Kapitel 4: Automatisierte Erkennung von Desinformationen

Autoren:

Oren Halvani
Wendy Freifrau Heereman von Zuydtwyck
Michael Herfert
Dr. Michael Kreutzer
Dr. Huajian Liu
Hervais-Clemence Simo Fhom
Prof. Dr. Martin Steinebach
Inna Vogel
Ruben Wolf
York Yannikos
Dr. Sascha Zmudzinski

Dieses Kapitel adressiert die Frage, wie Desinformationen automatisiert erkannt werden können. Die Notwendigkeit dieser Betrachtung leitet sich von der Menge an entsprechenden Inhalten ab, die potentiell erzeugt und verbreitet werden können. Ohne eine Möglichkeit, hier automatisch oder zumindest semi-automatisch Inhalte zu erkennen und zu filtern, sind die für entsprechende Vorgänge Verantwortlichen schnell überfordert.

Dabei wird von einem Anwender ausgegangen, der eine Beurteilung von eingehenden Meldungen durchführt und dabei technisch unterstützt wird. Dies adressiert insbesondere Redakteure und Medienschaffende. Es soll die Grundlage für ein Werkzeug oder treffender eine Sammlung von Werkzeugen geschaffen werden, mit denen Nutzende Hinweise für eine Manipulation der Inhalte einer Meldung oder ihre desinformierende Natur sammeln können. Diese Hinweise können beispielsweise darin bestehen, dass ein Foto bereits in einer früheren Pressemitteilung verwendet wurde und nun in einem anderen Kontext verwendet wird. Oder dass ein Foto Spuren von einer Bearbeitung aufweist, die möglicher Weise seine Aussage verändern. Das Erzeugen dieser Hinweise soll dabei automatisiert geschehen, die Interpretation hingegen muss heute noch durch Experten erfolgen. Die Technik assistiert also dem Anwender bei der Prüfung.

Die technischen Untersuchungen adressieren derzeit noch die Medientypen Text, Bild und Video jeweils für sich alleine. Jeder Medientyp weist dabei eigene Manipulationstypen und Erkennungsmethoden auf. Neben der Erkennung sind allerdings auch noch andere Aspekte zu beachten. Eine Automatisierung kann nur über eine Einbindung in ein Gesamtsystem geschehen. Die zu untersuchenden Medien müssen zuerst aus Quellen wie Social Networks oder Nachrichtenseiten gewonnen werden. Die gewonnenen Daten müssen datenschutzkonform behandelt werden, was beispielsweise eine sichere Ablagestrategie erfordert. Und die Daten müssen bereinigt werden, damit die eigentlichen Analysewerkzeuge nicht durch Rauschen gestört werden.

Bevor in den folgenden Abschnitten der Stand der Forschung und die eigenen Ergebnisse diskutiert werden, sollen zuerst noch einige Begriffe und ihre Verwendung in der Technik eingeführt werden. Die Echtheit eines Inhalts hat in der IT Sicherheit mehrere Aspekte. Eine Frage ist, ob das Dokument und sein Erzeuger die sind, die sie vorgeben zu sein. Dies ist die Frage der Authentizität. Ein Foto, welches von einem bekannten Pressefotografen stammt, wird ein hohes Vertrauen genießen. Es kann aber sein, dass entweder der Fotograf das Foto nicht erstellt hat, sondern seine Urheberschaft nur angegeben wurde, oder dass das Foto selbst ausgetauscht wurde. Ein Beispiel: In einer Kriegsberichtserstattung wird ein Foto verwendet, welches angeblich von dem bekannten (fiktiven) Fotografen Herrn Max Mustermann geschossen wurde. Es zeigt vorrückende Panzer. Die Authentizität stellt zum einen die Frage, ob Herr Mustermann wirklich der Ersteller des Fotos ist. Sie hinterfragt in dem Fall, dass Herr Mustermann tatsächlich Fotos aus dem Kriegsbericht erstellt hat, auch, ob genau dieses Foto auch tatsächlich aus der Menge der Fotos stammt. Weiterhin stellt sich auch die Frage der Integrität. Es ist denkbar, dass ein Foto verändert wurde, um seinen Inhalt zu manipulieren. Dabei ist es unwichtig, ob dies von einer dritten Partei oder dem ursprünglichen Fotografen stammt. Erst, wenn sowohl Authentizität als auch Integrität belegt sind, ist ein Dokument vertrauenswürdig.

A. Überblick über die Forschungslandschaft

Der Bedarf nach einer computergestützten Erkennung von Desinformation ist nicht neu. Insbesondere im englischsprachigen Raum ist hier, seitdem das Phänomen breite Beachtung erfährt, eine Reihe von Forschungsarbeiten durchgeführt worden. Unterschieden werden kann hierbei zwischen Ansät-

zen auf textueller Ebene und auf Basis von Metadaten. Erstere verwenden computerlinguistische Methoden, um Texte mit entsprechendem Inhalt zu erkennen. Letztere nutzen Informationen wie die Aktivitäten von Benutzerkonten oder IP-Adressen, um Bot-Netze zu identifizieren, welche zur Verbreitung eingesetzt werden.

Darüber hinaus sind aber auch Arbeiten aus der Multimedia-Forensik von Bedeutung. Auch die manipulierende Verwendung von Bildern und Videos wird betrachtet, daher sind auch Arbeiten zur Erkennung entsprechender Manipulationen von Bedeutung.

I. Erkennung von Desinformationen bei Texten

Es existieren Webseiten, die in der Vergangenheit aufgefallen sind, Desinformationen (im Folgenden auch „Fake News“ genannt) verbreitet zu haben. Im englischsprachigen Raum gehören zu solchen Webseiten beispielsweise denverguardian.com, wtoe5news.com oder ABCnews.com.co. Oft sind solche Webseiten professionell erstellt und kaum von großen Mainstream-Medienwebseiten zu unterscheiden. Der Hauptzweck der Fake-News-Webseiten ist allerdings die Verbreitung politischer Propaganda oder Profitgenerierung durch die Verbreitung von Desinformationen (beispielsweise durch „Clickbaiting“). Neben den Nachrichtentypen (Propaganda und Clickbaiting) existieren weitere Genres, welche an der Verbreitung von Fake News beteiligt sind z. B. Satire, Verschwörungstheorien, Hoax, „Promi-News“ oder Falschaussagen (z. B. von Politikern) (Rashkin et al., 2017; Rubin et al., 2015).

Allcott et al. (2017) auf der anderen Seite schließen 1) Fehlermeldungen, 2) unbestätigte Gerüchte über Menschen, Ereignisse oder Organisationen, 3) Verschwörungstheorien, 4) Satire, 5) falsche Aussagen von Politikern und 6) Artikel, die irreführend, aber nicht völlig falsch sind, aus dem Konzept von Fake News aus. Die folgende Grafik zeigt unterschiedliche Fake-News-Typen und wie diese ihrem Wahrheitsgehalt und ihrer Intention zugeordnet werden können.

Neben unterschiedlichen Typen existieren auch unterschiedliche Medienträger und Internet-Plattformen (z. B. Webseiten, Social-Networks-Kanäle, Instant-Messaging-Dienste etc.), welche zur Verbreitung von Fake News benutzt werden. An der Verbreitung können Menschen, aber auch sogenannte Bots oder Cyborgs beteiligt sein. Bots sind Computerprogramme, welche überwiegend in Social Networks automatisch Inhalte verbreiten. Cyborgs

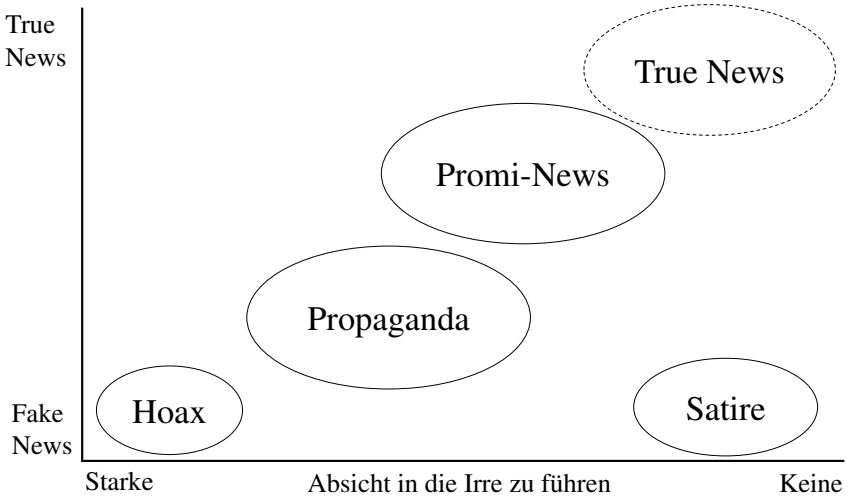

Abbildung 4.1: Nachrichtentypen angelehnt an Rashkin et al. (2017) und Rubin et al. (2015)

sind Accounts, welche zwar von Menschen betrieben werden, deren Inhaltsverbreitung allerdings automatisch abläuft.

Für wissenschaftliche Analysezwecke können sowohl physische, als auch Metadaten einer Meldung verwendet werden. Ein Nachrichtentext verfügt beispielsweise über einen Titel, Textkörper, Veröffentlichungsdatum oder auch zur Nachricht gehörende Fotos und Videos. Als nicht-physisch kann die Semantik des Textes zu Untersuchungszwecken herangezogen werden, beispielsweise der Inhalt der Nachricht, Emotionen, Stimmungen oder das Thema. Alle Eigenschaften (Features) der Nachricht können verwendet werden, um Nachrichtentexte ihrem Wahrheitsgehalt nach zu klassifizieren. Dabei werden die folgenden drei Analyseverfahren unterschieden: linguistic and semantic-based analysis, knowledge-based analysis und style-based analysis.

1. Knowledge-based Analysis

Beim wissensbasierten Ansatz wird der Wahrheitsgehalt eines Nachrichtenartikels direkt (manuell oder maschinell) geprüft (Shu et al., 2017). Bei der manuellen Überprüfung wird der Inhalt des Artikels von Experten z. B. Fachjournalisten überprüft, indem Fakten aus unterschiedlichen Quellen

verglichen werden (Banko et al., 2008). Im englischsprachigen Raum existieren unzählige „Fact-Checking"-Webseiten wie Snopes.com, PolitiFact.com und FactCheck.org. Die Inhalte der Nachrichtenartikel werden von Experten nach dem Wahrheitsgehalt der Behauptungen im Text klassifiziert. Solche Datenquellen sind essentiell für maschinelle Lernverfahren, da sie Trainingsdatenmaterial für den Algorithmus liefern.

Für den wissensbasierten Ansatz werden ebenfalls Crowdsourcing-Ressourcen verwendet. Dabei werden verdächtige Inhalte gemeldet und anschließend von Redakteuren geprüft. Auf diese Weise wird nicht jeder Text überprüft, sondern nur Texte, welche von Nutzenden gemeldet wurden, was einen geringeren Arbeitsaufwand bedeutet.

Neben dem manuellen und Crowdsourcing-Ansatz existieren auch automatische Systeme und Tools, welche Nachrichtenartikel aus dem Internet extrahieren und den Inhalt mit Informationen einer Wissensdatenbank abgleichen. Bestehen Widersprüche zwischen den Texten, wird dies vom System gemeldet (Shu et al., 2017; Banko et al., 2008). Pan et al. (2018) haben unterschiedliche automatische computerbasierte Ansätze zur Erkennung von Fake News getestet und konnten eine Gesamterkennungsrate von etwa 80 % erreichen. Dennoch adressieren die Autoren einige technische Herausforderungen und merken an, dass die rechenorientierte Faktenprüfung nicht umfassend genug ist, um alle Zusammenhänge abzudecken, die für die Erkennung gefälschter Nachrichten notwendig sind.

2. Style-based Analysis

Stilbasierte Ansätze zielen darauf ab, desinformierende Nachrichten anhand des Schreibstils des Verfassers zu erkennen. Hierfür werden syntaktische und semantische Textmerkmale wie etwa die Rechtschreibung, Grammatik, Interpunktion, Wortwahl, Satzbau, Absatzstruktur analysiert. Geeignete Merkmale, um den Schreibstil in Social Networks zu bestimmen sind Tokens wie etwa die Anzahl der URLs, Hashtags, @-Accountnennungen und die Verwendung von Großbuchstaben (Castillo et al., 2011; Horne & Adali 2017; Jin et al., 2016).

Es ist bekannt, dass Schöpfer von Falschnachrichten ihren Schreibstil versuchen zu verschleiern, um nicht als Autor identifiziert zu werden. Es werden aber auch Maschinen trainiert, welche automatisch Falschnachrichten generieren (Social Bots). Oder es werden Inhalte von Webseiten plagiiert, um

diese auf der eigenen Webseite anzubieten und auf diese Weise Einnahmen zu generieren (Potthast et al., 2016).

Um Verfasser von Fake News anhand ihrer Schreibstile zu überführen, haben Potthast et al. (2018) das Unmasking-Verfahren angewendet und auf diese Weise die Stilähnlichkeit zwischen Mainstream-Nachrichtentexten, Satire sowie extrem rechts- und linksseitigen Texten bestimmt. Unmasking entfernt iterativ die stärksten und schwächsten Features, sodass die Texte nach jeder Iteration anhand der übrig geblieben Merkmale verglichen werden. Mit den stärksten und schwächsten Merkmalen sind solche gemeint, die zur Unterscheidung der Texte am meisten bzw. wenigsten beitragen. Das Resultat sind sogenannte Degradationskurven, wobei jeder Text-zu-Text-Vergleich genau eine solche Degradationskurve darstellt. Kurven die stark abfallen bedeuten, dass die Texte nach der iterativen Entfernung von Merkmalen wesentlich schlechter unterschieden werden können (und dementsprechend vom selben Autor stammen) als Kurven, die kaum bis gar nicht abfallen.

Die extrem einseitig verfassten Texte („hyperpartisan"), weisen nach der Unmasking-Methode einen ähnlichen Schreibstil auf und können von Mainstream-Texten mit einer Genauigkeit von 78 % unterschieden werden. Satire kann zu 81 % richtig bestimmt werden. Shu et al. (2017) haben ebenfalls in ihrer Untersuchung gezeigt, dass Satire sowohl von extrem links- und rechtsseitigen Nachrichtentexten als auch Mainstream-Artikeln diskriminiert werden kann.

