldeverhalten der Benutzer und ermöglicht es, relevante Informationen über zeitliche Muster der Anmeldungen abzurufen. Die Verwendung von diesen Tabellen ermöglicht es den Benutzern, die Daten nach verschiedenen Kriterien zu sortieren, um gezielte Informationen aus den Ergebnissen der Fingerprint-Auswertung zu extrahieren.

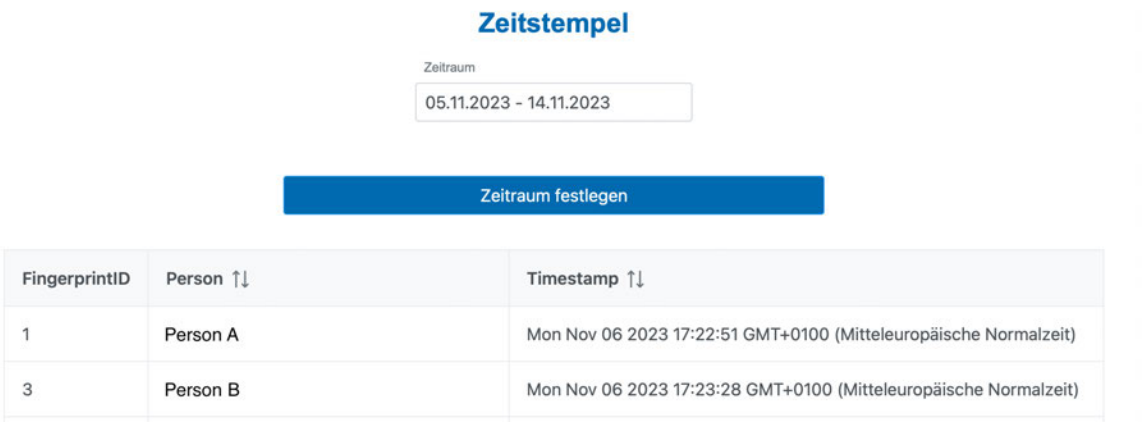

Abbildung 4.7: Tabelle für die Zeitstempel

Die Umsetzung der Auswertung im Backend ist von zentraler Bedeutung, da sie die Hauptfunktion für die Identifikation und Berechnung der Ähnlichkeit zwischen den Fingerprints bildet. Die Verwendung von Entropie und die Gewichtung der Attribute tragen dazu bei, eine präzise und zuverlässige Auswertung der Fingerprint-Daten sicherzustellen.

5 Evaluation

Die Evaluation erfolgte durch die freiwillige Teilnahme der Probanden, die sich mittels ihrer Hochschul-Accounts auf der Webseite anmeldeten. Nach dieser Anmeldung werden auf der Startseite sämtliche Fingerprint-Daten erfasst, wobei der Zugriff auf andere Seiten bewusst unterbunden wurde, da diese nicht relevant für diese Evaluation sind. Als weitere Maßnahme, um die Konstanz der Fingerprint-Daten zu prüfen, wurden die Teilnehmer aufgefordert, sich nach einer Woche erneut mit dem gleichen Gerät anzumelden. Zusätzlich dazu war es gewünscht, dass die Probanden sich mit verschiedenen Geräten oder unterschiedlichen Browsern auf der Webseite anmelden. Insgesamt beteiligten sich 45 Teilnehmer, von denen 72 Fingerprints gesammelt wurden. Nachfolgend werden die gesammelten Fingerprints und Attribute sowie die generierten IDs ausgewertet, bevor abschließend die Performanz des Algorithmus analysiert wird.

5.1 Fingerprints und Attribute

Die Verteilung der Anzahl der Fingerprints pro Person wird in dem Diagramm in Abbildung 5.1 visualisiert. Den größten Anteil stellten 32 Personen dar, von denen jeder einen einzigen Fingerprint aufweist. Es folgen 7 Personen mit jeweils 3 Fingerprints, während bei einer Person mit 7 Fingerprints die höchste Anzahl an Fingerprints gesammelt wurde. Die durchschnittliche Anzahl der Fingerprints pro Person beträgt 1,6, was auf eine geringe Varianz hinweist. Dieses Ergebnis legt nahe, dass der Hauptteil der Probanden sich nur einmal angemeldet hat oder lediglich ein Gerät und einen Browser verwendet hat.

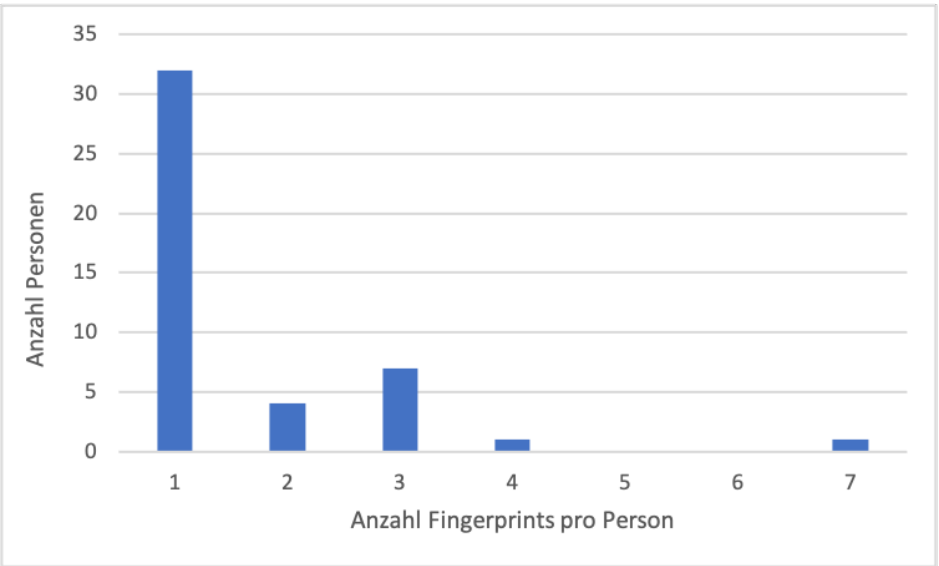

Abbildung 5.1: Anzahl der Fingerprints pro Person

Die Analyse der Fingerprint-Paare hinsichtlich ihrer Ähnlichkeit in Prozent, wie grafisch im Diagramm in Abbildung 5.2 dargestellt, zeigt, dass die Mehrheit der Fingerprint-Paare (893 von 2504) eine Ähnlichkeit im Bereich von 10-19% aufweist. Eine annähernd gleichmäßige Verteilung konnte für die Ähnlichkeiten von 0-9% und 20-29% mit 480 bzw. 498 Fingerprint-Paaren festgestellt werden. Signifikant bei dieser Analyse ist dabei, dass keines der Fingerprint-Paare eine Übereinstimmung

von 100% aufweist. Ebenso gibt es keine Paare mit einer Ähnlichkeit von 90-99%. Im Durchschnitt ähneln sich zwei Fingerprints zu etwa 24%, was impliziert, dass sie zu etwa 76% unterschiedlich sind. Diese Ergebnisse lassen darauf schließen, dass alle Fingerprints als einzigartig betrachtet werden können.

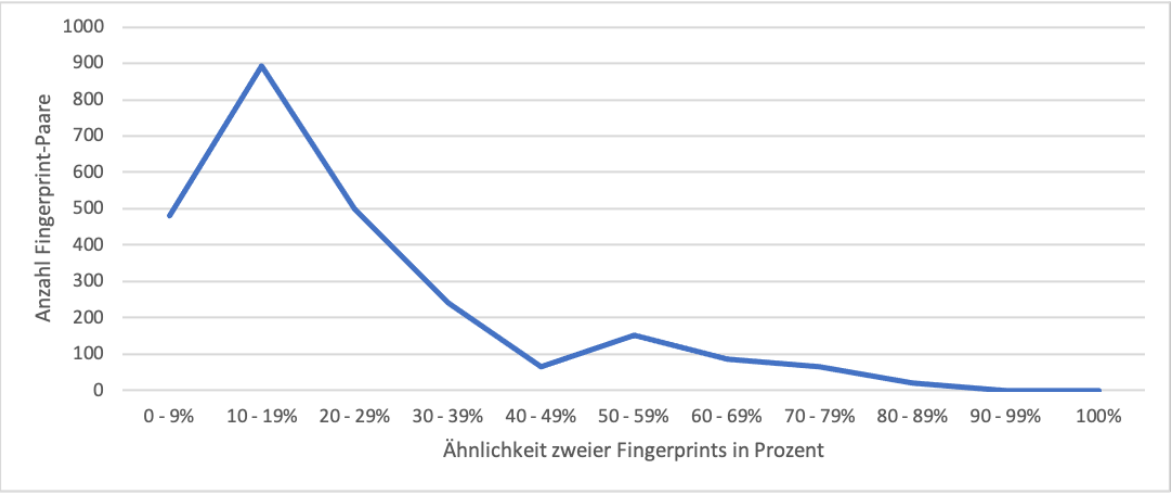

Abbildung 5.2: Anzahl der Fingerprint-Paare geordnet nach deren Ähnlichkeit in Prozent

Die Auswertung der Entropie der einzelnen Attribute, wie sie im Diagramm in Abbildung 5.3 dargestellt ist, liefert Einblicke in die Informationsdichte dieser Attribute. Hervorzuheben sind hierbei insbesondere der WebGL-Hash und die Screen Resolution, die mit Entropiewerten von 3,86 bzw. 3,80 die höchste Informationsdichte aufweisen. Dies bedeutet, dass bei diesen Attributen von den Probanden eine Vielzahl unterschiedlicher Werte erfasst wurde, was zur Einzigartigkeit aller Fingerprints beiträgt. Im Gegensatz dazu zeigen die Attribute Cookies enabled und Online Status eine Entropie von 0, was darauf hindeutet, dass alle Fingerprints bei diesen Attributen den gleichen Wert aufweisen. Somit tragen diese Attribute gar nicht zur Unterscheidung der Fingerprints bei. Insgesamt betrachtet liegt die durchschnittliche Entropie eines Attributs bei 1,7, was auf eine moderate Informationsdichte hindeutet, da die Kombination dieser Attribute das Ergebnis erzielt hat, dass jeder der 72 Fingerprints einzigartig ist.

