Julius-Maximilians-
UNIVERSITÄT
WÜRZBURG

Der Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen mittels maschineller Lernverfahren: Eine Analyse anhand der Kreditwürdigkeitsprüfung im Finanzwesen

Bachelorarbeit

Eingereicht von:	Ebner, Nicolas
Studiengang:	Wirtschaftsinformatik B.Sc.
Matrikelnummer:	2353610
Betreuer:	Prof. Dr. Frédéric Thiesse
Bearbeitungszeit:	von 11.06.2021
	bis 06.08.2021

Julius-Maximilians-Universität Würzburg

Lehrstuhl für Wirtschaftsinformatik und Systementwicklung

Sanderring 2, 97070 Würzburg

Zusammenfassung

Generative Adversarial Networks (GAN) haben in letzter Zeit zunehmend Aufmerksamkeit erhalten. Ein Grund ist ihre Fähigkeit, realistische Datensätze mit hohem Schutz der Privatsphäre zu generieren (Ma et al. 2020, 1). Bei der Anwendung von GANs auf sensible oder private Trainingsbeispiele, wie zum Beispiel finanziellen Aufzeichnungen, ist es dennoch ein Problem, dass persönliche Informationen von Personen preisgegeben werden können (Xie et al. 2018, 1; Ma et al. 2020, 1). So haben Hitaj et al. (2017) ein aktives Inferenz-Angriffsmodell eingeführt, welches originale Trainingsdaten aus dem generierten Datensatz rekonstruieren kann (Xie et al. 2018, 1). Eine mögliche Lösung für die Datenschutzbedenken bietet Differential Privacy (DP). Die Definition von DP kann anhand einer soliden mathematischen Formulierung eine Garantie für die Privatsphäre von Teilnehmenden einer Datenbank schaffen (Torfi et al. 2020, 1). DP wird durch das Hinzufügen von Zufälligkeit in ein System erreicht (Dwork und Roth 2014, 15). Je mehr Zufall einem System beigemengt wird, desto mehr wird die Privatsphäre der einzelnen Teilnehmenden geschützt (Dwork 2011, 338; Torfi et al. 2020, 4). Wird einem Datensatz jedoch mehr Rauschen hinzugefügt, entfernt sich die Verteilung dieses Datensatzes immer weiter von der des originalen Datensatzes. Darunter leidet schlussendlich die Effektivität der Daten (vgl. Jordon et al. 2019, 9). Bei Verwendung von Differential Privacy gibt es somit immer einen Trade-off zwischen Effektivität der Daten und der erreichten Privatsphäre (Hsu et al. 2014, 1). Dieser Trade-off wird in folgender Arbeit anhand eines Beispiels zur Kreditwürdigkeitsprüfung im Finanzwesen empirisch untersucht. Der Aufbau der Analyse folgt dem Referenzprozess für Big Data Analysen nach Müller et al. (2016).

Abstract

Generative Adversarial Networks (GAN) have recently received increasing attention. One reason is their ability to generate realistic datasets with high privacy protection (Ma et al. 2020, 1). Nevertheless, when applying GANs to sensitive or private training examples, such as financial records, it is a problem that personal information of individuals may be revealed (Xie et al. 2018, 1; Ma et al. 2020, 1). For example, Hitaj et al. (2017) introduced an active inference attack model that can reconstruct original training data from the generated dataset (Xie et al. 2018, 1). Differential privacy (DP) offers a potential solution to privacy concerns. The definition of DP can create a guarantee of privacy for participants in a databank using a solid mathematical formulation (Torfi et al. 2020, 1). DP is achieved by adding randomness to a system (Dwork and Roth 2014, 15). The more randomness added to a system, the more the privacy of individual participants is protected (Dwork 2011, 338; Torfi et al. 2020, 4). However, as more noise is added to a dataset, the distribution of that dataset moves further and further away from that of the original dataset. Ultimately, the effectiveness of the data suffers (cf. Jordon et al. 2019, 9). Thus, when using differential privacy, there is always a trade-off between the effectiveness of the data and the privacy achieved (Hsu et al. 2014, 1). In the following paper, this trade-off is empirically investigated using an example of credit scoring in finance. The structure of the analysis follows the reference process for Big Data analyses according to Müller et al. (2016).

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AUC	Area under the Curve
AUROC	Area under the Receiver Operating Characteristic Curve
DP	Differential Privacy
DP-WGAN	Differential Privacy – Wasserstein Generative Adversarial Network
FN	Falsch Negativ
FP	Falsch Positiv
FPR	Falsch-positiv Rate
GAN	Generative Adversarial Networks
PATE	Private Aggregation of Teacher Ensembles
RN	Richtig Negativ
ROC	Receiver Operating Characteristic
RP	Richtig Positiv
TPR	Richtig-positiv Rate
VAE	Variational Autoencoder
WGAN	Wasserstein Generative Adversarial Network

1 Einleitung

Im Folgenden wird ein Überblick über das Thema, die Problemstellungen, die Zielsetzung und die Struktur der Arbeit gegeben.

1.1 Motivation

Die Verwendung von Machine Learning Modellen hat sich bereits in vielen Branchen bewährt (vgl. Géron 2020, 5–7). Der Erfolg hängt jedoch stark von der Verfügbarkeit einer großen Menge an Trainingsdaten ab (Torfi et al. 2020, 1). Daher kann der Fortschritt beim Einsatz solcher Modelle in speziellen kritischen Bereichen, wie z. B. dem Finanzwesen, mit strengen Vorgaben bezüglich des Datenschutzes und der Involvierung großer Mengen an sensiblen Daten gehindert werden (Ma et al. 2020, 1; Jordon et al. 2019, 1). Um bestimmte, vielversprechende, datenintensive Methoden effektiv nutzen zu können, ist es daher notwendig, sich mit den Datenschutzproblemen in diesen Bereichen auseinanderzusetzen. Eine in der Praxis verwendete Methode, um die Datenschutzbedenken beim Umgang mit sensiblen Informationen zu bewältigen, ist die Anonymisierung personenbezogener Daten (Dwork und Roth 2014, 7). Solche Ansätze sind jedoch anfällig für De-Anonymisierungsangriffe (vgl. Narayanan und Shmatikov 2008). Ein effektiver Ansatz, Probleme mit privaten Daten zu bewältigen, ist die Generierung realistischer, synthetischer Daten, die eine praktisch akzeptable Datenqualität bieten (Torfi et al. 2020, 1). Aufgrund ihres jüngsten Erfolgs in anderen Domänen, wie der Generierung fotorealistischer Bilder (Zhang et al. 2019), haben Generative Adversarial Networks (GAN) in diesem Forschungsbereich viel Aufmerksamkeit auf sich gezogen. GANs sind nicht umkehrbar, d. h. es kann keine deterministische Funktion verwenden werden, um von den generierten Stichproben zu den realen Stichproben zu gelangen (Torfi et al. 2020, 1). Die Verwendung von GANs für die Generierung synthetischer Daten garantiert jedoch nicht, dass das System datenschutzkonform ist, da sich auch GANs bereits als anfällig erwiesen haben (Hayes et al. 2019). Eine angesehene Methode, um die Privatsphäre der Teilnehmenden einer Datenbank zu beschützen, ist die von Dwork et al. (2006) entwickelte Definition von Privatsphäre namens Differential Privacy (DP) (Sarwate und Chaudhuri 2013, 87). DP wird durch das Hinzufügen von Zufälligkeit in ein System erreicht (Dwork und Roth 2014, 15). Mit dem Einfügen von Zufälligkeit in einen Datensatz leidet jedoch die Qualität der Daten, da diese nicht mehr originalgetreu sind (vgl. Xie et al. 2018, 6; Jordon et al. 2019, 9). Es besteht also der Bedarf, die Privatsphäre der Teilnehmenden einer Datenbank zu schützen. Dem entgegengesetzt besteht ebenfalls das Bestreben, möglichst realistische Daten zu generieren.

1.2 Problemstellung

Wie viel Privatsphäre einem System mit Differential Privacy hinzugeführt wird, lässt sich einfach und mathematisch klar beschreiben. Die Wahl der Parameter und damit die Stärke der hinzugefügten Privatsphäre sind jedoch nicht trivial (Lee und Clifton 2011, 325). Forschende, die mit den zur Verfügung gestellten Daten arbeiten, erhoffen sich eine möglichst hohe Datenqualität, während die Teilnehmenden eines Datensatzes meist höhere Ansprüche an ihren eigenen Datenschutz stellen. Die Wahl der Parameter von Differential Privacy sind laut der Mitbegründerin der Definition somit eine soziale Frage (Dwork 2008, 3). Die richtige Festlegung dieser Parameter ist daher auch außerhalb des Untersuchungsbereichs dieser Arbeit. Es wird jedoch versucht, anhand eines praktischen Beispiels bei der Kreditwürdigkeitsprüfung im Finanzwesen einen Überblick über den Trade-off von Effektivität und Differential Privacy zu schaffen, so dass sich am Ende dieser Arbeit, der/die Lesende eine eigene Meinung über diesen Themenbereich bilden kann.

1.3 Zielsetzung

In dieser Arbeit werden aufbauend auf einem Datensatz zur Kreditwürdigkeitsprüfung synthetische tabellarische Daten mittels eines generativen Modells erstellt, welches mit unterschiedlichen Werten des Differential Privacy Parameters ε verwendet wird. Im Anschluss wird die Effektivität darauf trainierter Vorhersagemodelle und damit die Effektivität der entstehenden synthetischen Daten auf einem realen Testdatensatz evaluiert.

Das Ziel dieser Arbeit ist es, ein Überblick über den Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen bezüglich der Kreditwürdigkeitsprüfung im Finanzwesen darzulegen. Die Forschungsfrage lautet wie folgt:

Wie lässt sich der Trade-off zwischen Effektivität und Differential Privacy bei der synthetischen Datengenerierung mittels eines generativen Modells anhand des Beispiels der Kreditwürdigkeitsprüfung im Finanzwesen bewerten?

1.4 Vorgehensweise

Zu Beginn der Arbeit werden theoretische Grundlagen über die Methode von Differential Privacy erklärt. Dafür wird die Motivation und Funktionsweise dieser Technik verdeutlicht. Ebenfalls wird Differential Privacy definiert. Im Anschluss wird der Fokus auf Deep Generative Models gelegt. Dort wird insbesondere auf GANs eingegangen. Zusätzlich werden Probleme beim Erstellen synthetischer tabellarischer Daten im Finanzwesen dargestellt. Das Theoriekapitel wird durch die Vorstellung und Erklärung des in dieser Arbeit verwendeten generativen Modells abgeschlossen.

Nachdem die theoretischen Grundlagen für diese Arbeit gelegt sind, werden die vier Schritte der methodische Vorgehensweise nach Müller et al. (2016) in einem Methodik Kapitel erklärt.

Im darauffolgenden Kapitel wird diese Methode anhand von Daten zur Kreditwürdigkeitsprüfung durchgeführt. Im Abschnitt „Datenbeschaffung“ wird der Prozess der Wahl der Daten erläutert. Für diese Arbeit wird ein Datensatz von Kaggle.com verwendet, der Hauptbestandteil der Competition „Give me some Credit“ (Kaggle.com 2011) ist. Das Ziel ist es, anhand der Daten vorherzusagen, ob Teilnehmende der Datenbank in den nächsten zwei Jahren in eine finanzielle Delinquenz geraten.

In der Datenanalysephase werden die Originaldaten mit einer explorativen Datenanalyse untersucht. Gleichzeitig wird der Datensatz für die spätere Anwendung der Modelle vorbereitet. Es folgt ein Kapitel zu Evaluationsmetriken der binären Klassifikation, die auf diesen Datensatz anwendbar sind. Im darauffolgenden Kapitel wird auf die synthetische Datengenerierung eingegangen. Im Anschluss wird der Prozess der Modellentwicklung erläutert, wobei die verwendeten Klassifizierungsalgorithmen genannt und kurz erklärt werden.

Als Abschluss der Datenanalyse wird eine Evaluation durchgeführt. Hierfür werden die in den vorherigen Kapiteln erarbeiteten Schritte zusammengeführt. Auf den bereinigten Daten werden synthetische Daten mittels DP-WGAN erstellt. Diese werden anhand der AUROC Evaluationsmetrik mit den entwickelten Modellen auf die Effektivität hin untersucht.

Um den Trade-off zwischen Effektivität und Privatsphäre bewerten zu können, wird zuerst ein Vergleich der realen Daten und generierter Daten ohne DP gezogen. Im Anschluss wird die Effektivität der generierten Datensätze mit verschiedenen Größen des ε-Differential Privacy Faktors untersucht. Als Abschluss der Methodik, werden die Ergebnisse aus den vorherigen Schritten interpretiert.

2 Theoretische Grundlagen

Im Folgenden werden die theoretischen Grundlagen dieser Arbeit dargelegt. Dabei wird in Kapitel 2.1 auf Differential Privacy eingegangen. Hierfür wird diese zum einen definiert, und zum anderen folgen verschiedene Techniken, wie diese angewendet werden kann. Des Weiteren werden Herausforderungen bei der Anwendung beschrieben. Im darauffolgenden Kapitel 2.2 wird näher auf Deep Generative Models eingegangen. Es wird die Methode des Variational Autoencoders und die der Generative Adversarial Models dargestellt. Der Fokus liegt dabei auf zweiterem, da diese Methode im folgenden praktischen Teil angewendet wird.

2.1 Differential Privacy

Ein Beispiel für die nicht ordnungsgemäße Anonymisierung eines Datensatzes vor Veröffentlichung entstand, als Netflix im Jahr 2006 einen Datensatz von Benutzern/-innen und deren Bewertungen zu Filmen veröffentlichte. Das Ziel dieser Aktion war es, das Empfehlungssystem von Netflix zu verbessern (Dwork und Roth 2014, 7). Narayanan und Shmatikov (2008) haben bewiesen, dass die vorherige Anonymisierung der Daten nicht ausreichen war. Von Netflix wurden alle Attribute, die zu einer direkten Identifikation führen könnten, wie z. B. der Name und Adressen, vom Datensatz entfernt. Dennoch war es möglich, viele Nutzer/-innen mit einer sehr hohen Wahrscheinlichkeit zu identifizieren. Dafür verknüpften Narayanan und Shmatikov (2008) die veröffentlichten Daten von Netflix mit öffentlich verfügbaren Daten der Internet Movie Database (IMDb). Diese Art von Attacke nennt sich „linkage attack“ (Dwork und Roth 2014, 7). Hierbei wurde deutlich, dass bei einer Veröffentlichung von Daten eine einfache Anonymisierung nicht unbedingt ausreicht. Dies stellt ein Problem für Unternehmen dar, die bereit wären, ihre gesammelten Daten zu teilen. Sie würden nämlich so die Gefahr eingehen, dass die Daten ihrer Kundinnen und Kunden de-anonymisiert werden und somit an die Öffentlichkeit geraten. Dies ist ein Grund, weshalb nur wenige öffentliche Datensätze für sensible Daten, wie in dem Finanzwesen, vorliegen (Jordon et al. 2019, 1). Nach Dastile et al. (2020, 2) sind die zwei meistgenutzten Datensätze in öffentlichen, wissenschaftlichen Untersuchungen zu der Kreditwürdigkeitsprüfung, das „deutsche“ (Hofmann 1994) mit 1000 Einträgen und das „australische“ (UCI 1987) mit 690 Zeilen. Beide enthalten jeweils nur wenige Einträge und sind für das digitale, datengetriebene Zeitalter sehr alt. Im Jahr 2006 veröffentlichten Dwork et al. (2006) ihre Arbeit um das Konzept von Differential Privacy. Dies könnte eine Möglichkeit sein, zum einen Personen zu motivieren an einer Datenerhebung teilzunehmen und zum anderen die Gefahr der Verletzung der Privatsphäre beim öffentlichen Teilen eines Datensatz zu eliminieren (Dwork und Roth 2014, 5).

Der Begriff Differential Privacy hat in letzter Zeit viel Aufmerksamkeit erhalten. Da diese unter Verwendung einer soliden mathematischen Definition einen Ansatz zur Sicherstellung und

Quantifizierung der Privatsphäre eines Systems bietet (Torfi et al. 2020, 1). Die Stärke von DP liegt in der akkuraten mathematischen Darstellung, welche die Privatsphäre sicherstellt, ohne das Modell für genaue statistische Schlussfolgerungen einzuschränken. Zusätzlich kann mit deren Hilfe, der Grad der Privatsphäre eines Systems gemessen werden (Torfi et al. 2020, 1). Aus diesen Gründen wurde DP zum Defacto-Standard für die statistische Untersuchung von Datenbanken, die sensible private Daten enthalten (Hsu et al. 2014, 1).

Große öffentliche Aufmerksamkeit erhielt DP bei der Anwendung der Methode im Jahr 2020 bei der Volkszählung, dem „Census", in Amerika: „The 2020 Census will use a powerful new privacy protection system known in scientific circles as ‚differential privacy', designed specifically for the digital age. The Census Bureau is transitioning to this new, state-of-the-art privacy protection system to keep pace with emerging threats in today's digital world." (United States Census Bureau 2020).

2.1.1 Definition

DP ist eine mathematische Definition, welche das Datenschutzrisiko eines Individuums bei der Teilnahme an einer Datenbank beschreibt. Sie ist im Wesentlichen eine Garantie der Daten sammelnden Organisation oder Person gegenüber dem Individuum, dass sich die Folgen einer Datenbank, d. h. die Ergebnisse eines Algorithmus, der die Datenbank nutzt, nicht wesentlich ändern, ob diese die Nutzung der Daten zulässt oder nicht (Dwork und Roth 2014, 17).

Die Definition von DP basiert auf dem Vergleich zwei benachbarter Datensets.

Definition 2.1: Zwei Datasets *D, D´* werden als benachbart bezeichnet, wenn

$$\exists x \in D \; so\, dass\colon D\{x\} = D'$$

(Jordon et al. 2019, 3)

Zwei Datenbanken sind somit benachbart, wenn die eine Datenbank, eine echte Teilmenge der anderen ist und die größere Datenbank nur eine zusätzliche Zeile enthält. Sozusagen wenn *D* und *D'* sich in genau einem Element unterscheiden.

Definition 2.2: (ε-Differential Privacy) Ein randomisierter Algorithmus *M*, ist ε-differentiell privat, wenn für alle benachbarten Datensätze *D* und *D'*, die sich in höchstens einem Element unterscheiden und *S ⊆ Range(M)*, gilt

$$P[M(D) \in S] \leq e^{\varepsilon} * P[M(D') \in S]$$

Wobei *P* in Bezug auf die Zufälligkeit von *M* genommen wird.

