HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

BACHELORARBEIT

Herr/Frau
Vincent F. Grunwald

Ansätze für die Dunkelfeldforschung

Möglichkeiten von Simulationen auf Grundlage der PKS für den Deliktbereich Cyberkriminalität

2021

Fakultät: Angewandte Computer- und Biowissenschaften

BACHELORARBEIT

Ansätze für die Dunkelfeldforschung

Möglichkeiten von Simulationen auf Grundlage der PKS für den Deliktbereich Cyberkriminalität

Autor:
Herr Vincent F. Grunwald

Studiengang:
IT-Forensik und Cybercrime

Seminargruppe:
CC17

Erstprüfer:
Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:
Dr. rer. nat. Michael Spranger

Augsburg, 23.08.2021

Faculty of Applied Computer- and Biosciences

BACHELOR THESIS

Approaches for dark field research

Possibilities of simulations based on the PKS for the crime area of cybercrime

author:
Mr. Vincent F. Grunwald

course of studies:
IT-Forensics and Cybercrime

seminar group:
CC17

first examiner:
Prof. Dr. rer. nat. Dirk Labudde

second examiner:
Dr. rer. nat. Michael Spranger

submission:
Augsburg, 23.08.2021

Bibliografische Angaben

Grunwald, Vincent F.

Ansätze für die Dunkelfeldforschung: Möglichkeiten für Simulationen auf Grundlage der PKS für den Deliktbereich Cybercrime

Approaches for dark field research: possibilities of simulations based on the PKS for the crime area of cybercrime

34 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Medien, Bachelorarbeit, 2011

Abstract

Das Ziel der vorliegenden Arbeit war es, den Deliktbereich Cybercrime auf Vorhersagbarkeit seiner Verbrechensrate zu testen. Diese Testung erfolgte, indem nach Trends und Gesetzmäßigkeiten innerhalb der bisherigen Geschehnisse des Deliktbereichs gesucht wurde, wozu Daten der Polizeilichen Kriminalstatistik herangezogen wurden. Die Suche nach Trends und Gesetzmäßigkeiten erfolgte, indem versucht wurde, auf Basis eines Hidden-Markov-Modells diejenigen Parameter zu identifizieren, anhand derer sich die aussagekräftigsten Gesetzmäßigkeiten bilden ließen. Die Arbeit basiert auf PUX, eine vom Autor erstellte Webtechnologie-basierte Anwendung, mit der PKS-Daten gespeichert und mit Hilfe von Visualisierung und Simulation erforscht werden können.

Inhaltsverzeichnis

Abkürzungsverzeichnis

AQ: Aufklärungquote
BKA: Bundeskriminalamt
HMM: Hidden Markov Modell
HZ: Häufigkeitszahl: Fälle pro 100.000 Einwohner
KI: Künstliche Intelligenz
ML: Machine Learning
PKS: Polizeiliche Kriminalstatistik
UI: User Interface

Abbildungsverzeichnis

Tabellenverzeichnis

1 Vorhersage von Verbrechen als Präventionsstrategie

Internet Of Things, Industrie 4.0, Machine Learning - diese Begriffe fallen in den letzten Jahren immer häufiger. Grund dafür ist die Ausweitung der Digitalisierung in sämtlichen Lebensbereichen. Dies zog Entwicklungen wie papierloses, digitalisiertes Arbeiten, Smart Homes - mit dem Internet verbundene und vernetzte Alltagsgegenstände im Haus - und elektronische Dokumente, sowie Zahlvorgänge nach sich. Insbesondere im Zuge der Corona-Pandemie hat die Digitalisierung zunehmend an Bedeutung gewonnen. Veränderungen durch Social Distancing, wie das Homeoffice oder Homeschooling und beständige Kontaktdatenerfassung führten zur Beschleunigung und Normalisierung der Digitalisierung. [1] Dennoch schafft die Digitalisierung nicht nur Erleichterung und Sicherheit. Mit der wachsenden Anzahl digitalisierter Lebensbereiche eröffnen sich gleichermaßen mehr Möglichkeiten für Angriffsvektoren und somit für digitale Verbrechen. Zudem trägt die Digitalisierung dazu bei, dass digitale Verbrechen lukrativer werden und sich somit häufen. [2]

Um dieser ungewollten Entwicklung entgegenzuwirken, existieren verschiedene Ansätze. Diese bestehen insbesondere aus der Prävention und im Verbrechensfall aus der Verfolgung der Straftat. Die Prävention betreffend liegt dabei die Verantwortung primär bei Privatpersonen und Unternehmen selbst, durch die Implementation von Sicherheitsmaßnahmen, wie Firewalls, Updates, Intrusion-Detection-Systems und Anti-Viren-Programmen. Die Reaktion auf ein bereits erfolgtes Verbrechen erfolgt hingegen nach der Meldung dessen vorrangig von Seiten der staatlichen Behörden. Forensische Analysen und die Auswertung kompromittierter Systeme sind dabei Vorgehensweisen, um zur Verbrechensaufklärung beizutragen. [3] Dennoch existieren auch für den Staat Möglichkeiten, präventiv einzugreifen, wie beispielsweise die Dunkelfeldforschung. Diese versucht, die erfassten Daten, auch Hellfeld genannt, auf die tatsächlichen Zahlen, welche das Dunkelfeld abbilden, auszuweiten. [4] In der folgenden Arbeit soll versucht werden, den Deliktbereich Cybercrime auf Vorhersagbarkeit der Verbrechensrate zu testen. Dies soll erfolgen, indem nach Trends und Gesetzmäßigkeiten innerhalb der bisherigen Geschehnisse des Deliktbereichs gesucht wird. Diese Suche erfolgt, indem auf Basis eines Hidden-Markov-Modells diejenigen Parameter identifiziert werden, anhand derer sich die aussagekräftigsten Gesetzmäßigkeiten bilden lassen.

2 Theorie der Kriminalität

Im folgenden Abschnitt werden verschiedene theoretische Aspekte zur Computerkriminalität betrachtet. Zuerst wird auf die Mikro- und die Makro-Ebene der zu betrachtenden Kriminalität und die Simulation der Kriminalität eingegangen, anschließend erfolgt eine Definition der Computerkriminalität im engeren und im weiteren Sinne. Ergänzend dazu wird das Predictive Policing erläutert und eine Unterscheidung zwischen Hellfeld und Dunkelfeld skizziert.

2.1 Kriminalität

Die Entstehung der Kriminalität unterliegt verschiedenen Faktoren. Kriminalität entsteht zum einen daraus, dass eine Machtinstanz oder ein Gesetzgeber gewisse Handlungen als Bedrohung für die soziale Ordnung einstuft - wobei diese Bedrohung weiter gefasst werden kann, als nur die direkte Bedrohung - und als Versuch, diese Handlungen zu verhindern, ein Regelwerk definiert, in dem diese Handlungen als Verbote festgelegt werden. Um diesen Verboten eine Wirkung zu verleihen, wird jedem Verstoß gegen besagte Verbote eine Strafe oder Konsequenz zugeordnet. Um zusätzlich eine abschreckende und somit präventive Wirkung zu erzeugen, werden diese Strafen für Verstöße öffentlich diskutiert und verhängt. Zum anderen gibt es keine Kriminalität ohne eine Handlung, die gegen diese Verbote verstößt. Die soziale Ordnung ist jedoch nicht das einzige zu schützende bzw. Gegenstand der Verbote. Sogenannte Rechtsgüter spielen eine ebenso große Rolle als Faktor, gegen welchen Kriminalität ausgeübt werden kann. Somit ist die Verletzung, das Gefährden oder das Angreifen solcher Rechtsgüter nach den durch den Gesetzgeber festgelegten Regularien ebenfalls kriminell. Des Weiteren wird nach vier verschiedenen Definitionen der Kriminalität unterschieden. Die Handlungen, welche vom Gesetzgeber als strafbar festgelegt wurden, fallen unter die Kriminalität im strafrechtlichen und theoretischen Sinne. Betrachtet man zudem die Situationen oder Handlungen, welche von einem beliebigen Betrachter als unrechtmäßig, gegen das Gesetz verstoßend oder strafbar beurteilt werden, unabhängig von der tatsächlichen Gesetzgebung und Rechtsprechung, so spricht man von der Kriminalität im moralunternehmerischen Sinne. So wird deutlich, dass die Definition von Kriminalität aus Sicht des Gesetzes nicht immer mit der moralgeleiteten Empfindung von Recht und Kriminalität innerhalb der Gesellschaft übereinstimmt. Fasst man die Ereignisse, welche unter die Kriminalität im strafrechtlichen, beziehungsweise theoretischen Sinne fallen, allerdings noch nicht von gesetzgebenden und Kontrolle ausübenden Instanzen erfasst wurden, zusammen, so spricht man von Kriminalität, welche informell definiert ist. Im Gegensatz dazu steht die Kriminalität, welche formell definiert ist. Unter diesem Begriff werden all diejenigen Handlungen und Ereignisse zusammengefasst, welche ebenfalls unter Kriminalität im strafrechtlichen oder theoretischen Sinne verbucht werden, allerdings von gesetzgebenden und Kontrolle ausübenden Instanzen bereits erfasst wurden und somit in der Kriminalstatistik sichtbar werden. [5]

2.2 Mikro- und Makro-Ebene der Kriminalität

Eine große Schwierigkeit im Aufstellen von kriminologischen Theorien besteht darin, alle auf den Kontext der jeweiligen Theorie einflussnehmenden Phänomene zu berücksichtigen. Beispiele für solche Phänomene sind etwa Recht, Verbrechen, Strafe, gesellschaftliche Normen, soziale Strukturen und individuelle Bedürfnisse der Bürger. Ein solch komplexes Konstrukt versucht man, mittels eines Makro-Mikro-Makro-Modells zu beschreiben. In diesem Modell beschreibt die Makro-Ebene die Kriminalitätsrate und die Mikro-Ebene das kriminelle Handeln von Einzelpersonen. Es wird deutlich, dass die beiden Ebenen direkt miteinander verbunden sind. Begeht eine Einzelperson ein Verbrechen, so steigt natürlicherweise die Verbrechensrate. Mit diesem Modell soll allerdings insbesondere versucht werden, jedes Einzelgeschehen als Folge und wiederum als Faktor für nachfolgende Ereignisse zu betrachten. Das Ziel dadurch ist es, den Gesamtkontext und Zusammenhänge zwischen den einzelnen Geschehen zu berücksichtigen. Hierbei wird insbesondere die Wechselwirkung von Akteuren zueinander betrachtet und Gewichtungen zugesprochen. Diese Akteure können prinzipiell in drei Arten unterteilt werden. Zum einen gibt es die Einzelpersonen, welche kriminell handeln. Schließen sich mehrere dieser Einzelpersonen zusammen, bilden sich daraus kriminelle Gruppen oder Verbände, wie beispielsweise Schwarzmärkte oder Clans. Die Bildung solcher kriminellen Gruppen zieht gleichsam die Bildung der dritten Gruppe, der Gruppe der Kriminalitätsbekämpfung durch Gesetze und Polizei und Kriminalamt mit sich. Durch diese Wechselwirkungen werden die Außeneinflüsse auch stetig verändert, was zur Folge hat, dass die zu Anfang betrachteten, einflussnehmenden Phänomene sich mit verändern. [5]

2.3 Simulation von Kriminalität

Kombiniert man die Mikro- und Makro-Ebene der Kriminalität mit weiteren Modellmöglichkeiten, wie beispielsweise den Multi-Agenten-Systemen und den zellulären Automaten, so eröffnet sich eine Möglichkeit, eine Simulation der Kriminalität zu erstellen. Unter zellulären Automaten versteht man einen zu simulierenden Raum, welcher in einzelne Teilgebiete aufgeteilt ist. Diese Teilgebiete folgen alle denselben Regelmäßigkeiten und stehen in unmittelbarer nachbarschaftlicher Beziehung, woran sich Dynamiken und interagierende Prozesse optimal darstellen lassen. [6, 7] Ein Multi-Agenten-System bietet ebenfalls die Möglichkeit zur Simulation diverser Zusammenhänge und Systeme. Hierbei geht man von mehreren unabhängigen Akteuren, auch Agenten genannt, aus, welche selbstständig, in Kontakt mit anderen Agenten und mit der Fähigkeit, über Zusammenhänge zu lernen, innerhalb eines Systems agieren. Kombiniert man diese beiden Möglichkeiten zur Modellierung, gelingt es, starre Umweltbedingungen und selbstständig handelnde Akteure miteinander in Verbindung zu bringen. [7, 8, 9] Wie in 2.2 beschrieben, üben die Mikro- und die Makro-Ebene Einfluss auf diverse Phänomene aus, welche ihrerseits wiederum diverse Abläufe innerhalb der Kombination aus zellulären Automaten und Multi-Agenten-Systemen beeinflussen. Aus einer gesamten Betrachtung der Auswirkungen, welche die Mikro- und Makro-Ebene, die einflussnehmenden Phänomene und das System aus zellulären Automaten und Multi-Agenten-Systemen aufeinander haben, lassen sich in letzter Instanz Vorhersagen

und Erkenntnisse gegenüber dem Kriminalitätsgeschehen, welches die Simulation darstellt, treffen. [7]

2.4 Computerkriminalität im engeren und weiteren Sinne

Der Begriff „Cybercrime“ findet in etlichen Wissenschaften Verwendung und ist dementsprechend in diversen Kontexten unterschiedlich zu verstehen. Die unterschiedlichen Zusammenhänge, welchen den Begriff prägen, können von technischer, rechtlicher, wirtschaftlicher, aber auch psychologischer, soziologischer und insbesondere kriminologischer Natur sein. Der tatsächliche Begriff des „Cybercrime“ ist eine Zusammensetzung mehrerer Begriffe, welche ihre Bedeutungen in einem Wort vereinen. Der erste Bestandteil ist somit „Cyber“, vom englischen Begriff „Cyberspace“, eine Art virtueller Realität. Der zweite Begriff hingegen ist „Crime“, Englisch für „Verbrechen“. Somit zeigt sich schon anhand der Wortzusammensetzung, worum es sich bei Cybercrime tatsächlich handelt: Verbrechen und Straftaten, welche mithilfe von Informations- und Kommunikationstechnik begangen werden. [10] Im engeren Sinne fallen unter diese Definition alle Delikte, die es in keiner Variante offline gibt. Diese Kategorie des Cybercrime umfasst unter anderem die Verletzung der Vertraulichkeit, Integrität und Verfügbarkeit von Netzwerken, sowie von Geräten, Daten und Services innerhalb dieser Netz- werke. Dazu zählen Hacking, Cyber-Vandalismus und die Verbreitung von Viren. [10, 11] Der Begriff des Cybercrime kann allerdings noch weiter gefasst werden, wenn man gewöhnliche Verbrechen betrachtet, welche durch den Einsatz von Informations- und Kommunikationstechnik an Reichweite oder Gravidität gewinnen. Dazu zählen beispielsweise diverse Formen des Betrugs, des Diebstals und der sexuellen Verbrechen. So wird traditioneller Betrug zu Betrug über Online-Banking, Sexualverbrechen weiten sich aus auf Kinderpornographie und Kontaktaufnahme zu Kindern über das Internet und Diebstahl erweitert sich von rein materiellen Gütern zu Daten und Kreditkarteninformationen. [12]