Das Diagramm 5.4 vergleicht die normalisierten Entropien der Studien ”Panopticlick”[10], ”AmIUnique”[11] und ”Hiding in the Crowd” [12] mit LabCon. Dabei werden nur die gemeinsamen Attribute, die sowohl in den Studien, als auch für LabCon verwendet wurden, betrachtet. Diese Analyse zeigt Erkenntnisse über die Informationsdichte und -vielfalt der gemeinsamen Attribute. Es wird die normalisierte Entropie (siehe Formel 2.3) für den Vergleich zwischen den Studien und LabCon verwendet, da so die Entropien der Attribute miteinander verglichen werden können unabhängig von der Anzahl der gesammelten Fingerprints. LabCon besitzt bei 5 von 8 gemeinsamen Attributen die höchste normalisierte Entropie, was darauf hinweist, dass bei diesen Attributen die meisten unterschiedlichen Werte im Vergleich zu den anderen Studien gesammelt wurden.

Bei allen Attributen außer Plugins und Screen Resolution sind die normalisierten Entropien aller Studien und LabCon annähernd ähnlich, mit einer Standardabweichung von 0,1 oder darunter. Im Kontrast dazu weisen Plugins eine höhere Standardabweichung von 0,3 auf und die Screen Resolution eine Standardabweichung von 0,18. Die niedrige normalisierte Entropie von LabCon bei den Plugins, welche für diese höhere Standardabweichung sorgt, könnte darauf zurückzuführen

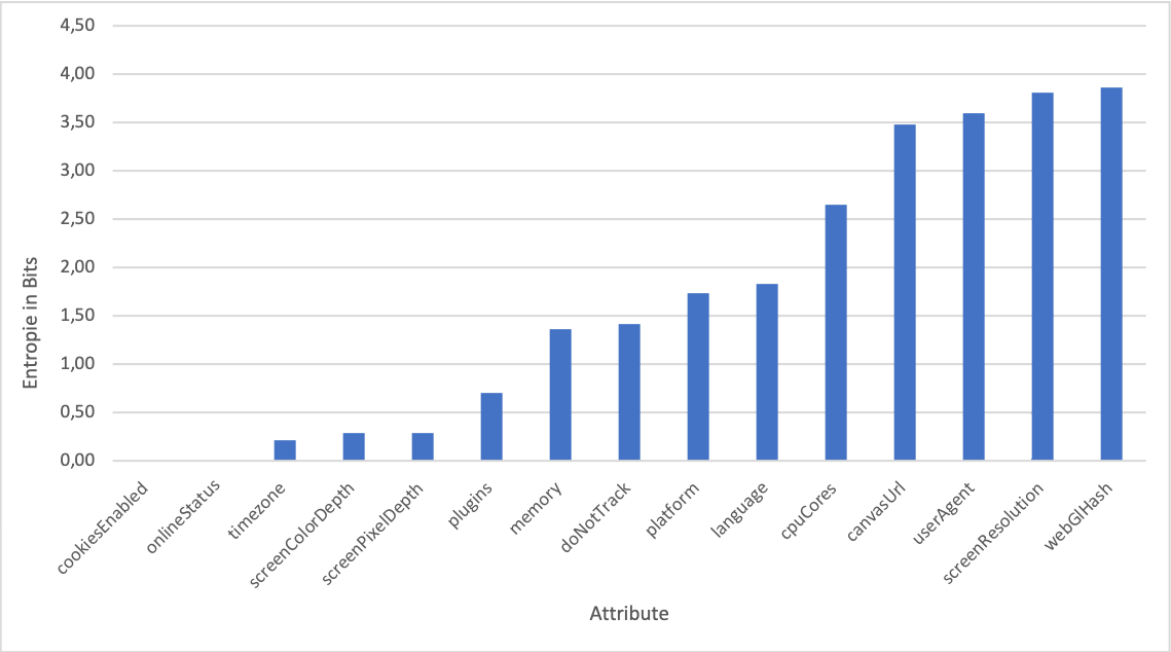

Abbildung 5.3: Entropie der jeweiligen Attribute

sein, dass das Navigator-Objekt für Plugins nur die fünf Standard-Plugins findet und zurückgibt, die die Inline-Ansicht von PDFs ermöglichen. Individuell hinzugefügte Plugins der Teilnehmer werden daher nicht erfasst. Die sehr hohe normalisierte Entropie von LabCon bei der Screen Resolution, die dort ebenfalls die höhere Standardabweichung erzeugt, könnte durch die zunehmende Vielfalt der Technik, einschließlich verschiedener Laptops, Monitore, Handys usw., in den letzten Jahren erklärt werden.

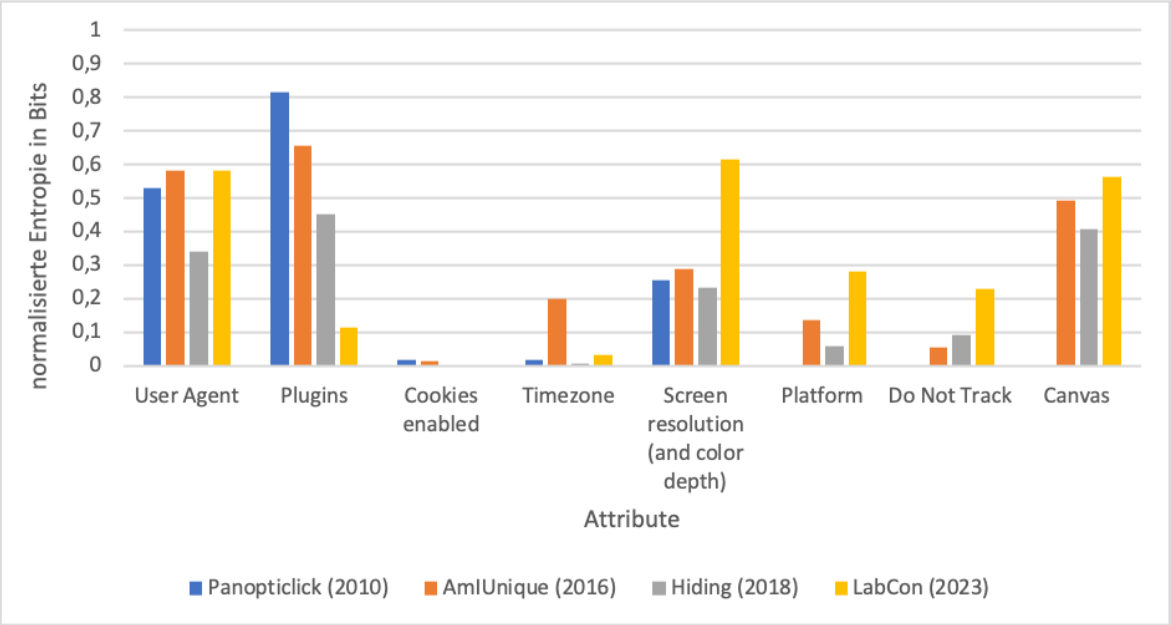

Abbildung 5.4: Normalisierte Entropien der jeweiligen Attribute aus den Studien ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] (s. Tabelle 2.1) und LabCon im Vergleich

Die Diagramme in den Abbildungen A.1 bis A.6 im Anhang A bieten einen detaillierten Einblick in die Anzahl der Fingerprints für die unterschiedlichen Werte der jeweiligen Attribute. Dabei wird insbesondere deutlich, dass die Attribute WebGl-Hash und Screen Resolution in Abbildung A.1 sowie

User Agent und Canvas URL in Abbildung A.2 eine hohe Entropie aufweisen. Dies resultiert aus der Vielzahl von unterschiedlichen Werten (zwischen 21 und 27 Stück) bei einer geringen Anzahl von Fingerprints pro Wert (maximal 24 Stück pro Wert). Diese Vielfalt an einzigartigen Werten trägt zu einer hohen Entropie bei. Im Gegensatz dazu zeigt Abbildung A.6, dass für die Attribute Online Status und Cookies Enabled lediglich ein einziger Wert gesammelt wurde, der von allen 72 Fingerprints angenommen wurde. Dies bedeutet, dass diese Attribute eine Entropie von 0 aufweisen, da alle Fingerprints denselben Wert für diese Attribute aufweisen.