(Dwork 2008, 2; Jordon et al. 2019, 3)

Wie aus Definition 2.2 zu erkennen ist, wird die Verteilung der Ergebnisse auf den zwei benachbarten Datensätzen durch den variablen Parameter ε unterschieden. Je größer ε gewählt ist, desto weiter können die Verteilungen sich unterscheiden. Daraus lässt sich ableiten, dass ein kleinerer Wert von ε einen höheren Grad an Privatsphäre sichert (Torfi et al. 2020, 4). Ein Mechanismus *M*, der diese Definition erfüllt, adressiert die Bedenken, die jeder Anwesende im Datensatz über das Durchsickern seiner persönlichen Informationen haben könnte (Dwork 2008, 2). Selbst wenn eine teilnehmende Person ihre Daten aus dem Datensatz entfernt, würde keine Ausgabe wesentlich wahrscheinlicher oder unwahrscheinlicher werden (Dwork 2008, 2). Wenn die Datenbank beispielsweise von einer Bank konsultiert wird und diese Abwägungen über die Kreditwürdigkeit einer Person trifft, dann hat das Vorhandensein oder Fehlen der Daten dieser Person in der Datenbank keinen signifikanten Einfluss auf ihre Chancen einen Kredit zu erhalten (vgl. Dwork 2008, 2).

Da ε-DP hohe Anforderungen an Mechanismen stellt, verlieren diese zum Teil stark an Nutzen. ε-DP gewährleistet, dass bei jedem Durchlauf des Mechanismus *M* die beobachtete Ausgabe mit (fast) gleicher Wahrscheinlichkeit bei *D* und *D'* beobachtet wird. Die Erweiterung (ε, δ)-DP besagt, dass dies nicht bei jedem, sondern nur bei fast jedem Durchlauf zutrifft (Dwork und Roth 2014, 18). Diese Definition erlaubt, dass Voraussetzungen bis zu einem gewissen Grad unerfüllt bleiben. Somit wird die Definition von ε-DP etwas gelockert. Das wird erreicht, indem an die Formel von ε-DP der zusätzliche Parameter δ addiert wird:

Definition 2.3: (ε, δ)-Differential Privacy

$$P[M(D) \in S] \leq e^{\varepsilon} * P[M(D') \in S] + \delta$$

(Dwork und Roth 2014, 17)

Dabei sind beide Parameter $\varepsilon, \delta > 0$. Der Parameter δ ist in der Literatur typischerweise sehr klein, kleiner als der Kehrwert eines beliebigen Polynoms in der Größe der Datenbank (Dwork 2011, 339). Anderenfalls wäre die Toleranzgrenze zu groß und die Definition von Privatsphäre nicht mehr gegeben.

2.1.2 Anwendung

Datenschutz kann laut Dwork und Roth (2014, 16) nicht durch die Anwendung einer deterministischen Funktion auf die Daten erreicht werden. Die Randomisierung ist ein essenzieller Teil, um eine nicht triviale Datenschutzgarantie zu schaffen. Dies argumentieren sie anhand folgenden Beispiels. Es wird angenommen, dass zwei benachbarte Datenbanken existieren. Die Abfrage einer deterministischen Funktion auf beide Datenbanken führt zu zwei unterschiedlichen Ergebnissen. Eine angreifende Person, welcher diese Tatsache bewusst ist, kann aus den unterschiedlichen Ergebnissen die Werte der einen zusätzlichen Spalte erfahren (Dwork und Roth 2014, 16).

DP bietet Privatsphäre durch Prozess, dafür wird in Ergebnisse eines Mechanismus Zufall eingefügt (Dwork und Roth 2014, 15). Ein aus Dwork und Roth (2014, 15f.) entnommenes Beispiel für Privatsphäre durch Zufallsprozess ist eine in den Sozialwissenschaften entwickelte Technik namens „Randomized Response" (randomisierte Antwort) (Warner 1965). Diese Methode kann angewendet werden, um statische Informationen über peinliches oder illegales Verhalten zu sammeln. Es wird das Vorhandensein einer Eigenschaft *P* erfasst und die Teilnehmenden der Studie werden aufgefordert wie folgt zu antworten:

1. Wirf eine Münze
2. Wenn Zahl, dann sage die wahre Antwort
3. Wenn Kopf, wirf die Münze ein zweites Mal und antworte „Ja" bei Kopf und „Nein" bei Zahl

Wenn es sich bei der Eigenschaft *P* um ein illegales Verhalten handelt, ist selbst eine „Ja" Antwort nicht belastend, da der Teilnehmende nur zu 50% mit einer wahren Aussage geantwortet hat. Die anderen 50% sind zufällige „Ja" oder „Nein" Antworten. Die Privatsphäre kommt von der plausiblen Bestreitbarkeit jedes Ergebnisses (Dwork und Roth 2014, 15f.). Die Genauigkeit ergibt sich aus genügend Teilnehmenden und dem Verständnis des hinzugefügten Rauschens (Dwork und Roth 2014, 15f.).

Es ist wichtig zu verstehen, dass DP eine Definition und kein Algorithmus ist (Dwork und Roth 2014, 6). So gibt es für eine gegebene Aufgabe, unterschiedliche Methoden/ Algorithmen, um ε-DP zu erreichen, manche mit besserer Genauigkeit als andere (Dwork und Roth 2014, 6). Bei der traditionellen Anwendung von Differential Privacy gibt es jedoch zwei mögliche Modelle, die unterschieden werden können (Kang et al. 2020, 1). Das lokale Modell einerseits konzentriert sich darauf, DP auf den Trainingsdaten zu erreichen, bei einem zentralen Modell andererseits wird DP auf das verwendete Machine Learning Modell angewendet (Kang et al. 2020, 1). Bei „Randomized Response" wird das Rauschen direkt bei der Datenerhebung hinzugefügt und damit ist es eine der ersten Anwendungen eines lokalen Privatsphäre Modells (Dwork und Roth 2014, 233)

Bei einem zentralen Modell wird DP nicht auf den verwendeten Datensatz erreicht. Es muss somit immer einen vertrauenswürdigen Datenbankadministrator geben, der direkten Zugang zu den realen privaten Daten hat (Dwork und Roth 2014, 232). Da DP auf das verwendete Machine Learning Modell angewendet wird, können bei dieser Methode nur die Ergebnisse der differentiell privaten Algorithmen veröffentlich werden (Dwork und Roth 2014, 232). Um die Privatsphäre der Teilnehmenden zu schützen, muss gewährleistet werden, dass ein(e) potentielle Angreifer/-in nur Zugriff auf die Ausgaben des Algorithmus hat und nicht die realen Daten einsehen kann (Dwork und Roth 2014, 232).

Bei einem lokalen Modell wird DP direkt auf den Datensatz angewendet. Ein großer Vorteil dieser Methode ist, dass das entstehende, differentiell private Datenset mit der Garantie von

Differential Privacy frei geteilt werden kann (Dwork und Roth 2014, 232). Da ein Algorithmus, der auf ein differentiell privates Datenset trainiert wird, nie die originalen/ realen Daten zu verarbeiten bekommt, folgen alle Ergebnisse dieser Algorithmen den Richtlinien von DP (Dwork und Roth 2014, 233f.). Dies wird durch das „Post-Processing" Theorem von DP bestätigt, welches kurzgefasst besagt, dass weitere Berechnungen auf differentiell privaten Ergebnissen, ohne die Verwendung von Originaldaten, die Privatsphäre der Ergebnisse nicht mindern können (Dwork und Roth 2014, 19). Der Datensatz kann dementsprechend verwendet werden, um ihn mit anderen Forscher-/innen zu teilen und jegliche weitere Berechnungen darauf durchzuführen (Jordon et al. 2019, 1; Nguyen et al. 2016, 87f.). Der Unterschied zu einem zentralen Modell liegt darin, dass für ein lokales Modell das Rauschen dem Datensatz bereits bei der Datenerhebung zugeführt werden muss (Wang et al. 2020, 1).

Die in dieser Arbeit verwendete Methode ist dem zentralen Modell zuzuordnen. Es existiert ein realer Datensatz, auf dem ein Machine Learning Modell mit den Einschränkungen von DP trainiert wird. Das Besondere ist, dass in dieser Arbeit ein generatives Modell verwendet wird, welches genutzt werden kann, um neue synthetische Daten zu erstellen. Der Unterschied von diskriminativen und generativen Modellen wird im Folgenden Kapitel 2.2 erklärt. Die neu erstellten synthetischen Daten mit DP haben alle Vorteile, die einem Datensatz aus einem lokalen Modell zuzuschreiben sind und können ebenfalls mit der Garantie von DP frei geteilt werden (Jordon et al. 2019, 1).

2.1.3 Was Differential Privacy nicht leistet

„[Differential Privacy] does not promise unconditional freedom from harm. Nor does it create privacy where none previously exists. More generally, differential privacy does not guarantee that what one believes to be one's secrets will remain secret" (Dwork und Roth 2014, 22). Die Definition von DP befasst sich mit dem Paradoxon, möglichst nützliche Informationen über eine Gruppe von Teilnehmenden zu erhalten, dabei jedoch nichts über ein einzelnes Individuum zu erfahren (Dwork und Roth 2014, 5). Dies wird verdeutlich anhand eines Beispiels, welches aus Dwork und Roth (2014, 5f.) entnommen ist und auf den Kontext dieser Arbeit angepasst wird.

Zum Beispiel kann die Studie einer Bank uns lehren, dass eine alleinstehende Person eher dazu neigt, einen Kredit aufzunehmen, welchen diese nicht zurückzahlen kann. Eventuell wägt eine verheiratete Person mit mehreren Kindern das Risiko vor der Kreditaufnahme eher ab und kann diesen zuverlässiger zurückzahlen. Anhand dieser Information kann die Kreditvergabe einer Bank beeinflusst werden. Ist eine alleinstehende Person durch diese Analyse geschädigt worden? Es ist möglich, dass die Konditionen der Kreditvergabe für diese Person negativ beeinflusst werden, da sie alleinstehend ist. Ist die Privatsphäre der alleinstehenden Person gefährdet? Die Resultate der Studie enthalten statistische Informationen, die Aussagen über die zwei

betrachteten Personengruppen beinhalten. Anhand der Ergebnisse sind somit nach der Studie mehr Informationen über die alleinstehende Person bekannt als vorher. Bei DP wird jedoch die Ansicht vertreten, dass dies keine Verletzung der Privatsphäre ist (Dwork und Roth 2014, 6). Dies wird damit begründet, dass die Auswirkungen auf die alleinstehende Person die gleichen seien, unabhängig davon, ob sie an der Studie teilgenommen hat oder nicht. Es ist nicht die An- oder Abwesenheit in der Datenbank, welche die Person beeinflusst, sondern es sind die in der Studie erzielten Schlussfolgerungen (Dwork und Roth 2014, 5).

2.1.4 Herausforderungen

Obwohl DP eine wichtige mathematische Garantie für die Privatsphäre von Personen ist, birgt sie auch einige Herausforderungen und daraus entstehende Einschränkungen. Im folgenden Abschnitt wird ein kurzer Überblick über diese dargestellt.

Das Kompositionstheorem von DP besagt, dass wenn zwei oder mehr differentiell private Algorithmen auf den originalen Daten entstehen, bewahrt eine Kombination dieser immer noch die Privatsphäre, allerding nehmen die Datenschutzgarantien mit der Mehrfachverwendung der Daten ab (Dwork und Roth 2014, 41–43). Angenommen der Mechanismus M_1 ist ε_1*-differentiell privat* und M_2 ist ε_2*-differentiell privat*, dann ist die Komposition $M_{1,2} = (M_1, M_2)$ gleich *(*$\varepsilon_1 + \varepsilon_2$*)-differentiell privat* (Dwork und Roth 2014, 41–43). Somit verschlechtert das sequentielle Abfragen von DP Mechanismen auf den selben Daten das gesamte Datenschutzniveau (Kairouz et al. 2013, 1).

Die Menge an Privatsphäre, die einem Datensatz hinzugeführt werden muss, um ε-DP zu erreichen, ist nicht trivial (Lee und Clifton 2011, 325; Nguyen et al. 2016, 86). Es ist abhängig von der Größe des Datensatzes und von der Art der Anwendung. Denn ε ist kein absolutes Maß für den Datenschutz, sondern eher ein relatives (Lee und Clifton 2011, 325). Selbst bei gleichem Wert von ε sind die von der differentiellen Privatsphäre erzwungenen Datenschutzgarantien je nach Anwendungsfall unterschiedlich (Lee und Clifton 2011, 325).

Des Weiteren bleibt die Wahl des Parameters ε von DP eine soziale Frage (Dwork 2011, 338). Offensichtlich sind kleinere Werte für ε vorteilhafter, da diese einen höheren Grad an Privatsphäre bieten. Jedoch muss dem System mehr Zufälligkeit eingeführt werden, um diese zu erreichen. Folglich nimmt die Qualität der Daten dabei ab (Xie et al. 2018, 6). Dies kann so weit gehen, dass der nun Privatsphäre schützende Mechanismus keinen statistisch relevanten Zusammenhang mehr bietet. Aus diesem Grund ist die Wahl der Parameter von Differential Privacy immer eine Abwägung zwischen mehr Datenschutz oder höhere Qualität der Ergebnisse. Es ist nicht möglich, beides gleichzeitig zu erhalten (Hsu et al. 2014, 1).

2.2 Deep Generative Models

Aufgrund der rasanten Entwicklung von Computerhardware und -software sind in den letzten Jahren immer mehr Daten in unterschiedlichen Anwendungsbereichen verfügbar geworden (Xie et al. 2018, 1). Daraufhin sind viele Methoden entwickelt worden, um diese großen Datensätze zu analysieren. Ein repräsentatives Beispiel, welches typischerweise eine große Menge an Trainingsdaten benötigt, um vielversprechende Ergebnisse zu liefern, ist Deep Learning (Xie et al. 2018, 1; Torfi et al. 2020, 1f.). Es gibt jedoch Bereiche, in denen es schwierig ist, so viele Daten zu erhalten, wie benötigt werden. Ein Beispiel dafür sind medizinische Daten oder auch Finanzdaten (Xie et al. 2018, 1; Ma et al. 2020, 1; Torfi et al. 2020, 1). Zum einen ist die Zielvariable in den Daten häufig stark untervertreten. So benötigt ein Algorithmus der darauf trainiert werden soll, eine seltene Krankheit zu erkennen, genug Beispiele auf der er diese Krankheit auch erkennen kann (vgl. Xie et al. 2018, 1). Genauso müssen für die Kreditwürdigkeitsprüfung genug Beispiele gesammelt werden, bei denen ein Kredit ausgegeben wird, welcher von der Kundin oder dem Kunden nicht zurückgezahlt werden kann. Da ein Kredit häufig über mehrere Jahre ausgegeben wird, muss ebenfalls die Datenerfassung über einen solchen Zeitraum gestreckt sein. Eine zweite Herausforderung, an solche Daten zu gelangen, ist der Datenschutz. Die Verbreitung eines Datensatz mit persönlichen Informationen über die gesundheitliche oder finanzielle Lage einzelner Personen ist ohne Einschränkungen nicht erlaubt (Torfi et al. 2020, 1).

Eine wichtige Unterteilung beim maschinellen Lernen ist die generative gegenüber der diskriminativen Modellierung (Kingma und Welling 2019, 308; Kalin 2018, 8). Dabei sind diskriminative Modelle die Methode mit der die meisten Personen im Machine Learning Kontext vertraut sind (Kalin 2018, 8f.). Eine beispielhafte Anwendung ist die Klassifikation von Daten. Bei der diskriminativen Modellierung wird versucht, ein Modell zu entwickeln, welches Vorhersagen anhand der zur Verfügung stehenden Beobachtungen trifft (Kingma und Welling 2019, 308). Bei der generativen Modellierung ist das Ziel, eine gemeinsame Verteilung über alle Variablen zu lösen. Dementsprechend versucht ein solches Modell nachzustellen, wie die Daten in der realen Wert erzeugt werden (Kingma und Welling 2019, 308). Damit bieten generative Modelle einen vielversprechenden Ansatz, um das Problem der Datenknappheit zu lindern (Ma et al. 2020, 1). Des Weiteren können sie eine Lösung für die Problemstellung um den Datenschutz der Rohdaten darstellen (Torfi et al. 2020, 1). Generative Modelle versuchen, aus einem Trainingsdatensatz die Muster und die Art der Datenverteilung zu erlernen. Soweit das Modell dies geschafft hat, können damit Abschätzungen über die Verteilung eines Datensatzes gemacht werden und das Modell kann ebenfalls genutzt werden, um realistische synthetische Daten zu generieren. Diese bilden im Idealfall die exakte Verteilung der realen Daten ab, können jedoch die Privatsphäre schützen, indem sie keine Informationen über die einzelnen Einträge der Trainingsdaten preisgeben (vgl. Xie et al. 2018, 1).

Im Machine Learning Kontext haben sich zwei Methoden zu Erstellung generativer Modelle durchgesetzt. Zum einen Variational Autoencoders (VAE) und zum anderen Generative Adversarial Networks (Kingma und Welling 2019, 311). Auf diese zwei Methoden wird im Folgenden genauer eingegangen. Dabei wird in dieser Arbeit der Fokus auf die Erstellung eines generativen Modells mittels GANs gelegt.

2.2.1 Variational Autoencoders

Die Architektur des VAE wurde von Kingma und Welling (2013) entworfen. Für diese Arbeit ist es ausreichend, das Framework in nur einem groben Überblick darzustellen. Die Erklärung beginnt mit der Beschreibung eines normalen Autoencoders. Wie der Name bereits vermuten lässt, wird dieser verwendet, um Daten automatisch zu kodieren (Langr und Bok 2019, 18). Autoencoder bestehen aus den zwei Bestandteilen: Encoder und Decoder die in den meisten Anwendungen jeweils als neuronales Netz implementiert sind (Langr und Bok 2019, 18).

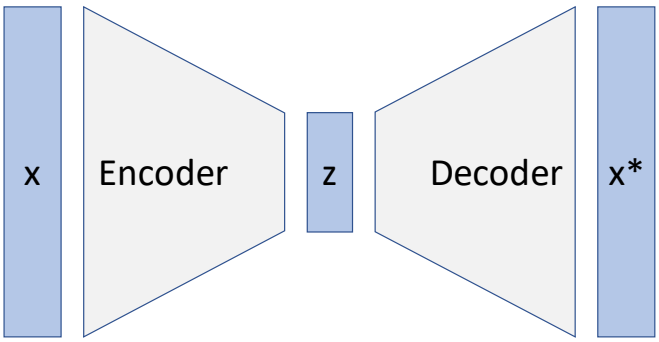

Abbildung 1: Architektur eines Basis Autoencoders (Langr und Bok 2019, 20)

Die folgenden Erklärungen der Autoencoder und der VAE Architektur wird aus Langr und Bok (2019, 19–25) abgeleitet. Das Encoder-Netzwerk erhält als Input einen Vektor x mit der Größe y. Hierbei könnte es sich um ein Bild, eine Audio-Datei oder eine andere Art Datei handeln. In dieser Arbeit würde x den Datensatz repräsentieren. Der Encoder reduziert/ komprimiert die Dimension des Vektors x von y nach z. Der Raum z in der Mitte von Abbildung 1 wird als latenter Raum bezeichnet, der typischerweise eine Darstellung einer kleineren Dimension des Ursprungsvektors x ist und als Zwischenschritt dient. Mit Hilfe des Decoders werden die Informationen aus dem latenten Raum in die ursprüngliche Dimension rekonstruiert. Es beschreibt somit den umgekehrten Prozess der Kodierung. Beim Training eines Autoencoders wird der Rekonstruktionsverlust von den originalen Daten x zu den rekonstruierten Daten x* mit einer Abstandsmessung der zwei Verteilungen bestimmt. Es wird dann versucht, die Parameter des Encoder und Decoder so einzustellen, dass der Verlust minimiert wird (Langr und Bok 2019,

19f.). Ein Variational Autoencoder unterscheidet sich von einem normalen Autoencoder im latenten Raum. Während bei einem Autoencoder der latente Raum aus einer unsortierten Menge von Zahlen besteht, wird beim VAE der latente Raum als eine Verteilung mit einem gelernten Mittelwert und einer Standardabweichung repräsentiert. Typischerweise wird hier eine multivariate Gauß-Verteilung gewählt (Langr und Bok 2019, 25). Das Sortieren des latenten Raums ermöglicht es, realistische neue Datensätze zu erzeugen. Zum Generieren von neuen Einträgen, wird der Encoder abgeschaltet und dem Decoder ein zufälliger Vektor aus dem latenten Raum in der Form von z übergeben. Da dieser im Trainingsprozess gelernt hat, die Originalverteilung aus einen Vektor des latenten Raums herzustellen, entstehen aus dem zufälligen Vektor neue, zu den Originaldaten ähnliche Verteilungen (Langr und Bok 2019, 24).