Im Umfeld des Cybercrimes spielen einige Akteure eine tragende Rolle. Offensichtlich gibt es bei einem Verbrechen immer mindestens zwei beteiligte Parteien, das Opfer und den Täter. Schnell wird aber ersichtlich, dass diese beiden Parteien nicht in einem Vakuum existieren und daher nicht die einzigen Akteure in diesem Umfeld sind. Bei genauerer Betrachtung sind auch die offenkundigen Parteien nicht einfach zwei Personen, wie es bei einem klassischen Einbruch beispielsweise der Fall wäre. Die Rolle des Einbrechers am Beispiel des Einbruchs ließe sich im Bereich des Cybercrimes auf einen Hacker, einen Staat, eine Firma oder andere Personen mit krimineller Absicht übertragen. Bei dem Opfer des Cyberverbrechens kann es sich beispielsweise auch um eine Gruppe Privatpersonen oder eine Firma, oder einen Staat handeln. [10]

In den Medien wird im Zusammenhang mit Cyber-Kriminalität häufig ein bestimmtes Wort genannt: Hacker. Ein Hacker ist nach der ursprünglichen Definition eine Person, die kreative

Löungen für Probleme mathematischer oder informatischer Natur findet. Im allgemeinen Sprachgebrauch wird darunter aber meist einen Kriminellen mit technischem Hintergrund. Befasst man sich genauer mit der Materie, gibt es einige spezifizierte Begriffe, die unter dem Begriff des Hackers zusammengefasst werden. Zunächst wird in 3 Gruppen unterschieden, dem Whitehead, dem Greyhead und dem Blackhead, wobei unter Whiteheads Ethical Hacker, Penetrationstester und weitere Security-Experten, die aktiv nach Sicherheitslücken suchen, mit der Absicht diese Lücken dann zu schließen, verstanden werden. Greyheads betitelt die Gruppe der Hacker, die nicht im Sinne des Gesetzes handeln, aber dennoch moralische Werte innehalten, wie Hacktivisten. Die letzte Gruppe beinhaltet den Personenkreis, der gemeinhin von der Mehrheit als Hacker verstanden wird. Dieser umfasst Kriminelle, die auf ihren eigenen Vorteil handeln und Daten stehlen, mit Hilfe von Ransomware Firmen erpressen oder sich auf ähnliche Weise illegal bereichern. Unabhängig dieser Klassifizierungen, die einen Hinweis auf die Motivation der Hacker gibt, gibt es weitere Kategorisierungen, die den Fokus eher auf das Level der Fähigkeiten legt. Hierbei wird in Scriptkiddy, Spammer, Exploit-/Malware-Coder und zuletzt den Cyber- Krieger unterschieden, wobei hier dem Skill-Level aufsteigend aufgezählt wurde. [10]

2.5 Predictive Policing

Der Wunsch, Verbrechen verhindern zu können, noch bevor sie eintreten wird auch medial, beispielsweise in dem Film Minority Report mit Tom Cruise aufgegriffen und ein fiktives Szenario in der Zukunft gezeichnet. In diesem Szenario ist die Polizei in der Lage, präzise Vorher- sagen zum zukünftigen Verbrechensgeschehen zutreffen, und auf Grund dieser Vorhersagen Festnahmen durchführt, wobei die Verdächtigen auch auf Basis noch nicht begangener Verbrechen verurteilt werden. [13] Würde man eine völlig gläserne Bevölkerung erlangen wol- len, um am richtigen Zeitpunkt am richtigen Ort sein zu können, und um Verbrecher auf frischer Tat ertappen zu können, würde es sich um eine Herausforderung handeln, die nicht zuletzt einige moralische und ethische Fragen aufwerfen wird. [14] Wie in der Hinleitung schon umrissen, versuchen die Behörden auch auf dem Feld der Prävention tätig zu werden. Diese Präventionsmaßnahmen werden im modernen Sprachgebrauch unter Predictive Policing zusammengefasst. Wortwörtlich heißt dies so viel wie "Vorhersagende Polizeiarbeit", und soll verdeutlichen, dass diese Art der Polizeiarbeit nicht beginnt, nachdem ein Verbrechen bereits geschehen ist, sondern ansetzt bevor das kriminelle Ereignis erfolgt. [15] Um tatsächliche Vorgänge ersichtlicher zu machen und um Unschärfe zu minimieren, kann die Dunkelfeldforschung zu Rate gezogen werden. Hierbei werden zum Teil anonyme Befragungen durchge- führt, um auch Verbrechen auf die Spur zu kommen, die aus persönlichen Gründen nicht zur Anzeige kamen. [16, 4] Unter dem Begriff Predictive Policing verbirgt sich zu großen Teilen keine abstrakte KI, die vorhersagt, wo und zu welcher Uhrzeit welches Verbrechen begangen wird, sondern vielmehr Ansätze wie das Bewerten von Vierteln, um die Polizeipräsenz anzupassen, um weitere Verbrechen durch Abschreck-Maßnahmen zu vereiteln oder schneller rea- gieren zu können und um somit die Verbrechensrate zu senken. An den Erweiterungen dieser Ansätze wird stetig gearbeitet. [17] Das Feld spannt sich jedoch weiter als die bloße Organisation von Polizeipräsenz. In Chicago geht das Konzept sogar so weit, dass einmal straffällig Gewordene von einer KI beurteilt werden und einen Score

zugeteilt bekommen, der aussagt, wie wahrscheinlich es ist, dass sie wiederholt straftätig werden. Basierend auf diesem Score wird dieser Bürger daraufhin streng überwacht. [18] Dieses Vorgehen der Behörden hat schon erheblich mehr gemein mit der fiktiven Zukunft aus dem Film.

2.6 Vergleich Hellfeld und Dunkelfeld

Im Verlauf der Arbeit werden Thematiken behandelt, in denen das Dunkel-, ebenso wie das Hellfeld erhebliche Relevanz erfahren. Somit soll im Folgen- den betrachtet werden, was unter diesen beiden Begriffen verstanden wird, und worin sich die beiden Bereiche unterscheiden. Als Hellfeld werden in einem Modell, welches Wissen auf eine spezifische Thematik bezogen, abbilden soll, alle erfassten Informationen und verstanden. Bezieht man sich auf das Themengebiet der Kriminalität, so fallen Analogien zwischen dem Hellfeld und der in 2.1 erwähnten Kriminalität, welche formell definiert ist, auf. Man spricht von bereits erfassten Informationen, beziehungsweise in dieser Thematik von bereits erfassten Kriminalfällen. Abbildung 1 stellt das Hellfeld dem Dunkelfeld gegenüber. Der helle Kreis in der Abbildung symbolisiert dabei das Hellfeld. Wie deutlich erkennbar wird, ist das Hellfeld ein Ausschnitt aus dem absoluten Dunkelfelds, in der Abbildung als graues Rechteck dargestellt. Das Dunkelfeld an sich umfasst alle Daten, die erfasst werden könnten, oder präziser ausgedrückt, alle existierenden Fälle und Informationen. Dies inkludiert auch die nicht erfassten und unbekannten Fälle. Dabei das Dunkelfeld allerdings noch einmal unterteilt, in das eben schon genannte absolute Dunkelfeld und das relative Dunkelfeld. Das absolute Dunkelfeld stellt dabei das Dunkelfeld in seiner Ge- samtheit dar, das relative Dunkelfeld beinhaltet nur den Teil, der mit Hilfe von Dunkelfeldforschung aufgehellt werden konnte. Das Dunkelfeld zeigt, ebenso wie das Hellfeld, Parallelen zu einer der in 2.1 genannten Definitionen der Kriminalität. Das Dunkelfeld ähnelt durch die Tatsache, dass es noch nicht erfasste Informationen und Fälle beinhaltet, stark der Kriminalität, welche informell definiert wird, da auch diese aus noch nicht erfassten Kriminalfällen besteht. Wie ebenfalls aus der Grafik ersichtlich wird, überlappen sich auch das relative Dunkelfeld sowie das Hellfeld in geringen Teilen, sind aber im Endeffekt beides Teile des absoluten Dunkelfelds, welches, angewandt auf die Thematik der Kriminalität, seinerseits noch einmal in kriminelle Ereignisse und Ereignisse, welche nicht als Kriminalität gewertet werden, unterteilt ist. Letztere werden nicht vom Strafrecht als kriminelle Ereignisse gewertet, werden aber von Opfern oder Zeugen als Kriminalität gewertet und auch von der Polizei als solche fest- gehalten. [19] Ziele der Dunkelfeldforschung sind hauptsächlich die Ermittlung der realen Viktimisierung der Bevölkerung. [20, 21, 22]

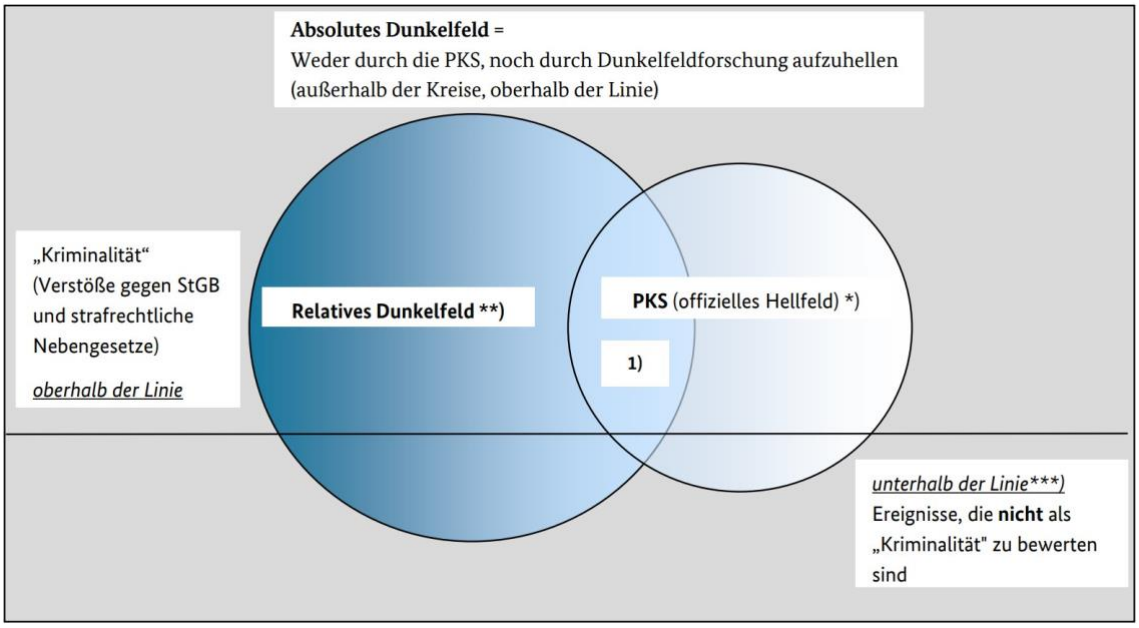

Abbildung 1: Hell- und Dunkelfeld aus PKS
[19]

3 Theorie zu Markov

Eine nicht zwar alte, aber neuentdeckte Art an Probleme heranzugehen, erfreut sich zurzeit wachsender Beliebtheit. Die Rede ist von Machine Learning oder kurz ML. Der Anwendungsbereich des ML wächst stetig. Er erstreckt sich von Image-Recognition, also dem Erkennen von Dingen in Bildern, der Anpassung von Werbevorschlägen, Spracherkennung, Sprachsteuerung, Bildkorrekturen, automatischen Antworten bei Emails und Übersetzungen über autonomes Fahren, anomaliebasierte Angriffserkennungssysteme, Chatbots, Robotics und persönliche Assistenten. [23, 24] Es existieren einige Ansätze des Machine Learnings, einer der bekanntesten ist der Ansatz der neuronalen Netzwerke [25]. Ein weiterer Ansatz ist das Hidden- Markov-Modell, es findet unter anderem in der Spracherkennung heute schon Anwendung [26]. Andrei Markov war ein russischer Mathematiker, der ein stochastisches Modell entwickelte, mit dem sich voneinander Abhängige definierte Zustände im Kontext von komplexeren Systemen beschreiben lassen [27]. Im Folgenden wird zunächst auf das Machine Learning als Form der künstlichen Intelligenz und im Anschluss daran auf Markov-Ketten und auf das Hidden- Markov-Modell als Möglichkeit zur Beschreibung von eingegangen.

3.1 Machine Learning als Form der künstlichen Intelligenz

Maschinelles Lernen, oder auch Machine Learning, ist eine Form der künstlichen Intelligenz. Obwohl diese beiden Begriffe häufig als Synonyme für einander verwendet werden, gibt es doch Unterschiede und beide betiteln unterschiedliche Bereiche. Unter künstlicher Intelligenz werden Computerprogramme zusammengefasst, die selbstständig in der Lage sind Probleme, meist aus einer Problem-Familie stammend, zu lösen. Mit dem Begriff Machine Learning werden Algorithmen klassifiziert, die aus Daten, meist Big-Data, lernen können und durch die erlernten Muster in der Lage sind Beziehungen zu erkennen. Die Ideen oder mathematischen Ausarbeitungen sind nicht neu. Über künstliche Intelligenz wurde bereits in den 1950er Jahren gesprochen. Im Bereich des maschinellen Lernens werden Algorithmen in zwei Unterkategorien unterteilt. Beim überwachten Lernen bzw. supervised learning gibt es einen Lehrer, der in der Trainingsphase überwacht, ob die Ausgabe Werte des Systems mit den bekannten richtigen Werten übereinstimmen. Dem gegenüber steht das unsupervised learning oder zu Deutsch unüberwachtes Lernen und somit die Algorithmen, die aus gegebenen Daten ein stochastisches Modell aufbauen, welches Zusammenhänge abbildet und darauf basierend Vorhersagen treffen kann. Diese Modelle passen ihre Parameter beim Lernen an. [23, 28] Ein Vertreter der unsupervised learning-Kategorie bildet das HMM, welches ein solch beschriebenes stochastisches Modell bildet [28].