Diagramm 5.5 bietet eine Darstellung der Anzahl der Fingerprints, die sich nach einer Woche entweder nicht verändert haben oder Veränderungen aufweisen. Die Gesamtzahl der Teilnehmer, die sich nach einer Woche erneut angemeldet haben, beträgt 12. Unter diesen sind 8 Fingerprints unverändert geblieben, während sich 4 Fingerprints verändert haben. Dieses Ergebnis ist positiv zu bewerten, da in diesem Fall doppelt so viele Fingerprints nach einer Woche konstant geblieben sind im Vergleich zu den Fingerprints, die sich verändert haben.

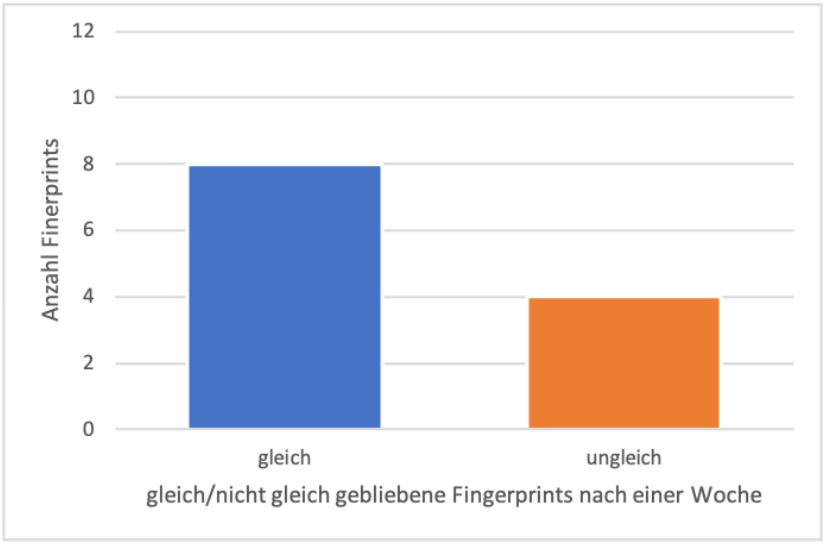

Abbildung 5.5: Anzahl der Fingerprints, die nach einer Woche gleich geblieben/nicht gleich geblieben sind

Die Diagramme aus Abbildung 5.6 und 5.7 veranschaulichen, welche Attribute sich nach einer Woche bei den Teilnehmern verändert haben. Die Analyse unterscheidet zwischen den 4 Teilnehmern, deren Fingerprint nach einer Woche nicht konstant geblieben ist, und den 7 von 8 Personen, die neben ihrem konstanten Fingerprint weitere Fingerprints besitzen.

Für die 4 Personen, deren Fingerprint nicht konstant geblieben ist, sind bei allen 5 Attribute gleich geblieben, während sich insgesamt 10 Attribute verändert haben. Personen H und J zeigen jeweils 8 veränderte Attribute, während es bei Person B nur 3 Attribute sind, die sich geändert haben. Bei allen 4 Personen hat sich das Attribut Screen Resolution verändert.

Bei den 7 Personen mit einem konstanten und zusätzlichen Fingerprints blieben bei allen ebenfalls 5 Attribute gleich, während sich insgesamt 10 Attribute verändert haben. Person E weist dabei 10 veränderte Attribute auf, während es bei Person G 2 Attribute sind. Es hat sich kein Attribut bei allen 7 verändert, aber WebGL-Hash, User Agent und Canvas URL haben sich bei allen Personen außer Person G geändert.

Die jeweils 5 Attribute, die sich bei beiden Betrachtungen geändert haben, sind bis auf eines, was unterschiedlich ist (Timezone und Plugins), die gleichen. Dabei handelt es sich insgesamt um die 6 Attribute mit der geringsten Entropie. Dahingegen sind die Attribute, die sich in beiden Betrachtungen bei den meisten Personen geändert haben, auch die mit der höchsten Entropie.

Die Ergebnisse legen nahe, dass die Attribute, die sich bei den meisten Personen geändert haben, darauf hinweisen, dass bei der erneuten Anmeldung möglicherweise ein anderer Browser oder ein anderes Gerät verwendet wurde. Bezogen auf die Screen Resolution könnte dies beispielsweise bedeuten, dass ein Laptop verwendet wurde, der auch mal an einen externen Monitor angeschlossen wurde.

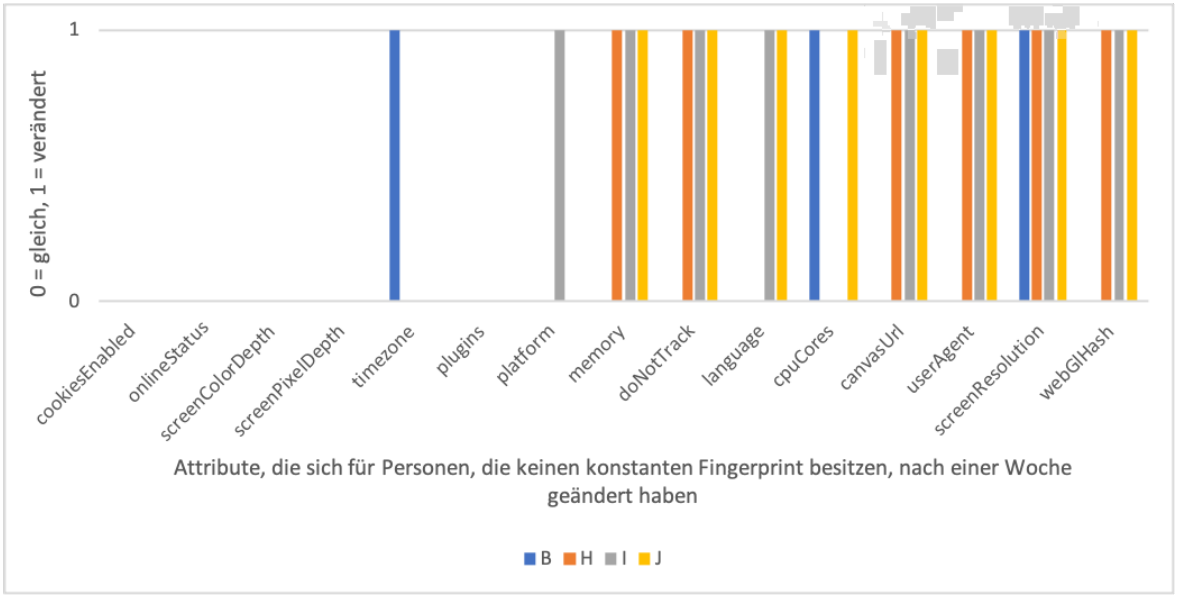

Abbildung 5.6: Attribute, die sich nach einer Woche geändert haben/gleich geblieben sind, für die Personen, die keinen konstanten Fingerprint besitzen

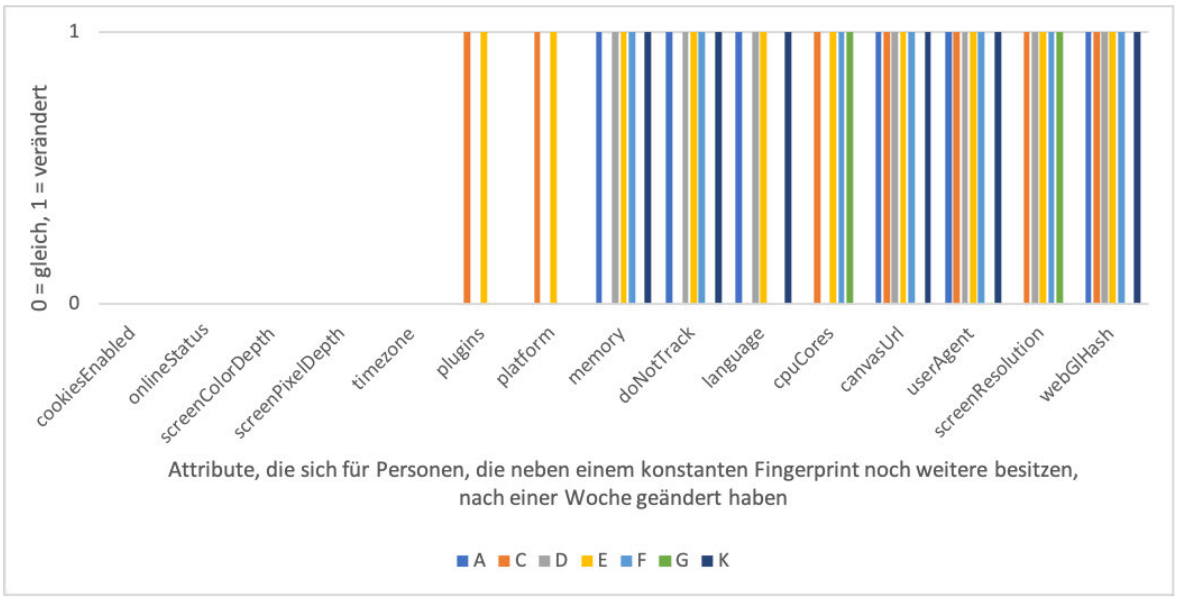

Abbildung 5.7: Attribute, die sich nach einer Woche geändert haben/gleich geblieben sind, für die Personen, die neben einem konstanten Fingerprint noch weitere besitzen

