



Universität  
Bremen

# Interpretable Machine Learning and Generative Modeling with Mixed Tabular Data

Advancing Methodology from the Perspective of Statistics

Kristin Blesch

## Dissertation

zur Erlangung des akademischen Grades  
*Doktor der Naturwissenschaften* Dr.rer.nat.

Universität Bremen

Fachbereich 03: Mathematik und Informatik

Januar 2024



Erster Gutachter: Prof. Dr. Marvin N. Wright  
Zweite Gutachterin: Prof. Dr. Barbara Hammer  
Tag des Kolloquiums: 05.04.2024



## Acknowledgments

*I am deeply grateful to the many individuals whose help, support, guidance, and advice made this thesis possible. In particular, I would like to express my sincere thanks to ...*

- ... Prof. Dr. Marvin N. Wright for his support and excellent supervision throughout my PhD journey. Thank you for always having an open ear, engaging in constructive discussions, and giving me the trust and freedom I needed to thrive in research. I am grateful that you took me on as one of your research group's first members, and I am overjoyed to have seen this group blossom over time!*
- ... Prof. Dr. Barbara Hammer, for agreeing to act as the secondary examiner of this thesis.*
- ... Dr. David S. Watson and the Department of Informatics at King's College London for hosting me for a six-month research stay in the second year of my PhD. This experience helped me find my research identity and become an independent researcher.*
- ... The Leibniz Institute for Prevention Research and Epidemiology – BIPS for being a supportive employer, promoting my academic adventures, and enabling me to conduct independent research.*
- ... my close working colleagues at BIPS from the working groups of Prof. Dr. Marvin N. Wright and Prof. Dr. Vanessa Didelez for the nice working atmosphere, sense of togetherness over great lunch breaks, and even better coffee rounds. Thank you for the fun team activities in the working context and in our free time!*
- ... funds that supported activities that brought me in contact with other researchers. This boosted my motivation and would not have been possible without financial support. In particular, thanks to ERASMUS+ and the Minds Media Machines Integrated Graduate School at the University of Bremen.*
- ... the joint journal club with the IML/XAI working group at LMU Munich for fruitful discussions. I found great motivation through the regular meetings, allowing me to establish meaningful connections with other researchers.*
- ... fellow PhD students and other academics I met at conferences, workshops, journal clubs, etc., for being a source of academic inspiration and becoming friends over time – many thanks in particular to the proofreaders of this thesis!*
- ... my friends in Bremen and scattered around Europe for the support throughout this PhD. I am deeply thankful for sharing frustrating moments and celebrating successes together.*
- ... my family for unlimited support during this PhD and my entire life. In particular, I want to thank my parents for always believing in me, my sister for regularly reminding me that there's a life to enjoy beyond research, and my aunt for magnificent local support in Bremen.*



## Summary

Artificial intelligence algorithms are ubiquitous in today's world and administer various tasks. However, comprehending the behavior of such algorithms is challenging for human beings. As a remedy, the academic field of explainable artificial intelligence (XAI), also referred to as interpretable machine learning (IML), has emerged to help generate insights into an algorithm's actions.

Developments in IML are shaped by academic perspectives, e.g., from the field of computer science, that frequently overlook statistical concerns. However, statistical considerations can inspire favorable improvements for IML methods. This thesis examines IML methodology from the perspective of statistics and advances methodology accordingly. Through the lens of statistics, it is apparent that IML methods often disregard the unique requirements imposed by real-world data. Specifically, empirical data sets regularly consist of diverse data types and exhibit dependency structures. Statistical method development routinely considers such traits, however, such awareness is yet to be established in IML. This thesis demonstrates the far-reaching implications of this disregard and advances methodology to counter this: data-adequate methods can yield favorable outcomes in interpretability and machine learning more broadly. Further, the work highlights the necessity for providing readily available software to transfer methodological advancements to real-world data applications.

In specific, this thesis centers around mixed tabular data. That is, data in a table-like format consisting of continuous and categorical features, i.e., a mixture of both data types. Previous literature mainly focuses on only one of the two data types, typically all-continuous data for which theoretical properties are often more straightforward to prove. Even though some methodological work states it would extend to categorical features or mixed data, this rarely provides concrete, ready-to-use methods. Consequently, practitioners encounter difficulties in real-world applications with mixed tabular data. Workarounds may work technically, but the consequences of, e.g., dummy encoding categorical features and treating them as continuous, remain largely unexplored.

A contributing paper of this thesis proposes a specialized method to measure conditional feature importance with mixed tabular data. The paper evaluates the consequences of alternative procedures that rely on workarounds and reveals that the method proposed achieves higher statistical power in testing for conditional feature importance with mixed tabular data.

Another contributing paper of this thesis highlights that embracing the traits of mixed tabular data can further leverage simplified algorithms: the introduction of adversarial random forests contributes to the field of generative modeling and may likewise be helpful in subroutines of IML methods. The work demonstrates that for mixed tabular data, data synthesis relying on this tree-based procedure rather than deep learning-based alternatives can yield competitive results more straightforwardly and efficiently. To encourage empirical applications, an accompanying software implementation in the *Python* programming language and a tutorial-style paper enrich the methodological contribution, promoting its use for a broad audience.