3.2 Markov-Ketten

Ein System, ungeachtet seines Kontexts, bildet dann einen Markov-Prozess ab, wenn es Zustände in der Vergangenheit nicht berücksichtigt, was zur Folge hat, dass auch seine Übergangswahrscheinlichkeiten nicht im Zusammenhang mit vergangenen Zuständen stehen. Zeitgleich muss ein System linear sein, was bedeutet, dass es sich um stationäre Übergangswahrscheinlichkeiten handelt, welche nicht von der Zeit und dem aktuellen Zustand abhängig sind. [29, 30]

Bevor geklärt werden kann, was ein Hidden-Markov-Model ist, wird zunächst darauf eingegangen, was eine Markov-Kette ist. In einer Markov-Kette, welche einen Markov-Prozess abbildet, stehen einige Zustände im Verhältnis zueinander. In diesen Verhältnissen geht es um die Wahrscheinlichkeiten der einzelnen Zustände und deren Übergänge ineinander. Die Zustände werden zu diskreten Zeitpunkten betrachtet und bilden einen zählbaren, und somit endlichen Zustandsraum *S.* Im Kontext dieser Arbeit sind diese Zeitpunkte einmal im Jahr. Es wird also für eine definierte Sequenz an Zeitpunkten $T = \Delta t, 2\Delta t, ...$ ein Modell λ betrachtet, welches zu jedem Zeitpunkt einen diskreten Zustand $S = s1, s2, ...$ einnimmt. Der Übergang von einem Zeitpunkt zum nächsten zieht den Übergang von einem Zustand in den nächsten mit sich. Dieser Zustandswechsel ist abhängig von der Übergangswahrscheinlichkeit *pij u*nd dem aktuellen Zustand *Si* [30]. Ein typisches Beispiel für eine Markov-Kette ist die Abbildung des Wetters. Hier wird angenommen, dass an einem bestimmten Ort definierte Wahrscheinlichkeiten der jeweiligen Wetterzustände existieren. Zum Beispiel scheint zu 50% die Sonne, zu 27% ist es bewölkt und zu 23% regnet es. Des Weiteren sind die Übergangswahrscheinlichkeiten der einzelnen Zustände ebenfalls definiert. Beispielsweise liegt die Wahrscheinlichkeit, dass nach dem Regen die Sonne wieder scheint, bei 70%. Diese Wahrscheinlichkeiten werden auf Basis bisher erfasster Wetterdaten ermittelt. Im Kontext dieser Arbeit wird das das Modell auf die Veränderung von Verbrechen übertragen. Daraus ergeben sich die möglichen Zustände „steigend", "gleichbleibend“ und "fallend", also besagt ein Zustand, ob das Verbrechensaufkommen, im Vergleich zum Vorjahr, verändert ist. Für die genannten Zustände könnte eine denkbare Markov-Kette aussehen, wie in Abbildung 2 visualisiert. Dieselben Beziehungen können auch in einer Tabelle dargestellt werden. Die passende Tabelle zu der Markov-Kette aus Abbildung 2 wurde er- stellt und ist in Tabelle 1 aufgeführt. Zusammenfassend sei also gesagt, dass eine Markov-Kette ein stochastisches Modell ist, um komplexe Zusammenhänge zu simplifizieren und zu beschreiben. Dies wird erreicht, indem die Abhängigkeiten reduziert werden, sodass nur der vorausgehende Zustand und die allgemeinen, beziehungsweise stationären Übergangswahrscheinlichkeiten die Abhängigkeit bilden.

	Up	Same	Down
Up	0.4	0.1	0.5
Same	0.5	0.3	0.2
Down	0.1	0.	0.3

Tabelle 1: Übergangswahrscheinlichkeiten
[Quelle: eigene Abbildung]

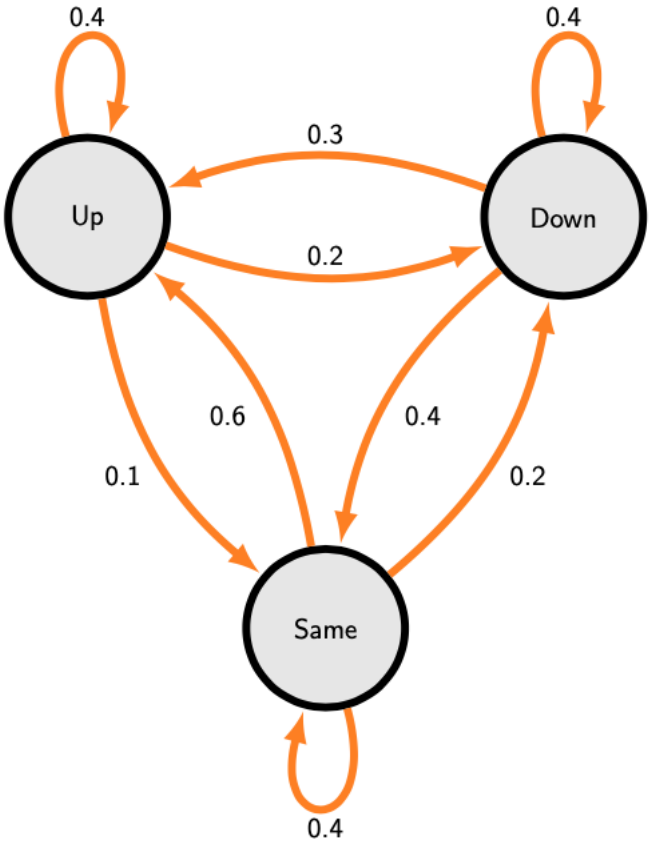

Abbildung 2: Markov-Kette
[Quelle: eigene Abbildung]

3.3 Das Hidden-Markov-Modell

Das Hidden-Markov-Modell beschreibt zwei Zufallsexperimente, das eine sichtbar, das andere unsichtbar. Das HMM setzt sich zusammen aus den verborgenen Zuständen des unsichtbaren Zufallsexperiments, den Beobachtungen, auch sicht- bare Variablen genannt und den Übergangswahrscheinlichkeiten der einzelnen Zustände ineinander. Weitere Bestandteile sind eine Emissionswahrscheinlichkeiten, die angeben, ob es zu jedem Zustand immer eine Beobachtung gibt, und einer Initialisierungswahrscheinlichkeit der verborgenen Zustände. Die verborgenen Zustände sind eine Markov-Kette mit der Besonderheit, dass sie in der Erweiterung unbekannt sind und nur ihre Auswirkungen beobachtbar sind. [31] Das Problem, welches das HMM zu adressieren versucht, ist die Wahrscheinlichkeit der verborgenen Zustände aus gemachten Beobachtungen zu ermitteln und die wahrscheinlichsten Zustände definieren zu können [32]. Ergänzend erfolgt die Optimierung der Parameter, um die Wahrscheinlichkeitsbestimmung anzupassen. Wie schon unter dem Punkt 3.2 erwähnt, sind wieder zwei Voraussetzung für die Modellierung mit dem HMM vorausgesetzt. Zum einen ist der aktuelle, in diesem Falle unsichtbare Zustand allein von dem vorangegangenen Zustand abhängig und zum anderen sind wieder die Wahrscheinlichkeiten, die die Übergänge der Zustände ineinander beschreiben stationär. [31] Im Kontext des Hidden-Markov-Models finden einige Algorithmen Anwendung. Mit dem Forward-Algorithmus wird die Wahrscheinlichkeit, mit der eine bestimmte Beobachtung in dem gegebenen Modell gemacht werden kann, berechnet. Da- bei kann die Beobachtung auch eine Reihe an Beobachtungen sein, also eine Beobachtungssequenz. [33] Gegeben sein müssen für den Algorithmus zum ersten das Hidden-Markov-Modell und zum anderen die zu untersuchende Folge an Beobachtungen. Zuerst werden für alle möglichen Zustände in Kombination mit allen möglichen Beobachtungen die Wahrscheinlichkeiten gebildet und dies von vorne nach hinten, also mit dem ersten Zustand beginnend. Die Wahrscheinlichkeiten werden berechnet durch die Multiplikation der Anfangswahrscheinlichkeiten mit den Emissionswahrscheinlichkeiten der Zustände. Diese Berechnung wird nun für jede mögliche Beobachtung aufgestellt. Damit ist der erste Schritt abgeschlossen. Der nächste Schritt ergibt sich aus der Summe der im ersten Schritt berechneten Ergebnisse, jeweils multipliziert mit der Übergangswahrscheinlichkeit der nächsten Beobachtung. Auch dies wird wieder für jeden Zustand wiederholt. Der zweite Schritt wird nun so lange wiederholt, bis die Werte für jeden Zustand und jede Beobachtung der Sequenz stehen. Alle diese Werte werden daraufhin betrachtet und verglichen und der größte Wert ist dann die wahrscheinlichste Sequenz von Zuständen. Die Programmiermethode, die dieser Algorithmus nutzt, ist die der dynamischen Programmierung. Die dynamische Programmierung beschäftigt sich mit dem Lösen von Optimierungsproblemen. Das klassische Beispiel für ein solches Problem ist das Rucksackproblem, bei dem nur ein begrenztes Volumen und Traglast zur Verfügung steht. Hierbei wird versucht, aus dieser gegebenen Situation das größtmögliche Potential herauszuholen. Auch der Viterbi-Algorithmus gehört zur Kategorie der dynamischen Programmierung. Im Kontext von HMMs wird der Viterbi- Algorithmus genutzt, um effizient die wahrscheinlichste Sequenz der verborgenen Zustände zu einer bekannten Reihe von Beobachtungen zu ermitteln. Der Viterbi-Algorithmus wurde zur Decodierung von Faltungscodes entwickelt und hat seine Anwendungsbereiche in der Fehlerkorrektur bei Signalen in sämtlichen Bereichen der Informatik und Nachrichtentechnik. Der Viterbi-

Algorithmus unterscheidet sich von dem Forward-Algorithmus nur minimal. Er nutzt die gleichen Eingabeparameter wie der Forward-Algorithmus und auch die Schritte sind fast identisch. Der entscheidende Unterschied ist, dass die Summe, die aus den verschiedenen Zustandswahrscheinlichkeiten mit den Übergangswahrscheinlichkeiten multipliziert, besteht, mit dem Maximum ersetzt wird. Daraus resultiert auch seine abweichende Ausgabe der Reihe der wahrscheinlichsten Zustände zu allen Beobachtungen. [33] Neben dem Forward- und dem Viterbi-Algorithmus findet auch der Baum-Welch Algorithmus Verwendung bei dem Training eines Hidden-Markov-Modells. Es wird versucht das Modell so zu justieren, dass das gezielte System besser beschrieben werden kann. Als Ansatz hierzu wird das bestehende Modell mit einer Sequenz an richtigen Beobachtungen optimiert. [33] Da bei einer richtigen, das heißt tatsächlich gemachten, Beobachtung die Wahrscheinlichkeit für eine Übereinstimmung 1 ist, wird auf dieser Grundlage die Optimierung aufgebaut. Der Baum-Welch Algorithmus knüpft an die Berchnungen des Forward Algorithmus an und nutzt die Ergebnisse der einzelnen Schritte. Über diese Werte werden neue Übergangswahrscheinlichkeiten ermittelt. Indem die Übergangswahrscheinlichkeiten auch für die erste Beobachtung gebildet werden, erhält man so auch angepasste Anfangswahrscheinlichkeiten. Die Emissionswahrscheinlichkeiten werden ebenfalls verbessert. Diese Optimie- rungen können nun so lange wiederholt werden, bis die neuen Werte sich von den vorherigen Werten kaum noch unterscheiden. Wenn der Durchgang dann einige Male die Parameter des HMMs angepasst hat, ist das HMM sehr gut auf die Trainingsdaten abgestimmt. Auch hier soll versucht werden, die Trainingsdaten so umfassend wie möglich zu halten, um ein Overfitting durch zu starke Anpassung an die Trainingsdaten zu vermeiden [34].

4 Die Polizeiliche Kriminalstatistik

Das Bundeskriminalamt, kurz auch BKA genannt, erhebt eine Statistik über alle erfassten Kriminalfälle der jeweiligen Jahre, kategorisiert nach verschiedenen Gesichtspunkten. Diese Erhebung hat die erfassten Daten der 16 Landeskriminalämter zur Basis. Die hier genutzten Daten stellt das BKA online zur Verfügung. Die Tabellen mit den gemeldeten und erfassten Verbrechensfällen heißt Polizeiliche Kriminalstatistik, kurz PKS. Für jedes Jahr existieren mehrere Tabellen in unterschiedlichen Formaten. Die erste PKS, die auf der Website des BKAs zur Verfügung steht, ist aus dem Jahre 1953. [35]

4.1 Beschaffung

Über die Website des Bundeskriminalamts können die PKS-Tabellen nach Jahren sortiert ausgewählt werden und über den Unterpunkt der thematischen Gliederung bis hin zu einzelnen PKS Bund-Fallzahlen Tabellen navigiert werden. Hier werden unter „T05 Grundtabelle - Straftaten mit Tatmittel „Internet“-Fälle (V1.0)“ Daten zu Verbrechen mit dem Tatmittel Internet zur Verfügung gestellt. Für jedes Jahr muss die entsprechende Tabelle ausfindig gemacht werden und heruntergeladen werden. Nachdem die Tabelle heruntergeladen ist und lokal vorliegt, kann sie in einem Tabellenkalkulationsprogramm wie beispielsweise Excel oder OpenOffice geöffnet werden. Nun wird nach den relevanten Informationen, den auf Computerkriminalität bezogenen Daten, gesucht. Dies wird erheblich erleichtert durch einen eindeutigen Schlüssel, der zur Identifikation der Daten, die zur Computerkriminalität gehören, dient. Somit muss lediglich nach diesem Schlüssel gesucht werden, woraufhin man dann Zugriff auf die Daten, welche der Computerkriminalität zugehörig sind, erhält.

4.2 Aufbereitung

Die Datentabellen werden entweder im PDF-Format als Tabelle grafisch dargestellt oder in einem XLSX-Format oder CSV-Format als Download zur Verfügung gestellt. Die Daten werden anschließend manuell in das JSON-Format überführt. Danach können die Daten über die VSCode-Extension Thunder Client über die REST-API Endpoints unseres Express-Services in die Datenbank übertragen werden. REST ist eine definierte Art der Kommunikation, die für Maschine- zu-Maschine-Kommunikation entwickelt ist und auf Web-Technologien aufsetzt. [36]

Nach dem Übertragen folgen die Daten dem folgenden Schema:


```
{
        "Year": 1987,
        "CaseCount": 3067,
        "HZ": 5,
        "TrysA": 241,
        "TrysP": 79,
        "DetectionRate": 448,
        "Suspects": 1112,
        "NonGermanSuspectsA": 159,
        "NonGermanSuspectsP": 14.3
}
```


Abbildung 3: Datenschema JSON-Format

[Quelle: eigene Abbildung]

Die Datenbank baut aus den JSON-Dateien dann jene Struktur auf:


```
[
    {
      "_id": "60aaee9d46603e001d5afa03",
      "year": 1987,
      "caseCount": 3067,
      "hZ": 5,
      "triesA": 241,
      "triesP": 7.9,
      "detectionRate": 44.8,
      "suspects": 1112,
      "nonGermanSuspectsA": 159,
      "nonGermanSuspectsP": 14.3,
      "__v": 0
    },
    {
      "_id": "60aaeec746603e001d5afa04",
      "year": 1988,
      "caseCount": 3355,
      "hZ": 5.5,
      "triesA": 343,
      "triesP": 10.2,
      "detectionRate": 43.7,
      "suspects": 1378,
      "nonGermanSuspectsA": 206,
      "nonGermanSuspectsP": 14.9,
      "__v": 0
    },
        ...
]
```


Abbildung 4: Datenbankstruktur

[Quelle: eigene Abbildung]

Nachdem die Daten in der Datenbank sind, können weitere Aufbereitungen, Sortierungen und Filterungen vorgenommen werden.