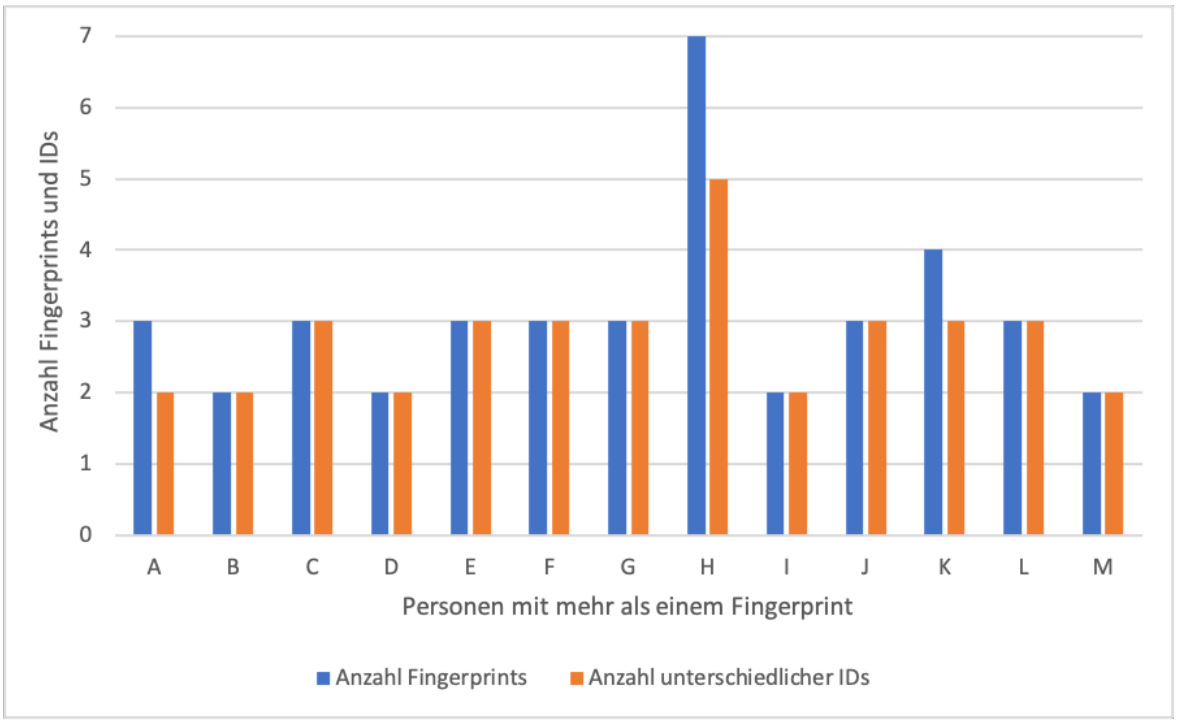

Abbildung 5.8: Die Anzahl an unterschiedlichen Fingerprints und IDs für Personen mit mehr als einem Fingerprints

5.2 Generierte IDs

Die generierten IDs sind die IDs, die für jeden Nutzer in Cookies, Local Storage und IndexedDB gespeichert wurden. Diese Daten wurden erfasst, indem bei jeder Anmeldung die Fingerprint-ID, die aktuelle generierte ID und der Zeitstempel der Anmeldung gespeichert wurden.

Das in Abbildung 5.8 dargestellte Diagramm zeigt die Anzahl an unterschiedlichen Fingerprints und IDs von den Personen, die mehr als einen Fingerprint besitzen. Betrachtet man die Anzahl der IDs dieser 13 Personen, dann zeigt sich, dass alle bis auf drei Personen genau so viele unterschiedliche Fingerprints wie IDs besitzen. Ein neuer Fingerprint kann entstehen, wenn entweder kleine Änderungen in den Browsern oder auf den Geräten der Personen vorgenommen wurden oder wenn unterschiedliche Geräte verwendet wurden. Im ersten Fall kann eine neue ID nur dann generiert werden, wenn die Personen ihre Speichermöglichkeiten gelöscht haben. Im zweiten Fall wird eine neue ID bei der Erstanmeldung über ein neues Gerät generiert. Jedoch kann bei der einmaligen Verwendung eines neuen Gerätes keine konkrete Aussage getroffen werden, ob der Nutzer seine Speicher löscht oder nicht. Bei den drei Personen, die mehr unterschiedliche Fingerprints als IDs besitzen, bedeutet das, dass sie mindestens einen Fingerprint besitzen, der durch eine Änderung im Browser oder auf dem Gerät entstanden ist und sie ihre Speicher nicht gelöscht haben. Beispielsweise sind es bei Person J zwei neue Fingerprints, welche die gleiche vorherige ID behalten haben, also wo die Cookies, der Local Storage oder die IndexedDB nicht gelöscht wurden.

Das Diagramm aus Abbildung 5.9 zeigt eine Zuordnung des Grundes einer neu generierten ID zu diesen 13 Personen. Auf einige Personen, wie beispielsweise auf Person E, treffen dabei beide Gründe unabhängig voneinander zu. Das bedeutet, dass ein Teil der neu generierten IDs zustande kommt, weil ein neues Gerät verwendet wurde und der andere Teil, weil der Speicher gelöscht wurde.

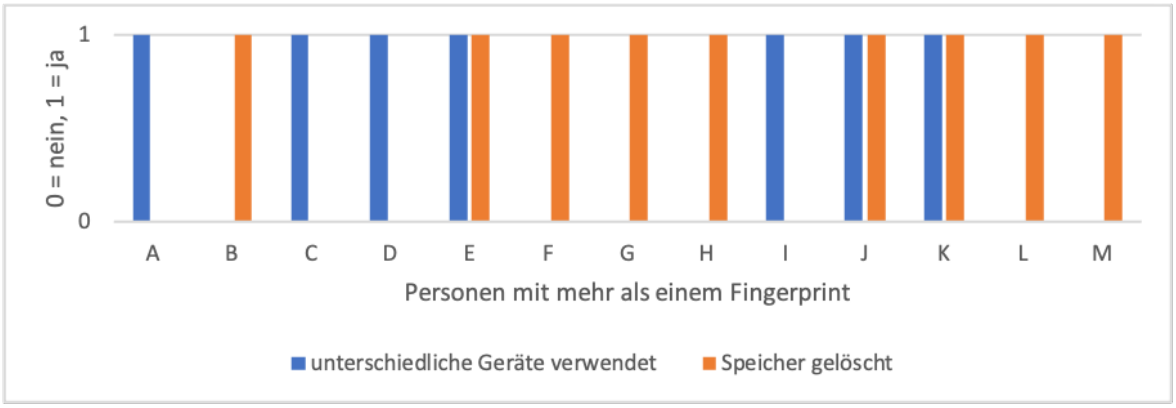

Abbildung 5.9: Personen mit mehr als einem Fingerprint, deren IDs entweder neu generiert wurden, weil die Personen unterschiedliche Geräte verwendet haben oder weil sie ihre Cookies/Local Storage/IndexedDB gelöscht haben

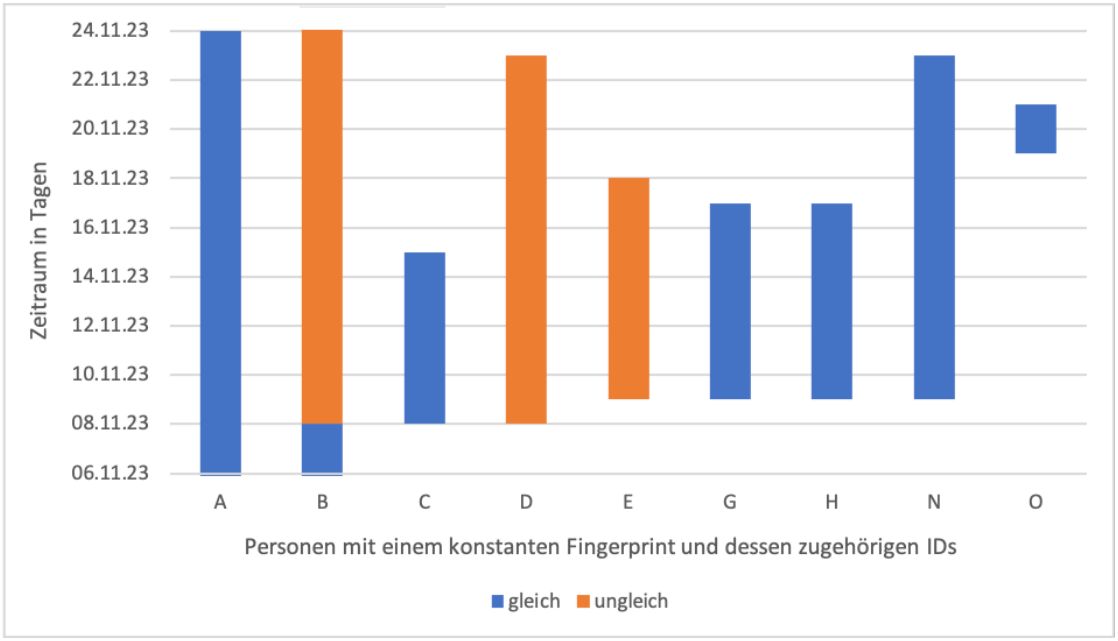

Abbildung 5.10: Die Veränderung generierter IDs, die zu konstanten Fingerprints unterschiedlicher Personen gespeichert wurden, abgebildet auf einen Zeitraum von 18 Tagen