Besides the mixed data type, dependency structures are crucial in applications with real-world tabular data. Often, IML methods implicitly assume mutual independence of features. However, ignoring the dependency structure between features can result in misleading outcomes. This misguidance concerns the interpretation of the IML explanations and their robustness against

manipulations. For example, a disregard for dependency structures opens up the possibility for adversarial attackers to generate arbitrary, i.e., manipulated, explanations. A contributing paper of this thesis demonstrates that respecting dependency structures is decisive to prevent adversarial attacks from unfolding. So-called model-X knockoffs (newly generated synthetic features that mimic the structure of some given data) satisfy desirable statistical properties that are particularly useful to effectively prevent adversarial attacks.

In sum, this thesis emphasizes that statistical considerations for mixed tabular data with a dependency structure are vital for IML. The contributing papers of this cumulative thesis introduce methods alongside open-source software to advance statistical adequacy for real-world applications.



## Zusammenfassung

Künstliche Intelligenz ist in der heutigen Welt allgegenwärtig und wird für verschiedenste Aufgaben eingesetzt, jedoch ist es für Menschen herausfordernd die Verhaltensweisen der zu Grunde liegenden Algorithmen zu verstehen. Um dem Abhilfe zu schaffen und Einblicke in das Verhalten der Algorithmen zu erlangen, ist das Forschungsfeld der erklärbaren künstlichen Intelligenz, auch als interpretierbares maschinelles Lernen (IML) bezeichnet, aufgekommen.

Entwicklungen in IML sind geprägt von akademischen Perspektiven, wie beispielsweise aus der Informatik, welche über statistische Aspekte oftmals hinwegsehen, allerdings können Überlegungen aus dem Fachbereich der Statistik zu vorteilhaften Weiterentwicklungen der IML Methoden beitragen. Diese Dissertation untersucht IML-Methoden in Hinblick auf statistische Betrachtungsweisen und erweitert die Methoden dem entsprechend. Durch statistische Einblicke wird deutlich, dass IML-Methoden oftmals die speziellen Gegebenheiten von empirischen Datensätzen übersehen, denn diese beinhalten meist verschiedene Datentypen und sind gekennzeichnet durch Abhängigkeitsstrukturen. Statistische Methoden berücksichtigen standardmäßig solche Gegebenheiten, wohingegen dies bei IML-Methoden bisher nicht etabliert ist. Diese Dissertation legt die weitreichenden Folgen dieser Missachtung dar und entwickelt Lösungen dem entgegenzuwirken: Adäquate Methoden, welche diese Gegebenheiten berücksichtigen, liefern vorteilhafte Erkenntnisse sowohl in IML, als auch dem maschinellen Lernen generell. Des Weiteren trägt diese Arbeit dem Transfer methodischer Weiterentwicklungen bei, indem sie Software bereitstellt, deren leichte Zugänglichkeit für eine Anwendung mit realen Daten als essentiell betrachtet wird.

Im Detail beschäftigt sich diese Dissertation mit gemischten tabularen Daten. Dies sind Daten, die in einem Tabellenformat vorliegen und sowohl kontinuierliche als auch kategorielle Variablen beinhalten, also eine Mischung dieser Datentypen umfassen. Die bisherige Literatur setzt den Fokus typischerweise auf nur einen der genannten Datentypen, meist kontinuierliche Daten, da hierfür das Zeigen theoretischer Eigenschaften oftmals vereinfacht ist. Obwohl in der Literatur vorgeschlagene Methoden stellenweise behaupten, dass diese auch auf den Fall von kategoriellen oder gemischten Daten erweitert werden können, fehlt hierbei häufig die Bereitstellung konkreter Methoden und Software. Aufgrund dessen begegnen Anwender, die mit gemischten Daten in ihrer Anwendung konfrontiert sind, häufig Herausforderungen. Eine Umgehung durch Workarounds mag aus technischer Sicht funktionieren (beispielsweise eine Dummy-Kodierung von Variablen und deren Verwendung als kontinuierliche Variablen), aber die Konsequenzen dessen sind bisher weitgehend unerforscht.

Ein Beitrag dieser kumulativen Dissertation schlägt eine spezialisierte Methode für das Messen von bedingter Variablenwichtigkeiten mit gemischten tabularen Daten vor. Der Beitrag analysiert die Auswirkungen von alternativen Methoden die auf Workarounds beruhen und demonstriert, dass die vorgestellte, spezialisierte Methode für das Testen von bedingter Variablenwichtigkeit eine verbesserte statistische power liefert.

Ein weiterer Beitrag dieser Dissertation unterstreicht, dass das Ausnutzen der besonderen Eigenschaften gemischter Daten vorteilhaft sein kann, um vereinfachte Algorithmen zu entwickeln: Die vorgestellte Methode „Adversarial Random Forests“ ermöglicht das Generieren synthetischer Daten auf vereinfachte Weise, was eine nützliche Subroutine in IML-Methoden darstellt. Der Beitrag veranschaulicht, dass gemischte synthetische tabulare Daten durch baumbasierte Methoden in vereinfachter und effizienterer Weise erzeugt werden können, als durch konkurrierende Methoden, welche auf Deep Learning basieren. Um die Anwendung der vorgestellten Methode für

ein breites Publikum zugänglich zu machen, ist die Bereitstellung als Software Implementierung in der Programmiersprache *Python*, inklusive eines Tutorials für dessen Verwendung, Teil dieser Dissertation.