5 Implementierung

Balzert definiert den Sinn der Implementierung darin, „aus der vorgegebenen technischen Lösung in Form einer Softwarearchitektur ein lauffähiges Softwaresystem durch Programmierung der einzelnen Subsysteme und Komponenten zu erstellen“ [37]. Im Folgenden werden die Aspekte der Entwicklungsumgebung, die verwendeten Technologien, die Architektur und die Realisierung der Implementierung betrachtet. Zudem wird das Vorgehen bei der Umsetzung erläutert.

5.1 Entwicklungsumgebung

Für die Organisation und Verwaltung von Programmier-Projekten ist eine Plattform wie Github optimal geeignet. Mit Github ist es möglich, Features in Boards zu planen, den Quellcode in der Cloud zu speichern und von jedem Gerät drauf zu arbeiten. Zudem kann man automatische Tests über Github laufen lassen. Für die Versionskontrolle wird, wie bei jedem Software-Projekt, Git verwendet. Das Projekt wurde als Repository in Github angelegt. Um den Master-Branch immer auf einem sauberen, das bedeutet funktionierenden Stand zu halten, wird für die Entwicklung mit Feature-Branches gearbeitet. Konkret bedeutet dies, dass jedes Mal, wenn an einer Funktion gearbeitet wird, vom Master-Branch ein neuer Branch abgezweigt und auf diesem gearbeitet wird. Jede Änderung wird dann commitet und auf diesen neuen Feature-Branch gepusht. Wenn die Funktion komplett ist, wird ein Merge-Request in den Master-Branch erstellt und durchgeführt.

Die Implementierung wurde teils auf einem MacBook Pro, teils auf einem Windows Personal Computer vorgenommen. Daher musste die Umgebung plattformunabhängig gehalten werden. Unter Windows wird hauptsächlich aus Gründen der Annehmlichkeit die Windows-Funktion WSL aktiviert. Diese beinhaltet eine der Windows 10 Versionen und installiert das Linux-Subsystem unter Windows. Somit kann als Standart-Shell auf beiden Systemen Fish, die "friendly interactive shell", genutzt werden. Diese Shell ist ein Kommandozeileninterpreter und hat eine Autovervollständigung, einfache Navigation und weitere Funktionen, die den Umgang mit der Kommandozeile intuitiver gestalten. Als Terminal-Emulator, um den Zugriff auf Fish zu ermöglichen, wird Hyper genutzt, was plattformunabhängig funktioniert. Auf diese Weise steht eine gleiche Umgebung unabhängig von dem Betriebssystem zur Verfügung. Dies hat vor allem den Vorteil, dass bekannte Tools genutzt werden können und auf beiden Systemen gleich produktiv gearbeitet werden kann. Zugleich können auf beiden Systemen die gewohnten Vorzüge von Linux und dieselben Befehle genutzt werden.

Als Editor für den Quellcode wurde Visual Studio Code gewählt. Dies ist ein von Microsoft entwickelter Opensource Editor, der in Teilen Merkmale einer integrierten Entwicklungsumgebung aufweist. Somit können zahlreiche für die Softwareentwicklung benötigten Aktionen aus einem Tool heraus ausgeführt werden. VS Code ist ein häufig genutzter Quelltext-Editor, sodass zahlreiche Erweiterungen, sogenannte Extensions, zur

Verfügung stehen. Von diesen Extensions wurde auch in diesem Projekt Gebrauch gemacht. Mit MongoDB for VS Code lässt sich die verwendete MongoDB-Datenbank untersuchen und manipulieren. GitLens hilft durch Vereinfachung der Bedienung von Git bei der Versionskontrolle. API Requests lassen sich mit der Thunder Client unkompliziert und direkt vom Editor aus stellen und mit Docker können die Container innerhalb des Editors organisiert werden. Ventur sorgt für Syntax-Hervorhebung und statische Code-Analyse durch Linting in Vue Dateien. Um die Möglichkeit zu haben, die Services in Docker-Containern laufen zu lassen, wird Docker installiert. In der Windows Umgebung ist dies Docker Desktop für Windows, in der Mac Umgebung wird Docker Desktop für Mac verwendet. Docker Desktop stellt nicht nur die Voraussetzungen für die Virtualisierung und nützliche Tools, wie Docker-Compose, sondern auch eine grafische Oberfläche, mit der die Images verwaltet werden können, zur Verfügung. Für Nuxt.js und Express.js wird weiterhin Node.js, und damit der Paketmanager npm und yarn installiert. Node.js ist eine Laufzeitumgebung, um zu ermöglichen, dass JavaScript auch auf dem verwendeten Betriebssystem laufen kann. JavaScript ist in ihrem Ursprung eine Scriptsprache, die im Webbrowser läuft. Durch Node.js wird JavaScript auch Zugriff auf Betriebssystemfunktionen und auf das Filesystem gewährt. Um einen noch besseren Überblick, eine weitere Hilfe bei dem Debugging und weitere Manipulationsmöglichkeiten zur Verfügung stehen zu haben, wird weiter MongoDB Compass installiert. Als Webbrowser wurde Brave aufgrund einiger Vorteile in Gesichtspunkten wie Performance und unterstützen Technologien genutzt. Da Brave auf Chromium aufgebaut ist, werden alle Industriestandards zuverlässiger als beispielsweise bei Safari oder Firefox unterstützt. Im Browser wurde die Extension Vue DevTools hinzugefügt, um die Nuxt.js Anwendung besser debuggen zu können.

5.2 Technologien

Da die Softwarenentwicklung bereits seit geraumer Zeit besteht, existieren mittlerweile Möglichkeiten zur Automatisierung der Implementierung von Möglichkeiten. Die modernen Programmiersprachen übernehmen beispielsweise einige Probleme wie das Reservieren und Freigeben von Speicher. Darüber hinaus bestehen zahlreiche Libraries und Technologien, auf welche einfach zugegriffen werden kann. Von diesen Technologien wurden bei der Umsetzung in dieser Arbeit mongoDB, Express.js, Nuxt.js, Vue.js, sowie Docker Compose eingesetzt. MongoDB ist eine NoSQL-Datenbank, die die Datensätze in Dokumenten mit JSON-Format speichert. Diese Art der Speicherung erspart Objekttransformationen, da in PUX immer mit JSON-Objekten gearbeitet wird. Darüber hinaus ist MongoDB performant, weitverbreitet, was bei der Benutzung hilft, da auf eine große Nutzerbasis zugegriffen werden kann, und mit einigen unterstützenden Tools versorgt. Es gibt nützliche Erweiterungen für VS Code, den verwendeten Code-Editor, mit Hilfe dessen die Datenbank komfortabel manipuliert, ausgelesen oder gelöscht werden kann. Dies ist gerade in der Entwicklungsphase ein großer Vorteil. Express.js ist ein auf Node.js aufsetzender Webserver. Als UI- Framework wurde auf Vue.js gesetzt, da Vue.js sehr schmal und ohne viel Overhead auskommt. Es vereint ähnlich wie React Styling, Templating und Scripting in einer Datei. Um Vue.js um einige benötigte Funktionen zu erweitern und das Aufsetzten des Projektes zu beschleunigen, wird Nuxt.js genutzt. Um die einzelnen Services in einer unabhängigen und wohl definierten Umgebung laufen lassen zu können, werden die einzelnen Services je als Docker Container organisiert.

Docker ist eine Virtualisierung, welche die einzelnen Umgebungen isoliert voneinander hält. Dennoch ist Docker so konzipiert, dass Kernel-Funktionen mitgenutzt werden. Kernel-Funktionen werden genutzt, um, ohne ein gesamtes Betriebssystem zu nutzen, Funktionen des Kerns eines Betriebssystems zu nutzen. Dies sorgt dafür, dass ein Container dem Verwendungszweck entsprechend klein ist, weil nicht das gesamte Betriebssystem mitgeliefert werden muss. Mit der Installation von Docker Desktop kommen eine Reihe von Tools für Docker. Eines dieser Tools ist Docker Compose und dieses Tool dient der Organisation und Verwaltung mehrerer Container. Über eine YAML-Datei kann eine Reihenfolge definiert werden, die festsetzt welcher Dienst an welcher Stelle startet, Netzwerke konfiguriert werden, welche die Container isolieren oder miteinander verbinden und weitere Konfigurationen vorgenommen werden. Die in der Arbeit eingesetzten Libraries sind buefy, typescript, mary-markov, chart.js, mongoose und cors. Sie fungieren als Sammlungen von Funktionen, welche sich meist auf ein gemeinsames Problemfeld berufen. Buefy ist eine UI-Komponenten-Library und wird verwendet, um eine Vielzahl an fertigen UI-Komponenten bereit zur Verwendung zu haben. Buefy ist jedoch gegenüber Material sehr schlank und für schlichtere Benutzeroberflächen ausreichend. Typescript macht aus Javas- cript eine besser lesbare und organisierter typisierte Script-Sprache, die einige Vorteile bietet. Es ist unter anderem wesentlich leichter, Fehlerquellen und Bugs beim Programmieren zu finden, da Typescript, wie oben genannt bei dem Schreiben schon prüft, ob ein Widerspruch vorliegt und einer Funktion ein Parameter vom falschen Typ übergeben wird. Das npm-Paket, beziehungsweise die Library mary-markov stellt die Funktionen zur Verfügung, die benötigt wer- den, um mit den Markov-Ketten und Modellen zu arbeiten. Weil dem Benutzer die Simulationsergebnisse visualisiert werden sollten, wurde für die Generierung des Diagramms Chart.js verwendet. Chart.js ist eine Schnittstelle, auch Wrapper genannt, für die Library D3.js und macht das Erstellen von Diagrammen und Graphen somit unkomplizierter und schneller. Mongoose ist das das npm-Paket, welches die Verbindung zu der MongoDB aufbaut und die Queries und Manipulationen verarbeitet. Da Abfragen aus dem Frontend an das Application Programming Interface (API), oder andere Services gemacht werden, müssen die CORS-Regeln angepasst werden. CORS ist eine Abkürzung aus dem Webumfeld und steht für Cross-Origin-Ressource-Sharing. Dies soll durch eine Sicherheitsfunktion vermieden werden, um User davor zu schützen, dass von einer Seite Content, wie ein Script, von einer anderen Webadresse ausgeführt wird, als die er aufgerufen hat. Für diesen Usecase gibt es aber auch legitime Gründe, daher muss für diesen Usecase dann eine Ausnahme spezifiziert werden [38]. Um eine solche Ausnahme in dem Express-Server definieren zu können wird die Libary cors genutzt.

5.3 Architektur

Die Architekturen, welche in ihrer Anwendung als am sinnvollsten betrachtet wurden, sind monolithisch, Client Server oder Microservices. Der erste Ansatz war, nur mit einem Frontend zu arbeiten, die Daten in einem In-Memory-Store zu halten und die Business-Logik mit in die Skripte von Vue zu integrieren. Die Software-Architektur von PUX ist eine Microservice-Architektur. Es gibt je einen Container für das Frontend, die Datenbank, das Initialisieren der Daten, den API-Server und das Backend. Microservices bilden einen moderneren, übersichtlicheren Ansatz im Gegensatz zu monolithischen Anwendungen. Mit Microservices lassen sich komplexe Strukturen in zahlreiche kleinere und simpler gestaltete Strukturen aufteilen. Microservice-Architektur wird in der Regel immer dann genutzt, wenn die Anwendung mit dem Problem der Skalierung umgehen muss. Im Falle von PUX ist dies jedoch nicht der Fall. Der Hauptgrund für die Verwendung von Microservice-Architektur war in diesem Fall die Flexibilität in den einzelnen Services. Die einzelnen Services innerhalb der Microservice-Architektur sind einfach austauschbar und variabel hinsichtlich der eingesetzten Technologien und Sprachen. Mit dieser Architektur-Entscheidung kann jeder Service optimal an seine Aufgabe angepasst werden. So müssen, im Gegensatz zu anderen Entscheidungen, keine Kompromisse aufgrund der Programmiersprache gemacht werden.

Docker Hub bietet eine Menge an vordefinierten Docker-Containern, beziehungsweise Docker-images, welche, ähnlich wie Templates bei virtuellen Maschinen, vorgefertigte Umgebungen bereitstellen, auf denen beispielsweise schon Software installiert ist. Mongo ist das offizielle image von MongoDB auf Docker Hub. Mongo wurde im vorliegenden Projekt als Service für die Datenbank gewählt. Mit Hilfe des Services mongo-seed, welcher auf dem offiziellen mongo-image basiert, wird die eigentliche MongoDB mit den gesammelten Daten initialisiert, damit die Daten nicht bei jedem Start der Container-Umgebung erneut eingebunden werden müssen. Node ist das offizielle image, welches Node.js auf DockerHub zur Verfügung stellt. Express ist der Service, welcher auf dem offiziellen node-image basiert. Express.js wurde hierbei als Technologie verwendet, um den Webserver aufzusetzen. Über den Service Express wird eine Verbindung zur Datenbank ermöglicht. Nuxt basiert ebenfalls auf dem offizielle node-image und bildet als Service das Frontend. Nuxt nutzt Nuxt.js als Frontend-Framework. Basierend auf dem offiziellen node-image wurde der letzte Service namens HMM entwickelt, welcher einen Express.js-Server darstellt. Dieser HMM-Service fungiert als Backend, welches die Hidden-Markov-Berechnungen übernimmt.

Die aufgeführten Services, beziehungsweise Docker-Container sind unter Abbildung 3 noch einmal schematisch dargestellt.

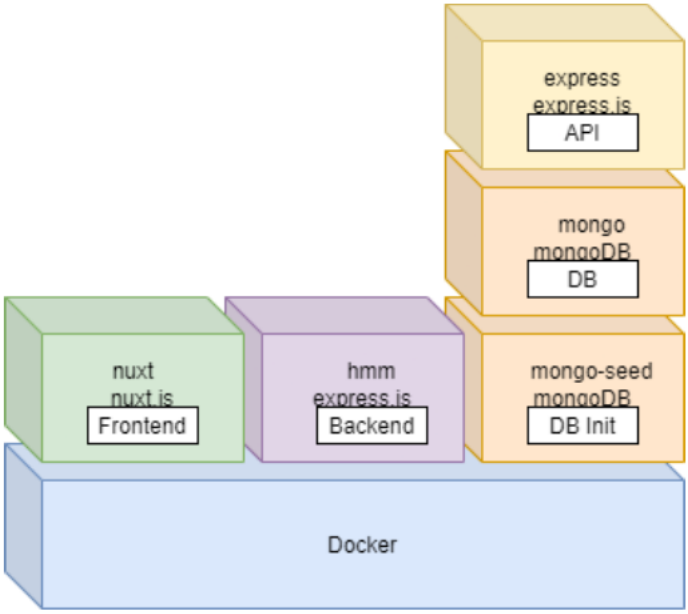

Abbildung 5: Docker-Architektur

[Quelle: eigene Abbildung]

5.4 Realisierung der Services

Zu Beginn wurde festgelegt, dass PUX auf Web-Technologie aufbauen sollte. Dies hat zum Vorteil, dass die Anwendung an verschiedenen Orten laufen kann, in der Cloud, in einem internen Netzwerk oder lokal. In Konsequenz muss ein Endanwender sich nicht um eine Installation kümmern, Updates können automatisch ausgerollt werden und es wird nur ein moderner Browser benötigt, um PUX zu bedienen. Nachdem dies feststand, musste eine Entscheidung bezüglich des zugrundeliegenden Frameworks getroffen werden. Wie in 5.2 schon näher erläutert, wurde sich aus benannten Gründen für Vue entschieden. Bei der Erstellung von Pux wurde mit der Umsetzung des Frontends angefangen. Hier wurde mithilfe der UI-Komponenten-Bibliothek Buefy zunächst das Grundgerüst entworfen. Die Benutzeroberfläche wurde simpel gehalten, eine einfache Navigationsleiste an der linken Seite strukturiert das Programm und in der Mitte der Seite wird der Content in Card-Komponenten angezeigt. Die Anwendung wurde in vier Seiten aufgeteilt. Die Seiten entsprechen den jeweils einzelnen Schritten die nötig sind, um zum HMM zu gelangen und dann mit dem Modell experimentieren zu können. Die erste Seite dient nur als Einstiegspunkt. Es wurde überlegt, wie man einen User bestmöglich an PUX heranführt und ihm die Funktionsweise näherbringt. Dies wurde versucht zu realisieren, indem das Durchführen gameifiziert wurde, und als eine Art kleines Tutorial den User spielend durch die Aufgabe jeder Seite mit einer kleinen Erklärung führt. Das Tutorial ist eine angepasste Stepper-Komponente aus der verwendeten UI-Komponenten-Bibliothek. Die nächste Aufgabe, die im Rahmen der Forschung bewältigt werden musste, war die Erfassung bzw. die Speicherung der Daten, die aus der PKS kommen.