Das Diagramm aus Abbildung 5.10 illustriert die Veränderung der generierten IDs, die konstanten Fingerprints zugeordnet wurden, die zu unterschiedlichen Personen gehören, über einen Zeitraum von 18 Tagen. Dabei sind die Personen mit dem gleichen Buchstaben die gleichen wie aus den beiden Diagrammen zuvor. Die Personen I bis M tauchen nicht auf, da diese keinen konstanten Fingerprint besitzen. Die Personen N und O sind neu hinzugekommen, da die beiden Personen jeweils nur einen Fingerprint besitzen. Betrachtet man diese Abhängigkeit zwischen den drei Diagrammen, so lässt sich beispielsweise erkennen, dass Person A zwei unterschiedliche IDs besitzt, aber 3 Fingerprints (s. Abbildung 5.8). Also ist eine der beiden IDs über zwei Fingerprints konstant geblieben. Die zweite ID ist Aufgrund der Verwendung eines neuen Gerätes entstanden (s. Abbilung 5.9) und der konstante Fingerabdruck über 18 Tage hat immer die gleiche ID (s. Abbildung 5.10), was beides darauf hindeutet, dass diese Person ihre Speichermöglichkeiten nicht löscht. Von den 9 Fingerprints, bei denen mehrere Anmeldezeiten vorliegen und die zu unterschiedlichen Personen gehören, haben 6 eine generierte ID, die konstant geblieben ist. Das bedeutet, dass diese Personen ihre lokalen

Speichermöglichkeiten nicht geleert haben. Die generierte ID von Fingerprint A konnte über den vollen Zeitraum von 18 Tagen beobachtet werden und blieb konstant. Die ID von Fingerprint I wurde über 3 Tage beobachtet und blieb ebenfalls konstant. Dahingegen blieb die ID von Fingerprint B zunächst über 2 Tage konstant und änderte sich dann im Zeitraum von 16 Tagen bis zur letzten Anmeldung am 24.11.2023. Die ID von Fingerprint D änderte sich vom Zeitpunkt der ersten Anmeldung am 08.11.2023 bis zur letzten Anmeldung am 23.11.2023 ebenfalls, was bedeutet, dass die Person innerhalb dieses Zeitraums ihre Speicher gelöscht hat.

Keine dieser generierten IDs einer Person ist identisch mit der einer anderen Person, was beutet, dass kein Verdacht auf Betrug vorliegt. Dies stimmt auch mit den Ergebnissen des Vergleichs zweier Fingerprints aus Abbilung 5.2 überein, da kein Fingerprint-Paar existiert, welches zu 100% gleich ist.

5.3 Performanz

Das Diagramm in Abbildung 5.11 veranschaulicht die Performanz des Algorithmus in Abhängigkeit von dem zuvor eingestellten Ähnlichkeitsschwellwert in Prozent, die angibt, welche Fingerprint-Paare ihrer Ähnlichkeit nach ausgegeben werden sollen und der damit verbundenen Anzahl an ausgegebenen Fingerprint-Paaren. Der Test wurde über den Server durchgeführt und nicht mehr lokal, was zu insgesamt schnelleren Ergebnissen führte. Die orangene Linie repräsentiert die Dauer der Auswertung in Sekunden (rechte Achse), während die blauen Balken die Anzahl der ausgegebenen Fingerprint-Paare zeigen (linke Achse). Die Ergebnisse sind nach der eingestellten Prozentzahl der Ähnlichkeit sortiert.

Bei 0% Ähnlichkeit werden alle Fingerprint-Paare ausgegeben, da sich alle Fingerprints zu 0% oder ähnlicher sind. In diesem Fall sind es insgesamt 2504 Fingerprint-Paare und die Auswertung dauert ca. 18 Sekunden.

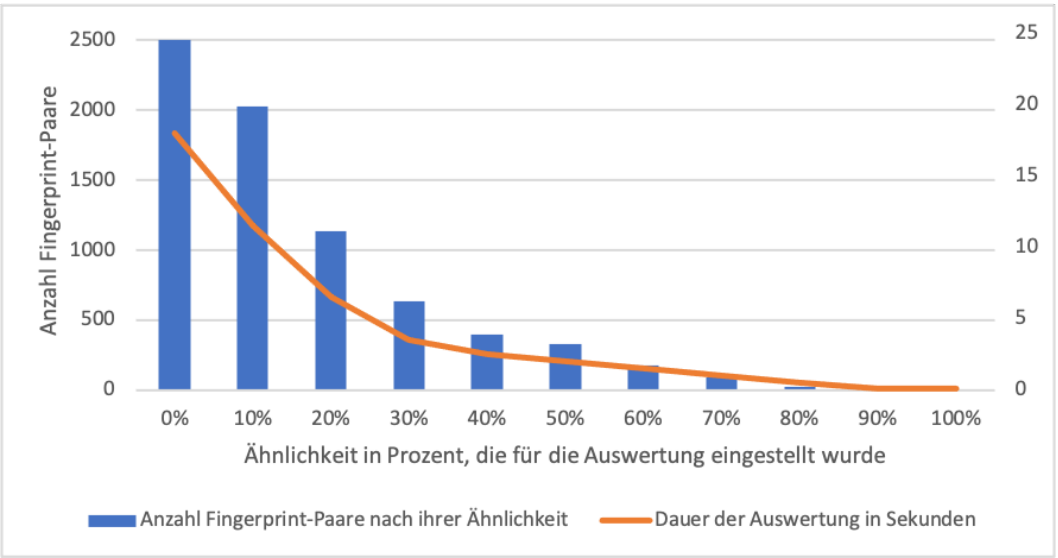

Abbildung 5.11: Performanz des Algorithmus zur Auswertung der Fingerprints in Abhängigkeit von dem eingestellten Ähnlichkeitsschwellwert in Prozent und der damit verbundenen Anzahl an ausgegebenen Fingerprint-Paaren.

Je höher der Ähnlichkeitsschwellwert, desto weniger Fingerprint-Paare werden ausgegeben, und desto weniger Sekunden benötigt der Algorithmus für die Auswertung und Anzeige dieser Fingerprint-Paare. In den Bereichen von 10% bis 30% sowie von 40% bis 50% verhält sich die Dauer in Sekunden annähernd direkt proportional zur Anzahl der ausgegebenen Fingerprint-Paare mit einem Faktor von 176,5 (für den Bereich 10% bis 30%) bzw. 160,5 (für den Bereich 40% bis 50%).

Bei höheren Prozenten sinkt die Zeit langsamer als die Anzahl der Fingerprints. Bei einer Auswertung von 50% dauert die Auswertung in diesem Fall 2 Sekunden, was für den Kontext von LabCon ausreichend schnell ist, insbesondere wenn der Dozent seinen Kurs in ähnlicher Größe wie die Teilnehmeranzahl dieser Evaluation (45) auswerten möchte.

6 Zusammenfassung und Ausblick

In diesem abschließenden Kapitel werden die wichtigsten Erkenntnisse und Ergebnisse der vorliegenden Arbeit zusammengefasst. Im Anschluss daran werden mögliche Erweiterungen des entwickelten Prototyps beleuchtet. Abschließend folgt das Fazit, das eine Gesamtbewertung der Arbeit präsentiert.

6.1 Zusammenfassung

Die Arbeit präsentiert einen funktionsfähigen Prototypen, der in das Remote-Lab-System LabCon integriert wurde. Das Ziel war der Entwurf und die Implementierung zur Sammlung, Speicherung und Auswertung von Digital-Fingerprinting-Daten zur eindeutigen Identifikation von Studenten der Ingenieurwissenschaften, die ihre Online-Praktika über LabCon als Prüfungsvorleistung absolvieren. Die verwendeten Fingerprint-Techniken und -Daten basieren zum Großteil auf den Studien ”Panopticlick” [10], ”AmIUnique” [11] und ”Hiding in the Crowd” [12] und wurden um einige Attribute erweitert, insbesondere um Attribute des WebGL-Fingerprintings.

