

## Bachelor-Thesis

# Analyse von Fakeshops und ihren verwendeten Systemen mit Hilfe von Python

eingereicht von:

Stefan Scheibe

geboren am 09. April 1984 in Hannover  
Studiengang IT-Forensik

Betreuer:

Prof. Dr.-Ing. Antje Raab-Düsterhöft

weitere Gutachter:

Prof. Dr.-Ing. habil. Andreas Ahrens

Hannover, den 16. Juli 2023

## **Aufgabenstellung**

In der Bachelor-Thesis ist eine Analyse real betriebener Fakeshops bzgl. der verwendeten Systeme durchzuführen. Es wird angestrebt, Trends in der Verbreitung von Fakeshops zu erkennen.

Dazu ist zu untersuchen

- welche Systeme zum Betreiben eines Fakeshops eingesetzt,
- welche Zielgruppe anhand der verwendeten Sprache favorisiert und
- in welchen Ländern Fakeshops gehostet werden.

Das Ziel ist es, Gemeinsamkeiten in den Systemen zu finden.

## Kurzreferat

Die vorliegende Bachelor-Thesis befasst sich mit der Analyse von Fakeshops und den von ihnen verwendeten Systemen unter Verwendung von Python. Fakeshops stellen heute ein ernsthaftes Problem im Bereich des Online-Betrugs dar und können für Verbraucher erhebliche finanzielle Schäden verursachen. Das Ziel dieser Arbeit ist es, anhand erstellter Methodiken und Algorithmen auf Basis von Python Fakeshops auf die verwendeten Systeme, Sprache, IP-Adresse, Server-Standort und mögliche Erweiterungen zu analysieren.

Nach der Analyse umfangreicher Daten, die mithilfe von Python-basierten Techniken wie Web-Scraping und Datenanalyse extrahiert und verarbeitet wurden, werden sowohl Ergebnisse dargestellt als auch Trends und Entwicklungen aufgezeigt.

Die Ergebnisse dieser Thesis sollen einen wertvollen Einblick in die Funktionsweise von Fakeshops geben und dazu beitragen, das Bewusstsein für die Problematik zu schärfen. Die Methoden und Algorithmen können von Strafverfolgungsbehörden, E-Commerce-Plattformen und Verbraucherschutzorganisationen genutzt werden, um Trends in der Verbreitung von Fakeshops zu erkennen. Anhand der erzielten Ergebnisse können genannte Institutionen vorhandene präventive Maßnahmen weiter ausbauen und potenzielle Opfer schützen.

**Abstract**

This bachelor thesis deals with the analysis of fakeshops and the systems they use, using Python. Fakeshops are a serious problem in the field of online fraud today and can cause significant financial damage to consumers. The aim of this work is to analyze fakeshops for the systems used, language, IP address, server location and possible extensions using created methodologies and algorithms based on Python.

After analyzing extensive data extracted and processed using Python-based techniques such as web scraping and data analysis, results are presented as well as trends and developments are shown.

The results of this thesis are intended to provide valuable insight into how fakeshops work and to help raise awareness of the problem. The methods and algorithms can be used by law enforcement agencies, e-commerce platforms and consumer protection organizations to identify trends in the spread of fakeshops. Based on the results obtained, said institutions can further develop existing preventive measures and protect potential victims.

## Inhalt

Aufgabenstellung.....	3
Kurzreferat.....	4
Abstract .....	5
Inhalt.....	6
1 Einleitung.....	9
2 Grundlagen.....	13
2.1 Online-Shops .....	13
2.1.1 Definition .....	13
2.1.2 Funktionen und Merkmale .....	13
2.1.3 Entwicklung von Online-Shopping .....	15
2.1.4 Kategorien von Onlineshop-Systemen .....	16
2.1.5 Verbreitung von Onlineshop-Systemen.....	18
2.2 Fakeshops.....	19
2.2.1 Definition .....	19
2.2.2 Arten von Fakeshops.....	20
2.2.3 Gesetzliche Grundlage .....	22
2.2.4 Maßnahmen.....	22
2.3 Merkmale .....	25
2.3.1 Unabhängige Merkmale.....	25
2.3.2 Systemabhängige Merkmale .....	26
2.4 Automatische Erkennung von Fakeshops .....	28
2.4.1 Merkmale und Datenquellen.....	28
2.4.2 Naives Bayes als Klassifikationsverfahren .....	29
2.4.3 Evaluation der Klassifikation.....	29
2.4.4 Herausforderungen und zukünftige Entwicklungen .....	29
3 Methodik.....	31
3.1 Methodik und Inhaltlicher Aufbau .....	31
3.2 Erläuterung der Forschungsfragen .....	32
3.2.1 Frage 1.....	32
3.2.2 Frage 2.....	33
3.2.3 Frage 3.....	33
3.2.4 Ziel .....	33
4 Vorgehen und Aufbau .....	34
4.1 Vorgehen.....	34
4.1.1 Auffinden aktueller Fakeshops .....	34

---

4.1.2	Auswahl relevanter Daten.....	38
4.1.3	Nutzung von APIs zur Ergänzung der Analysedaten .....	41
4.1.4	Extrahierung der Informationen von Fakeshops .....	43
4.1.5	Überprüfung der Daten .....	43
4.2	Aufbau .....	43
4.2.1	Datenbankumgebung .....	44
4.2.2	Teilautomatisierung mit Python .....	45
4.2.3	Python-Skript bt_fs_main.py .....	47
4.2.4	Python-Skript bt_fs_analyze.py .....	48
4.2.5	Python-Skript bt_fs_functions.py .....	49
4.2.6	Crontab .....	60
4.2.7	Sicherung der Daten .....	61
4.2.8	CSV zur Auswertung der Daten .....	62
5	Durchführung .....	63
5.1	Zeitraum .....	63
5.2	Analyse mit Python-Skripten .....	63
5.3	Limitationen .....	64
5.4	Validierung und Datenbereinigung .....	64
6	Ergebnis und Trends .....	68
6.1	Ergebnis .....	68
6.1.1	Verwendete Systeme .....	69
6.1.2	Systeme mit Erweiterung .....	72
6.1.3	Zuordnung Systeme und Erweiterungen in Online-Shop Kategorien .....	73
6.1.4	Versionen von System WordPress und PHP .....	75
6.1.5	System Shopware in Version 5 und 6 .....	77
6.1.6	Einsatz von Cloudflare .....	79
6.1.7	Systeme die Cloudflare nutzen .....	80
6.1.8	Server-Standorte .....	81
6.1.9	IP-Adressen .....	85
6.1.10	IP-Adresse 23.227.38.32 .....	89
6.2	Trends .....	89
7	Fazit und Ausblick .....	92
	Literaturverzeichnis .....	94
	Abbildungsverzeichnis .....	98
	Tabellenverzeichnis .....	100
	Verzeichnis der Abkürzungen .....	101
	Selbstständigkeitserklärung .....	102

Anhangverzeichnis .....	103
bt_fs_functions.py .....	A
bt_fakeshops_csv.py .....	I

## 1 Einleitung

### Motivation

Im Zeitalter der Digitalisierung ist die Bedeutung von Internetseiten unumstritten. Das Erstellen von Internetseiten wie Lifestyleblogs, Foren, Nachrichtenseiten oder Online-Shops ist einfacher als je zuvor, da es zahlreiche Tools und Plattformen gibt, die es auch Ungeübten oder Laien ermöglichen, eine Internetseite in kurzer Zeit zu erstellen. Das Unternehmen VeriSign gab in einem Report für das 4. Quartal 2022 bekannt, dass weltweit über 350 Millionen Domains registriert sind. [1]

Mit der Zunahme von Internetseiten wie Online-Shops steigt auch die Anzahl betrügerischer Webseiten wie so genannte Fakeshops. Fakeshops präsentieren sich als vertrauenswürdige Online-Händler und ziehen Kunden mit äußerst attraktiven Angeboten an. Obwohl die Kunden im Voraus bezahlen, wird die Ware meist nicht geliefert. [2]

Gemäß der Polizeilichen Kriminalstatistik (PKS) von 2022 wurden 396.184 der erfassten Straftaten als Delikte mit dem Tatmittel Internet eingestuft. Dabei entfielen 22,7% auf Warenbetrug, welcher gemäß § 263 StGB strafbar ist. [3] Zum Warenbetrug zählen auch so genannte Fakeshops. Die Zunahme von Fakeshops stellt somit eine zunehmende Bedrohung für Verbraucher und den Online-Handel dar.

In der Projektarbeit „Analyse von Fakeshops anhand des jeweiligen eingesetzten Content-Management-System mit Abfrage von gefundenen Merkmalen auf Basis von „Python“ im Modul Forensik Projekt 2 wurde sich intensiv mit der Untersuchung von Merkmalen von Fakeshops auf den Systemen WordPress und Shopware in der Version 5 beschäftigt. [4]

Aufbauend auf den Erkenntnissen der Projektarbeit wird in dieser Thesis nun der nächste Schritt getätigt und Fakeshops hinsichtlich ihrer verwendeten Shop- bzw. Content-Management Systeme analysiert. Durch diese Analyse kann ein Überblick darüber geschaffen werden, welche Systeme von Fakeshop-Betreibern eingesetzt werden.



Der Stand der Forschung zu dem behandelten Themenkomplex ist stark begrenzt, da zum gegenwärtigen Zeitpunkt keine einschlägige Literatur existiert.

Das Österreichische Institut für angewandte Telekommunikation (ÖIAT) strebt durch den Einsatz von künstlicher Intelligenz die Erkennung von Fakeshops im Internet an. Mit einem eigens entwickelten Fakeshop-Detector werden Konsumenten in Echtzeit beim Onlineshopping geschützt. Identifizierte Fakeshops werden auf der Plattform [www.watchlist-internet.at](http://www.watchlist-internet.at) veröffentlicht. Mit einem präventiven Ansatz zielt das ÖIAT darauf ab, Konsumenten im Onlinehandel vor Betrug zu schützen, da der entstandene Schaden oft erst festgestellt wird, wenn eine Rückerstattung des im Voraus bezahlten Betrags nicht mehr möglich ist. [5]

Die Realisierung des Fakeshops-Detectors wurde in der wissenschaftlichen Ausarbeitung „Real-Time Detection of Fake-Shops through Machine Learning“ veröffentlicht. [6]

Weitere wissenschaftliche Ausarbeitungen zum Thema Fakeshops sind zum Zeitpunkt der Erstellung dieser Bachelor-Thesis nicht publiziert.

## **Ausgangssituation und Problemstellung**

Im Rahmen der Bachelor-Thesis ist es von zentraler Bedeutung, reale Fakeshops zu untersuchen, die bereits in Betrieb sind. Ein Fakeshop gilt in diesem Zusammenhang als identifiziert, wenn er auf den Portalen [www.auktionshilfe.info](http://www.auktionshilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) als Fakeshops bzw. betrügerischer Online-Shop aufgeführt ist.

Um die identifizierten Fakeshops möglichst effizient zu verarbeiten, soll im Rahmen der Thesis ein teilautomatisierter Prozess mittels Python entwickelt werden. In einem ersten Schritt werden dafür die Domains der identifizierten Fakeshops in einer einfachen Datenbank gespeichert. Im zweiten Schritt werden dann mithilfe weiterer Python-Skripte die in der Datenbank erfassten Fakeshops untersucht, um Informationen über das verwendete Shop- bzw. Content-Management-System, die Version, eventuelle Shop-Erweiterungen, die verwendete Sprache des Webseiten-Inhalts, den Standort des Servers, die IP-

Adresse sowie eine mögliche Absicherung über den Dienst Cloudflare zu erfassen und den Datenbestand zu vervollständigen.

Für die Durchführung der Bachelor-Thesis wird die Umgebung der Datenbank sowie der Python-Skripte auf einem Raspberry Pi unter Ubuntu Server 20.04.6 LTS betrieben.

Die Herausforderung, die sich bei dieser Bachelor-Thesis ergibt, betrifft insbesondere die zeitliche Komponente. Eine umfassende Datenbasis ist von entscheidender Bedeutung, um ein aussagekräftiges Ergebnis zu erzielen. Da es sich bei den analysierten Online-Shops um reale Fakeshops handelt, besteht die Möglichkeit, dass ein Fakeshop nach der Identifizierung zeitnah offline genommen wird. Es ist daher wichtig, dass die Untersuchung unmittelbar und möglichst vollständig durchgeführt wird. Die Datenerhebung erstreckt sich über einen längeren Zeitraum und erfolgt mehrmals pro Tag, weshalb eine präzise Teilautomatisierung unverzichtbar ist. Es muss jedoch beachtet werden, dass manuelle Nacharbeiten am Datenbestand erforderlich sind, die binnen kurzer Zeit nach der Aufnahme durchgeführt werden müssen, um so eine möglichst genaue Analyse sicherzustellen.

### **Zielsetzung und Grenzen**

Das Ziel dieser Bachelor-Thesis ist es, einen umfassenden Überblick über die bei realen Fakeshops eingesetzten Systeme zu erhalten. Dazu werden die Domains von Fakeshops über einen Zeitraum von ca. 10 Wochen ausgelesen und anschließend mit der Unterstützung von selbst erstellten Python-Skripten analysiert. Anhand dieses Vorgehens soll aufgezeigt werden, welche Shop- bzw. Content-Management-Systeme eingesetzt, welche Zielgruppe fokussiert und in welchen Ländern die Fakeshops gehostet werden. Basierend auf diesen Informationen soll die Verbreitung bestimmter Systeme bei Fakeshops aufgezeigt werden.

Um dem Umfang dieser Arbeit zu entsprechen, wird im Folgenden nicht konkret auf die automatische Erkennung von Fakeshops eingegangen. Weiter erfolgt keine Betrachtung und eingehendere Erläuterung der identifizierten Shop- bzw. Content-Management-Systeme. Die Inbetriebnahme und Installation der

technischen Umgebung, wie beispielsweise der eingesetzte Raspberry Pi, das verwendete Betriebssystem Ubuntu Server sowie die Einrichtung der Datenbankumgebung, werden ebenfalls nicht im Detail erläutert.

## **2 Grundlagen**

### **2.1 Online-Shops**

#### **2.1.1 Definition**

Der Begriff Online-Shop bezeichnet eine Webseite, eine Art digitalen Geschäftsraum im Internet, auf welcher Waren und Dienstleistungen angeboten und verkauft werden. Der Begriff Online-Shop ist nicht mit dem Begriff Electronic Commerce (E-Commerce) zu verwechseln. E-Commerce bezeichnet den elektronischen Markt für den Vertrieb von Waren und Dienstleistungen mit dessen Hilfe Vertriebswege verkürzt, Geschäftsprozesse vereinfacht und neue Geschäftsbereiche abgedeckt werden können. [7]

Während sich E-Commerce auf alle Aspekte des Online-Geschäftsbetriebs bezieht, deckt der Online-Handel über Online-Shops den Verkauf und Kauf von Waren und Dienstleistungen ab. [8]

#### **2.1.2 Funktionen und Merkmale**

In einem Online-Shop werden Konsumenten Waren und Dienstleistungen angeboten. Diese Produkte werden häufig mit Bildern und aussagekräftigen Beschreibungen präsentiert. Neben Preisen für die Produkte werden in der Regel auch die Verfügbarkeiten sowie Lieferzeiten angegeben.

Durchgeführte Bestellungen liefern Unternehmen große Datenmengen. Anhand der Datenmengen können Kundenverhalten und Warenkorbanalysen für die Erkennung von Trends und Wahrscheinlichkeiten berechnet werden. Dabei werden Zusammenhänge zwischen den Produkten analysiert. Im Ergebnis einer Warenkorbanalyse werden Assoziationsregeln aufgestellt, mit welchen Kunden anschließend Produktangebote unterbreitet werden können. Kennzahlen der Assoziationsanalyse sind der Support und die Konfidenz von Produkten. Kauft ein Kunde Produkt A, so gibt die Konfidenz die Wahrscheinlichkeit wieder, dass der Kunde auch Produkt B kaufen wird. Viele Online-Shops nutzen diese Daten und zeigen Kunden beim „Stöbern“ bereits an, dass Kunden, welche sich Produkt

A gekauft haben, auch dieses Produkt B kaufen würden. [9]

Die wichtigsten Funktionen in einem Online-Shop sind der Bestellprozess und die Zahlungsabwicklung. Über ein Warenkorbsystem können Kunden Produkte dem Warenkorb hinzufügen. Bei der Bestellung müssen in der Regel vorab persönliche Daten für die Liefer- und Rechnungsanschrift im Online-Shop hinterlegt werden. Anschließend folgt die Auswahl der gewünschten Zahlungsmethode. Professionelle Online-Shops bieten eine Vielzahl von Zahlungsmöglichkeiten an. Diese reichen von E-Wallets wie PayPal, Zahlung per Rechnung, Zahlung über Dienstleister wie Klarna, Kreditkarte oder häufig auch per Vorkasse bzw. Banküberweisung. In einer durchgeführten Umfrage aus dem Jahr 2022 gaben 64% der Befragten an, dass die Zahlung via E-Wallet die bevorzugte Zahlungsmethode sei. Gefolgt von Zahlung per Rechnung (44%) und das Begleichen der Rechnung über den Zahlungsdienstleister Klarna mit 28%.

Obwohl heute viele Zahlungsmöglichkeiten existieren, ist die Banküberweisung mit 25% weiterhin eine beliebte und gegenwärtige Zahlungsmethode. [10]

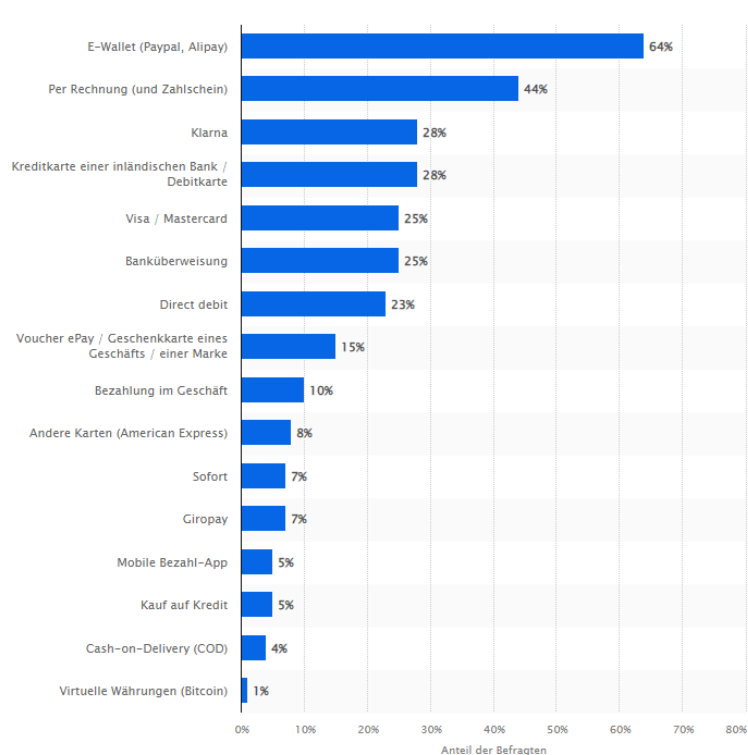


Abbildung 1: Übersicht bevorzugter Zahlungsmethoden 2022

Mit dem Fortschritt der Digitalisierung hat sich der Kauf von Produkten über das Internet auf das Smartphone verlagert. Entsprechend ist es von großer Bedeutung, dass ein Online-Shop für mobile Endgeräte optimiert ist oder sogar über eine eigene Applikation verfügt. In einer Umfrage aus dem Jahr 2022 beantworteten 70% der 1984 Befragten, dass sie das Smartphone am häufigsten für den Einkauf im Internet nutzen. [11]

### **2.1.3 Entwicklung von Online-Shopping**

Ab dem Jahr 2000 bis zum Jahr 2021 herrschte ein kontinuierlicher Anstieg beim Umsatz mit Waren im Bereich E-Commerce. Die Corona Pandemie beschleunigte die Umsatzsteigerung enorm und Einzelhändler waren gezwungen, den Schritt zum eigenen Online-Shop neben dem Betrieb eines stationären Ladengeschäftes zu wagen. Im Jahr 2021 lag der Umsatz im Bereich E-Commerce in Deutschland mit 99,1 Milliarden Euro am höchsten. Nach 21 Jahren durchgängiger Umsatzsteigerung war 2022 das erste Jahr, in dem ein Rückgang der Umsätze zu verzeichnen war. [12]

Das Unternehmen APPINIO GmbH führte 2022 eine Umfrage in fünf Altersgruppen durch. Inhalt der Umfrage war das Thema Onlineshopping und wie häufig die Verbraucher online shoppen oder online nach neuen Produkten suchen. Mit mehr als 30% gaben alle Altersgruppen an, dass sie mehrmals die Woche im Internet nach neuen Produkten suchen oder diese kaufen. Mit 17 bis 21% folgte die Angabe, dass fast täglich dem Onlineshopping nachgegangen wird. [13]

Die Entstehung und Entwicklung von Systemen für Online-Shops stehen in engem Zusammenhang mit dem Aufstieg des E-Commerce. Durch die zunehmende Nutzung des Internets und die dadurch entstandene Akzeptanz für Online-Shops, wurden zahlreiche Onlineshop-Systeme entwickelt. Im Laufe der Jahre haben sich die Funktionalitäten der verschiedenen Systeme kontinuierlich erweitert und weiterentwickelt. [14]

### 2.1.4 Kategorien von Onlineshop-Systemen

Durch den Aufstieg des E-Commerce und die damit voranschreitende Entwicklung von Onlineshop-Systemen, können die verschiedenen Systeme heute in fünf Kategorien unterteilt werden:

- Self-Hosted Onlineshop-Systeme
- Cloud-basierte Onlineshop-Systeme
- Open-Source Onlineshop-Systeme
- Plugin-Software als Erweiterung
- Eigenentwicklungen

#### **Self-Hosted Onlineshop-Systeme**

Ein Self-Hosted System erfordert, dass der Betreibende das Shopsystem auf einem eigenen Server hosted. Dabei ist es unerheblich, ob das Hosting On-Premise stattfindet oder ob das System bei einem Dienstleister gehostet wird. Bei dieser Art besitzt der Betreibende die volle Kontrolle über den Shop und die Funktionalitäten. Die Wartung des Online-Shops, wie das Einspielen von Updates oder die Erstellung von Datensicherungen müssen durch den Betreibenden selbst durchgeführt werden. Verbreitete Self-Hosted Onlineshop-Systeme sind Magento, Shopware und PrestaShop.

#### **Cloud-basierte Onlineshop-Systeme**

Cloud-basierte Systeme werden vom Anbieter selbst gehostet und gewartet. Der Betreibende hat keinen Zugriff auf die Serverinfrastruktur und besitzt bei dieser Art von Systemen nicht die komplette Kontrolle. Der Vorteil gegenüber Self-Hosted Systemen ist zum einen die Skalierbarkeit, welche schneller an Bedürfnisse angepasst werden kann und zum anderen, dass die Wartung vom Anbieter selbst durchgeführt wird. Damit verbunden sind jedoch die finanziellen Aufwände.

Shopify, Wix Stores und BigCommerce sind weit verbreitete Cloud-basierte

Shopsysteme. Der Anbieter Ecwid zählt ebenfalls zu den Cloud-basierten Lösungen. Mit Ecwid können Betreibende einen Online-Shop in bestehende Webseiten, soziale Medien und andere Online-Plattformen integrieren. Die Integration kann in Open-Source Systeme sowie in Content-Management-Systemen erfolgen.

### **Open-Source Onlineshop-Systeme**

Bei Open-Source Onlineshop-Systemen liegt der Quellcode offen und kann von Entwicklern bearbeitet und angepasst werden. Diese Art von Systemen sind kostenlos und bieten Betreibenden eine hohe Flexibilität, da sie an die Bedürfnisse eines Online-Shops angepasst werden können. Durch die Weiterentwicklung vorhandener Communitys werden Open-Source Systeme stetig verbessert und erhalten häufig neue Funktionen oder Erweiterungen in Form von Plugins. Das Betreiben kann auf eigenen Servern oder Cloud-basierte Servern stattfinden. Bekannte Vertreter der Open-Source Onlineshop-Systeme sind OpenCart, osCommerce und Zen Cart.

### **Plugin-Software als Erweiterung**

Wie bei Open-Source Onlineshops wird bei kostenfreien Content-Management-Systemen mit einer vorhandenen Community die Software stetig erweitert und verbessert. Mit zusätzlicher Plugin-Software kann ein CMS erweitert werden. Mit Hilfe von Plugin-Software wie WooCommerce kann das CMS WordPress leicht um einen eigenen Online-Shop erweitert werden.

### **Eigenentwicklungen**

Eine weitere Möglichkeit ist die Entwicklung von eigenen Shop- oder Content-Management-Systemen. Eigenentwicklungen können auf die eigenen Bedürfnisse zugeschnitten werden, sind damit flexibel und unabhängig. Die Entwicklung ist jedoch zeitintensiv und kann bei externer Entwicklung hohe Kosten verursachen. Eigenentwickelte PHP-Umgebungen können mit der Implementierung von Warenkorb-Funktionalitäten (Cart functionality) ähnlich wie



WordPress in Verbindung mit WooCommerce um einen Online-Shop erweitert werden.

### 2.1.5 Verbreitung von Onlineshop-Systemen

Onlineshop-Systeme bilden die Grundlage eines Online-Shops und damit das Fundament einer Webseite. Das Unternehmen BuiltWith, ein Website-Profiler zur Wettbewerbsanalyse, gibt an, dass in Deutschland 724.796 Online-Shops (Stand 24. April 2023) registriert sind. Ein Online-Shop wird dann als Online-Shop eingestuft, sobald der Webseiten-Crawler von BuiltWith die entsprechenden Systeme auf der überprüften Webseite vorfindet.

Laut BuiltWith sind Shopify (24%), Ecwid (22%) und WordPress in Verbindung mit WooCommerce (11%) die führenden Systeme der Online-Shops in Deutschland. Wix Stores (10%) und Shopware (6%) belegen die Platzierungen vier und fünf. [15]

#### eCommerce Usage Distribution in Germany

Distribution for websites using eCommerce technologies

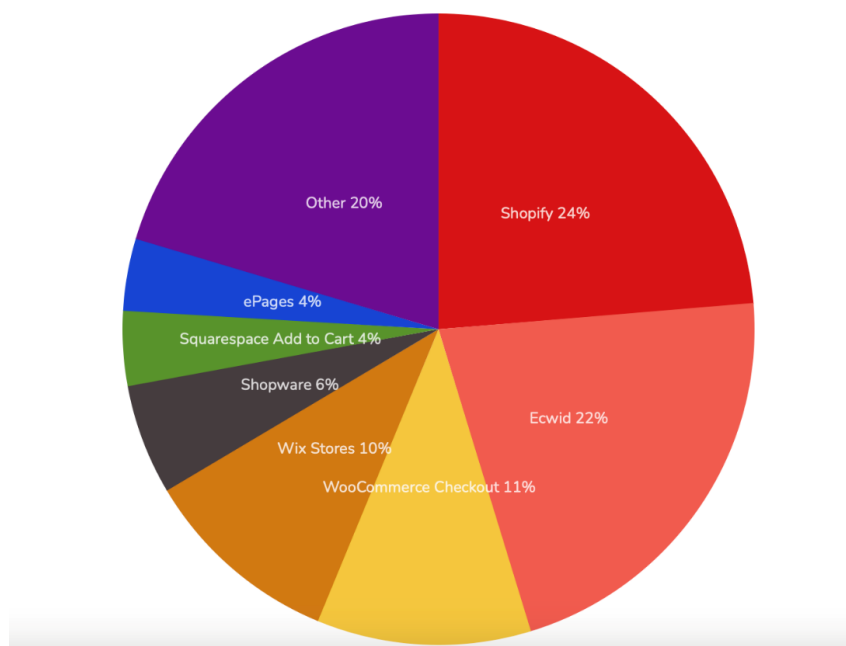


Abbildung 2: Verteilung e-commerce Lösungen auf Webseiten (DE)

Durch die Vielzahl von Tools und Plattformen zur Erstellung von Online-Shops ist es heute, im Zeitalter der Digitalisierung, relativ einfach, einen eigenen Online-Shop zu erstellen. Programmierkenntnisse sind nicht notwendig, da Betriebsmodelle wie SaaS (Software-as-a-Service) dem Kunden ermöglichen, in kurzer Zeit einen kompletten Online-Shop zu betreiben. Vorgefertigte Templates, integrierte Zahlungsoptionen und Drag-and-Drop-Funktionalitäten ermöglichen auch Benutzern ohne spezielle Kenntnisse einen eigenen Shop zu erstellen. Auch Self-hosted Onlineshop-Systeme können in der heutigen Zeit von Laien aufgesetzt und betrieben werden. Auf Plattformen wie Youtube oder durch Suchmaschinen können Problemstellungen schnell gesucht, gefunden und gelöst werden.