Neben den gemischten Datentypen spielen Abhängigkeitsstrukturen eine wichtige Rolle in empirischen Anwendungen mit tabularen Daten. Häufig treffen IML-Methoden jedoch die Annahme, dass die Variablen untereinander unabhängig seien. Ein Missachten der Abhängigkeitsstrukturen kann in fehlgeleiteten Resultaten münden. Diese Fehlleitung betrifft sowohl die Interpretation der Resultate, als auch deren Robustheit gegenüber Manipulationsversuchen. Beispielsweise eröffnet das außer Acht lassen von Abhängigkeitsstrukturen die Möglichkeit willkürliche Erklärungsergebnisse durch IML-Methoden zu generieren, sie also zu manipulieren. Ein Beitrag dieser Dissertation demonstriert, dass das Beachten und Respektieren von Abhängigkeitsstrukturen ausschlaggebend ist um Manipulationsversuche zu unterbinden. Sogenannte Model-X Knockoffs (synthetische Variablen die die Datenstruktur der echten Variablen imitieren) erfüllen vorteilhafte statistische Eigenschaften, welche es ermöglichen solche Manipulationsversuche abzuwehren.

Zusammenfassend lässt sich festhalten, dass statistische Überlegungen für gemischte tabulare Daten mit Abhängigkeitsstrukturen essenziell für IML-Methoden sind. Die Beiträge dieser Dissertation bringen spezialisierte Methoden für solche Daten hervor und stellen öffentlich zugängliche Software bereit um die statistische Angemessenheit für IML Anwendungen zu verbessern.

# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Motivation	1
1.2. Context and Scope: Mixed Tabular Data with Dependency Structures	3
1.3. Outline	5
<b>2. Background</b>	<b>7</b>
2.1. Interpretable Machine Learning	7
2.1.1. Supervised Machine Learning	7
2.1.2. Interpretability	8
2.1.3. Feature Importance Measurement	9
2.1.4. A Selection of Post Hoc Model-agnostic Interpretability Methods	11
2.2. Generative Modeling	16
2.2.1. Model-X Knockoffs	18
<b>3. Gap in Research and Contribution</b>	<b>23</b>
3.1. Gap in Research	23
3.2. Contribution Overview	25
3.3. Individual Contributions: Parts I, II and III	25
3.4. Connections between Parts: Joint Contribution	30
<b>I. Conditional Feature Importance Measurement with Mixed Tabular Data</b>	<b>33</b>
Paper 1. Conditional Feature Importance for Mixed Data	35
<b>II. Adversarial Attack Robustness in IML</b>	<b>57</b>
Paper 2. Unfooling SHAP and SAGE: Knockoff Imputation for Shapley Values	59
<b>III. Generative Modeling with Mixed Tabular Data</b>	<b>77</b>
Paper 3. Adversarial Random Forests for Density Estimation and Generative Modeling	79
Paper 4. Arfpy: A Python Package for Density Estimation and Generative Modeling with Adversarial Random Forests	101
<b>IV. Conclusion and Discussion</b>	<b>111</b>
<b>4. Conclusion</b>	<b>113</b>
<b>5. Discussion and Future Work</b>	<b>115</b>
<b>References</b>	<b>119</b>



# 1. Introduction

## 1.1. Motivation

Artificial intelligence (AI) plays a pivotal role in shaping the modern world. Through recent advancements in computing power and an increased amount of digitized data, AI has become omnipresent, and a large variety of fields nowadays apply powerful machine learning algorithms for diverse tasks, e.g., in finance (Dixon et al., 2020), medicine (Rajkomar et al., 2019), justice (Deeks, 2019) or engineering (Thai, 2022). The disruptive role of this technology particularly concerns designing new ways of modeling data relationships, predicting outcomes, and generating new data. For example, tools like ChatGPT (OpenAI, 2023) use large language models, a special kind of AI, to generate compelling texts, which has received considerable attention recently and propelled the topic of AI at the heart of many contemporary societal and scientific debates. The public frequently discusses not only hopes, but also fears associated with AI and may even create a dystopian prophecy around AI (Cave and Dihal, 2019). A factor contributing to such an AI anxiety is the opacity, i.e. lack of transparency, in AI systems (Li and Huang, 2020).

As a remedy to this black box nature, concepts for explainability and interpretability in AI and machine learning have emerged (Du et al., 2019; Gilpin et al., 2018; Guidotti et al., 2018; Molnar, 2020; Murdoch et al., 2019). Such approaches aspire to enhance human understanding and while the origins of the field extend further into the past (Molnar et al., 2020), they have attracted considerable interest in recent years (Adadi and Berrada, 2018; Leblanc and Germain, 2023; Saeed and Omlin, 2023). However, explainability and interpretability are defined inconsistently in the literature and the term interpretable machine learning (IML) may or may not be used interchangeably with explainable artificial intelligence (XAI) (Doshi-Velez and Kim, 2017; Gilpin et al., 2018; Leblanc and Germain, 2023). With a particular focus on machine learning algorithms, this thesis adheres to the term IML, and understands IML as a tool for extracting relevant knowledge from a machine learning model on the relationships learned by it or present in the data (Murdoch et al., 2019). That is, IML aspires to unveil relevant mechanisms in machine learning algorithms to provide valuable insights for various stakeholders.