Zusätzlich zu den Fingerprint-Daten werden auch die Anmeldezeiten der Teilnehmer erfasst und individuelle ID’s generiert, die in verschiedenen lokalen Speichern wie Cookies, Local Storage und IndexedDB abgelegt werden. Diese zusätzlichen Kontrollmechanismen ermöglichen eine weitergehende Überprüfung, insbesondere wenn Teilnehmer regelmäßige Anmeldezeiten besitzen, die dann auffällig unterbrochen wurden oder wenn sie nicht ihre lokale Speicher leeren und die generierten IDs daher konstant eindeutig zugeordnet werden können.

Die Webseite bietet den Dozenten einige Einstellungsmöglichkeiten, um den Identifikationsprozess an ihre Anforderungen anzupassen. So kann beispielsweise der Prozentwert für die Ähnlichkeit der anzuzeigenden Fingerprint-Paare voreingestellt werden und es kann ein Zeitraum für die Anzeige der bei der Anmeldung erfassten Zeitstempel ausgewählt werden.

Im Rahmen einer Evaluation mit 45 Teilnehmern und 72 Fingerprints konnte eine Einzigartigkeit aller Fingerprints erreicht werden. Die Auswahl eines Algorithmus zur Auswertung der Fingerprint-Daten aus zwei implementierten Algorithmen erfolgte zugunsten der Performanz, in Bezug auf die Geschwindigkeit der Auswertung, mit dem minimalen Verlust an Genauigkeit bei der Ermittlung der Ähnlichkeit zweier Fingerprints. Die Auswertung aller Fingerprint-Daten dauert dabei weniger als 20 Sekunden.

6.2 Ausblick

Im Hinblick auf Erweiterungsmöglichkeiten des Systems ergeben sich verschiedene Ansätze zur Verbesserung. Eine Möglichkeit besteht darin, den Algorithmus zu verfeinern, indem längere String-Attribute präziser miteinander verglichen werden, beispielsweise durch die Anwendung des Levenshtein-Algorithmus. Zusätzlich könnte eine Methode implementiert werden, die den Vergleich der Attribute für jedes Fingerprint-Paar speichert. Die Idee besteht darin, Attribute miteinander zu vergleichen und

einen Vektor aus Nullen (Attribute sind ungleich) und Einsen (Attribute sind gleich) zu generieren. Dieser Vektor wird für jedes Fingerprint-Paar gespeichert. Dadurch ist es bei der Auswertung nicht erforderlich, alle Fingerabdrücke erneut zu vergleichen, sondern nur diejenigen, die neu hinzugekommen sind. Die gespeicherten Vektoren werden anschließend in der Auswertung verwendet, indem sie mit den neu berechneten Entropien der Attribute multipliziert werden. Dieser Ansatz könnte die Auswertung der Fingerprints noch effizienter gestalten.

Eine Exportfunktion für die Fingerprint-Daten und Zeitstempel würde den Nutzern die Möglichkeit bieten, die Daten individuell zu sichten. Dabei könnte es möglich sein, dass sie die gewünschten Attribute der Fingerabdrücke für den Export auswählen und somit die Daten an ihre Bedürfnisse anpassen.

Die Erweiterung der Fingerprint-Attribute könnte durch das Hinzufügen weiterer Verfahren erfolgen. Beispielsweise könnten Browser Extensions der Nutzer aufgelistet werden, obwohl dies aufgrund des Fehlens einer API etwas aufwändiger sein könnte. Eine Studie von Alexander Sjösten, Steven Van Acker und Andrei Sabelfeld (2017) [18] könnte hierbei als Grundlage dienen. Ein weiterer Ansatz wäre die Anwendung von CSS-Abfragen, um einzigartige CSS-Variablen für bestimmte Browser zu identifizieren. Diese Techniken könnten durch die Umsetzung von Browser Fingerprinting-Methoden, wie in der Arbeit von T. Unger et al. (2013) [19] beschrieben, realisiert werden.

6.3 Fazit

Der implementierte und integrierte Prototyp zur Teilnehmeridentifikation kann für LabCon verwendet und gegebenenfalls erweitert werden. Die gewählte Kombination an Fingerprint-Attributen hat in der Evaluation für ein eindeutiges Ergebnis gesorgt, da jeder Fingerprint einzigartig war, jedoch lassen sich diese Attribute noch um weitere erweitern, um eine noch höhere Einzigartigkeit der einzelnen Fingerprints zu ermöglichen. Dies ist vor allem wichtig, um die Fingerprints der Studenten noch genauer voneinander unterscheiden zu können, falls diese identische Geräte und Browsereinstellungen verwenden sollten.

Anhang A: Anzahl der Fingerprints für die unterschiedlichen Werte eines Attributes

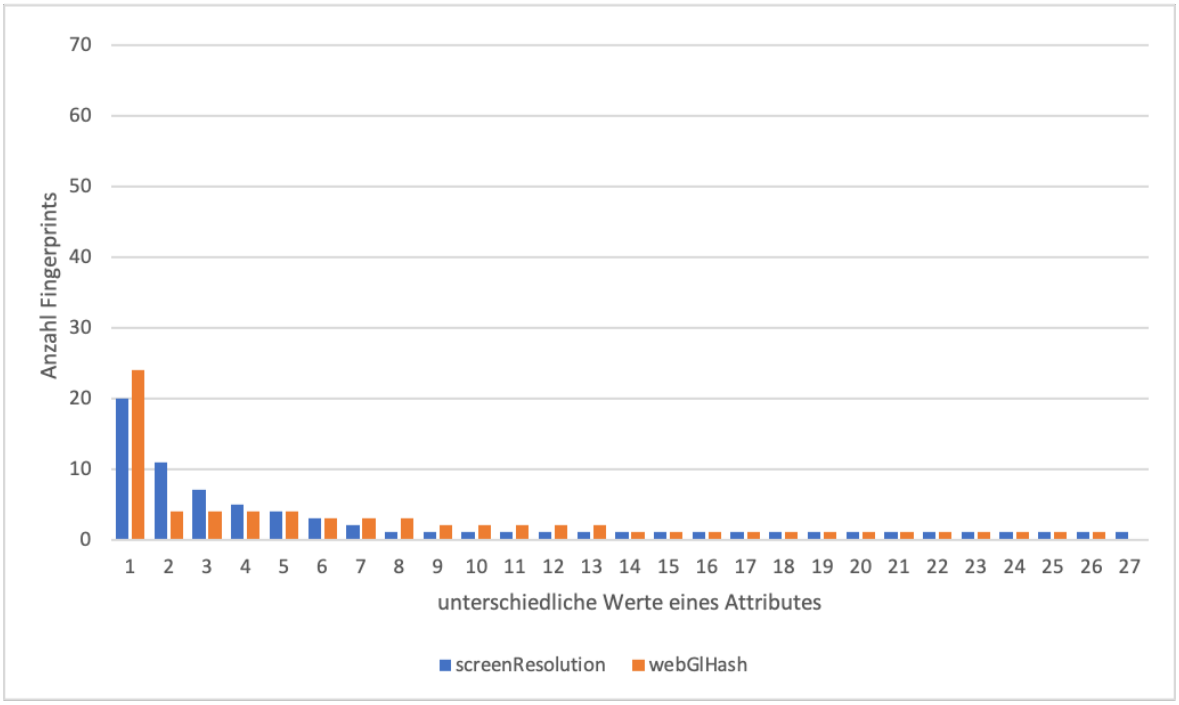

Abbildung A.1: Anzahl der Fingerprints für die unterschiedlichen Werte von Screen Resolution und WebGL-Hash

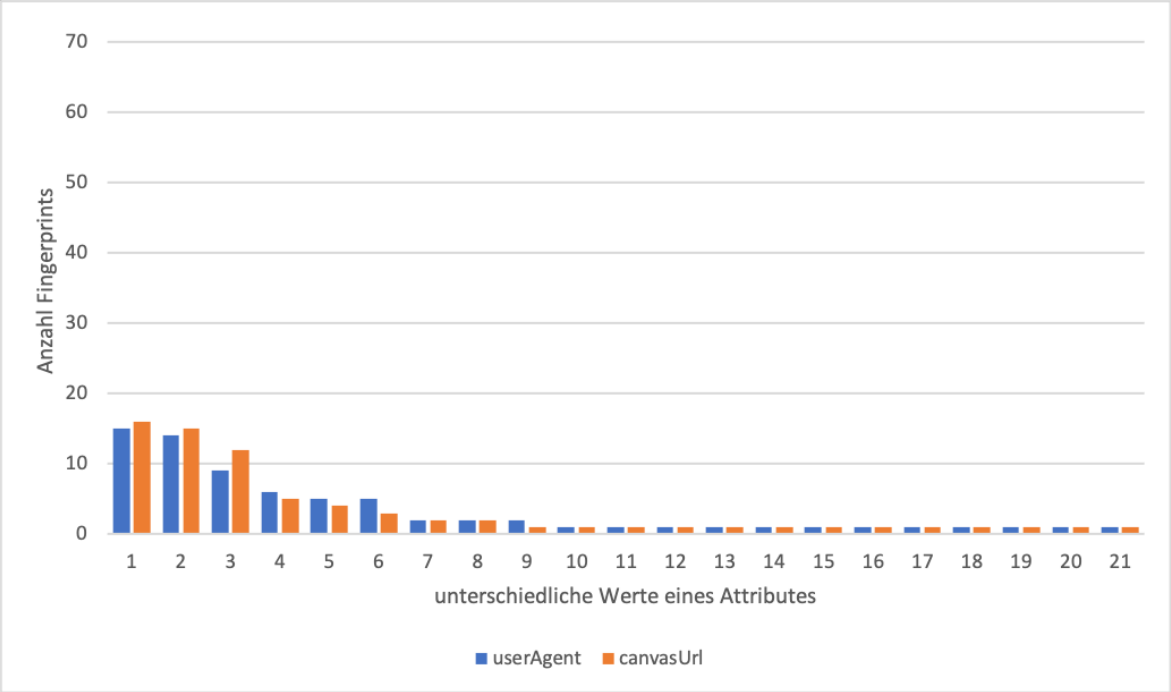

Abbildung A.2: Anzahl der Fingerprints für die unterschiedlichen Werte von User Agent und Canvas URL