Das Framework der VAE hat eine breite Palette von Anwendungen von generativer Modellierung, semi-supervised learning bis hin zu representation learning (Kingma und Welling 2019, 312). Zusammenfassend ist die allgemeine Idee von Autoencodern recht einfach und besteht darin, einen Encoder und einen Decoder als neuronale Netze zu setzen und das beste Kodier-Dekodier-Schema durch einen iterativen Optimierungsprozess zu erlernen. Bei jeder Iteration erhält der Autoencoder zuerst einige Samples an Daten, die kodierte-dekodierte Ausgabe wird danach mit den Anfangsdaten verglichen und der Fehler wird genutzt, um die Gewichte der Netzwerke zu aktualisieren. Der Decoder kann im Anschluss verwendet werden, um neue Samples zu generieren.

2.2.2 Generative Adversarial Networks

Dieses Modell wurde von Goodfellow et al. (2014) entwickelt und ebnete den Weg für unterschiedlichste Variationen, welche auf der Grundidee von GANs aufbauen. GANs verfolgen eine andere Strategie, generative Modelle zu modellieren. Dabei werden zwei neuronale Netze gegenüber gestellt, die in einem Zwei-Spieler Minimax Spiel gegeneinander antreten (Goodfellow et al. 2014, 1). Zum einen gibt es den Generator mit dem Ziel, Daten zu generieren, die den echten Daten täuschend ähnlich sind und nicht von den Trainingsdaten zu unterscheiden sind und zum anderen benötig es einen Diskriminator, welcher das Ziel verfolgt echte und vom Generator erzeugte Daten zu unterscheiden, indem er die vorherigen Erfahrungen nutzt (Goodfellow et al. 2014, 2f.). Das Spiel und damit das Modell ist beendet, wenn das globale Optimum erreicht ist. Das heißt, wenn die Datenverteilung, welche vom Generator erzeugt wird, gleich der unserer Trainingsdaten ist (Goodfellow et al. 2014, 1). Essenzieller Baustein in der GAN Struktur sind dafür die Verlustfunktionen der beiden neuronalen Netze (Kalin 2018, 19). Der Prozess des Trainings besteht darin, die Gewichte so einzustellen, dass die Verlustfunktionen für den gegebenen Problemsatz optimiert werden. Die für das neuronale Netz gewählte Verlustfunktion ist daher entscheidend dafür, dass das neuronale Netz gute Ergebnisse liefert und konvergiert (Kalin 2018, 19).

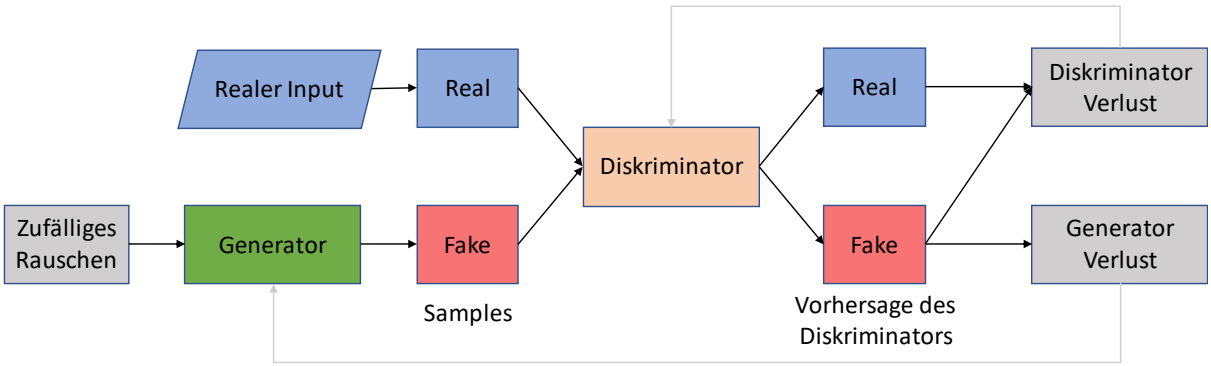

Abbildung 2: Darstellung einer Basis GAN Architektur angelehnt an (Saxena und Cao 2021, 5)

Aus Abbildung 2 wird deutlich, dass der Generator zufälliges Rauschen als Input erhält, woraus unechte Samples generiert werden. Der Diskriminator erhält als Input die unechten Samples des Generators und die realen Samples, aus dem Trainingsset. Der Diskriminator trifft eine Aussage darüber, ob der erhaltene Input real oder gefälscht ist (Kalin 2018, 15). Durch Backpropagation liefert die Klassifizierung des Diskriminators ein Signal, das zur Aktualisierung der Gewichte der beiden neuronalen Netze verwendet wird (Kalin 2018, 22). Backpropagation beschreibt einen Algorithmus zur Durchführung des Gradientenabstiegs bei neuronalen Netzen. Zunächst werden die Ausgangswerte jedes Knotens in einem Vorwärtsdurchlauf berechnet (und zwischengespeichert). Dann wird die partielle Ableitung des Fehlers in Bezug auf jeden Parameter in einem Rückwärtsdurchlauf durch den Graphen berechnet (Kalin 2018, 13)

2.2.2.1 Aufbau

Im Folgenden wird auf die einzelnen Bestandteile der GAN Struktur genauer eingegangen. Der Diskriminator ist ein einfaches Klassifikationsmodell (Kalin 2018, 18). Dieses versucht, reale Daten und Daten, die von dem Generator erzeugt werden, zu unterscheiden (Goodfellow et al. 2014, 2f.). Aus Abbildung 2 wird deutlich, dass die Trainingsdaten des Diskriminators aus zwei Datenquellen kommen. Zum einen handelt es sich um reale Daten, die der Diskriminator als positive Trainingsdaten während des Trainings verwendet und zum anderen werden die generierten Daten des Generators, die als negative Beispiele verwendet werden, genutzt. Da die Input Daten des Diskriminators gelabelt sind - das heißt, es ist klar, welcher Ausprägung sie angehören -, kann die Diskriminator-Verlustfunktion basierend auf den richtigen oder falschen Aussagen des Diskriminators angepasst werden (Langr und Bok 2019, 6). Der Diskriminator-Verlust bestraft somit den Diskriminator für die Fehlklassifizierung einer echten Instanz als falsch oder einer falschen Instanz als echt. Der Diskriminator aktualisiert seine Gewichte durch Backpropagation vom Diskriminator-Verlust (Kalin 2018, 22). Durch diese Anpassung wird der Diskriminator optimiert, so dass dieser immer zuverlässiger zwischen realen und vom Generator erzeugten Daten unterscheiden kann (Kalin 2018, 17–19).

Das Ziel des Generators besteht darin, Ergebnisse zu erzeugen, die die Eigenschaften des Trainingsdatensatzes so gut abbilden, dass die erzeugten Beispiele von den Trainingsdaten nicht zu unterscheiden sind (Goodfellow et al. 2014, 2f.). Der Generator kann als ein umgekehrtes Objekterkennungsmodell betrachtet werden (Langr und Bok 2019, 6). Objekterkennungsalgorithmen lernen die Muster in Bildern, um den Inhalt eines Bildes zu erkennen. Anstatt die Muster zu erkennen, lernt der Generator sie im Wesentlichen von Grund auf neu zu erstellen. So ist der Input des Generators im Standard GAN nicht mehr als ein Vektor von Zufallszahlen (Langr und Bok 2019, 6). Der Generator lernt durch das Feedback der Generator-Verlustfunktion, die von den Klassifizierungen des Diskriminators beeinflusst wird (Kalin 2018, 22).

Während der Generator also besser darin wird, realistisch aussehende Daten zu erzeugen, wird der Diskriminator geübter darin, gefälschte Daten von echten zu unterscheiden und beide Netzwerke verbessern sich gleichzeitig weiter (Langr und Bok 2019, 6). Sie haben jedoch konkurrierende Ziele, genauer ausgedrückt, befinden sie sich in einem Nullsummenspiel (Langr und Bok 2019, 11f.). Dies beschreibt eine Situation, in der die Gewinne des einen Spielenden dem Verlust des anderen Spielenden entsprechen. Alle Nullsummenspiele haben ein Nash-Gleichgewicht, d. h. einen Punkt, an dem keiner der Spielenden die Situation oder den Gewinn durch Änderung ihrer Aktionen verbessern kann (Langr und Bok 2019, 11). Ein GAN erreicht ein Nash-Gleichgewicht, wenn der Generator gefälschte Beispiele erstellt, die von den echten Daten im Trainings-Datensatz nicht zu unterscheiden sind. Der Diskriminator kann bestenfalls zufällig erraten, ob ein bestimmtes Beispiel echt oder gefälscht ist (Langr und Bok 2019, 11). Wenn ein GAN das Gleichgewicht erreicht hat, heißt es, dass der GAN konvergiert ist (Langr und Bok 2019, 12). In der Praxis ist es für GANs jedoch fast unmöglich das Nash-Gleichgewicht herzustellen, somit bleibt die GAN-Konvergenz eine der wichtigsten offenen Fragen in der GAN-Forschung (Langr und Bok 2019, 12).

2.2.2.2 Herausforderungen beim Erstellen synthetischer tabellarischer Daten im Finanzwesen mittels Generative Adversarial Networks

Mehrere einzigartige Eigenschaften von tabellarischen Daten stellen eine Herausforderung für den Entwurf eines GAN-Modells dar (Xu et al. 2019, 2f.). Eine Fragenstellung, die in einer realen Domäne besonders hohen Stellenwert hat, entsteht aus der Bewertung der Qualität synthetischer Daten, die mittels GANs erstellt werden. Es stellt sich die Frage, wie die Qualität synthetisch erzeugter Daten gemessen werden kann (Torfi et al. 2020, 2)? Besondere Vorsicht muss hierbei in Bereichen wie dem Finanzwesen ausgeübt werden. Bei der Verwendung von synthetischen Daten mit geringer Qualität zur Erstellung von Vorhersagemodellen, können daraus schwerwiegende Folgen entstehen. So könnte eine gewisse Personengruppe auf Grund schlecht generierter Daten bei der Kreditwürdigkeitsprüfung benachteiligt werden. Für die Banken könnten fehlerhafte Modelle zu schwerwiegenden finanziellen Fehlentscheidungen führen.

Tabellarische Daten bestehen in der realen Welt aus gemischten Datentypen. Viele der veröffentlichten GAN Strukturen haben Schwierigkeiten, wenn ein Datensatz diskrete Daten beinhaltet, da GAN-Modelle von Natur aus für die Generierung kontinuierlicher Werte ausgelegt sind (Torfi et al. 2020, 2). Insbesondere die Mischung aus kontinuierlichen und diskreten Daten stellt eine Herausforderung für die Generierung synthetischer tabellarischer Daten anhand von GANs dar (Xu et al. 2019, 3; Torfi et al. 2020, 2).

Ein weiteres Problem von tabellarischen Daten für einige GAN Strukturen sind zeitliche und lokale Korrelation zwischen den Merkmalen. Diese werden zum Teil von den Modellen ignoriert (Torfi et al. 2020, 2). Bei Finanzdaten sind jedoch häufig zeitliche Informationen und korrelierende Merkmale mitinbegriffen. Die Einbeziehung dieser Korrelationen ist in diesem Bereich wichtig, da sich zum Beispiel der Finanzstatus häufig und schnell über die Zeit der Datenerhebung hinweg verändern kann. Die Qualität der generierten Daten kann durch die Einbeziehung solcher Abhängigkeiten in die Daten erheblich verbessert werden (Torfi et al. 2020, 2).

Es ist Problematisch, dass die Verteilung eines Features bei tabellarischen Daten häufig ungleichmäßig ist (Xu et al. 2019, 3). Während bei Bildern die Pixel meist eine Verteilung ähnlich zu der Gaußschen Verteilung annehmen, ist dies für tabellarische Daten nicht der Fall. Häufig gibt es hochgradig unausgewogenen diskrete Spalten, wobei die dominante Kategorie in mehr als 90% der Zeilen vertreten ist (Xu et al. 2019, 3).

Die Herausforderung, die schlussendlich auch in diese Arbeit angesprochen wird, ist das Schützen der Privatsphäre. Die Mehrheit der vorhandenen Arbeiten trainiert das Modell nicht auf eine datenschutzfreundliche Weise, im besten Fall wird versucht, die Privatsphäre mit einigen statistischen oder auf maschinellem Lernen basierenden Messungen zu berücksichtigen (Torfi et al. 2020, 2). GANs haben sich jedoch bereits als anfällig erwiesen, so haben Hitaj et al. (2017) ein aktives Inferenz-Angriffsmodell eingeführt, das originale Trainingsdaten aus dem generierten Datensatz rekonstruieren kann (Xie et al. 2018, 1). In Bereichen wie dem Finanzwesen hat der Datenschutz jedoch besondere Bedeutung und muss explizit beachtet werden. Es ist daher wichtig, generative Modelle zu erstellen, die nicht nur qualitativ hochwertige Daten erzeugen, sondern auch die Privatsphäre der Trainingsdaten schützen (Xie et al. 2018, 1).

2.2.2.3 Differential Private – Wasserstein Generative Adversarial Network

Seit dem Beitrag von Goodfellow et al. (2014) sind viele neue Architekturen von generativen Modellen entstanden, die aufbauend auf dieser Originalidee sind wie zum Beispiel StyleGAN, der besonders realistische Bilder von menschlichen Gesichtern erstellen kann (Karras et al. 2018). CTGAN hingegen ist darauf ausgerichtet, möglichst realistische tabellarische Daten zu erstellen (Xu et al. 2019) und DiscoGAN verfügt über gute Fähigkeiten, domänenübergreifenden Beziehungen beim unbewachten Lernen zu erreichen (Kim et al. 2017). Dies ist nur eine

kleine Auswahl an entstandenen GAN Strukturen, die versuchen, einen bestimmten Forschungsbereich über generative Modelle zu verbessern.

In dieser Arbeit wird ein Differential Privacy – Wasserstein GAN (DP-WGAN) verwendet. Die Idee des Wasserstein GAN (WGAN) hatte Arjovsky et al. (2017). Diese Struktur ist eine Alternative zu traditionalem GAN Training. Ziel des WGANs ist es, die Stabilität des GAN Trainings zu verbessern und somit Probleme wie den Mode Collapse zu verringern (Arjovsky et al. 2017, 16). Mode Collapse ist ein häufiges Problem in GAN Strukturen. Dieses unerwünschte Ereignis tritt auf, wenn das Modell auf nur einige wenige Einstellungen der Gewichte fixiert wird, was zur Folge hat, dass der Generator sehr ähnliche Samples produziert. Dabei wird die Mehrheit der möglichen Verteilungen des Modells ignoriert (Durall et al. 2020, 1). Der Hauptunterschied zwischen Wasserstein- und dem original GAN ist die Verwendung einer anderen Metrik zur Quantifizierung der Ähnlichkeit zwischen zwei Wahrscheinlichkeitsverteilungen (vgl. Arjovsky et al. 2017, 2). Die Verlustfunktion des original GANs misst die Jensen-Shannon-Divergenz zwischen den Verteilungen der originalen und der vom Generator erstellen Daten (Goodfellow et al. 2014, 4f.). Diese Metrik liefert jedoch keinen sinnvollen Wert, wenn zwei Verteilungen disjunkt sind (Arjovsky et al. 2017, 4f.). In der WGAN Struktur wird dafür die „Earth-Mover“ oder auch Wasserstein-1 Distanz verwendet.

Definition 2.4: „Earth-Mover“ oder Wasserstein-1 Distanz

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}[\| x - y \|]$$

(Arjovsky et al. 2017, 4)

Dabei bezeichnet $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ die Menge aller gemeinsamen Verteilungen $\gamma(x, y)$, deren Randwerte jeweils $\mathbb{P}_r$ und $\mathbb{P}_g$ sind. $\gamma(x, y)$ gibt an, wie viel „Masse“ von x nach y transportiert werden muss, um die Verteilung $\mathbb{P}_r$ in die Verteilung $\mathbb{P}_g$ zu transformieren. Die „Earth-Mover“ Distanz sind dann die Kosten des optimalen Transportplans (Arjovsky et al. 2017, 4).

Die Einführung von Differential Privacy in den WGAN kam von Xie et al. (2018), wobei die Technik auch leicht auf andere GAN-Strukturen erweitert werden kann (Xie et al. 2018, 2). Diese Methode konzentriert sich auf die Wahrung der Privatsphäre während des Trainingsverfahrens, anstatt Rauschen zu den endgültigen Parametern hinzuzufügen (Xie et al. 2018, 3). Im DP-WGAN wird das Rauschen auf den Gradienten der Wasserstein-Distanz in Bezug auf die Trainingsdaten hinzugefügt (Xie et al. 2018, 3). Dadurch sind die Klassifizierungsergebnisse des Diskriminators differentiell privat. Da der Generator nur mittels des Diskriminators beeinflusst wird, und nicht in Kontakt mit den realen Daten gerät, folgen die generierten Ergebnisse des Generators aufgrund des „Post-Processing“ Theorems (vgl. Kapitel 2.1.2) ebenfalls den Richtlinien von Differential Privacy (Xie et al. 2018, 3). Das Ergebnis ist eine GAN Struktur, die es ermöglicht, differentiell private, synthetische Daten zu erstellen.

3 Methodische Vorgehensweise

Um die Forschungsfrage der vorliegenden Abschlussarbeit zu beantworten, erfolgt eine datenbasierte Analyse von synthetisch generierten Daten, die mit unterschiedlich starken Differential Privacy-Einschränkungen erstellt werden. Anhand von prädiktiven Modellen wird die Effektivität dieser analysiert und evaluiert. Dazu wird der von Müller et al. (2016) vorgeschlagene Referenzprozess für Datenanalysen in der Informationssystemforschung angewendet. Es ist für das weiter Verständnis dieser Arbeit nützlich, die Vorgehensweise der Datenanalyse zu verstehen. Aus diesem Grund werden die Bedeutung und Maßnahmen der vier Phasen der Datenanalyse nach Müller et al. (2016) im folgenden Kapitel zusammengefasst.