A broad spectrum of use cases can benefit from IML explanations. For example, in software engineering, IML can help to uncover flaws in algorithms, which assists in debugging code. To illustrate this, we can think of IML techniques for image recognition that illuminate whether the classification of an object was driven by meaningful contents of the image or, in fact, by watermarks only (Lapuschkin et al., 2019). In another context, IML can help fulfill legal requirements for deploying black box machine learning models to the real world. For example, the European data protection regulation grants individuals subject to an automated decision process, e.g., a decision reached by the prediction of a machine learning model, a 'right to explanation' (Goodman and Flaxman, 2017) and IML can help deduct such explanations for individuals. On a more societal level, it might be desirable to assess whether machine learning models fulfill ethical standards in decision-making, e.g., whether models are discriminatory against population subgroups. Such

ethical considerations may be evaluated in auditing procedures of legal authorities, yet may also be misused for fairwashing (Aivodji et al., 2019). Still, IML techniques can uncover a model’s behavior and, by doing so, aid in avoiding undesirable outcomes. Such applications are highly relevant given the increased usage of machine learning generally and in high-risk domains such as health or justice (Adadi and Berrada, 2018). In sum, IML can assist in improving the entire model deployment pipeline, including debugging, testing, and auditing the model prior to real-world deployment (Murdoch et al., 2019).

Numerous IML methods have been proposed to shed light on the diverse aspects of machine learning, with a standpoint from computer science contributing to many proposals. It appears natural to seek advice about the inner workings of the algorithms from the researchers who developed machine learning algorithms in the first place. Consequently, computer scientists have engaged widely in developing methodology to explain the black box machine learning models. IML may even be categorized as a subdiscipline of computer science (Watson, 2022b). This dynamic thus shaped the relatively young field of IML from a computer science viewpoint. However, explaining a model’s behavior and modeling data is also studied in other research fields; hence, valuable insight from different areas could be passed along to IML methodology.

The field of statistics is, just like machine learning, concerned with modeling data, yet it follows a fundamentally different standpoint. Machine learning models typically focus on predictive performance, whereas statistical models prioritize inference and immediate interpretability. Machine learning models enhance predictive performance by flexibly adapting to the data. However, this flexibility comes at the cost of yielding so-called black box models that are no longer comprehensible to humans. Statistical models, on the contrary, are designed for inherent interpretability. A linear regression model, for example, allows for direct interpretation of model coefficients. Some authors argue that if interpretability is of interest, directly interpretable models (white box models) should be given priority over deploying and attempting to explain black box models (Rudin, 2019). Nonetheless, depending on the application, more flexible black box models may still be favored.

Instead of framing different viewpoints as competing approaches, bridging fields could lead to favorable outcomes. Shifting the focus away from the task of model design itself helps to understand the mechanisms for accomplishing this: a statistical perspective on IML concentrates on data characteristics, such as data types and distributions, and their alignment with a method’s assumptions. Thus far, the statistical adequacy of IML methods is barely acknowledged, with far-reaching implications for the resulting explanations. If real data mismatches the implicit assumptions of IML methods or implications of the method’s underlying assumptions are disregarded, IML explanations are prone to be misleading. After all, to deliver trustworthy and meaningful interpretability results, IML methods themselves must be trustworthy, and statistical considerations can advance IML methodology to help reach this goal.

A central part of many IML methods requires modeling data distributions and generating feature values accordingly to deduct explanations. In other words, IML methods frequently rely on replacing feature values with newly sampled values, for which obeying the data distribution is often pivotal. The learning of data distributions and subsequent synthesis of new data points is at the core of a related research field – generative modeling.

Generative modeling has received outstanding popularity through tools like ChatGPT for text generation (OpenAI, 2023) and DALL-E for image generation (Ramesh et al., 2022) and is a

## 1.2 Context and Scope: Mixed Tabular Data with Dependency Structures

---

widely discussed topic in both society and academic research. However, as with IML, generative modeling is a relatively young research area shaped mainly by computer scientists and statistical considerations such as data adequacy have thus far received little attention. In particular, the specifics of data represented in a table format (as is the standard pattern in statistics) have been neglected under a one-fits-all spirit of deploying deep learning architectures to any data, including image, text, audio, or tabular data. In supervised machine learning, a similar dynamic is apparent. However, in this research area authors recently advocated for a more careful evaluation, yielding provocatively titled works such as “Tabular data: deep learning is not all you need” (Shwartz-Ziv and Armon, 2022). Similarly in generative modeling, algorithms other than deep learning approaches may better suit applications with tabular data. Leveraging the statistical, data-centered perspective to a generative modeling, it becomes apparent that embracing data-specific characteristics is not necessarily an obstacle to overcome but can serve as a means to yield beneficial algorithms.

### 1.2. Context and Scope: Mixed Tabular Data with Dependency Structures

This thesis discusses, develops, and advances methods for improved data adequacy in IML and generative modeling through the perspective of statistics. This work focuses on mixed tabular data that exhibits dependency structures, which is a frequently occurring data type in real-world applications. This subsection exposes the unique characteristics of such data and relates them to machine learning methodology. Doing so narrows down the context and scope of this thesis and sets the groundwork for respecting and leveraging these traits in method development.

**Tabular Data** Tabular data generally describes data that appears in a table format. That is, data that intrinsically arrives in a structured form, with each row indicating an individual observation (instance) and each column corresponding to an attribute of this observation (feature). Tabular data is heterogeneous, exhibiting unique traits that differ from homogeneous data such as image or text data,<sup>1</sup> which poses unique challenges on machine learning algorithms (Borisov et al., 2022).

The specialties of tabular data are vital to be taken into account for the development and application of machine learning methodology. For example, deep neural networks may outperform other state-of-the-art methods on image classification tasks (Krizhevsky et al., 2012). However, for the prediction of tabular data, tree-based methods may be preferable over neural networks (Borisov et al., 2022; Grinsztajn et al., 2022; Shwartz-Ziv and Armon, 2022). As another example, interpretability with image data may benefit from using heatmaps that highlight relevant pixels in an image (Bach et al., 2015). However, for tabular data, which is represented by a data matrix filled with feature values, scores and rankings might be more helpful in explaining a black box model. These examples show that different data types have different requirements, and adequate methodology is needed to match those.