In der Forschung wurden unterschiedliche maschinelle Lernverfahren angewendet, um Fake News zu erkennen. Zu den klassischen Verfahren gehören beispielsweise Entscheidungsbäume (Decision Trees), Support Vector Machines (SVM), die Logistische Regression und der Nächste-Nachbarn-Klassifikator (KNN) (Afroz et al., 2012; Castillo et al., 2011; Davis et al., 2016; Horne & Adali, 2017; Kwon et al., 2013; Tacchini et al., 2017; Yang & Counts, 2010). Neben den unterschiedlichen Lernverfahren wurden ebenfalls unterschiedliche Datensets und Textmerkmale verwendet, um Fake News zu erkennen. Ebenfalls Anwendung finden „Deep Learning Algorithmen", welche nicht auf explizit definierte Textmerkmale angewiesen sind und Nachrichtenkontextinformationen selbständig erlernen.

Überwachte Lernverfahren hängen stark von der Qualität des Datensatzes ab. Aus folgenden Gründen ist es jedoch schwierig, einen qualitativ hochwertigen Datensatz zur Erkennung von Falschnachrichten zu generieren: Die Daten im Internet sind oft unstrukturiert, verrauscht und unvollständig (Shu et al., 2017). Jeden Tag erhöht sich die Menge an Falschinformationen, welche mit unterschiedlichen Absichten verfasst werden. Unüberwachte Lernmodel-

le sind daher praktikable Lösungen für solche Probleme aus der Praxis. Es gibt jedoch nur wenige Studien, die unüberwachte Lernverfahren zur Erkennung von Fake News anwenden. Die meisten von ihnen konzentrieren sich auf semantische Ähnlichkeits- oder Sentimentanalysen. Ahmed (2017) schlägt in seiner Dissertation ein unüberwachtes Ähnlichkeitsmessverfahren vor zur Erkennung von gefälschten Rezensionen. Für sein Verfahren verwendete er Wort- und Wortfolgenähnlichkeiten als Textmerkmale und konnte auf diese Weise ähnliche Rezensionen bzw. Duplikate mit einer hohen Erkennungsrate klassifizieren. Dieses Verfahren könnte künftig für die Erkennung von Falschinformationen eingesetzt werden, da Nachrichten von Qualitätsmedien oft plagiiert und in leicht abgewandelter Form für bestimmte politische oder soziale Ziele missbraucht werden.

Abschließend lässt sich festhalten, dass viele Techniken zur automatischen Erkennung gefälschter Nachrichten vorgeschlagen und angewendet werden. Die Faktenüberprüfung obliegt nichtsdestotrotz immer noch dem Wissen der Menschen.

II. Erkennung von Bildmanipulationen

Die Identifizierung von Bildmanipulationen ist ein komplexes Feld, insbesondere, wenn man auch die Wiederverwendung von Bildern bereits als potenziellen Angriff betrachtet. Es existieren aber auch zahlreichen Methoden zur Bildmanipulation, die die Bedeutung eines Bildes verändern. Hinzu kommt die Wiederverwendung von Bildern, die nicht verändert, sondern in einem anderen Kontext verwendet werden. Bildmontagen sind eine Kombination aus beiden Ansätzen: Ein bestehendes Bild wird außerhalb des Kontextes verwendet, aber auch manipuliert, indem ein weiteres Objekt aus anderen Bildern hinzugefügt wird, um seine Bedeutung zu ändern.

Es gibt viele Methoden zur Erkennung von Manipulationen in digitalen Bildern. Heute lassen sie sich grob zwei Klassen zuordnen: Methoden, die auf vom Menschen definierten Modellen und Mustern basieren, und Methoden, die auf überwachtem maschinellen Lernen basieren. Für die erste Gruppe sind bereits mehrere Veröffentlichungen, die einen Überblick zum Thema bieten, beispielsweise von Bayram (2008) oder Birajdar (2013), verfügbar.

Ein generischer Ansatz besteht darin, Unterschiede in den Stärken von Fehlern zu berechnen, die durch unterschiedliche Kompressionshistorien von Bereichen verursacht werden, wie beispielsweise von Luo (2010) beschrieben.

Die Re-Identifikation von Bildern ist eine Domäne, in der robuste Hashverfahren (Shin, 2013) oder wahrnehmungsbasierte Hashverfahren (Niu, 2008) die besten Ergebnisse liefern.

Die Herausforderung bei der Re-Identifikation besteht darin, Bilder zu identifizieren, die Kopien einer bekannten Quelle sind, ohne zu empfindlich auf die Ablehnung von Bildänderungen durch übliche Bildverarbeitung, z. B. verlustbehaftete Kompression, zu reagieren. Andererseits muss man es vermeiden, ähnliche Bilder als Duplikate zu identifizieren, da die Ähnlichkeit nicht für eine tatsächliche Anerkennung der Wiederverwendung qualifiziert.

1. Manipulationserkennung durch maschinelles Lernen

Es gibt zwei Arten von Manipulationen an digitalen Bildern: zufällige Manipulationen und absichtliche Manipulationen. Ersteres beinhaltet die durch die gängige Bildverarbeitung verursachten Änderungen wie Rauschfilterung, Skalierung, Komprimierung und so weiter. Letzteres bezieht sich in der Regel auf Bildfälschungen, die auf Inhaltsänderungen wie das Einfügen und Entfernen von Objekten abzielen. Ziel der Bildverarbeitung ist es nicht, den Bildinhalt zu verändern, sondern die Bildqualität oder Speichereffizienz zu verbessern. Die Bildverarbeitung ändert nur die binäre Darstellung des Bildes, während die zugrundeliegende semantische Bedeutung intakt bleibt. Daher wird die zufällige Manipulation auch als inhaltserhaltende Manipulation bezeichnet. Im Gegensatz dazu beabsichtigt die Bildfälschung, die semantische Bedeutung des Bildes zu bearbeiten, die auch als inhaltsverändernde Manipulation oder böswillige Manipulationen bezeichnet wird.

Bildforensik zielt auf die Erkennung beider Arten von Manipulationen ab. Die Erkennung absichtlicher Manipulationen kann die manipulierten Teile lokalisieren und die Absicht der Fälschung ableiten. Das Erkennen von zufälligen Manipulationen kann den Verarbeitungsverlauf aufdecken und auf mögliche Manipulationen hinweisen. In der Praxis erfolgt in der Regel nach der Bildverfälschung eine weitere Bildverarbeitung, um die Bearbeitungsspuren zu verbergen. Zum Beispiel zeigt das Erkennen von Skalierung und Filterung an, dass ein Bild nach der Aufnahme bearbeitet wurde. Die Existenz der doppelten JPEG-Komprimierung verrät, dass ein Bild nicht die Originalkopie ist.

Klassische bildforensische Techniken versuchen, bestimmte Bildoperationen, wie z. B. Medianfilterung, Skalierung, doppelte JPEG-Komprimierung, Copy-Move, etc., durch Untersuchung der entsprechenden Spuren zu er-

kennen [1-2]. Die Erkennung basiert auf der manuellen Analyse von manipulierten Bildern und extrahierten Merkmalen, deren Eigenschaften die eindeutigen Spuren der Manipulationen zeigen. Die Merkmale sind in der Regel mit der Manipulation verbunden und ein bestimmtes Merkmal wird verwendet, um eine bestimmte Manipulation zu erkennen. Daher ist jede klassische forensische Technik auf eine bestimmte Bildoperation ausgerichtet. Zum Beispiel kann der lokale Rauschpegel verwendet werden, um Image Splicing zu erkennen, indem die Rauschvarianz innerhalb kleinerer Bildblöcke bewertet wird. Die statistischen Merkmale der DCT-Koeffizienten in JPEG werden allgemein angewendet, um doppelte Komprimierung zu erkennen und die Manipulationen zu lokalisieren. Darüber hinaus werden einige allgemeine Low-Level-Merkmale, wie z. B. Pixel-Kookkurrenz, auch zur Unterscheidung verschiedener Bildverarbeitungsoperationen verwendet.

Im Gegensatz dazu verwenden Deep Learning forensische Methoden neuronale Netze, um wichtige Merkmale automatisch aus einem großen Trainingsdatensatz zu lernen [3-5]. Aufgrund seines sehr guten Ergebnisses für die Objekterkennungsaufgabe in digitalen Bildern wird das Convolutional Neural Network in der Bildforensik zur Erkennung von Manipulationen eingesetzt. Eine große Herausforderung für eine effektive Erkennung von Bildmanipulationen stellt die verbreitete JPEG-Komprimierung dar, die sowohl für klassische als auch für tief lernende Ansätze gilt. Die JPEG-Komprimierung verwirft Bildinformationen im Hochfrequenzbereich, was dazu führt, dass die Spuren, die bei Manipulationsoperationen hinterlassen werden, verringert oder zerstört werden, während gleichzeitig neue spezifische Spuren eingeführt werden. Abhängig vom Qualitätsfaktor eliminiert die JPEG-Komprimierung mehr oder weniger Bilddetails.

2. Merkmalserkennung

Die Merkmalserkennung besteht darin, sogenannte Schlüsselpunkte (Keypoints) an mehreren Stellen in einem Bild mit einem Detektor zu erkennen und Deskriptoren mit einem Merkmals-Extraktor zu extrahieren. In einem weiteren Schritt, dem Merkmalsvergleich, werden die gefundenen Merkmale mit Merkmalen eines anderen Bildes verglichen. Wenn beide Bilder nun das gleiche Objekt enthalten, sollten die Merkmale idealerweise messbar ähnlich sein. Ein Merkmal selbst ist definiert als ein „interessanter" Teil des Bildes. Was genau als „interessanter" Teil des Bildes verstanden wird, variiert je nach Merkmalsdetektor. Der Bildteil, in dem ein Merkmal extrahiert wird,

ist oft entweder ein isolierter Punkt, eine kontinuierliche Kurve oder ein verbundener Bereich.

Der Scale Invariant Feature Transform (SIFT) (Lowe, 2004) Algorithmus ist wahrscheinlich einer der bekanntesten und am häufigsten verwendeten Merkmals-Detektoren. SIFT findet Schlüsselpunkte in einem Bild mit dem Difference-of-Gaussian (DOG) Operator. DOG wird verwendet, um über mehrere Bildgrößen nach lokalem Extrema zu suchen. So können Schlüsselpunkte gefunden werden, die auch bei einer Änderung der Bildgröße erhalten bleiben. Für jeden dieser Schlüsselpunkte wird dann die stärkste Orientierung aus der Nachbarschaft zugewiesen.

Der Speeded Up Robust Features (SURF) Detektor (Bay, 2006) ist teilweise von SIFT inspiriert und ist ein Versuch, schneller und robuster zu sein als SIFT. Wie SIFT basiert auch SURF auf dem Difference-of-Gaussian (DOG). Um Schlüsselpunkte zu erkennen, verwendet SURF eine ganzzahlige Approximation eines Bildobjekt-Detektors.

3. Erkennung von Deepfake-Videos

Eine besondere Art der Manipulation zielt in Bildern und, vor allem, in Videos auf darin abgebildete Personen: Mit der sog. Deepfake-Technologie kann das Gesicht einer Person A automatisiert durch das Gesicht einer anderen Person B beinahe „lebensecht" ersetzt werden: Mimik und Mundbewegungen, die Kopfhaltung und auch das passende Größenverhältnis und die perspektivische Ansicht werden für das ausgetauschte Gesicht in jedem Einzelbild des Videos automatisch eingepasst.

Hiermit lassen sich innerhalb weniger Stunden bis Tage gefälschte Videos erzeugen, die die (Ziel-) Person B scheinbar in einer kompromittierenden Situation zeigen, indem ihr Gesicht in u. U. brisantes Videomaterial nachträglich eingefügt wird (etwa in Pornografie).

Die Technologie ist frei verfügbar, beispielsweise als kostenfreie Software von Kowalski (2018), deepfakes (2019), shaoanlu/clarle (2019), als „Deepfakes FakeApp" aus diversen Internetquellen oder – noch bequemer – als Online-Dienstleistung bei Deepfakes web β (2019).

Der Name „Deepfake" leitet sich davon ab, dass Techniken des sog. Deeplearnings aus dem Gebiet des maschinellen Lernens genutzt werden.

Auch Verfahren zur Erkennung dieser Deepfake-Videos sind bekannt: Einige basieren darauf, dass die Bildbereiche der gefälschten Gesichter einen anderen „technischen Lebenszyklus" durchlaufen haben als die unveränder-

ten Bildbereiche des Bildhintergrundes. Das betrifft die Anzahl und Parameter der Video- und Bildkompressionen (Bianchi & Piva, 2012) oder die Vergrößerung und/oder Rotation der eingepassten Bildinhalte (Li, Yuan et al., 2009), welche in den echten und gefälschten Bildbereichen voneinander abweichende forensische Spuren hinterlassen.

Andere Verfahren können ggf. eine unnatürliche Häufigkeit des Augenblinzelns der Personen im Video erkennen, so etwa von Li, Chang et al., 2018.

Einen generalisierten Ansatz verfolgen Arbeiten, die ihrerseits – ebenso wie der Deepfake-Algoritmus – maschinelles Lernen einsetzen, so etwa von Rössler, Cozzolino et al. (2018), Afchar, Nozcik et al. (2018) und Chollet (2016).

III. Bot-Erkennung

Bei der Bot-Erkennung geht es um die Frage, ob die Aktivitäten von Nutzerprofilen einer Online-Plattform von Menschen stammen oder ob sie programmgesteuert sind. Wenn hinter den Interaktionen von Profilen ein Computerprogramm steckt (wenn auch nur zeitweise), handelt es sich um ein (halb-)automatisches Profil bzw. um einen (Social) Bot. Der transparente Einsatz von Bots kann viele positive Effekte haben, beispielsweise hat er ein Rationalisierungspotenzial für Anbieter und auf Kundenseite kann er zu einem Gewinn an Gebrauchstauglichkeit führen.

In Zusammenhang mit Desinformation werden Bots jedoch in intransparenter Weise zur Verstärkung der Verbreitung von Fake News eingesetzt, hier kommen „Malicious Social Bots“ zum Einsatz. Das an „one man, one vote“ angelehnte Prinzip „one man, one voice“ wird durch diese Bots gebrochen: Das Ziel ihres Einsatzes ist die Manipulation von Meinungen mittels Fake News, die scheinbar von einer Masse von Menschen für wahr gehalten und verbreitet wird.