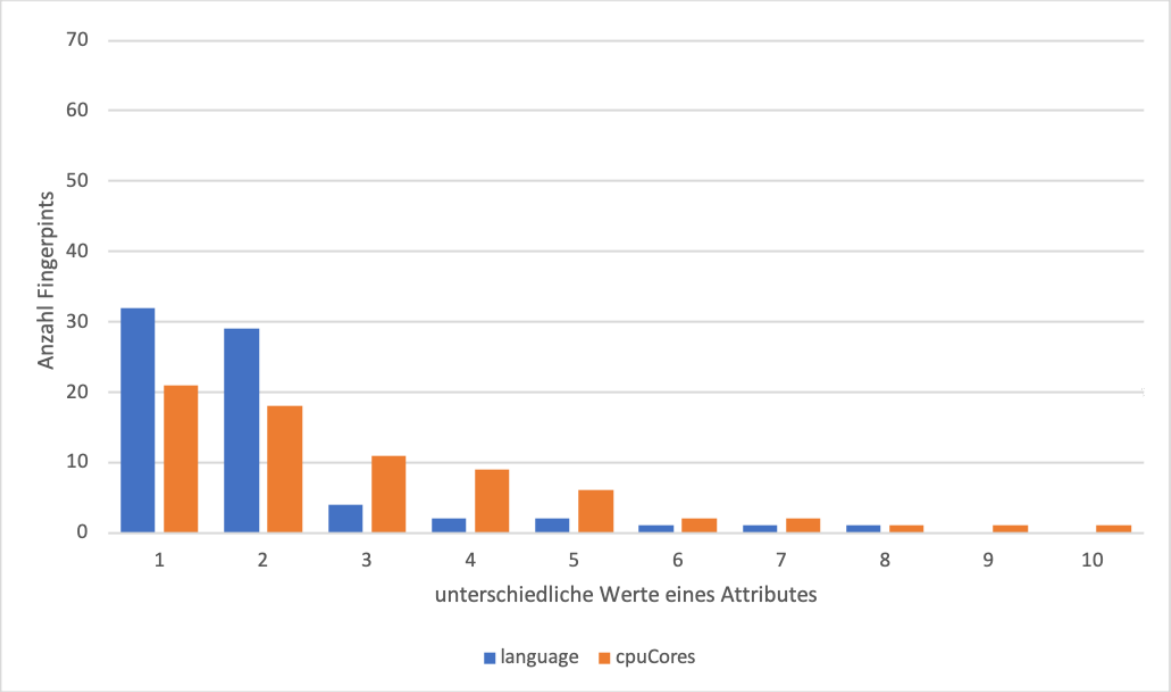

Abbildung A.3: Anzahl der Fingerprints für die unterschiedlichen Werte von Language und CPU Cores

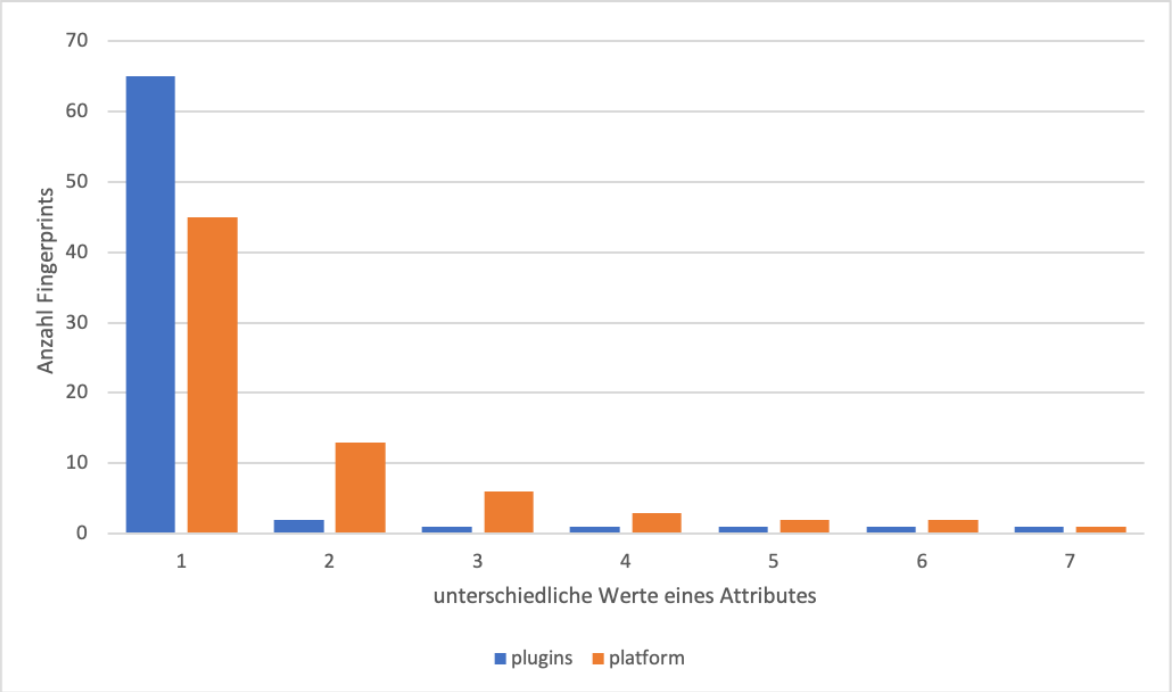

Abbildung A.4: Anzahl der Fingerprints für die unterschiedlichen Werte von Plugins und Platform

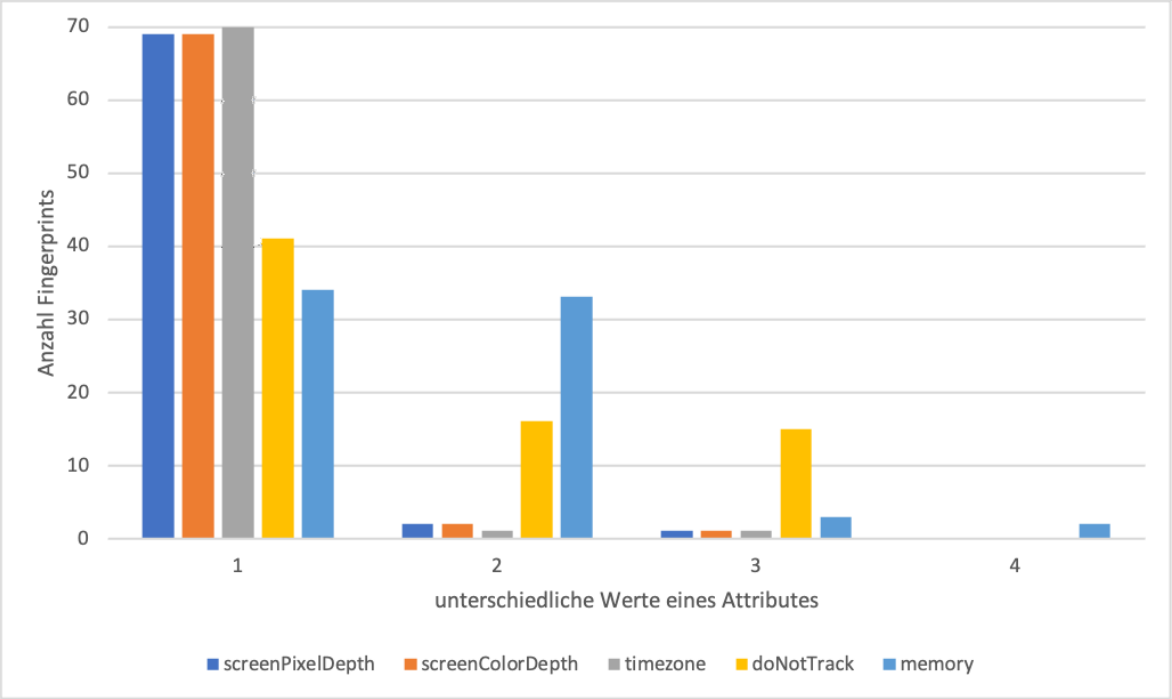

Abbildung A.5: Anzahl der Fingerprints für die unterschiedlichen Werte von Screen Pixel Depth, Screen Color Depth, Timezone, Do Not Track und Memory