---

<sup>1</sup>For example, tabular data sets often exhibit dense numerical and sparse categorical features with correlations that are weaker than those caused by spatial or semantic relationships in images or texts, respectively (Borisov et al., 2022).

**Mixed Tabular Data** A more nuanced view reveals that within the tabular data type, a further refinement – mixed tabular data – necessitates tailored algorithms. Mixed tabular data specifies tabular data that includes at least one continuous and at least one categorical feature. As the name suggests, continuous features take on values on a continuous scale, whereas categorical features only exhibit  $c$  distinct nominal values (category levels) that often have no inherent ordering. Research frequently proposes methodology suitable for only continuous features or categorical features (Aas et al., 2021; Chen et al., 2020; Romano et al., 2020; Sesia et al., 2018). The mixed data case has thus far received considerably less attention, even though some branches of machine learning recognize the relevance of this issue, e.g., in Bayesian optimization (Ru et al., 2020). Methodology proposed may – in principle – extend to the mixed case; however, concrete methods and software that is ready to use for practitioners are often lacking in the initial proposition of a method and, instead, is developed by follow-up research such as Luo et al. (2021) or Redelmeier et al. (2020). If suitable methods for mixed tabular data are absent, practitioners are forced to apply workarounds using encoding schemes (see Hancock and Khoshgoftaar (2020) and Pargent et al. (2022) for an overview) that work technically but may be disadvantageous from a statistical perspective. For example, performing one-hot encoding on categorical features with values of 0 or 1 and then treating them as continuous violates the actual data distribution and inflates the dimensionality of the data matrix. Another example is integer encoding, where category levels are assigned an integer number, which induces an artificial ordering of categories. This thesis devotes special attention to mixed tabular data, investigating the implications of such workarounds and proposing ready-to-use methods for practitioners faced with mixed tabular data.

**Dependency Structures** Tabular data can be examined more closely by considering how the features in a data set relate to each other. Due to an underlying dependency structure between features, some feature values may be affected by other feature values. For example, assessing the Pearson correlation coefficient between features reflects the degree to which feature values linearly change in the same (or opposite) direction, i.e., indicates linear associations between features. More generally, statistical (in-)dependencies between features reflect essential aspects of the data generating process and may even indicate causal relationships.<sup>2</sup> For machine learning models, the dependency structure between features in an empirical data set can play a crucial role. For example, generative models have to aptly pick up the dependency structure in the data in order to synthesize data that appears realistic.

For interpretability in machine learning, dependency structures between features have far-reaching implications on the interpretation and robustness of the methods, further discussed in Chapter 3. In brief, an ignorance of dependency structures can result in misleading explanations. However, dependency structures are scarcely discussed in IML and, therefore, are prone to remain unrecognized in practical application. Some algorithms even assume independence across features, e.g., to calculate approximations (Lundberg and Lee, 2017), which often mismatches the state of real data. This thesis discusses the consequences of disregarding dependency structures in IML and proposes methodological advancements to account for dependency structures.

---

<sup>2</sup>Note that causal discovery based on feature independencies is a separate field of research that requires further assumptions and should not be confused with insights deducted by IML. For an introductory overview of causality research and causal structure learning, see Pearl (2009) or Peters et al. (2017).



## 1.3 Outline

---

**Relevance for Real Data Applicability** In many real-world applications, tabular data sets exhibit both phenomena – mixed features *and* dependency structures. For example, socio-economic data sets may include information on the monthly salary (continuous) and level of education (categorical) of people, or medical data sets may incorporate the age of a person (continuous) and information on whether the person is vaccinated against the flu or not (categorical). In both examples, it is reasonable to suspect dependency structures arising in the empirical data, e.g., the monthly salary may, on average, be higher for higher levels of education, and older adults may be more likely to have received a flu vaccine.

Developing methods that account for mixed tabular data with a dependency structure is, therefore, a pursuit of aligning theory with the circumstances practitioners face in application. This thesis contributes to generating valuable insights and developing methods for IML and generative modeling by addressing the unique requirements of mixed tabular data with a dependency structure. Applicability to real-world problems is further encouraged by making methods accessible to practitioners through user-friendly, open-source software implementations.

### 1.3. Outline

The remainder of this thesis is structured as follows. Chapter 2 introduces relevant methodological concepts in IML (Section 2.1) and generative modeling (Section 2.2). After a discussion on interpretability in the context of supervised machine learning (Sections 2.1.1 and 2.1.2), the focus is directed to feature importance measurement (Section 2.1.3). Section 2.1.4 presents several methods for model-agnostic, post hoc explanations. Further, Section 2.2 draws attention to generative modeling at a conceptual level before delving into the subfield of model-X knockoffs in Section 2.2.1.

Chapter 3 examines the gap in research this thesis centers around (Sections 3.1 and 3.2). Subsequently, the aspects addressed by each of the three parts contributing to this thesis individually (Section 3.3) and in connection to each other (Section 3.4) are summarized. Following this, three parts present the contributing papers of this thesis in full length.

Part I discusses the measurement of conditional feature importance measurement with mixed tabular data and proposes a specialized method to do so (Paper 1). Part II highlights the consequences of neglecting dependency structures in feature importance measurement in the context of manipulations to the explanation (adversarial attacks) and presents a strategy to defend against such manipulations by accounting for the dependency structure in the data (Paper 2). Finally, Part III extends considerations on mixed tabular data with dependency structures to generative modeling and introduces a method that is particularly suitable for this kind of data (Paper 3). Moreover, Part III provides a user-friendly software implementation in the *Python* programming language and a tutorial on its usage (Paper 4) to enhance applicability to real-world scenarios.