Es wurde zu Beginn angenommen, dass die gesamte Applikation mit nuxt.js umgesetzt werden könnte. Dies stellte sich jedoch als Trugschluss heraus. Schnell zeichnete sich ab, dass die Anwendung komplexer war, und die Anforderungen nicht nur von dem UI-Framework realisiert werden konnten. Das Halten der Datenstruktur und -menge gestaltete sich zunehmend schwieriger und auch hier wurde der Anwendungsbereich des In-Memory-Speichers überschätzt. Daher wurde ein Update auf eine Datenbank vollzogen. Um auf die Daten zugreifen zu können, und weitere hinzufügen zu können, wurde per mongoose eine Express API mit der Datenbank verbunden. Alle anderen Services können somit auf die Daten per REST-Calls zugreifen und weitere Datensätze der Datenbank hinzufügen. Die gesamte Kommunikation der einzelnen Services ist in Abbildung 6 visualisiert. Um nicht jedes Mal beim Hochfahren der Docker Umgebung alle PKS-Daten erneut einspielen zu müssen, wurde ein weiter Service angelegt. Dieser hat die Aufgabe, die Datenbank beim Hochfahren mit den Daten zu initialisieren. Sobald die Database-Page aufgerufen wird, sendet PUX aus dem Frontend einen Aufruf an den express Service, um alle vorhandenen PKS-Daten zu erhalten. Diese werden dann unformatiert im JSON-Format, also exakt so, wie sie in der Datenbank liegen, angezeigt. Durch das Anzeigen der Daten können diese von dem User manuell überprüft werden. Da ein User auch in der Lage sein muss, neue Daten zu den bereits bestehenden Daten hinzuzufügen, ist eine Eingabe implementiert worden. Hierzu wurde auch wieder, um dem User verständlicher zu machen, wie er die Daten einzugeben hat, eine Hilfestellung an die Hand gegeben. Der erste Schritt dafür sind die Placeholder in den einzelnen Feldern, die anzeigen, welche Angaben in welchen Input gehören. Als zweiten Schritt wurden die Input-Felder mit einer Eingabevalidierung ergänzt. Auf die Inputvalidation wird bei der Benutzung noch näher eingegangen. Da nun die Voraussetzungen geschaffen wurden, indem die Daten vorliegen, kann nun mit der Markov-Kette weiter gemacht werden. Die Markov-Kette ist im Kontext dieser Arbeit mit dem Hellfeld gleichzusetzten und wird daher thematisch auf der Brightfield-Page behandelt. Hier wird dem User auf der Karte, die an erster Stelle steht, eine Datenhülle angezeigt, um dem User zu suggerieren, dass hier Handlungen von ihm benötigt werden, um die leeren Stellen zu befüllen. Der Karten-Komplex darunter, bestehend aus zwei Karten, bildet eine Einflussnahme des Benutzers auf das Modell. Auf der ersten Karte werden alle Veränderungen in der Anzahl der Verbrechen zum Vorjahr aus den PKS-Daten aufsteigend sortiert und mit der geringsten Veränderung beginnend, aufgelistet. Das hat zum Vorteil, dass die Auswahl des Benutzers transparenter und nachvollziehbarer wird. Auf der Karte darunter finden sich die Controls zur Manipulation der Werte. Der Anwender bekommt die Option, die Schwellenwerte zu bestimmen, die festlegen, ab wann eine Veränderung als fallend, gleichbleibend oder steigend zu beurteilen ist. Diese Aufgabe wurde zunächst mit zwei Input-Feldern umgesetzt. Dies war simpel in der Umsetzung, jedoch war für einen Anwender nicht erkenntlich, was diese Werte zu bedeuten haben. Daher wurde die Karte von Grund auf erneuert. Die erste Neuerung war, die Werte nicht per Eingabefeld, sondern über einen, beziehungsweise zwei Slider zu setzen. Jeder Slider bekam eine überarbeitete Überschrift, die zum besseren Verständnis beitragen sollte, und um die Übersicht noch weiter zu erhöhen, wurde unter den Slidern eine Grafik eingesetzt, die das Verhältnis der einzelnen Einstufungen farblich anzeigt. Diese Grafik ist so umgesetzt, dass sie sich in Echtzeit aktualisiert. Mit der Funktion, die mit dem Klick auf Go aufgerufen wird, wird, gemäß der vom User eingestellten Schwellenwerte, jedem Datensatz der entsprechende Wert zugeordnet. Anschließend wird über die Daten iteriert und die Vorkommen von je steigend,

gleichbleibend und fallend gezählt und durch die Gesamtanzahl der Datensätze geteilt, um so die Wahrscheinlichkeiten für jeden Wert zu ermitteln. Die Anzahl der jeweiligen Zustände wird in einer Variablen gespeichert. Bei der Iteration wird zusätzlich gezählt, wie oft welcher Zustand folgt. Diese Werte werden nach der Iteration durch die Anzahl der einzelnen Zustände geteilt, um so die Übergangswahrscheinlichkeiten zu erhalten. Diese Wahrscheinlichkeiten werden in Kombination mit den Zuständen und den Initialisierungs-Wahrscheinlichkeiten benötigt, um die Markov-Chain bilden zu können. Sobald alle Berechnungen erfolgreich terminiert sind, werden die Wahrscheinlichkeiten in einer Matrix angezeigt, und die Markov-Kette wird visualisiert und simuliert den Wahrscheinlichkeiten nach Zustandsübergänge. Mit der fertigen Markov-Kette hat ein Anwender nun alle erforderlichen Voraussetzungen geschaffen, um mit der eigentlichen Simulation zu beginnen. Die Seite, die diesem letzten Abschnitt zugeordnet ist, ist die Darkfield-Page. Um die Simulationen bestmöglich erfassen zu können, wurde zunächst Chart.js genutzt, um die Datensätze zu visualisieren. Als Anhaltspunkt sind die realen Daten zum einen in rot und zum anderen als diskretes Liniendiagramm dargestellt, im Gegensatz zu den Balkendiagrammen, mit denen die simulierten Daten angezeigt werden. Das Hidden-Markov-Modell kann frei angepasst werden. Im weiteren Verlauf der Implementierung, wie oben angedeutet, stellte sich heraus, dass für die Umsetzung des Hidden-Markov-Modells noch ein eigener Service benötigt wird. Als naheliegende Annahme wurde als Libary für diese Aufgabe zunächst an Tensorflow gedacht, im speziellen an Tensorflow.probabilistics. Da diese Bibliothek allerdings nicht mehr erhalten ist, musste nach einer Alternative gesucht werden, welche gefunden wurde. Nach etwas Recherche wurde das npm-Paket ‚mary-markov' gefunden. Eine der größten Schwierigkeiten bei der Umsetztung war es, die zusammengesetzten Observer aus den Observergrößen "Versuche", Äufklärungs Quote", "HZ" und "Verdächtige" zu bilden. Die letztliche Realisierung wurde über den Durchschnitt aus den einzelnen Wahrscheinlichkeiten der jeweiligen Observergrößen errechnet. Zur Veranschaulichung wird einmal eine exemplarische Rechnung aufgeführt. Angenommen, die Observergröße „Versuche" ist [0.5; 0.2; 0.3] und die Observergröße „Aufklärungsquote" ist [0.2; 0.4; 0.4] dann bildet sich die zusammengesetzte Observergröße „VersucheAufklärungsquote" wie folgt: [(0.5 + 0.2) / 2; (0.2 + 0.4) / 2; (0.3 + 0.4) / 2] daraus ergibt sich die Größe [0.35; 0.3; 0.35]. Mithilfe der oben genannten, getroffenen Entscheidungen wurde somit versucht, eine optimale Benutzerfreundlichkeit und Funktionsweise für PUX zu erlangen.

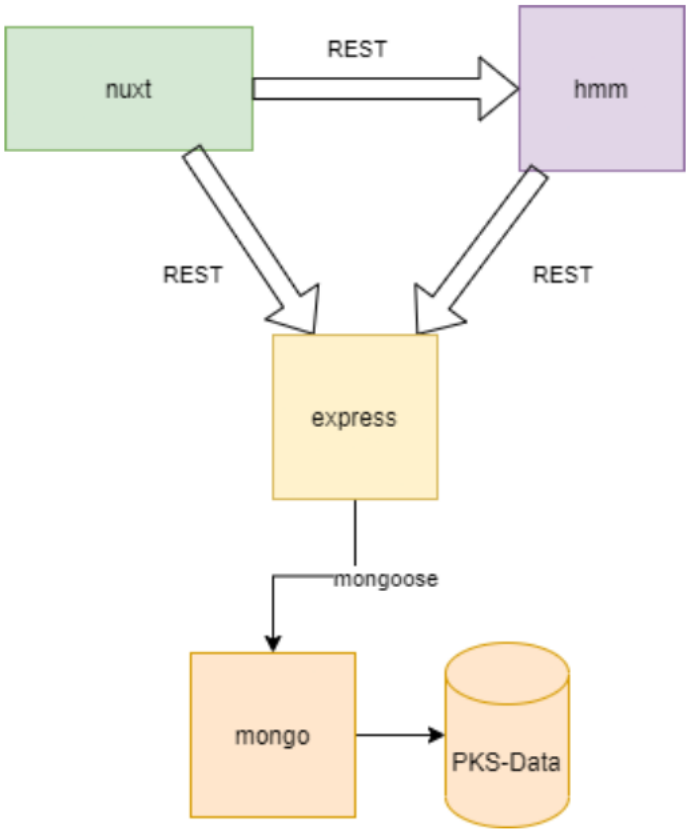

Abbildung 6: Service-Kommunikation

[Quelle: eigene Abbildung]

5.5 Nutzung

PUX setzt sich aus vier Hauptseiten zusammen. Bevor jeweils auf die Nutzung jeder Seite im Detail eingegangen wird, wird im Folgenden zunächst die Vorbereitung zur Nutzung von PUX erwähnt. Im Anschluss daran werden die Homepage, die Database-Page, die Brightfield-Page und die Darkfield-Page näher erläutert.

5.5.1 Vorbereitung

Um mit PUX arbeiten zu können, müssen die benötigten Dateien heruntergeladen, und einige Tools vorinstalliert werden. Diese Dateien können zum einen auf Github als Zip-Archiv oder mit Hilfe von git heruntergeladen werden. Nachdem der Download abgeschlossen ist, muss, sofern die Entscheidung auf das Zip-Archiv gefallen ist, dieses zunächst entpackt werden. Anschließend wird über ein Terminal-Emulator, wie der Powershell auf Windows-Systemen oder dem Terminal auf MacOS-Systemen in das gerade heruntergeladene Verzeichnis navigiert. Dies geht mit dem ‚cd'-command welches für change directory steht. Sofern man sich im richtigen Verzeichnis befindet und alle benötigten Tools heruntergeladen sind, kann

mit dem Befehl ‚docker compose build' die Anwendung gebaut werden. Nachdem dieses Kommando erfolgreich abgeschlossen ist, können mit dem nächsten Befehl ‚docker compose up' die einzelnen Container hochgefahren werden. Der Zusammenschluss der einzelnen Container bzw. Services bildet in seiner Gesamtheit PUX, und ist somit dem Starten von PUX gleichzusetzen. Nun kann also auf PUX zugegriffen werden.

5.5.2 Homepage

Um das User Interface von PUX aufzurufen, wird ein Browser der Wahl gestartet, oben in das Adressfeld die URL ‚http://localhost:8080' eingegeben und mit der Enter-Taste aufgerufen. Daraufhin eröffnet sich dem User die Landingpage oder Homepage. Wie auf der Abbildung 7 zu sehen, kann von dieser Seite aus zum einen über die links befindliche Navigation zu jeder anderen Seite navigiert werden, und zum anderen unter dem Logo ein kleines Tutorial durchgeklickt werden, welches eine kurze Beschreibung zu jeder Seite enthält und eine zusätzliche Option der Navigation bietet. Wie die Abbildung 7 zeigt, kann man das Tutorial mit den Next- und Previous-Buttons durchklicken. Ergänzend kann der Benutzer aber auch auf die einzelnen Steps klicken, um direkt die Informationen zur gewählten Seite zu erhalten.