Phase eins beschreibt die angemessene Rahmung der Datenanalyse anhand der Forschungsfragen. Dabei wird zwischen zwei Ansätzen zur Modellierung unterschieden. Die erklärende Modellierung zielt darauf ab, theoriegeleitete Hypothesen unter Verwendung empirischer Daten statistisch zu testen. Prädiktive Modelle, wollen Vorhersagen über zukünftige oder unbekannte Ereignisse treffen, indem Modelle auf historischen Daten trainiert werden. Forschende müssen offen für Überraschungen in den Daten sein, sie iterativ analysieren und sich auf relevante Konzepte fokussieren, die einen Beitrag leisten können. Ein Modell soll frei und nicht gezwungenermaßen entstehen, darum sollen Forschende eine Phase der theoretischen Triangulation in Betracht ziehen, in der die verschiedenen identifizierten Konzepte mit der bestehenden Theorie verglichen werden. Das Testen der entstehenden Forschungsfrage gegen die bereits vorhandenen Theorien kann helfen, Überschneidungen sowie Forschungslücken zu identifizieren und sicherzustellen, dass die entstehende Forschungsfrage wirklich einzigartig und neuartig ist.

In der zweiten Phase wird die Datenbeschaffung beschrieben. Big Data zielt darauf ab, in Bezug auf Breite und Tiefe der Daten umfassender und in der Auflösung genauer zu sein als herkömmliche Daten, wobei in der Regel einzelne Personen indiziert werden, anstatt Daten auf Gruppenebene zu aggregieren (Kitchin 2014, zitiert nach Müller et al. 2016, 292). Diese Daten sind ein Nebenprodukt des technologischen Fortschritts und werden meist ohne einen bestimmten Zweck erhoben. Für Forscher/-innen gibt es jedoch große Unterschiede in den Zugriffsebenen auf diese Datenquellen, denn solche Daten werden meist von Unternehmen gesammelt, die diese Daten häufig nur mit ausgewählten Forschenden teilen. Im Gegensatz zu herkömmlichen Datenerhebungsmethoden, wie Interviews, Umfragen oder Experimenten, werden die Daten von Nutzenden generiert und können somit mögliche Instrumentenverzerrungen reduzieren, da Verhaltensweisen und Einstellungen auf unauffällige Weise und zu dem Zeitpunkt und in dem Kontext erhoben werden, in dem sie auftreten. In der Datenbeschaffungsphase sollte die Auswahl von Big Data als primäre Datenquelle begründet werden, die Beschaffenheit der Daten im Hinblick auf ihre Validität und Zuverlässigkeit diskutiert werden, der Datenerhebungsprozess detailliert dokumentiert und möglicherweise der Zugang zu den Daten der Arbeit gewährt werden.

Phase drei beschreibt die Datenanalyse. Um auf einen Datensatz gewisse Methoden für wissenschaftliche Arbeiten anwenden zu können, ist die Datenvorverarbeitung ein wichtiger Schritt. Dieser sorgt dafür, dass die angewendeten Methoden zu den bestmöglichen Ergebnissen führen. Diese Phase muss in dem Beitrag im Detail dokumentiert werden, damit er für Lesende der Arbeit nachvollziehbar und reproduzierbar ist. Da das Spektrum der verfügbaren Algorithmen bereits sehr umfangreich ist und sich ständig weiterentwickelt, ist hier ebenfalls die Sicherstellung der Nachvollziehbarkeit der zugrundeliegenden Algorithmen von zentraler Bedeutung. Dabei müssen sich Informationssystemforscher/-innen auf andere Disziplinen, besonders auf die Informatik und das maschinelle Lernen verlassen oder mit diesen kooperieren. Soweit es möglich ist, sollte auch der Zugang zu den für die Datenanalyse verwendeten Algorithmen gewährt werden. Um die statistische Validität der Analyse sicherzustellen, sollte eine empirische Triangulation durchgeführt werden. Dafür werden verschiedene Methoden oder Sichtweisen auf das gleiche Phänomen angewendet. Dabei sollten geeignete Bewertungskriterien gewählt werden, die eine Vergleichbarkeit mit anderen Studien gewährleisten können.

Die letzte Phase der methodischen Vorgehensweise nach Müller et al. (2016) beschreibt die Ergebnisinterpretation. Algorithmen aus dem maschinellen Lernen bieten zwar häufig eine bessere Vorhersagegenauigkeit als statistische Modelle, haben aber meist den Nachteil, dass die Vorhersagen wie eine „Black-Box“ getroffen werden (Martens und Provost 2014, zitiert nach Müller et al. 2016, 294). Es gibt keine Erklärung, warum eine gewisse Vorhersage gemacht wurde. Dies ist für bestimmte Entscheidungen in Bereichen wie dem Finanzwesen allerdings essenziell. Hier verlangen regulatorische Anforderungen, dass alle automatisierten Entscheidungen, die von Kreditwürdigkeits-Algorithmen getroffen werden, transparent und begründbar sind, um zum Beispiel die Diskriminierung von Personen zu vermeiden (Müller et al. 2016, 294). Eine explizite Erklärungsphase, die in den Forschungsprozess eingefügt wird, ist daher notwendig, um die Interpretierbarkeit von Vorhersagemodellen zu gewährleisten.

4 Datenanalyse im Kontext der Kreditwürdigkeitsprüfung

Im folgenden Kapitel wird die Methode nach Müller et al. (2016) durchgeführt und der Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen empirisch untersucht. Die vier Phasen der Methodik werden jeweils in einem eigenen Unterkapitel behandelt.

4.1 Forschungsfrage

Der Ansatz dieser Arbeit ist der erklärenden Modellierung zuzuordnen. In den theoretischen Grundlagen wird bereits ein Verständnis dafür vermittelt, dass bei einem niedrigeren Differential Privacy Parameter ε dem Datensatz mehr Rauschen hinzugeführt wird (vgl. Xie et al. 2018, 6). Diese Aussage wird anhand eines Beispiels zur Kreditwürdigkeitsprüfung im Finanzwesen empirisch untersucht. Dabei soll ein Überblick über den Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen dargestellt werden und die Forschungsfrage beantwortet werden. Die Forschungsfrage, die in den folgenden Kapiteln bearbeitet wird, lautet:

Wie lässt sich der Trade-off zwischen Effektivität und Differential Privacy bei der synthetischen Datengenerierung mittels eines generativen Modells anhand des Beispiels der Kreditwürdigkeitsprüfung im Finanzwesen bewerten?

4.2 Datenbeschaffung

Um den Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen auf eine empirische Weise bewerten zu können, ist die Durchführung anhand eines Datensatzes nötig. Daten, die in einem betriebswirtschaftlichen Kontext besonders auf Privatsphäre hin geschützt werden müssen, sind Finanzdaten (Ma et al. 2020, 1). Aus diesem Grund wurde festgelegt, dass in dieser Arbeit anhand eines Datensatzes zur Kreditwürdigkeitsprüfung im Finanzwesen gearbeitet wird. Um eine Reproduzierbarkeit der Arbeit zu ermöglichen, wird ein öffentlich zugänglicher Datensatz von Kaggle.com verwendet. Er ist Teil der Competition „Give Me Some Credit“ (Kaggle.com 2011). Das Ziel und die Motivation der Competition wird der Beschreibung des Wettbewerbs entnommen und im Folgenden kurz dargestellt.

Banken spielen in Marktwirtschaften eine bedeutende Rolle. Sie entscheiden, wer zu welchen Konditionen Finanzmittel erhält und können Investitionsentscheidungen treffen oder verhindern. Damit Märkte und die Gesellschaft funktionieren, brauchen Einzelpersonen und Unternehmen Zugang zu Krediten. Mit Hilfe von Kreditscoring-Algorithmen, die eine Schätzung der Ausfallwahrscheinlichkeit vornehmen, entscheiden Banken, ob ein Kredit gewährt wird oder nicht. Der Wettbewerb verlangt von den Teilnehmenden die Wahrscheinlichkeit vorherzusagen, ob jemand in eine finanzielle Notlage gerät und damit den Stand der Technik bei der Kreditwürdigkeitsprüfung zu verbessern.

Da der Datensatz nicht vom Ersteller der Arbeit erhoben wird, kann keine nähere Aussage über den Prozess und die Qualität der Datenerhebung getroffen werden. Es wird jedoch davon ausgegangen, dass die Validität und Zuverlässigkeit des Datensatzes gegeben ist, da dieser Teil eines Wettbewerbs mit Preisgeld war.

Der Datensatz besteht aus 150.000 Einträgen und hat zusätzlich zu der Zielvariablen zehn Features, anhand deren eine Vorhersage getroffen werden soll. Des Weiteren ist die Spalte ***Unnamed: 0*** in diesem enthalten, die eine ID des jeweiligen Eintrags beschreibt. Der Datensatz besteht aus acht Integer und vier Float-Spalten. Somit sind bereits alle Spalten in numerischer Form vorhanden, weshalb kein Encoding der Spalten durchgeführt werden muss.

Der Datensatz beinhaltet Spalten, in denen für gewisse Zeilen kein Wert verfügbar ist. So gibt es in der Spalte ***NumberOfDependents*** 3.924 Einträge und in der Spalte ***MonthlyIncome*** 29.731 Zeilen, für die kein Wert eingetragen ist. Diese fehlenden Werte können nicht unbeachtet bleiben und werden in der folgenden Datenanalyse verarbeitet.

4.3 Datenanalyse

Dieses Kapitel beschreibt Phase drei nach Müller et al. (2016) und beinhaltet den Prozess der Datenanalyse. Dafür werden die Daten in einem ersten Schritt explorativ untersucht und vorbereitet. Im Anschluss wird die Wahl der Evaluationsmetriken und die Durchführung der synthetischen Datengenerierung beschrieben. Es folgt ein Unterkapitel zur Modellentwicklung und eine Evaluation der Ergebnisse.

4.3.1 Explorative Datenanalyse und Datenvorbereitung

Die Zielvariable heißt ***SeriousDlqin2yrs*** und beschreibt, ob eine Person innerhalb der nächsten zwei Jahren eine schwerwiegende Delinquenz (Zahlungsverzug von mindestens 90 Tagen) aufweisen wird. Die Zielvariable ist in dem Datensatz ungleichmäßig verteilt. So wird der Großteil der Teilnehmenden nicht in Verzug geraten und kann den Kredit ordnungsmäßig zurückzahlen.

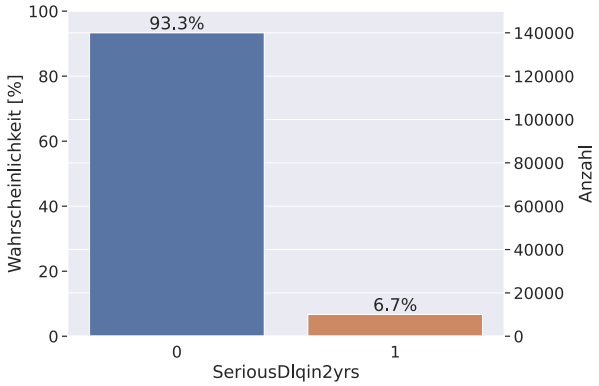

Abbildung 3: Verteilung der Zielvariable *SeriousDlqin2yrs*

Ein Wert von 0 für die Zielvariable bedeutet, dass es keine schwerwiegende Delinquenz geben wird. Sollte in einer Zeile dieser Wert auf 1 stehen, wird diese Person innerhalb der nächsten zwei Jahren in einen Zahlungsverzug von mindestens 90 Tagen geraten. Abbildung 3 zeigt, dass die Wahrscheinlichkeit in dieser Spalte für den Wert 0 bei 93,3% liegt und somit in etwa jeder 14te Teilnehmende der Datenbank in Verzug gerät.

Um ein Machine Learning Modell zu trainieren, sind zehn Features gegeben, die Informationen enthalten können, ob eine Person in Verzug geraten wird. In Anhang 1 kann ein Überblick der Features mit jeweiliger Kurzbeschreibung gefunden werden. Im Folgenden werden diese Features einzeln untersucht. Falls dabei Ausreißer identifiziert werden können, werden diese direkt verarbeitet und beeinflussen damit auch die Analyse des darauffolgenden Features.

Als erstes wird die einfach verständliche Spalte ***age***, die das Alter der Teilnehmenden der Datenbank beschreibt, analysiert.

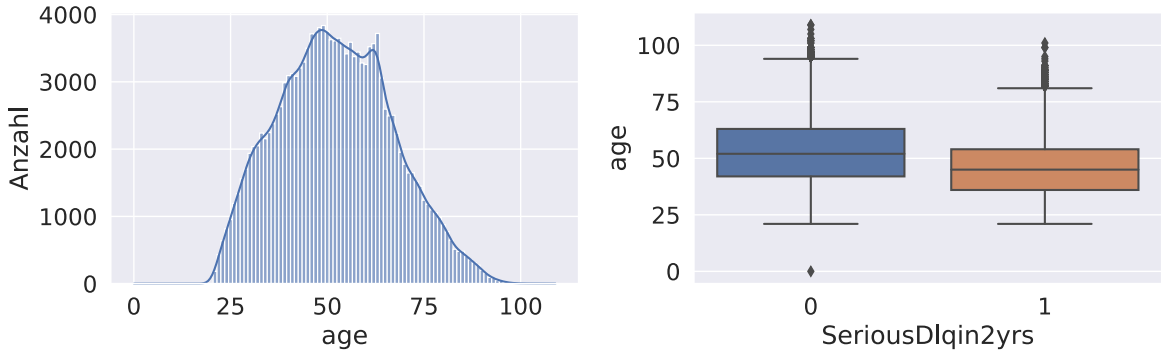

Abbildung 4: Verteilung des Features *age*

Das Durchschnittsalter liegt hier bei 52,3 Jahren. Bei der Verteilung des Features in Bezug auf die Zielvariable ist auffällig, dass vermehrt jüngere Personen dazu neigen, in den nächsten zwei Jahren in Verzug zu geraten. Des Weiteren ist in der rechten Grafik von Abbildung 4 zu erkennen, dass es Ausreißer gibt, bei denen ein Alter von 0 Jahren eingetragen ist. Bei genauerer Betrachtung fällt auf, dass es sich hierbei um exakt einen Eintrag handelt. Dieser wird ohne weitere Beachtung vom Datensatz entfernt.

Die drei Spalten ***NumberOfTime30-59DaysPastDueNotWorse***, ***NumberOfTime60-89DaysPastDueNotWorse*** und ***NumberOfTimes90DaysLate*** sind in ihrer Bedeutung annähernd identisch. Sie beschreiben die Anzahl der Fälle, in denen der Kreditnehmer in den letzten 2 Jahren mit der Zahlung überfällig war und unterscheiden sich lediglich in dem Zeitraum. Bei genauerer Betrachtung der drei Spalten ist auffällig, dass sich das 99%-Quantil bei allen drei im einstelligen Bereich bewegt. Der Maximalwert liegt jedoch jeweils bei 98. Werden alle möglichen Ausprägungen der drei Spalten angezeigt, ist deutlich, dass die Werte zwischen dem Intervall [0; 17] liegen. Nach einer großen Lücke folgen dann die Ausprägungen 96 und 98.

Insgesamt gibt es 269 Zeilen, bei denen die drei Spalten identisch mit 96 oder 98 gefüllt sind. Diese Einträge können als Fehleinträge klassifiziert werden. Es ist jedoch auffällig, dass die Zielvariable hier mit einer durchschnittlichen Wahrscheinlichkeit von 54,6% deutlich häufiger auftritt als im Normalfall. Aus diesem Grund werden die Zeilen nicht als nutzlos angesehen. Die Fehleinträge werden mit dem jeweiligen Maximalwert aus den drei Spalten, der kleiner als 96 ist, aufgefüllt. Für ***NumberOfTime30-59DaysPastDueNotWorse*** entspricht dies 13, ***NumberOfTime60-89DaysPastDueNotWorse*** mit 11 und für ***NumberOfTimes90DaysLate*** 17.

Der Verschuldungsgrad, als Spalte ***DebtRatio*** in diesem Datensatz, beschreibt die monatlichen Schuldenzahlungen, Alimente und Lebenshaltungskosten geteilt durch das monatliche Bruttoeinkommen. Ein Verschuldungsgrad von 1 bedeutet somit, dass sämtliche Ausgaben exakt dem Bruttoeinkommen entsprechen. Das 75% Quantil hat einen Wert von 0.87 und erscheint demnach realistisch. Der Maximalwert der Spalte mit 329.664 lässt jedoch auf Eintragungsfehler deuten. Bei genauerer Betrachtung der Einträge mit einem Verschuldungsgrad von größer als 1 ist auffällig, dass die Spalte ***MonthlyIncome***, die das monatliche Einkommen der Person beschreibt, für nur 7.233 von insgesamt 35.137 Einträgen einen Wert hat. Des Weiteren ist das 25%-Quantil der Spalte ***MonthlyIncome*** für die übrigen Spalten, die einen Wert eingetragen haben, 1. Dies könnte bedeuten, dass das monatliche Einkommen bei diesen Einträgen nicht bekannt ist, oder dass diese Personen wirklich kein Einkommen haben. Umgekehrt betrachtet nimmt das ***DebtRatio***, wenn das Monatseinkommen nicht bekannt, 0 oder gleich 1 ist, zu mehr als 99% einen anormalen Wert an. Der Wert ist nicht in dem Bereich eines normalen Prozentsatzes, sondern entweder gleich 0 oder größer 1. Dieser Zusammenhang lässt sich dadurch erklären, dass der Verschuldungsgrad abhängig vom Einkommen einer Person ist. Möglicherweise stellt sich beim ***DebtRatio***, sobald kein Einkommen bekannt ist, der Rohwert der Schulden dar, wodurch die vielen hohen Werte in dieser Spalte erklärt werden könnten. Wenn der Wert in dieser Spalte 0 ist, könnte das daran liegen, dass auch der rohe Wert der Schulden unbekannt ist. Der Durchschnitt der Zielvariablen ***SeriousDlqin2yrs*** bleibt bei den Einträgen ohne monatliches Einkommen im Vergleich zum gesamten Datensatz ohne große Veränderung. Die Variable hat bei den genannten Zeilen einen Durchschnitt von 5,61%. Es gäbe nun die Möglichkeit, die fehlenden und falsch eingetragenen Werte mit dem Median zu füllen. Des Weiteren wäre es denkbar, anhand eines prädiktiven Modells zu versuchen, für die betroffenen Spalten die fehlenden Einträge zu generieren. Da sich jedoch die Wahrscheinlichkeit der Zielvariable kaum vom restlichen Datensatz unterscheidet, die fehlenden und falsch eingetragenen Werte mehrere Spalten betreffen und für diese Arbeit ein realistisches Datenset wichtig ist, werden Einträge, für die kein oder nur ein anormaler Wert in den Spalten ***MonthlyIncome*** oder ***DebtRatio*** angegeben ist, gelöscht.