## **2.2 Fakeshops**

### **2.2.1 Definition**

Eine einheitliche Definition für den Begriff „Fakeshop“ ist nicht existent. Der Begriff Fakeshop ist jedoch allgemein anerkannt und wird in verschiedenen Kontexten verwendet. Bei einem Fakeshop handelt es sich um eine eigenständige Internetseite, in der Regel um einen täuschend echt aussehenden Online-Shop, auf welcher Waren zum Kauf angeboten werden. Die Waren werden dabei äußerst günstig angeboten, was Besuchende suggeriert, beim Stöbern auf ein Schnäppchen zu stoßen. Dabei handelt es sich häufig um Waren, welche hochpreisig sind und eine hohe Nachfrage besitzen. Produktbilder, Beschreibungen und die Allgemeinen Geschäftsbedingungen sind häufig von anderen Online-Shops kopiert und täuschen Besuchende so erheblich. Das Impressum wurde in der Regel frei erfunden und bei den genannten Personen oder Unternehmen handelt es sich um gefälschte Profile. Durch platzierte Gütesiegel, teilweise frei erfunden oder auch mit bekannten Anbietern wird zusätzlich die Seriosität des Online-Shop suggeriert. Wird der Besuchende zum Käufer, so werden entweder minderwertige Waren von den Betreibenden oder aber nach Vorauszahlung keine Waren versendet. [16]

Fakeshops geben vor, Zahlungsmethoden wie E-Wallets, Kreditkarte oder Bezahlung durch einen Zahlungsdienstleister zu akzeptieren. Beim Bestellvorgang steht jedoch nur die Möglichkeit der Bezahlung mit Vorkasse zur Verfügung.

Sobald der Kaufende eine Überweisung an den Fakeshop-Betreiber durchführt und sich das Empfängerkonto im Europäischen Wirtschaftsraum befindet, darf der Vorgang gemäß § 675s BGB nur einen Bankarbeitstag andauern. [17]

Betreibende von Fakeshops erhalten so einen Vorteil, da eine selbstständig durchgeführte Überweisung nur dann rückgängig gemacht werden kann, wenn der Zahlungsdienstleister den Vorgang noch nicht ausgeführt hat. [18]

Kaufende haben somit unter einen Bankarbeitstag Zeit, nach der Durchführung zu erkennen, dass es sich um einen Fakeshop handelt und den Vorgang zu unterbinden. Fakeshop-Betreibende sind sich dem bewusst und täuschen oftmals Lieferschwierigkeiten vor, um Betroffene an der Unterbindung zu hindern. [19]

### **2.2.2 Arten von Fakeshops**

Es existieren verschiedene Arten von Fakeshops, welche nach aktuellem Stand in vier Kategorien eingeteilt werden können:

- Fake-Onlineshop
- Gefälschte Marken-Shops
- Phishing-Shops
- Phishing-Seiten von echten Online-Shops

#### **Fake-Onlineshops**

Bei einem Fake-Onlineshop handelt es sich wie in Abschnitt 2.2.1 beschrieben, um einen Online-Shop, welcher Waren zu äußerst günstigen Preisen anbietet. Bei den tatsächlichen Waren handelt es sich um Produkte von minderwertiger Qualität oder die Waren werden mitunter gar nicht ausgeliefert. Das Ziel dieser Art von Fakeshops ist das Geld der Opfer. Diese Fakeshops bestehen nur aus Profitgier der Betreibenden.

### **Gefälschte Marken-Shops**

Gefälschte Marken-Shops haben das Merkmal, dass die URL (Uniform Resource Locator) häufig der originalen URL einer bekannten Marke ähnlich sind. Ein Beispiel anhand des Sportartikelherstellers Nike mit der originalen URL [www.nike.com](http://www.nike.com) könnte z.B. [www.nike-outlet.com](http://www.nike-outlet.com) sein. Somit wird dem potenziellen Opfer suggeriert, dass es sich bei dem Online-Shop um einen realen Online-Shop der Marke Nike handelt.

Diese Art von Online-Shops haben große Ähnlichkeiten zum Original und bieten die Produkte zum äußerst günstigen Preis an.

### **Phishing-Shops**

Ein Phishing-Shop tarnt sich ähnlich wie ein Fake-Onlineshop als legitimer Online-Shop, verfolgt aber das Ziel, an die persönlichen Daten wie Kreditkarteninformationen oder Passwörter zu gelangen. Diese Daten werden mitunter dazu verwendet, Identitätsdiebstahl zu begehen oder Bestellungen auf Kosten der Opfer zu tätigen.

### **Phishing-Seiten von echten Online-Shops**

Diese Art von Fakeshops ist eine Kombination aus Gefälschten Marken-Shops und Phishing-Shops. Dabei werden real existierende Marken-Shops imitiert und die aufgesetzten Online-Shops verfolgen keinem monetären Hintergrunde, sondern haben das Ziel, an die persönlichen Daten (Login-Daten, Zahlungsdaten) der Opfer zu gelangen.

In dieser Bachelor-Thesis wird der Fokus auf Fakeshops aus der Kategorie Fake-Onlineshops und Gefälschte Marken-Shops gelegt. Fakeshops aus den Kategorien Phishing-Shops und Phishing-Seiten von echten Online-Shops sind nicht Bestandteil dieser Ausarbeitung.

### **2.2.3 Gesetzliche Grundlage**

Das Betreiben von Fakeshops gilt in Deutschland als Betrug und ist im Strafgesetzbuch (StGB) geregelt. Gemäß § 263 Abs. 1 StGB wird bestraft „wer in der Absicht, sich oder einem Dritten einen rechtswidrigen Vermögensvorteil zu verschaffen, das Vermögen eines anderen dadurch beschädigt, dass er durch Vorspiegelung falscher oder durch Entstellung oder Unterdrückung wahrer Tatsachen einen Irrtum erregt oder unterhält“

Bereits der Versuch des Betruges ist strafbar, was in § 263 Abs. 2 StGB geregelt wird. Wird dabei eine andere Person in wirtschaftliche Not gebracht, so greift § 263 Abs. 3 StGB.

Das Landgericht München I hat 2017 in einem Urteil (19 KLS 30 Js 18/15) entschieden, einen Fakeshop-Betreiber zu einer Gesamtfreiheitsstrafe über fünf Jahren und fünf Monaten zu verurteilen, da der Betreibende mit 19 erstellten Fakeshops insgesamt ca. 430.000 EUR erbeutet hat. In den Fakeshops wurden hochwertige Elektronikartikel angeboten, welche jedoch niemals ausgeliefert wurden. Die Opfer haben die Zahlung per Vorkasse überwiesen.

### **2.2.4 Maßnahmen**

Solange ein Fakeshop online ist, muss damit gerechnet werden, dass Verbraucher Opfer von Betrug werden können. Eine Maßnahme zur Eindämmung ist es, dass ein Fakeshop vom Netz genommen wird. Je nachdem wie der Fakeshop betrieben und in welchem Land der Fakeshop gehostet wird, kann die Außerbetriebnahme einige Zeit andauern. In der Zwischenzeit können die Betreiber den Fakeshop ungehindert weiter betreiben.

Das Bundesamt für Sicherheit in der Informationstechnik (BSI), verschiedene Strafverfolgungsbehörden und der Verbraucherschutz machen auf ihren Internetseiten auf Fakeshops aufmerksam. Es wird versucht, mit präventiven Maßnahmen Verbraucher zu informieren und aufzuzeigen, worauf Verbraucher beim Onlineshopping achten sollten.

Es folgt eine Auflistung der Hinweise, worauf Verbraucher beim Onlineshopping achten sollten und welche durch die o.g. Behörden angegeben werden: [20] [21] [22]

### **Internetadresse**

- URL besitzt Ähnlichkeiten zu einer bekannten Marke und weicht nur leicht vor der richtigen URL ab
- Unübliche Endungen wie -com.de oder -outlet.de
- URL passt nicht zu den Angebotenen Artikel

### **Unsichere Zahlungsweisen**

- Beim Bestellvorgang wird nur Vorkasse angeboten

### **Gefälschte Gütesiegel**

- Erfundene Gütesiegel
- Trusted Shop wird als Gütesiegel angezeigt, Zertifikat aber nicht vorhanden
- Trusted Shop wird als Gütesiegel angezeigt, Weiterleitung erfolgt auf erfundene URL

### **Auffallend günstige Preise**

- Hochpreisige Artikel werden besonders günstig angeboten
- Alle Waren sind deutlich günstiger als in gängigen Online-Shops

### **Schlecht übersetzte AGBs**

- AGB weisen Übersetzungsfehler auf
- AGB wurden von anderen Webseiten kopiert und beinhalten andere Namen von Unternehmen

### **Gefälschte Kundenbewertungen**

- Bewertungen/Erfahrungen sind frei erfunden

### **Impressum**

- Impressum ist nicht vorhanden
- Impressum ist unvollständig

### **Keine HTTPS-Verschlüsselung**

- Die Webseite besitzt kein SSL-Zertifikat und die Domain kann nur unverschlüsselt aufgerufen werden

Wie in Abschnitt 2.1.2 angegeben, verlagert sich das Onlineshopping zunehmend auf das Smartphone. Dies hat den Nachteil, dass Recherchen nach den genannten Hinweisen deutlich aufwendiger und für Verbraucher schwieriger durchzuführen sind. Es ist davon auszugehen, dass die genannten Hinweise für Verbraucher dementsprechend nicht ausreichend genügend sind. Weiter könnten folgende Hinweise und Tipps für Verbraucher aufgenommen werden, um noch besser vor Fakeshops zu schützen:

- Überprüfung, ob der Shop als Fakeshop auf den Portalen auktionshilfe.info und/oder watchlist-internet.at gelistet ist
- Eingabe der URL des Shops über eine Suchmaschine. Zur Überprüfung, ob der Shop bereits als Fakeshop gelistet ist
- Verweis auf den Fakeshop-Finder der Verbraucherzentrale
- Alter der Domain über Dienste wie whois.com überprüfen – Domains geringer 90 Tage weisen auf einen Fakeshop hin
- Domain über Dienste wie nslookup.io überprüfen – Wenn Standort nicht in Deutschland, dann ist Vorsicht geboten
- Fake-Shop Detector von fakeshop.at als Browser-Add-On installieren
- Unternehmen aus Impressum im Handelsregister suchen
- Adresse des Unternehmens über Portalen wie Google Maps suchen und

- überprüfen, ob an dem besagten Ort das Unternehmen einen Sitz hat
- Nur auf sicheren Zahlungsverkehr wie E-Wallets (PayPal) mit Käuferschutz vertrauen
- Überprüfen, ob folgende URL vorhanden ist:
  - Shop URL + /hallo-welt oder /hello-world (WordPress)
  - Shop URL + /defektes-produkt (Shopware)
  - Shop URL + /registerFC/index/sValidation/H (Shopware)

## 2.3 Merkmale

Auf den Internetseiten verschiedener Behörden werden Merkmale (Abschnitt 2.2.4) vorgegeben, an denen Verbraucher einen Fakeshop erkennen können. Neben den präventiven Merkmalen existieren auch technische Merkmale, welche häufiger bei Fakeshops zu finden sind. In der Projektarbeit „Analyse von Fakeshops anhand des jeweiligen eingesetzten Content-Management-System mit Abfrage von gefundenen Merkmalen auf Basis von Python“ wurden allgemeine, System unabhängige Merkmale sowie Merkmale gefunden, welche abhängig vom jeweils eingesetzten System zu finden sind. [4]

Diese Merkmale sind Anzeichen, welche wiederholt bei Fakeshops auftreten.

### 2.3.1 Unabhängige Merkmale

#### **Domain wird im Ausland gehostet**

In der Projektarbeit wurde festgestellt, dass Fakeshops überwiegend im Ausland gehostet werden. Wenn ein Online-Shop deutschsprachigen Inhalt anbietet und dabei im Ausland gehostet wird, gilt dies als Merkmal für einen Fakeshop.

#### **Geringes Alter der Domain**

Ebenfalls konnte herausgefunden werden, dass die Domain von Fakeshops einen kurzen Aktivierungszeitraum inne hat. Wenn ein Online-Shop erst wenige Tage (60 bis 90 Tage) existiert, gilt dies ebenfalls als ein Indiz für einen Fakeshop.



## **Cloudflare Protect**

Ein weiteres Merkmal, welches im Rahmen der Projektarbeit herausgefunden wurde, ist dass der Schutz über den Dienst Cloudflare bei Fakeshops weit verbreitet ist. Cloudflare bietet neben seinem Content-Delivery-Netzwerk (CDN), verschiedener Sicherheitsdienste und der Performance-Optimierung eine Reverse-Proxy-Funktion an, mit welcher die IP-Adresse und damit die Identität des eigentlichen Host-Servers verborgen wird. Als Ziel wird die IP-Adresse eines Cloudflare-Servers angegeben, was die Ermittlung der realen Identität deutlich erschwert.

### **2.3.2 Systemabhängige Merkmale**

WordPress Blog-Artikel Hello World und Hallo Welt

Ein systemabhängiges Merkmal sind die Blog-Artikel Hello World und Hallo Welt, welche als so genannter Demo-Content bei WordPress nach der Installation als Standard installiert werden. Es ist davon auszugehen, dass professionell betriebene Online-Shops diese Blog-Artikel von der Internetseite entfernen. Im Rahmen der Projektarbeit wurden diese Blog-Artikel bei Fakeshops immer wieder vorgefunden.

## **WordPress Blog-Archiv**

Betreibende von Fakeshops nutzen das Grundgerüst von WordPress auch für weitere Fakeshops nutzen. Entsprechend sind so genannte Blog-Archive, älter > 1 Jahr vorhanden. Dabei handelt es sich jedoch nicht um ein aussagekräftiges Merkmal. Sobald aber das Merkmal „Geringes Alter der Domain“ und dieses Merkmal vorhanden sind, könnte dies ein weiterer Hinweis für einen Fakeshop sein.

## **WordPress Plugin Contact Form 7**

Professionell betriebene Online-Shops nutzen in der Regel kostenpflichtige Premium-Plugins. Vor allem dann, wenn es um die Korrespondenz mit

potenziellen Käufern geht. Diese bieten häufig zusätzliche Funktionen und der Nachrichtenfluss kann auch über das WordPress-Backend nachvollzogen werden.

Ein Indiz für einen Fakeshop wurde in der Projektarbeit identifiziert, dass Fakeshops sehr häufig das Standard Kontaktformular-Plugin Contact Form 7 einsetzen. Dies wird lediglich genutzt, um den Schein zu vermitteln, dass Kunden oder potenzielle Käufer sich mit Fragen an die Betreibenden wenden können.

### **Shopware Händler-Login**

Ähnlich wie die Blog-Artikel Hello World und Hallo Welt bei WordPress existiert bei dem System Shopware in der Version 5 nach der Installation Platzhalter-Inhalt. Auf einer Unterseite können sich Händler für potenzielle Partnerschaften registrieren. Bei der Suche nach Merkmalen ist in der Projektarbeit aufgefallen, dass auf dieser Unterseite immer die gleiche Fax-Nummer vorhanden ist. Die angegebene Fax-Nummer gehört zur Shopware AG und dient nur als Platzhalter.

### **Shopware „defektes Produkt“**

Im Verlauf der Projektarbeit wurde eine weitere Unterseite gefunden, welche als Demo-Content installiert wird. Die Unterseite „defektes Produkt“ gibt eine Standard-Formulierung vor, welche bei seriösen Online-Shops nicht vorgefunden wurde.

### **Shopware Cloudflare E-Mail Protection**

Neben den Schutz der IP-Adresse konnte bei Fakeshops mit den Systemen Shopware ein E-Mail-Sicherheitsdienst von dem Unternehmen Cloudflare gefunden werden. Dieser Schutz ist bei der Suche nach Merkmalen vielfach vorgekommen und dient ebenfalls als Indiz für einen Fakeshop.

## 2.4 Automatische Erkennung von Fakeshops

Eine automatische Erkennung von Fakeshops ist nicht ohne weiteres möglich. Anhand von typischen Merkmalen kann eine Vermutung aufgestellt werden, dass es sich bei einem Online-Shop um einen Fakeshop handle. Das es sich dann schlussfolgernd um einen Fakeshop handelt, kann erst dann bestätigt werden, wenn nach einer aufgegebenen Bestellung minderwertige oder keine Ware geliefert wird. Bei einer Klassifikation, ob es sich bei einem Online-Shop um einen Fakeshop handelt, können als Ergebnis zwei bzw. drei Ergebnisse herauskommen, wie die folgenden Beispiele zeigen: Ja, Nein und Vielleicht.

Eine Möglichkeit, die Wahrscheinlichkeit zu ermitteln, ob es sich bei einem Online-Shop um einen Fakeshop handelt, könnte die Klassifikation sein. Das Ziel der Klassifikation besteht darin, neue Daten oder Objekte in bereits existierende Datensätze (Klassen oder Kategorien) einzuordnen. [9]

### 2.4.1 Merkmale und Datenquellen

Um Fakeshops zu klassifizieren, müssen Merkmale identifiziert werden, welche üblich für einen Fakeshop sind. Da diese Merkmale nicht immer auftreten und von Fakeshop zu Fakeshop unterschiedlich sein können, können diese mit Ja (vorhanden) und Nein (nicht vorhanden) oder 1 und 0 in einen Datensatz eingeordnet werden. Aus verschiedenen Datenquellen wie IP-Adresse, Serverstandort, Quellcode, Inhalt des Online-Shops, Bewertungen und weiteren technischen Merkmalen werden Trainingsdaten gebildet, welche den Klassen oder Kategorien zugeordnet werden. Neben Daten von Fakeshops werden zusätzlich auch Daten von seriösen Online-Shops benötigt. Diese sind vor allem dann wichtig, wenn anhand der Daten vorausgesagt werden soll, wie hoch die Wahrscheinlichkeit ist, dass es sich bei einem Online-Shop um einen Fakeshop handelt. Die genannten Merkmale können als Grundlage für die Klassifikation von Fakeshops dienen.

### **2.4.2 Naives Bayes als Klassifikationsverfahren**

Ein auf die Wahrscheinlichkeit basiertes Verfahren ist der Naive-Bayes Algorithmus. Der Algorithmus basiert auf der Bayesschen Formel. Das Ziel des Algorithmus ist die Vorhersage der wahrscheinlichsten Klasse. Anhand von identifizierten Merkmalen werden Trainingsdaten gesammelt, welche sowohl Fakeshops als auch seriöse Online-Shops enthalten. Diese Trainingsdaten werden dem Datensatz zugeordnet. Sind genügend Trainingsdaten vorhanden, so erfolgt die Vorhersage der Wahrscheinlichkeit für einen neuen Online-Shop direkt aus den Trainingsdaten.

Dafür berechnet der Algorithmus für jedes Feature die Wahrscheinlichkeit, dass das Feature in einer bestimmten Klasse vorkommt. Für die Überprüfung eines neuen Online-Shops, werden die Daten erhoben und die Wahrscheinlichkeit der Zugehörigkeit dieser Daten zu einer Klasse berechnet. Als Ergebnis wird eine Vorhersage getroffen, ob es sich anhand der erhobenen Daten um einen Fakeshop handelt. Die Voraussetzung für die Funktion des Naive-Bayes-Algorithmus ist, dass alle Attribute unabhängig voneinander sind. [9]

### **2.4.3 Evaluation der Klassifikation**

Ein wichtiger Schritt, um die Leistung der Klassifikationsmethode zu bewerten ist die Evaluation. Die Ergebnisse müssen regelmäßig überprüft und das Modell bei Bedarf angepasst werden. Dabei sollte überprüft werden, ob die für Fakeshops üblichen Merkmale weiterhin zutreffen. Um die Klassifikationsgenauigkeit bewerten zu können, können verschiedene Metriken wie die Präzision, Recall und F1-Score herangezogen werden.

### **2.4.4 Herausforderungen und zukünftige Entwicklungen**

Es ist davon auszugehen, dass es sich bei der Klassifikation von Fakeshops um ein schnell entwickelndes Forschungsfeld handelt, welches in der Zukunft noch weitere Fortschritte machen wird. Obwohl die Klassifikation zur automatischen Erkennung ein vielversprechender Ansatz ist, gibt es somit auch die Herausforderung die Modelle regelmäßig zu aktualisieren, um neu aufkommende

Betrugsmethoden entgegenzuwirken. Auch legitime Online-Shops entwickeln sich weiter, sei es in ihrer Art und Weise oder auch in ihren Funktionen. Es können neue Systeme auf dem Markt erscheinen, es kann aber auch zu einer Veränderung des Handelns von Waren und Dienstleistungen kommen. Entsprechend müssen zukünftige Entwicklungen stets beobachtet und Modelle gegebenenfalls angepasst werden.

### 3 Methodik

In dieser Ausarbeitung erfolgt die Analyse von Fakeshops auf die verwendeten Systeme mit Hilfe von Python. Hierzu sollen real betriebene Fakeshops untersucht und die identifizierten Informationen erhoben werden. Die Programmiersprache Python soll dazu eingesetzt werden, den Prozess nach Möglichkeit zu automatisieren. In diesem Kapitel wird auf die Methodik eingegangen, bevor im folgenden Kapitel das Vorgehen und der technische Aufbau für die Realisierung behandelt werden.

#### 3.1 Methodik und Inhaltlicher Aufbau

In Abbildung 3 werden die Methodik und der inhaltliche Aufbau zur Beantwortung der Forschungsfragen dargestellt.

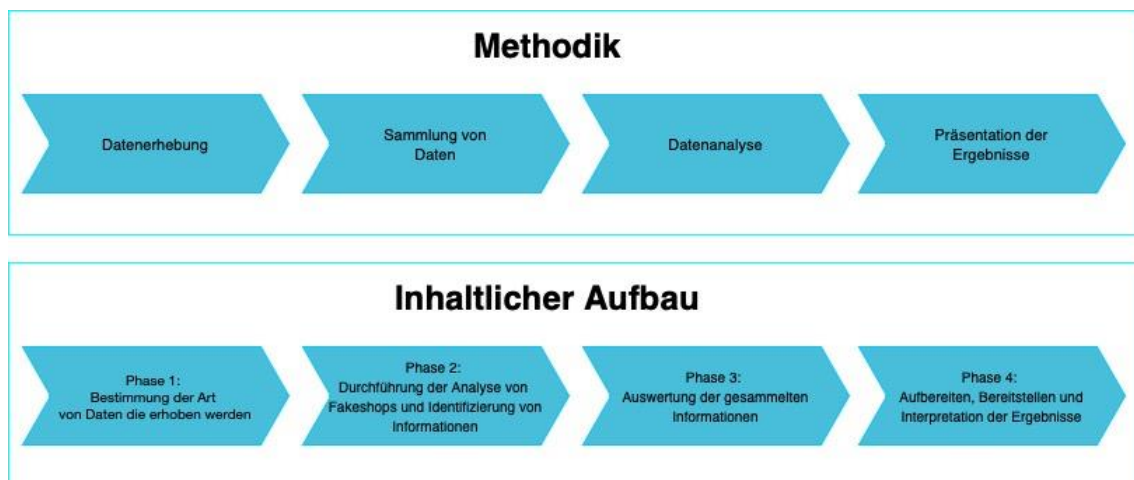


Abbildung 3: Methodik und Inhaltlicher Aufbau

Im ersten Schritt werden die Daten erhoben, welche für die Beantwortung der Forschungsfragen von Relevanz sind. Im zweiten Schritt werden Fakeshops auf die aus Schritt erhobenen Daten analysiert. Diese Daten werden erfasst und zusammengetragen. In der Datenanalyse werden die erfassten Informationen ausgewertet, bevor im vierten und letzten Schritt die Ergebnisse aufbereitet und bereitgestellt werden. Anhand der Ergebnisse sollen die Forschungsfragen

beantwortet werden. Die Ergebnisse können dann auch als Ausgangspunkt für weitere Forschungsarbeiten dienen.

### **3.2 Erläuterung der Forschungsfragen**

Im Folgenden werden die Forschungsfragen, die sich aus dem thematischen Kontext dieser Arbeit ergeben, zunächst benannt. Anschließend wird die Wichtigkeit der Beantwortung der einzelnen Fragen dargelegt.

1. Welche Systeme zum Betreiben eines Fakeshops werden eingesetzt?
2. Welche Zielgruppe anhand der verwendeten Sprache wird favorisiert?
3. In welchen Ländern werden Fakeshops gehostet?

Als Ziel dieser Thesis ist das Auffinden von Gemeinsamkeiten in den Systemen sowie das Anstreben Trends in der Verbreitung von Fakeshops zu erkennen.

#### **3.2.1 Frage 1**

Zum Betreiben eines Online-Shops oder Fakeshops sind wichtige Funktionalitäten wie in Abschnitt 2.1.2 erläutert essenziell. Heute existiert eine Vielzahl von Shop- bzw. Content Management-Systemen, welche sowohl bei legitimen Online-Shops als auch bei Fakeshops eingesetzt werden. Um mögliche Trends bei Fakeshops zu erkennen oder in weiteren Forschungsarbeiten zum Thema Fakeshops eine Zielrichtung vorzugeben, ist ein Überblick über die hier eingesetzten Systeme von hoher Relevanz. Durch Beantwortung dieser Frage können Fakeshops auf systemabhängige Merkmale (siehe Abschnitt 2.3) untersucht und Systeme zur automatischen Erkennung von Fakeshops ergänzt werden.

### **3.2.2 Frage 2**

Anhand der verwendeten nativen Sprache soll die Zielgruppe eines Fakeshops identifiziert werden. In Hinblick auf die Vergleichbarkeit, zwischen der auf deutschsprachige Verbraucher und internationale Verbraucher ausgerichtete Fakeshops kann, somit ein Trend von den verwendeten Systemen aus der ersten Frage im Ergebnis wiedergegeben werden. Die Beantwortung dieser Frage ist essenziell, um generelle Vergleiche zur Ausrichtung der Fakeshops wiedergeben zu können.

### **3.2.3 Frage 3**

Der Standort des Servers, auf welchem ein Fakeshop betrieben wird, soll dahingehend ermittelt werden, um auch hier Trends erkennen zu können. Es kann davon ausgegangen werden, dass Fakeshops überwiegend außerhalb von Deutschland gehostet werden. Dies erschwert Strafverfolgungsbehörden in Deutschland die Ermittlung der Täter bzw. Tätergruppen. Diese Frage steht im engen Zusammenhang mit Frage zwei, da Fakeshops mit der Zielgruppe auf deutschsprachige Verbraucher ebenfalls außerhalb von Deutschland betrieben werden.

### **3.2.4 Ziel**

Das Ziel dieser Thesis ist einen umfassenden Überblick zu erhalten, um Trends in der Verbreitung von Fakeshops zu erkennen sowie das Auffinden von Gemeinsamkeiten im Betreiben von Fakeshops.