Zur Bot-Erkennung werden verschiedene Verfahren erforscht. Sie lassen sich grob bezüglich ihrer jeweiligen Methodik (s. Karataş & Şahin, 2017) in drei Kategorien einteilen (wobei zur Detektion vielfach Kombinationsverfahren hiervon eingesetzt werden):

– Bot-Erkennung mittels Methoden der Strukturerkennung,
– Bot-Erkennung durch Crowdsourcing und
– Bot-Erkennung mit Methoden des maschinellen Lernens.

Bacciu et al. (2019) erzielen auf Twitter eine Bot-Erkennung von ca. 95 % bei englischsprachigen Texten.

Es ist offensichtlich, dass die Detektionsergebnisse durch die Betreiber wesentlich besser sind, als diejenigen, die von außerhalb des Netzes erzielt werden können. Die Betreiber haben – im Gegensatz zu dritten Parteien – globalen Vollzugriff mit beliebigen, eigenen Programmierschnittstellen (API = application programming interface) auf die Primärdaten der Profile, sie können wie keine andere Organisation Big-Data-Analysen (unter Wahrung des Datenschutzes) hierauf durchführen und in Echtzeit im Direktzugriff bei aktuell aktiven Profilen die Provenance der Aktivitäten analysieren.

Aufgrund der besonderen Bedeutung von Twitter für Nachrichtenmedien, der vielfältigen Möglichkeiten des Einsatzes von APIs bei Twitter und weil es bei Twitter keine Echtnamen-Pflicht gibt, werden im Folgenden Analyseergebnisse auf Twitter mittels maschinellen Lernens zusammengefasst.

Gilani et al. (2017) analysierten Verhaltensmerkmale von Bots und Menschen in Twitter-Daten, wie z. B. Vorlieben, Retweets, Antworten und @-Mentions, Aktivität oder die Menge der hochgeladenen Inhalte. Sie zählten auch die gemeinsamen URLs und das Verhältnis von Followern für jedes Twitter-Benutzerkonto. Für die Datenerfassung, Vorverarbeitung, Annotation und Analyse wurde das als Stweeler (Gilani et al., 2016) bekannte Framework verwendet.

Die Autoren von Gilani et al. (2017) beobachteten, dass Menschen neuere Inhalte generieren, während Bots auf Retweeting angewiesen sind. Außerdem haben Bots eine höhere Neigung, URLs zu teilen und Medien (wie Bilder und Videos) häufiger hochzuladen als Menschen. Varol et al. (2017) stellten ein Twitter-Bot-Erkennungsframework vor, das mehr als tausend Features aus sechs verschiedenen Klassen von Twitter-Benutzerdaten und Metadaten extrahiert. Extrahierte Features sind z. B. die Anzahl der Freunde des Twitter-Nutzers, der getwitterte Inhalt, die Stimmung im Text oder seine Aktivitätszeitreihe. Die extrahierten Merkmale wurden dann verwendet, um verschiedene Modelle zur Boterkennung zu trainieren. Durch ein 5-faches Kreuzvalidierungsverfahren erreichte der trainierte Random Forest Klassifikator eine Genauigkeit von 0,95 AUC.

B. Überblick über die eigene Forschung

Im Projekt DORIAN wurden zahlreiche Methoden untersucht, wie Desinformationen erkannt werden können. Dabei war die Absicht nicht, fertige

Werkzeuge zu entwickeln, mit denen bereits automatisiert Inhalte beurteilt werden können. Es galt vielmehr, die vielfältigen Möglichkeiten dieser Erkennung zu sondieren und sie im Rahmen erster Testimplementierungen auf ihre Eignung hin zu untersuchen. Dabei wurde auch ein Augenmerk auf die Gestaltung einer passenden Umgebung gelegt, die einen effizienten Zugriff auf Interneträume durch Crawler ermöglicht und dabei die Prinzipien des Privacy-by-Design (Cavoukian, 2009) beachtet.

I. Datenerfassung mittels Crawling-Technologie

Für die automatisierte Erfassung von Informationen aus verschiedenen Interneträumen wurde im Projekt DORIAN ein Framework konzipiert, bei dem der Einsatz von Crawling-Technologie im Fokus steht. Für einzelne Module des Frameworks wurden Testimplementierungen durchgeführt und evaluiert.

1. Definitionen

Crawler

Als „Webcrawler“ oder kurz „Crawler“ werden Programme bezeichnet, die sich automatisiert meist über Verlinkungen (URLs) zwischen Webseiten durch das Internet bewegen, um die auf den verschiedenen Webseiten vorhandenen Inhalte zu erfassen. Ein Crawler stellt dabei grundsätzlich zwei Mechanismen zur Verfügung: Einen Mechanismus zur Fortbewegung und Datenerfassung im Internet, also das Absetzen von Anfragen über das Hypertext Transfer Protocol (HTTP) und das Herunterladen der entsprechenden HTTP-Antworten, sowie einen Mechanismus zur Analyse der erfassten Inhalte, beispielsweise um die URL für die nächste Webseite zu ermitteln. Die Inhalte, die ein Crawler erfassen/herunterladen kann, sind dabei nicht nur auf HTML-Webseiten beschränkt, sondern schließen jegliche Dateitypen (bspw. Dokumente, Bilder, Videos, …) ein, solange diese über HTTP übertragen werden.

Scraper

Diejenige Teilkomponente eines Crawlers, die die Analyse der erfassten Inhalte übernimmt, wird nachfolgend „Webscraper“ oder kurz „Scraper“ bezeichnet. Scrapers sind zentrale Bestandteile eines Crawlers: Sie ermöglichen eine Selektion derjenigen Inhalte oder Teilbereiche von Webseiten, die für die jeweilige Datenerfassung relevant sind. Entsprechend kann hier bereits eine Filterung oder Säuberung größerer Datenmengen stattfinden, bspw. wenn eine Vielzahl von Webseiten erfasst, auf jeder Webseite aber lediglich die enthaltenen URLs gesammelt und alle anderen Inhalte ignoriert werden sollen. Scrapers analysieren und strukturieren dazu die erfassten HTML-Webseiten, um mittels geeigneter Abfragesprachen wie XPath die Selektion relevanter Inhaltsteile zu ermöglichen. Hier wird deutlich, dass ein Scraper stets webseitenspezifisch ist: Zwei im Aufbau grundsätzlich verschiedene Webseiten benötigen jeweils einen eigenen Scraper, da sich ihre Struktur und somit ebenfalls die Selektion der relevanten Inhalte unterscheidet. Entsprechend hoch ist der Aufwand der Entwicklung und Wartung eines Crawlers, der eine Vielzahl verschiedener Webseiten automatisiert bearbeiten können soll.

Dynamische Inhalte

Typischerweise verarbeitet ein einfacher Crawler mittels seiner integrierten Scrapers Webseiten in ihrer „rohen“ Form: Der Crawler lädt die Webseite als HTML herunter und übergibt diese seinem zuständigen Scraper zur Analyse und Strukturierung. Hierbei werden typischerweise eingebundene Inhalte wie bspw. Bilder, Videos, Stylesheets oder JavaScript-Dateien nicht weiter betrachtet. Dies hat zur Folge, dass insbesondere über JavaScript nachgeladene dynamische Inhalte nicht vom Scraper analysiert werden können – der Scraper verarbeitet entsprechend nur unvollständige Daten und liefert im schlimmsten Fall falsche Ergebnisse, bspw. in Form einer leeren Webseitenstruktur.

Um auch dynamische Inhalte in korrekter Form erfassen und verarbeiten zu können, ist es notwendig, Scraper mittels zusätzlicher Mechanismen zu erweitern. Hierbei bieten sich Test-Frameworks wie Selenium an: Dieses Framework ist eigentlich für Softwaretests von Webanwendungen entwickelt worden und ermöglicht quasi die automatisierte Steuerung eines vollständigen Webbrowsers. Hiermit wird sichergestellt, dass dynamische Inhalte nachgeladen und ggf. notwendige Interaktionen auf der Webseite (bspw.

Button-Klicks, Scrolling, etc.) durchgeführt werden können, um die Inhalte der Webseite vollständig zu erfassen. Die Integration von Selenium in eigene Scrapers ist für eine Vielzahl von Programmiersprachen möglich und einfach umzusetzen, hat jedoch zum Nachteil, dass die jeweiligen Scrapers mehr Ressourcen benötigen und entsprechend schwerfälliger ihre Prozesse abarbeiten.

Application Programming Interfaces (APIs)

Eine Application Programming Interface (API) im Web-Bereich stellt eine Schnittstelle dar, die von einem Inhaltsanbieter externen Entwicklern zur Verfügung gestellt wird, damit diese über (ebenfalls externe) Programme effektiv und effizient vordefinierte Prozesse auf den Inhalten des Anbieters ausführen können, bspw. Suchen im Datenbestand oder die Steuerung konkreter Verarbeitungsschritte. APIs sind typischerweise zugriffsgeschützt und benötigen einen entsprechenden Account beim Anbieter, dem die Nutzung der API gestattet ist. Zu seiner API liefert der Anbieter typischerweise eine umfangreiche Spezifikation und Dokumentation und legt außerdem die Kriterien für die Nutzung seiner API fest, bspw. Limitierungen der jeweiligen Abfragen, Datennutzungsbestimmungen oder Anforderungen für den Erhalt eines Zugangs zur API.

Aus Sicht eines Entwicklers ist der Zugriff auf die API eines Anbieters relevanter Inhalte stets zu bevorzugen, da typischerweise der Programmieraufwand durch die existierende API-Spezifikation deutlich reduziert wird. Jedoch können die Anforderungen eines Anbieters für die Nutzung seiner API durchaus so hoch sein, dass eine API-Nutzung aus technischer oder auch finanzieller Sicht impraktikabel wird.

2. Crawling-Framework für größerer Interneträume

Im Rahmen der Projektarbeit wurde ein Konzept für ein Framework entworfen, das in der Lage ist, in größeren Interneträumen Daten zu erfassen, potenziell relevante Inhalte zu selektieren und abzuspeichern. Die so erfassten Daten sollen eine Grundlage für nachfolgende Analysen hinsichtlich Desinformationen darstellen. Als Interneträume wurden exemplarisch drei große Plattformen (YouTube, Twitter und Facebook) betrachtet, auf denen Nutzende zahlreiche eigene Inhalte veröffentlichen und die Inhalte anderer

kommentieren können. Zur Datenerfassung wurden Testimplementierungen durchgeführt und evaluiert.

Das Crawling-Framework ist modular konzipiert, wobei verschiedene Scraping-Module für jeweils verschiedene Scraping-Tasks eingesetzt werden. Für die Speicherung der erfassten Daten wird ein relationales Datenbanksystem verwendet. Als relevante Inhalte werden auf den drei Plattformen jegliche Textinhalte betrachtet, die von Nutzenden erstellt worden waren. Das entspricht auf YouTube den Kommentaren unter den jeweiligen Videos, bei Twitter den Tweets und bei Facebook den Posts und Kommentaren der Nutzerinnen und Nutzer. Alle drei Pattformen unterscheiden sich in ihrer Struktur und anhand ihrer Inhalte signifikant, sodass jeweils ein Scraping-Modul je Plattform vorgesehen ist.

Nachfolgend werden die drei Plattformen einzeln betrachtet und die Möglichkeiten und Herausforderungen einer automatisierten Datenerfassung aufgezeigt.

YouTube-Modul

YouTube bietet über die YouTube-Data-API effizienten Zugriff auf Kommentare samt umfangreicher Metadaten zu Kommentaren, Videos und Nutzenden. Für einen API-Zugang wird lediglich ein Google-Account benötigt. Zur Datenerfassung auf YouTube eignet sich somit ein einfacher Scraper, der Anfragen an die API stellen kann. Zusätzliche Funktionalität für die Verarbeitung dynamischer Inhalte wird entsprechend nicht benötigt.

Es existieren zwar Limitierungen für die Nutzung der API, diese sind jedoch relativ großzügig bemessen: Pro Tag besitzt jeder API-Nutzende ein Kontingent von 10.000 Einheiten, wobei einzelne Abfragen zwischen 2-5 Einheiten verbrauchen.

Zusammenfassung

Relevante Daten auf YouTube lassen sich somit sehr einfach erfassen, eine API-Zugriffsberechtigung lässt sich schnell erlangen und der Implementierungsaufwand eines Scrapers ist gering.

Facebook-Modul

Wie YouTube und Twitter stellt auch Facebook seine Graph API für den Datenzugriff zur Verfügung, für den ein Facebook-Account mit entsprechender API-Berechtigung benötigt wird. Die Graph API von Facebook bietet zwar ebenfalls umfassenden Zugriff auf Nutzer- und Metadaten, jedoch werden selbst nach erteilter API-Zugriffsberechtigung weitere Berechtigungen von Facebook eingefordert, die die Nutzung der API quasi gänzlich impraktikabel machen: Facebook fordert, dass Nutzende der API, die Daten über Facebook-Nutzende erfassen möchten, von jedem dieser Facebook-Nutzende eine Einwilligung über diese Datenerfassung einholt. Dies kann automatisiert über eine Abfrage auf Seite des Facebook-Nutzenden geschehen und wird bspw. bei der Installation von Apps für Smartphones eingesetzt, die eine Facebook-Integration anbieten. Willigt der Facebook-Nutzende ein, so kann die entsprechende App bzw. das Unternehmen, das die App zur Verfügung stellt, die Daten des Facebook-Nutzenden über die Graph API erfassen. Ohne Einwilligung ist dies jedoch nicht möglich. Somit ist die automatisierte Erfassung von relevanten Daten auf Facebook nicht über die Graph API realisierbar, solange kein Weg gefunden wird, von sämtlichen Facebook-Nutzenden, die relevante Inhalte produzieren, eine Einwilligung zu erhalten.

Entsprechend wird für die Entwicklung eines Scrapers für Facebook die Betrachtung der Facebook-Webseite notwendig. Ein einfacher Scraper stößt hier jedoch schnell an seine Grenzen: Da von Facebook auf seiner Webseite sehr viele dynamische Inhalte eingesetzt werden (insbesondere zum Nachladen von Inhalten durch Paging, Scrolling), ist der zusätzliche Einsatz von Selenium notwendig. Hiermit lassen sich die Struktur der Facebook-Webseite analysieren und einzelne Elemente gezielt ansteuern. Da mit Selenium quasi ein vollständiger Webbrowser automatisiert gesteuert wird, leidet die Leistungsfähigkeit eines Scrapers, der Selenium einsetzt, sehr stark im Vergleich zu einem Scraper, der lediglich einer API-Spezifikation folgen muss und sehr effizient entsprechende Abfragen stellen kann.

