**Einleitung**

Derzeit stellen unausgeglichene Datensätze eine erhebliche Herausforderung im Bereich des maschinellen Lernens und des Data Minings dar. Denn die herkömmlichen Klassifizierungsverfahren neigen in der Regel dazu, die Mehrheitsklasse zu bevorzugen, welche bei einem binären Klassifizierungsproblem oft ein Vielfaches an Instanzen gegenüber der Minderheitsklasse besitzt. Hierdurch wird der Klassifikator in seiner Vorhersagefähigkeit, die Minderheitsinstanzen korrekt zu erkennen, stark beeinträchtigt, kann jedoch trotzdem eine hohe Accuracy aufweisen. Die Minderheitsklasse kann dabei auch völlig unerkannt bleiben. Diese Thematik ist insbesondere bei der Erkennung betrügerischer Transaktionen in Banken, Kreditrisikobewertung oder Erkennung von Firewall-Eingriffen von hoher Relevanz. Aufgrund dieser universellen Existenz von unausgewogenen Datensätzen wurden bereits viele Vorverarbeitungsmethoden vorgeschlagen, um die Imbalance zu bewältigen. Oversampling ist eine vielversprechende Technik für unausgewogene Datensätze, die neue Minderheiteninstanzen erzeugt, um den Datensatz auszugleichen. Unter den Oversampling-Methoden ist die Synthetic Minority-Oversampling-Technique (SMOTE) eine der bekanntesten Methoden, die künstliche Minoritätsinstanzen durch lineare Interpolation erzeugt. Eine jüngst veröffentlichte Adaption dessen liefert im Paper „ASN-SMOTE: a synthetic minority oversampling method with adaptive qualified synthesizer selection“ von Yi Xinkai et al. (2022) vielversprechende Ergebnisse, weshalb dieser Algorithmus im Folgenden auf dem ‚Creditcard‘ Datensatz angewendet und seine Performance mithilfe zweier geeigneter Classifier mit dem originalen SMOTE Algorithmus verglichen wird. Dafür wird nach der Data Exploration im 1. Kapitel die Funktionsweise des ASN-SMOTE sowie die Unterschiede zum klassischen SMOTE in Kapitel 2 erläutert. Anschließend werden die verwendeten Classifier und das Performancemaß in Kapitel 3 sowie die gewählte Cross-Validation im 4. Kapitel präsentiert. Darauf folgt in Kapitel 5 und 6 eine Methode des Undersamplings und die Durchführung der Hyperparameteroptimierung. Abschließend werden die Ergebnisse des ASN-SMOTE in Kapitel 6 analysiert, bevor ein Ausblick mit weiteren Ideen und Forschungsfragen folgt. Der Code des entsprechenden Abschnitts ist zur Nachvollziehbarkeit in der Programmiersprache R an jedes Kapitel angehängt.