## **4 Vorgehen und Aufbau**

Mit den in Kapitel 2 beschriebenen Grundlagen zu Online-Shops und Fakeshops wird in diesem Kapitel auf das Vorgehen zur Analyse von Fakeshops sowie auf den Aufbau zur Datenerhebung- und Sammlung eingegangen.

### **4.1 Vorgehen**

Für einen umfassenden Überblick der verwendeten Systeme und zur Beantwortung der Fragestellung ist es wichtig reale Fakeshops zu analysieren. Identifizierte Fakeshops müssen zeitnah aufgenommen, gespeichert und untersucht werden. Es folgt das Vorgehen, welche Daten relevant sind und wie diese aufgenommen werden sollen.

#### **4.1.1 Auffinden aktueller Fakeshops**

Zum Zeitpunkt der Erstellung dieser Bachelor-Thesis existieren keine empirischen Daten, wie lange ein Fakeshop durchschnittlich online ist.

Es kann jedoch davon ausgegangen werden, dass es sich um wenige Wochen handelt. Besteht der Verdacht, dass es sich bei einem Online-Shop um einen Fakeshop handelt, werden diese auf verschiedenen Portalen im Internet gemeldet. Für das Auffinden aktueller Fakeshops sind zwei Portale von großer Bedeutung.

##### **Forum Auktionshilfe.info**

Das Forum [www.auktionshilfe.info](http://www.auktionshilfe.info) wurde ursprünglich geschaffen, um Nutzern im Internet die Möglichkeit zu geben, sich über Auktionsplattformen auszutauschen. Heute bietet das Forum zusätzlich einen Bereich, in dem sich Nutzer über Gefahren im Internet informieren können. Im Bereich „Online-Shops“ werden von so genannten „Fakeshophuntern“ Online-Shops dahingehend untersucht, ob es sich dabei um Fakeshops handelt. Trifft dies zu, werden diese Fakeshops in dem Forum gemeldet.

Auktionshilfe.info - eBay - PayPal - Kle... Gefahren im Internet und Erfahrungen a...

## Online-Shops

Erfahrungen mit Online-Shops - Fake-Shops | Dubiose Online-Shops | Betrugsserien

<< 1 2 3 4 5 6 7 ... 545 >>

Thema	Antworten	Zugriffe	Letzte Antwort
<b>Fakeshop: gruber-autoreifen.com - Gruber Autoreifen GmbH - Auerstraße 100 - 01640 Coswig - 03523-324817 - 03523-324820 - service@gruber-autoreifen.com - DE433849881 - Fake Shop - Betrug</b> headsman - Freitag, 21:00	2	19	headsman Freitag, 21:02
<b>Fakeshop: singer-autoreifen.com - DS Reifenversand GmbH - Durlacher Straße 63 - 75179 Pforzheim - 07231-630575 - office@singer-autoreifen.com - DE749213729 - Fake Shop - LT893250060767583520 - BIC REVOLT21XXX - LT163120023810018452 BIC UAPULT22</b> headsman - 14. April 2023	20	4.512	The Punisher Freitag, 17:31
<b>Fakeshop: hein-reifenhandel.com - Hein Reifenhandel GmbH - Neue Straße 136 - 06493 Siptenfelde - 039483-167511 - 039483-167514 - service@hein-reifenhandel.com - DE878834737 - Fake Shop - Betrug</b> headsman - Donnerstag, 00:32	4	46	The Punisher Freitag, 17:20


Abbildung 4: Bereich Online-Shops in Forum www.auktionshilfe.info

Dies bietet den Vorteil, dass sobald ein Nutzer über eine Suchmaschine die URL eines enttarnten Fakeshop sucht, in den Suchergebnissen das Forum mit dem Zusatz „Fakeshop“ angezeigt wird. So werden Nutzer davor gewarnt, bei den jeweiligen Shops einzukaufen.

Google oelfabrik24.com

Alle Videos Bilder News Shopping Mehr Suchfilter

10 Ergebnisse (0,22 Sekunden)


**Auktionshilfe.info**  
<https://www.auktionshilfe.info> > ... > Online-Shops >

**Fakeshop oelfabrik24.com – Heizöl Fischer UG ...**

11.05.2023 · 9 Posts · 5 Autoren

Rechtswidriges Impressum, die Daten der darin genannten Firma wurden gestohlen und werden nun für betrügerische Zwecke missbraucht.

Abbildung 5: Eingabe URL von einem Fakeshop in einer Suchmaschine

Das Forum konzentriert sich vor allem auf Fakeshops mit deutschsprachigem Inhalt und gilt für viele weitere Listen von Fakeshops als Ausgangspunkt.

### Portal watchlist-internet.at

Das Portal watchlist-internet.at ist eine unabhängige Informationsplattform zum Thema Internetbetrug. Das Portal informiert Privatpersonen und Unternehmen zu aktuellen Betrugsfällen im Internet und gibt Nutzern Tipps mit, um sich davor zu schützen. Die Umsetzung erfolgt durch das ACR-Institut und das Österreichisches Institut für angewandte Telekommunikation (ÖIAT) in enger Zusammenarbeit mit der Internet Ombudsstelle in Österreich. [23]

Im Bereich „Betrügerische Online-Shops“ werden identifizierte Fakeshops aufgelistet, anhand sich Nutzer über die Fakeshops informieren können.

Ihre Suche lieferte **13762** Treffer

osterreich24rx.com	Seit <b>26.05.2023</b> gelistet	Betrügerische Shops	Mehr Infos
apoteknorsk24.com	Seit <b>26.05.2023</b> gelistet	Betrügerische Shops	Mehr Infos
tastefuloutdoorus.com	Seit <b>26.05.2023</b> gelistet	Betrügerische Shops	Mehr Infos
twoboxing.com	Seit <b>26.05.2023</b> gelistet	Betrügerische Shops	Mehr Infos
plus-running.com	Seit <b>26.05.2023</b> gelistet	Betrügerische Shops	Mehr Infos

Abbildung 6: Liste von Fakeshops auf www.watchlist-internet.at

Wird durch den in Abschnitt 1.1 genannten Fakeshop-Detector ein Online-Shop als potenzieller Fakeshop eingestuft, wird dieser vorab durch Expertinnen und Experten überprüft. Erst dann werden die Fakeshops auf dem Portal gelistet. [24]

Das Portal konzentriert sich auf Fakeshops aus der DACH-Region. Aber auch englischsprachige Fakeshops werden auf dem Portal gelistet.

## Gütesiegelanbieter Trusted Shops

Das in Köln gegründete Unternehmen bietet Online-Shops und Internetnutzern Leistungen wie Gütesiegel, Käuferschutz-Verfahren und ein System von bzw. für Kundenbewertungen. Potenzielle Kunden können sich über die Plattform über Online-Shops informieren und diese bewerten. Unternehmen können mit dem Gütesiegel Kunden aufzeigen, dass es sich um einen legitimen Online-Shop handelt. Da Fakeshops immer häufiger gefälschte Trusted-Shops Gütesiegel verwenden, hat auch Trusted-Shops selbst einen Bereich erstellt, auf welchem Fakeshops gelistet werden. Zusätzlich bietet Trusted Shops einen Scanner, in welchem Nutzer eine URL eines Online-Shops überprüfen können. Nach eigener Überprüfung (29.05.2023 - 07:17 Uhr) wurde jedoch festgestellt, dass dieser Scanner keine korrekten Ergebnisse liefert. Fakeshops, welche auf den o.g. Portalen bereits gemeldet wurden, wurden von dem Scanner nicht identifiziert und als „sicher“ eingestuft.

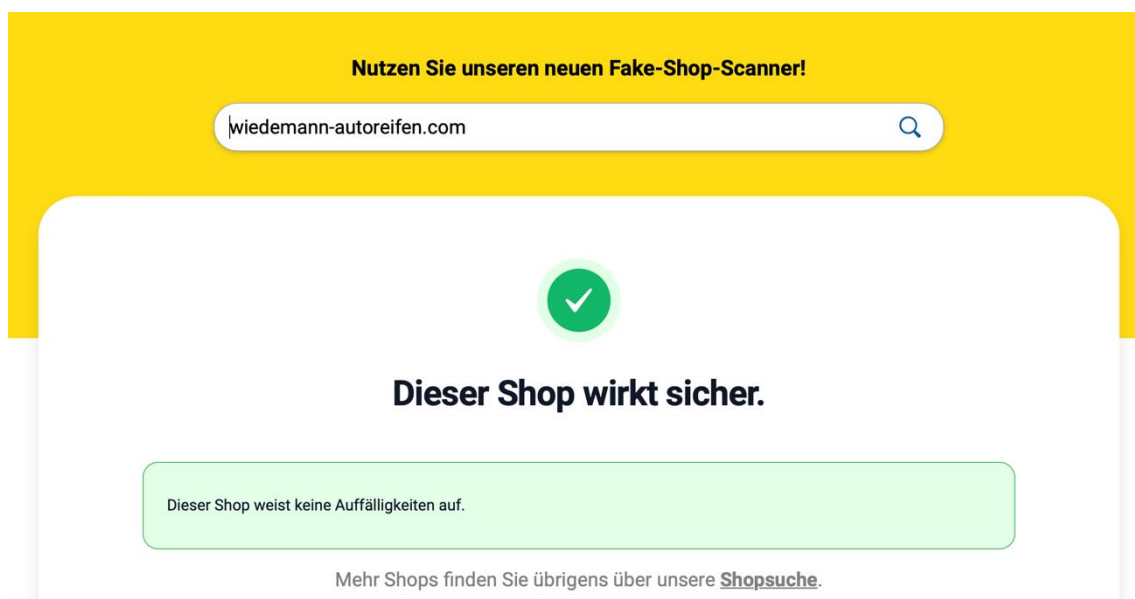


Abbildung 7: wiedemann-autoreifen.com wird als „sicher“ eingestuft



Abbildung 8: wiedermann-autoreifen.com - Gelistet auktionshilfe.info

Da die Auflistung der Fakeshops leicht veraltet ist (> 7 Tage), kann davon ausgegangen werden, dass die aufgeführten Fakeshops nicht eigenständig identifiziert, sondern von anderen Portalen wie die oben genannten übernommen werden.

### Nutzung von Portalen als Ausgangspunkt

Die Portale [www.auktionshilfe.info](http://www.auktionshilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) zeigen, dass die dort geführten Fakeshops durch verschiedene Methoden überwiegend selbstständig identifiziert und aufgenommen werden. Es ist nicht auszuschließen, dass die Portale als weitere Informationsquelle gegenseitig aufeinander zugreifen, was jedoch für diese Ausarbeitung kein Problem darstellt. Für das Vorgehen werden die auf den Portalen geführten Fakeshops als Ausgangslage genutzt und gelten im Rahmen dieser Thesis als identifizierter Fakeshop, sobald dieser gelistet wurde.

#### 4.1.2 Auswahl relevanter Daten

Zur Beantwortung der Forschungsfragen aus der Aufgabenstellung müssen unterschiedliche Daten erhoben werden. Dazu werden Fakeshops auf folgende Daten überprüft:

- Domain des Fakeshops
- Datum, wann der Fakeshop aufgenommen wurde
- Status, ob der Fakeshop Online bzw. Offline war
- Die verwendete native Sprache
- Das verwendete Shop- bzw. Content Management-System

- Version des Systems
- Mögliche Erweiterung/Plugin mit Shop-Funktion
- Server-Standort
- IP-Adresse des Servers
- Überprüfung, ob Cloudflare genutzt wird

### **Domain des Fakeshops**

Als Ausgangslage für die Domain eines Fakeshops werden die Portale [www.auktionhilfe.info](http://www.auktionhilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) genutzt. Neu gemeldete Fakeshops werden zur Untersuchung aufgenommen.

### **Datum, wann der Fakeshops aufgenommen wurde**

Sobald ein Fakeshop aufgenommen wurde, wird das jeweilige Tagesdatum ausgewählt.

### **Status, ob der Fakeshop Online bzw. Offline war**

Bei der ersten Überprüfung eines Fakeshops wird der Status überprüft. Wenn der Fakeshop online ist, wird der Status „Online“ gesetzt. Andernfalls wird „Offline“ gesetzt.

### **Die verwendete native Sprache**

Um die Zielgruppe des Fakeshops zu bestimmen, wird die native Sprache (Muttersprache) erhoben. Dazu ist eine API (Application Programming Interfaces) zu einem Anbieter notwendig, auf welchen in Abschnitt 4.1.3 näher eingegangen wird.

### **Das verwendete Shop- bzw. Content Management-System**

Wie in den Abschnitten 2.1.4 und 2.1.5 angegeben, existieren heute eine Vielzahl von Möglichkeiten, einen Online-Shop zu betreiben. Zur Bestimmung des

verwendeten Systems wird mittels API eine Abfrage durchgeführt, um auch zum Teil weniger bekannte Systeme identifizieren zu können.

### **Version des Systems**

Da bei Shopsystemen wie zum Beispiel Shopware in den Versionen 5 und 6 markante Unterschiede existieren, werden von allen genutzten Systemen, sofern möglich und existent, die für das eingesetzte System verwendete Version mit aufgenommen.

### **Mögliche Erweiterung/Plugin mit Shop-Funktion**

Eingesetzte Systeme wie das Content-Management-System WordPress werden häufig für den Aufbau von Internetseiten genutzt. Da das CMS WordPress aber keine Shop-Funktion bietet, ist es essenziell, zusätzlich zum verwendeten System einen Fakeshop dahingehend zu überprüfen, ob eine Shop-Erweiterung in Form eines Plugins oder ähnliches eingesetzt wird.

### **IP-Adresse des Servers**

Zur Ermittlung des Server-Standortes ist die IP-Adresse des Servers maßgeblich. Zusätzlich könnte die Möglichkeit bestehen, dass auf einem Server auch andere Fakeshops betrieben/gehostet werden.

### **Server-Standort**

Da Fakeshops überwiegend im Ausland gehostet werden, wird zusätzlich mithilfe einer API und der IP-Adresse des Servers überprüft, in welchem Land sich der Server befindet.

### **Überprüfung, ob Cloudflare genutzt wird**

Mit dem Dienst Cloudflare kann der reale Standort eines Servers und die Identität der Betreibenden verschleiert werden. Fakeshops werden dahingehend

analysiert, ob der Dienst Cloudflare aktiv eingesetzt wird.

#### **4.1.3 Nutzung von APIs zur Ergänzung der Analysedaten**

Da nahezu täglich neue Fakeshops auf den Portalen gemeldet werden, müssen die Fakeshops zeitnah analysiert werden. Dafür werden verschiedene Dienste verwendet, um an die notwendigen Informationen zu gelangen. Die Dienste bieten zusätzlich Schnittstellen an, welche in die Teilautomatisierung integriert werden sollen. Es folgt eine kurze Beschreibung der Dienste, welche im Rahmen dieser Thesis verwendet sowie eine Auflistung, welche Daten durch die eingesetzten Dienste erhoben werden.

##### **MeaningCloud**

Mit dem Anbieter MeaningCloud ([www.meaningcloud.com](http://www.meaningcloud.com)) wird ein Software-as-a-Service (SaaS) Produkt bereitgestellt, welches Textanalysen und semantische Verarbeitung in Systemen anbietet. Der Dienst bietet eine Vielzahl von Funktionen, um Text zu analysieren. Anhand von Textklassifizierungen können Texte wie Nachrichten zum Beispiel in Kategorien wie Politik, Sport, Wirtschaft oder weitere eingeteilt werden. Ein weiterer Dienst ist die Stimmungsanalyse, wodurch in sozialen Netzwerken, Bewertungen oder Umfragen die Stimmung von Kunden eingefangen und kategorisiert werden kann. Der Anbieter bietet noch viele weitere Dienste an, auf welche hier nicht näher eingegangen wird. Zur Analyse, welche native Sprache bei einem Fakeshop eingesetzt und welche Zielgruppe damit verfolgt wird, wird im Rahmen dieser Thesis die Sprachidentifikations-API von MeaningCloud genutzt. Anhand der API können Textinhalte aus einer Liste mit mehr als 180 Sprachen identifiziert werden. [25]

Daten die mit MeaningCloud erhoben werden:

- Die verwendete native Sprache



## **WhatCMS**

Über den Anbieter What CMS ([www.whatcms.org](http://www.whatcms.org)) kann das verwendete System einer Internetseite identifiziert werden. Mit der Eingabe einer URL werden Informationen wie das verwendete System, mögliche Erweiterungen, Version, Programmiersprache, Informationen zur Datenbank und zum Webserver offengelegt. Der Dienst bietet zusätzlich eine kostenpflichtige API, mit welcher Daten automatisiert, abgefragt werden können. Wird eine Domain abgefragt, so wird das Ergebnis der Abfrage in Form einer JSON-Datei zurückgegeben und kann ausgelesen werden.

Für die Erhebung der notwendigen Informationen wird die kostenpflichtige API genutzt um folgende Daten von Fakeshops möglich automatisiert zu identifizieren:

- Das verwendete Shop- bzw. Content-Management-System
- Version des Systems
- Mögliche Erweiterung/Plugin mit Shop-Funktion
- Überprüfung, ob Cloudflare genutzt wird

## **IP-Stack**

Anhand des Internetdienst ipstack ([www.ipstack.com](http://www.ipstack.com)) können Informationen zu einem Server aufgedeckt werden. Notwendig dafür ist eine IP-Adresse, anhand welcher anschließend der Standort (Kontinent, Land, Region, Stadt, Postleitzahl, Längen- und Breitengrad) und die Zeitzone angezeigt werden. Auch dieser Dienst bietet eine kostenpflichtige Schnittstelle, mit welcher Abfragen automatisiert, durchgeführt werden können. Wird eine IP-Adresse über die Schnittstelle abgefragt, so werden die Informationen in einer JSON-Datei zurückgegeben. Für die Ausarbeitung dieser Thesis ist der Standort des Server von Bedeutung, welcher hier als Information über die kostenpflichtige API erhoben wird:

- Server-Standort

#### **4.1.4 Extrahierung der Informationen von Fakeshops**

Die manuelle Extrahierung der Daten ist zeitaufwendig und kann zusätzlich zu fehlerhaften Resultaten führen, da eine Vielzahl von Shop- und Content-Management-Systemen existiert, welche fälschlicherweise nicht als solches erkannt werden könnten. Um den Prozess der Datenerhebung möglichst effizient zu gestalten, ist eine Teilautomatisierung unabdingbar. Dafür werden verschiedene Python-Skripte erstellt, auf welche in Abschnitt 4.2 näher eingegangen wird. Die identifizierten Informationen werden anschließend in einer Datenbank gespeichert.

#### **4.1.5 Überprüfung der Daten**

Die Daten werden im Zeitraum der Erhebung täglich kontrolliert und auf mögliche Fehler überprüft. Eine tägliche Überprüfung ist notwendig, da ein Fakeshop bereits nach kurzer Zeit wieder offline sein könnte. Werden Daten nicht korrekt erkannt oder weisen Unstimmigkeiten auf, so werden manuelle Nacharbeiten am Datenbestand durchgeführt. Es kann beispielsweise vorkommen, dass ein System nicht korrekt oder falsch erkannt wird. Das ist vor allem dann der Fall, wenn der Dienst Cloudflare eingesetzt und möglichst viele Informationen verschleiert werden. Auch die Version eines Systems wird nicht immer korrekt ermittelt. In den manuellen Nacharbeiten wird der Quellcode einer Webseite analysiert und der Datenbestand händisch korrigiert. Zusätzlich wird der Dienst von dem Anbieter [www.builtwith.com](http://www.builtwith.com) zur Nachkontrolle eingesetzt, auf welchem anhand einer URL-Informationen zum System, mögliche Erweiterungen und zur Sprache eingesehen werden können. Dieser Dienst wird ebenfalls für Stichproben verwendet, welche zusätzlich zu den manuellen Nacharbeiten mehrfach pro Woche durchgeführt werden.

### **4.2 Aufbau**

Wie in Abschnitt 4.1 beschrieben, folgt nun der Aufbau der Umgebung. Um die Anforderungen zur Analyse der Daten zu erfüllen, werden eine Datenbank zur Speicherung der Daten sowie verschiedene Python-Skripte erstellt. Es wird eine Datenbank auf einem Raspberry Pi mit dem Betriebssystem Ubuntu 20.04 LTS

eingrichtet. Zusätzlich werden die Python-Skripte auf dem Raspberry Pi ausgeführt.

Die technische Umgebung, wie der Aufbau erstellt wird, ist in Abbildung 9 zu erkennen.

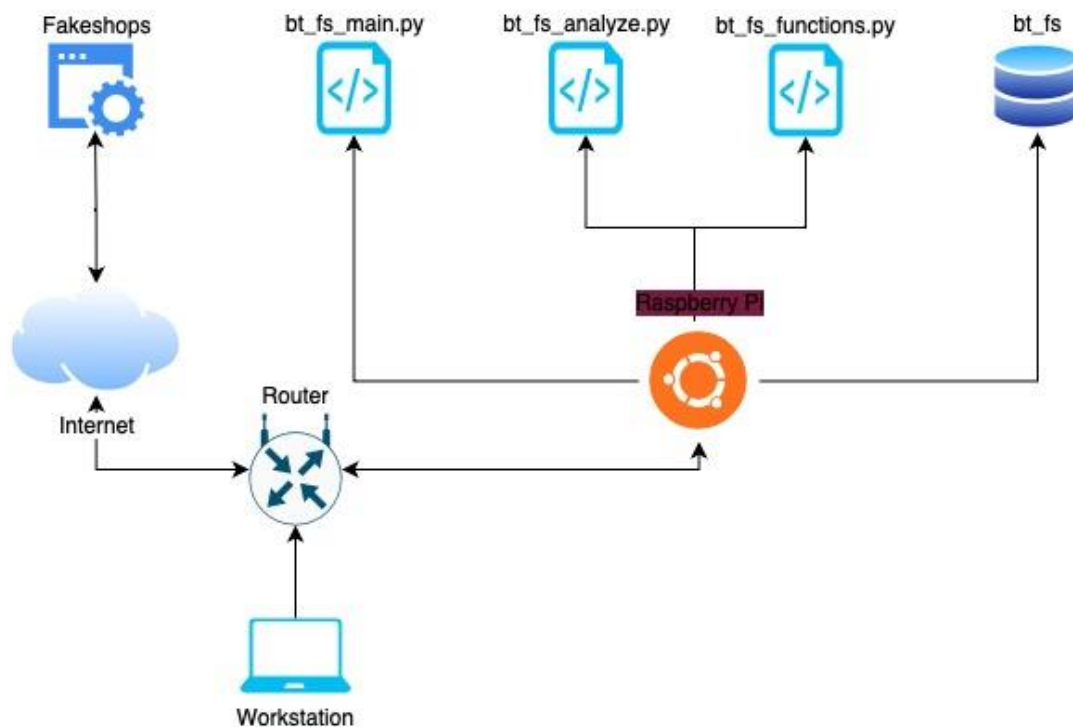


Abbildung 9: Aufbau tech. Umgebung

#### 4.2.1 Datenbankumgebung

Für die Realisierung wird MariaDB in der Version 10.11.2 eingesetzt, da die Syntax von MySQL und MariaDB größtenteils identisch ist. MariaDB wurde als Fork von MySQL entwickelt und basiert auf dem gleichen SQL-Dialekt. [26]

Zur Speicherung der Daten wird eine Datenbank mit der Bezeichnung *bt\_fs* (bachelorthesis fakeshops) erstellt, welche eine Tabelle mit der Bezeichnung *fakeshops* enthält.

Die Tabelle enthält folgende Columns:

- ID (Primärschlüssel)
- Domain
- Datum
- Status
- Sprache
- System
- Version
- Erweiterung
- IP
- Standort
- Cloudflare

Im Rahmen dieser Thesis wird eine einfache Datenbankumgebung mit einer einzigen Tabelle verwendet. In diesem Fall wird auf die Normalisierung der Daten verzichtet, da es sich um eine reine Speicherung von Daten handelt. Diese Entscheidung ermöglicht eine einfachere und schnellere Datenabfrage, jedoch auf Kosten der Datenkonsistenz und der Vermeidung von Redundanz.

#### **4.2.2 Teilautomatisierung mit Python**

Um den Prozess der Analyse von Fakeshops umzusetzen, wird die Programmiersprache Python eingesetzt. Im Zuge dieser Thesis wurde sich für Python entschieden, da die Programmiersprache sich zum scraping (Auslesen von Informationen) von Webseiten eignet. [27]

Python ist vielseitig und eignet sich vor allem für Automatisierungsprozesse. Es existieren eine Vielzahl an Python-Bibliotheken, welche die Erstellung der notwendigen Skripte vereinfacht. Zum Auslesen der Fakeshops werden die Bibliotheken Requests und BeautifulSoup verwendet, um die Daten (XML- und HTML-Dokumente) zu parsen und zu extrahieren. [28]

Für die Teilautomatisierung werden folgende Python-Skripte erstellt:

- `bt_fs_main.py`
- `bt_fs_analyze.py`
- `bt_fs_functions.py`

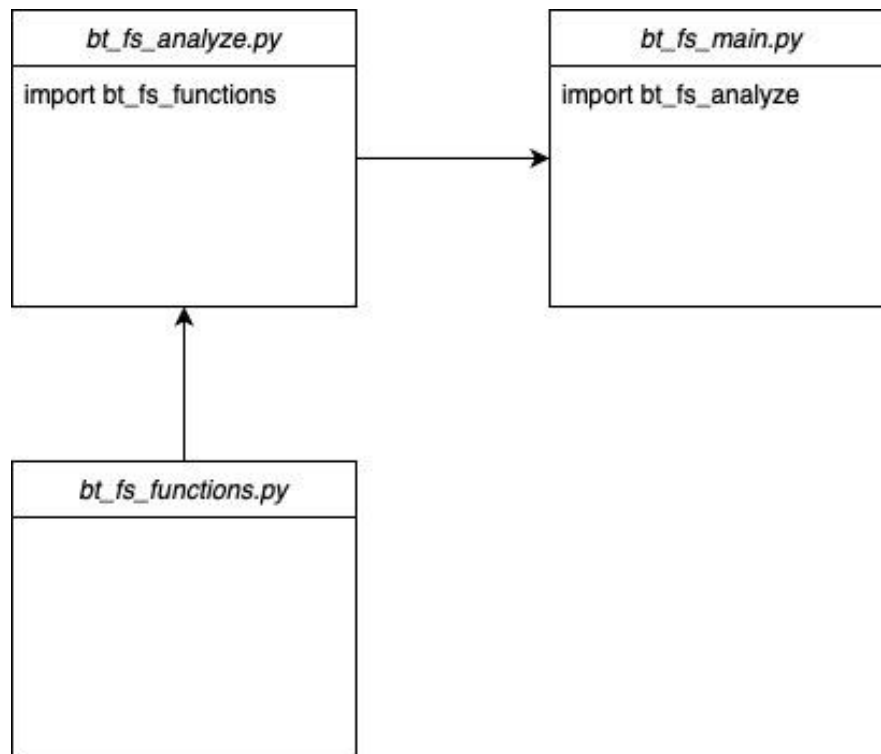


Abbildung 10: Zusammenhang Python-Skripte

Wie in Abbildung 10 dargestellt, ist das Python-Skript `bt_fs_main.py` das Haupt-Skript, welches zur Ausführung gestartet werden muss. Das Skript greift auf eine Funktion aus dem Skript `bt_fs_analyze.py` zu. In der angesprochenen Funktion sind die einzelnen Funktionen enthalten, die in dem Python-Skript `bt_fs_functions.py` definiert wurden. Dieses Vorgehen hat den Vorteil, dass jederzeit neue Funktionen hinzugefügt oder veraltete entfernt werden können und die Skalierbarkeit für das Vorgehen gegeben ist.

Zur Verdeutlichung wird in den nächsten Abschnitten auf die einzelnen Skripte und Funktionen eingegangen.

### 4.2.3 Python-Skript `bt_fs_main.py`

Mit dem Skript `bt_fs_main.py` wird der Ausgangspunkt zum Starten der Analyse gesetzt. Dafür wird auf die Inhalte des Skriptes `bt_fs_analyzse.py` zugegriffen. Der Aufbau wird in Abbildung 11 aufgezeigt.