Zusammenfassung

Relevante Daten auf Facebook sind über die Graph API quasi nicht erfassbar, da Facebook hier sehr einschränkende Bedingungen stellt. Scraper, die die Graph API umgehen und stattdessen über die Webseite von Facebook Daten

erfassen wollen, müssen zwangsläufig Frameworks wie Selenium einsetzen, wodurch der Scraper vergleichsweise nur sehr langsam Daten erfassen kann. Der Implementierungsaufwand ist verhältnismäßig hoch, da eine Vielzahl von Elementen der Facebook-Webseite angesteuert werden müssen, um die dynamischen Inhalte nachladen zu können.

3. Evaluation von Testimplementierungen

Im Rahmen des Projekts wurden erste Testimplementierungen für jedes Scraping-Modul der drei großen Plattformen nach dem oben beschriebenen Konzept durchgeführt und evaluiert. Hierbei konnte die Funktionsfähigkeit jedes Moduls bestätigt werden, wobei sich teilweise erhebliche Leistungsunterschiede hinsichtlich der Scraping-Dauer zeigten. Während pro Sekunde auf YouTube fast 90 Kommentare und auf Twitter bis zu 1600 vollständige Tweets erfasst werden konnten, dauerte die Erfassung eines einzelnen Kommentars auf Facebook bis zu 3 Sekunden. Hier zeigte sich deutlich der Overhead eines Scrapers, der (im Gegensatz zu Scrapers, die auf eine API zugreifen können) einen vollständigen Webbrowser steuern und umständlich auf Webseiten mit dynamischen Inhalten navigieren muss, um an die gewünschten Inhalte zu gelangen.

II. Datenschutzrechtliche Aspekte der automatischen Erkennung von Desinformationen für wissenschaftliche Forschungszwecke

Im Rahmen der automatischen Erkennung von Desinformationen werden öffentlich gemachte Artikel und Meldungen in Online-Nachrichtenplattformen sowie Beiträge in Social Networks erhoben und analysiert. Die verarbeiteten Daten enthalten in der Regel personenbezogene Daten[1] (z. B. Name des Autors, Inhalte, Abbildungen, Zitate einer natürlichen Person), bei deren Verarbeitung die Anforderungen der Datenschutz-Grundverordnung (nach-

1 Personenbezogene Daten sind diejenigen Informationen, die eine natürliche Person („betroffene Person“) direkt oder indirekt identifizieren oder identifizierbar machen (Art. 4 Nr. 1 DSGVO). Z. B. Vor- und Nachname, die Telefon- und Kreditkartennummer sowie die Hobbies und Interessen einer natürlichen Person. Auch pseudonymisierte Daten eröffnen den Anwendungsbereich der Regeln des Datenschutzrechts. Auf anonymisierte Daten, bei denen eine Identifizierung natürlicher Personen nicht (mehr) möglich ist, finden die Anforderungen des Datenschutzrechts jedoch keine Anwendung.

folgend DSGVO) einzuhalten sind. Nachfolgend werden die wichtigsten Anforderungen der DSGVO näher erläutert, die bei der automatischen Erkennung von Desinformationen für Forschungszwecke zu beachten sind.

1. Zulässigkeit der Verarbeitung

Die automatische Sammlung und Auswertung personenbezogener Daten verstößt grundsätzlich gegen das Grundprinzip der DSGVO, wonach die Verarbeitung personenbezogener Daten auf das notwendige Maß zu beschränken ist (Art. 5 Abs. 1 lit. c, Datenminimierung). Daher kann sie nur in Ausnahmefällen zulässig sein.[2]

Eine solche Ausnahme liegt vor, wenn die Verarbeitung personenbezogener Daten für wissenschaftliche Forschungszwecke erforderlich ist (Art. 89 Abs. 1 DSGVO). Unter wissenschaftliche Forschung ist die technologische Entwicklung sowie die Grundlagenforschung, die angewandte Forschung und die privat finanzierte Forschung zu verstehen.[3] Die Verarbeitung für Forschungszwecke wird in der Datenschutz-Grundverordnung sowie in sonstigen anwendbaren Datenschutzregelungen privilegiert[4] (z. B. hinsichtlich der Zweckbindung und Speicherbegrenzung sowie hinsichtlich der Rechte der betroffenen Personen). Grundvoraussetzung ist, dass die forschende Stelle zusätzliche Garantien zum Schutz der Rechte der Freiheiten der betroffenen Personen vorsieht. Mit technischen und organisatorischen Maßnahmen soll sichergestellt werden, dass bei der Durchführung der Verarbeitungstätigkeiten insbesondere das Prinzip der Datenminimierung eingehalten wird. Die Maßnahmen sind bereits bei der Entwicklung der Mechanismen, d. h. vor der tatsächlichen Verarbeitungstätigkeit zur Erkennung von Desinformationen zu treffen (Art. 25 Abs. 1). Zu den Maßnahmen kann die Pseudonymisierung und die Anonymisierung gehören (Art. 89 Abs. 1 Satz 3). Insofern hat die forschende Stelle zunächst zu prüfen, ob der Forschungszweck auch mit anonymisierten Daten zu erreichen ist. Im Rahmen des DORIAN-Projekts hat sich jedoch gezeigt, dass die Verarbeitung mit anonymisierten Daten für die automatische Erkennung von Desinformationen ungeeignet ist. Die

2 Hoeren, Skript Internetrecht, Stand November 2018, S. 492; Schulz, in: Gola, Datenschutz-Grundverordnung 2018, Art. 6 Rn. 257-258.

3 Buchner/Tinnefeld, in: Kühling/Buchner, Datenschutz-Grundverordnung, Bundesdatenschutzgesetz: DS-GVO/BDSG 2017, Art. 89 Rn. 13.

4 Z. B. Datenschutzgesetze der Länder, Bundesdatenschutzgesetz, Landeskrankenhausgesetze, Meldegesetze.

Verarbeitung personenbezogener Daten ist erforderlich, um u.a. Verbreiter von Desinformationen zu identifizieren sowie Korrelationen zu erkennen. Hier sollte anstelle der Anonymisierung eine Pseudonymisierung der Daten erfolgen. Die pseudonymisierten Daten sollten darüber hinaus verschlüsselt gespeichert und übertragen werden. Ist die Verarbeitung personenbezogener Daten zu einem späteren Zeitpunkt des Projekts nicht mehr erforderlich, sind die Daten unverzüglich zu löschen bzw. zu anonymisieren. Die Weiterverarbeitung der Daten für andere Forschungszwecke sollte ebenfalls ohne Personenbezug erfolgen (Art. 89 Abs. 1 Satz 4).

2. Dokumentationspflichten

Darüber hinaus hat die forschende Stelle ihren Dokumentations- und Nachweispflichten nachzukommen (Art. 5 Abs. 2 DSGVO). Neben einer Verfahrensbeschreibung (Art. 30 Abs. 1 DSGVO) hat die forschende Stelle Störungen in der Sicherheit der Verarbeitung bei der zuständigen Behörde zu melden (Art. 33 Abs. 1 DSGVO) und in den Fällen, in denen ein hohes Risiko für die Rechte und Freiheiten der betroffenen Personen vorliegt, eine Datenschutzfolgenabschätzung durchzuführen (Art. 35 DSGVO).[5] Hier werden die möglichen Risiken analysiert und bewertet und folglich die effektive Gegenmaßnahmen für die bestehenden Risiken ermittelt und implementiert.

3. Informationspflicht und Rechte der betroffenen Personen

Zur Gewährleistung der Transparenz der Verarbeitung regelt die DSGVO Informationspflichten des Verantwortlichen (Art. 13 und 14) sowie Rechte der betroffenen Personen (z. B. Recht auf Auskunft oder Löschung, siehe Art. 15 ff.). Für die Forschung sieht die DSGVO eine Ausnahme der Informationspflicht vor, wenn die Erteilung dieser Information sich als unmöglich erweist oder einen unverhältnismäßigen Aufwand erfordern würde (Art. 14 Abs. 5 lit. b). Bei der automatischen Erhebung und Auswertung personenbezogener Daten für Forschungszwecke wird in der Regel diese Ausnahme

5 Hierzu: Leitlinien zur Datenschutz-Folgenabschätzung (DSFA) und Beantwortung der Frage, ob eine Verarbeitung im Sinne der Verordnung 2016/679 „wahrscheinlich ein hohes Risiko mit sich bringt“, Datenschutzgruppe nach Art. 29, 4.10.2017, 17/DE WP 248 Rev. 01.

Anwendung finden. Darüber hinaus regelt die DSGVO eine Einschränkung des Löschungsrechts der betroffenen Personen, wenn die Verwirklichung des Rechts die Verarbeitung für wissenschaftliche Forschungszwecke unmöglich macht oder ernsthaft beeinträchtigt (Art. 17 Abs. 3 lit. d). Hier ist in jedem Einzelfall zu prüfen, ob die Voraussetzungen vorliegen. Weitere Einschränkungen der Rechte sowie Ausnahmen der Informationspflichten werden in nationalen Gesetzen vorgesehen (z. B. § 27 Abs. 2 BDSG, § 24 Abs. 2 HDSIG.).

4. Verarbeitung besonderer Kategorien von Daten

Bei der automatischen Erkennung von Desinformationen ist eine Verarbeitung von besonderen Kategorien von Daten nicht auszuschließen. Gegenstand der Analyse können beispielsweise Inhalte sein, die Informationen über die politische oder religiöse Überzeugung einer natürlichen Person enthalten. Gemäß Art. 9 DSGVO ist die Verarbeitung dieser Art Daten grundsätzlich verboten. Diese Daten sind als besondere Kategorien von personenbezogenen Daten[6] einzustufen, die besonders schützenswert sind. Soweit die betroffene Person, die sie betreffende besondere Kategorien von Daten offensichtlich öffentlich gemacht hat, besteht allerdings keine besondere Schutzbedürftigkeit mehr und eine Verarbeitung diese Daten kann beim Vorliegen eines Erlaubnistatbestandes gemäß Art. 6 Abs. 1 erfolgen.[7] Die Verarbeitung besondere Kategorien von Daten für Forschungszwecke ist auch ohne Einwilligung der betroffenen Personen zulässig, wenn die Verarbeitung für die Erfüllung des Forschungszweckes erforderlich ist und zusätzliche Maßnahmen zum Schutz der Rechte und Freiheiten der betroffenen Personen getroffen wurden. (u.a. § 27 Abs. 1 BDSG i. V. m. Art. 9 Abs. Abs. 2 lit. j). Grundsätzlich sollten aber besondere Kategorien personenbezogener Daten nie unverschlüsselt gespeichert und übertragen werden sowie vor unberechtigtem Zugriff geschützt werden.

6 Gemäß Art. 9 Abs. 1 DSGVO sind besondere Kategorien von Daten solche, aus denen die rassische und ethnische Herkunft, politische Meinungen, religiöse oder weltanschauliche Überzeugungen oder die Gewerkschaftszugehörigkeit hervorgehen, sowie genetische und biometrische Daten zur eindeutigen Identifizierung einer natürlichen Person, Gesundheitsdaten und Daten zum Sexualleben oder der sexuellen Orientierung.

7 Schulz, in: Gola Datenschutz-Grundverordnung 2018, Art. 9, Rn. 25 - 26.

5. Zusammenfassung

Die Verarbeitung personenbezogener Daten einschließlich besondere Kategorien von Daten zur Erkennung von Desinformationen für Forschungszwecke ist grundsätzlich zulässig, wenn die Rechte und Freiheiten der betroffenen Personen durch technische und organisatorische Maßnahmen (insb. Pseudonymisierung, Anonymisierung, Verschlüsselung, Zugriffskontrolle und Protokollierung) gewahrt werden. Demnach ist eine wichtige Aufgabe der forschenden Stelle, die Risiken der Verarbeitung für die Rechte der betroffenen Personen rechtzeitig zu erkennen und mit den erforderlichen Maßnahmen zu adressieren. Ist die Verarbeitung personenbezogener Daten zu einem späteren Zeitpunkt des Projekts nicht mehr erforderlich, sind die Daten zu anonymisieren bzw. zu löschen. Auch die Weiterverarbeitung der Daten für andere Forschungszwecke sollte in anonymisierter Form erfolgen. Neben den Regelungen der DSGVO hat die forschende Stelle ebenfalls das auf sie anwendbare nationale Datenschutzgesetz (z. B. das Datenschutzgesetz des Landes oder das Bundesdatenschutzgesetz) ergänzend zu beachten. Hier werden u.a. konkrete Maßnahmen zum Schutz der betroffenen Personen geregelt, die die forschende Stelle zusätzlich zu implementieren hat (siehe z. B. § 22 Abs. 2 i. V. m. § 27 BDSG).

III. Erkennung Bildmanipulation

Wenn Bilder aus unbekannten oder unseriösen Quellen auf Echtheit überprüft werden sollen, reicht eine visuelle Überprüfung durch einen Menschen heute nicht mehr aus. Ein Bild kann genau dann als nicht mehr echt angesehen werden, wenn es nachträglich bearbeitet worden ist.

1. Erkennung der Wiederverwendung

Die einfachste Art, mit Bildern Desinformationen zu betreiben, ist es, diese einfach aus ihrem Kontext heraus neu zu verwenden. Ein bekanntes Beispiel für diese Vorgehensweise sind Bilder aus Kriegsgebieten. Angeblich wird hier häufig auf Archivmaterial zurückgegriffen, da ein schnelles Beschaffen aktueller Bilder nicht oder nur unter großer Gefahr möglich ist.

Diese Wiederverwendung kann erkannt werden, wenn die Bilder über eine inverse Bildersuche in bekannten Portalen wie Google Image Search oder

TinEye gefunden werden und die Treffer in der Vergangenheit liegen. Diese Suchmaschinen sind allerdings nicht resistent gegen bewusste Verschleierung. So kann es reichen, Ausschnitte aus Bildern zu erzeugen oder diese zu spiegeln.