Dafür werden in einem ersten Schritt alle Einträge, die bei ***MonthlyIncome*** keinen Wert haben, entfernt. Als nächstes werden Zeilen gelöscht, die für die Spalte ***DebtRatio*** einen Wert größer als 10 haben. Dieser Wert wurde vom Verfasser der Arbeit festgelegt. Ein ***DebtRatio*** von 10

würde bedeuten, dass die monatlichen Schuldenzahlungen, Alimente und Lebenshaltungskosten zehnfach so hoch sind, wie das monatliche Bruttoeinkommen der Person. Einträge die größer als 10 sind, erscheinen somit sehr ungewöhnlich und sind nicht zu erwarten. Da die Wahrscheinlichkeit, dass diese Einträge innerhalb der nächsten zwei Jahre eine schwerwiegende Delinquenz erleiden, geringer als bezogen auf den gesamten Datensatz ist, werden diese als Fehleinträge klassifiziert und vom Datensatz entfernt. Denn die Einträge beschreiben eher den Rohwert der Schuldzahlungen als den Verschuldungsgrad. Als letztes werden alle Zeilen entfernt, in denen das ***DebtRatio*** und das ***MonthlyIncome*** jeweils den Wert 0 enthält.

Nachdem die Spalte ***MonthlyIncome*** bereinigt ist, wird nun die Korrelation zwischen dieser und der Zielvariablen analysiert. Leider ist über den Datensatz nicht bekannt, in welchem Land die Datenerhebung durchgeführt wurde oder welche Währung für das monatliche Einkommen verwendet wird. Aus diesem Grund kann auch in dieser Arbeit keine Aussage darüber getroffen werden und die Werte aus dieser Spalte werden ohne Währung angegeben.

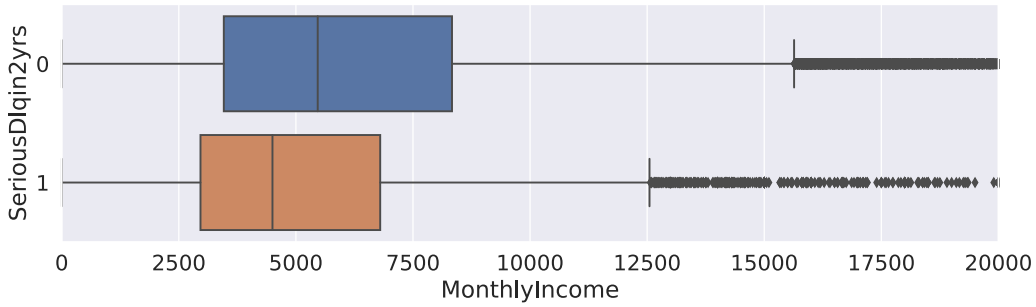

Abbildung 5: Verteilung des Features *MonthlyIncome*

Die x-Achse aus Abbildung 5 wurde auf ein maximales monatliches Einkommen von 20.000 gesetzt. Sonst ist die Grafik auch unter logarithmischer Skala aufgrund einiger Ausreißer, die ein sehr hohes monatliches Einkommen haben, nicht lesbar. Wie zu vermuten und aus Abbildung 5 abzulesen, neigen Personen mit einem niedrigeren Einkommen eher dazu, den Kredit nicht zurückzahlen zu können. Des Weiteren ist jedoch auch sichtbar, dass der Unterschied nicht immens groß ist und auch einzelne Personen mit einem überdurchschnittlich hohen Einkommen, die Schulden nicht immer begleichen können. Dies könnte daran liegen, dass in dieser Betrachtung, die Höhe der vergebenen Kredite nicht beachtet wird. So werden besserverdienende Personen vermutlich häufiger einen durchschnittlich höheren Kredit anfragen als Personen mit einem niedrigeren Einkommen.

Das Feature ***RevolvingUtilizationOfUnsecuredLines*** beschreibt den Gesamtsaldo von Kreditkarten und persönlichen Kreditlinien außer Immobilien und Ratenschulden wie Autokredite, geteilt durch die Summe der Kreditlimits. Es stellt also grundsätzlich das Verhältnis des geschuldeten Betrags zum Kreditlimit einer Person dar. Solange das Kreditlimit nicht

überschritten ist, ist dieser Wert kleiner oder gleich 1. Das 75%-Quantil mit einen Wert von 0.58 erscheint somit realistisch. Jedoch lässt auch hier der Maximalwert mit 50.708 darauf deuten, dass Fehleinträge in dieser Spalte existieren. Bei genauerer Betrachtung ist zu erkennen, dass ab einem Wert von 10 die Einträge in dieser Spalte enorm ansteigen. Ein Wert von 10, könnte theoretisch möglich sein, jedoch erscheint es unwahrscheinlich in Bezug darauf, dass die Zielvariable bei den Einträgen, die ihr Kreditlimit über das zehnfache überziehen, nicht deutlich ansteigt. Aus diesem Grund werden die übrigen 170 Einträge, mit einem Wert von 10 oder höher in der Spalte ***RevolvingUtilizationOfUnsecuredLines*** aus dem Datensatz entfernt.

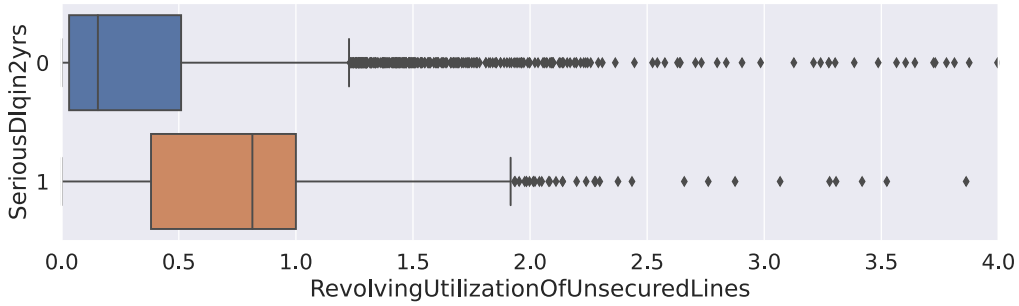

Abbildung 6: Verteilung des Features *RevolvingUtilizationOfUnsecuredLines*

Bei der Verteilung der Spalte ***RevolvingUtilizationOfUnsecuredLines*** lässt sich aus Abbildung 6 ein eindeutiger Trend erkennen. Personen, die an oder auch über die Grenze ihres Kreditlimits gehen, neigen eher dazu, in den nächsten zwei Jahren in Verzug zu geraten.

Die Spalte ***NumberOfDependents*** beschreibt die Anzahl der unterhaltsberechtigten Personen in der Familie, ausgenommen der eigenen Person, also zum Beispiel Ehepartner/-in und Kinder. In dieser Spalte gab es wie während der Datenbeschaffung festgestellt, 3.924 Einträge ohne einen Wert. Nachdem der Datensatz in den vorherigen Schritten von einigen Ausreißern und den leeren Zeilen aus ***MonthlyIncome*** bereinigt wurde, sind diese Zeilen ebenfalls gelöscht worden. So befindet sich im aktuellen Datensatz keine Zeile mehr, mit einem fehlenden Wert.

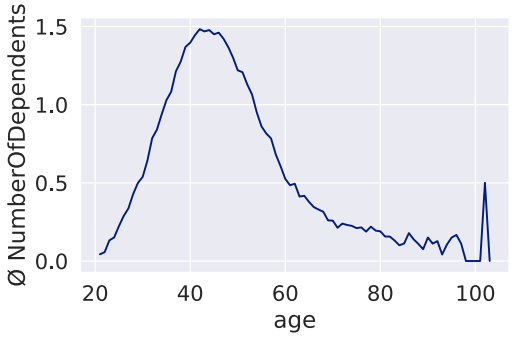

Abbildung 7: Verhältnis *age* zu Durchschnitt *NumberOfDependents*

Im Vergleich der Variablen ***NumberOfDependents*** und ***age*** (Abbildung 7) ist auffällig, dass besonders Teilnehmer-/innen in den mittleren Jahren eine höhere Anzahl an unterhaltsberechtigten Personen in der Familie haben. Das liegt vermutlich daran, dass junge Personen noch keine Kinder haben und die Kinder von älteren Personen nicht mehr unterhaltsberechtigt sind. Der Ausreißer in Abbildung 7 bei einem ***age*** von 102 liegt daran, dass es nur zwei Personen mit diesem Alter in der Datenbank gibt. Eine Person davon hat eine unterhaltberechtigte Person in der Familie, also ist der Durchschnitt hier bei 0,5. Die Werte, in der Spalte ***NumberOfDependents*** nehmen alle eine plausible Form an. Es gibt lediglich zwei Teilnehmer-/innen, die in dieser Spalte eine besonders hohe Anzahl von 13 und von 20 haben. Es wird jedoch angenommen, dass dies mögliche und reale Werte sind. Aus diesem Grund müssen für dieses Feature keine Ausreißer verarbeitet werden.

Als letztes gibt es die Spalten ***NumberOfOpenCreditLinesAndLoans*** und ***NumberRealEstateLoansOrLines***. Erstere beschreibt laut der Spaltenbeschreibung in Kaggle.com die Anzahl an offenen Krediten z. B. Ratenzahlungen wie Autokredite oder Hypotheken und offenen Kreditlinien wie z. B. Kreditkarten. Zweitere beinhaltet die Anzahl der Hypotheken- und Immobilienkredite einschließlich Home-Equity-Kreditlinien. Beide Spalten würden somit Hypotheken umfassen. Anhand der Spaltenbezeichnung kann man eher davon ausgehen, dass Hypotheken nur in ***NumberRealEstateLoansOrLines*** mitgezählt werden und es sich bei der Beschreibung um einen Fehler handelt. Leider können hierüber keine weiteren Informationen gefunden werden. Für den weiteren Verlauf stellt dies jedoch keine Einschränkung dar.

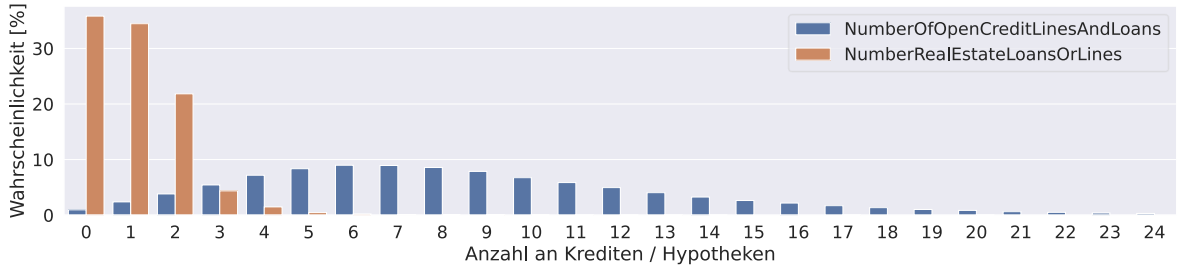

Abbildung 8: Verteilung von *NumberOfOpenCreditLinesAndLoans* und *NumberRealEstateLoansOrLines*

Abbildung 8 wurde ebenfalls auf der x-Achse um ein paar Einträge gekürzt, um die Grafik besser erkennbar zu machen. Die maximale Anzahl an offenen Krediten und Kreditlinien einer Person in dieser Datenbank liegt bei 58. In Abbildung 8 wird deutlich, dass die Teilnehmenden des Datensatzes am häufigsten sechs oder sieben offene Kredite haben. Durchschnittlich haben die teilnehmenden Personen 8,87 offene Kredite. Die Verteilung der Anzahl an Hypotheken (***NumberRealEstateLoansOrLines***) ist deutlich rechtsschief. So haben die meisten Personen keine oder nur eine offene Hypothek. Insgesamt liegt die durchschnittliche Anzahl an Hypotheken bei 1,06 und die maximal Anzahl bei 54. Für beide Spalten sind für alle Zeilen Werte

vorhanden, welche ebenfalls realistisch erscheinen. Somit müssen hier keine Zeilen verarbeitet werden und der Datensatz ist nun von allen Ausreißern und nicht verfügbaren Werten bereinigt.

Insgesamt sind noch 117.886 Einträge im Datensatz vorhanden. Aus diesen Einträgen werden nun 25% der Daten entnommen, die zum Testen und Evaluieren der erstellten Modelle genutzt werden. Somit stehen noch 88.414 Einträge zur Verfügung, auf Basis deren Modelle trainiert und synthetische Daten generiert werden.

4.3.2 Evaluationsmetriken

Der Use Case in dieser Arbeit ist eine Klassifizierung, ob eine Person in den nächsten zwei Jahren eine finanzielle Delinquenz erleiden wird oder nicht. Die Klassifikation ist somit binär, wobei 0 bedeutet, dass diese in keine schwerwiegenden finanziellen Schwierigkeiten geraten wird und 1 dem Gegensatz entspricht. Die Zielvariable ist in dem Datenset stark ungleichmäßig verteilt. So würde ein Modell, das für alle Einträge einen Wert von 0 vorhersagt mit einer Genauigkeit von 93,3% (Wahrscheinlichkeit, dass der Eintrag eine 0 ist) richtig liegen. Die Genauigkeit ist deshalb kein gutes Maß für die Evaluation der Modelle.

Eine weitere Möglichkeit zum Auswerten der Vorhersageleistung eines Klassifikators ist das Betrachten einer Konfusionsmatrix (Géron 2020, 93–95). In dieser wird das Ergebnis des Modells in eine der folgenden vier Kategorien eingeteilt:

- RN (richtig Negativ): Eintrag ist korrekt als 0 eingestuft
- FP (falsch Positiv): Eintrag ist 0, wird vom Modell fälschlicherweise als 1 eingestuft
- FN (falsch Negativ): Eintrag ist 1, wird vom Modell fälschlicherweise als 0 eingestuft
- RP (richtig Positiv): Eintrag ist korrekt als 1 eingestuft

(vgl. Géron 2020, 93–95; Langr und Bok 2019, 41f.).

Mit der Konfusionsmatrix kann man einen genauen Überblick über die Klassifizierung der Modelle schaffen. In den meisten Fällen wird jedoch ein etwas kompakteres Qualitätsmaß benötigt (Géron 2020, 94f.). Aus den Kategorien der Konfusionsmatrix lassen sich verschiedene Leistungsmetriken ableiten. Ein Maß, um die Genauigkeit der positiven Vorhersagen zu bestimmen, nennt sich Relevanz (Precision). Es ermöglicht eine Aussage, welcher Anteil der positiven Identifikationen auch tatsächlich korrekt ist (Géron 2020, 94f.). Angewendet auf einen Kreditscoring-Algorithmus beschreibt es, welcher Anteil der Einträge, für die eine finanzielle Notlage vorhergesagt wird, korrekt sind. Die Relevanz steht üblicherweise mit einem zweiten Maß namens Sensitivität (Recall) in Zusammenhang. Diese ist der Anteil positiver Datenpunkte, die vom Klassifikator entdeckt wurden (Géron 2020, 95). Die Sensitivität beschreibt somit, wie viele von den Einträgen, die in Verzug geraten, vom Modell entdeckt werden.

Definition 4.1, 4.2: Relevanz und Sensitivität

$$Relevanz = \frac{RP}{RP + FP} \qquad Sensitivität = \frac{RP}{RP + FN}$$

(Géron 2020, 94f.)

Wenn das Ziel ist, die Anzahl der Fehlalarme zu minimieren, muss eine hohe Relevanz gewählt werden. Ist es wichtig, möglichst viele positive Einträge zu identifizieren, ist die Sensitivität von höherer Wichtigkeit. Leider kann nicht beides gleichzeitig erzielt werden, ein Erhöhen der Relevanz senkt die Sensitivität und umgekehrt (Géron 2020, 97). Ein Betrachten der Relevanz oder der Sensitivität allein ist im Normalfall nicht sinnvoll. Sollte ein Algorithmus beispielsweise nur auf die Relevanz hin optimiert werden, kann dieser mit einer einzigen positiven wahren Vorhersage einen perfekten Wert für die Relevanz von 1 erhalten (Géron 2020, 97f.). Das Interesse liegt eher auf einer Kombination dieser Metriken. Des Weiteren ist es bei einem Vergleich zwischen zwei oder mehr Klassifikatoren von Vorteil, dies anhand einer einzigen Metrik zu erledigen. So ist es möglich, diesen Wert direkt für eine Aussage zu verwenden, welches Modell besser oder schlechter ist.

Der F_1-Score ist der harmonische Mittelwert von Relevanz und Sensitivität. Während der gewöhnliche Mittelwert alle Werte gleichbehandelt, erhalten beim harmonischen Mittelwert niedrigere Mittelwerte ein weitaus höheres Gewicht. Daher erhält ein Klassifikator nur dann einen hohen F_1-Score, wenn sowohl Relevanz als auch Sensitivität hoch sind (Géron 2020, 96).

Definition 4.3: F_1-Score

$$F_1 = 2 * \frac{Relevanz * Sensitivität}{Relevanz + Sensitivität}$$

(Géron 2020, 96)

Eine weitere Metrik, der binären Klassifikation ist die ROC-Kurve (Receiver Operating Characterisitc). Diese ist gemäß Dastile et al. (2020, 14) nach der Genauigkeit, die für diesen Datensatz nicht anwendbar ist, die am häufigsten verwendete Metrik in Arbeiten zu Kredit-Scoring. Anstatt Relevanz gegen Sensitivität zu vergleichen, zeigt die ROC-Kurve die Richtig-positiv-Rate (TPR), die lediglich ein anderer Name für Sensitivität ist, gegen die Falsch-positiv-Rate (FPR) (Géron 2020, 100). Die FPR beschreibt den Anteil negativer Datenpunkte, die fälschlicherweise als positiv eingestuft werden (Géron 2020, 100). Um Klassifikatoren zu vergleichen, wird die Größe des Bereichs unter der Kurve, genannt AUC (Area under the Curve), gemessen (Géron 2020, 100f.). Ein völlig zufälliger Klassifikator kann nicht zwischen den zwei Raten unterscheiden und dessen Kurve gleicht somit einem linearen Verlauf von (0,0) bis (1,1). Damit hat die Fläche unter der „Kurve“ eines zufälligen Klassifikators einen AUROC (Area under the Receiver Operating Characteristic Curve) von 0,5, während ein perfekter Klassifikator einen Wert von genau 1 erreicht (Géron 2020, 101).

Definition 4.4: AUROC

$$AUROC = \frac{1}{2}\left(1 + \frac{RF}{FP + RN} - \frac{RP}{RP + FN}\right)$$

(Dastile et al. 2020, 10)

Da die ROC-Kurve, abgesehen von der Genauigkeit, die am häufigsten verwendete Metrik in Machine Learning Arbeiten über Kredit-Scoring ist und da diese ebenfalls häufig in Arbeiten, die dieser ähnlich sind genutzt wird (z. B. Torfi et al. 2020; Jordon et al. 2019; Xie et al. 2018), wird diese auch hier als Hauptentscheidungskriterium eingesetzt um eine Vergleichbarkeit der Studien zu gewährleisten. Der F_1-Score wird zusätzlich bei der Evaluation berücksichtigt.