1. Data Exploration und SMOTE Algorithmus

Der zu behandelnde Datensatz „Creditdata“ beinhaltet europäische Kreditkartentransaktionen von September 2013 im CSV-Format und besitzt pro Transaktion 31 Features/Spalten, wobei die letzte binäre Spalte die Target darstellt. Diese definiert, ob ein Sample Fraud (1) ist oder kein Fraud (0). Das Ziel hierbei ist es ein Machine Learning Algorithmus zu trainieren, welcher mit Kreditkartenbetrug erkennen soll, um die Kunden vor unautorisierten Belastungen zu schützen. Die Daten sind ausschließlich numerisch und abgesehen von „Time“ und „Amount“ aus Datenschutzgründen mit PCA transformiert worden. Zu Beginn werden die notwendigen Packages geladen sowie die Working Directory in den Ordner gesetzt, worin sich der „Creditdata“ Datensatz befindet und mit read.csv eingelesen. Anschließend wird eine NA-Analyse durchgeführt und einige Statistiken der einzelnen Spalten berechnet. Die Imbalance des Datensatzes wird deutlich, wenn wir die Anzahl der Minority Samples mit den 284.315 der Majority vergleichen. Es befinden sich lediglich 492 Fraud Samples in Creditdata, was einem Prozentsatz von 0,17% entspricht. Dies deutet auf eine extreme Imbalance zwischen Majority und Minority hin und macht eine Preprocessingmethode umso bedeutender. Aufgrund der hohen Imbalance wird ein 90 zu 10 Split der Daten in Trainings- und Testdatensatz durchgeführt, worauf eine Trennung der Features von der Target Variable folgt. Da das Feature „Time“ als nicht aussagekräftig genug erscheint, wird dieses aus dem Datensatz entfernt und ebenfalls ein Scaling der Spalte „Amount“ vorgenommen, da diese nicht mit PCA transformiert wurde. Dies geschieht in Trainings- und Testdatensatz separat, um den Mittelwert und die Standardabweichung des Testdatensatzes nicht mit in den Trainingsdatensatz aufzunehmen, was andernfalls zu Overfitting führen würde. Beim Korrelationsplot der Features mit der Target Variable fallen einige positive als auch negative Korrelation beim Feature „Amount“ beispielsweise mit „V2“ und „V7“ auf. Die Target Value weist lediglich mit den Features „V12“, „V15“ und „V17“ einige niedrige negative Korrelation auf. Im Anschluss an die Data Exploration wird der in R bereits implementierte SMOTE Algorithmus aus dem Package smotefamily auf den Trainingsdatensatz angewendet, um eine Balance im Datensatz herzustellen. Nach der Generierung von xxx synthetischen Trainingsdaten beträgt der Anteil der Minority Samples im Train Set 49,97%.

**ASN-SMOTE**

Allerdings können unsachgemäß erzeugte Minderheiteninstanzen, d. h. Rauschinstanzen, das Lernen des Klassifikators stören und Klassifikators beeinträchtigen und sich negativ auf ihn auswirken. Aus diesem Grund schlagen wir in diesem Papier einen einfachen und effektiven Oversampling-Ansatz vor, der als ASN-SMOTE, das auf der k-nearest neighbors und der synthetischen Minderheiten-Oversampling-Technologie (SMOTE) basiert.

Um die Wirksamkeit zu belegen, wurde ASN-SMOTE auf drei verschiedene Klassifikatoren angewandt und umfassende Experimente wurden mit 24 unausgewogenen Benchmark-Datensätzen durchgeführt. ASN-SMOTE wird auch ausführlich mit neun namhaften Oversampling-Algorithmen verglichen. Die Ergebnisse zeigen, dass ASN-SMOTE in den meisten Datensätzen die besten Ergebnisse erzielt. Die ASN-SMOTE-Implementierung ist verfügbar unter: https://www.github.com/yixinkai123/ASN-SMOTE/.

Allerdings, SMOTE und einige andere bestehende Methoden konzentrieren sich jedoch hauptsächlich auf das Ungleichgewicht zwischen den Klassen, lösen aber nicht effektiv das Problem des Ungleichgewichts innerhalb der Klasse und kleiner Disjunktionen [3,13,30]. Dies kann zu unsachgemäß synthetisierten Minoritätsinstanzen führen, wie z. B. Rauschen und überlappende Instanzen, die zu einer Leistungsverschlechterung statt zu statt zu verbessern. Obwohl es einige Methoden gibt, die Minoritätsinstanzen um die Entscheidungsgrenze Instanzen um die Entscheidungsgrenze herum verwenden und die die oben genannten Probleme weitgehend lösen können, doch wird dadurch die Entscheidungsgrenze noch undeutlicher und unschärfer, und kann das Problem verursachen, dass die neu synthetisierten Beispiele in der Region der Mehrheitsklasse angezeigt werden. Dies verstärkt das Rauschen und verschlechtert die Leistung der Klassifikatoren [56]. Aus diesem Grund wird in dieser Arbeit eine einfache und gleichzeitig effektive synthetische Minderheiten-Oversampling-Technologie (SMOTE) vorgeschlagen, bekannt als ASN-SMOTE, die adaptiv qualifizierte Minderheitsnachbarn nominiert, um Minderheitsinstanzen zu synthetisieren. Genauer gesagt, werden für jede Instanz der Minderheitenklasse die Euklidische Distanzen zwischen ihr und allen anderen Instanzen zunächst berechnet. Danach wird die Minderheiteninstanz, deren nächstgelegene Nachbar eine Mehrheitsinstanz ist, als Rauschen betrachtet und herausgefiltert. Ist hingegen der nächste Nachbar der Minderheiteninstanz ebenfalls eine Minderheiteninstanz, so wird  
der nächste Nachbar der Minderheiteninstanz ebenfalls eine Minderheit ist, wird der qualifizierte Synthesizer k-nächsten Nachbarn (KNN) anhand der euklidischen Abstände zu ihnen. Schließlich werden diese qualifizierten Nachbarn verwendet, um Minderheiteninstanzen mit Hilfe des SMOTE-Algorithmus [13] zu erzeugen. Die synthetisierten Minoritätsinstanzen werden mit einem unausgewogenen Datensatz kombiniert und in einen traditionellen überwachten Klassifikator zum Trainieren. ASN-SMOTE kann auf reale Datensätze angewendet werden und die Genauigkeit der überwachten Klassifikatoren ohne Änderungen.