Chapter 4 concludes the findings of this thesis before discussing them in a broader context and outlining promising directions for future research in Chapter 5.



## 2. Background

### 2.1. Interpretable Machine Learning

#### 2.1.1. Supervised Machine Learning

Given a data set  $\mathcal{D} = \{(\mathbf{x}^{(i)}, y^{(i)})\}_{i=1}^n$ , resembling  $n$  observations of the  $p$  random features  $\mathbf{X} = \{X_1, \dots, X_p\}$  and target  $Y$ , sampled i.i.d. from the joint probability distribution  $\mathbb{P}_{\mathbf{X}, Y}$ . A supervised machine learning model aims to learn the functional relationship  $f$  that maps the feature space  $\mathcal{X} = \{\mathcal{X}_1 \times \dots \times \mathcal{X}_p\}$  to the target space  $\mathcal{Y}$ , i.e.,  $f : \mathcal{X} \rightarrow \mathcal{Y}$ . As the name suggests, the target  $Y$  supervises the fitting of a model  $\hat{f}$  from the model class (also known as learner or inducer)  $\mathcal{M}$ .

The training of  $\hat{f}$  can be formulated as a task to minimize the expected loss by the risk  $\mathcal{R} = \mathbb{E}[\mathcal{L}(\hat{f}(\mathbf{X}), Y)]$ . The loss function  $\mathcal{L}$  measures the discrepancy between the  $i$ -th realization  $y^{(i)}$  and its prediction by the model  $\hat{y}^{(i)} = \hat{f}(\mathbf{x}^{(i)})$ . The actual training process of the model  $\hat{f}$  is performed by minimizing the average loss across observations in a training data set  $\mathcal{D}_{train} \subset \mathcal{D}$ , i.e., minimizing  $\frac{1}{|\mathcal{D}_{train}|} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}_{train}} \mathcal{L}(\hat{f}(\mathbf{x}^{(i)}), y^{(i)})$ . Conversely, the model evaluation is conducted on a separate test data set  $\mathcal{D}_{test} = \mathcal{D} \setminus \mathcal{D}_{train}$ , such that the empirical risk  $\hat{\mathcal{R}}_{emp} = \frac{1}{|\mathcal{D}_{test}|} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}_{test}} \mathcal{L}(\hat{f}(\mathbf{x}^{(i)}), y^{(i)})$  is a suitable estimate for the generalization error.

$\mathcal{D}_{train}$  and  $\mathcal{D}_{test}$  are disjoint sets following the same distribution  $\mathbb{P}_{\mathbf{X}, Y}$ . To determine  $\mathcal{D}_{train}$  in practice, users can draw random samples from  $\mathcal{D}$  without replacement, and nowadays, more advanced procedures such as cross-validation are commonly used as a standard (Hastie et al., 2009). For further details on supervised machine learning, see Bishop and Nasrabadi (2006), Hastie et al. (2009) or Mohri et al. (2018).

Choosing a model class  $\mathcal{M}$  for fitting  $\hat{f} = \mathcal{M}(\mathcal{D}_{train})$  can be a complex challenge as many candidate models are available, and the optimum model choice will depend on the context of application (Ding et al., 2018). Popular supervised machine learning model classes are neural networks, tree-based approaches (e.g., decision trees, random forests, gradient-boosted models), or support vector machines; see Hastie et al. (2009) for an overview. Within a given model class  $\mathcal{M}$ , users can typically specify several hyperparameter configurations, which enlarges the number of model options further.

The so-called Rashomon effect additionally complicates the model choice, which describes the phenomenon that models with fundamentally different approaches may achieve similar performance on an empirical data set (Breiman, 2001b; Müller et al., 2023). Hence, fitting several models to an empirical data set may not clearly indicate which model to proceed with. Still, the model that minimizes the empirical risk – even by just a small fraction – might be selected. However, this ignores the degree to which a model is interpretable and intuitive for human understanding (Ding et al., 2018).

### 2.1.2. Interpretability

For human beings, it can be challenging to comprehend why and how supervised machine learning models reach their predictions. For this reason, machine learning algorithms are often referred to and treated as a black box. In an attempt to open up this black box, the field of IML has emerged (Du et al., 2019; Gilpin et al., 2018; Guidotti et al., 2018; Molnar, 2020; Murdoch et al., 2019).

However, interpretability is a “broad, poorly defined concept” (Murdoch et al., 2019). There is no consensus in the literature on a definition of interpretability, and the “notion of interpretability also depends on the target audience” (Ribeiro et al., 2016). This inconsistency transfers onto definitions of IML, for which various proposals exist (Miller, 2019; Mohseni et al., 2021). This thesis follows the definition of Murdoch et al. (2019), characterizing IML as “the extraction of relevant knowledge from a machine-learning model concerning relationships either contained in data or learned by the model.”