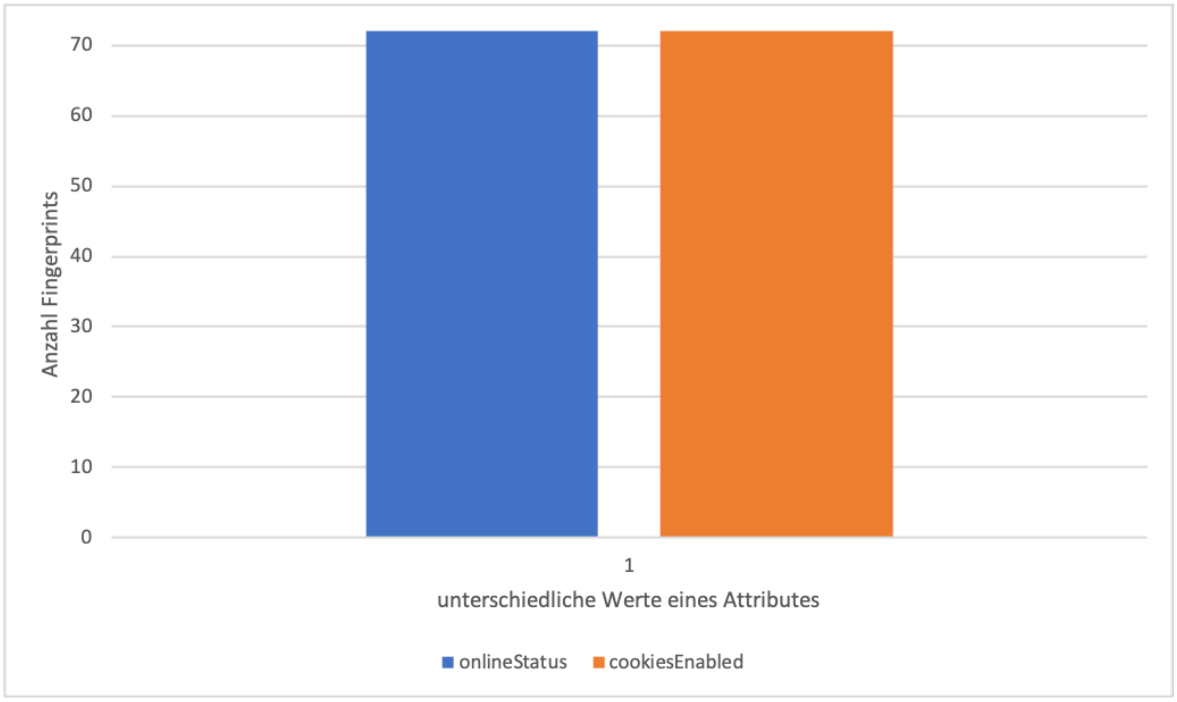

Abbildung A.6: Anzahl der Fingerprints für die unterschiedlichen Werte von Online Status und Cookies Enabled

Literaturverzeichnis

[1] KEYSIGHT Technologies. „PathWave Lab Operations for Remote Learning, PW9112EDU“. (2021), Adresse: https://www.keysight.com/de/de/assets/3121-1016/data-sheets/PathWave-Lab-Operations-for-Remote-Learning-PW9112EDU.pdf (besucht am 26. 11. 2023).

[2] M. Achilles u. a., „Enhancing Digital Learning: A User Management and Access System for Remote Laboratories“, in *eLmL 2023, The Fifteenth International Conference on Mobile, Hybrid, and On-line Learning*, Venice, Italy, 2023. Adresse: https://www.thinkmind.org/index.php?view=article&articleid=elml_2023_2_50_50016 (besucht am 26. 11. 2023).

[3] A-SIT Zentrum für sichere Informationstechnologie – Austria. „Digitale Identitäten: Elektronische Identifikation im Netz“. (Mai 2022), Adresse: https://www.onlinesicherheit.gv.at/Services/News/Digitale-Identitaeten.html (besucht am 25. 11. 2023).

[4] T. Laor u. a., „DRAWNAPART: A Device Identification Technique based on Remote GPU Fingerprinting“, *ArXiv*, Jg. abs/2201.09956, 2022. Adresse: https://api.semanticscholar.org/CorpusID:246276013 (besucht am 26. 11. 2023).

[5] P. Laperdrix, N. Bielova, B. Baudry und G. Avoine, „Browser Fingerprinting: A survey“, *ACM Transactions on the Web (TWEB)*, Jg. 14, Nr. 2, S. 1–33, 2020. Adresse: https://scholar.google.fr/citations?view_op=view_citation&hl=fr&user=hcXaYRAAAAAJ&citation_for_view=hcXaYRAAAAAJ:eQOLeE2rZwMC (besucht am 26. 11. 2023).

[6] IONOS. „Browser-Fingerprinting: Grundlagen und Schutzmöglichkeiten“. (2021), Adresse: https://www.ionos.de/digitalguide/online-marketing/web-analyse/browser-fingerprinting-tracking-ohne-cookies/ (besucht am 26. 11. 2023).

[7] K. Mowery und H. Shacham, „Pixel Perfect: Fingerprinting Canvas in HTML 5“, 2012. Adresse: https://api.semanticscholar.org/CorpusID:1399943 (besucht am 26. 11. 2023).

[8] BrowserLeaks. „WebGL Browser Report“. (2011 – 2023), Adresse: https://browserleaks.com/webgl (besucht am 26. 11. 2023).

[9] Y. Cao, S. Li und E. Wijmans, „(Cross-)Browser Fingerprinting via OS and Hardware Level Features“, in *Network and Distributed System Security Symposium*, 2017. Adresse: https://yinzhicao.org/TrackingFree/crossbrowsertracking_NDSS17.pdf (besucht am 26. 11. 2023).

[10] P. Eckersley, „How Unique is Your Web Browser?“, in *Proceedings of the 10th International Conference on Privacy Enhancing Technologies*, Ser. PETS'10, Berlin, Germany: Springer-Verlag, 2010, S. 1–18, ISBN: 3642145264. Adresse: https://link.springer.com/chapter/10.1007/978-3-642-14527-8_1 (besucht am 26. 11. 2023).

[11] P. Laperdrix, W. Rudametkin und B. Baudry, „Beauty and the Beast: Diverting modern web browsers to build unique browser fingerprints“, in *37th IEEE Symposium on Security and Privacy (S&P 2016)*, San Jose, United States, Mai 2016. Adresse: https://inria.hal.science/hal-01285470 (besucht am 26. 11. 2023).

[12] A. Gómez-Boix, P. Laperdrix und B. Baudry, „Hiding in the Crowd: An Analysis of the Effectiveness of Browser Fingerprinting at Large Scale“, in *Proceedings of the 2018 World Wide Web Conference*, Ser. WWW '18, Lyon, France: International World Wide Web Conferences Steering Committee, 2018, S. 309–318, ISBN: 9781450356398. Adresse: https://doi.org/10.1145/3178876.3186097 (besucht am 26. 11. 2023).

[13] MDN Web Docs. „Using HTTP cookies“. (2023), Adresse: https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies (besucht am 06. 12. 2023).

[14] MDN Web Docs. „Web Storage API“. (2023), Adresse: https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API (besucht am 06. 12. 2023).

[15] MDN Web Docs. „IndexedDB API“. (2023), Adresse: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API (besucht am 06. 12. 2023).

[16] E. F. Foundation, *Cover Your Tracks*. Adresse: https://coveryourtracks.eff.org (besucht am 26. 11. 2023).

[17] P. Leach, M. Mealling und R. Salz, *A Universally Unique IDentifier (UUID) URN Namespace*, Internet Engineering Task Force, RFC 4122, 2005. Adresse: https://www.ietf.org/rfc/rfc4122.txt (besucht am 07. 12. 2023).

[18] Sjösten, Alexander and Van Acker, Steven and Sabelfeld, Andrei, „Discovering Browser Extensions via Web Accessible Resources“, in *Proceedings of the Seventh ACM on Conference on Data and Application Security and Privacy*, Ser. CODASPY '17, Scottsdale, Arizona, USA: Association for Computing Machinery, 2017, S. 329–336, ISBN: 9781450345231. Adresse: https://doi.org/10.1145/3029806.3029820 (besucht am 26. 11. 2023).

[19] T. Unger, M. Mulazzani, D. Frühwirt, M. Huber, S. Schrittwieser und E. Weippl, „SHPF: Enhancing HTTP(S) Session Security with Browser Fingerprinting“, in *2013 International Conference on Availability, Reliability and Security*, 2013, S. 255–261. Adresse: https://doi.org/10.1109/ARES.2013.33 (besucht am 26. 11. 2023).

Eidesstattliche Erklärung

Hiermit versichere ich – Emilia Cora Weinhold – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 11. Dezember 2023

Ort, Datum

Emilia Cora Weinhold