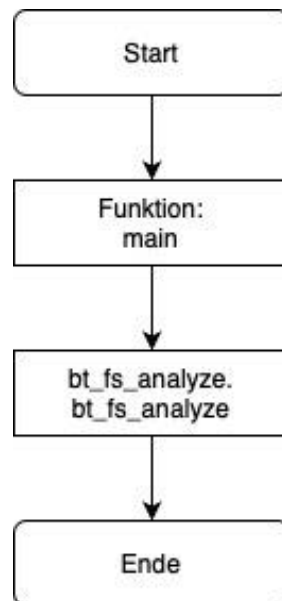


Abbildung 11: Ablaufdiagramm Skript `main.py`

Anschließend folgt der Inhalt des Skriptes `bt_fs_main.py`:

```
import bt_fs_analyze

def main():
    bt_fs_analyze.bt_fs_analyze()
```

#### 4.2.4 Python-Skript `bt_fs_analyze.py`

Das Skript `bt_fs_analyze.py` enthält eine Funktion, in der die einzelnen Funktionen durchgegangen werden, welche im Python-Skript `bt_fs_functions.py` definiert wurden. Dafür wird das Skript über die Import-Funktion hinzugefügt. Das hat den Vorteil, dass neue Funktionen sehr einfach hinzugefügt, entfernt oder ausgetauscht werden können. Jede Funktion wird nacheinander durchgegangen, was in Abbildung 12 verdeutlicht wird.

Das Skript dient der Vereinfachung und Skalierung, um Änderungen/Anpassungen zeitnah umsetzen zu können. Auf den Inhalt, Aufbau und Ablauf der Funktionen zur Analyse von Fakeshops wird in Abschnitt 4.2.5 eingegangen.

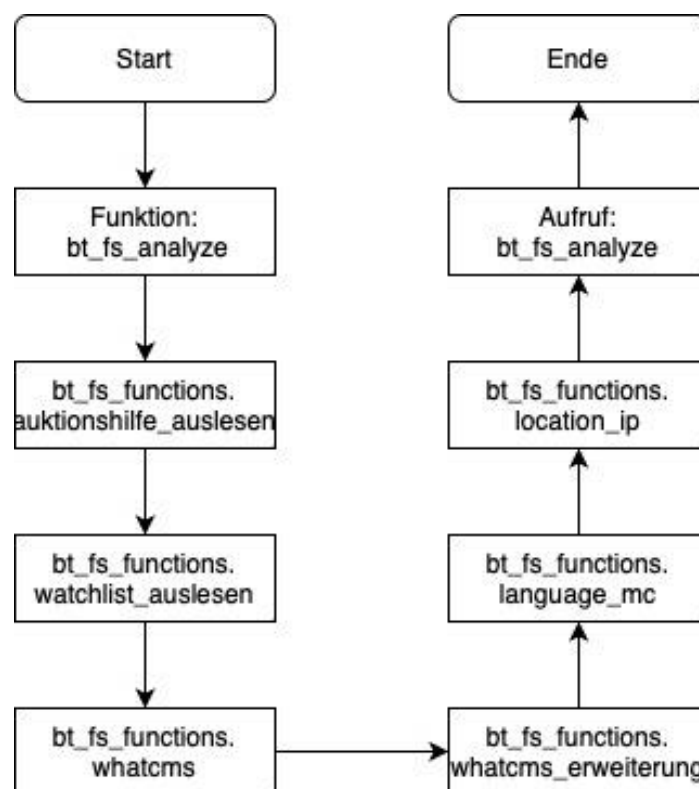


Abbildung 12: Ablaufdiagramm `bt_fs_analyze.py`

```
import bt_fs_functions

print("-----")
print("Starte Analyse von Fakeshops")
print("-----")

# Funktion, welche Funktionen aus bt_fs_functions aufruft

def bt_fs_analyze():
    # Starte Funktionen aus
    bt_fs_functions.auktionshilfe_auslesen()
    bt_fs_functions.watchlist_auslesen()
    bt_fs_functions.whatcms()
    bt_fs_functions.whatcms_erweiterung()
    bt_fs_functions.language_mc()
    bt_fs_functions.location_ip()

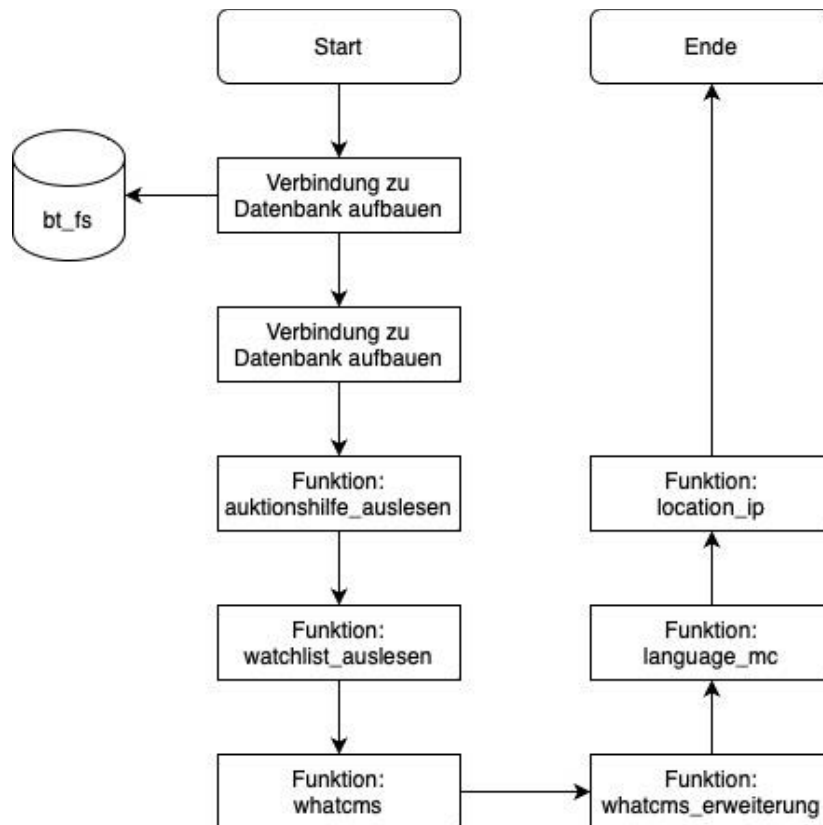
bt_fs_analyze()
```

#### 4.2.5 Python-Skript bt\_fs\_functions.py

Mit dem Skript bt\_fs\_functions.py werden alle notwendigen Funktionen für die Analyse von Fakeshops definiert. Das Skript besteht aus folgenden Funktionen:

- auktionshilfe\_auslesen
- watchlist\_auslesen
- whatcms
- whatcms\_erweiterung
- language\_mc
- location\_ip



Abbildung 13: Aufbau Skript `bt_fs_functions.py`

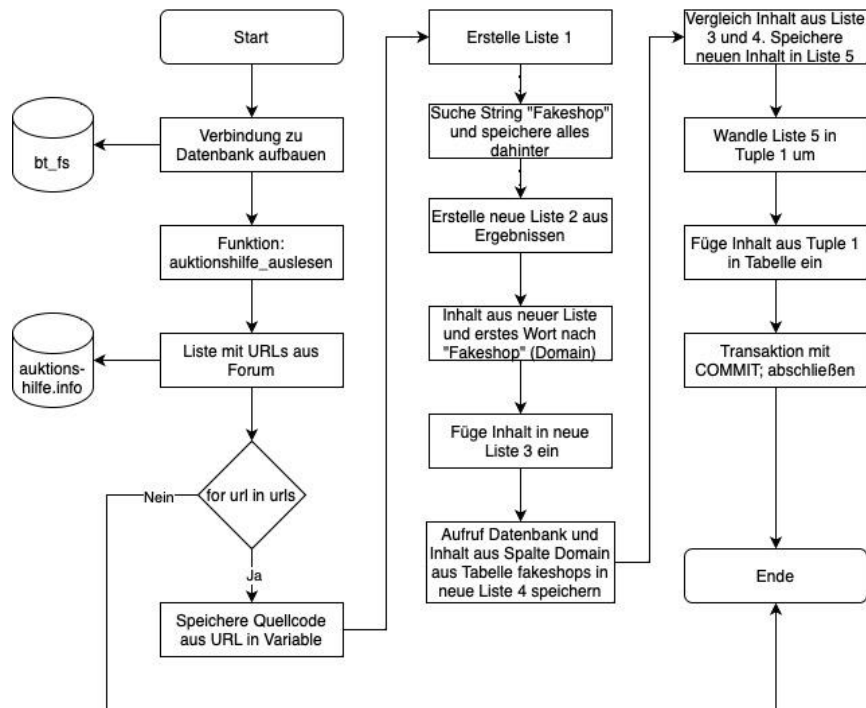
Der Programmiercode wird in Anhang A.1 wiedergegeben.

Folgend wird auf die Funktionen zur Analyse der Fakeshops eingegangen.

### Funktion `auktionshilfe_auslesen`

Die Funktion `auktionshilfe_auslesen()` liest die Domains der Fakeshops aus dem Forum `www.auktionshilfe.info` aus. Dazu wird eine Liste erstellt, welche folgende URLs enthält:

- <https://www.auktionshilfe.info/board/22-online-shops/?pageNo=1>
- <https://www.auktionshilfe.info/board/22-online-shops/?pageNo=2>
- <https://www.auktionshilfe.info/board/22-online-shops/?pageNo=3>
- <https://www.auktionshilfe.info/board/22-online-shops/?pageNo=4>
- <https://www.auktionshilfe.info/board/22-online-shops/?pageNo=5>

Abbildung 14: Ablauf Funktion `auktionshilfe_aufllesen`

Jede URL wird durch eine for-Schleife einmal untersucht. Dabei wird die URL aufgerufen und der Quellcode in einer Variable gespeichert. Die Variable wird anschließend nach den Posts im Forum durchsucht und die Domains der Fakeshops aus den Titeln der Forum-Posts extrahiert. Die Domains werden in einer neuen Liste gespeichert.

Nun wird auf die Datenbank mit der Tabelle *fakeshops* zugegriffen und alle vorhandenen Domains aus der Datenbank in einer Liste gespeichert.

Anschließend werden beide Listen miteinander verglichen und nur die Fakeshops, welche noch nicht in der Datenbank vorhanden sind, in eine neue Liste gespeichert.

Diese Liste wird dann in ein Tuple umgewandelt, damit die neuen Domains der Datenbank mit einem INSERT INTO hinzugefügt werden können.

Abschließend wird mit einem COMMIT; die Transaktion abgeschlossen und festgeschrieben.

Der Programmiercode der Funktion `auktionshilfe_aufllesen()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.

### **Funktion watchlist\_auslesen**

Mit der Funktion `watchlist_auslesen()` werden ähnlich wie bei der Funktion `auktionshilfe_auslesen()` die Domains der Fakeshops von der Webseite `www.watchlist-internet.at` ausgelesen. Zum Auslesen der Domains wird eine Liste erstellt, welche folgende URLs enthält:

- <https://www.watchlist-internet.at/liste-betruegerischer-shops/>
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=2](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=2)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=3](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=3)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=4](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=4)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=5](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=5)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=6](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=6)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=6](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=6)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=7](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=7)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=8](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=8)
- [https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx\\_solr%5Bpage%5D=9](https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=9)

Wie bei der Funktion `auktionshilfe_auslesen()` wird mit einer `for`-Schleife jede URL einmal untersucht. Der Quellcode wird in einer Variable gespeichert und anschließend werden Links untersucht, welche der Klasse `„site-item__link“` angehören. In diesem String befinden sich die Domains der Fakeshops, welche folgend extrahiert und in einer neuen Liste gespeichert werden. Der weitere

Ablauf ist identisch zur Funktion `auskursionshilfe_auslesen()` und wird in Abbildung 15 verdeutlicht.

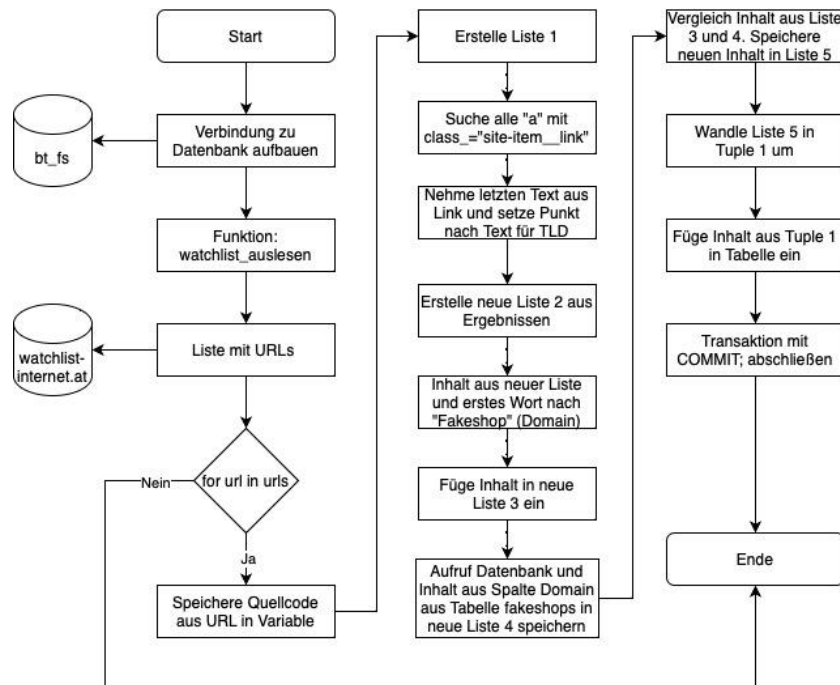


Abbildung 15: Ablauf Funktion `watchlist_auslesen()`

Der Programmiercode der Funktion `watchlist_auslesen()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.

## Funktion `whatcms`

Mit der Funktion `whatcms()` wird das Shop- bzw. Content-Management System und die jeweilige Version eines Fakeshops untersucht. Zusätzlich wird überprüft, ob der Dienst Cloudflare zum Einsatz kommt.

Dazu werden die Domains der Datenbank `fakeshops` aufgerufen, die in der Column System keinen Eintrag besitzen. Diese Domains werden in einer Liste gespeichert. Es folgt die Einleitung einer Schleife mit einem Indexwert und den Domains aus der vorher erstellten Liste. Anschließend wird mit einem API-Key die Verbindung zum Dienst von `www.whatcms.org` hergestellt. Die Domains der Fakeshops werden nacheinander über den Dienst überprüft. Gibt `whatcms.org`

den Status Code 201 zurück, so wurde kein System und Version erkannt und die Tabelle der Datenbank wird dahingehend mit „Unbekannt“ aktualisiert. Wird der Status Code 202 zurückgegeben, so ist der Fakeshop bereits offline und der Status in der Tabelle wird auf „offline“ und alle anderen Columns auf „Unbekannt“ gesetzt.

Mit dem Status Code 200 wurde ein System erkannt und der Status wird auf „Online“ gesetzt. Anschließend wird das System ausgelesen und die Version bestimmt. Danach wird der Fakeshop dahingehend überprüft, ob der Dienst Cloudflare eingesetzt wird. Mit den Ergebnissen wird die Tabelle in der Datenbank aktualisiert und es folgt die Überprüfung der nächsten Domain, bis alle Domains aus der zu Anfang erstellten Liste überprüft wurden.

Der Programmiercode der Funktion `whatcms()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.

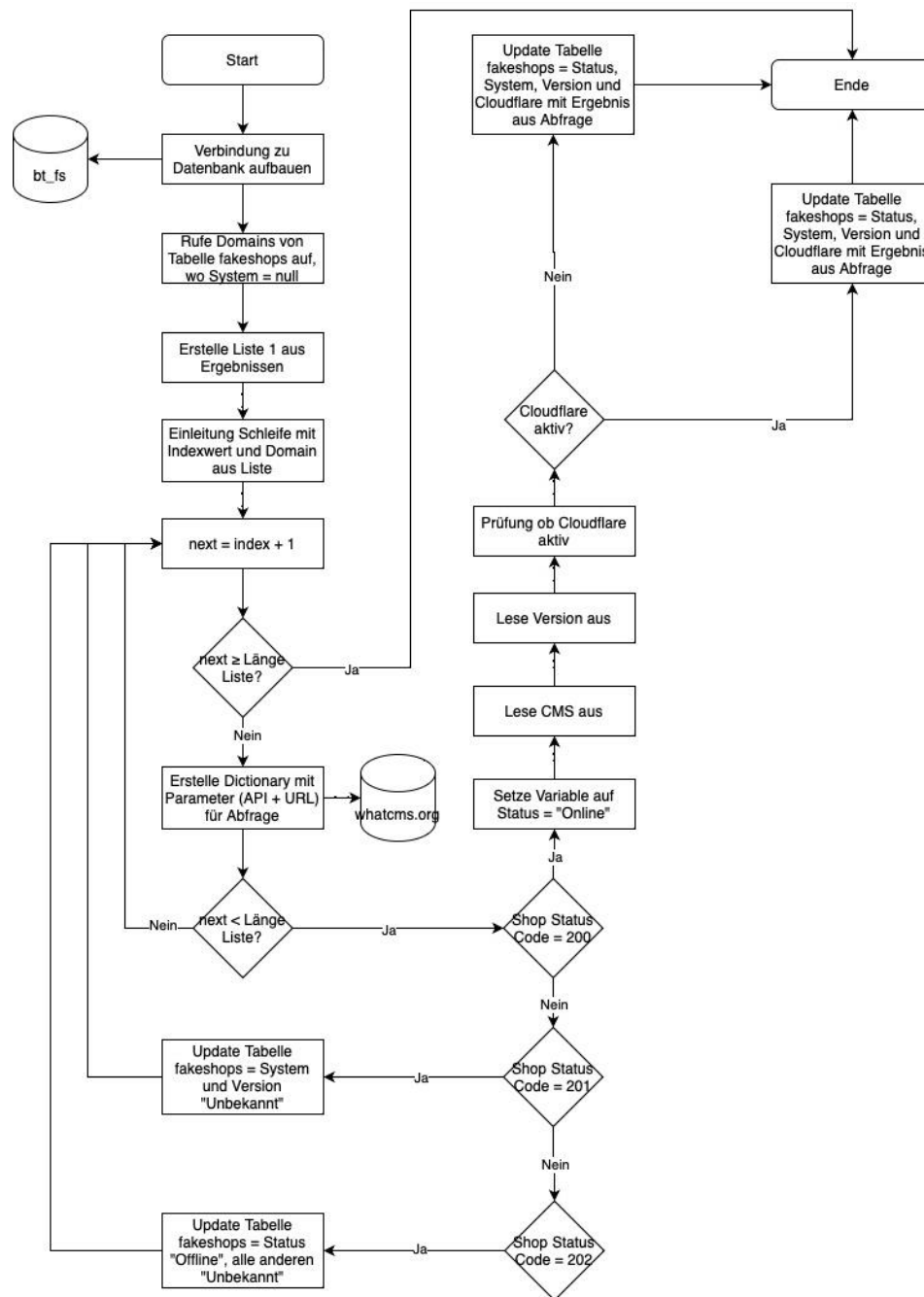


Abbildung 16: Ablauf Funktion whatcms()

### Funktion whatcms\_erweiterung

Mit der Funktion `whatcms_erweiterung` wird das Content-Management-System WordPress dahingehend überprüft, ob ein Plugin als Erweiterung zur Shop-Funktion eingesetzt wird.

Dazu werden die Domains aus der Tabelle `fakeshops` aufgerufen, welche das System „WordPress“ besitzen und gleichzeitig die Column Erweiterung leer ist.

Diese Domains werden in einer Liste gespeichert. Mit der Einleitung einer Schleife mit einem Indexwert und der Domains aus der Liste wird die Überprüfung einer Domain gestartet. Dazu wird mit dem API-Key der Dienst [www.whatcms.org](http://www.whatcms.org) aufgerufen und der Status Code der Domain überprüft. Ist der Status Code 200, dann wird die Erweiterung ausgelesen und die Tabelle fakeshops dahingehend aktualisiert. Ist der Status Code nicht 200, so wird die nächste Domain aus der Liste gewählt und die Überprüfung erneut gestartet. Wurden alle Domains überprüft, so endet die Funktion.

Der Programmiercode der Funktion `whatcms_erweiterung()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.

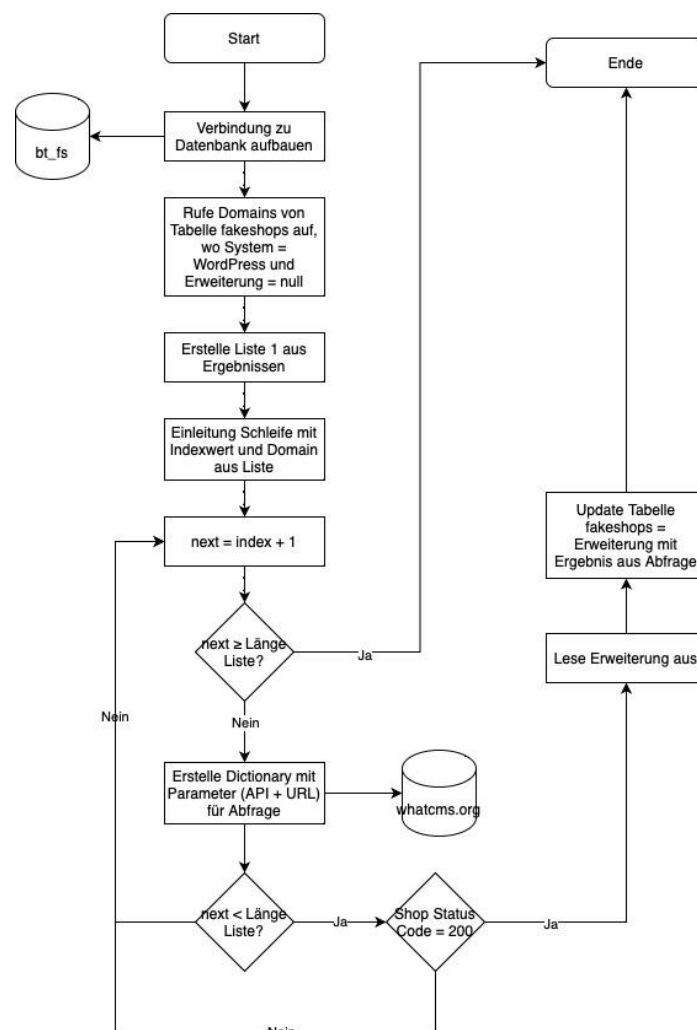


Abbildung 17: Ablauf Funktion `whatcms_erweiterung()`

**Funktion `language_mc`**

Die Funktion `language_mc()` dient dazu, dass die native Sprache und damit die Zielgruppe eines Fakeshops identifiziert wird.

Der Ablauf der Funktion ähnelt dem der vorherigen Funktionen. Es wird eine Verbindung zur Datenbank hergestellt und alle Domains der Tabelle `fakeshops` aufgerufen, welche in der Column `Sprache` keinen Inhalt besitzen. Diese Domains werden in einer Liste gespeichert. Mit der Einleitung einer Schleife wird ein Indexwert erstellt und die Domains aus der Liste nacheinander überprüft. Zur Überprüfung wird mit einem API-Key die Verbindung zu `www.meaningcloud.com` hergestellt. Über den Dienst wird der Fakeshop aufgerufen und der Text analysiert. Das Ergebnis wird anschließend in der Tabelle aktualisiert. So werden alle Fakeshops nacheinander analysiert und die Funktion endet dann, wenn alle Domains aus der Liste überprüft wurden.

Der Programmiercode der Funktion `language_mc()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.



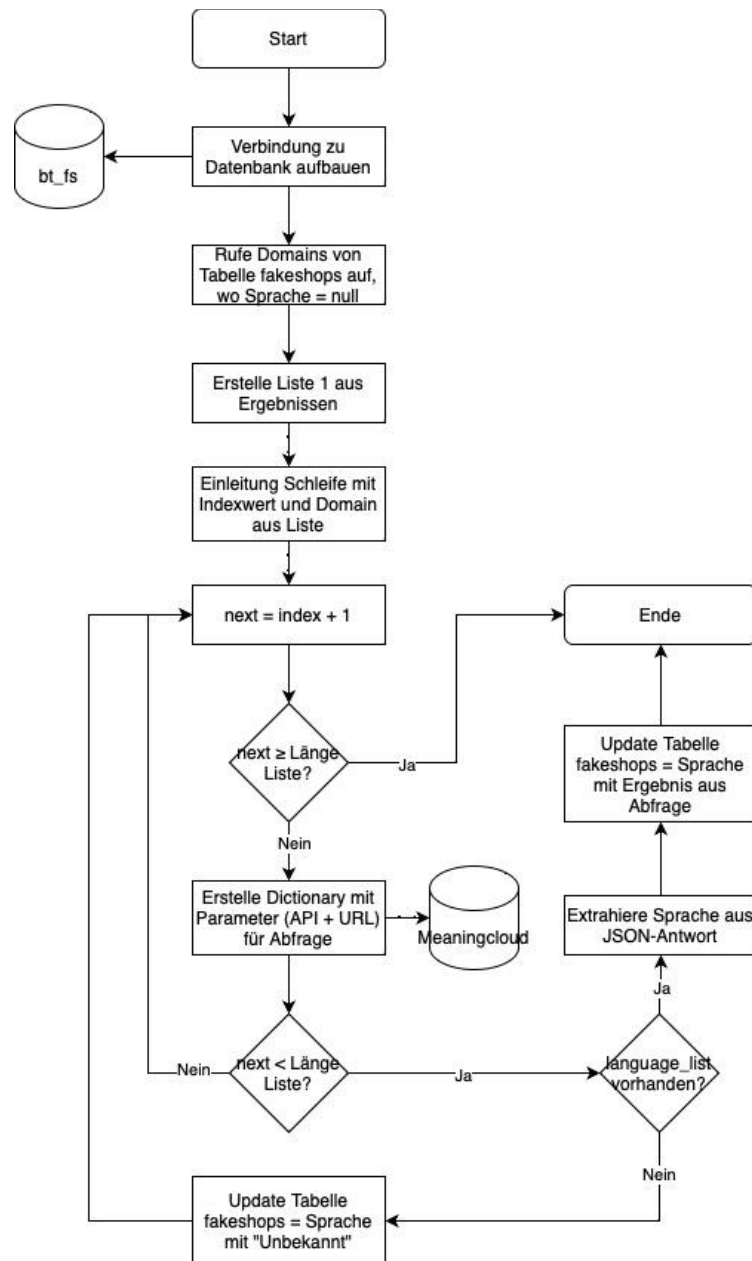


Abbildung 18: Ablauf Funktion language\_mc

### Funktion location\_ip

Die Funktion `location_ip()` dient dazu, die IP-Adresse und den Server-Standort eines Fakeshops zu identifizieren. Dafür wird eine Verbindung mit der Datenbank aufgebaut und die Tabelle `fakeshops` aufgerufen. Alle Domains, wo der Standort keinen Eintrag besitzt, werden in einer Liste gespeichert. Auch hier folgt die Einleitung einer Schleife mit einem Indexwert und Domains aus der Liste. So wird nacheinander über den Dienst `www.ipstack.com` der Standort des Servers

identifiziert und anschließend IP-Adresse und das Land des Server-Standortes in der Tabelle aktualisiert. Wird dabei der HTTP-Status Code 200 nicht zurückgegeben, so ist der Fakeshop nicht mehr zu erreichen und der Standort und die IP-Adresse werden in der Tabelle mit „Unbekannt“ aktualisiert. Die Analyse der Fakeshops erfolgt in einer sequenziellen Abfolge, wobei die Funktion erst dann endet, wenn sämtliche Domains aus der Liste überprüft wurden.