Sollen Bilder auch nach entsprechenden Verschleierungsschritten noch erkannt werden, sollten robuste Methoden zur Bilderkennung eingesetzt werden. Dazu gehören zum einen sogenannte „robuste Hashverfahren", im Englischen auch als image fingerprints oder perceptual hashes bezeichnet. Sie haben gemeinsam, dass sie eine effiziente Wiedererkennung von Bildern ermöglichen und resistent gegen verschiedene Veränderungen am Bild sind. Ein einfaches Beispiel ist der Blockhash aus Steinebach (2012). Hier wird ein Bild in Graustufen umgerechnet und auf 16 x 16 Pixel herunter skaliert. Aus den Pixeln wird anhand des Abstands vom Medien der Helligkeit der Pixel ein Vektor aus 256 Bit errechnet, der den robusten Hash des Bildes darstellt. Dieser Hash ist robust gegen Skalierung, leichte Bildmanipulationen und verlustbehaftete Kompression.

Ein Alternative zu den Hashverfahren sind Merkmalsvektoren wie SIFT. Sie erkennen Bilder über eine Repräsentation prägnanter Stellen wieder und sind insbesondere resistent gegen Verzerrung und Beschneiden von Bildern.

2. Erkennung von Veränderungen

Viele bestehende Algorithmen verwenden unkomprimierte Bilder für Netzwerktraining und Testing. In der Praxis ist es jedoch selten, dass unkomprimierte Bilder zugänglich sind, insbesondere für forensische Untersuchungen. In der Regel sind nur JPEG-Bilder als Original- oder manipulierte Bilder verfügbar. Um die Erkennungsleistung für JPEG-Bilder zu verbessern, haben wir ein praktischeres Szenario entwickelt, bei dem sowohl die Trainingsbilder als auch die Testbilder im JPEG-Format gespeichert werden. Wir haben die möglichen Gründe analysiert, warum das CNN-Netzwerk in [3] bei JPEG-Bildern im praktischen Szenario schlecht abschneidet. Basierend auf der Analyse haben wir ein neues Fusionsnetzwerk entwickelt, um die Erkennungsleistung bei JPEG-Bildern zu verbessern. Das Fusionsnetzwerk ist ein Verbund aus einem Inception-ResNet basierten Netzwerk und einem DCT-basierten Netzwerk, das die Stärken der beiden Netzwerke kombiniert und ihre Schwächen ausgleicht.

Das neue Fusionsnetzwerk wird zunächst im Basistest auf seine Leistungsfähigkeit bei der Identifizierung neuer Bilder mit den gleichen JPEG-

Tabelle 4.1: Ergebnisse des Basistests

	Netzwerk in [3]	**Fusionsnetzwerk**	**Δ Netzwerk in [3]-Fusion**
F1 Score	0,6958	0,9001	29,35 %
Precision	0,6916	0,9005	30,22 %
Recall	0,7001	0,8996	28,49 %

Tabelle 4.2: Ergebnisse des generalisierten Tests

	Netzwerk in [3]	**Fusionsnetzwerk**	**Δ Netzwerk in [3]-Fusion**
F1 Score	0,6973	0,8544	22,53 %
Precision	0,7022	0,8579	23,89 %
Recall	0,6925	0,8509	21,18 %

Qualitätsfaktoren wie in der Trainingsphase evaluiert. Wie in Tabelle 4.1 dargestellt, verbessert sich das Fusionsnetzwerk um etwa 30 % und die Genauigkeit erreicht 89,96 %. Weiterhin wird im generalisierten Test die Generalisierbarkeit auf neue Bilder mit unterschiedlichen Qualitäten bewertet. Die Leistung zur Erkennung von Manipulationen von Bildern mit unbekannten JPEG-Qualitätsfaktoren ist in Tabelle 4.2 dargestellt. Die Genauigkeit beträgt 85,09 % und eine Verbesserung von ca. 22 % wird erreicht. Darüber hinaus zeigt Tabelle 4.3 die Verbesserung der Erkennung für jede Art der Manipulation.

3. Erkennung von Montagen

Eine Art der digitalen Manipulation ist die Fotomontage. Eine Fotomontage kann definiert werden als die Erstellung eines Ausgangsbilds, dass mindestens Bildinhalte aus zwei verschiedenen Eingangsbildern enthält. Auch wenn die kopierten Bildinhalte prinzipiell beliebig sein können, sind für diese Aufgabenstellung eher die Bildinhalte wichtig, die sich für Desinformationen eignen. Relevante Bildinhalte sind deswegen vor allem Menschen und größere Objekte.

Der in der Praxis relevanteste Fall einer Fotomontage ist daher jener, bei dem ein relevanter Bildinhalt wie zum Beispiel eine Person in einem

Tabelle 4.3: Verbesserung der Ergebnisse für jede Art der Manipulation

	Δ F1 Score Netzwerk in [3]-Fusion	**Δ Precision Netzwerk in [3]-Fusion**	**Δ Recall Netzwerk in [3]-Fusion**
Double JPEG	125,07 %	77,05 %	168,38 %
Gaussian Blurring	8,58 %	7,99 %	9,17 %
Median Filtering	4,35 %	3,32 %	5,45 %
Resampling	17,32 %	25,59 %	8,80 %
Gaussian Noise	55,13 %	67,91 %	42,32 %

Eingangsbild in ein zweites Eingangsbild kopiert wird. So kann ein Ausgangsbild erzeugt werden, welches zum Beispiel die Person fälschlicherweise in einer Situation darstellt, die die Aussage einer ebenfalls dazu erfundenen Falschnachricht untermauert, um die Person zu diffamieren. Es ist aber auch genau das Gegenteil möglich, um zum Beispiel eine Person in einer Situation wichtig erscheinen zu lassen. Eine Fotomontage muss sich jedoch nicht nur auf das Kopieren von Fremdbildinhalten beschränken. So kann der Ersteller der Montage zusätzlich noch bestimmte Bildmanipulationen durchführen, um so das Ausgangsbild authentischer wirken zu lassen oder aber auch, um einer automatisierten Erkennung zu entgehen.

Abbildung 4.2: Ein Beispiel für eine Montage. Links findet sich die Fälschung, in der Putin auf den leeren Stuhl im Bild auf der rechten Seite einkopiert wurde. Quellen: Links Twitter, Rechts Getty Images.

Für die Erkennung von Kollagen wurde ein Ansatz gewählt, der Bilder anhand von Merkmalen vergleicht. Da Merkmale über den gesamten Bildinhalt hinweg erkannt werden können, ist es durch einen merkmalsbasierten Ansatz relativ einfach, die Ähnlichkeit ganzer Bilder oder auch einzelner Bildteile miteinander zu vergleichen.

Konzept

Die Montageerkennung selbst ist in zwei Hauptkomponenten unterteilt: Die erste Hauptkomponente ist die Initialisierung, bei der eine Bilddatenbank einmalig zu einem durchsuchbaren Index verarbeitet wird. Sie stellt also die Datenbasis dar, auf der später eine Suche durchgeführt werden kann. Im Umfeld von Desinformationen könnte dies ein Archiv von Aufnahmen sein, die bereits in der Presse verwendet wurden. Neue Bilder werden hier kontinuierlich nachgepflegt.

Zuerst wird hierzu eine Datenbank für die Bildkennungen (beispielsweise Dateinamen und Dateipfad in einem Archiv) und ihre Merkmale angelegt. Für die Generierung der Merkmale verwenden wir die Verfahren SIFT und SURF. Mit einem Detektor für diese Merkmale werden die Schlüsselpunkte für jedes Bild erstellt. Anschließend erfolgt eine Filterung, die die Anzahl der zu verwendenden Schlüsselpunkte reduziert. Anschließend werden die Deskriptoren der einzelnen Schlüsselpunkte ermittelt. Deskriptoren entstehen aus der Verarbeitung der Merkmale und stellen Vektoren dar, die die Merkmale in einer normalisierten und somit robust vergleichbaren Form speichern. Aus den Deskriptoren wird dann mit einer ausgewählten Indizierungsmethode ein Index erstellt.

Die zweite Hauptkomponente ist die Abfrage, bei der ein Eingangsbild verarbeitet und mit dem Index verglichen wird. Das Ergebnis der Verarbeitung kann dabei sowohl für die Abfrage verwendet werden, aber auch für das Einfügen des Bildes in die Datenbank aus der Indexierung, falls das Bild noch nicht bekannt ist.

Zunächst wird hierzu ein Eingabebild bereitgestellt, für welches eine Merkmalserkennung durchgeführt wird. Wie bei der Initialisierung wird dann die Filterung durchgeführt und die Deskriptoren extrahiert. Anschließend erfolgt der Abgleich der Deskriptoren mit den Deskriptoren im Index über das Indizierungsverfahren. Das Ergebnis ist eine Reihe von Übereinstimmungen, die Deskriptoren einander zuordnen. Eine Übereinstimmung besteht aus einem Deskriptor aus dem Ausgangsbild und dem ähnlichsten Deskriptor aus

einem Bild in der Datenbank. Wenn genügend Übereinstimmungen gefunden wurden, die sich auf das gleiche Bild beziehen, ist es wahrscheinlich, dass ein Objekt erkannt wird. In einem weiteren Schritt wird dies jedoch nochmals überprüft, indem die Merkmale im Detail hinsichtlich ihrer geometrischen Ähnlichkeit miteinander verglichen werden.

Optimierung der Merkmale

Eine Montage besteht aus einem oder mehreren Objekten aus verschiedenen Bildern. Die Erkennung der einzelnen Objekte erfolgt über die Merkmalserkennung. Merkmalsdetektoren reagieren in der Regel besonders auf inhomogene Oberflächen und finden Merkmale auf Bildern ohne homogene Oberflächen über das gesamte Bild. Homogene Oberflächen sind monotone Oberflächen ohne Struktur, wie z. B. ein wolkenloser Himmel oder ein niedrig aufgelöstes Bild einer Straße. Der weit verbreitete Merkmalsdetektor SIFT findet 30.000 - 40.000 Merkmale auf einem 1000 x 1000 Pixel Bild ohne homogene Flächen. Eine so hohe Anzahl von Features ist in unserem Anwendungsfall unnötig und würde dazu führen, dass der Bildindex bis zu einem Punkt wächst, der den Speicherverbrauch inakzeptabel macht.

Deshalb verwenden wir eine Filtermethode, um nur eine kleine Anzahl von Features auszuwählen und den Rest zu verwerfen. Merkmalsdetektoren bewerten die Stärke der gefundenen Merkmale. Mit einem geeigneten Filterverfahren kann sichergestellt werden, dass bei beiden Bildern nur die stärksten Merkmale ausgewählt werden und die passende Quote nicht reduziert wird.

Eine einfache Filterung der Merkmale führt aber schnell zu einem Problem im Zusammenhang mit der Montageerkennung. Wenn nur die stärksten Merkmale erhalten bleiben, bleiben oft keine oder nur noch sehr wenigen Merkmale in Teilen eines Bildes übrig. Bei einer Montage kann es nun vorkommen, dass es keine oder nicht genügende Merkmale auf einem Objekt gibt und dieses Objekt nicht mehr als kopierter Bildinhalt erkannt werden kann. Daher muss die Filterung eine Verteilung der Merkmale über das gesamte Bild gewährleisten. Durch die gleichmäßige Verteilung der Merkmale ist es sehr wahrscheinlich, dass nach der Filterung noch genügend Merkmale auf dem Objekt verbleiben. Im Idealfall reduziert dies die Anzahl der Features, ohne eine verminderte Erkennungsrate zu verursachen.

Die Anzahl der pro Bild zu verwendenden Merkmale ist eine der wichtigsten Parameter. Sie hat einen starken Einfluss auf den Abruf auf der einen Seite und auf den Speicherbedarf auf der anderen Seite. Daher muss hier ein

geeigneter Kompromiss gefunden werden. Um den Speicherverbrauch zu reduzieren, wird bei der Indizierung und Datenbankerstellung eine geringere Anzahl von Merkmalen gespeichert als bei der Suche.

In unserer Implementierung stellte sich heraus, dass 500 Merkmale bei der Indexierung und 2000 Merkmale der Suche den besten Kompromiss darstellen. Dies führt immer noch zu einer hohen Erkennungsrate, aber gleichzeitig auch zu einem akzeptableren Speicherverbrauch.

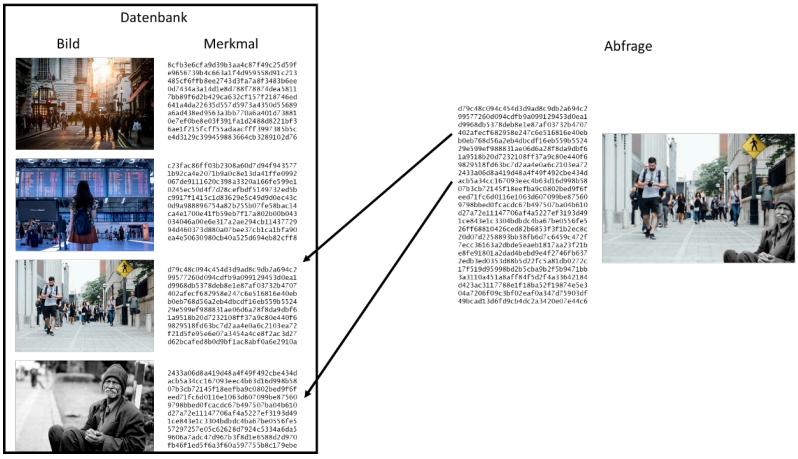

Abbildung 4.3: In der Datenbank werden Bilder und Merkmale gespeichert. Die Abfrage erzeugt aus dem Abfragebild eine Menge von Merkmalen. Diese werden in der Datenbank gesucht. Eine Übereinstimmung zeigt, dass Teile eines Bildes in der Datenbank in dem Bild der Abfrage vorkommen.

Evaluierung

Zur Evaluierung des Ansatzes wurde eine synthetische Erstellung von Kollagen implementiert, die eine ausreichend große Menge an Testmaterial erstellen konnte. Dazu wurden jeweils Objekte aus einer freien Bibliothek in ein Hintergrundbild kopiert. So entstand ein Bild mit dem Objekt, zu dem aber gleichzeitig bereits ein perfekt ausgeschnittenes Objekt bekannt war. Dieses Objekt wurde nun in ein zweites Hintergrundbild kopiert. Dieser Vorgang simuliert die Montage: In der Praxis würde das Objekt aus dem ersten Bild entnommen und in das zweite eingefügt. In der Datenbank sind

nun zwei Bilder gespeichert: Das erste Hintergrundbild mit dem Objekt und das zweite Hintergrundbild. Die Abfrage erfolgt dann mit dem zweiten Hintergrundbild, in welches das Objekt hineinkopiert wurde. Nun muss der Algorithmus in der Lage sein, das Objekt aus dem Abfragebild im ersten Bild der Datenbank zu finden und den Hintergrund des Abfragebildes als das zweiten Hintergrundbild in der Datenbank identifizieren. Um den Vorgang zu erschweren, können Bilder und Objekte verrauscht, gedreht und skaliert werden.