4.3.3 Synthetische Datengenerierung

Im folgenden Kapitel wird auf den Prozess der Datengenerierung eingegangen. Diese gestaltet sich mit den Anforderungen an diese Arbeit äußerst schwierig. Eine Eigenimplementierung des Verfahrens stellt aufgrund des Umfangs und des möglichen Fehlerpotentials hier keine Option dar. Es wurde somit nach einem Package gesucht, anhand dessen dieser Schritt durchgeführt werden kann. An die Implementierung wurden folgende Anforderungen gestellt:

- Generatives Modell aufbauend auf der GAN-Struktur
- Anwendbar auf tabellarische Daten
- Beachten von Differential Privacy

Es gibt viele unterschiedliche Implementierungen für die jeweils einzelnen Anforderungen oder auch für Kombinationen aus diesen. Da es sich hierbei jedoch um ein sehr neues Forschungsgebiet handelt (der DP-WGAN wurde von Xie et al. (2018) vorgeschlagen), wurde keine weit verbreitete und angesehene Implementierung gefunden, welche alle drei Anforderungen erfüllt. Es wurde allerdings ein vielversprechendes, freies und einfach zugängliches Package gefunden, welches den Anforderungen entspricht. Hierbei handelt es sich um die „Private Data Generation Toolbox“ von BorealisAI (2019) auf GitHub. Dieses beinhaltet fünf Implementierungen für generative Modelle, die Differential Privacy beachten. Davon sind zwei auf der GAN-Struktur aufgebaut. Eine Implementierung basiert auf dem Framework „Private Aggregation of Teacher Ensembles“ (PATE), die von Jordon et al. (2019) auf die GAN Struktur angewendet wurde und sich PATE-GAN nennt. Als zweite Implementierung ist ein Differential Privacy-Wasserstein GAN in der Toolbox vorhanden. Dieser ist bereits aus den theoretischen Grundlagen bekannt. Es wurde die Anwendung des DP-WGAN gewählt, da ein Wasserstein GAN stabileres Training ermöglichen soll (siehe Kapitel 2.2.2.3) und da dieser in der verwendeten Implementierung auch im „non private“ Modus gestartet werden kann. Es ist somit möglich, einen Datensatz, ohne die Privatsphäre-Einschränkungen von Differential Privacy zu generieren. Damit kann im Anschluss besser der Trade-off von Effektivität und DP analysiert werden.

Zuerst wurde versucht, Daten anhand der unskalierten Originaldaten zu generieren. Die Ergebnisse des DP-WGAN waren hierbei selbst im „non private“ Modus, trainiert in bis zu 3000 Epochen, unbrauchbar. Die generierten Daten waren zum größten Teil in einem Bereich von [-1, 1], jedoch gab es hier auch einige Einträge, die mit Werten bis zu 20 weit außerhalb dieser Skala lagen. Diese Verteilung ist jedoch weit von den Originalwerten entfernt. Da die Daten keiner bestimmten Skalierung folgen, z. B. nur zwischen [-1, 1], war es nicht möglich, diese Daten auf die Verteilung der Originaldaten zu skalieren. Umgekehrt konnten die Originaldaten nicht auf diese Skala verteilt werden. Das verwendete Package hat für die generierten Daten einige Downstream Classifier implementiert, die einen direkten Überblick über die Funktionalität der Daten geben. Im beschriebenen Beispiel, ohne Privatsphäre-Einschränkungen mit 3000 Epochen, lieferten die Classifier einen AUROC von ca. 0,5. Dies entspricht der Qualität eines Dummy Classifier, der nichts über die Daten lernt, sondern lediglich zufällige Werte schätzt. Daran lässt sich erkennen, dass die generierten Daten bei originaler Skalierung keinen Nutzen bringen.

Im Anschluss wurde versucht die Originaldaten, bevor sie vom DP-WGAN zum Generieren von Daten verwendet werden, zu skalieren. Es werden hier die von Scikit-Learn (Pedregosa et al. 2011) im Modul „Preprocessing“ verfügbaren Skalierungsmethoden getestet. Die Evaluierung wird anhand der Downstream Classifier der „Private Data Generation Toolbox“ durchgeführt. Dafür wird der DP-WGAN mit dem jeweils skalierten Datenset im „non private“ Modus für 700 Epochen trainiert. Die Downstream Classifier erstellen im Anschluss ein Modell auf den synthetischen Daten, die mit dem AUROC Score auf das reale Hold-out Set getestet werden. Die genauen Resultate dieses Tests können in Anhang 2 nachgelesen werden. Der Test hat gezeigt, dass unter vorheriger Skalierung, der DP-WGAN verwendbare Ergebnisse liefert. Den höchsten AUROC Score erreicht das generative Modell, bei vorherige Skalierung mit dem StandardScaler. Das Standardisieren von Features wird erreicht, indem die Werte so verschoben werden, dass der Mittelwert der Spalte gleich 0 und die Standardabweichung eine Einheitsvarianz von 1 erlangt (scikit-learn 2020a). Da diese Skalierung bei den Downstream Classifiern die besten Werte erreicht hat, werden für die Generierung und Evaluierung der synthetischen Daten standard-skalierte Daten verwendet.

Darauffolgenden werden Daten mit Differential Privacy generiert. Wie in den theoretischen Grundlagen erklärt, ist die eingefügte Menge an Rauschen in das Datenset nicht trivial. So kann der DP-WGAN nicht für eine bestimmte Anzahl an Epochen gestartet werden, um eine bestimmte Größe von Differential Privacy ε zu erreichen. Das Rauschen wird schrittweise dem Datensatz zugefügt. Mit jeder Epoche, die der DP-WGAN trainiert, verwendet dieser erneut die Originaldaten. Aufgrund der Kompositionseigenschaft von DP steigt somit bei jeder trainierten Epoche die Größe von ε an. In dieser Implementierung kann ein Zielwert für ε festgelegt werden. Der Algorithmus läuft dann dementsprechend viele Epochen, bis das gewünschte ε erreicht ist. Die Anzahl der Epochen kann jedoch beeinflusst werden, indem unterschiedliche Werte für

Sigma, das den Varianzmultiplikator für das Gaußsche Rauschen beschreibt, eingesetzt werden. Ein größeres Sigma führt dazu, dass in einer Epoche mehr Rauschen hinzugefügt wird und der Wert für ε pro Epoche, kleiner ist (entspricht mehr Privatsphäre der Teilnehmenden). Das Modell trainiert somit bei größerem Sigma, für gleiches ε, mehr Epochen. Es wurde versucht, Sigma für den jeweiligen ε-Wert so zu wählen, dass die Anzahl der Trainingsepochen nahe an dem liegt, was ein „non-private" Training des Modells für eine zufriedenstellende Leistung erfordert. Aus dem Code der verwendeten DP-WGAN Implementierung wird deutlich, dass sich während des Trainings der erreichte ε-Wert aus den variablen Parametern Batchgröße, Sigma und DP-δ berechnet. DP-δ muss wie in Kapitel 2.1.1 erklärt, kleiner als der Kehrwert der Datenbankgröße sein. Somit muss für δ ein Wert, der kleiner als $\frac{1}{88.414}$ ist, gewählt werden. Hier wird der Wert von δ auf 10^{-5} gesetzt. Das Erstellen von Datensätzen mit (ε,0)-DP, also für δ ein Wert von 0, ist in dieser Implementierung nicht möglich. Mit einer sehr kleinen Batchgröße und einem sehr großen Sigma, beginnt der Wert von ε in der ersten Epoche bei etwas über 0,01. Der kleinste Wert für ε, der mit diesem Package und den verwendeten Daten erreicht werden kann, ist somit in etwa 0,02.

Der Clip Koeffizient, die Lower- und die Upper Clamp des Wasserstein GAN werden auf den Standardwerten des Packages von 0,1 und [-0,01; 0,01] gelassen.

4.3.4 Modellentwicklung

Im Folgenden werden die in dieser Arbeit verwendeten deskriptiven Modelle zur Evaluation der entstehenden Datensätze erklärt. Wie bereits in Kapitel 3 dargestellt, ist für die methodische Vorgehensweise die Sicherstellung der Nachvollziehbarkeit der zugrundeliegenden Algorithmen von zentraler Bedeutung (Müller et al. 2016, 293). Bei der Kreditwürdigkeitsprüfung handelt es sich um ein überwachtes Lernproblem (supervised learning), genauer gesagt um ein binäres Klassifizierungsproblem. Für diese Klassifikation werden die Implementierungen der Modelle aus der Library Scikit-Learn (Pedregosa et al. 2011) verwendet. Die Beschreibungen der einzelnen Modelle werden auf der Dokumentation von Scikit-Learn (scikit-learn 2020b) aufgebaut.

Das erste deskriptive Modell, welches in dieser Arbeit verwendet wird, ist die logistische Regression. Das Lernverfahren ist vermutlich aufgrund der Interpretierbarkeit seiner Entscheidung das meistverwendete statistische Modell in der Kreditwürdigkeitsprüfung (Dastile et al. 2020, 6). Die logistische Funktion ist eine sigmoide Funktion, die eine s-förmige Ausprägung hat und zwischen 0 und 1 liegt. An dieser Funktion kann man die Wahrscheinlichkeit eines Ereignisses ablesen. Ohne weitere Einstellung liegt die Entscheidungsgrenze bei 50%. Die logistische Regression kann mittels der maximalen Anzahl an Iterationen und dem Kehrwert der

Regularisierungsstärke optimiert werden. Des Weiteren kann die Gewichtung der einzelnen Klassen gesetzt werden.

Neben dem statistischen Modell werden auch maschinelle Lernverfahren für die Evaluierung der Daten verwendet. Der KNeighborsClassifier ist eine Art des Instanz-basierten Lernens. Es wird nicht versucht ein allgemeines internes Modell zu konstruieren, sondern es werden Instanzen der Trainingsdaten gespeichert. Die Klassifikation wird aus einer einfachen Mehrheitsabstimmung der nächsten Nachbarn jedes Punktes berechnet. Als Hyperparameter dieses Modells kann die Anzahl der betrachteten Nachbarn unterschieden werden. Als nächstes Modell wird ein DecisionTreeClassifier verwendet. Das Ziel eines Entscheidungsbaums ist es, aus den Datenmerkmalen Entscheidungsregeln abzuleiten, anhand deren es möglich ist den Wert der Zielvariablen vorherzusagen. Der Classifier kann mittels der maximalen Tiefe der entstehenden Bäume getunt werden. Durch die Einführung von Zufälligkeit wird beim RandomForestClassifier versucht, die Varianz eines einzelnen Entscheidungsbaums zu verringern. Die Vorhersage des Modells wird als die gemittelte Vorhersage der einzelnen Klassifikatoren angegeben. Als letzter Klassifizierungsalgorithmus wird ein GradientBoostingClassifier verwendet. Dieser baut in einem iterativen Prozess ein additives Modell auf. Es beruht auf der Intuition, dass das bestmögliche nächste Modell, wenn es mit den vorherigen Modellen kombiniert wird, den gesamten Vorhersagefehler minimiert. Zunächst wird ein erstes Modell an die Daten angepasst. Dann wird ein zweites Modell erstellt, das sich auf die genaue Vorhersage der Fälle konzentriert, in denen das erste Modell schlecht abschneidet. Diese Modelle werden danach zusammengefügt und es wird erwartet, dass die Kombination dieser beiden Modelle besser ist als das jeweilige Modell allein. Dieser Prozess wird mehrfach wiederholt. Jedes nachfolgende Modell versucht, die Unzulänglichkeiten der kombinierten vorherigen Modelle zu korrigieren. Die einzelnen kleinen Modelle sind im GradientBoostingClassifier Regressionsbäume. Der Classifier kann mittels der Lernrate optimiert werden. Diese besagt, wie groß der Einfluss eines neuen Regressionsbaum ist. Des Weiteren kann die Anzahl der auszuführenden Boosting-Stufen gesetzt werden. Es wird zusätzlich ein DummyClassifier in die Evaluation miteinbezogen. Dieser gibt mit Beachten der vorherrschenden Verteilung zufällige Klassifizierungen aus.

Für das Hyperparametertuning wird, ebenfalls aus der Scikit-Learn Library, die Funktion GridSearchCV verwendet, der ein Machine Learning Modell übergeben wird, welches optimiert werden soll. Des Weiteren werden für dieses Modell passende vordefinierte Werte für die Hyperparameter angegeben. GridSearchCV wendet alle Kombinationen der angebotenen Hyperparametern an und bewertet das Modell für jede Kombination mithilfe der Kreuzvalidierung. Die Kreuzvalidierung spaltet den Trainingsdatensatz bei den hier verwendeten Standardeinstellungen zufällig in fünf unterschiedliche Teilmengen, genannt Folds, auf. Im Anschluss trainiert und evaluiert es das angegebene Modell fünfmal hintereinander, wobei jedes Mal ein anderer Fold zur Evaluierung genutzt wird, während auf den übrigen vier Folds trainiert wird (Géron 2020, 76–81). Das Ergebnis der Kreuzvalidierung ist ein Array mit fünf Scores aus denen der

Durchschnitt an die GridSearchCV weitergegeben wird. Das Resultat der GridSearchCV ist der durchschnittliche Score aus der Kreuzvalidierung für jede Kombination der vorher definierten Hyperparametern. Aus diesem Ergebnis sind nun die besten Hyperparameter mit dem erreichten Score abzulesen. Der „scoring“ Parameter der GridSearchCV wird auf „roc_auc“ gesetzt, wodurch die AUROC Evaluierungsmetrik genutzt wird. GridSearchCV gibt die Kombination an Hyperparametern aus, für die das ausgewählte Modell bei dieser Metrik den höchsten Score erreicht hat. Sollte die am besten performende Kombination an Hyperparametern am Rand des vorher definierten Bereichs liegen, könnte die Möglichkeit bestehen, dass Parameter außerhalb dieses Bereichs noch bessere Ergebnisse liefern. Aus diesem Grund wird in einem solchen Fall der definierte Bereich erweitert und die GridSearchCV erneut gestartet. Eine vollständige Liste der untersuchten Hyperparameter für die jeweiligen Modelle kann in Anhang 3 gefunden werden. Die Modelle werden mit den vordefinierten Hyperparametern in einer Pipeline aufgebaut. So könne die originalen- und die generierten Datensätze mit unterschiedlichen Werten für ε automatisiert auf mehrere Modelle trainiert und evaluiert werden. Dabei wird für jeden Datensatz ebenfalls automatisch Hyperparametertuning durchgeführt, um optimale Ergebnisse zu erzielen.

4.3.5 Evaluation

Nachdem der Datensatz vorbereitet, die Evaluationsmetrik festgelegt, die synthetischen Daten generiert und die Auswahl der Modelle getroffen ist, folgt hier die Evaluierung der Ergebnisse. Dafür wird in einem ersten Schritt das bestmögliche Modell auf den realen Daten gesucht. Hierbei wird eine Abwägung der Effektivität der realen Daten zu denen mit Standardskalierung gemacht. Im Anschluss wird die Effektivität synthetischer Daten ohne Differential Privacy im Gegensatz zu den realen Daten geprüft. Als letztes folgt ein Vergleich der Effektivität von synthetischen Datensätzen, die mit unterschiedlichem Differential Privacy Parameter ε erstellt werden.

4.3.5.1 Vergleich realer Datensatz und Datensatz mit Standardverteilung

Im folgenden Abschnitt wird untersucht, welches Ergebnis ein Modell auf den realen Daten maximal erreichen kann. Des Weiteren wird ein Überblick über den Einfluss der Standardskalierung auf die Effektivität der Daten dargestellt. Dafür wird die Pipeline zum einen auf den originalen Daten, zum anderen auf den transformierten Originaldaten gestartet. Wie in Kapitel 4.3.4 erläutert, wird das Modell auf die AUROC Metrik optimiert. Der F_1-Score wird zusätzlich bei den Ergebnissen genannt. Hierbei handelt sich es um den F_1-Score, der mit dem Modell optimiert auf den AUROC Score entsteht. Bei dieser Metrik könnten dementsprechend noch bessere Ergebnisse erzielt werden, wenn die Modelle auf diesen optimiert wären.

Modell	reales Datenset		standard-skaliertes Datenset	
	AUROC	F_1	AUROC	F_1
Logistic Regression	0,8412	0,3253	0,8426	0,3283
KNeighbors	0,6268	0,0010	0,8264	0,2257
Decision Tree	0,8288	0,2465	0,8288	0,2465
Random Forest	0,8520	0,2496	0,8518	0,2526
Gradient Boosting	0,8530	0,2824	0,8527	0,2843
Dummy	0,4985	0,0649	0,4991	0,0768
Durchschnitt	**0,8004**	**0,221**	**0,8405**	**0,2675**

Tabelle 1: Performance der Modelle auf den Originaldaten

Auf den realen Datensätzen erreichen die besten Classifier einen AUROC Wert von 0,85. Dabei hat sowohl auf den transformierten und auch auf den unskalierten Datensatz der GradientBoostingClassifier den höchsten Wert erzielt. Der Durchschnitt wird ohne den DummyClassifier berechnet, da diese Angabe nur zur Orientierung dient, welche Ergebnisse ein zufälliger Klassifikator aufweist. Die Skalierung auf eine Standardverteilung liefert im Durchschnitt bessere Ergebnisse als das reale Datenset, wobei der Durchschnitt auf das reale Datenset hauptsächlich aufgrund des schlecht performenden KNeighborsClassifier negativ beeinflusst wird. Die Ergebnisse der anderen Classifier sind beinahe identisch mit denen des skalierten Datensets.

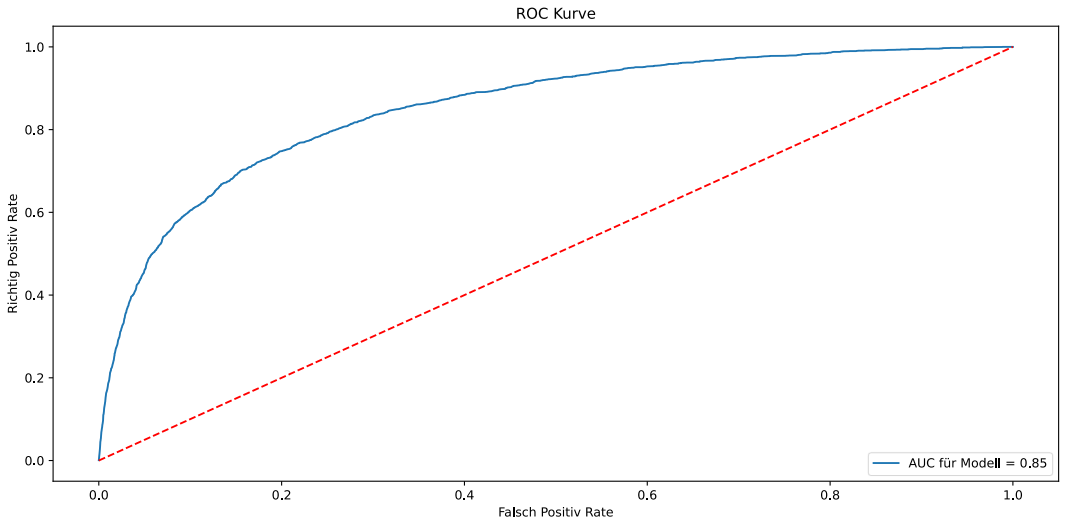

Abbildung 9: ROC Kurve des GradientBoostingClassifier auf dem skalierten Datenset

An der ROC Kurve des am besten performenden Modells auf den skalierten Daten (Abbildung 9) ist offensichtlich, dass das Modell fähig ist, eine höhere Richtig-positiv-Rate zu erreichen

als die Falsch-positiv-Rate. Die rot gestrichelte Linie symbolisiert einen zufälligen Klassifikator.