Der Programmiercode der Funktion `location_ip()` wird in Anhang A.1 im Python-Skriptes `bt_fs_functions.py` wiedergegeben.

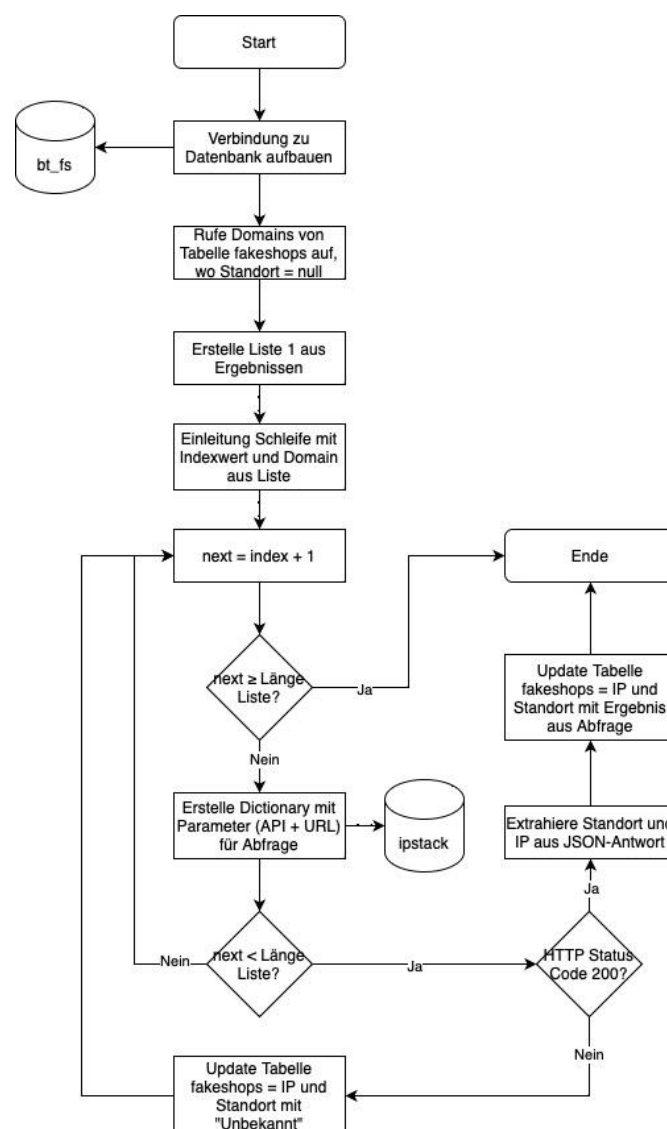


Abbildung 19: Ablauf Funktion `location_ip`

#### 4.2.6 Crontab

Um über die Portale [www.auktionshilfe.info](http://www.auktionshilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) möglichst aktuelle Fakeshops aufnehmen zu können, wird das Python-Skript `bt_fs_main.py` über einen Cronjob zeitbasiert ausgeführt. Zur Realisierung wird ein Shellskript erstellt, welches den Befehl zur Ausführung enthält.

```
cd /home/ubuntu/BT/  
python3 bt_fs_main.py
```

Abbildung 20: Inhalt Shellskript zur Ausführung Befehl

Mit dem Ubuntu Dienst Cron Daemon wird zu jeder vollen Stunde das Shellskript ausgeführt, welches anschließend das Python-Skript `bt_fs_main.py` ausführt.

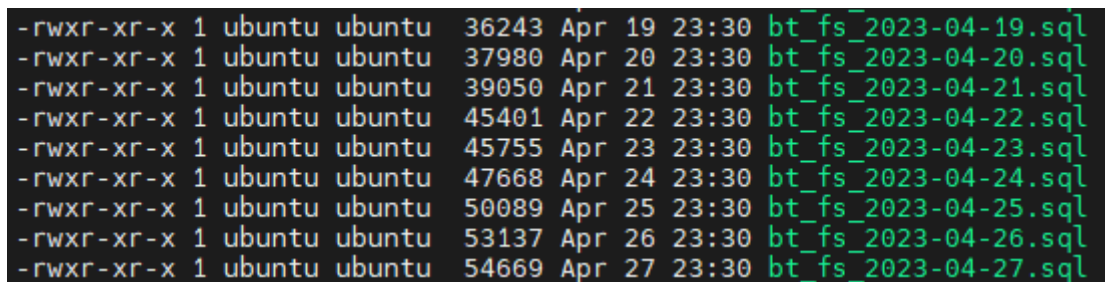
```
# For example, you can run a backup of all your user accounts  
# at 5 a.m every week with:  
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/  
#  
# For more information see the manual pages of crontab(5) and cron(8)  
#  
# m h dom mon dow   command  
  
0 * * * * /usr/local/bin/fakeshop_liste.sh  
30 23 * * * /usr/bin/python3 /home/ubuntu/BT/bt_daily_sql-dump.py
```

Abbildung 21 Crontab zur automatisierten Ausführung: `fakeshop_liste.sh`

So wird sichergestellt, dass möglichst alle Fakeshops aufgenommen werden. Der Dienst Cron Daemon ist in jeder Ubuntu Distribution nach der Installation als Standard enthalten. [29]

#### 4.2.7 Sicherung der Daten

Die Datenbank wird täglich einer Sicherung unterzogen, um einen möglichen Datenverlust vorzubeugen. Dazu wird wie in Abbildung 22 gezeigt, das Python-Skript `bt_daily_sql-dump.py` täglich um 23:30 Uhr ausgeführt. Durch das Python-Skript wird ein SQL-Dump erstellt, welches das jeweilige Tagesdatum (Abbildung 22) im Namen enthält. So kann die Datenbank auch zu einem früheren Zeitpunkt wiederhergestellt werden.



```
-rwxr-xr-x 1 ubuntu ubuntu 36243 Apr 19 23:30 bt_fs_2023-04-19.sql
-rwxr-xr-x 1 ubuntu ubuntu 37980 Apr 20 23:30 bt_fs_2023-04-20.sql
-rwxr-xr-x 1 ubuntu ubuntu 39050 Apr 21 23:30 bt_fs_2023-04-21.sql
-rwxr-xr-x 1 ubuntu ubuntu 45401 Apr 22 23:30 bt_fs_2023-04-22.sql
-rwxr-xr-x 1 ubuntu ubuntu 45755 Apr 23 23:30 bt_fs_2023-04-23.sql
-rwxr-xr-x 1 ubuntu ubuntu 47668 Apr 24 23:30 bt_fs_2023-04-24.sql
-rwxr-xr-x 1 ubuntu ubuntu 50089 Apr 25 23:30 bt_fs_2023-04-25.sql
-rwxr-xr-x 1 ubuntu ubuntu 53137 Apr 26 23:30 bt_fs_2023-04-26.sql
-rwxr-xr-x 1 ubuntu ubuntu 54669 Apr 27 23:30 bt_fs_2023-04-27.sql
```

Abbildung 22: SQL-Dumps der Datenbank `bt_fs`

Es folgt der Programmiercode für das Python-Skript `bt_daily_sql-dump.py`:

```
import subprocess
import datetime

def backup_database(user, database_name, dumpfile):
    command = f"sudo mysqldump -u {user} {database_name} > {dumpfile}"
    subprocess.run(command, shell=True, check=True)

if __name__ == "__main__":
    # DB Daten
    user = "root"
    database_name = "bt_fs"

    # Generiere Backup-File mit Name von Tagesdatum
    today = datetime.date.today().strftime("%Y-%m-%d")
    backup_directory = "/home/ubuntu/BT/Backups/SQL_Dumps/"
    dumpfile = f"{backup_directory}{database_name}_{today}.sql"

    # Backup der Datenbank
    backup_database(user, database_name, dumpfile)
```

#### **4.2.8 CSV zur Auswertung der Daten**

Für die spätere Auswertung sollen die Inhalte der Datenbank in eine Datei im CSV-Format exportiert werden. Hintergrund ist, dass mit dem Programm Excel von Microsoft Office die Aufbereitung erfolgen soll. Dafür wird ein Python-Skript erstellt, welche die Daten im geforderten Format bereitstellt.

Der Programmiercode wird in Anhang A.1 wiedergegeben.

## 5 Durchführung

In diesem Kapitel wird die konkrete Umsetzung zur Analyse von Fakeshops beschrieben. Es wird erläutert, wie und über welchen Zeitraum die Identifizierung der relevanten Informationen erfolgte, wie die Daten validiert wurden und welche Hindernisse bei der Analyse aufgetreten sind.

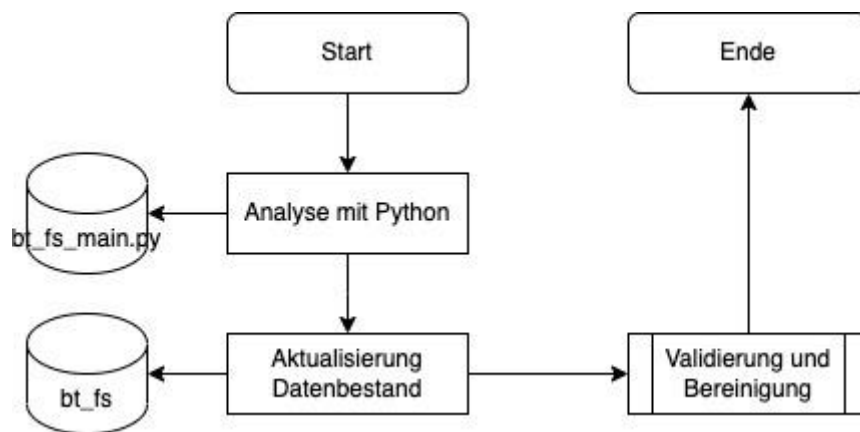


Abbildung 23: Ablauf der Durchführung

### 5.1 Zeitraum

Um eine möglichst große Datenbasis zu erhalten und damit die Forschungsfragen aus der Aufgabenstellung beantworten zu können, wurden Fakeshops über einen Zeitraum von 75 Tagen analysiert. Die Durchführung der Analyse wurde am 07.04.2023 gestartet und am 21.06.2023 beendet.

### 5.2 Analyse mit Python-Skripten

Die Durchführung wurde anhand der Methodik aus Kapitel 3 durchgeführt. Das Python-Skript `bt_fs_main.py` wurde zu jeder vollen Stunde am Tag ausgeführt. Sobald auf den Portalen [www.auktionshilfe.info](http://www.auktionshilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) neue Fakeshops gelistet wurden, wurden diese durch das Python-Skript in die

Datenbank bt\_fs überführt.

Anschließend wurden mit den Schnittstellen zu den Diensten aus Abschnitt 4.1.3 die Analysen durchgeführt und die Datenbank aktualisiert.

### **5.3 Limitationen**

Bei der Durchführung ist aufgefallen, dass ein verwendetes System durch den Dienst von [www.whatcms.org](http://www.whatcms.org) häufig nicht korrekt erkannt wird. Wird ein Fakeshop durch den Dienst von Cloudflare geschützt, so wird das System nicht immer erkannt, da der Dienst eine Identifizierung verhindern kann. Ebenfalls wird der Quellcode zum Teil anders dargestellt, was den Dienst von [www.whatcms.org](http://www.whatcms.org) vor große Herausforderungen stellt. So wird als System häufig Cloudflare, Apache, Nginx oder PHP aufgenommen. Auffällig ist auch, dass so bald ein Fakeshop das Content-Management-System WordPress verwendet, die Erweiterungen WooCommerce oder Elementor erkannt werden können. Wird das CMS WordPress verwendet, wird die Version in großen Teilen der Analyse nicht korrekt angegeben.

### **5.4 Validierung und Datenbereinigung**

Um die Zuverlässigkeit der aufgenommenen Daten zu gewährleisten, ist eine tägliche Validierung und Bereinigung der Daten unabdingbar. Dafür wird der Datenbestand über den gesamten Zeitraum täglich manuell überprüft. Der Fokus liegt dabei auf den Systemen, den mögliche Erweiterungen dieser sowie auf der nativen Sprache gelegt. Werden Systeme nicht korrekt erkannt, erfolgt eine händische Überprüfung, wobei der Quellcode des jeweiligen Fakeshops gesichtet und auf bekannte Shop- bzw. Content-Management-Systeme überprüft werden. Zusätzlich wird das Portal [www.builtwith.com](http://www.builtwith.com) hinzugezogen und der Fakeshop über den Dienst kontrolliert. Anschließend erfolgt eine Aktualisierung des Datenbestands. Das Verfahren wird ebenfalls bei Stichproben im Datenbestand angewendet. Dafür werden neue Fakeshops, welche korrekt

erkannt wurden ebenfalls manuell über die Sichtung des Quellcodes und den Dienst von [www.builtwith.com](http://www.builtwith.com) verifiziert.

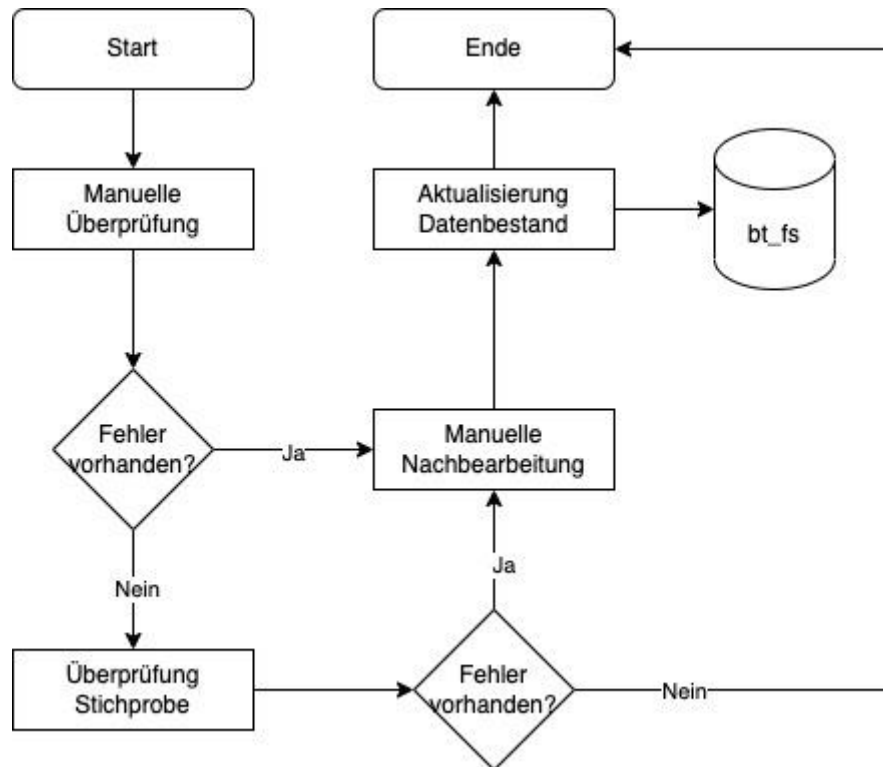


Abbildung 24: Teilprozess Validierung und Datenbereinigung

Anhand eines Beispiels wird aufgezeigt, wie die Datenbereinigung durchgeführt wird. In Abbildung 25 wurde der Fakeshop [lexer-rasenmaher.com](http://lexer-rasenmaher.com) in die Datenbank aufgenommen. Der Dienst von [whatcms.org](http://whatcms.org) erkennt das System Elementor, wobei es sich jedoch um eine Erweiterung für das Content-Management-System WordPress und kein eigenständiges System handelt. Außerdem wird die native Sprache English angegeben. Entsprechend muss eine manuelle Nachbearbeitung erfolgen.

lexer-rasenmaher.com	2023-06-20	Online	English	Elementor	3.13.4	NULL
----------------------	------------	--------	---------	-----------	--------	------

Abbildung 25: Fakeshop [lexer-rasenmaher.com](http://lexer-rasenmaher.com) - Falsches System

Dazu wird der Fakeshop manuell über einen Browser aufgerufen und der Quellcode untersucht. Wie in Abbildung 26 gezeigt, handelt es sich bei dem



System um das System WordPress mit der Shop-Erweiterung WooCommerce.

```

79 <link rel="stylesheet" id="elementor-icons-shared-0-css" href="https://www.lexer-rasenmaher.com/wp-content/plugins/elementor/assets/css/elementor-icons-shared-0.css" type="text/css">
80 <link rel="stylesheet" id="elementor-icons-fa-solid-css" href="https://www.lexer-rasenmaher.com/wp-content/plugins/elementor/assets/css/elementor-icons-fa-solid.css" type="text/css">
81 <link rel="stylesheet" id="elementor-icons-fa-regular-css" href="https://www.lexer-rasenmaher.com/wp-content/plugins/elementor/assets/css/elementor-icons-fa-regular.css" type="text/css">
82 <link rel="preconnect" href="https://fonts.gstatic.com/" crossorigin><script type="text/template" id="tmpl-variation-template">
83 <div class="woocommerce-variation-description">{{ data.variation.variation_description }}</div>
84 <div class="woocommerce-variation-price">{{ data.variation.price_html }}</div>
85 <div class="woocommerce-variation-availability">{{ data.variation.availability_html }}</div>
86 </script>
87 <script type="text/template" id="tmpl-unavailable-variation-template">
88 <p>Dieses Produkt ist leider nicht verfügbar. Bitte wähle eine andere Kombination.</p>
89 </script>
90 <script type="text/javascript" src="https://www.lexer-rasenmaher.com/wp-includes/js/jquery/jquery.min.js?ver=3.6.4" id="jquery"></script>
91 <script type="text/javascript" src="https://www.lexer-rasenmaher.com/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.4" id="jquery-migrate"></script>
92 <script type="text/javascript" src="https://www.lexer-rasenmaher.com/wp-content/plugins/razzi-addons/modules/buy-now/assets/js/buy-now.js" id="buy-now"></script>
93 <!--[if lt IE 9]>
94 <script type="text/javascript" src="https://www.lexer-rasenmaher.com/wp-content/themes/razzi/assets/js/plugins/html5shiv.js" id="html5shiv"></script>
95 <![endif]>
96 <!--[if lt IE 9]>
97 <script type="text/javascript" src="https://www.lexer-rasenmaher.com/wp-content/themes/razzi/assets/js/plugins/respond.min.js" id="respond"></script>
98 <![endif]>
99 <link rel="https://api.w.org/" href="https://www.lexer-rasenmaher.com/wp-json/" /><link rel="alternate" type="application/json+oembed" href="https://www.lexer-rasenmaher.com/wp-json/oembed/1.0/embed?url=https://www.lexer-rasenmaher.com/" />
100 <link rel="wlmmanifest" type="application/wlmmanifest+xml" href="https://www.lexer-rasenmaher.com/wp-includes/wlmmanifest.xml" />
101 <meta name="generator" content="WordPress 6.2.2" />
102 <meta name="generator" content="WooCommerce 7.8.0" />
103 <link rel="canonical" href="https://www.lexer-rasenmaher.com/" />
104 <link rel="shortlink" href="https://www.lexer-rasenmaher.com/" />
105 <link rel="alternate" type="application/json+oembed" href="https://www.lexer-rasenmaher.com/wp-json/oembed/1.0/embed?url=https://www.lexer-rasenmaher.com/" />
106 <link rel="alternate" type="text/xml+oembed" href="https://www.lexer-rasenmaher.com/wp-json/oembed/1.0/embed?url=https://www.lexer-rasenmaher.com/" />
107 </style>

```

Abbildung 26: WordPress und WooCommerce als System + Erweiterung

Weiter wird der Dienst builtwith.com hinzugezogen und die URL über den Dienst überprüft. Der Dienst gibt wie in Abbildung 27 zu erkennen ist an, dass es sich bei der nativen Sprache um Deutsch handelt und gibt auch noch einmal wieder, dass die Erweiterung WooCommerce durch den Fakeshop genutzt wird.

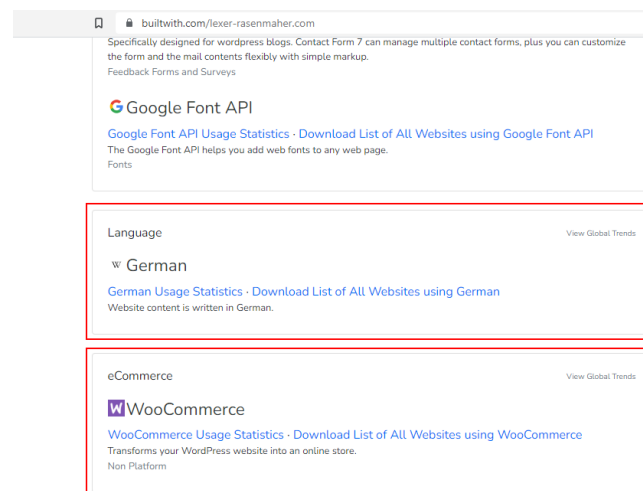


Abbildung 27: Überprüfung Fakeshop mit Dienst builtwith.com

Über SQL-Statements wird der Datenbestand aktualisiert. Dafür muss das

System, die Erweiterung, die Version und die Sprache in der Datenbank für die ID 1571 aktualisiert werden. Die SQL-Statements werden in Abbildung 28 wiedergegeben.

```
Update fakeshops set System = "WordPress" where id in (1571);  
Update fakeshops set Erweiterung = "WooCommerce" where id in (1571);  
Update fakeshops set Version = "6.2.2" where id in (1571);  
Update fakeshops set Sprache = "German" where id in (1571);
```

Abbildung 28: SQL-Statements zur Aktualisierung Datenbestand

## **6 Ergebnis und Trends**

### **6.1 Ergebnis**

Die Analyse der Fakeshops und deren verwendeten Systeme mit Hilfe von Python bringt Einblicke in die Vorgehensweisen der Betreibenden. Insgesamt wurden 1569 Fakeshops gefunden, wovon 1300 Fakeshops während der Analyse online waren. Die 1300 Fakeshops wurden untersucht und hinsichtlich ihrer Systeme analysiert. Bei der Analyse der Systeme konnten 17 verschiedene Shop bzw. Content-Management-Systeme gefunden werden. Zusätzlich zeigt sich, dass Fakeshops auch mit verschiedenen PHP-Frameworks erstellt werden, welche keinem bestimmten System zugeordnet werden können. PHP ist eine serverseitige Skriptsprache, welche häufig für die Erstellung von Internetseiten verwendet wird. Daher wird bei Systemen auf Basis von PHP-Frameworks „PHP“ als Ergebnis angegeben.

Die Mehrheit der Fakeshops basiert auf bestimmte E-Commerce-Plattformen, wobei 34,54 % der überprüften Fakeshops das System WordPress verwenden und 20,77 % das System Shopware.

In der Analyse der 1300 Fakeshops verwenden 651 (50,08%) Shops als native Sprache deutsch und 649 (49,92%) Fakeshops andere native Sprachen. Somit können 651 Fakeshops der deutschsprachigen Zielgruppe zugeordnet werden.

Für jeden Unterpunkt in diesem Kapitel erfolgt nachfolgend zunächst ein umfangreicher Überblick. Anschließend wird das spezifische Ergebnis für den deutschsprachigen Raum präsentiert, sofern dies für einen Vergleich notwendig ist. Darauf folgt eine Betrachtung auf internationaler Ebene, um die Zielgruppen aus der DACH-Region und dem internationalen Raum zu vergleichen.

### 6.1.1 Verwendete Systeme

In der Analyse konnte festgestellt werden, dass 450 der 1300 untersuchten Fakeshops als System WordPress einsetzen. Das entspricht einen Anteil von 34,62 %. Dahinter folgt das System Shopware mit 20,77 % (270 von 1300).

Bei 228 der untersuchten Fakeshops konnte kein direktes System gefunden werden, da zur Realisierung PHP-Frameworks eingesetzt wurden.

In Tabelle 1 werden die fünf häufigsten Systeme aufgezeigt und anschließend in Abbildung 29 die Gesamtverteilung der eingesetzten Systeme bei Fakeshops.

Platzierung	System	Anteil von 1300	Anteil in Prozent
1.	WordPress	450	34,62
2.	Shopware	270	20,77
3.	PHP	228	17,54
4.	Laravel	113	8,69
5.	Zen Cart	101	7,77

Tabelle 1: Verwendete Systeme – Gesamt

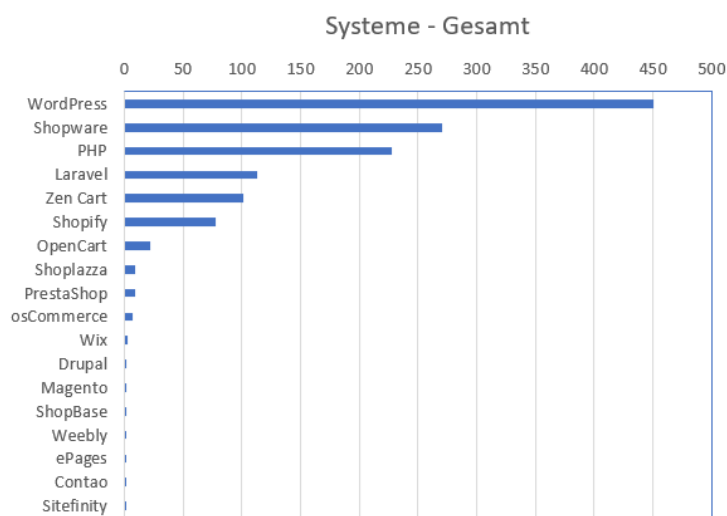


Abbildung 29: Verwendete Systeme - Gesamtverteilung

Im deutschsprachigen Raum werden ebenfalls die Systeme WordPress (39,02%) und Shopware (32,87%) am häufigsten verwendet, wobei der Anteil im deutschsprachigen Raum gegenüber dem Gesamtüberblick größer ist.

In Tabelle 2 werden die fünf häufigsten Systeme mit der Zielgruppe deutsch aufgezeigt und anschließend in Abbildung 30 die Verteilung der eingesetzten Systeme mit der Zielgruppe deutsch.

Platzierung	System	Anteil von 651	Anteil in Prozent
1.	WordPress	251	38,56
2.	Shopware	214	32,87
3.	PHP	76	11,67
4.	Shopify	38	5,84
5.	Zen Cart	33	5,07

Tabelle 2: Verwendete Systeme – Zielgruppe deutsch

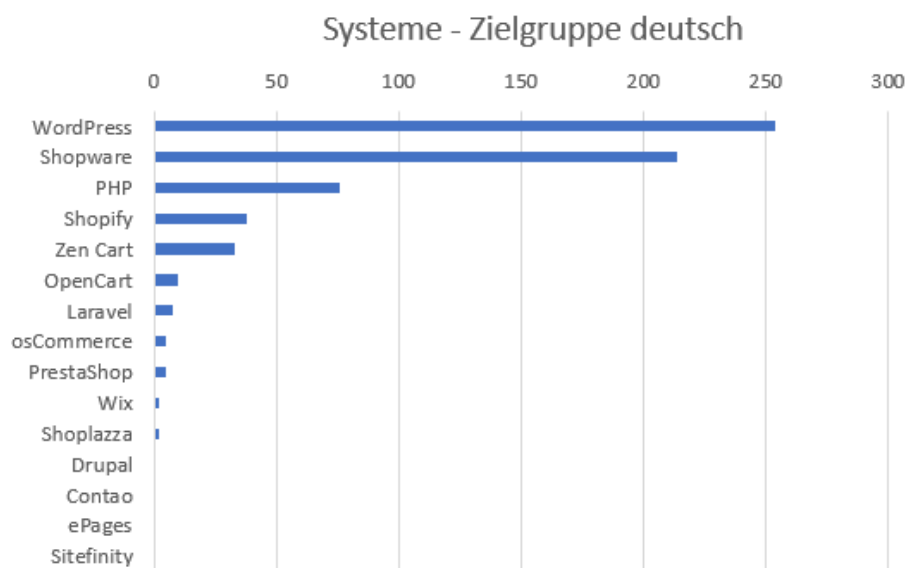


Abbildung 30: Verwendete Systeme - Zielgruppe deutsch

Auch im internationalen Raum liegt WordPress mit 30,05 % vorn. Dahinter folgen Fakeshops, welche mit PHP-Frameworks (23,57 %) erstellt wurden sowie Fakeshops mit dem System Laravel (16,18 %)

In Tabelle 3 werden die fünf häufigsten Systeme mit der Zielgruppe international aufgezeigt und anschließend in Abbildung 31 die Verteilung der eingesetzten Systeme mit der Zielgruppe international.