Ergebnisse

Die Evaluierung belegt, dass eine Erkennung von Montage mit der beschriebenen Vorgehensweise sehr zuverlässig erfolgen kann. Die folgende Tabelle zeigt einige ausgewählte Ergebnisse für die Erkennung nach verschiedenen Operationen wie Skalierung, Rotation und Rauschen, jeweils mit unterschiedlicher Stärke. Die Bilder hatten dabei eine Auflösung von 1000 mal 1000 Pixeln. Es wurde durchgehend eine Precision von mindestens 0,99 erreicht, es werden also fast ausschließlich tatsächliche Montagen vom System erkannt.

4. Erkennung von Deepfakes

Für die Erkennung von Deepfake-Angriffen wurde ein in der Fachliteratur bekannter Bildforensik-Ansatz, der sog. JPEG-Ghost-Effekt, für Video weiterentwickelt.

JPEG-Ghost-Effekt

Dieser Effekt wurde ursprünglich für digitale (Einzel-) Bilder vorgestellt (siehe Farid, 2009). Er tritt auf, wenn Bilder, Videos oder Ausschnitte davon mehrfach JPEG-komprimiert werden.

Angenommen, ein gegebenes Bild liegt als JPEG-Datei in der Qualitätsstufe Q1 vor. Letztere wird oftmals auf einer Skala von 0 (starke Kompression, entspricht schlechter Bildqualität, dafür kleiner Dateigröße) bis 100 (entspricht sehr guter Bildqualität) angeben. Von diesem Bild wird jetzt testweise durch eine weitere JPEG-Kompression bei der Qualitätsstufe Q2 eine neue Version erzeugt. Voruntersuchungen der Originalautoren haben gezeigt, dass

Tabelle 4.4: Beispiele Precision und Recall bei Manipulation von Montagen

		Precision		Recall	
		SIFT	**SURF**	**SIFT**	**SURF**
Bild Skalieren	50 %	0,9986	0,9993	0,9567	0,9913
	40 %	0,9993	0,9987	0,8947	0,9893
	30 %	1	0,9993	0,798	0,98
	20 %	0,9988	0,9993	0,546	0,9433
Rotation	10°	0,9993	0,9993	0,9367	0,974
	-10°	0,9972	0,9993	0,958	0,97
	20°	0,9993	0,9993	0,9433	0,9533
	-20°	0,9993	0,9993	0,9453	0,956
	30°	0,9993	0,9993	0,9373	0,9427
	40°	0,9993	0,9986	0,9327	0,9367
	60°	0,9979	1	0,946	0,9287
	90°	0,9951	0,9986	0,956	0,9787
	180°	0,9972	0,9993	0,9527	0,966
Rauschen	Kein	0,9986	0,9993	0,9567	0,9567
	Schwach	0,9986	0,9993	0,9527	0,9527
	Medium	0,9979	1	0,944	0,944
	Stark	0,9965	0,9993	0,9493	0,9493

sich die Farbwerte der Bildpixel dieser beiden Bildversionen am wenigsten unterscheiden, wenn Q2 = Q1 gewählt wird. Im Umkehrschluss werden sich die beiden Versionen Pixel für Pixel immer stärker unterscheiden, je „stärker“ die zweite Kompression ist (also je ausgeprägter Q2 < Q1 ist). Berechnet man also das „Differenzbild“ der beiden Versionen durch pixelweise Subtraktion der Farbwerte, wird dieses daher umso dunkler (entspricht kleinerer Differenz) je ähnlicher Q1 und Q2 sind.

Deepfake-Detektion

Für die Anwendung auf Deepfake-Videos wird dies sinngemäß auf Video-enkodierte Daten übertragen. Auch bei Video-Kompression, etwa im H.264-Codec, sind verschiedene Qualitätsstufen technisch möglich und der Ghost-Effekt kann ebenfalls beobachtet und genutzt werden. Dies wird im vorgestellten Verfahren folgendermaßen angewendet:

1. Input: Ausgangspunkt ist die zu untersuchende Videodatei. Diese kann Bildbereiche mit eingefügten Deepfake-manipulierten Gesichtern enthalten – oder auch nicht.
2. Video-Dekodierung: die Input-Videoframes (Qualität Q1 unbekannt) werden als Einzelbilder dekodiert und temporär abgespeichert.
3. Re-Enkodierung: Hieraus werden temporäre Videodateien re-enkodiert, wobei man ihre Qualitätsstufe Q2 über viele Stufen hinweg iteriert
4. Vergleich: für jede Q2-Stufe wird das Differenzbild der Input-Videoframes und des temporären Videoframes berechnet.
5. Ghost-Effekt: Bei derjenigen Q2-Stufe, bei der das Differenzbild insgesamt am dunkelsten ist, wurde demnach die Inputvideo-Qualität am besten „erraten" ($Q2 \approx Q1$). Dieses Differenzbild zeigt also den Ghost-Effekt.
6. Detektion: Falls sich im Inputvideo auch Deepfake-manipulierte Bildbereiche befunden haben, werden diese im Differenzbild als heller sichtbare Bereiche signalisiert. Dieses Graustufen-Differenzbild wird abschließend gegen einen geeigneten Schwellwert in eine reine Schwarz-Weiß-Darstellung binarisiert.
7. Gesichtserkennung: das binarisierte Differenzbild wird nur in Bildbereichen ausgewertet, die überhaupt ein Gesicht zeigen. Hierzu wird, parallel zur Schritt 2.-6., eine Gesichtserkennung durchgeführt.

Die Gesichtserkennung ist notwendig, da die Untersuchungen gezeigt haben, dass im Bildhintergrund oftmals Fehlalarme beobachtet werden. Eine genauere Analyse (z. B. bei Videos aus Nachrichtensendungen) hat gezeigt, dass diese häufig von eingeblendeten Bildern, Laufschriften oder computergenerierten Inhalten ausgelöst werden. Für die Gesichtserkennung werden externe Bibliotheken genutzt, so etwa von Bulat (2019) oder Geitgey (2019).

Es muss gesagt werden, dass die Detektion nur erfolgreich ist, wenn die eingefügten Bildinhalte bei einer von Q1 abweichenden Qualitätsstufe enkodiert worden waren, also einem anderen Kompressions-„Lebenszyklus" unterworfen waren als der unveränderte Hintergrund.

Beispiel

Die Effektivität des Verfahrens kann man an den folgenden Beispielbildern erkennen: das erste Bildpaar zeigt ein Einzelbild des Originalvideos und des Deepfake-manipulierten Videos.

Abbildung 4.4: links: originaler Video-Frame; rechts: Deepfake-Manipulation und Gesichtserkennung (rot)

Im folgenden Bildpaar ist hierfür das Detektionsergebnis zu sehen. Man kann erkennen, dass die als Fälschung signalisierten Pixel im tatsächlich manipulierten Bild (rechts) viel dichter liegen als im unverfälschten Original-Frame (links). Durch geeignete Wahl eines Schwellwerts für diese Dichte wird eine eindeutige Klassifikation möglich.

Abbildung 4.5: links: Detektionsergebnis für original Video-Frame, rechts: für Deepfake-Mainpulation

IV. Erkennung von Desinformationen in Texten

Um eine linguistische Analyse vornehmen zu können, um darauf aufbauend maschinelle Lernverfahren zu trainieren, bedarf es eines Gegenkorpus (zu den Fake-News-Referenztexten aus Kap. 2.B) mit inhaltlich wahren Nachrichten. Diese wurden von den Online-Auftritten deutscher Zeitungen wie Süddeutsche Zeitung und Frankfurter Allgemeinen Zeitung bezogen (gecrawlt). Es wurden nur Artikel gecrawlt, die im gleichen Zeitraum wie die Referenztexte publiziert wurden. Zudem wurde darauf geachtet, dass die Artikel thematisch

mit dem Inhalt der verifizierten Fake News zusammenhängen (z. B. „Migration", „Innere Sicherheit", „Europapolitik", „US Präsidentschaftswahl 2016", etc.). Potthast et al. (2018) hatten in einer Studie gezeigt, dass keiner der von ihnen untersuchten Mainstream-Texte komplett falsch war. 97,5 % der Nachrichten aus Mainstream-Texten basieren auf seriöser und wahrhaftiger Berichterstattung. Nur 8 von 826 Texten (ca. 1 %) hatten eine Mischung aus korrekten und falschen Angaben. Diese Abweichung wurde akzeptiert und sich auf die Wahrhaftigkeit der Artikel verlassen.

Die Untersuchungen haben einige Forschungsergebnisse, welche auf englischen Daten vorgenommen wurden, bestätigt, andere konnten wiederlegt werden. Fake-News-Titel sind durchschnittlich länger als die Titel von True News. Untersuchungen haben gezeigt, dass vor allem Social-Networks-Nutzende (z. B. auf Facebook) dazu tendieren nur den Titel statt den gesamten Artikel zu lesen (Horne & Adali, 2017). Aus diesem Grund werden Fake News-Artikel, welche eine bestimmte politische Agenda vorantreiben, so verfasst, dass so viel Inhalt wie möglich im Titel zusammengefasst wird. Die Verwendung von Großbuchstaben kann hauptsächlich in Titeln von Clickbaiting-Artikeln beobachtet werden, welche versuchen, auf diese Weise Aufmerksamkeit des Lesers auf sich zu ziehen.

Potthast et al. (2018) haben gezeigt, dass der Textkörper von Fake News in der Regel kürzer ist als der von Mainstream-News. Unsere Forschungsergebnisse mit deutschen Daten haben ebenfalls gezeigt, dass Fake-News-Texte etwas kürzer verfasst sind, jedoch nicht signifikant. Es konnte nicht bestätigt werden, dass True News komplexere und längere Wörter verwenden (Rashkin et al., 2017).

Zudem wurde die Verteilung von Falschaussagen in Fake News analysiert. Wenn der Artikel eine Falschaussage enthält, wird diese in 45 % der Fälle bereits im Titel offeriert, in 39 % der Fälle im Textkörper. Weitere Fake-Statements können im Text verortet werden. In nur etwa 10 % der Fälle tauchen Falschaussagen im Vorspann auf oder können in Bildern verortet werden (5,3 %).

Rashkin et al. (2017) und Horne und Adali (2017) haben gezeigt, dass Fake News in ihren Artikeln häufiger Personalpronomen verwenden, um den Leser persönlicher zu adressieren. Im Projekt wurden die relativen Häufigkeiten von Personalpronomen in allen Nachrichtentexten untersucht. Fake News verwenden nicht signifikant mehr Personalpronomen. „Ich" wird beispielsweise häufiger in Nachrichten von Qualitätsmedien verwendet. Dies könnte aber damit zusammenhängen, dass diese Nachrichten häufiger Direktzitate

verwenden. Das Personalpronomen „wir“ kommt in beiden Referenzdaten gleich häufig vor.

1. Datenanalyse mit Machine Learning und Natural Language Processing Verfahren

Um die mithilfe des Crawling-Frameworks erfassten Textdaten verarbeiten und analysieren zu können, wurde im Rahmen des Projekts ein Analyse-Modul entwickelt, mit dessen Hilfe Texte strukturiert, repräsentiert und anschließend analysiert werden können.

Datenaufbereitung (Datenbereinigung und Datenstrukturierung)

Die gecrawlten Textdaten müssen vor ihrer Analyse entsprechend aufbereitet werden. Dies schließt im Wesentlichen eine Säuberung ein, die nicht auf der Metadatenebene, sondern auf der Inhaltsebene durchgeführt wird. Im Folgenden werden die einzelnen Schritte der Säuberung erläutert.

Da im Rahmen des Projekts nur deutschsprachige Texte untersucht werden sollten, galt es bezüglich der gecrawlten Datenströme nur relevante Texte zu extrahieren. Alle anderen Textdaten, deren Sprache nicht Deutsch war, wurden für die weitere Verarbeitung ausgeschlossen. Als Spracherkenner verwendeten wir eine eigene Implementierung, die auf Funktionswörter und Zeichen-N-Gramme basiert und eine nahezu perfekte Erkennung (> 99 %) bzgl. 50 Sprachen aufweist.

Nach der Extraktion deutschsprachiger Texte aus den Datenströmen galt es im nächsten Schritt die Texte entsprechend zu säubern. Hierfür verwendeten wir eine Pipeline, die die Texte sequentiell nach vorgegebenen Regeln säubert. Konkret wurden die Texte von Grundrauschen wie Hyperlinks, Überschriften, Bildsignaturen und Ähnlichen bereinigt. Des Weiteren wurden Datumsangaben, Währungen und anderweitige numerische Textstellen durch einheitliche Dummy-Tokens normalisiert (z. B. 21.5.2018 → <DATUM>, 20:12 → <UHRZEIT>). Dies hat den Effekt, dass Machine Learning Verfahren (kurz ML-Verfahren) von spezifischen Zahlen abstrahieren sollen, um einer Überanpassung (Overfitting) entgegenzuwirken, gleichzeitig aber die abstrakten Angaben nutzen können, um syntaktische Muster zu lernen, die hinsichtlich einer Unterscheidung zwischen Fake-/Nicht-Fake-Texten hilfreich sein können.

Nach Bereinigung der Textdaten galt es, diese mithilfe von Natural Language Processing (kurz NLP) Verfahren zu strukturieren, um bezüglich der Analyse auf die unterschiedlichsten Merkmale zuzugreifen. Zu solchen Merkmalen gehören z. B. Wortklassen (Funktionswörter/ Inhaltswörter), Wortarten (Nomen, Adjektive, Verben, Adverbien, etc.), Entitäten, semantische Relationen, Zitate, Redewendungen, etc. Es entstand hierfür ein gesondertes Framework, mit dessen Hilfe eine effiziente MerkmalEDsextraktion durchgeführt werden konnte. Als zugrundeliegende NLP-Werkzeuge dienten hierbei unter anderem Tokenizer, Part-of-Speech-Tagger, Chunker sowie Named Entitey Recognizer.