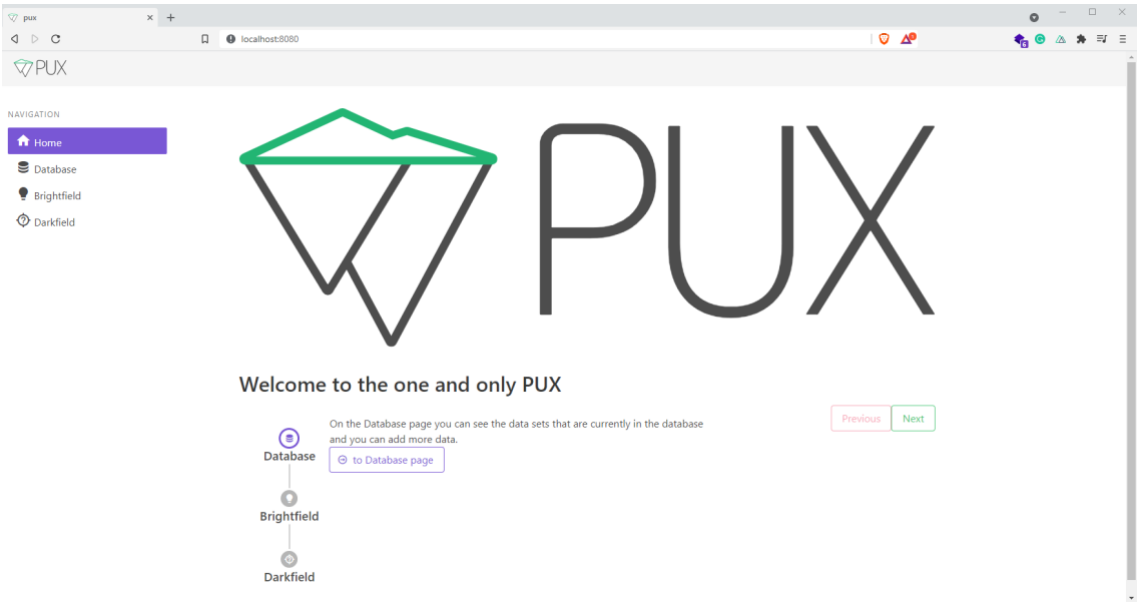

Abbildung 7: Homepage
[Quelle: eigene Abbildung]

5.5.3 Database-Page

Auf der thematisch nächsten Seite, der Database-Page, werden auf der ersten Karte alle Datensätze, die von der Polizeilichen Kriminalstatistik in die Datenbank übertragen wurden, wie in Abbildung 8 erkennbar, aufgelistet. In der Karte darunter hat der Benutzer die Möglichkeit, noch weitere Datensätze der Datenbank hinzuzufügen. Hierzu stehen dem User

eine Reihe an Input-Feldern zur Verfügung. Um die Benutzerfreundlichkeit zu erhöhen, wurde eine Inputvalidation eingesetzt, die den User auf nicht ausführbare oder unvollständige Eingaben hinweist. Diese prüft die Eingaben auf erwarteten Typ und auf Plausibilität. Wenn ein Benutzer beispielsweise wie in Abbildung 9 zu sehen, in das Feld ‚Year' ‚E' eingeben würde, würde die Inputvalidation bemerken, dass die Eingabe nicht vom erwartetem Typ Zahl ist, und dem User einen Hinweis anzeigen, der klar aussagt, dass nur Zahlen eine gültige Eingabe sind, wie ebenfalls auf der Abbildung zu erkennen. Wenn für das Jahr ‚2' eingegeben würde, würde die Inputvalidation wieder erkennen, dass die Eingabe vermutlich ein Versehen des Users war und ihn darauf hinweisen, dass die getätigte Eingabe zu gering ist. Als gültige Werte wurde eine Spanne, wie auf der Abbildung 10 zu lesen, zwischen 1800 und 9999 festgelegt, da PKS-Daten vermutlich nicht zuvor Relevanz haben, und 9999 als Grenze, da es unwahrscheinlich ist, dass die gleichen Regeln in einer so weit entfernten Zukunft gelten und daher angenommen werden kann, dass der User sich bei den Angaben vertippt hat. Wenn nun alle Eingabe-Felder ausgefüllt sind, und die Inputvalidierung keine Unstimmigkeiten feststellt, kann der eingetragene Datensatz über den Save-Button in der Datenbank abgespeichert werden.

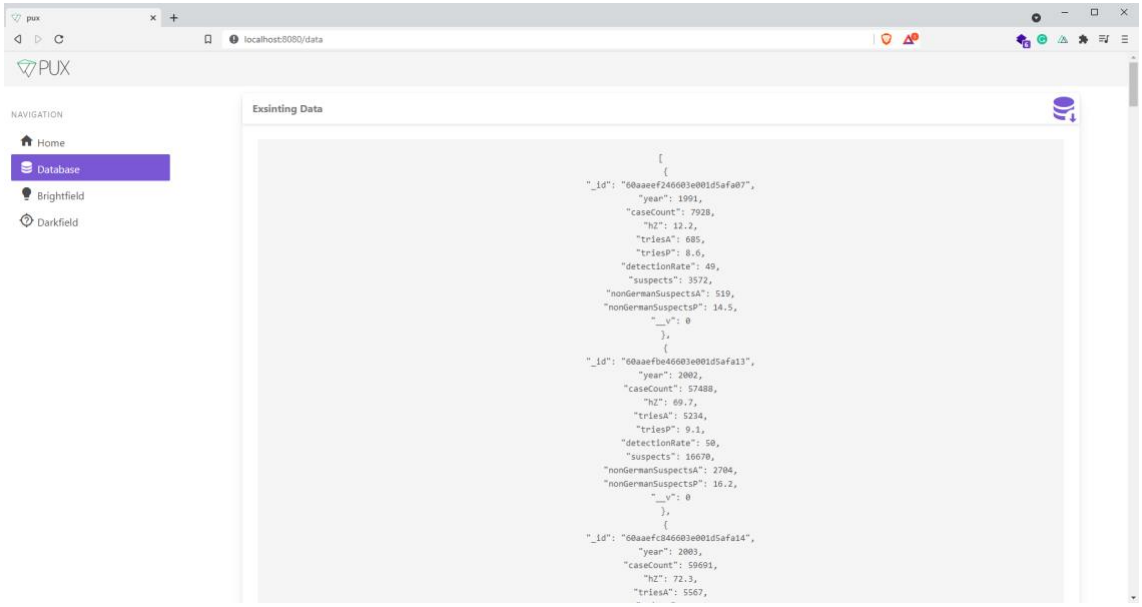

Abbildung 8: Database-Page mit vorhandenen Daten

[Quelle: eigene Abbildung]

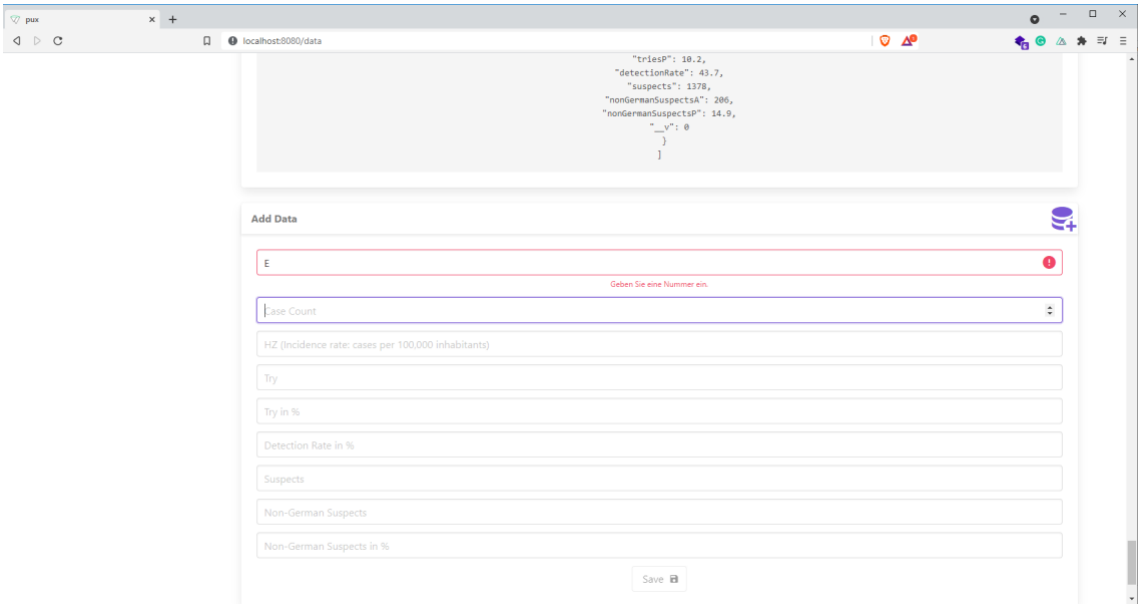

Abbildung 9: Database-Page Inputvalidation falscher Typ

[Quelle: eigene Abbildung]

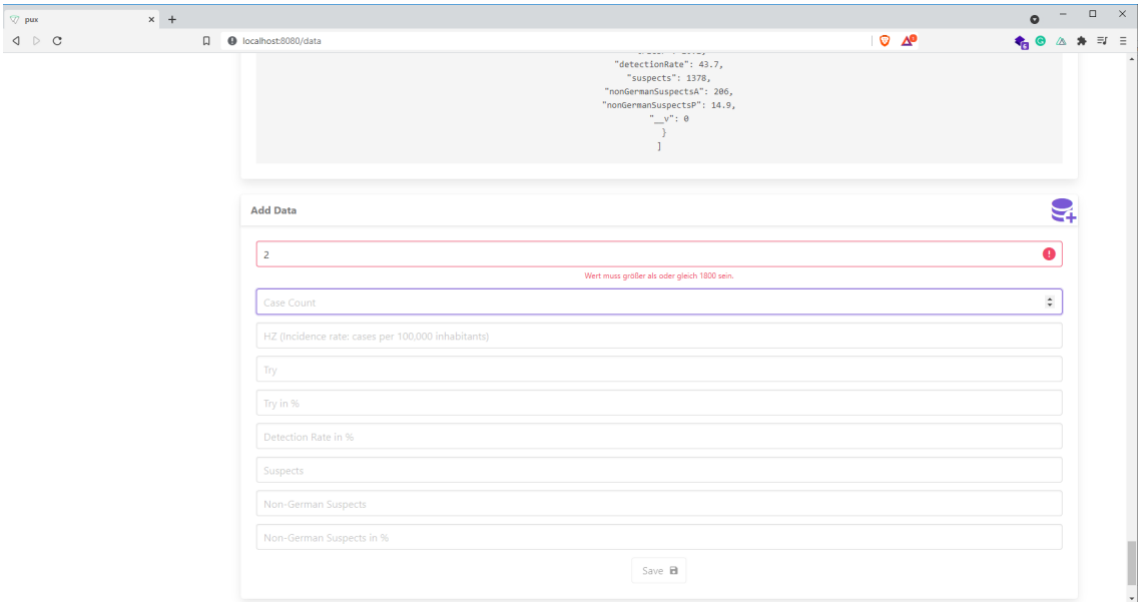

Abbildung 10: Database-Page Inputvalidation Out Of Range

[Quelle: eigene Abbildung]

5.5.4 Brightfield-Page

Nachdem die gewünschten Datensätze der PKS in die Datenbank übertragen sind, kann auf der Brightfield-Page, welche über die Navigation angesteuert werden kann, das Hellfeld mit einer Markov-Kette modelliert werden. Dafür legt der User fest, wie er seine Trend-Thresholds haben möchte und setzt dementsprechend die Slider auf den gewünschten Wert. Beim Verschieben wird der aktuelle Wert sichtbar, und für eine bessere Übersicht wird der Wert zusätzlich links des jeweiligen Sliders dauerhaft angezeigt. Der obere Slider in Abblidung 11 dient dabei dem Festlegen des unteren Thresholds, der angibt bis zu welchem Wert das Modell einen Wert als fallend zu betrachten hat. Der untere Slider legt fest bis, beziehungsweise ab welchem Wert die Zahl als gleichbleibend, beziehungsweise steigend betrachten werden soll. Um den User bei seiner Entscheidung zu unterstützen, werden über der Karte mit den Slidern noch einmal alle absoluten Werte aufsteigend sortiert angezeigt. Dies ist in der Abbildung 11 ebenfalls erkennbar. Unter den beiden Slidern aktualisiert sich noch eine Grafik in Echtzeit, damit das Verhältnis der drei Bereiche zueinander auf einen Blick erfasst werden kann. Mit den drei Bereichen sind die drei Einstufungen gemeint, ob ein Wert als fallend, gleichbleibend oder steigend zu bewerten ist. Die Slider werden voreingestellt mit einer Verteilung von etwa 25:50:25, da mit dieser Einteilung die besten Ergebnisse erzielt wurden. Die gesetzten Werte werden mit dem grünen Go-Button, unten in der Abbildung 11, bestätigt, und daraufhin wird eine Liste mit den so umgesetzten Trends angezeigt und die Markov-Kette gebildet. Diese wird daraufhin in einer neuen Karte, die sich oben als erste neu einfügt, visualisiert und darunter wird die Kette noch einmal in Zahlen dargestellt. Die Visualisierung, wie sie auch in Abbildung 12 zu erkennen ist, ist eine Animation, die den Wahrscheinlichkeiten nach dem Zustand wechselt, was mit dem Kreis dargestellt wird. Wenn eine Anpassung der Thresholds vom User gewünscht wird, ist dies jederzeit mit den ihm bekannten Schritten problemlos möglich. Er muss dafür die Slider auf die neuen Werte schieben und erneut mit dem Go-Button bestätigen.

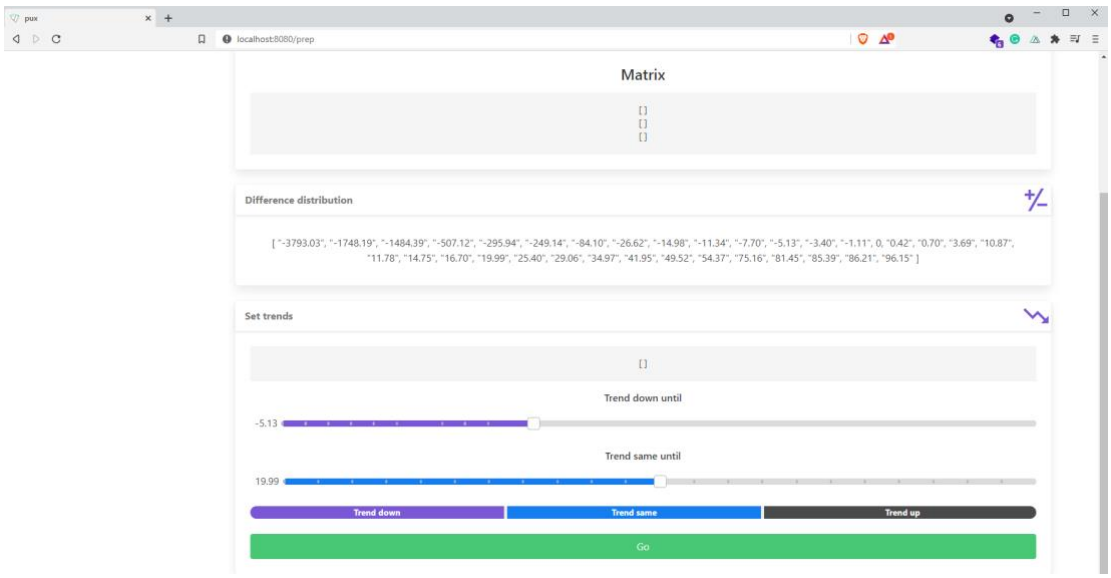

Abbildung 11: Brightfield Page

[Quelle: eigene Abbildung]