Platzierung	System	Anteil von 649	Anteil in Prozent
1.	WordPress	196	30,20
2.	PHP	152	23,42
3.	Laravel	76	11,71
4.	Zen Cart	68	10,48
5.	Shopware	56	8,63

Tabelle 3: Verwendete Systeme - Zielgruppe international

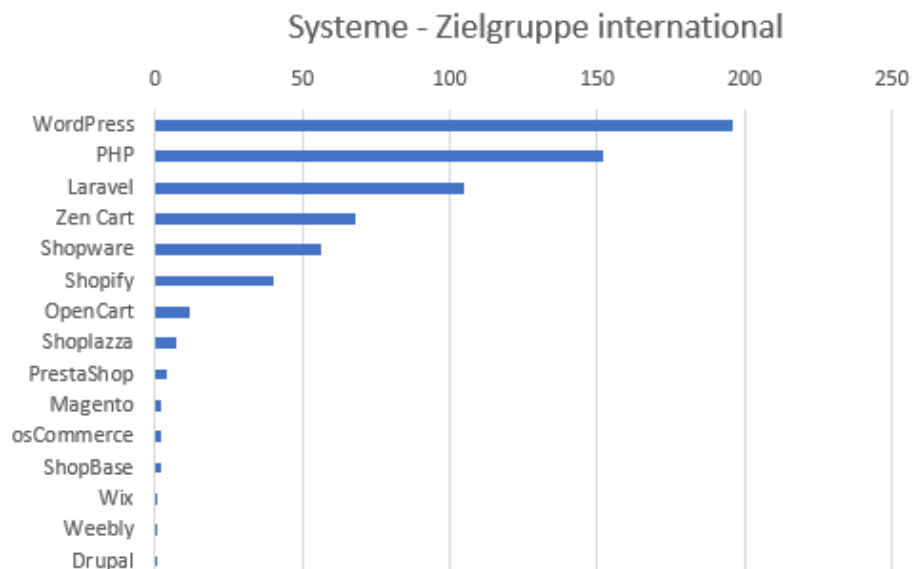


Abbildung 31: Verwendete Systeme - Zielgruppe international

Es ist deutlich zu erkennen, dass WordPress sowohl im Gesamten als auch im deutschsprachigen und internationalen Raum als System am häufigsten eingesetzt wird. PHP-Frameworks kommen im deutschsprachigen Raum weniger vor, während andere Länder diese vermehrt umsetzen, vermutlich da diese leicht dupliziert werden können. Das System Shopware wird im Vergleich von der Zielgruppe deutsch zu international vermehrt verwendet. Das System Laravel findet hingegen bei Fakeshop Betreibenden mit der Zielgruppe deutsch nur geringfügige Beachtung.

### 6.1.2 Systeme mit Erweiterung

Wie in Abschnitt 6.1 angegeben, konnte WordPress bei insgesamt 450 Fakeshops als verwendetes System identifiziert werden. 430 der Fakeshops mit dem System WordPress setzen zusätzlich die Erweiterung WooCommerce ein, um dem Content-Management-System die Shop-Funktionalität hinzuzufügen.

Damit nutzen 95,56 % der Fakeshops WordPress als System inklusive einer Erweiterung zur Realisierung eines Online-Shops .

Das System WordPress wird bei Fakeshops somit fast immer in Verbindung mit der Erweiterung WooCommerce eingesetzt.

Platzierung	Erweiterung	Anteil von 450	Anteil in Prozent
1.	WooCommerce	430	95,56

Tabelle 4: Erweiterung bei System WordPress

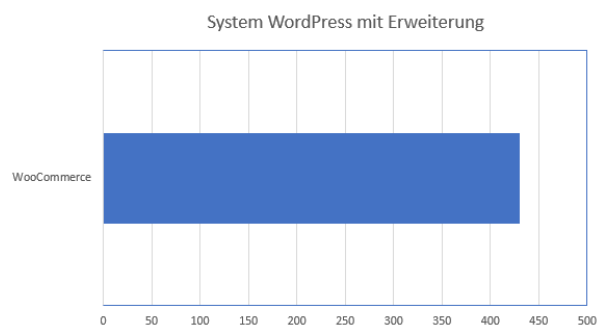


Abbildung 32: Erweiterung WooCommerce bei System WordPress

PHP-Frameworks verfügen in der Regel nicht über Shop-Funktionalitäten. In der Analyse konnten 228 Fakeshops mit dem System „PHP“ gefunden werden. 117 der überprüften Fakeshops setzen zusätzlich zum System „PHP“ die Erweiterung „Cart functionality“ ein, womit einer Internetseite die Funktionalität für Online-Shops hinzugefügt wird. Die ist ein Anteil von 51,32 % bei Fakeshops auf Basis von PHP-Frameworks.

Platzierung	Erweiterung	Anteil von 228	Anteil in Prozent
1.	Cart functionality	117	51,32

Tabelle 5: Erweiterung bei System PHP

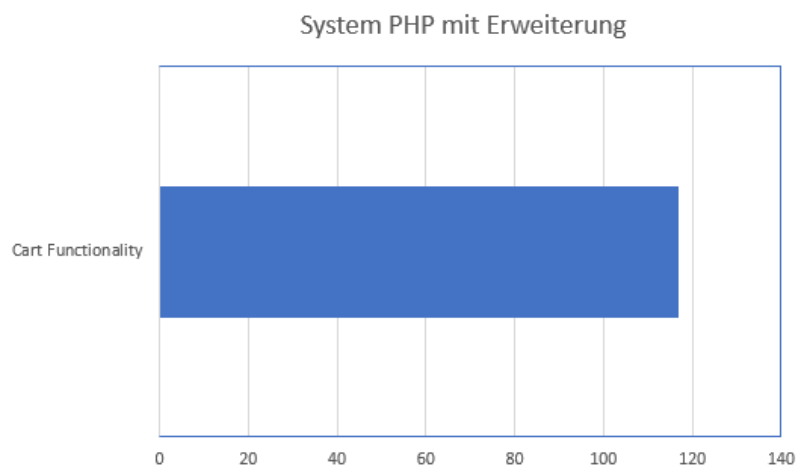


Abbildung 33: Erweiterung Cart functionality bei System PHP

### 6.1.3 Zuordnung Systeme und Erweiterungen in Online-Shop Kategorien

Wie in Kapitel 2 angegeben, können Online-Shop Systeme heute in fünf Kategorien unterteilt werden. Da es sich bei Fakeshops mit dem System „PHP“ sowohl um Eigenentwicklungen als auch um fertige Frameworks handeln kann, wird das System PHP der Kategorie Eigenentwicklung zugeordnet. In Tabelle 6 folgt eine Übersicht der bei den Fakeshops gefundenen Systeme mit der Zuordnung der Kategorien aus Kapitel 2. Zusätzlich werden die Erweiterungen WooCommerce und Cart functionality der Tabelle hinzugefügt.



System	Anteil von 1300	Anteil in Prozent	Kategorie
<b>WordPress</b>	450	34,62	Open-Source
<b>Shopware</b>	270	20,77	Self-Hosted
<b>PHP</b>	228	17,54	Eigenentwicklung
<b>Laravel</b>	113	8,69	Self-Hosted
<b>Zen Cart</b>	101	7,77	Open-Source
<b>Shopify</b>	78	6,00	Cloud-basiert
<b>OpenCart</b>	22	1,69	Open-Source
<b>Shoplazza</b>	9	0,69	Cloud-basiert
<b>PrestaShop</b>	9	0,69	Self-Hosted
<b>osCommerce</b>	7	0,54	Open-Source
<b>Wix</b>	3	0,23	Cloud-basiert
<b>Drupal</b>	2	0,15	Open-Source
<b>Magento</b>	2	0,15	Self-Hosted
<b>ShopBase</b>	2	0,15	Cloud-basiert
<b>Weebly</b>	1	0,08	Cloud-basiert
<b>ePages</b>	1	0,08	Cloud-basiert
<b>Contao</b>	1	0,08	Open-Source
<b>Sitefinity</b>	1	0,08	Self-Hosted
<b>Erweiterung</b>			
<b>WooCommerce</b>			Plugin-Software
<b>Cart funct.</b>			Plugin-Software

Tabelle 6: Zuordnung Systeme zu Online-Shop Kategorien

Zur Veranschaulichung werden die Anteile der gefundenen Systeme bei den 1300 Fakeshops den Kategorien zugeordnet und zusammengefasst. Dabei ist zu erkennen, dass Systeme auf Basis von Open-Source mit einem Anteil von 45 % am häufigsten verwendet werden. Es folgen Self-Hosted Systeme mit einem Anteil von 30 %. Cloud-basierte Systeme werden hingegen am wenigsten verwendet. Diese machen einen Anteil von 7 % aus. In Abbildung 34 wird die Verteilung der Kategorien wiedergegeben.

### Verteilung Systeme bei Fakeshops auf Online-Shop Kategorien

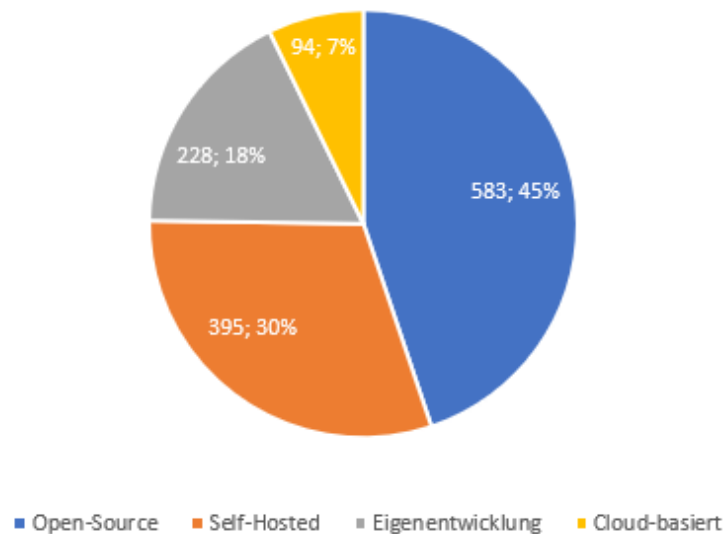


Abbildung 34: Verteilung der Systeme auf Online-Shop Kategorien

#### 6.1.4 Versionen von System WordPress und PHP

Im Zuge der Analyse wurden, soweit möglich, die Versionen der eingesetzten Systeme überprüft. Da die Systeme WordPress (Anteil: 34,54 %) und „PHP“ (Anteil: 17,54 %) häufig von Fakeshop Betreibenden eingesetzt werden, wird veranschaulicht, welche Versionen bei den genannten Systemen identifiziert werden konnten. Bei dem System WordPress konnten 19 verschiedene Versionen ausfindig gemacht werden. Bei 12 Fakeshops mit dem System WordPress konnte keine genaue Version identifiziert werden. Die Versionen 6.2.2 und 6.2 konnten regelmäßig gefunden werden.

In Abbildung 35 werden die Versionen und die jeweiligen Anteile bei dem System WordPress aufgezeigt.

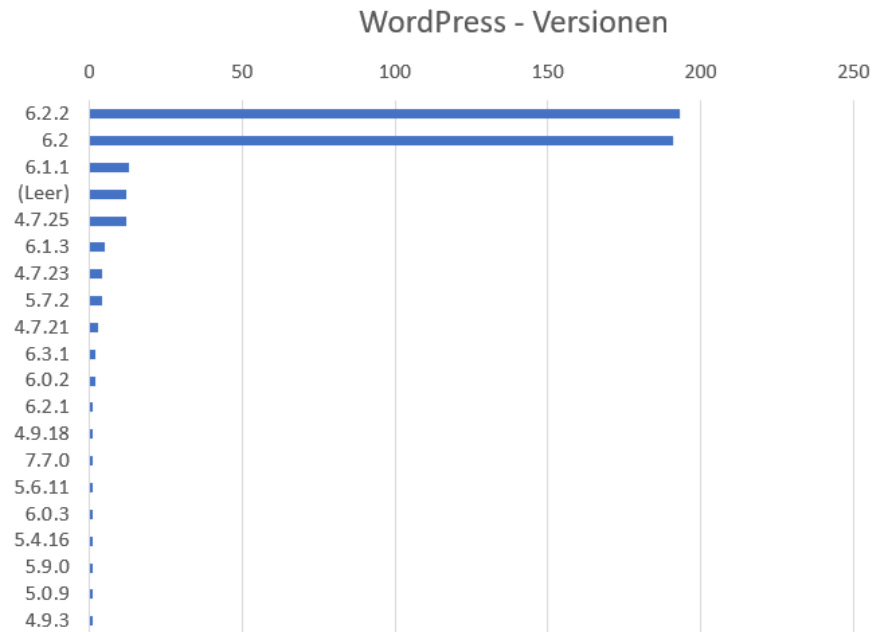


Abbildung 35: WordPress Versionen und Anteile

Bei Fakeshops mit dem System „PHP“ ist die Identifizierung der Version deutlich schwieriger. Entsprechend konnte bei 140 Fakeshops mit dem System „PHP“ keine Version ermittelt werden. Insgesamt wurden 15 verschiedene Versionen festgestellt. In Abbildung 36 werden die Versionen und Anteile wiedergegeben.

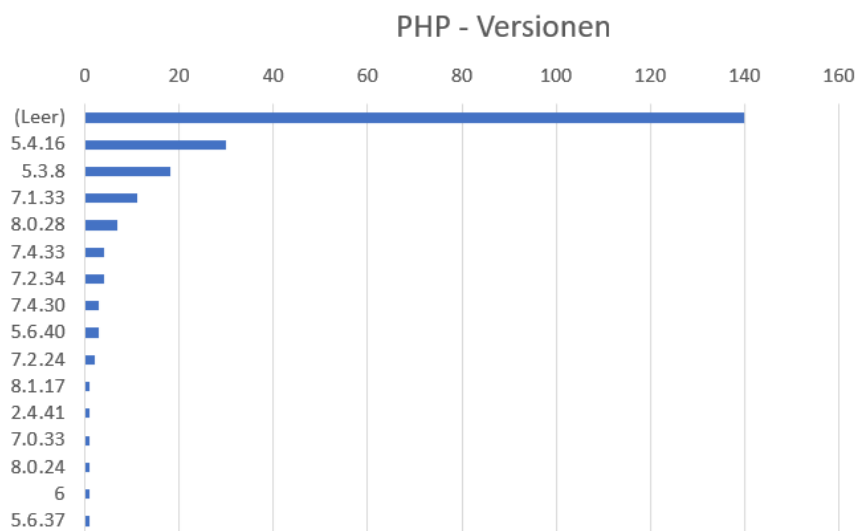


Abbildung 36: PHP-Versionen und Anteile

### 6.1.5 System Shopware in Version 5 und 6

Die Verwendung der Software Shopware als System konnte bei 270 Fakeshops festgestellt werden. Die Software wird in den Versionen 5 und 6 verwendet, welche sich vom inhaltlichen Aufbau wesentlich unterscheiden. Entsprechend wird in diesem Abschnitt aufgezeigt, wie häufig die Software als System eingesetzt wird und wie die Aufteilung der Versionen sich dabei zusammensetzen.

Von den 270 Fakeshops, welche das System Shopware einsetzen, nutzen 202 Fakeshops die Version 5 und 68 Fakeshops die Version 6.

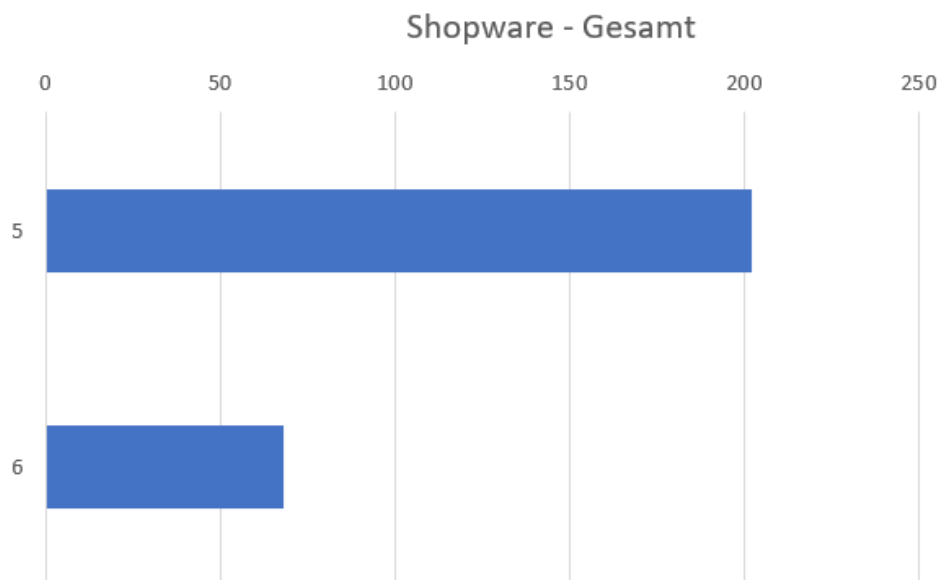


Abbildung 37: Shopware Verteilung der Version 5 und 6

Mit 214 der 270 Fakeshops, wird Shopware als System überwiegend mit der nativen Zielsprache deutsch eingesetzt. Dabei wird die Version 5 deutlich häufiger eingesetzt, wie in Abbildung 38 zu verdeutlicht wird.

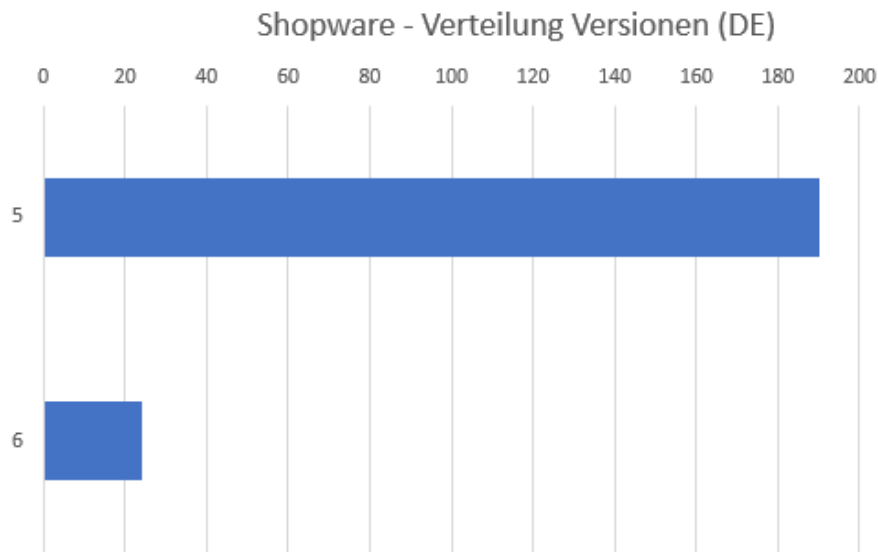


Abbildung 38: Shopware Versionen - Verteilung bei Zielsprache deutsch

Im internationalen Raum war das System Shopware deutlich seltener vorzufinden. Nur bei 56 der 649 Fakeshops im anders sprachigen Raum wird Shopware eingesetzt. Die Version 5 wurde nur 12-mal gefunden und 44-mal wurde die Version 6 gefunden.

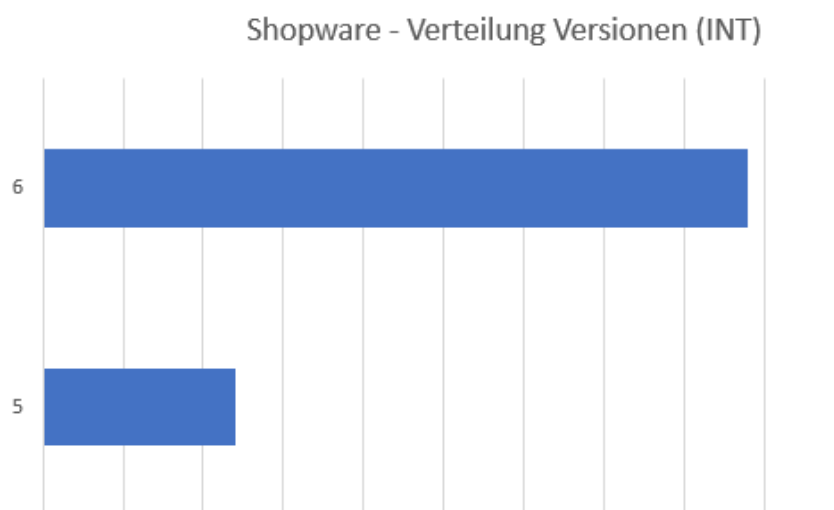


Abbildung 39: Shopware Versionen - Bei Zielsprache nicht deutsch

Wie in den Ergebnissen zu erkennen ist, wird Shopware bei Fakeshops mit der Zielsprache deutsch deutlich häufiger eingesetzt als im internationalen Vergleich. Dabei wird insgesamt die Version 5 am häufigsten genutzt.

### 6.1.6 Einsatz von Cloudflare

Die Dienst Cloudflare zur Verschleierung der Identität wurde von 939 der 1300 überprüften Fakeshops eingesetzt. Das macht einen Anteil von 72,23 % aus.

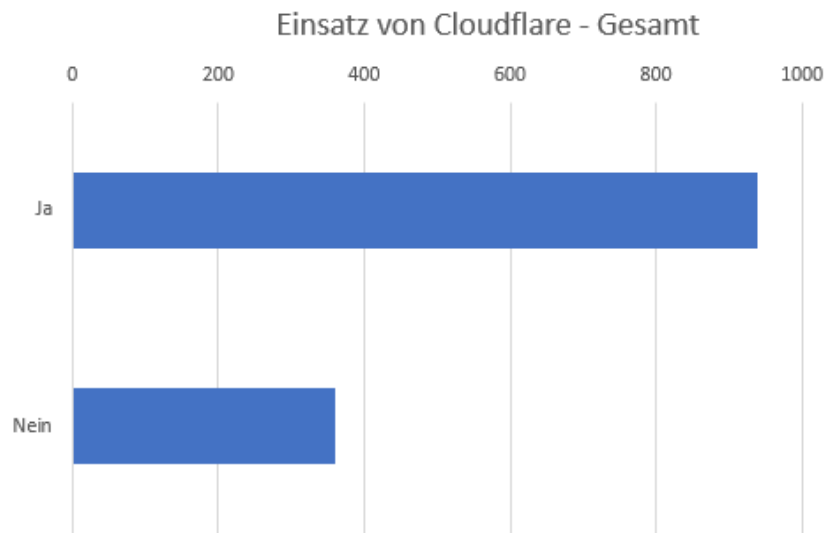


Abbildung 40: Einsatz von Cloudflare bei Fakeshops

Wie in Abbildung 41 zu erkennen ist, setzen 478 der 651 Fakeshops mit der Zielsprache deutsch den Dienst von Cloudflare ein. Das ist ein Anteil von 73,43 %.

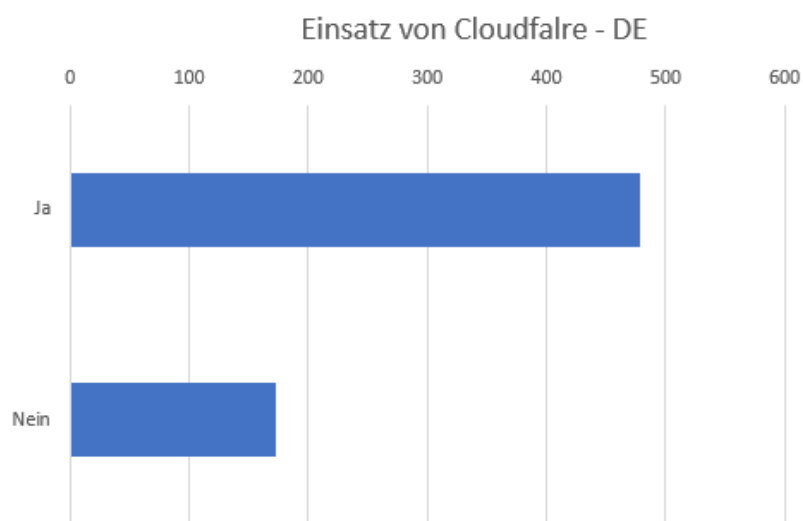


Abbildung 41: Einsatz von Cloudflare bei Fakeshops – DE

Im internationalen Raum wird Cloudflare ebenfalls häufig zur Verschleierung der Identitäten genutzt. 461 der 649 Fakeshops (Anteil von 71,03 %), mit den Zielsprachen englisch und weitere setzen den Dienst ein.

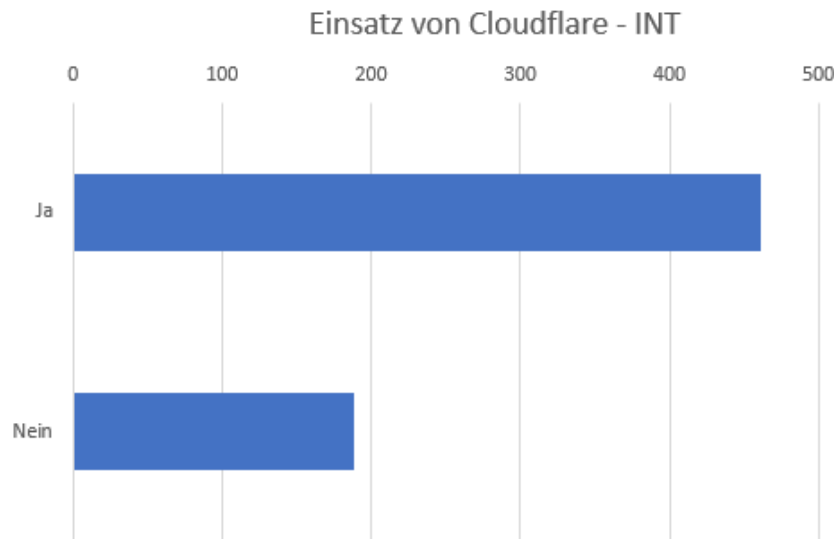


Abbildung 42: Einsatz von Cloudflare bei Fakeshops – INT

Sowohl bei deutschsprachigen Fakeshops als auch im internationalen Vergleich wird der Dienst von Cloudflare eingesetzt. Damit setzen fast 3 von 4 Fakeshops Cloudflare ein.

### 6.1.7 Systeme die Cloudflare nutzen

Die Systeme WordPress, Shopware, Laravel, Shopify, Zen Cart und PHP-Frameworks setzen auf den Dienst von Cloudflare. Mit einem Gesamtanteil von 97,98 % wurde Cloudflare auf Fakeshops gefunden, welche die o.g. Systeme einsetzen. Es folgt eine Auflistung der Ergebnisse mit dem Gesamtanteil des Systems, dem Anteil wie häufig Cloudflare bei dem System eingesetzt wird und welcher Anteil dies ausmacht. Die Auflistung wird in Tabelle 7 wiedergeben.

Das Ergebnis der Tabelle 7 zeigt, dass die Systeme Shopware, Laravel und Shopify, welche mindestens 50-mal als System identifiziert werden konnten, mit einem Anteil von über 95% auch Cloudflare zur Verschleierung der Identitäten einsetzen.