Die Hauptvorteile von ASN-SMOTE sind dreifach:

1. Es kann Rauschen herausfiltern und die Effektivität von Over-Sampling verbessern.
2. Es nutzt die Informationen des unausgewogenen Datensatzes, um adaptiv qualifizierte Synthesizer auszuwählen, was die synthetisierten Minderheiteninstanzen  
   genauere Anpassung an die praktische Datenverteilung.
3. (3) Es kann mit jedem Klassifikationsalgorithmus kombiniert werden und es besteht keine Notwendigkeit die Klassifizierungsalgorithmen zu ändern.

Um die Leistung von ASN-SMOTE zu bewerten, wurden umfangreiche vergleichende Experimente durchgeführt. Das vorgeschlagene ASN-SMOTE wurde auf 24 öffentliche Benchmark-Datensätze angewandt mit einem Ungleichgewichtsverhältnis von 1,82 bis 41,4. Drei Klassifikatoren, nämlich der k-nearest neighbor (KNN) [18], SVM und Random Forest (RF), wurden für die Klassifizierung verwendet.

**SMOTE**

**1. Einführung**  
Ein Datensatz ist unausgewogen, wenn die Klassen nicht annähernd gleich stark vertreten sind. Unausgewogenheit  
in der Größenordnung von 100 zu 1 ist bei der Betrugserkennung weit verbreitet, und Ungleichgewichte von bis zu 100.000 zu  
c 2002 AI Access Foundation und Morgan Kaufmann Publishers. Alle Rechte vorbehalten.  
Chawla, Bowyer, Hall & Kegelmeyer  
1 wurde in anderen Anwendungen berichtet (Provost & Fawcett, 2001). Es gab bereits  
Versuche, mit unausgewogenen Datensätzen in Bereichen wie betrügerischen Telefonanrufen umzugehen  
(Fawcett & Provost, 1996), Telekommunikationsmanagement (Ezawa, Singh, & Norton,  
1996), Textklassifikation (Lewis & Catlett, 1994; Dumais, Platt, Heckerman, & Sahami,  
1998; Mladeni'c & Grobelnik, 1999; Lewis & Ringuette, 1994; Cohen, 1995a) und Erkennung  
von Ölverschmutzungen in Satellitenbildern (Kubat, Holte, & Matwin, 1998).  
Die Leistung von Algorithmen für maschinelles Lernen wird in der Regel anhand der Vorhersagegenauigkeit bewertet.  
Genauigkeit bewertet. Dies ist jedoch ungeeignet, wenn die Daten unausgewogen sind und/oder die Kosten der  
verschiedenen Fehlern stark variieren. Als Beispiel sei hier die Klassifizierung von Pixeln in Mammogrammbildern als möglicherweise krebsartig genannt (Woods, Doss, Bowyer, Solka, Priebe, & Kegelmeyer,  
1993). Ein typischer Mammographiedatensatz könnte 98% normale Pixel und 2% abnormale Pixel enthalten.  
Pixel. Eine einfache Standardstrategie, bei der die Mehrheitsklasse erraten wird, würde eine Vorhersagegenauigkeit von 98 % ergeben. Die Art der Anwendung erfordert jedoch eine ziemlich hohe Rate korrekter  