The specifics of the knowledge humans aim to extract from the model and procedures to deduct such explanations will depend on the application setting. Formulating a precise research question in advance narrows down the aspect of knowledge extraction that is of interest to the given application and is a vital prerequisite for choosing an interpretability method in application. This step proves a major conceptual challenge in IML (Lipton, 2017; Watson, 2022b), and we will see in Section 3.3 that, for example, knowledge extraction of relationships learned by the model in contrast to those contained in the data requires different methods. Further, diverse forms can present IML explanations, for example, visualizations, text descriptions, or mathematical equations, for which, again, an optimal choice will depend on the context and audience (Murdoch et al., 2019). Evidently, the workflow of applying IML techniques depends on various factors, which may be challenging to navigate for users (Vermeire et al., 2021). For this reason, recent research has engaged in developing tools like *eXplego* (Jullum et al., 2023) to guide practitioners through the process. For a broad overview of IML, I refer interested readers to Du et al. (2019), Guidotti et al. (2018) or Molnar (2020).

This thesis focuses on a central component of interpretability that analyzes the role of features in a supervised machine learning prediction task – feature attributions. Feature attributions may be categorized in feature effect and feature importance measurement (Molnar et al., 2022): feature effect measurement is concerned with measuring the effect size (magnitude) and sign (direction) of a feature on the value of the predicted outcome, e.g., as in local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016), Shapley additive explanations (SHAP) (Lundberg and Lee, 2017), partial dependence plots (PDP) (Friedman, 2001) or accumulated local effects (ALE) (Apley and Zhu, 2020). On the contrary, feature importance measurement gives insights into the impact of a feature (relevance) for the model (Molnar, 2020; Molnar et al., 2022; Murdoch et al., 2019). Feature importance typically relates to the importance of a feature for the predictive performance of the model (Breiman, 2001a; Covert et al., 2020; Fisher et al., 2019). However, it can also relate to the importance regarding the predicted value itself<sup>1</sup> or the model’s predictive uncertainty (Watson et al., 2023). While these the concepts of feature effects and feature importance appear similar at first sight, they, in fact, target different aspects. Feature effects focus on the influence of a feature (on the prediction), whereas feature importance focuses on the relevance of the feature. It is worth noting that it is possible to refine both concepts by

<sup>1</sup>For example, by taking the absolute values of feature effect attributions like SHAP values (Marcilio and Eler, 2020).

## 2.1 Interpretable Machine Learning

---

further analyzing which components of a feature’s effect (or importance) are due to interactions with other features (Herbinger et al., 2023; Tsang et al., 2020).

This thesis centers around feature importance measurement and this subsection introduces this concept in more detail. Several categorizing factors of interpretability methods (local versus global, intrinsic versus post hoc, model-specific versus model-agnostic explanations) streamline the remainder of this section.

### 2.1.3. Feature Importance Measurement

Feature importance in IML centers around the relevance of a feature in a supervised machine learning task (Molnar, 2020; Molnar et al., 2022; Murdoch et al., 2019). Intuitively speaking, this addresses the question of ‘how important is a feature for the prediction?’. This stream of research evaluates the magnitude to which a feature impacts quantities associated with the prediction task. As mentioned above, such quantities may relate to the predictive performance, the prediction target’s value, or the model’s predictive uncertainty. Commonly, and in this thesis (if not stated otherwise), feature importance is evaluated with respect to the predictive performance as characterized in terms of loss on a test data set.

The rationale behind analyzing this kind of feature importance is that changes in the model’s predictive performance will reflect the degree to which the information a feature provides is vital for the model in reaching apt predictions. If erasing a feature’s information from the model decreases predictive performance, this indicates an important feature (Fisher et al., 2019). Feature importance measurement is not limited to analyzing the importance of individual features but can also be extended to groups of features, see Au et al. (2022).

At first glance, the concept of feature importance might seem straightforward, but a statistical perspective reveals that it incorporates various facets. For example, researchers may be interested in determining whether a feature has *any* importance, i.e., desire to test the significance of nonzero feature importance scores. The application of statistical testing procedures can assist with that, and further, concepts from the field of feature selection may be related.<sup>2</sup> Moreover, from a statistical viewpoint, a more nuanced notion of feature importance differentiates whether a feature’s importance is irrespective of or conditional on other features in the model, as further discussed in Chapter 3. This aspect relates to whether a method should assess the importance of features for the machine learning model or the underlying data generating process (Williamson and Feng, 2020)<sup>3</sup> and the controversy of whether IML methods should be true to the model or true to the data (Chen et al., 2020).

The methodology for feature importance measurement is diverse, which mirrors the variety of aspects that an IML method may evaluate. The remainder of this section discusses how interpretability methods generally, and feature importance measures more specifically, may differ in

---

<sup>2</sup>Feature selection is a stream of research in statistics that aims to determine relevant features to include in a model, see further Hastie et al. (2017) and Miao and Niu (2016). However, the focus of feature selection is slightly different as the goal is to find a model and not to explain an existing model, which is the target of IML methods considered in this thesis.

<sup>3</sup>Explanations on the data generating process typically involve strong assumptions, such as assuming that a machine learning model aptly captures the structure of the data. Suitable IML methods that give explanations of such learned dependency structure may be helpful for some users. However, for learning causal structures in the data, specialized methodology from this field should be applied instead of IML methodology.

the level of explanation (local versus global), at which point of the machine learning workflow they are assessed (intrinsic versus post hoc) and to which model class they can be applied (model-specific versus model-agnostic). This section concludes by presenting several methods for post hoc, model-agnostic feature importance measurement on a primarily global level.