Bei der F_1 Metrik erzielt jeweils die logistische Regression die besten Ergebnisse mit einen Wert von 0,32. Dieser Wert ist ebenfalls deutlich besser, als wenn eine zufällige Klassifikation, mit Berücksichtigung der vorherrschenden Verteilung auf die Daten angewendet wird.

Das Ergebnis dieses Abschnitts ist, dass ein Modell auf den realen Daten nützliche Ergebnisse liefert. Es ist fähig, einen Unterschied zwischen den zwei Resultaten der Zielvariable mit einer AUROC Genauigkeit von 0,85 festzustellen. Ob der Datensatz auf die Standardverteilung skaliert ist oder nicht, hat bis auf den KNeighborsClassifier, keinen deutlichen Einfluss auf das Ergebnis der Modelle. Da das generative Modell bessere Ergebnisse auf den skalierten Datensatz gezeigt hat, wird in den folgenden Abschnitten nur noch anhand von Datensätzen in Standardverteilung gearbeitet.

4.3.5.2 Vergleich realer Datensatz und generierte Daten ohne Differential Privacy

Im folgenden Abschnitt wird eine Abwägung der Effektivität von realen Daten und generierten Daten des DP-WGAN ohne Privatsphäre-Einschränkungen gemacht. Diese Betrachtung ist wichtig, um zu verstehen, ob das generative Modell die Beziehungen zwischen den Merkmalen widerspiegeln kann. Dafür werden die deskriptiven Modelle auf generierten Daten des DP-WGAN trainiert und im Anschluss auf das reale Hold-out Set getestet. Für den Vergleich wird ebenfalls ein Modell auf den realen Daten trainiert, welches mit demselben Hold-out Set getestet wird.

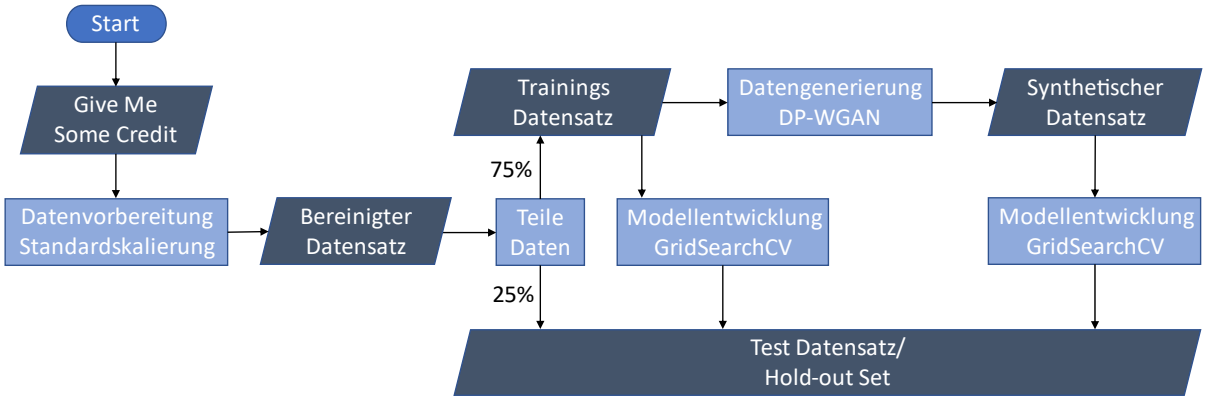

Abbildung 10: Flowchart des Versuchsaufbau

Die synthetischen Daten werden mittels DP-WGAN im „non-private“ Modus generiert. Bei der Datengenerierung ohne Privatsphäre-Einschränkung bietet sich der Vorteil, dass die Anzahl der zu trainierenden Epochen frei gewählt werden kann. Dies ist wie in Kapitel 4.3.3 erklärt nicht möglich, sobald Datensätze mit Differential Privacy erstellt werden. Es ist daher interessant, wie sich die Anzahl der trainierten Epochen auf die Qualität der Daten auswirkt. Dafür wird der

Schritt der Datengenerierung häufiger mit unterschiedlicher Anzahl an Epochen durchgeführt. Im Anschluss wird auf jeden der entstehenden Datensätze die Evaluierungspipeline gestartet. Die Ergebnisse der einzelnen Modelle auf die unterschiedlichen Datensätze werden in Tabelle 2 dargestellt. Hierbei wird der jeweils erreichte AUROC Score aufgetragen.

Modelle \ Epochen	50	100	300	500	800	Real
Logistic Regression	0,7298	0,7617	0,6145	0,6967	0,5533	0,8426
KNeighbors	0,6588	0,6947	0,4974	0,6534	0,5969	0,8264
Decision Tree	0,6328	0,6379	0,5660	0,5843	0,5254	0,8288
Random Forest	0,6868	0,6687	0,6501	0,6770	0,6491	0,8518
Gradient Boosting	0,6863	0,6584	0,6846	0,7227	0,6543	0,8527
Dummy	0,5020	0,5031	0,4968	0,5001	0,5022	0,4991
Durchschnitt	**0,6789**	**0,6843**	**0,6025**	**0,6668**	**0,5958**	**0,8405**

Tabelle 2: Performance der Modelle auf synthetischen Daten ohne Differential Privacy

Wie aus Tabelle 2 erkannt werden kann, ist die Effektivität der synthetischen Daten, mittels DP-WGAN ohne Differential Privacy erstellt, erkennbar schlechter als die des realen Datensatzes. Dieser Zusammenhang ist logisch, da ein Modell trainiert auf den originalen Daten, diese auch am besten unterscheiden kann. Es ist allerdings ebenfalls ersichtlich, dass die verwendeten Klassifizierungsalgorithmen fähig sind, bessere Ergebnisse zu erzielen als der zufällige Klassifikator. Dadurch wird deutlich, dass das generative Modell zumindest teilweise im Stande ist, die Verteilung der originalen Daten zu erkennen und sie in einem synthetischen Datenset wiederzugeben (vgl. Jordon et al. 2019, 8). Das beste Ergebnis, abgesehen von den Originaldaten, erreicht die logistische Regression auf dem Datensatz, der innerhalb von 100 Epochen mittels DP-WGAN erstellt wurde. Hier wird ein AUROC Score von 0,76 erreicht. Im Vergleich zu dem besten Ergebnis auf den realen Daten ist dieser Wert um 0,0910 schlechter.

Die durchschnittlichen Ergebnisse der Klassifizierungsmodelle auf den unterschiedlichen Datensätzen sind nicht linear. Das heißt, es kann nicht gesagt werden, dass mehr Epochen automatisch ein besseres Ergebnis liefern. Die Resultate unterscheiden sich in der Qualität der Daten. Da kein Zusammenhang zwischen trainierten Epochen und Datenqualität zu erkennen ist, wird davon ausgegangen, dass der DP-WGAN ungleichmäßig gute Ergebnisse liefert. Das Training eines Wasserstein GAN sollte stabiler funktionieren, als das Training eines originalen GANs (Arjovsky et al. 2017, 16). Diesem Ergebnis zufolge ist es dennoch ein Problem, dass Schwankungen zwischen den Ergebnissen bestehen.

4.3.5.3 Vergleich mit unterschiedlichen Privatsphäre-Einschränkungen

In diesem Abschnitt wird ein Vergleich der Effektivität der Datensätze bei unterschiedlichen Einstellungen für den Differential Privacy Parameter ε durchgeführt. Der Versuchsaufbau ist ähnlich zu dem in Abschnitt 4.3.5.2. Bei der Datengenerierung mittels DP-WGAN wird nun das Modell im Differential Privacy Modus gestartet. Somit entstehen differentiell private Datensets. Auf diesen Datensätzen werden ebenfalls die diskriminativen Modelle trainiert und auf den realen Daten evaluiert. Wenn Differential Privacy aktiviert ist, lässt sich die Anzahl der zu trainierenden Epochen nicht frei wählen. Da die Ergebnisse des DP-WGAN schwanken, wird für jeden der untersuchten ε Werte das generative Modell dreimal gestartet. Dabei wird Sigma unterschiedlich gewählt, so dass die Anzahl der trainierten Epochen zwischen [100, 700] liegt. Aus diesem Ansatz entstehen für jeden ε-Wert drei Datensätze, die in unterschiedlich vielen Epochen erstellt werden. Im Anschluss wird auf allen Datensätzen mittels der Klassifizierungsmodelle die Effektivität der Daten getestet. Die Evaluierungspipeline wird einmal mit allen Hyperparametern aus Anhang 3 gestartet. Um Overfitting beim Hyperparametertuning auf die synthetischen Daten entgegenzuwirken, werden die Modelle ebenfalls einmal nur mit den Hyperparametern der besten Modelle auf den realen Daten (Anhang 4) und einmal mit den Standardeinstellungen der Klassifikatoren getestet. Für jedes ε wird das Ergebnis verwendet, welches den höchsten durchschnittlichen AUROC Wert erreicht hat. Somit wird nur der am besten performende Datensatz für die Ergebnisse des folgenden Kapitels genutzt. Durch die mehrfache Generierung von Datensätzen wird das Risiko vermindert, dass die Resultate durch einen schlechten Durchlauf des DP-WGAN verfälscht werden. Es hat sich nämlich auch hier gezeigt, dass die vom generativen Modell erstellen Datensätze ebenso für selbe ε Werte mit identischen Einstellungen unterschiedlich gute Ergebnisse liefern.

Epsilon ε / Modelle	0,02	0,05	0,1	1	3	5	8	12	16
Logistic Regression	0,5857	0,7044	0,4920	0,6696	0,5209	0,7060	0,6261	0,6950	0,7263
KNeighbors	0,5383	0,5207	0,5642	0,4863	0,5844	0,7018	0,6912	0,6485	0,6314
Decision Tree	0,6952	0,5003	0,5398	0,6111	0,5259	0,5977	0,6884	0,5722	0,6900
Random Forest	0,6473	0,5068	0,4693	0,6316	0,4794	0,7177	0,7429	0,6240	0,7370
Gradient Boosting	0,6255	0,5881	0,4255	0,6763	0,5466	0,6655	0,6802	0,7147	0,6995
Dummy	0,5032	0,5016	0,4979	0,5006	0,5017	0,5007	0,4971	0,4948	0,5021
Durchschnitt	**0,6184**	**0,5641**	**0,4982**	**0,6150**	**0,5314**	**0,6777**	**0,6858**	**0,6509**	**0,6968**

Tabelle 3: Performance der Modelle bei unterschiedlichen ε-Werten

Den höchsten Score auf einen Datensatz mit DP erreicht der RandomForestClassifier auf dem Datenset, welches mit ε=8 trainiert wurde (Tabelle 3). Mit einem AUROC Score von 0,7429 wird deutlich, dass das generative Modell fähig ist, die Verteilung der originalen Daten zumindest teilweise zu reproduzieren. Den höchsten Durchschnittswert erzielen die deskriptiven Modelle auf dem Datensatz, der mit ε=16 erstellt wurde. Dieser ist mit 0,6968 sogar höher als der höchste Durchschnittswert auf den synthetischen Daten ohne DP (Tabelle 2). Aus diesem Grund ist zu vermuten, dass die Auswirkung von ε=16 auf einen Datensatz dieser Größe, die Effektivität nicht einschränken. Ebenfalls haben ε=5 und ε=8 gute Durchschnittswerte erreicht, die mit dem Ergebnis im „non-private" Modus zu vergleichen sind. Bei den dreimal gestarteten DP-WGAN mit ε=3 funktionierte es bei keinem der erstellten Datensätze die Originaldaten bestmöglich aufzufassen.

Ein ebenfalls sehr interessantes Ergebnis ist die logistische Regression trainiert auf dem Datensatz mit ε=0,05. Diese erreichte einen Score von 0,7044 und ist damit ähnlich effektiv wie das Modell auf Datensätzen mit höheren ε-Werten. Des Weiteren performen die anderen Klassifizierungsalgorithmen auf diesem Datensatz deutlich schlechter. So weisen die übrigen Modelle einen durchschnittlichen AUROC Score von 0,5290 auf, was nur etwas besser als ein DummyClassifier ist.

Abgesehen von diesen Ausreißern lässt sich eine eindeutige Tendenz erkennen. Im Durchschnitt performen Modelle besser, wenn sie auf einem Datensatz mit höherem ε trainiert sind.

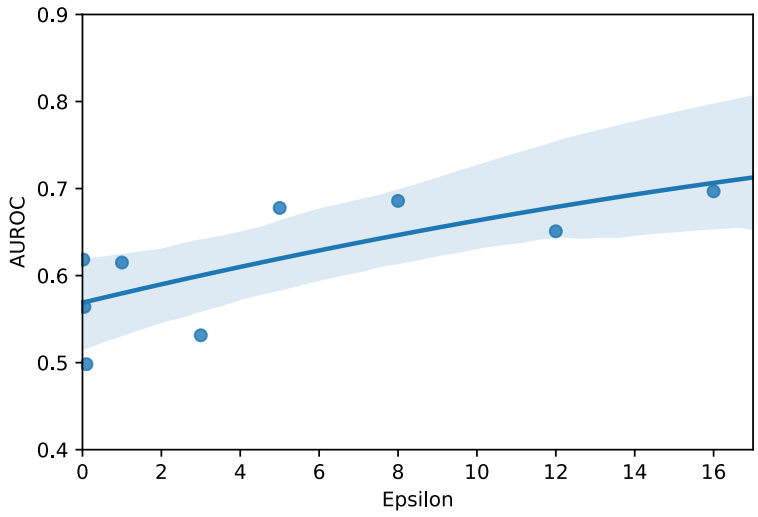

Abbildung 11: Ergebnisse der Modelle bei unterschiedlichen ε

Obwohl der Datensatz mit ε=0,02 Ergebnisse aufweist, die deutlich besser als ein komplett zufällig erstellter Datensatz sind, wird deutlich, dass im Durchschnitt die Datensätze mit niedrigem ε schlechter performen als die mit höheren ε (Abbildung 11). Somit wird hier der Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen offensichtlich.

Bei höheren Werten für ε nimmt die Effektivität und damit auch die Nutzbarkeit der Daten zu. Dies ist zu Vergleichen mit den Ergebnissen von anderen Studien (vgl. Jordon et al. 2019, 9f.; Torfi et al. 2020, 9–11). Im Vergleich zu Modellen, die auf realen Daten trainiert sind, ist die Qualität merkbar schlechter, trotzdem können die Ergebnisse von Datensätzen mit DP für nützliche Erkenntnisse verwendet werden.

4.4 Interpretation der Ergebnisse

Es stellt sich nun die Frage, wie dieser Trade-off zu bewerten ist. Dafür wird als erstes geklärt, wie nützlich die Ergebnisse auf den Originaldaten sind. In der Finanzbranche verlangen beispielsweise regulatorische Anforderungen, dass alle automatisierten Entscheidungen, die von Kreditscoring-Algorithmen getroffen werden, transparent und begründbar sind, um zum Beispiel die Diskriminierung von Personen zu vermeiden (Müller et al. 2016, 294). Aus diesem Grund wird für folgendes Beispiel die logistische Regression verwendet, welche verständliche und reproduzierbare Ergebnisse liefert.

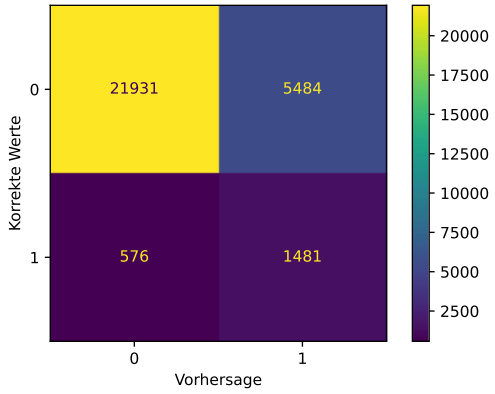

Abbildung 12: Konfusionsmatrix der Logistischen Regression auf den realen Daten

Die Konfusionsmatrix aus Abbildung 12 zeigt, dass das Ergebnis zwar eine große Anzahl an falsch Positiv Einträgen enthält, die Relevanz (Precision) liegt bei 0,21. Es zeigt jedoch ebenfalls, dass ein großer Anteil der Personen, die in eine finanzielle Delinquenz geraten, erfolgreich erkannt werden, so liegt die Sensitivität (Recall) bei 0,72. Diese Erkenntnis ist für Kreditgebende sicherlich eine nützliche Information.

Der Zugriff auf die realen Daten ist jedoch den meisten Forschenden nicht gestattet, da diese aus Datenschutzgründen nicht geteilt werden können (Jordon et al. 2019, 1). Die in dieser Arbeit erstellten synthetischen Datensätze schützen mit der Garantie von Differential Privacy die Privatsphäre der einzelnen Teilnehmenden. Die erstellten Datensätze können genutzt werden, um darauf verschiedene Algorithmen zu trainieren oder sie an externe Forscher-/innen

weiterzugeben, ohne die Privatsphäre des ursprünglichen Datensatzes zu gefährden (Jordon et al. 2019, 1).

Es hat sich gezeigt, dass die Ergebnisse auf den synthetischen Daten zwar schlechter als auf den realen Daten sind, dennoch können sie für Personen oder Organisationen, die sonst keinen Zugriff auf solche Daten hätten, nützliche Erkenntnisse erbringen. Inwieweit die synthetischen Daten genutzt werden können, um darauf basierend tatsächliche Entscheidungen über die Kreditwürdigkeit einer Person zu treffen, ist eine juristische Frage, auf die nicht weiter eingegangen wird. Gewiss können die Daten jedoch verwendet werden, um bestimmte Korrelationen aufzudecken, z. B. das Personen, die an oder auch über die Grenze ihres Kreditlimits gehen, eher dazu neigen, in den nächsten zwei Jahren in Verzug zu geraten (Abbildung 6). Je niedriger ε gewählt ist, desto mehr Rauschen wird dem Datensatz zugeführt. Die Effektivität und damit die Nützlichkeit der Daten nimmt folglich ab. Es ist wichtig, dass die Privatsphäre der Teilnehmenden geschützt wird, jedoch müssen die Daten noch genug Informationen enthalten, dass die darauf erzielten Ergebnisse nützliche Erkenntnisse erbringen, denn sonst wäre der Prozess der Datenerhebung und Auswertung nicht sinnvoll. Die Höhe eines angemessenen ε und damit eines angemessenen Trade-off zwischen Effektivität und Differential Privacy bleibt weiterhin eine soziale Frage (Dwork 2008, 3).

5 Diskussion der Ergebnisse

Aufgrund der unausgewogenen Testdaten (die Zielvariable hat einen Durchschnitt von 0,067) wird hier die AUROC Metrik als Maß der Evaluation gewählt. Die besten Testergebnisse auf den realen Daten erreichen dabei einen Wert von 0,85 (Tabelle 1). Bei Klassifikatoren, die auf generierten Daten trainiert werden, sinkt die Effektivität auf das bestmögliche Ergebnis von 0,76 (Tabelle 2). Wird bei der Datengenerierung Differential Privacy miteinbezogen, erzielt das beste Modell einen Score von 0,74 (Tabelle 3).

Trotz der Tatsache, dass in den meisten Fällen Klassifikatoren, die auf realen Daten trainiert werden besser abschneiden, als Klassifikatoren, die auf generierten Daten trainiert werden, können die erstellten Daten nützliche Erkenntnisse erbringen. Diese Datensätze können mit der Garantie von DP die Privatsphäre der Teilnehmer-/innen schützen und könnten somit an Dritte weitergegeben werden. Die Daten könnten so von verschiedenen Forscher-/innen verwendet werden, um weitere Forschungen damit durchzuführen (vgl. Jordon et al. 2019, 1).