Platzierung	System	Gesamt	Anteil Cloudflare	Anteil in
1.	WordPress	450	309	68,67
2.	Shopware	270	259	95,93
3.	PHP	228	126	55,26
4.	Laravel	113	109	96,46
5.	Shopify	78	76	97,44
6.	Zen Cart	101	41	40,59
7.	OpenCart	22	9	40,91
8.	Shoplazza	9	5	55,56
9.	Drupal	2	2	100
10.	ShopBase	2	2	100
11.	Contao	1	1	100

Tabelle 7: Ergebnis Cloudflare mit Anteil bei Systemen

### 6.1.8 Server-Standorte

Insgesamt konnten 34 verschiedene Server-Standorte lokalisiert werden. Mit großem Abstand wurden die meisten Fakeshops in den USA gehostet. 943 der 1300 Fakeshops wurden in den USA gehostet. Das entspricht einen Gesamtanteil von 72,54%. Darauf folgen die Standorte Kanada mit 6% und die Niederlande mit 5,77%.

Zusätzlich konnten 37 der überprüften 1300 Fakeshops gefunden werden, welche in Deutschland gehostet werden. Mit 2,85% jedoch ein sehr geringer Anteil. In Tabelle 8 werden die fünf häufigsten Standorte aufgezeigt und in Abbildung 43 die gesamten Standorte mit ihrer Verteilung.



Platzierung	Standort	Anteil von 1300	Anteil in Prozent
1.	USA	943	72,54
2.	Kanada	78	6,00
3.	Niederlande	75	5,77
4.	Deutschland	37	2,85
5.	Ukraine	26	2,00

Tabelle 8: Fünf häufigste Server-Standorte

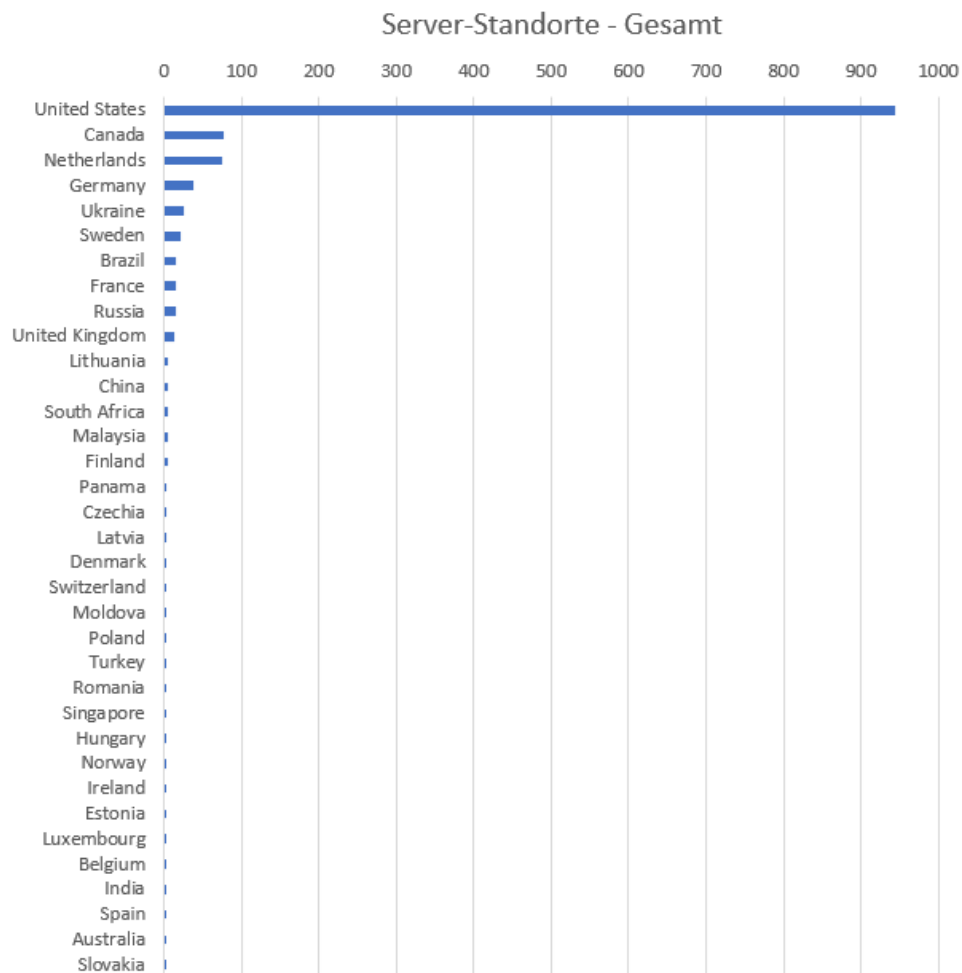


Abbildung 43: Server-Standorte - Gesamt

Fakeshops, welche als Zielgruppe die DACH-Region verfolgen konnten an 29 verschiedenen Standorten lokalisiert werden. Auch hier ist die USA mit 484 führend, gefolgt von den Niederlanden (40), Kanada (38), Deutschland (29) und Brasilien (8).

Platzierung	Standort	Anteil von 651	Anteil in Prozent
1.	USA	484	74,35
2.	Niederlande	40	6,14
3.	Kanada	38	5,84
4.	Deutschland	29	4,45
5.	Brasilien	8	1,23

Tabelle 9: Fünf häufigste Server-Standorte – DE

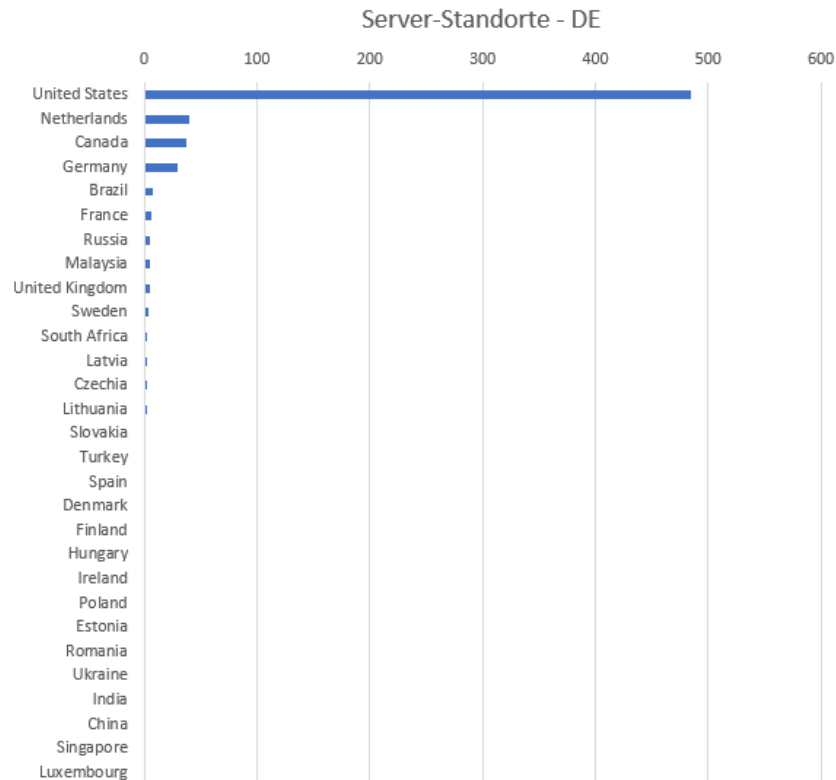


Abbildung 44: Server-Standorte – DE

Anderssprachige Fakeshops werden ebenfalls am häufigsten in den USA gehostet. 459 der 649 Fakeshops konnten dort lokalisiert werden, was einen Anteil von 70,72% entspricht. Die Auswertung wird in Tabelle 10 und Abbildung 45 wiedergegeben.

Platzierung	Standort	Anteil von 649	Anteil in Prozent
1.	USA	459	70,72
2.	Kanada	40	6,16
3.	Niederlande	35	5,39
4.	Ukraine	25	3,85
5.	Schweden	25	3,85

Tabelle 10: Fünf häufigste Server-Standorte – INT

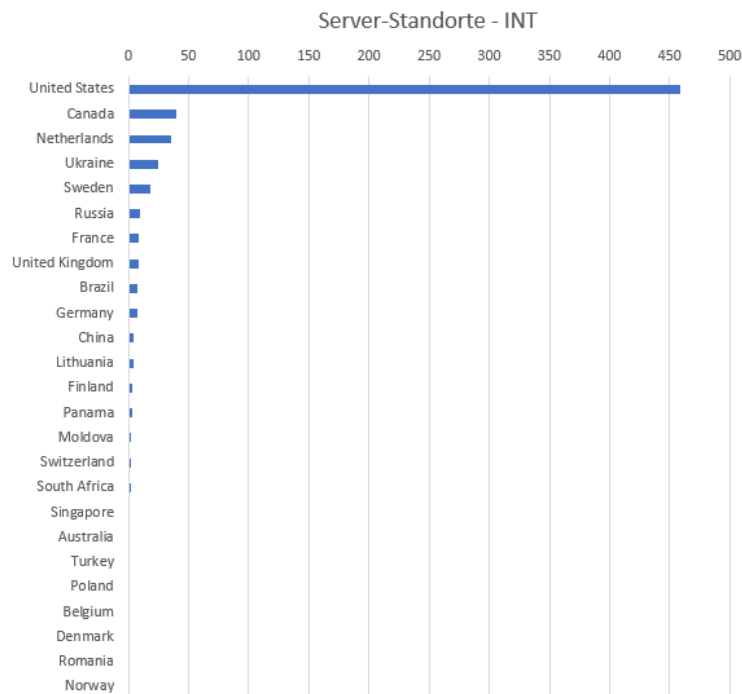


Abbildung 45: Server-Standorte – INT

Anhand der Auswertung wird deutlich, dass der Server-Standort USA sowohl im internationalen als auch im Vergleich für Fakeshops die auf die Zielgruppe mit deutscher Sprache ausgerichtet sind, der am häufigsten gewählte Standort ist. Die Ukraine als Server-Standort konnte bei Fakeshops mit verwendeter native Sprache deutsch lediglich einmal lokalisiert werden. Die Niederlande und Kanada werden mit den zweit- und dritthöchsten Anteilen hingegen ebenfalls häufig gewählt.

#### 6.1.9 IP-Adressen

Bei der Untersuchung der Fakeshops wurden auch gleiche IP-Adressen gefunden. Insgesamt wurden 1117 verschiedene IP-Adressen erfasst.

53 IP-Adressen konnten mehr als einmal gefunden werden.

Die IP-Adresse 23.227.38.65 konnte 42 Fakeshops zugeordnet werden, was einen Anteil von 3,76% entspricht.

Platzierung	IP-Adresse	Anteil von 1300	Anteil in Prozent
1.	23.227.38.65	42	3,76
2.	185.254.198.235	23	2,06
3.	104.16.198.133	23	2,06
4.	185.225.115.62	15	1,34
5.	23.227.38.32	10	0,90

Tabelle 11: Fünf häufigsten IP-Adressen

Eine Untersuchung der IP-Adressen aus Tabelle 11 ergibt, dass die IP-Adressen über den Dienst Whois ([www.whois.com](http://www.whois.com)) lokalisiert und teilweise zugeordnet werden können.

### IP-Adresse 23.227.38.65

Der IP-Bereich 23.227.32.0 bis 23.227.63.225 kann dem Unternehmen Shopify (Herkunft Kanada) zugeordnet werden, wie in Abbildung 46 verdeutlicht wird. Die IP-Adresse 23.227.38.65 wurde durchweg bei Fakeshops mit dem System Shopify gefunden. Werden der Bereich und die gefundenen Fakeshops betrachtet, so können 73 Fakeshops dem genannten IP-Bereich zugeordnet werden. Die restlichen 5 Fakeshops, welche Shopify einsetzen, werden in den USA gehostet und haben eine andere IP-Adresse.

### Whois IP 23.227.38.65

Updated 5 days ago

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
#

NetRange:      23.227.32.0 - 23.227.63.255
CIDR:          23.227.32.0/19
NetName:       SHOPIFY-NET
NetHandle:     NET-23-227-32-0-1
Parent:        NET23 (NET-23-0-0-0-0)
NetType:       Direct Allocation
OriginAS:      AS62679
Organization:  Shopify, Inc. (SHOPI-1)
RegDate:       2013-09-19
Updated:       2021-12-14
Ref:           https://rdap.arin.net/registry/ip/23.227.32.0
```

Abbildung 46: Whois IP-Adresse 23.227.38.65

### IP-Adresse 185.254.198.235

Die IP-Adresse 185.254.198.235 kann dem Unternehmen Virtual System LLC (Herkunft Ukraine) zugeordnet werden. Das Unternehmen besitzt den IP-Bereich 185.254.196.0 bis 185.254.199.255. Das Unternehmen ist ein Webhosting-Unternehmen mit Sitz in Kiew. Bei 23 der 26 identifizierten Fakeshops mit dem Server-Standort in der Ukraine kann die IP-Adresse 185.254.198.235 diesen Fakeshops zugeordnet werden. In Abbildung 47 wird die Auflösung der IP-

Adresse über den Dienstleister Whois.com angezeigt. Alle Fakeshops, welche die genannte IP-Adresse verwenden, basieren auf dem System Zen Cart und verwenden dabei nicht den Dienst von Cloudflare.

## Whois IP 185.254.198.235

Updated 13 minutes ago

```
% This is the RIPE Database query service.
% The objects are in RPSL format.
%
% The RIPE Database is subject to Terms and Conditions.
% See http://www.ripe.net/db/support/db-terms-conditions.pdf

% Note: this output has been filtered.
%       To receive output for a database update, use the "-B" flag.

% Information related to '185.254.196.0 - 185.254.199.255'

% Abuse contact for '185.254.196.0 - 185.254.199.255' is 'abuse@v-sys.org'

inetnum:        185.254.196.0 - 185.254.199.255
netname:        UA-VSYS-20180411
country:        UA
org:            ORG-VSL22-RIPE
admin-c:        VS10657-RIPE
tech-c:         VS10657-RIPE
status:         ALLOCATED PA
mnt-by:         VSYS-MNT
mnt-by:         RIPE-NCC-HM-MNT
created:        2021-05-25T06:57:27Z
last-modified:  2021-05-25T06:57:27Z
source:         RIPE

organisation:   ORG-VSL22-RIPE
org-name:       Virtual Systems LLC
country:        UA
org-type:       LIR
address:        Laboratorna str., building 33/37
address:        03150
address:        Kiyv
address:        UKRAINE
```

Abbildung 47: Whois IP-Adresse 185.254.198.235

## IP-Adresse 104.16.198.133

Der IP-Bereich 104.16.0.0 bis 104.31.255.255 kann dem Unternehmen Cloudflare aus den USA zugeordnet werden. Die IP-Adresse 104.16.198.133 wurde bei den Systemen Zen Cart und PHP gefunden, wobei PHP den größten Anteil ausmacht. Alle Fakeshops, welche der IP-Adresse zugeordnet werden konnten, nutzen entsprechend auch der Dienst von Cloudflare.

In Abbildung 48 werden die Informationen zur IP-Adresse wiedergegeben.

## Whois IP 104.16.198.133

Updated 30 minutes ago

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
#

NetRange:      104.16.0.0 - 104.31.255.255
CIDR:          104.16.0.0/12
NetName:       CLOUDFLARENET
NetHandle:     NET-104-16-0-1
Parent:        NET104 (NET-104-0-0-0-0)
NetType:       Direct Allocation
OriginAS:      AS13335
Organization:  Cloudflare, Inc. (CLOUD14)
RegDate:       2014-03-28
Updated:       2021-05-26
Comment:       All Cloudflare abuse reporting can be done via https://www.cl
Ref:           https://rdap.arin.net/registry/ip/104.16.0.0

OrgName:       Cloudflare, Inc.
OrgId:         CLOUD14
Address:       101 Townsend Street
City:          San Francisco
StateProv:     CA
PostalCode:    94107
Country:       US
```

Abbildung 48: Whois IP-Adresse 104.16.198.133

## IP-Adresse 185.225.115.62

Die IP-Adresse 185.225.115.62 kann keinem direkten Unternehmen zugeordnet werden. Laut Informationen über Whois.com wird die Adresse im Bereich 185.225.114.0 bis 185.225.115.255 verwendet, welcher auf die Seychellen verweist. Eine weitere Untersuchung wird im Rahmen dieser Thesis nicht durchgeführt.

```
inetnum:      185.225.114.0 - 185.225.115.255
netname:      SC-IPFIB-SE
country:      SE
admin-c:      IA6784-RIPE
tech-c:       IA6784-RIPE
status:       ASSIGNED PA
mnt-by:       IPFIB-MNT
created:      2022-10-05T15:18:43Z
last-modified: 2023-01-31T11:29:52Z
source:       RIPE

role:         IPFIB-RIPE
address:      SEYCHELLES
```

Abbildung 49: Whois IP-Adresse 185.225.115.62

### 6.1.10 IP-Adresse 23.227.38.32

Die IP-Adresse 23.227.38.32 liegt ebenfalls im Bereich 23.227.32.0 – 23.227.63.225, wo bereits die IP-Adresse 23.227.38.65 zugeordnet werden konnte. Dieser Bereich ist dem Unternehmen Shopify aus Kanada zugeordnet. Entsprechend wird hier nicht weiter auf die IP-Adresse eingegangen.

#### Whois IP 23.227.38.32

Updated 4 days ago

```
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2023, American Registry for Internet Numbers, Ltd.
#

NetRange:      23.227.32.0 - 23.227.63.255
CIDR:          23.227.32.0/19
NetName:       SHOPIFY-NET
NetHandle:     NET-23-227-32-0-1
Parent:        NET23 (NET-23-0-0-0-0)
NetType:       Direct Allocation
OriginAS:      AS62679
Organization:  Shopify, Inc. (SHOPI-1)
RegDate:       2013-09-19
Updated:       2021-12-14
Ref:           https://rdap.arin.net/registry/ip/23.227.32.0

OrgName:       Shopify, Inc.
OrgId:         SHOPI-1
Address:       151 O'Connor Street, Ground floor
City:          Ottawa
StateProv:     ON
PostalCode:    K2P 2L8
Country:       CA
```

Abbildung 50: Whois IP-Adresse 23.227.38.32

## 6.2 Trends

Die Ergebnisse verdeutlichen, dass das System WordPress unter Fakeshops am weitesten verbreitet ist. Mit einem Anteil von fast 35 % ist jeder dritte untersuchte Fakeshops auf Basis von WordPress. Da es sich bei WordPress um ein Content-Management-System ohne Shop-Funktionalität handelt, kommt bei über 95 %



dieser Fakeshops die Plugin-Software WooCommerce zum Einsatz. Neben WordPress werden auch bekannte Systeme wie Shopware, Shopify, Zen Cart und Laravel eingesetzt. Eine Schwierigkeit besteht darin, dass PHP-Frameworks eingesetzt werden, welche keinem standardisierten System zugeordnet werden können. Diese Fakeshops verwenden zwar häufig die Erweiterung Cart functionality, jedoch handelt es sich bei diesen Shops zum Zeitpunkt dieser Ausarbeitung um eine weitestgehend unbekannte Komponente, wenn es um typische Merkmale dieser Fakeshops geht.

Die gefundenen Systeme können Online-Shop Kategorien aus Kapitel 2 zugeordnet werden. Dabei wird deutlich, dass Open-Source- und Self-Hosted-Systeme mit einem Gesamtanteil von 75 % die große Mehrheit ausmachen. Eigenentwicklungen und Cloud-basierte-Systeme sind mit 25% deutlich weniger verbreitet. Es kann davon ausgegangen werden, dass auch in Zukunft die Anzahl der Fakeshops auf Basis von Open-Source und Self-Hosted-Systemen zunehmen wird, da Anbieter wie Shopify und Co., welche den Cloud-basierten-Systemen zugeordnet werden, die Verbreitung der Fakeshops und damit verbundenen Straftaten in ihrer eigenen Cloud erschweren bzw. unterbinden werden.

Hingegen entstehen fast täglich neue Anleitungen und Tutorials, wie eigene Online-Shops auf Basis von Open-Source- und Self-Hosted-Systemen aufgebaut und optimiert werden können. Die Einfachheit diese Online-Shops zu erstellen und im Ausland zu hosten, wo deutsche Strafverfolgungsbehörden nicht handlungsfähig sind, wird auch in Zukunft die Verbreitung von Fakeshops nicht verhindern.

Mit 651 von 1300 Fakeshops wurden in der Analyse Fakeshops untersucht, welche als native Sprache deutsch einsetzen und damit die Zielgruppe deutschsprachiger (DACH-Region) Verbraucher fokussieren. Insgesamt wurden 15 verschiedene Sprachen identifiziert, wovon die Sprache deutsch mit 651 und die Sprache Englisch mit 605 Fakeshops den größten Anteil (96,54 %) ausmachen. Es wurden auch Fakeshops mit den nativen Sprachen russisch, französisch, spanisch und weitere gefunden, welche jedoch mit 3,46 % einen sehr geringen Anteil ausmachen.

Eine weitere wichtige Erkenntnis ist, dass der Dienst Cloudflare bei über 72 % der untersuchten Fakeshops zum Einsatz kommt. Dies gestaltet zusätzlich die Analyse von Fakeshops umständlicher, da der Quellcode zum Teil nicht automatisiert untersucht werden kann. Außerdem können Betreibende damit die eigene Identität verschleiern und die Verfolgung durch deutsche Strafverfolgungsbehörden erschweren.

Ein weiterer Aspekt ist, dass die untersuchten Fakeshops mit einem großen Anteil von 72,54 % in den USA gehostet werden. In Deutschland wurden lediglich 37 der analysierten Fakeshops lokalisiert. Entsprechend ist der Gesamtanteil der Fakeshops die im Ausland gehostet werden bei 97,15 %. Damit verlagern sich die Straftaten immer weiter ins Ausland, was auch hier deutsche Strafverfolgungsbehörden vor eine große Herausforderung stellt.

Im Gesamten festzuhalten ist, dass die Anzahl der untersuchten Fakeshops mit einer Anzahl von 1569 in nur 75 Tagen aufzeigt, dass im Durchschnitt rund 21 Fakeshops täglich hinzukommen. Zumindest dann, wenn davon ausgegangen wird, dass alle Fakeshops auf den Portalen [www.auktionshilfe.info](http://www.auktionshilfe.info) und [www.watchlist-internet.at](http://www.watchlist-internet.at) gelistet werden. Da diese Portale sich aber überwiegend auf die DACH-Region konzentrieren und die Dunkelziffer mit hoher Wahrscheinlichkeit sehr hoch sein wird, ist die Anzahl von 21 als Mindestanzahl zu verstehen.

## 7 Fazit und Ausblick

Zusammenfassend ist festzuhalten, dass die Analyse von Fakeshops und ihren verwendeten Systemen mithilfe von Python ein vielversprechender Ansatz ist, um das Ausmaß und die Methoden von Fakeshops zu verstehen. Durch die Erstellung und Anwendung verschiedener Python-basierter Skripte konnten relevante Daten von Fakeshops extrahiert, analysiert und Muster identifiziert werden, die zur Erkennung potenzieller Fakeshops beitragen und Trends bei Fakeshops aufzeigen. Die Erkenntnisse dieser Arbeit haben sowohl theoretische als auch praktische Implikationen.

Auf der theoretischen Seite tragen die Ergebnisse dazu bei, dass Verständnis der Funktionsweise von Fakeshops und der verwendeten Taktiken, die von Betreibenden genutzt werden, bei. Zum Beispiel verwenden Fakeshops überwiegend bekannte Shop- und Content-Management-Systeme wie WordPress oder Shopware. Die überprüften Fakeshops sind mit einem Anteil von über 75 % der Kategorie Open-Source und Self-Hosted-Systemen zuzuordnen. Anhand der nativen Sprache wurde festgestellt, dass 651 von 1300 Fakeshops die Zielgruppe deutschsprachig fokussieren. Fakeshops weisen häufig ähnliche Strukturen auf, was sich auch darin widerspiegelt, dass mit einem Anteil von über 97,15 % Fakeshops im Ausland gehostet werden, wobei die USA mit einem Anteil von 72,54 % den größten Anteil ausmacht.

Diese Erkenntnisse ermöglichen es, präventive Maßnahmen zu entwickeln, um Verbraucher besser vor Online-Betrug zu schützen. Mit gezielten Aufklärungskampagnen und verbesserter Sensibilisierung können Verbraucher besser über Fakeshops informiert werden.

Auf der praktischen Seite könnten die aufgezeigten Methoden und entwickelten Algorithmen von Strafverfolgungsbehörden, Verbraucherschutzorganisationen oder E-Commerce-Plattformen genutzt werden, um Fakeshops weiter zu analysieren und zukünftige Aktivitäten einzudämmen. Identifizierte Fakeshops könnten automatisiert bei Diensten wie Cloudflare gemeldet werden, um die Zeit, wie lange ein Fakeshop online ist, möglichst gering zu halten. Anhand der IP-

Adressen könnten Hosting-Provider ausfindig gemacht werden und durch Kooperationen die Schließung von Fakeshops beschleunigt werden. Dies kann die Sichtbarkeit des Fakeshops einschränken und mögliche Opfer vor finanziellen Verlust schützen.

Eine zusätzliche Möglichkeit besteht darin, Verbraucher noch mehr über die Existenz von Fakeshops und deren Erkennungsmerkmale aufzuklären. Verbraucherorganisationen, Strafverfolgungsbehörden und das BSI versuchen mit präventiven Maßnahmen vor Online-Betrug zu schützen. Diese Maßnahmen sollten angepasst und überarbeitet werden.

Die Arbeit soll eine solide Grundlage für weitere Forschungsarbeiten im Bereich der Fakeshop-Analyse bieten. Zukünftige Ausarbeitungen könnten sich beispielsweise darauf konzentrieren, die Effektivität der entwickelten Algorithmen weiter zu verbessern und die Teilautomatisierung auszubauen, damit händische Verbesserungen am Datenbestand künftig kaum bis gar nicht notwendig wären. Mit künftigen Analysen und anhand der erarbeiteten Datenbasis könnten fortgeschrittene Machine-Learning-Techniken wie Deep Learning eingesetzt werden, um noch präzisere Modelle zur Fakeshop-Erkennung zu entwickeln. Zudem könnte der Einsatz von künstlicher Intelligenz und automatisierter Systeme eine kontinuierliche Überwachung des Online-Raums nach Aktivitäten von Fakeshops ermöglichen. Ein Ansatz hierzu wäre, Google-Shopping Ergebnisse automatisiert zu crawlen, um auch hier neue Fakeshops frühzeitig erkennen zu können. Des Weiteren besteht die Chance darauf, dass bereits identifizierte Fakeshops auf die IP-Adresse untersucht und anhand der Ergebnisse über verschiedene Dienste weitere Online-Shops mit der gleichen IP-Adresse auf verdächtige Aktivitäten überprüft werden.

Diese kurz angerissenen Ansätze können dazu beitragen, die Reaktionszeit auf das Auftreten neuer Fakeshops zu verkürzen und den Schutz der Verbraucher weiter zu verbessern.