Erkennungsrate in der Minderheitenklasse und erlaubt eine kleine Fehlerrate in der Mehrheitsklasse, um  
um dies zu erreichen. Eine einfache Vorhersagegenauigkeit ist in solchen Situationen eindeutig nicht angemessen. Die Receiver Operating Characteristic (ROC)-Kurve ist eine Standardtechnik zur  
Klassifikatorleistung über einen Bereich von Kompromissen zwischen wahren positiven und falschen  
positiven Fehlerraten (Swets, 1988). Die Fläche unter der Kurve (Area Under the Curve, AUC) ist eine anerkannte traditionelle Leistungsmetrik für eine ROC-Kurve (Duda, Hart, & Stork, 2001; Bradley, 1997; Lee,  
2000). Die konvexe ROC-Hülle kann auch als robuste Methode zur Identifizierung potenziell  
optimale Klassifikatoren (Provost & Fawcett, 2001). Wenn eine Linie durch einen Punkt auf der konvexen Hülle  
konvexen Hülle verläuft, dann gibt es keine andere Linie mit der gleichen Steigung, die durch einen anderen Punkt mit einem  
größeren wahr-positiven (TP) Achsenabschnitt. Somit ist der Klassifikator an diesem Punkt optimal unter allen  
Verteilungsannahmen in Verbindung mit dieser Steigung

Die Gemeinschaft des maschinellen Lernens hat das Problem des Klassenungleichgewichts auf zwei Arten angegangen.  
Die eine besteht darin, den Trainingsbeispielen unterschiedliche Kosten zuzuweisen (Pazzani, Merz, Murphy, Ali, Hume, &  
Brunk, 1994; Domingos, 1999). Die andere besteht darin, den ursprünglichen Datensatz erneut zu beproben, indem entweder die Minderheitenklasse überbeprobt und/oder die Mehrheitsklasse unterbeprobt wird (Kubat & Matwin,  
1997; Japkowicz, 2000; Lewis & Catlett, 1994; Ling & Li, 1998). Unser Ansatz (Chawla,  
Bowyer, Hall, & Kegelmeyer, 2000) verbindet die Unterauswahl der Mehrheitsklasse mit einer  
speziellen Form der Überstichprobe der Minderheitenklasse. Experimente mit verschiedenen Datensätzen und  
dem C4.5 Entscheidungsbaum-Klassifikator (Quinlan, 1992), Ripper (Cohen, 1995b) und einem Naive Bayes  
Klassifikator zeigen, dass unser Ansatz besser ist als andere frühere Ansätze für Re-Sampling, Modifizierung der Verlustquote  
Ratio und Klassenprioritäten unter Verwendung der konvexen AUC- oder ROC-Hülle.  
Abschnitt 2 gibt einen Überblick über die Leistungsmaße. Abschnitt 3 gibt einen Überblick über die wichtigsten  
eng verwandten Arbeiten, die sich mit unausgewogenen Datensätzen befassen. Abschnitt 4 enthält die Einzelheiten  
unseres Ansatzes. In Abschnitt 5 werden experimentelle Ergebnisse vorgestellt, die unseren Ansatz mit anderen  
Re-Sampling-Ansätzen. In Abschnitt 6 werden die Ergebnisse diskutiert und Vorschläge für zukünftige  
Arbeit vor.