Es wird deutlich, dass die AUROC Werte der Modelle auf den generierten Daten mit dem Abnehmen von ε (mehr Rauschen hinzugefügt) schlechtere Ergebnisse erzielen. Das liegt daran, dass beim Generieren von Daten das Training des Diskriminators gestört wird und damit der Generator indirekt beeinflusst wird, was zu einer Abweichung der Ausgabeverteilung von den realen Daten und einer schlechteren Testleistung der deskriptiven Modelle führt (Xie et al. 2018, 8f.). Es lässt sich sagen, dass höhere Privatsphäre-Einschränkungen durch Differential Privacy bei einem generativen Modell dazu führen, dass die interdimensionalen Beziehungen der Daten weniger gut erfasst werden.

Die Qualität der vom DP-WGAN erstellen Datensets schwankt deutlich, selbst bei gleichen Einstellungen der Hyperparameter. So erreichten die Klassifizierungsmodelle häufig unterschiedlich gute Ergebnisse auf zwei Datensets, obwohl diese mit identischen Einstellungen erstellt wurden. Die Dateninhaberin bzw. der Dateninhaber sollten somit bevor, sie synthetische Daten mit Differential Privacy Privatsphäre-Einschränkungen veröffentlicht, die Qualität dieser Daten kontrollieren.

Die Frage der richtigen Wahl des Parameters ε von Differential Privacy bleibt weiterhin offen. Offensichtlich sind kleinere Werte für ε im Allgemeinen besser, da sie einen höheren Schutz der Privatsphäre des Einzelnen garantieren. Es ist dennoch schwer zu bestimmen, welche Werte klein genug sind, denn wie in dieser Arbeit gezeigt, sinkt damit auch die Effektivität der Daten. Des Weiteren ist ε kein absolutes, sondern eher ein relatives Maß, für den Datenschutz. Selbst bei gleichem Wert von ε sind die von der differentiellen Privatsphäre erzwungenen Datenschutzgarantien je nach Anwendungsfall unterschiedlich (Lee und Clifton 2011, 325).

6 Schlussfolgerung

Die Schlussfolgerung beinhaltet eine Zusammenfassung der in dieser Arbeit verwendeten Vorgehensweise und die erreichten Ergebnisse. Zudem wird auf Limitationen dieser Arbeit eingegangen und ein Ausblick auf mögliche weiterführende Forschung dargestellt.

6.1 Zusammenfassung

Das Ziel dieser Arbeit ist es, einen Überblick über den Trade-off zwischen Effektivität und Differential Privacy bei Vorhersagemodellen mittels maschineller Lernverfahren dazustellen. Dies wird anhand von Daten zur Kreditwürdigkeitsprüfung aus dem Finanzwesen untersucht. Zu Beginn werden in den theoretischen Grundlagen die Konzepte um Differential Privacy und Deep Generative Models erklärt. Differential Privacy ist im Wesentlichen eine Garantie der Datensammlerin bzw. des Datensammlers gegenüber dem Individuum, dass sich die Folgen einer Datenbank, d. h. die Ergebnisse eines Algorithmus, der die Datenbank nutzt, nicht wesentlich ändern, ob es die Nutzung der Daten zulässt oder nicht (Dwork und Roth 2014, 17). Mit Differential Privacy kann somit anhand einer soliden mathematischen Definition eine Garantie für die Privatsphäre von Teilnehmenden einer Datenbank gegeben werden (Torfi et al. 2020, 1). Deep Generative Models sind Methoden, die das Erlernen einer gemeinsamen Verteilung über alle Variablen, versuchen zu modellieren. Ein generatives Modell simuliert, wie die Daten in der realen Welt erzeugt werden (Kingma und Welling 2019, 308). Mit einem gut erstellten Modell können im Anschluss synthetische Daten generiert werden, die im besten Fall die Verteilung der Originalen widerspiegeln. Diese zwei Konzepte werden in dem DP-WGAN vereint. Die entstehenden Ergebnisse sind differentiell private, synthetische Datensets.

Die Datenanalyse in dieser Arbeit wird nach dem wissenschaftlichen Framework für Big Data Analysen von Müller et al. (2016) durchgeführt. Der Prozess der Forschung wird hierfür in die Phasen: Forschungsfrage, Datenbeschaffung, Datenanalyse und Ergebnisinterpretation aufgeteilt. Für die Ergebnisse dieser Arbeit werden synthetische Datensets mittels DP-WGAN erstellt. Dabei wird die Höhe des Differential Privacy Parameters ε unterschieden. Auf den realen und auf den vom DP-WGAN erstellten Datensätzen werden diskriminative Modelle trainiert und im Anschluss die Effektivität der Modelle auf dem realen Testdatensatz evaluiert. Die Ergebnisse zeigen, dass auf den realen Daten ein AUROC Wert von bis zu 0,85 erreicht werden kann. Wird ein Modell auf synthetischen Daten ohne DP trainiert, erreicht das beste Modell einen Score von 0,76. Bei der Betrachtung von Datensätzen mit unterschiedlichen Werten für ε wird deutlich, dass hier ein höherer Wert im Durchschnitt bessere Ergebnisse liefert. Dies ist der Fall, da für das Erreichen eines niedrigeren ε dem Algorithmus mehr zufälliges Rauschen hinzugefügt wird. Das Rauschen stört die Ergebnisse des Diskriminators und beeinflusst damit indirekt die erstellten Daten des Generators, was zu einer Abweichung der Ausgabeverteilung

von den realen Daten führt (Xie et al. 2018, 8f.). Das beste Modell auf synthetischen Daten mit Differential Privacy erreicht einen AUROC Score von 0,74 welcher beweist, dass die Ergebnisse der Klassifizierungsalgorithmen weiterhin nützliche Informationen beinhalten.

6.2 Limitationen

Es war nicht möglich mit dem verwendeten Package und Datensatz synthetische Daten mit einem kleineren Wert für ε als 0,02 zu erstellen. Hätte dies gewährleistet werden können, so hätte der Trade-off zwischen Effektivität und Differential Privacy noch besser dargestellt werden können. Da die entstehenden Daten mit ε=0,02 aufgrund des großen Datensatzes immer noch bessere Ergebnisse erzielen als es ein komplett zufällig erstellter Datensatz.

Das Erstellen synthetischer Daten mittels DP-WGAN benötigt einiges an Rechenleistung. So hat das Training eines einzigen Datensatzes häufig mehrere Stunden an Rechenzeit in Anspruch genommen. Somit konnte bei der Generierung der Daten kein Hyperparametertuning durchgeführt werden. Es wurden dieselben Hyperparameter verwendet, die ebenfalls von den Erstellern/-innen der „Private Data Generation Toolbox“ für dasselbe Datenset verwendet werden.

Da die Qualität der erstellten Daten schwankte, ist es bei einer erneuten Durchführung des Forschungsprozess möglich, dass andere Ergebnisse als die hier dargestellten entstehen. Es sollte jedoch immer eine Tendenz zu erkennen sein, dass die Qualität der Daten bei einem Anstieg von ε ebenfalls steigt.

6.3 Weiterführende Forschung

Interessant in Bezug auf mögliche weiterführende Untersuchungen ist eine Abwägung vergleichbarer generativer Modelle, die ebenfalls Differential Privacy verwenden. Es könnte hierbei zwischen unterschiedlichen Arten von generativen Modellen wie Variational Autoencoder und Generative Adversarial Networks unterschieden werden. Des Weiteren könnten auch verschiedene GAN Modelle miteinander verglichen werden. Ein weiterer Faktor, der untersucht werden könnte, ist der Einfluss des Differential Privacy Parameters δ, der die Fehlertoleranz des Algorithmus beschreibt, auf die Qualität der Daten.

Weitere interessante Fragestellungen im Bezug zu Differential Privacy entstehen aus den Auswirkungen der Volkszählung, dem „Census“ 2020 in Amerika. Hier hat DP erstmals große Aufmerksamkeit in der Öffentlichkeit erhalten. Es muss analysiert werden, wie erfolgreich das Anwenden der Methode bei einem solch großen, staatlichen Vorhaben ist. Welche Probleme traten auf und wie wurde die Anwendung von Experten/-innen und der Öffentlichkeit aufgenommen? Sollte Differential Privacy bei der nächsten Volkszählung im Jahr 2030 weiterverwendet werden und wenn ja, wie sollte die Anwendung angepasst werden?

Literaturverzeichnis

Arjovsky, M. et al. (2017): *Wasserstein GAN*. Verfügbar unter: https://arxiv.org/pdf/1701.07875.

BorealisAI (2019): *A toolbox for differentially private data generation*. In: https://github.com/BorealisAI/private-data-generation, zugegriffen am 23.07.2021.

Dastile, X. et al. (2020): *Statistical and machine learning models in credit scoring: A systematic literature survey*. In: Applied Soft Computing, 91, 106263.

Durall, R. et al. (2020): *Combating Mode Collapse in GAN training: An Empirical Analysis using Hessian Eigenvalues*. Verfügbar unter: https://arxiv.org/pdf/2012.09673.

Dwork, C. et al. (2006): *Calibrating Noise to Sensitivity in Private Data Analysis*. In: Theory of Cryptography, 265–284.

Dwork, C. (2008): *Differential Privacy: A Survey of Results*. In: Theory and Applications of Models of Computation (4978), 1–19.

Dwork, C. (2011): *Differential Privacy*. In: van Tilborg, H.; Jajodia, S. (Hg.): Encyclopedia of cryptography and security, 338–340.

Dwork, C.; Roth, A. (2014): *The Algorithmic Foundations of Differential Privacy*. In: Foundations and Trends® in Theoretical Computer Science, 9 (3-4), 211–407.

Géron, A. (2020): *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow. Konzepte, Tools und Techniken für intelligente Systeme*. Unter Mitarbeit von Kristian Rother und Thomas Demmig. 2. Auflage, O'Reilly (Animals), Heidelberg.

Goodfellow, I. J. et al. (2014): *Generative Adversarial Networks*. Verfügbar unter: https://arxiv.org/pdf/1406.2661.

Hitaj, B. et al. (2017): *Deep Models Under the GAN: Information Leakage from Collaborative Deep Learning*. Verfügbar unter: http://arxiv.org/pdf/1702.07464v3.

Hofmann, H. (1994): *UCI Machine Learning Repository: Statlog (German Credit Data) Data Set*. In: https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data), zugegriffen am 22.07.2021.

Hsu, J. et al. (2014): *Differential Privacy: An Economic Method for Choosing Epsilon*. Verfügbar unter: https://arxiv.org/pdf/1402.3329.

Jordon, J. et al. (2019): *PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees*. In: 7th International Conference on Learning Representations. Verfügbar unter: https://openreview.net/forum?id=S1zk9iRqF7.

Kaggle.com (2011): *Give Me Some Credit*. In: https://www.kaggle.com/c/GiveMeSomeCredit, zugegriffen am 23.07.2021.

Kairouz, P. et al. (2013): *The Composition Theorem for Differential Privacy*. Verfügbar unter: https://arxiv.org/pdf/1311.0776.

Kalin, J. (2018): *Generative Adversarial Networks Cookbook. Over 100 Recipes to Build Generative Models Using Python, TensorFlow, and Keras*, Packt Publishing Ltd, Birmingham.

Kang, Y. et al. (2020): *Input Perturbation: A New Paradigm between Central and Local Differential Privacy*. Verfügbar unter: https://arxiv.org/pdf/2002.08570.

Karras, T. et al. (2018): *A Style-Based Generator Architecture for Generative Adversarial Networks*. Verfügbar unter: https://arxiv.org/pdf/1812.04948.

Kim, T. et al. (2017): *Learning to Discover Cross-Domain Relations with Generative Adversarial Networks*. Verfügbar unter: https://arxiv.org/pdf/1703.05192.

Kingma, D. P.; Welling, M. (2013): *Auto-Encoding Variational Bayes*. Verfügbar unter: https://arxiv.org/pdf/1312.6114.

Kingma, D. P.; Welling, M. (2019): *An Introduction to Variational Autoencoders*. In: Foundations and Trends® in Machine Learning, 12 (4), 307–392.

Langr, J.; Bok, V. (2019): *GANs in Action: Deep learning with Generative Adversarial Networks*. 1. Aufl., Manning Publications, Shelter Island.

Lee, J.; Clifton, C. (2011): *How Much Is Enough? Choosing ε for Differential Privacy*. In: Xuejia Lai, Jianying Zhou und Hui Li (Hg.): Information Security, Bd. 7001. Berlin, Heidelberg: Springer Berlin Heidelberg (Lecture Notes in Computer Science), 325–340.

Ma, C. et al. (2020): *RDP-GAN: A Rényi-Differential Privacy based Generative Adversarial Network*. Verfügbar unter: https://arxiv.org/pdf/2007.02056.

Müller, O. et al. (2016): *Utilizing big data analytics for information systems research: challenges, promises and guidelines*. In: European Journal of Information Systems, 25 (4), 289–302.

Narayanan, A.; Shmatikov, V. (2008): *Robust De-anonymization of Large Sparse Datasets*. In: IEEE Symposium on Security and Privacy, 111–125.

Nguyen, H. H. et al. (2016): *Detecting Communities under Differential Privacy*. Verfügbar unter: https://arxiv.org/pdf/1607.02060.pdf.

Pedregosa, F. et al. (2011): *Scikit-learn: Machine learning in Python*. In: Journal of Machine Learning Research (12), 2825–2830.

Sarwate, A. D.; Chaudhuri, K. (2013): *Signal Processing and Machine Learning with Differential Privacy: Algorithms and challenges for continuous data*. In: IEEE signal processing magazine, 30 (5), 86–94.

Saxena, D.; Cao, J. (2021): *Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions*. In: ACM Computing Surveys, 54 (3), 1–42.

scikit-learn (2020a): *6.3. Preprocessing data*. In: https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-scaler, zugegriffen am 01.08.2021.

scikit-learn (2020b): *Supervised learning*. In: https://scikit-learn.org/stable/supervised_learning.html, zugegriffen am 28.07.2021.

Torfi, A. et al. (2020): *Differentially Private Synthetic Medical Data Generation using Convolutional GANs*. Verfügbar unter: http://arxiv.org/abs/2012.11774.

UCI (1987): *UCI Machine Learning Repository: Statlog (Australian Credit Approval) Data Set*. In: http://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval), zugegriffen am 22.07.2021.

United States Census Bureau (2020): *Statistical Safeguards*. In: https://www.census.gov/about/policies/privacy/statistical_safeguards.html, zugegriffen am 16.07.2021.

Wang, T. et al. (2020): *A Comprehensive Survey on Local Differential Privacy toward Data Statistics and Analysis*. In: Sensors (Basel, Switzerland), 20 (24), 7030.

Warner, S. L. (1965): *Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias*. In: Journal of the American Statistical Association, 60 (309), 63–69.

Xie, L. et al. (2018): *Differentially Private Generative Adversarial Network*. Verfügbar unter: https://arxiv.org/pdf/1802.06739.

Xu, L. et al. (2019): *Modeling Tabular data using Conditional GAN*. Verfügbar unter: https://arxiv.org/pdf/1907.00503.

Zhang, H. et al. (2019): *StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks*. In: IEEE transactions on pattern analysis and machine intelligence, 41 (8), 1947–1962.

Anhang

Anhang 1: Kurzbeschreibung der Features von Give Me Some Credit

age	Alter des Kreditnehmers in Jahren
DebtRatio	Monatliche Schuldenzahlungen, Alimente, Lebenshaltungskosten geteilt durch das monatliche Bruttoeinkommen
MonthlyIncome	Das monatliche Einkommen
NumberOfDependents	Anzahl der unterhaltsberechtigten Personen in der Familie ohne sich selbst
NumberOfOpenCredit-LinesAndLoans	Anzahl der offenen Kredite (Ratenkredit wie Autokredit oder Hypothek) und Kreditlinien (z. B. Kreditkarten)
NumberOfTime30-59DaysPastDueNotWorse	Anzahl der Fälle, in denen der Kreditnehmer in den letzten 2 Jahren 30-59 Tage überfällig, aber nicht schlechter war.
NumberOfTime60-89DaysPastDueNotWorse	Anzahl der Fälle, in denen der Kreditnehmer in den letzten 2 Jahren 60-89 Tage überfällig, aber nicht schlechter war.
NumberOfTimes90Days-Late	Anzahl der Male, in denen der Kreditnehmer 90 Tage oder mehr im Verzug war.
NumberRealEstateLo-ansOrLines	Anzahl der Hypotheken- und Immobilienkredite einschließlich Home-Equity-Kreditlinien
RevolvingUtilizationO-fUnsecuredLines	Gesamtsaldo von Kreditkarten und persönlichen Kreditlinien außer Immobilien und keine Ratenschulden wie Autokredite, geteilt durch die Summe der Kreditlimits.
SeriousDlqin2yrs	Person wird in den nächsten zwei Jahren in Verzug von 90 Tage oder länger geraten

Anhang 2: Test des generativen Modells mit unterschiedlichen Skalierungen

	Ohne Skalierung	MinMax-Scaler	MaxAbs-Scaler	Robust-Scaler	Standard-Scaler
Logistic Regression	0,457	0,8	0,754	0,642	0,761
Random Forest	0,284	0,686	0,786	0,722	0,769
Neural Network	0,435	0,640	0,708	0,688	0,787
Gaussian NB	0,469	0,680	0,776	0,586	0,753
Gradient Boosting	0,291	0,738	0,757	0,783	0,790
Gesamt	**0,387**	**0,709**	**0,756**	**0,684**	**0,772**

Anhang 3: Alle verwendeten Hyperparameter für das Hyperparametertuning

LogisticRegression	
max_iter	50, 80, 100, 150, 200, 300
C	0.1, 0.5, 0.8, 1, 3, 5, 10, 15
class_weight	balanced, None
KNeighborsClassifier	
n_neighbors	2, 3, 5, 10, 15, 20, 30, 40, 55, 60, 70
weights	uniform, distance
DecisionTreeClassifier	
max_depth	2, 5, 10, 20, 30, None
min_samples_split	2, 5, 10
criterion	gini, entropy
RandomForestClassifier	
n_estimators	50, 100, 200, 300
max_depth	5, 10, 20, None
GradientBoostingClassifier	
n_estimators	100, 200, 300, 400
learning_rate	0.05, 0.07, 0.1
DummyClassifier	
strategy	stratified

Anhang 4: Hyperparameter der besten Modelle auf realen, skalierten Daten

Modell	**AUROC**	**Hyperparameter**
LogisticRegression	0.8426	C: 1, class_weight: balanced, max_iter: 200
KNeighborsClassifier	0.8264	n_neighbors: 55, weights: uniform
DecisionTreeClassifier	0.8288	criterion: entropy, max_depth: 5
RandomForestClassifier	0.8518	max_depth: 10, n_estimators: 200
GradientBoostingClassifier	0.8529	learning_rate: 0.05, n_estimators: 400
DummyClassifier	0.4991	strategy: stratified

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Würzburg, den 6. August 2021 _____

Unterschrift