Datenrepräsentation

Ausgehend von dem Strukturierungsframework entstand ein weiteres Werkzeug mit dessen Hilfe Texte geeignet repräsentiert werden konnten, um diese seitens von ML-Verfahren untersuchen zu können. Hierfür entwickelten wir Repräsentationen basierend auf Bag-of-Features, Embeddings sowie Language Models. Die Repräsentationen können hierbei als Modelle aufgefasst werden, die entsprechende Texteinheiten numerische Werte zuweisen (Beispiel: absolute/relative Häufigkeiten von bestimmten Wörtern oder die Auftretenswahrscheinlichkeit eines Folgewortes in einer Sequenz von Wörtern).

Klassifikationsverfahren

Die zentrale Komponente des Analyse-Moduls sind sogenannte Klassifikatoren, mit deren Hilfe gegebene Texte automatisiert hinsichtlich vorgegebener Klassen eingeordnet werden können. Die Klassen, die im Vordergrund stehen sind „Fake“ und „Nicht-Fake“. Im Rahmen des Projekts entstanden mehrere Klassifikatoren, von denen ein Verfahren (genannt OCCAV) auf einer angesehenen internationalen Konferenz für Information Retrieval vorgestellt wurde (ECIR 2018). OCCAV basiert dabei auf einer Language Model Repräsentation und hat den markanten Vorteil, dass es ohne Training auskommt und Textdaten daher ohne Vorwissen direkt klassifizieren kann. Dies ist insbesondere in solchen Szenarien wichtig, in denen entweder keine Trainingsdaten existieren oder vorliegende Daten nicht geeignet sind. Vereinfacht ausgedrückt erlernt OCCAV ein Language Model aus einer Menge gegebener

Texte, die eine Klasse X repräsentieren, und versucht dieses Modell in einem unbekannten Text Y wiederzufinden. Wenn dies erfolgreich gelingt, wird angenommen, dass Y zur Klasse X gehört, ansonsten handelt es sich bei Y um eine andere Klasse. Übertragen auf das Projekt stellt X die Klasse der Fake-Texte, sodass wenn Y dieser Klasse zugewiesen wird, es sich ebenfalls um einen Fake-Text handelt, andernfalls um einen Nicht-Fake-Text.

Evaluierung

In einer Reihe von Experimenten testeten wir die Anwendung unserer Klassifikatoren zum Zwecke der Erkennung von Fake-News bzw. zur Diskriminierung zwischen Fake und Nicht-Fake.

Experiment: Unterscheidung von Fake News vs. True News auf Basis syntaktischer Strukturen

In diesem Experiment stellten wir einen Korpus zusammen, der 100 Klassifikationsfälle umfasste. Diese unterteilten sich in 50 Fälle, bei denen X mit Y übereinstimmt (unbekannte und bekannte Texte gehören der Klasse „Fake" an), und weitere 50 Fälle, in denen X mit Y nicht übereinstimmt (bekannte Texte = „Fake", unbekannter Text = „Nicht-Fake"). Als Klassifikator wählten wir hierbei OCCAV. Als Vorverarbeitungsschritt maskierten wir zunächst themenbehaftete Wörter, sodass nur noch syntaktische Strukturen wie Funktionswortphrasen inklusive Interpunktionszeichen in den Texten verblieben.

Beispiel

„Die fast regelmäßg gemeldeten Pannen der Bundeswehr lassen eigntlich aufhorchen, doch es scheint niemanden der Abgeordneten aus dem gleichgeschalteten Bundesparlament ernsthaft zu interessieren."

*„Die * * * * der * * * *, * es * * der * aus dem * * * zu *."*

Die Fragestellung, der somit nachgegangen werden soll, ist, ob eine Unterscheidung von Fake News gegenüber True News alleine auf Basis syntaktischer Strukturen möglich ist. Mit anderen Worten bedeutet dies, dass der Klassifikator sich nicht auf themenbehaftete Inhalte fokussieren soll. Als

Baselines wählten wir drei One-Class-Verfahren (PCA, SOS und LOCI) aus einem existierenden Framework (PyOD). Alle drei erzielten hinsichtlich des gegebenen Korpus ein zufälliges Klassifikationsergebnis (50 %). OCCAV erzielte hingegen eine Erkennungsgenauigkeit von 69 %. Dies zeigt, dass es möglich ist, eine Unterscheidung unabhängig von dem eigentlichen Inhalt der Nachrichtentexte zu erzielen, auch wenn das Ergebnis sicherlich verbesserungsfähig ist.

V. Malicious-Bot-Erkennung

Der Einsatz von Botnetzen ist ein Mechanismus, der auch im Kontext von Desinformationen zunehmend relevant ist. Wir beschreiben hier erste Ergebnisse und Ansätze einer automatisierten Bot-Erkennung.

1. Beschreibung des Datensatzes

Der bereitgestellte Trainingsdatensatz der Bots- und Gender-Profiling-Aufgabe auf PAN 2019 (Rangel et. al, 2019) besteht aus 4.120 englischen und 3.000 spanischen Twitter-Accounts. Jede dieser XML-Dateien enthält 100 Tweets pro Autor. Jeder Tweet wird in einem „Dokument“ XML-Tag gespeichert.

Jeder Autor wurde mit einer alphanumerischen Autoren-ID kodiert. Der englischsprachige Ordner enthält 2.060 Bot-Texte, 1.030 weibliche und die gleiche Anzahl männlicher Texte. Der spanische Ordner ist kleiner als der englische und umfasst 1.500 Bot-Texte und 750 Texte pro Geschlecht.

Um eine Überanpassung beim Training eines Klassifikators zu vermeiden, werden die Daten in ein Trainings- (70 %) und Test- (30 %)-Set aufgeteilt - wie von den PAN-Organisatoren empfohlen.

Bei Betrachtung des binären Klassifizierungsproblems „Bot vs. Human“ ist der Datensatz ausgeglichen. Wird sie jedoch zu einem Dreiklassenproblem umformuliert, dominiert die „Bot“-Klasse über die beiden Geschlechterklassen „männlich“ und „weiblich“. Unsymmetrische Daten beziehen sich auf eine ungleiche Verteilung von Klasseninstanzen. Dieses Ungleichgewicht kann durch den Einsatz der „Undersampling“-Technik weitgehend reduziert werden. Durch die zufällige Entfernung von Texten aus der Mehrheitsklasse ermöglicht diese einfache Methode die Erstellung ausgewogener Datensätze, die theoretisch zu eine Klassifikation führen, die nicht auf eine bestimmte

Klasse fokussiert ist. Durch das Undersampling der Bot-Klasse sind wir das Risiko eingegangen, wichtige Instanzen auszulassen, die wichtige Unterschiede zwischen den drei Klassen aufweisen können. Dadurch wurde die Anzahl der englischen Bot-Texte von 2.060 auf 1.030 reduziert, je nach Größe der Autoren und Autorinnen pro Klasse. Die spanischen Bot-Texte wurden von 1.500 auf 750 Instanzen reduziert. Zusätzlich haben wir den Trainingsdatensatz in drei kleinere Mengen aufgeteilt. 50 % der Daten wurden für das Training, 25 % für die Validierung und 25 % für die Testung der SVM verwendet.

2. Methodik

Im Folgenden wird für jede Sprache der gleiche Ansatz angewendet. Zuerst werden die Twitter-Daten vorverarbeitet, um Textbesonderheiten wie Hashtags, URLs und Benutzererwähnungen zu behandeln. Anschließend werden Wort-Unigramme und Bigramme sowie Zeichen-N-Gramme im Bereich von 3 bis 5 als Merkmale extrahiert, die als Input für das Training einer Support Vector Machine (SVM) dienen.

Vorverarbeitung

Die Vorverarbeitungspipeline ist für beide Sprachen (Englisch und Spanisch) nahezu gleich. Die folgenden Schritte werden durchgeführt, um die Tweets zu reinigen und zu strukturieren:

1. Konkatenierung aller 100 Tweets pro Autor zu einer langen Zeichenkette.
2. Kleinschreibung aller Zeichen.
3. Entfernen von Leerzeichen.
4. Ersetzen von URLs durch den Platzhalter <URL>.
5. Löschen von irrelevanten Zeichen, z. B. „+,*,/,/,“.
6. Ersetzen aller Hashtags und angehängter Token durch den Platzhalter < HashTag>. 7. Ersetzen von @-Mentions (z. B. @username) durch den Platzhalter <UsernameMention>.
7. Sequenzen mit gleichen Zeichen und einer Länge von mehr als drei werden entfernt.
8. Entfernen von Wörtern mit weniger als drei Zeichen.

9. Entfernen von Stoppwörtern mit Hilfe der NLTK (Natural Language Toolkit) Bibliothek.
10. Um die Wörter zu tokenisieren, haben wir den TwitterTokenizer aus der NLTK-Bibliothek verwendet.

Merkmale

Da die beiden Sprachen unterschiedliche Datensätze haben, wurden zwei separate Klassifikationsmodelle für jede Sprache trainiert. Wir haben verschiedene Feature-Sets getestet und mit Hyperparameter-Tuning experimentiert, manuell und mit der Grid-Suchfunktion von scikit-learn. Die Hyperparameter wurden für jedes Sprachmodell separat abgestimmt. Verschiedene Experimente werden in Abschnitt 6 diskutiert.

Nach der Vorverarbeitung wurde eine Wortfrequenzanalyse auf beiden Datensätzen durchgeführt. Wir haben die Trainings-, Validierungs- und Testsets zusammengeführt. Die drei am häufigsten verwendeten Token von Bots sind:

a. URLs (Token <URL>)
b. Hash-Tags (Token < HashTag>)
c. und @-Mentions (Token < UsernameMention >)

Während Bots eine höhere Neigung haben, URLs zu teilen, neigen Menschen dazu, sich in erster Linie auf andere Nutzende (oder Konten) zu beziehen, indem sie @-Mentions verwenden (markiert als <UsernameMention>-Token). Neben dem Verweis auf andere Nutzende verwenden Menschen am zweithäufigsten URLs. Der dritthäufigste Token, den Menschen auf Twitter verwenden, ist die Verwendung von Hashtags. Diese Analyse zeigt, dass diesen Token besondere Aufmerksamkeit geschenkt werden sollte, wenn Twitter-Texte vorverarbeitet werden.

Je nach Häufigkeitsverteilung wurden die 10.000 am häufigsten verwendeten Token im Trainingsset in einem Dictionary gespeichert. Beim Aufbau des Vokabulars wurden Begriffe mit einer Dokumentenfrequenz von weniger als 2 ignoriert.

Es wurden TF-IDF zum Vektorisieren verwendet, um eine Vektor-Pipeline für jede Sprache aufzubauen. Die folgenden N-Gramme für beide Sprachen wurden verwendet:

a. Wortunigramme und Bigramme

b. Zeichen-N-Gramme im Bereich von 3 bis 5

Die Art und Weise, wie die Wort- und Zeichenauswahl durchgeführt wurde, ist von Daneshvar und Inkpen (2018) inspiriert. Die Autoren präsentierten ihren Gender Identification-Ansatz für Twitter-Texte bei der PAN Challenge im Jahr 2018, bei der ihr Modell auf dem zweiten Platz landete.

3. Algorithmus des maschinellen Lernens

Um einen Klassifikator zu trainieren, verwendeten wir eine lineare SVM mit verschiedenen Wort- und Zeichen-N-Grammen als Features. Da wir die Aufgabe als ein Multiklassen-Problem betrachten, wurde die Entscheidungsfunktion OVR („One-vs.-Rest“) verwendet. OVR kombiniert mehrere binäre SVMs zur Lösung der Klassifikationsaufgabe mit dem Training einer Vielzahl von Klassen. Die drei zu trainierenden Klassen sind: „Bot“, „Männlich“ und „Weiblich“. Mit OVR klassifiziert jede SVM eine Klasse gegen alle anderen Klassen.

Um eine Überanpassung beim Experimentieren mit dem Trainingsset zu vermeiden, haben wir die von den Veranstaltern zur Verfügung gestellten Daten in drei Teile gegliedert. Für das Training haben wir 50 % der Daten verwendet. Die andere Hälfte des Datensatzes wurde zu gleichen Teilen als Validierungs- und Testsatz aufgeteilt (jeweils 25 % der Textdaten). Während der Experimente sah das Modell den Testsatz nicht. Die Parametereinstellung wurde am Validierungsdatensatz durchgeführt. Schließlich wurde jedes Modell auf dem offiziellen PAN 2019 Testset für den Author-Profiling-Task auf der TIRA-Plattform getestet.

VI. Ergebnisse

Die folgende Tabelle zeigt die Ergebnisse des Verfahrens, die mit dem vorläufigen Trainingsset erzielt wurden, sowie die Genauigkeitswerte mit dem offiziellen Testset. Die Genauigkeitswerte wurden für jede Sprache einzeln berechnet. Zuerst wurde die Genauigkeit bei der Identifizierung von Bot und Mensch berechnet. Dann, im Falle eines Menschen, wurde die Genauigkeit der Vorhersage ob Mann oder Frau berechnet. Jedes Modell wurde auf 50 % der Testdaten trainiert. Die Hyperparameter wurden auf dem 25 %igen Vali-

Tabelle 4.5: Genauigkeitswerte für Bot- und Geschlechtererkennung am „Early Bird“ und am offiziellen PAN 2019 Testdatensatz

	„Early Bird“		Testdatensatz	
Sprache	DE	ES	DE	ES
Bot vs. Mensch	0,97	0,97	0,92	0,91
Männlich vs. weiblich	0,94	0,93	0,82	0,78

dierungssplit angepasst. Schließlich wurde das eingereichte Modell auf dem offiziellen PAN 2019 Testset auf der TIRA-Plattform getestet.

VII. Weitere getestete Methoden und Merkmale

Neben den bereits beschriebenen Schritten der Vorverarbeitung und Merkmalsauswahl wurden auch andere Merkmale und Datenstrukturtechniken untersucht. Neben der vorgestellten SVM mit einem linearem Kernel wurden auch andere Klassifikatoren getestet, nämlich CNN und den Random Forest Classifier. In den Experimenten konnten diese beiden Klassifikatoren in Bezug auf die Leistung nicht mit der linearen SVM mithalten.

In den Experimenten wurden Twitter-Daten wie folgt bereinigt: Entfernung aller URLs, Hashtags, Retweets (RT) und @-Mentions. Experimente haben gezeigt, dass diese Features für die Bot-Erkennung von Twitter-Daten unerlässlich sind. Um die Token zu vektorisieren, wurde zunächst mit gesamten und relativen Wortfrequenzen sowie mit der Konvertierung der Tokens in tf-idf gearbeitet. Die Vektorlänge lag zwischen 1.000 und 10.000 der häufigsten vorkommenden Token. Die Experimente zeigten, dass die Genauigkeit abnahm, wenn die Hyperparameter mit der Rastersuchfunktion von scikit-learn angepasst wurden. Die Tabelle zeigt die Ergebnisse der Experimente mit dem Testdatensatz „Early Bird“.