**ASN-SMOTE Algorithmus**

ASN-SMOTE filtert zunächst das Rauschen in der Minderheitenklasse heraus, indem es feststellt, ob der nächste Nachbar jeder Minderheiteninstanz zu der Minderheiten- oder Mehrheitsklasse gehört. Danach verwendet ASN-SMOTE die nächstgelegene Mehrheitsinstanz jeder Minderheitsinstanz, um die Entscheidungsgrenze wahrzunehmen, innerhalb derer die qualifizierten Minderheiteninstanzen für jede Minderheiteninstanz adaptiv ausgewählt werden durch das vorgeschlagene adaptive Nachbarschaftsauswahlverfahren ausgewählt werden, um eine neue Minderheiteninstanz zu synthetisieren.

**2. Leistungsmessungen**  
Die Leistung von Algorithmen für maschinelles Lernen wird in der Regel anhand einer Konfusionsmatrix bewertet  
wie in Abbildung 1 (für ein Problem mit 2 Klassen) dargestellt. Die Spalten sind die vorhergesagte Klasse und die  
Zeilen sind die tatsächliche Klasse. In der Verwechslungsmatrix ist T N die Anzahl der negativen Beispiele  
  
Abbildung 1: Konfusionsmatrix  
richtig klassifiziert (True Negatives), F P ist die Anzahl der negativen Beispiele, die fälschlicherweise  
Beispiele, die fälschlicherweise als positiv klassifiziert wurden (False Positives), F N ist die Anzahl der positiven  
negativen Beispiele (False Negatives) und T P ist die Anzahl der positiven Beispiele, die richtig  
klassifizierten positiven Beispiele (True Positives).  
Die Vorhersagegenauigkeit ist das Leistungsmaß, das im Allgemeinen mit Algorithmen des maschinellen Lernens assoziiert wird, und ist definiert als Genauigkeit = (T P + T N)/(T P + F P + T N + F N). Im Kontext  
Zusammenhang mit ausgeglichenen Datensätzen und gleichen Fehlerkosten ist es sinnvoll, die Fehlerrate als  
Leistungsmaßstab zu verwenden. Die Fehlerrate ist 1 - Genauigkeit. Bei unausgewogenen Datensätzen  
mit ungleichen Fehlerkosten ist es sinnvoller, die ROC-Kurve oder andere ähnliche Techniken zu verwenden  
Verfahren zu verwenden (Ling & Li, 1998; Drummond & Holte, 2000; Provost & Fawcett, 2001; Bradley,  
1997; Turney, 1996).  
ROC-Kurven können als die Familie der besten Entscheidungsgrenzen für  
relativen Kosten von TP und FP. Auf einer ROC-Kurve stellt die X-Achse %F P = F P/(T N+F P)  
und die Y-Achse stellt %T P = T P/(T P +F N) dar. Der ideale Punkt auf der ROC-Kurve wäre  
(0,100) sein, d. h. alle positiven Beispiele werden richtig klassifiziert und keine negativen Beispiele werden  
falsch als positiv klassifiziert. Eine Möglichkeit, eine ROC-Kurve auszufahren, besteht darin, die  
Gleichgewicht der Trainingsbeispiele für jede Klasse im Trainingssatz manipuliert wird. Abbildung 2 zeigt eine Veranschaulichung.  
Die Linie y = x stellt das Szenario einer zufälligen Schätzung der Klasse dar. Fläche unter der ROC  
Kurve (AUC) ist ein nützliches Maß für die Leistung des Klassifikators, da es unabhängig vom gewählten Entscheidungskriterium  
gewählten Entscheidungskriterium und den vorherigen Wahrscheinlichkeiten ist. Der AUC-Vergleich kann eine Dominanz  
Beziehung zwischen den Klassifikatoren herstellen. Wenn sich die ROC-Kurven überschneiden, ist der Gesamt-AUC ein  
Durchschnittsvergleich zwischen den Modellen (Lee, 2000). Für einige spezifische Kosten- und Klassenverteilungen  
Verteilungen kann der Klassifikator mit dem höchsten AUC tatsächlich suboptimal sein. Daher berechnen wir  
berechnen wir auch die konvexen ROC-Hüllen, da die Punkte, die auf der konvexen ROC-Hülle liegen  
potentiell optimal sind (Provost, Fawcett, & Kohavi, 1998; Provost & Fawcett, 2001).