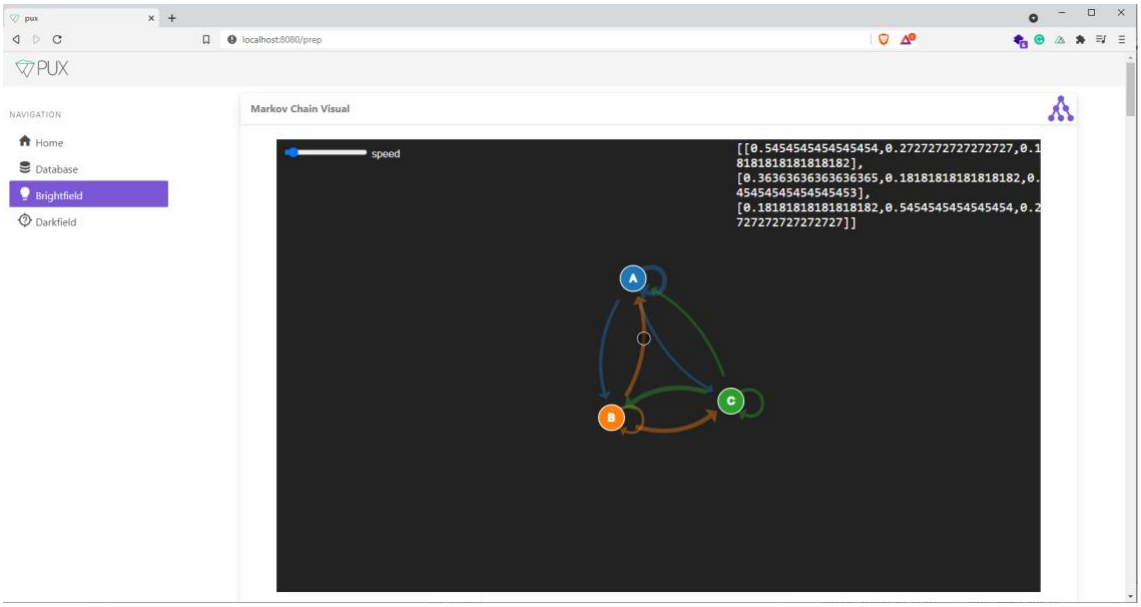

Abbildung 12: Visualisierung Markov-Chain

[Quelle: eigene Abbildung]

5.5.5 Darkfield-Page

Wenn die Darkfield-Page aufgerufen wird, bevor über die Brightfield-Page die Schwellenwerte gesetzt wurden, ist die Seite leer und weist den Benutzer darauf hin, erst die Brightfield-Page zu besuchen und dort die erforderlichen Werte zu setzen. Die Seite sieht dann aus wie in Abbildung 13. Wenn aber die Darkfield-Page aufgerufen wird, nachdem die Thresholds gesetzt wurden, bekommt der User wie bisher gewohnt eine Seite mit Karten. Auf der ersten Karte kann der User die realen Daten der PKS mit den gesetzten Thresholds in einem Diagramm betrachten. Das Diagramm ist in Abbildung 14 in rot zu erkennen. Auf der Y-Achse kann der Graph diskret drei Werte annehmen. Eins steht für einen Anstieg zum Vorjahr, Null für ein Gleichbleiben und minus Eins für einen Abstieg der Verbrechensrate im Deliktbereich Cybercrime. Die X-Achse zeigt an, in welchem Jahr die Veränderung liegt. In der darauffolgenden Karte wird noch einmal die Matrix des aktuellen Modells angezeigt und es stehen zwei Buttons zur Verfügung. Der erste Button mit dem Text „Reset“ setzt das Modell auf das Modell zurück, mit welchem man gestartet hat. Der zweite Button mit dem Text „Build“ erstellt ein neues Modell aufgrund der ausgewählten Parameter auf der Seite. Auf der dritten Karte können die bisher gebauten Modelle miteinander auf Basis ihrer Präzision verglichen werden. Die Präzision liegt zwischen null und eins und sagt aus, wie viele Vorhersagen mit den Realdaten übereinstimmen. Das nächste Control auf der Seite ist ein Slider, mit dem eingestellt werden kann, welche Jahre, beziehungsweise welche Jahresdaten gewählt werden sollen, um das Modell zu bauen. Wie aus 3.3 bekannt, werden für ein HMM auch Observer benötigt. Hinter den vier Checkboxen verbirgt sich, welche Observer betrachtet werden sollen. Dabei ist es möglich, jede Kombination der Observables zu wählen. Standardmäßig sind alle viel Möglichen vorausgewählt. Die letzte Karte auf der Darkfield-Page

bietet die Option, eine Reihe von Observern zu wählen, und daraufhin die Wahrscheinlichkeiten für diese Sequenz zurück geliefert zu bekommen. Die Observables, die hierbei ausgewählt werden können, hängen direkt von den für das Modell gewählten Observern ab. Dies bedeutet, dass falls nur die Versuche betrachtet werden sollen, nur die Observer für die Versuche der Sequenz angefügt werden können. Je mehr Observables für das Modell gewählt werden, desto komplexer werden die Observables. Da das HMM nur eine Beobachtung in einem Zeitpunkt zulässt, es aber mehre Beobachtungsgrößen gibt, müssen diese in einem Observable zusammengefasst werden.

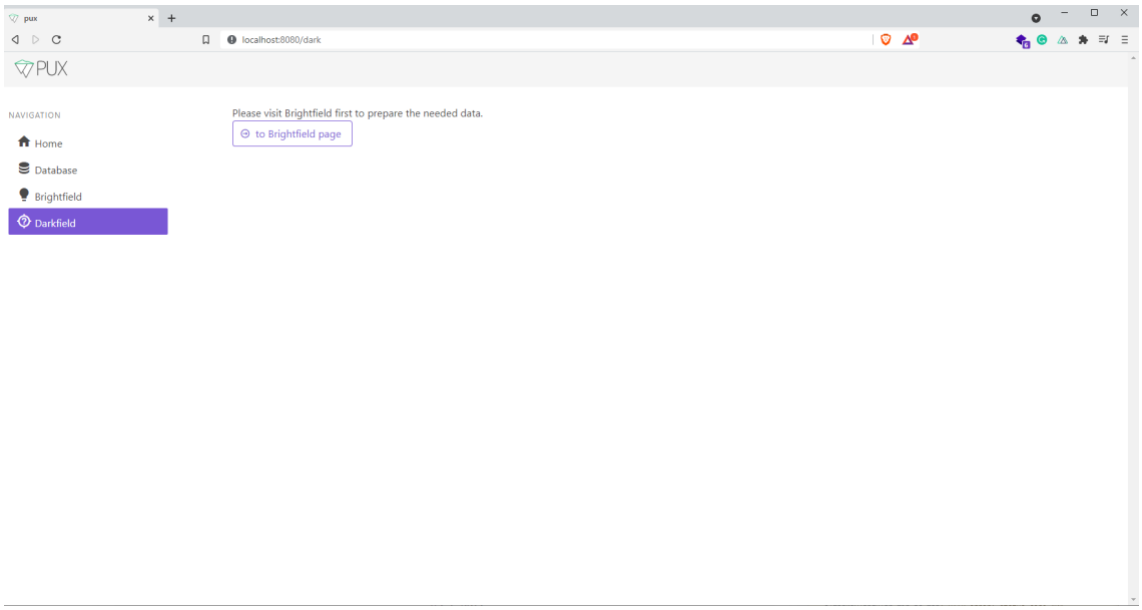

Abbildung 13: Darkfield-Page leer

[Quelle: eigene Abbildung]

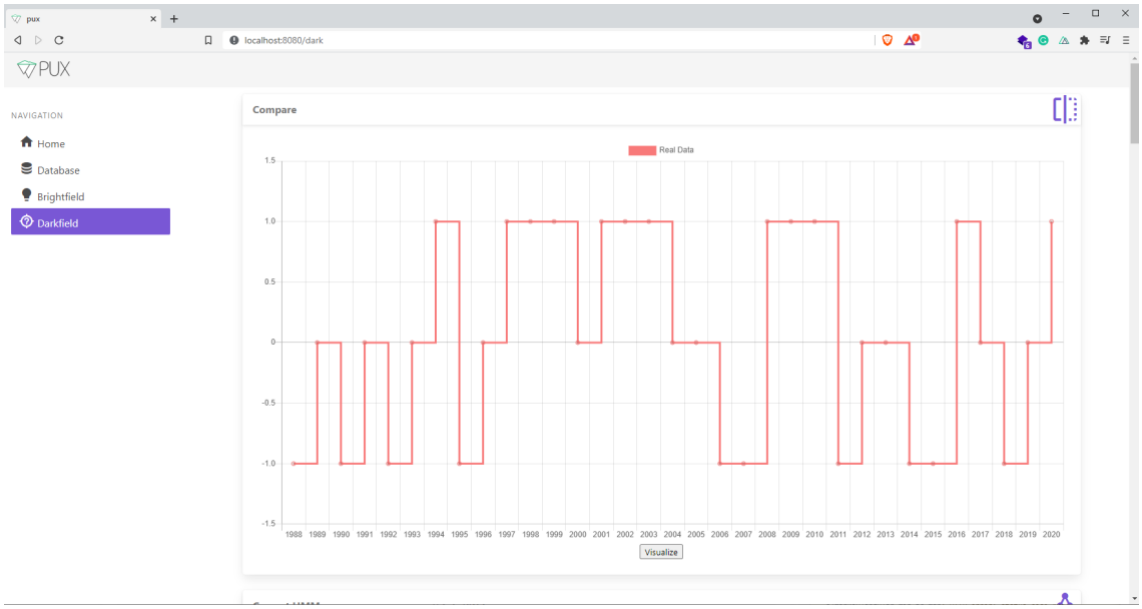

Abbildung 14: Darkfield-Page Graph

[Quelle: eigene Abbildung]

6 Auswertung

Die PKS als Datenbasis ist eine sehr umfassende Sammlung an Verbrechensdaten. Was die Arbeit mit der PKS erheblich unkomfortabel macht, ist der Zustand in dem die PKS sich befindet. Die Daten sind inkonsistent erfasst, mal als Tabelle mit den Zusatzangaben und mal mit anderen. Die Tabellen sind auch nicht immer gleich aufgebaut. An den Daten der PKS werden nachträglich auch immer wieder Änderungen vorgenommen, so dass aktuell zu bleiben eine Herausforderung ist. Des Weiteren umfasst die PKS nur das Hellfeld. Das Dunkelfeld ist leider nur schwer greifbar und einige Verbrechen bleiben unentdeckt, oder kommen nicht zur Anzeige aus den verschiedensten Gründen. Mittels Dunkelfeldforschung erhellt man die Wissenslücken und kann somit auch mehr über die Zusammenhänge der Kriminalität verstehen. Wenn dieses Modell und insbesondere die verwendeten Daten betrachtet werden, wird deutlich, dass die Datenbasis nicht diversifiziert ist. In Faktoren darüber hinaus stehen für den Deliktbereich Cyberkriminalität nur eine begrenzte Anzahl an Daten zur Verfügung. Dies resultiert aus der Tatsache, dass dieser Deliktbereich noch sehr jung ist. Wirklich verwertbare Datensätze wurden prinzipiell erst seit den 1990er Jahren erhoben, was dazu führt, dass weniger als 40 Datensätze als Basis für die Modellierung herangezogen werden können. Auch sind in diesem Modell keinerlei anderer Faktoren berücksichtigt worden, da die polizeiliche Kriminalstatistik nicht alle Faktoren, welche zu Verbrechen führen oder dazu beitragen, betrachtet. Beispiele für solche Faktoren sind für den Deliktbereich Cybercrime etwa der Grad der Digitalisierung oder wirtschaftliche Aspekte oder auch gesetzliche Regelungen. Eine der größten Herausforderungen stellte das Kombinieren der Observergrößen zu einem Observer dar. Die gewählten Observergrößen „Trys“, „Aufklärungs-Quote“, „HZ“, „Suspects“ können jeweils 3 Zustände einnehmen, ähnlich wie die Zustände von dem HMM. Es wird mit dem Vorjahr verglichen und daraus ergeben sich die drei Zustände steigend, gleichbleibend und fallend. Nun wird für jede Größe über die Datenbank iteriert und gezählt wie viele Zustände vorkommen. Daraus erben sich Aussagen wie es gibt fünfmal den Zustand „steigend“ für die Größe „Try“. Anschließend wird erneut über die Datenbank iteriert und gezählt, wie oft diese Zustände vorkommen, wenn Das System einen bestimmten Zustand hat. Daraus kann abgeleitet werden, dass beispielsweise zweimal bei dem Zustand „Gleichbleibend“ die Observergröße den Zustand „steigend“ hat. Aus diesen Werten können daraus dann die Wahrscheinlichkeiten für die einzelnen Observergrößen berechnet werden. Wenn auf den steigenden Zustand zwei steigende „Try“-Größen kommen, und es insgesamt fünfmal vorkommt, dass „Try“ steigend ist, dann ist die Wahrscheinlichkeit für steigende Versuche bei steigendem System 2/5. Auf diese Weise müssen als Bedingung alle Wahrscheinlichkeiten ermittelt werden. Daraus resultieren bei vier Größen mit je drei Zuständen und variablen Größen, das heißt, dass die Größen nach Belieben betrachtet oder ignoriert werden können, über 200 mögliche Observer. Eine Observer-Gruppe unterliegt einigen Regeln, denen sie folgen muss. Eine dieser Regeln ist, dass die Summe aller Observer 1 ergeben muss, denn die Wahrscheinlichkeit aller Beobachtungen, die gemacht werden können, für einen Zustand, sind zu 100% alles was sein kann. Unter diesen Regeln und mit dieser Menge an Observern zu arbeiten und hierbei keine Fehler zu machen ist wahrlich eine Herausforderung. Wie die verschiedenen Observergrößen zum einem Observer kombiniert werden, wurde unter der Berücksichtigung, dass wenn eine Observergröße sehr

wahrscheinlich oder sehr unwahrscheinlich ist, diese Wahrscheinlichkeit ins Gewicht fallen solle, der Mittelwert der Größen gebildet. Damit löst man zwei Probleme auf einmal, zum Ersten bleiben die Verhältnisse erhalten und die Summe der Observer bleibt bei 1 und die Wahrscheinlichkeiten behalten ihren Wert. Immer wieder im Laufe der Entwicklung schlichen sich Bugs ein, die mittels Debuggen lokalisiert und dann gefixt werden mussten. Ein Bug sorgte dafür, dass die Wahrscheinlichkeiten für die gleichbleibenden Observer immer Null waren. Dies lag daran, dass bei dem Key ein Tippfehler entstanden war, welcher eine richtige Berechnung verhinderte. Daraus resultierten Simulationen die ausgaben, dass alle Zustände immer steigend oder fallend waren, sodass dieses Problem behoben werden musste. Nachdem das Modell die größten Schwierigkeiten überwunden hatte und der Hauptteil der Programmierung abgeschlossen war, lagen die Simulationsdaten sehr weit von den Echten Daten entfernt. Die Präzision lag etwa bei 0.3, das bedeutet das Modell hatte eine ähnlich hohe Trefferquote, wie diejenige, die man erreicht hätte, sofern man geraten hätte. Durch die Justierung der Parameter ließ sich die Präzision in der vorhandenen Zeit bis auf 60 % steigern.