## Literaturverzeichnis

1. Verisign. (2023, März 9). Verisign Domain Name Industry Brief: 350.4 Million Domain Name Registrations in the Fourth Quarter of 2022. *Verisign Blog*. Abgerufen 26. März 2023, von <https://blog.verisign.com/domain-names/verisign-q4-2022-the-domain-name-industry-brief/>
2. NDR. (o. J.). Fake Shops: So schützen sich Kunden vor Internet-Betrug. Abgerufen 26. März 2023, von <https://www.ndr.de/ratgeber/verbraucher/Fake-Shops-So-schuetzen-sich-Kunden-vor-Internet-Betrug,fakeshop104.html>
3. Straftaten im Bereich Internetkriminalität 2022. (o. J.). *Statista*. Abgerufen 28. Mai 2023, von <https://de.statista.com/statistik/daten/studie/12802/umfrage/internet-kriminalitaet-in-deutschland-anteile-der-delikte/>
4. S. Scheibe. Analyse von Fakeshops anhand des jeweiligen eingesetzten Content Management System mit Abfrage von gefundenen Merkmalen auf Basis von Python.pdf. 2023 (o. J.).
5. ÖIAT – Mit künstlicher Intelligenz Fake-Shops aufspüren – ACR. (2021, Mai 14). Abgerufen 28. Mai 2023, von <https://www.acr.ac.at/awards/innovationspreis-2020/fake-shops-aufspueren/>
6. Beltzung, L., Lindley, A., Dinica, O., Hermann, N., & Lindner, R. (2020). Real-Time Detection of Fake-Shops through Machine Learning. In *2020 IEEE International Conference on Big Data (Big Data)* (S. 2254–2263). Gehalten auf der 2020 IEEE International Conference on Big Data (Big Data). <https://doi.org/10.1109/BigData50022.2020.9378204>
7. Strobel, C. *Web-Technologien in E-Commerce-Systemen*. Walter de Gruyter GmbH & Co KG. 2017. ISBN: 978-3-486-81464-4
8. Admin, 4iM Kay Schoenewerk. (2022, Oktober 11). Der Unterschied zwischen E-Commerce und Online-Shopping. *4iMEDIA GmbH*. Abgerufen 3. Mai 2023, von <https://www.4imedia.com/ecommerce/der-unterschied-zwischen-e-commerce-und-online-shopping/>
9. Cleve, J., Lämmel, U., *Data Mining*. (2020) (3. Aufl.). De Gruyter. 3. Auflage, Walter de Gruyter GmbH & Co KG, 2020. ISBN 978-3-11-067623-5

10. Online-Kauf - Bevorzugte Zahlungsmethoden in Deutschland 2022. (o. J.). *Statista*. Abgerufen 3. Mai 2023, von <https://de.statista.com/statistik/daten/studie/677035/umfrage/bevorzugte-zahlungsmethoden-beim-online-einkauf-in-deutschland/>
11. Online-Shopping: Nutzung von Endgeräten 2022. (o. J.). *Statista*. Abgerufen 3. Mai 2023, von <https://de.statista.com/statistik/daten/studie/1340760/umfrage/online-shopping-nach-endgeraeten-in-deutschland/>
12. Online-Handel - Umsatz mit Waren in Deutschland 2022. (o. J.). *Statista*. Abgerufen 26. März 2023, von <https://de.statista.com/statistik/daten/studie/71568/umfrage/online-umsatz-mit-waren-seit-2000/?locale=de>
13. Online-Shopping: Häufigkeit nach Altersgruppen 2022. (o. J.). *Statista*. Abgerufen 3. Mai 2023, von <https://de.statista.com/statistik/daten/studie/1340699/umfrage/haeufigkeit-von-online-shopping-nach-altersgruppen-in-deutschland/>
14. Kompalka, K., & Ebel, D. (2018). Onlineshop-Systeme zur Digitalisierung des Handels.
15. eCommerce technologies Web Usage Distribution in Germany. (o. J.). Abgerufen 7. Mai 2023, von <https://trends.builtwith.com/shop/country/Germany>
16. Abzocke online: Wie erkenne ich Fake-Shops im Internet? (o. J.). *Verbraucherzentrale.de*. Abgerufen 7. Mai 2023, von <https://www.verbraucherzentrale.de/wissen/digitale-welt/onlinehandel/abzocke-online-wie-erkenne-ich-fakeshops-im-internet-13166>
17. Kann ich eine Überweisung zurückholen? - Volksbank Raiffeisenbank. (o. J.). Abgerufen 28. Mai 2023, von <https://www.vr.de/privatkunden/news/ueberweisung-zurueckholen.html>
18. Internet-Betrug: So können Sie versuchen, Ihr Geld zurückzuholen. (o. J.). *Verbraucherzentrale.de*. Abgerufen 28. Mai 2023, von <https://www.verbraucherzentrale.de/wissen/digitale-welt/onlinehandel/internetbetrug-so-koennen-sie-versuchen-ihr-geld-zurueckzuholen-33155>

19. Fake-Shops locken mit unschlagbaren Preisen in die Falle. (o. J.). Abgerufen 7. Mai 2023, von <https://www.polizei-beratung.de/themen-und-tipps/gefahren-im-internet/e-commerce/fake-shops/>
20. Abzocke online: Wie erkenne ich Fake-Shops im Internet? (o. J.). *Verbraucherzentrale.de*. Abgerufen 7. Mai 2023, von <https://www.verbraucherzentrale.de/wissen/digitale-welt/onlinehandel/abzocke-online-wie-erkenne-ich-fakeshops-im-internet-13166>
21. publisher. (o. J.). Woran erkenne ich sichere Onlineshops? *Bundesamt für Sicherheit in der Informationstechnik*. Abgerufen 28. Mai 2023, von <https://www.bsi.bund.de/DE/Themen/Verbraucherinnen-und-Verbraucher/Informationen-und-Empfehlungen/Online-Banking-Online-Shopping-und-mobil-bezahlen/Online-Shopping/Worauf-beim-Online-Einkauf-zu-achten-ist/worauf-beim-online-einkauf-zu-achten-ist.html?nn=131600>
22. Fakeshops. (o. J.). *RATGEBER INTERNETKRIMINALITÄT*. Abgerufen 28. Mai 2023, von <https://www.polizei-praevention.de/themen-und-tipps/straftaten-im-netz/fakeshops>
23. Über uns. (o. J.). *Watchlist Internet*. Abgerufen 29. Mai 2023, von <https://www.watchlist-internet.at/ueber-uns/>
24. Datenquellen. (o. J.). *Fakeshops*. Abgerufen 29. Mai 2023, von <https://www.fakeshop.at/ueber-fake-shop-detector/datenquellen/>
25. Sprachidentifikations-API | MeaningCloud. (o. J.). Abgerufen 29. Mai 2023, von <https://www.meaningcloud.com/developer/language-identification>
26. MariaDB und MySQL – Vergleich relationaler Open-Source-Datenbanken – AWS. (o. J.). *Amazon Web Services, Inc.* Abgerufen 2. Juni 2023, von <https://aws.amazon.com/de/compare/the-difference-between-mariadb-vs-mysql/>
27. Web Scraping With Python - Full Guide to Python Web Scraping. (2018, November 14). *Edureka*. Abgerufen 2. Juni 2023, von <https://www.edureka.co/blog/web-scraping-with-python/>

28. Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation. (o. J.). Abgerufen 3. Juni 2023, von <https://beautiful-soup-4.readthedocs.io/en/latest/>
29. Cron › Wiki › ubuntuusers.de. (o. J.). Abgerufen 4. Juni 2023, von <https://wiki.ubuntuusers.de/Cron/>



---

**Abbildungsverzeichnis**

Abbildung 1: Übersicht bevorzugter Zahlungsmethoden 2022 .....	14
Abbildung 2: Verteilung e-commerce Lösungen auf Webseiten (DE).....	18
Abbildung 3: Methodik und Inhaltlicher Aufbau .....	31
Abbildung 4: Bereich Online-Shops in Forum <a href="http://www.auktionshilfe.info">www.auktionshilfe.info</a> .....	35
Abbildung 5: Eingabe URL von einem Fakeshop in einer Suchmaschine .....	35
Abbildung 6: Liste von Fakeshops auf <a href="http://www.watchlist-internet.at">www.watchlist-internet.at</a> .....	36
Abbildung 7: <a href="http://wiedermann-autoreifen.com">wiedermann-autoreifen.com</a> wird als „sicher“ eingestuft.....	37
Abbildung 8: <a href="http://wiedermann-autoreifen.com">wiedermann-autoreifen.com</a> - Gelistet <a href="http://auktionshilfe.info">auktionshilfe.info</a> .....	38
Abbildung 9: Aufbau tech. Umgebung .....	44
Abbildung 10: Zusammenhang Python-Skripte .....	46
Abbildung 11: Ablaufdiagramm Skript <code>main.py</code> .....	47
Abbildung 12: Ablaufdiagramm <code>bt_fs_analyse.py</code> .....	48
Abbildung 13: Aufbau Skript <code>bt_fs_functions.py</code> .....	50
Abbildung 14: Ablauf Funktion <code>auktionshilfe_auslesen</code> .....	51
Abbildung 15: Ablauf Funktion <code>watchlist_auslesen()</code> .....	53
Abbildung 16: Ablauf Funktion <code>whatcms()</code> .....	55
Abbildung 17: Ablauf Funktion <code>whatcms_erweiterung()</code> .....	56
Abbildung 18: Ablauf Funktion <code>language_mc</code> .....	58
Abbildung 19: Ablauf Funktion <code>location_ip</code> .....	59
Abbildung 20: Inhalt Shellskript zur Ausführung Befehl.....	60
Abbildung 21 Crontab zur automatisierten Ausführung: <code>fakeshop_liste.sh</code> .....	60
Abbildung 22: SQL-Dumps der Datenbank <code>bt_fs</code> .....	61
Abbildung 23: Ablauf der Durchführung .....	63
Abbildung 24: Teilprozess Validierung und Datenbereinigung .....	65

---

Abbildung 25: Fakeshop lexer-rasenmaher.com - Falsches System.....	65
Abbildung 26: WordPress und WooCommerce als System + Erweiterung .....	66
Abbildung 27: Überprüfung Fakeshop mit Dienst builtwith.com .....	66
Abbildung 28: SQL-Statements zur Aktualisierung Datenbestand .....	67
Abbildung 29: Verwendete Systeme - Gesamtverteilung .....	69
Abbildung 30: Verwendete Systeme - Zielgruppe deutsch.....	70
Abbildung 31: Verwendete Systeme - Zielgruppe international.....	71
Abbildung 32: Erweiterung WooCommerce bei System WordPress .....	72
Abbildung 33: Erweiterung Cart functionality bei System PHP.....	73
Abbildung 34: Verteilung der Systeme auf Online-Shop Kategorien .....	75
Abbildung 35: WordPress Versionen und Anteile.....	76
Abbildung 36: PHP-Versionen und Anteile.....	76
Abbildung 37: Shopware Verteilung der Version 5 und 6 .....	77
Abbildung 38: Shopware Versionen - Verteilung bei Zielsprache deutsch .....	78
Abbildung 39: Shopware Versionen - Bei Zielsprache nicht deutsch .....	78
Abbildung 40: Einsatz von Cloudflare bei Fakeshops .....	79
Abbildung 41: Einsatz von Cloudflare bei Fakeshops – DE .....	79
Abbildung 42: Einsatz von Cloudflare bei Fakeshops – INT.....	80
Abbildung 43: Server-Standorte - Gesamt .....	82
Abbildung 44: Server-Standorte – DE .....	83
Abbildung 45: Server-Standorte – INT .....	84
Abbildung 46: Whois IP-Adresse 23.227.38.65.....	86
Abbildung 47: Whois IP-Adresse 185.254.198.235 .....	87
Abbildung 48: Whois IP-Adresse 104.16.198.133.....	88
Abbildung 49: Whois IP-Adresse 185.225.115.62 .....	88
Abbildung 50: Whois IP-Adresse 23.227.38.32 .....	89

**Tabellenverzeichnis**

Tabelle 1: Verwendete Systeme – Gesamt .....	69
Tabelle 2: Verwendete Systeme – Zielgruppe deutsch .....	70
Tabelle 3: Verwendete Systeme - Zielgruppe international .....	71
Tabelle 4: Erweiterung bei System WordPress .....	72
Tabelle 5: Erweiterung bei System PHP .....	73
Tabelle 6: Zuordnung Systeme zu Online-Shop Kategorien .....	74
Tabelle 7: Ergebnis Cloudflare mit Anteil bei Systemen .....	81
Tabelle 8: Fünf häufigste Server-Standorte.....	82
Tabelle 9: Fünf häufigste Server-Standorte – DE.....	83
Tabelle 10: Fünf häufigste Server-Standorte – INT .....	84
Tabelle 11: Fünf häufigsten IP-Adressen .....	85

**Verzeichnis der Abkürzungen**

AGB	Allgemeine Geschäftsbedingungen
API	Application Programming Interface
BGB	Bürgerliches Gesetzbuch
BSI	Bundesamt für Sicherheit in der Informationstechnik
CDN	Content-Delivery Network
CMS	Content Management System
CSV	Comma-separated values
DACH	Deutschland Österreich Schweiz
E-Commerce	Electronic Commerce
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
LCC	Limited Liability Company
LTS	Long term support
ÖIAT	Österreichische Institut für angewandte Telekommunikation
PHP	Hypertext Preprocessor
PKS	Polizeiliche Kriminalstatistik
SaaS	Software as a Service
SSL	Secure Sockets Layer
StGB	Strafgesetzbuch
URL	Uniform Resource Locator
USA	United States of America
XML	Extensible Markup Language

## **Selbstständigkeitserklärung**

Hiermit erkläre ich, dass ich die hier vorliegende Arbeit selbstständig, ohne unerlaubte fremde Hilfe und nur unter Verwendung der in der Arbeit aufgeführten Hilfsmittel angefertigt habe.

Hannover, 16.07.2023      Stefan Scheibe

Ort, Datum                      (Unterschrift)

**Anhangverzeichnis**

Anhang 1: Python-Skripte .....	A
bt_fs_functions.py .....	A
bt_fakeshops_csv.py .....	I

## Anhang 1: Python-Skripte

### bt\_fs\_functions.py

```
import re
import requests
from bs4 import BeautifulSoup
from datetime import date
import mariadb
import json
import time
import sys

# Connect to MariaDB Platform
try:
    conn = mariadb.connect(
        user="fs",
        password="*****",
        host="localhost",
        port=3306,
        database="bt_fs"
    )
except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
    sys.exit(1)

today = date.today()

# Funktion zum Auslesen von Fakeshops aus Plattform auktionenhilfe.info

def auktionenhilfe_auslesen():
    print("Starte Auslesen von Fakeshops von auktionenhilfe.info")
    print("-----")
    # Seite 1 bis 5 in Liste URLs einfügen
    urls = [ "https://www.auktionshilfe.info/board/22-online-shops/",
"https://www.auktionshilfe.info/board/22-online-shops/?pageNo=2",
"https://www.auktionshilfe.info/board/22-online-shops/?pageNo=3",
"https://www.auktionshilfe.info/board/22-online-shops/?pageNo=4",
"https://www.auktionshilfe.info/board/22-online-shops/?pageNo=5"
]

    # Posts hat tr mit class wbbThread jsClipboardObject new
    # Da drin (aus der Foren Übersicht (Vorteil, es muss kein Post
    geöffnet werden) ist ein a tag mit
```

```
# class messageGroupLink wbbTopicLink
# URLs durchgehen in Inhalt URLs extrahieren
# Domain auslesen und Liste übergeben
# Abgleich mit Datenbank und neue FS hinzufügen
for url in urls:
    page = requests.get(url)
    soup = BeautifulSoup(page.content, "html.parser")
    body = soup.find_all(
        "tr", attrs={"class": "wbbThread jsClipboardObject new"})
    list = []
    # Liste enthält immer 15 Einträge (eine Seite im Forum)
    for x in body:
        check = x(text=re.compile("Fakeshop"))
        list.append(check)

    # Eine Liste mit allen Ergebnissen erstellen (shops)
    second_list = [word for sentence in list for word in sentence]

    # Aus der Liste nun jeweils aus jedem Listenelement das Wort
    (Domain) nach Fakeshop nehmen
    # damit eine neue Liste (Shops) mit Domains erstellen
    list_shops_web = [word for words in second_list for i,
                      word in enumerate(words.split()) if i == 1]
    # print(list_shops_web)

    # SQL Abfrage, SELECT Domain FROM Domains
    # Inhalt in zweite Liste (zum späteren Vergleich) speichern
    mycursor = conn.cursor()
    mycursor.execute("SELECT Domain FROM fakeshops")
    list_shops_db = [item[0] for item in mycursor.fetchall()]
    # print(list_shops_db)

    # Listen vergleichen und nicht vorhandene Domains in neue Liste
    schreiben
    s = set(list_shops_db)
    list_shops_new = [x for x in list_shops_web if x not in s]
    # print(list_shops_new)

    # Die Liste mit neuen Domains wird in Tuple umgewandelt
    tuple_shops = tuple(i for i in list_shops_new)
    print(tuple_shops)

    mycursor = conn.cursor()
    for tupl in tuple_shops:
        query = "INSERT INTO fakeshops (domain, datum) VALUES ('%s',
'%s')" % (tupl, today)
        mycursor.execute(query)
```



```

conn.commit()

# Funktion zum Auslesen von Fakeshops aus Plattform watchlist-internet.at

def watchlist_auslesen():
    print("-----")
    print("Starte Auslesen von Fakeshops von Watchlist Internet")
    print("-----")
    # Seite 1 bis 9 in Liste URLs einfügen
    urls = [ "https://www.watchlist-internet.at/liste-betruegerischer-shops/", "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=2",
    "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=3", "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=4", "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=5",
    "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=6", "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=7", "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=8",
    "https://www.watchlist-internet.at/liste-betruegerischer-shops/?tx_solr%5Bpage%5D=9"
    ]

    # URLs durchgehen in Inhalt URLs extrahieren
    for url in urls:
        page = requests.get(url)
        soup = BeautifulSoup(page.content, "html.parser")
        body = soup.find_all("a", class_="site-item__link")
        list = []
        # Domains von Fakeshops in Liste aufnehmen
        for link in body:
            href = link['href']
            text = link.text.strip()
            list.append(text)

        # SQL Abfrage, SELECT Domain FROM Domains
        # Inhalt in zweite Liste (zum späteren Vergleich) speichern
        mycursor = conn.cursor()
        mycursor.execute("SELECT Domain FROM fakeshops")
        list_shops_db = [item[0] for item in mycursor.fetchall()]

        # Listen vergleichen und nicht vorhandene Domains in neue Liste schreiben
        s = set(list_shops_db)
        list_shops_new = [x for x in list if x not in s]

        # Die Liste mit neuen Domains wird in Tuple umgewandelt

```

```
tuple_shops = tuple(i for i in list_shops_new)
print(tuple_shops)

mycursor = conn.cursor()
for tupl in tuple_shops:
    # Domain der Fakeshops und heutiges Datum in Datenbank
    query = "INSERT INTO fakeshops (domain, datum) VALUES ('%s',
    '%s')" % (tupl, today)
    mycursor.execute(query)

conn.commit()

# Funktion zum Ermitteln System über Schnittstelle zu Whatcms.org

def whatcms():
    # SQL Abfrage, welcher Shop Status None/Null hat - Diesen dann
    # Alle Shops mit None/Null in Liste speichern
    überprüfen
    api_key = "*****"
    mycursor = conn.cursor()
    mycursor.execute("SELECT Domain FROM fakeshops where System is null")
    list_shops_db = [item[0] for item in mycursor.fetchall()]
    # Liste nacheinander durchgehen und auf CMS und Version prüfen
    mycursor = conn.cursor()
    for index, item in enumerate(list_shops_db):
        next = index + 1
        time.sleep(1)
        # Erstelle Dictionary mit Parameter für Abfrage der URL
        params = {
            "key": api_key,
            "url": item
        }
        if next < len(list_shops_db):
            response = requests.get("https://whatcms.org/API/Tech",
            params=params)
            cms_info = json.loads(response.text)
            if cms_info["result"]["code"] == 200:
                # Shop ist online
                status = "Online"
                data = response.json()
                # Lese das verwendete CMS aus
                cms = data["results"][0]["name"]
                # Lese die verwendete Version des CMS aus
                version = data["results"][0]["version"]
```

```
        # Prüfen ob Shop durch Cloudflare geschützt, wenn
        # vorhanden, dann String Ja einfügen. Wenn nicht vorhanden, dann String
        # Nein einfügen
        if "Cloudflare" in str(data):
            print(item + " " + status + " " + cms)
            query = "UPDATE fakeshops SET Status = '%s', System =
            '%s', Version = '%s', Cloudflare = 'Ja' WHERE Domain = '%s'" % (status,
            cms, version, item)
            mycursor.execute(query)
            conn.commit()
        # Datenbank mit den gefundenen CMS und der Version
        # aktualisieren
        else:
            print(item + " " + status + " " + cms)
            query = "UPDATE fakeshops SET Status = '%s', System =
            '%s', Version = '%s', Cloudflare = 'Nein' WHERE Domain = '%s'" % (status,
            cms, version, item)
            mycursor.execute(query)
            conn.commit()
        elif cms_info["result"]["code"] == 201:
            # Wenn CMS unbekannt ist, dann "Unbekannt" einfügen
            print(item + " CMS nicht bekannt")
            query = "UPDATE fakeshops SET System = 'Unbekannt',
            Version = 'Unbekannt' WHERE Domain = '%s'" % (item)
            mycursor.execute(query)
            conn.commit()
        elif cms_info["result"]["code"] == 202:
            # Wenn der Shop bereits offline ist, "Offline" einfügen
            print(item + " Fakeshop bereits offline")
            query = "UPDATE fakeshops SET Status = 'Offline', Sprache
            = 'Unbekannt', System = 'Unbekannt', Version = 'Unbekannt', Erweiterung =
            'Unbekannt', Standort = 'Unbekannt', IP = 'Unbekannt', Cloudflare =
            'Unbekannt' WHERE Domain = '%s'" % (item)
            mycursor.execute(query)
            conn.commit()
        else:
            print(f"{item} verwendet kein bekanntes CMS.")
            # Time Sleep von sechs, da nur alle fünf Sekunden eine
            # Abfrage bei whatcms.org möglich ist
            time.sleep(6)

# Funktion zum Ermitteln einer Erweiterung für das System über
# Schnittstelle zu Whatcms.org

def whatcms_erweiterung():
    # SQL Abfrage, welcher Shop WordPress ist und gleichzeitig Plugin auf
    # null steht
```

```
# Alle Shops mit None/Null in Liste speichern
api_key = "*****"
mycursor = conn.cursor()
mycursor.execute("SELECT Domain FROM fakeshops WHERE System =
'WordPress' AND Erweiterung is null")
list_shops_db = [item[0] for item in mycursor.fetchall()]
# Liste nacheinander durchgehen und auf CMS und Version prüfen
mycursor = conn.cursor()
for index, item in enumerate(list_shops_db):
    next = index + 1
    time.sleep(1)
    # Erstelle Dictionary mit Parameter für Abfrage der URL
    params = {
        "key": api_key,
        "url": item
    }
    if next < len(list_shops_db):
        response = requests.get("https://whatcms.org/API/Tech",
params=params)
        cms_info = json.loads(response.text)
        if cms_info["result"]["code"] == 200:
            data = response.json()
            plugin = data["results"][1]["name"]
            query = "UPDATE fakeshops SET Erweiterung = '%s' WHERE
Domain = '%s'" % (plugin, item)
            mycursor.execute(query)
            conn.commit()
        else:
            print(f"{item} verwendet kein bekanntes Plugin.")
            # Time Sleep von 15, da nur alle fünf Sekunden eine Abfrage
            # bei whatcms.org möglich ist
            time.sleep(15)

# Funktion zum Ermitteln der native Sprache über Schnittstelle zu
MeaningCloud

def language_mc():
    # SQL Abfrage, welcher Shop Status None/Null hat - Diesen dann
    # überprüfen
    # Alle Shops mit None/Null in Liste speichern

    # API-Schlüssel von MeaningCloud
    api_key = '*****'
    # API-Endpoint
    endpoint = 'https://api.meaningcloud.com/lang-4.0/identification'
    mycursor = conn.cursor()
```

```
mycursor.execute("SELECT Domain FROM fakeshops where Sprache is
null")
list_shops_db = [item[0] for item in mycursor.fetchall()]
# Liste nacheinander durchgehen und auf CMS und Version prüfen
mycursor = conn.cursor()
for index, item in enumerate(list_shops_db):
    next = index + 1
    time.sleep(1)
    params = {
        "key": api_key,
        "url": 'https://' + item
    }
    if next < len(list_shops_db):
        time.sleep(2)
        # GET-Request an die MeaningCloud Language Identification API
senden
        response = requests.get(endpoint, params=params)
        # JSON-Daten aus der HTTP-Response extrahieren
        json_data = response.json()
        if 'language_list' in json_data:
            # Sprache des Inhalts aus der JSON-Antwort extrahieren
            try:
                content_language =
json_data['language_list'][0]['name']
                print(item + " " + content_language)
                # Datenbank mit den gefundenen CMS und der Version
aktualisieren
                query = "UPDATE fakeshops SET Sprache = '%s' WHERE
Domain = '%s'" % (content_language, item)
                mycursor.execute(query)
                conn.commit()
            except:
                print(item + " Unbekannt")
            else:
                query = "UPDATE fakeshops SET Sprache = 'Unbekannt' WHERE
Domain = '%s'" % (item)
                mycursor.execute(query)
                conn.commit()
            else:
                print(f"{item} Abfrage zur Identifizierung der Sprache konnte
nicht durchgeführt werden")
                # Time Sleep von 5, da nur alle fünf Sekunden eine Abfrage
möglich ist
                time.sleep(2)

# Funktion zum Ermitteln der IP-Adresse und Standort des Hosts über
Schnittstelle zu ipstack.com
```

```
def location_ip():
    # SQL Abfrage, welcher Shop Status None/Null hat - Diesen dann
    # überprüfen
    # Alle Shops mit None/Null in Liste speichern
    # API-Schlüssel von IPStack
    api_key = '*****'
    mycursor = conn.cursor()
    mycursor.execute("SELECT Domain FROM fakeshops where Standort is
null")
    list_shops_db = [item[0] for item in mycursor.fetchall()]
    # Liste nacheinander durchgehen und auf CMS und Version prüfen
    mycursor = conn.cursor()
    for index, item in enumerate(list_shops_db):
        next = index + 1
        print(item)
        time.sleep(3)
        if next < len(list_shops_db):
            # GET-Request an die IPStack API senden
            response =
requests.get(f"http://api.ipstack.com/{item}?access_key={api_key}")
            if response.status_code == 200:
                data = response.json()
                country = data["country_name"]
                ip_address = data["ip"]
                print(item + " " + country)
                query = "UPDATE fakeshops SET Standort = '%s', IP = '%s'
WHERE Domain = '%s'" % (country, ip_address, item)
                mycursor.execute(query)
                conn.commit()
            else:
                print(item + " Unbekannt")
                query = "UPDATE fakeshops SET Standort = 'Unbekannt', IP
= 'Unbekannt' WHERE Domain = '%s'" % (item)
                mycursor.execute(query)
                conn.commit()
        else:
            print("Daten für Standort und IP-Adresse können nicht
ermittelt werden")
            # Time Sleep von 5, da nur alle fünf Sekunden eine Abfrage
möglich ist
            time.sleep(5)
```

**bt\_fakeshops\_csv.py**

```
import csv
import datetime
import os
import mariadb
import sys

# Connect to MariaDB Platform
try:
    conn = mariadb.connect(
        user="fs",
        password="*****",
        host="localhost",
        port=3306,
        database="bt_fs"
    )
except mariadb.Error as e:
    print(f"Error connecting to MariaDB Platform: {e}")
    sys.exit(1)

# Verzeichnis erstellen, falls es nicht existiert
output_dir = "/home/ubuntu/BT/csv"
if not os.path.exists(output_dir):
    os.makedirs(output_dir)

# Cursor-Objekt erstellen
cursor = conn.cursor()

# MySQL-Abfrage definieren
query = "SELECT * FROM fakeshops"

# MySQL-Abfrage ausführen
cursor.execute(query)

# Alle Zeilen aus dem Abfrageergebnis abrufen
rows = cursor.fetchall()

# CSV-Datei mit aktuellem Datum im Dateinamen erstellen und schreiben
filename =
"fakeshops_csv_{}.csv".format(datetime.datetime.now().strftime("%d_%m_%Y"
))
filepath = os.path.join(output_dir, filename)
with open(filepath, 'w', newline='') as file:
    writer = csv.writer(file)
    # Spaltennamen als erste Zeile schreiben
    writer.writerow([i[0] for i in cursor.description])
    # Daten schreiben
```

```
    for row in rows:
        writer.writerow(row)

# Cursor und Verbindung schließen
cursor.close()
conn.close()
```