C. Diskussion und Zusammenfassung

Die große Menge von Meldungen, die potentiell Desinformationen enthalten können, macht eine Unterstützung von menschlichen Beobachtern durch technische Maßnahmen notwendig. Diese Maßnahmen können heute noch nicht selbständig eine Aussage über Desinformationen machen und somit als

Tabelle 4.6: Genauigkeitswerte für Bot- und Geschlechtererkennungsexperimente auf dem Datensatz des PAN 2019 „Early Bird“ Testdatensatz.

	Bot vs. Human		Male vs. Female	
Sprache	DE	ES	DE	ES
Token Gesamthäufigkeit	0,91	0,83	0,65	0,64
Token Relative Frequenz (Grid Search Tuning)	0,73	0,83	0,53	0,64
Token TF-IDF Vektorisierung	0,92	0,78	0,81	0,61

autonomer Filter Kommunikationskanäle überwachen. Sie können aber einen Redakteur, der den Wahrheitsgehalt einer Meldung betrachtet, unterstützen.

Verschiedene Verfahren sind heute bereits so ausgereift, dass sie problemlos in der Praxis eingesetzt werden können. Das Wiedererkennen von Bildern und das Erkennen der Bestandteile von Bildmontagen weist nur noch Fehlerraten im Bereich von unter einem Promille und wenigen Prozent auf. Eine Nutzung erfordert hier allerdings den Aufbau entsprechender Referenzdatenbanken. Nur wenn die Bilder zuerst in einer Datenbank gespeichert sind, können sie auch wiedererkannt werden. Unterstützen können hier automatisierte Crawler, die selbständig Bilder finden und in die Datenbank einspeisen.

Andere Verfahren wie die Manipulationserkennung von Bildern und das Erkennen von Deepfake-Videos sind auf einem guten Weg, hier ist die Erkennung allerdings noch nicht so weit fortgeschritten, dass sie einfach einzusetzen sind. Sie können Hinweise geben und in geeigneten Fällen auch sehr präzise Bewertungen durchführen, sind aber noch nicht in der Lage, in allen Fällen eine Manipulation zu erkennen und neigen andererseits auch dazu, Fehlalarme auszulösen. Hier muss also der Anwender die Ergebnisse interpretieren und genug Fachkenntnis besitzen, eine abschließende Entscheidung zu treffen. Dies gilt ebenso für die Erkennung von Texten; sowohl linguistische Methoden als auch Ansätze aus dem Maschinellen Lernen zeigen, dass eine Erkennung von Desinformation und ähnlichen Inhalten mit hohen Trefferquoten möglich ist. Trotzdem muss bei Fehlerraten von 30 Prozent und mehr das abschließende Urteil von einem Anwender erfolgen.

Um hier in der Zukunft bessere Ergebnisse zu erzielen, ist das Schaffen einer besseren Trainingsgrundlage von großer Bedeutung. Maschinelles Lernen wird im Kontext von Desinformation erst dann wirklich erfolgreich sein können, wenn große Menge von Texten und auch anderen Medien, die als

Desinformation erkannt wurden, in Datenbanken abgelegt und entsprechend kommentiert sind. Auf diesen Daten können dann zukünftige Netze trainiert werden.

Die Nutzung erster Ergebnisse durch im Journalismus Tätige, also Fachanwender, kann in naher Zukunft erfolgen. Weiter in die Zukunft blickend ist aber auch eine Verbesserung der Verfahren bis zu einem Grad der Automatisierung denkbar, der die Methoden auch für den einzelnen Bürger verfügbar macht. Eine Integration in Browser würde dann Hinweise geben, dass beispielsweise eine Text einen Stil aufweist, in dem in der Vergangenheit schon zahlreiche belegte Desinformationen verfasst wurden, oder dass ein betrachtetes Bild Spuren von Manipulationen aufweist.

Ein automatisiertes Blockieren und Löschen von Inhalten durch entsprechende Verfahren hingegen wird immer problematisch sein, da hier Maschinen eine Aufgabe erhalten, die in der Praxis immer Fehlerraten aufweisen wird. Denkbar ist aber, dass im Falle der Verfügbarkeit entsprechender Methoden und der damit einhergehenden Erleichterung bei der Prüfung von Meldungen der Druck auch auf die Verbreiter von Nachrichten steigt, hier eine verantwortungsvolle Prüfung durchzuführen.

Literaturverzeichnis zu Kapitel 4

Afchar, D. und Nozick, V. und Yamagishi, J. & Echizen, I (2018). MesoNet: a Compact Facial Video Forgery Detection Network. In 2018 IEEE International Workshop on Information Forensics and Security (WIFS) (pp. 1-7). IEEE.

Afroz, S., Brennan, M., & Greenstadt, R. (2012). Detecting hoaxes, frauds, and deception in writing style online. In 2012 IEEE Symposium on Security and Privacy (pp. 461-475). IEEE.

Ahmed, H. (2017). Detecting opinion spam and fake news using n-gram analysis and semantic similarity, Ph.D. thesis, University of Victoria. https://dspace.library.uvic.ca//handle/1828/8796.

Allcott, H., Gentzkow, M. (2017), Social Media and Fake news in the 2016 Election. In: Journal of Economic Perspectives. 31(2), 211-236.

Bacciu, A., La Morgia, M., Mei, A., Nemmi, E. N., Neri, V., & Stefa, J. (2019). Bot and Gender Detection of Twitter Accounts Using Distortion and LSA. CLEF (Working Notes)

Banko, M., Etzioni, O., Soderland, S., Weld, D. S. (2008), Open information extraction from the web. IJCAI, Vol. 7, (pp. 2670-2676).

Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer, Berlin, Heidelberg.

Bayram, S., Sencar, H. T., & Memon, N. (2008, September). A survey of copy-move forgery detection techniques. In IEEE Western New York Image Processing Workshop (pp. 538-542). IEEE.

Belhassen Bayar and Matthew C. Stamm, A Deep Learning Approach to Universal Image Manipulation Detection Using a New Convolutional Layer, In Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, (pp. 5–10), 2016.

Belhassen Bayar and Matthew C. Stamm, Design Principles of Convolutional Neural Networks for Multimedia Forensics, Electronic Imaging, Media Watermarking, Security, and Forensics 2017, pp. 77-86(10), 2017.

Bianchi, T. und Piva, A. (2012). Image Forgery Localization via Block-Grained Analysis of JPEG Artifacts. IEEE Transactions on Information Forensics and Security. vol. 7, issue 3, S. 1003 ff. IEEE

Birajdar, G. K., & Mankar, V. H. (2013). Digital image forgery detection using passive techniques: A survey. Digital investigation, 10(3), 226-245.

Bulat, A. (2019) 2D and 3D Face alignment library build using pytorch. In: Github repository '1adrianb/face-alignment'. https://github.com/1adrianb/face-alignment

Castillo, C., Mendoza, M., Poblete, B. (2011). Information credibility on twitter. Proceedings of the 20th international conference on world wide web. ACM675–684.

Cavoukian, A. (2009). Privacy by design: The 7 foundational principles. Information and Privacy Commissioner of Ontario, Canada, 5.

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. Technischer Report. In: arXiv.org, Cornell University, Cornell University, Ithaca, NY, USA

Daneshvar, S., Inkpen, D.: Gender identification in twitter using n-grams and lsa: Notebook for pan at clef 2018. In: CLEF (2018)

Davis, C. A., Varol, O., Ferrara, E., Flammini, A., Menczer, F. (2016). Botornot: A system to evaluate social bots. Proceedings of the 25th international conference companion on world wide web. International World Wide Web Conferences Steering Committee273–274.

De Marneffe, M. C., MacCartney, B., & Manning, C. D. (2006, May). Generating typed dependency parses from phrase structure parses. In Lrec (Vol. 6, pp. 449-454).

deepfakes (Github user) (2019). Non official project based on original /r/Deepfakes [Reddit] thread. In: Github repository 'deepfake/faceswap'. https://github.com/deepfakes

Deepfakes web beta (2019). Create your own Deepfakes online. https://deepfakesweb.com/

Dickerson, J. P., Kagan, V., Subrahmanian, V. (2014). Using sentiment to detect bots on twitter: Are humans more opinionated than bots? Advances in social networks analysis and mining (asonam), 2014 IEEE/ACM international conference on. IEEE620–627.

Farid, H. (2009). Exposing Digital Forgeries From JPEG Ghosts. IEEE Transactions on Information Forensics and Security, vol. 4, issue 1, S. 154 ff., IEEE

Geitgey, A. (2019) The world's simplest facial recognition api for Python and the command line. In: Github repository 'ageitgey/face_recognition'. https://github.com/ageitgey/face_recognition

Gilani, Z., Farahbakhsh, R., Tyson, G., Wang, L., Crowcroft, J.: Of bots and humans (on twitter). In: Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017. pp. 349–354. ACM (2017)

Gilani, Z., Wang, L., Crowcroft, J., Almeida, M., Farahbakhsh, R.: Stweeler: A framework for twitter bot analysis. In: Proceedings of the 25th International Conference Companion on World Wide Web. pp. 37–38. International World Wide Web Conferences Steering Committee (2016)

Hany Farid, „Image forgery detection", IEEE Signal Processing Magazine, vol. 26, issue 2, pp. 16-25, 2009.

Horne, B. D., Adali, S. (2017): This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: International AAAI Conference on Web and Social Media, 759-766.Jin, Z., Cao, J., Zhang, Y., Luo, J. (2016). News verification by exploiting conflicting social viewpoints in microblogs. AAAI2972–2978.

Jason Bunk, et al, „Detection and Localization of Image Forgeries using Resampling Features and Deep Learning", CVPR Worshop on Media Forensics, July 2017.

Karataş, Arzum & Şahin, Serap. (2017). A Review on Social Bot Detection Techniques and Research Directions. ISCTurkey 10th International Information Security and Cryptology Conference, At Ankara, Turkey

Kowalski, M. (2018). 3D face swapping implemented in Python. In: Github repository 'MarekKowalski/FaceSwap'. https://github.com/MarekKowalski/FaceSwap

Kwon, S., Cha, M., Jung, K., Chen, W., Wang, Y. (2013). Prominent features of rumor propagation in online social media. Data mining (icdm), 2013 ieee 13th international conference on. IEEE1103–1108.

Li, W. und Yuan, Y. und Yu, N. (2009). Passive detection of doctored JPEG image via block artifact grid extraction. Signal Processing. vol. 89, issue 9, S. 1821 ff. Elsevier

Li, Y. und Chang, M.-C. und Lyu, S. (2018). In Ictu Oculi: Exposing AI Created Fake Videos by Detecting. Eye Blinking. Technischer Report. University at Albany, State University of New York, NY, USA.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

Luo, W., Huang, J., & Qiu, G. (2010). JPEG error analysis and its applications to digital image forensics. IEEE Transactions on Information Forensics and Security, 5(3), 480-491.

Niu, X. M., & Jiao, Y. H. (2008). An overview of perceptual hashing. Acta Electronica Sinica, 36(7), 1405-1411.

Pan, J. et al. (2018): Content based fake news detection using knowledge graphs. In International Semantic Web Conference (pp. 669-683). Springer, Cham.

Pawel Korus, „Digital image integrity – a survey of protection and verification techniques“, Digital Signal Processing 71, pp. 1 –26, 2017.

Potthast, M., Hagen, M., Stein, B. (2016): Author Obfuscation: Attacking the State of the Art in Authorship Verification. In: CLEF (Working Notes), 716-749.

Rangel, F., Rosso, P.: Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. In: Cappellato, L., Ferro, N., Losada, D., Müller, H. (eds.) CLEF 2019 Labs and Workshops, Notebook Papers. CEUR-WS.org.

Rashkin, H., Choi, E., Jang, J. Y., Volkova, S., Choi, Y. (2017): Truth of Varying Shades: Analyzing Language in Fake news and Political Fact-Checking. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, (pp. 2931-2937).

Raskin, V. (1987). Linguistics and natural language processing. Machine translation: Theoretical and methodological issues.. Cambridge University Press, Cambridge (pp. 42–58).

Rössler, A. und Cozzolino, D. und Verdoliva, L., Riess, C., Thies, J., & Nießner, M (2018). FaceForensics: A Large-scale Video Dataset for Forgery Detection in Human Faces. Technischer Report. In: arXiv.org, Cornell University, Ithaca, NY, USA

Rubin, V., Chen, Y., Conroy N. J. (2015): Deception detection for news: three types of fakes. In: Proceedings of the 78th ASIS & T Annual Meeting: Information Science with Impact: Research in and for the Community (ASIST '15). American Society for Information Science, Silver Springs, MD, USA, Article 83.

shaoanlu, clarle (Github users). (2019). A denoising autoencoder + adversarial losses and attention mechanisms for face swapping. In: Github repository 'shaoanlu/faceswap-GAN'. https://github.com/shaoanlu/faceswap-GAN

Shin, J., & Ruland, C. (2013, October). A survey of image hashing technique for data authentication in WMSNs. In 2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob) (pp. 253-258). IEEE.

Shu, K., Sliva, A., Wang, S., Tang, J., & Liu, H. (2017). Fake news detection on social media: A data mining perspective. ACM SIGKDD Explorations Newsletter, 19(1), 22–36.

Steinebach, M., Liu, H., & Yannikos, Y. (2012, February). Forbild: Efficient robust image hashing. In Media Watermarking, Security, and Forensics 2012 (Vol. 8303, p. 83030O). International Society for Optics and Photonics.

Tacchini, E., Ballarin, G., Della Vedova, M. L., Moret, S., de Alfaro, L. (2017). Some like it hoax: Automated fake news detection in social networks. arXiv:170407506.

Varol, O., Ferrara, E., Davis, C.A., Menczer, F., Flammini, A., (2017). Online human-bot interactions: Detection, estimation, and characterization. In: Eleventh international AAAI conference on web and social media.

Yang, J., Counts, S. (2010). Predicting the speed, scale, and range of information diffusion in twitter. ICWSM, 10(2010), 355–358.

Zhang, X., Ghorbani, A. A. (2019). An overview of online fake news: Characterization, detection, and discussion. Information Processing & Management.