Die Frage dieser Arbeit, die versucht wurde zu beantworten, in dem eine Anwendung entwickelt wurde, die als Tool zur Untersuchung der PKS-Daten eingesetzt wurde, ist zum Stand der Abgabe nicht eindeutig mit Ja oder Nein zu beantworten. Mit der Simulation wurde das gewünschte Ergebnis nur zu Teilen erreicht. Die Trefferquote für das Modell ist so hoch, dass man nicht mehr von Zufall sprechen kann, aber noch nicht so hoch, dass man mit Sicherheit sagen kann, dass die Kriminalität im Deliktbereich Cybercrime ein Markov-Prozess ist. Die 60%ige Präzision lässt die Vermutung zu, dass die Kriminalität stochastischen Gesetzmäßigkeiten folgt, aber die wurde nicht mit Sicherheit bestätigt. Um hier eine präzisiere Aussage treffen zu können, sind zum einen weitere Verbesserungen am Programm notwendig, und zum anderen ist es genauso wichtig eine größere Menge Daten zu haben. Hierbei muss allerdings beachtet werden, dass die nächsten 2 Jahre stark durch Covid-19 geprägt sein werden und hierdurch die Aussagekraft vermutlich zunächst verringert wird. Bei der Betrachtung der besten Beobachter hat sich herauskristallisiert, dass die Kombination aller Observergrößen die besten Ergebnisse liefert. Dies kann aber nach der Korrektur des Modells noch einmal genauer untersucht werden. Dennoch ist PUX eine benutzerfreundliche und performante Anwendung, die durch ihre Architektur dem User offen hält, welche Plattform er nutzen möchte, indem PUX komfortabel im Webbrowser bedient werden kann. Bei der Entwicklung von PUX wurde versucht die Bedienbarkeit so angenehm wie möglich zu gestalten. Bei intensiverer Nutzung der Software sind jedoch noch einige Verbesserungsideen aufgekommen. Mit mehr Zeit könnten die ein oder anderen Probleme bestimmt noch minimiert oder ganz behoben werden.

7 Ausblick

Um das Projekt abzuschließen, wird der Fokus auf die Zukunft von PUX gerichtet und es werden mögliche Anwendungsgebiete für die Software beleuchtet. Zudem werden Wege betrachtet, die weitere Entwicklungen und Potentiale aufzeigen.

7.1 Anwendungsgebiete von PUX

Im jetzigen Zustand ist PUX schon als ein Hilfsmittel zu gebrauchen, um ein Gefühl für die Kriminalität zu bekommen, indem man die Daten erkundet und mit den variablen Parametern experimentiert. Zum Zeitpunkt der Abgabe entspricht PUX aufgrund der begrenzten Zeit noch nicht allen ermittelten Anforderungen. Mit PUX soll ein Benutzer fähig sein, Kriminalität besser zu verstehen und Hypothesen an das Modell stellen zu können, deren Bestätigung zumindest erahnt werden kann. Durch das Arbeiten mit PUX und die Simulationen können Erkenntnisse über Verbindungen in der Kriminalität gewonnen werden, die wiederum helfen können, die Verbrechensbekämpfung anzupassen und zu verbessern. Mit einigen Erweiterungen ist es denkbar, dass PUX auch für weitere Deliktbereiche oder sogar ganz andere Bereiche verwendet werden kann. Insbesondere Instanzen mit Datenanalystenhintergrund, die die strategische Verbrechensbekämpfung leitet, könnt durch den Einsatz von einem Tool wie PUX einen Vorteil ziehen. Zusammenfassen kann gesagt werden, dass PUX einen eher akademischen Nutzen hat, als aktiv bei der Verbrechensbekämpfung eingesetzt zu werden, jedoch sind die Erkenntnisse, welche durch die Forschung mit PUX gewonnen werden können, wichtige Bestandteile, die nach einer Auswertung zur Verbrechensbekämpfung beitragen können.

7.2 Zukünftige Erweiterungsmöglichkeiten

In der Software-Entwicklung ist es unüblich, von einem definierten Punkt zu sprechen, an welchem ein Produkt „fertig“ ist. Es existieren in den meisten Fällen immer noch Funktionen, um welche das Produkt erweitert werden kann. Hierzu gehören auch Performance, die gesteigert werden kann, Fixes von Problemen oder Unschönheiten, wie Tippfehler oder Ähnliches und Verbesserungen im Bereich der Sicherheit oder Library-Updates, damit die Ausführung auch auf aktuellen Systemen möglich bleibt. PUX ist hier keinesfalls eine Ausnahme. Bei der Konzeption, der Realisierung und der Nutzung sind immer wieder solche Möglichkeiten für Verbesserungen aufgefallen und zum Teil auch angewandt worden und sollen im Folgenden näher erläutert werden. Beim Benutzen von PUX ist aufgefallen, dass der Wechsel zwischen den beiden Field-Seiten lästig ist, und das Experimentieren hindert. Um dieses Problem zu lösen, wäre es möglich die Trend-Controls mit auf die Darkfield-Page zu übernehmen. Ergänzend hierzu wäre es hilfreich, wenn die Slider den Wert behalten würden, und nicht immer wieder zurückgesetzt würden. Das Potential dieser Anwendung ist noch nicht ausgeschöpft, und vermutlich auch noch nicht vollständig überblickt. Mit einigen Ausblicken soll dieses Potential aber versucht werden, zu verdeutlichen. Die Usability und Accsessibility ist

mit Sicherheit noch ein Part, an welchem Verbesserungen vorgenommen werden könne. Für den Fall, dass Pux in unterschiedlichen Regionen eingesetzt werden sollte, könnte man die Texte der Benutzeroberfläche lokalisieren und zusätzliche Sprachen außer Englisch anbieten. Auch ist es denkbar, die Modelle noch weiter zu visualisieren und auch den Verlauf der Verbrechensentwicklung unter Berücksichtigung der Veränderung der einzelnen Parameter, grafisch anzuzeigen. Einfache Weiterentwicklungen von PUX würden eine bessere Interaktion mit der Datenbank beinhalten. Das Hinzufügen von mehreren Datensätzen könnte ergänzt werden. Die Datenbank-Seite hält außerdem die Möglichkeit inne, um eine Suche erweitert zu werden, über welche gleichzeitig die Option, den Datensatz zu verändern oder zu löschen, vorhanden wäre. Damit wäre die Anzeige aller Daten der Datenbank auch nicht mehr in ihrer Gänze notwending, sondern könnte mit einer schmaleren Grafik arbeiten, die beispielsweise nur anzeigt, für welche Jahre die Daten bereits existieren. PUX könnte zudem um einen Darkmode erweitert werden, der das Arbeiten auch in dunkleren Konditionen oder zu späteren Zeiten angenehmer gestalten würde. Um den Prozess des Einspielens der Daten zu vereinfachen, wäre es möglich, einen Import der PKS Tabellen aus beispielsweise dem CSV-Format zu implementieren. Dies hätte unter anderem den Vorteil, dass der Schritt zu einer automatischen Aktualisierung naheliegender wäre. Beim Starten der Anwendung könnte PUX somit über ein Art Crawler auf der BKA-Website suchen, ob eine neue PKS-Tabelle zur Verfügung gestellt wurde, diese herunterladen, die Daten in die Datenbank importieren und das Modell aktualisieren. Eine weitere denkbare Erweiterung von PUX wäre die Erweiterung auf mehrere Deliktbereiche. In diesem Falle wäre es denkbar, dass über eine Auswahl ein Deliktbereich gewählt werden könnte, und alle Seiten von PUX dann in diesem Kontext arbeiten würden. Diese Auswahl würde vermutlich zu einer Einstellung werden, die jederzeit von dem Benutzer angepasst werden könnte. Eventuell wäre es sogar möglich, die Suche nach den besten Observern zu automatisieren, in dem man die Simulationen automatisch laufen lassen würde und miteinander vergleichen könnte. Für die Herangehensweise der Suche hätte man mehrere Möglichkeiten, man könnte sich bei der Optimierung an bekannten Verfahren für Optimierungsprobleme orientieren. Die wohl größte Weiterentwicklung die an PUX vorgenommen werden könnte ist die Ausweitung von einem Hidden-Markov-Modell auf ein mehrdimensionales System. Somit könnten eventuell noch weitere Einflussfaktoren berücksichtigt werden. Zusammenfassend, befindet PUX sich in einem sehr frühen Stadium und hat Potential und einige Weiterentwicklungsoptionen, ist aber ein guter Startpunkt um tiefer in die Analyse der PKS und der Frage ob Dichtebereiche stochastischen Regeln folgen, nachzugehen.

Literaturverzeichnis

[1] M. B. Sleiman and S. Gerdemann, “Covid-19: A Catalyst for Cybercrime?”, International Cybersecurity Law Review, vol. 2, no. 1, pp. 37–45, 2021.

[2] B. Rohs,Sicherheit im Internet: Cybercrime, Cyberterror und Cyberwar, pp. 303–332. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.

[3] W. Honekamp, Cybercrime: Aktuelle Erscheinungsformen und deren Bekämpfung, pp. 47–59. Wiesbaden: Springer Fachmedien Wiesbaden, 2019.

[4] R. Haverkamp, “Ein Überblick zur Dunkelfeldforschung in Deutschland,” Begriff, Methoden und Entwicklung, SIAK-Journal- Zeitschrift für Polizeiwissenschaft und polizeiliche Praxis (2/2019), pp. 15–30, 2019.

[5] H. Hess and S. Scheerer, “Theorie der Kriminalität,” Kölner Zeitschrift für Soziologie und Sozialpsychologie. Sonderheft, vol. 43, pp. 69–92, 2004.

[6] M. Gerhardt and H. Schuster, Das digitale Universum: Zelluläre Automatenals Modelle der Natur. Springer-Verlag, 2013.[

7] D. Labudde,Sicherheit ist die Abwesenheit von Kriminalität - eine Hypothese, pp. 145–153. München: Herbert Utz Verlag, 2018.

[8] S. Grolik, T. Stockheim, O. Wendt, S. Albayrak, and S. Fricke, “Dispositive supply-web-koordination durch Multiagentensysteme,” Wirtschaftsinformatik, vol. 43, no. 2, pp. 143–155, 2001.

[9] D. Veit, W. Fichtner, and M. Ragwitz, “Multi-Agenten-Systeme als Methode zur Simulation von Entscheidungsprozessen in der Energiewirtschaft,” 2004.

[10] E. Huber, Cybercrime - Eine Einführung. Berlin Heidelberg New York: Springer-Verlag, 2019.

[11] S. Gordon and R. Ford, “On the definition and classification of Cybercrime,” Journal in Computer Virology, vol. 2, no. 1, pp. 13–20, 2006.

[12] M. McGuire and S. Dowling, “Cyber Crime: A review of the evidence,” Summary of key findings and implications. Home Office Research Report, Vol. 75, 2013.

[13] P. K. Dick, The Minority Report. New York, 2002.

[14] A.-K. Lück, Der gläserne Mensch im Internet: Ethische Reflexionen zur Sichtbarkeit, Leiblichkeit und Personalität in der Online-Kommunikation. Kohlhammer Verlag, 2013.

[15] B. Pearsall, “Predictive policing: The future of law enforcement,” National Institute of Justice Journal, vol. 266, no. 1, pp. 16–19, 2010.

[16] W. Heinz, “Kriminalität und Kriminalitätskontrolle in Deutschland,” in Handbuch der Forensischen Psychiatrie, pp. 1–133, Springer, 2009.49

[17] M. Rolfes, “Predictive Policing: Beobachtungen und Reflexionen zur Eiführung und Etablierung einer vorhersagenden Polizeiarbeit,” Potsdamer Geographische Praxis, no. 12, pp. 51–76, 2017.

[18] J. Saunders, P. Hunt, and J. S. Hollywood, “Predictions put into Practice: a quasi-experimental Evaluation of Chicago’s Predictive Policing Pilot,” Journal of Experimental Criminology, vol. 12, no. 3, pp. 347–371, 2016.

[19] Bundeskriminalamt, “PKS Jahrbuch 2019,” Jahrbuch 2019 (1/1.0), p. 6, 2019.

[20] W. Heinz, Kriminalität und Kriminalitätskontrolle in Deutschland, pp. 1–133. Heidelberg: Steinkopff, 2009.

[21] E. Huber and B. Pospisil, “Problematik der Hell-und Dunkelfeldanalyse im Bereich Cybercrime,” Cyberkriminologie. Kriminologie für das digitale Zeitalter, pp. 109–133, 2020.

[22] K. Liebl, Dunkelfeldstudien im Vergleich: Bewertung der Aussagekraft von Untersuchungen zur Kriminalitätsbelastung. Springer-Verlag, 2019.

[23] M. Hartmann, “Machine Learning und IT-Security,” Datenschutz und Datensicherheit-DuD, vol. 42, no. 4, pp. 231–235, 2018.

[24] D. B. Waltl, “Erklärbarkeit und Transparenz im Machine Learning,” Philosophisches Handbuch Künstliche Intelligenz, pp. 1–23, 2020.

[25] U. Köthe, “Tiefe Netze. Von Maschinen lernen,” Ruperto Carola, no. 16,pp. 76–85, 2020.

[26] H. Finster and H.-G. Hirsch, “Bestimmung der optimalen HMM-Parameter zur robusten, phonembasierten Spracherkennung,” Studientexte zur Sprachkommunikation: Elektronische Sprachsignalverarbeitung 2004, pp. 125–132,2004.

[27] N. Nagorny, “Andrei Markov and mathematical constructivism,” in Studies in Logic and the Foundations of Mathematics, vol. 134, pp. 467–479, Elsevier, 1995.

[28] B. Wiesmüller, “Verkehrscharakterisierung durch Methoden des Maschinellen Lernens,” Innovative Internet Technologies and Mobile Communications(IITM), p. 80, 2009.

[29] F. Woelk, Signaldetektion bei verrauschten Markov-Prozessen. PhD Thesis, Diplomarbeit, Kiel, 2000.

[30] D. Meintrup and S. Schäffler, “Markov-ketten,” in Stochastik, pp. 227–265,Springer, 2005.50

[31] H. Wunsch, Der Baum-Welch Algorithmus für Hidden Markov Models, ein Spezialfall des EM-Algorithmus. PhD Thesis, Master’s Thesis, Universität Tübingen, 2001.

[32] S. R. Eddy, “What is a Hidden Markov Model?” Nature Biotechnology, vol. 22, no. 10, pp. 1315–1316, 2004.

[33] B. Wichern, Hidden-Markov-Modelle zur Analyse und Simulation von Finanzzeitreihen. PhD Thesis, Universität zu Köln, 2001.

[34] T. Dietterich, “Overfitting and Undercomputing in Machine Learning,”ACMcomputing surveys (CSUR), vol. 27, no. 3, pp. 326–327, 1995.

[35] Bundeskriminalamt, “Polizeiliche Kriminalstatistik (pks) 1953,” 1954.

[36] L. Richardson and S. Ruby,Web-services mit REST. O’Reilly Germany, 2007.

37] H. Balzert, Lehrbuch der Softwaretechnik: Entwurf, Implementierung, Installation und Betrieb. Springer-Verlag, 2011.

[38] M. Gede and J. Jeney, “Thematische Kartierung mit Verwendung von cesium,” KN-Journal of Cartography and Geographic Information, vol. 67,no. 4, pp. 210–213, 2017.

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum Vorname Nachname