**Local versus Global Explanations** Interpretability methods can address distinct levels of explanations, i.e., seek different sorts of explanations concerning the granularity. Local methods yield explanations for individual instances, e.g., a row in a tabular data set, whereas global interpretability methods aim to generate insights into the expected model behavior overall (Adadi and Berrada, 2018; Molnar, 2020; Murdoch et al., 2019). Well-known examples of local interpretability methods are LIME (Ribeiro et al., 2016) and SHAP (Lundberg and Lee, 2017), whereas widespread global measures are, for example, permutation feature importance (PFI) (Breiman, 2001a) and Shapley additive global explanations (SAGE) (Covert et al., 2020).

The dissimilarity between local and global feature importance measures can be illustrated vividly through the example of credit scoring, which will reoccur in Part II of this thesis. The setting is a loan company that uses a credit scoring algorithm to determine whether an applicant should be granted a loan. If a customer asks for insights on which characteristics of hers were decisive in granting (or denying) her loan application, local methods will deliver valuable insights. However, suppose a regulatory authority wants to assess through an auditing procedure based on IML techniques whether the credit scoring algorithm violates ethical fairness standards (Alikhademi et al., 2021; Walzl and Vogl, 2018). In that case, global feature importance measures can help reveal the model’s systematic use of potentially discriminatory features.

**Intrinsic versus Post Hoc Explanations** The attempt to enhance the interpretability of predictive models, including measuring feature importance, can be undertaken at various stages of the machine learning workflow.

Intrinsic IML techniques – also called model-based approaches – enforce interpretability when fitting the machine learning model (Molnar, 2020; Murdoch et al., 2019). One approach to do so is choosing to use a so-called white box model instead of a black box model for the prediction task. White box models, such as a linear regression model, decision tree, or simple rule-based algorithms, can be interpreted directly by humans without the need of IML methods.

There is an ongoing debate in the literature on whether such white box models are preferable over black box models for tasks where interpretability is of interest. For example, Rudin (2019) questions that using black box models and explaining them afterward is adequate in high-stake decisions. Instead, Rudin (2019) argues that white box models can often achieve comparable predictive performance, making them suitable models while being transparent, and hence may be preferable. In a similar spirit, machine learning model choices may draw on Occam’s razor, favoring the simplest of similarly performing models (Bargagli Stoffi et al., 2022).

However, if a satisfactory predictive performance requires more flexible (hence, complex) models, intrinsic interpretability can be enhanced by setting complexity constraints. That is, a hyperparameter may regularize model complexity to balance off a model’s interpretability with flexibility (Ustun and Rudin, 2014). For example, practitioners could use a pre-specified tree-depth when fitting a decision-tree model or approaches like GA<sup>2</sup>M (Lou et al., 2013). However, a reasonable

## 2.1 Interpretable Machine Learning

---

cut-off point reflecting the complexity level to which a model is still interpretable will heavily depend on the context and audience.

Post hoc interpretability methods, on the contrary, aim for deriving explanations *after* fitting a model. In some use cases, questions on the interpretability of the model may only arise after a model was fitted and deployed, making post hoc methods a favorable choice for this. Such methods allow practitioners to obtain explanations on any supervised machine learning model deployed and typically work by analyzing the behavior of the model's predictions under data manipulations, e.g., as in PFI (Breiman, 2001a). While post hoc methods, in principle, give practitioners the freedom to use any model they find most suitable (irrespective of interpretability concerns), refined post hoc methods that focus on specific model classes can leverage unique model characteristics for deducing explanations.

**Model-specific versus Model-agnostic Explanations** As the name suggests, model-specific methods are specific to a particular machine learning model class. Such methods derive explanations from a model's specific traits and internal structures. On the one hand, this limits the method to the specific model class, but on the other hand, this allows for efficient deduction of tailor-made explanations. An example of this is the layer-wise relevance propagation method (Bach et al., 2015), which exploits the layer-based architecture of neural networks. Another example is TreeSHAP (Lundberg et al., 2018), which efficiently calculates SHAP values by actively taking advantage of the tree-based model structure.

Conversely, model-agnostic methods work with any predictive model. Model-agnosticism is feasible because such methods respect the black box character of the models in the sense that the IML method needs access to only the model's predictions. The model's predictive behavior is illuminated by querying the supervised machine learning model to be explained for various altered inputs, e.g., as in the conditional predictive impact (CPI) (Watson and Wright, 2021). Therefore, practitioners have unlimited flexibility in model choice, i.e., they can choose any model that fits the data best and effectively respects data-specific requirements. Further, with model-agnostic methods, explanations between different models can be compared to each other, which may be helpful for IML researchers and regulatory offices when assessing different models via the same interpretability method.

### 2.1.4. A Selection of Post Hoc Model-agnostic Interpretability Methods

This thesis aims to develop IML methods that give practitioners high flexibility in choosing appropriate models and, therefore, focuses on model-agnostic, post hoc interpretability methods. To introduce relevant background knowledge of several methods discussed in this thesis, this section presents an assortment of feature importance methods for post hoc, model-agnostic importance measurement focusing mainly on global-level explanations.

**Permutation Feature Importance** A straightforward approach to measuring the importance of a feature to the prediction is PFI, i.e., permutation feature importance (Breiman, 2001a; Fisher et al., 2019). For a given model, PFI assesses the importance of a feature by evaluating the change in loss  $\mathcal{L}$  when permuting the feature of interest in the data set. The idea is to quantify the change in model performance when removing the predictive information provided by the feature

of interest. With PFI, a random permutation of the feature’s values in the data set removes the predictive information. The random permutation wipes out the dependency structure between the feature and target<sup>4</sup> and hence, will reduce the model’s performance (increase in  $\mathcal{L}$ ) if the feature is important for reaching apt predictions of the target. Repeating this procedure for multiple rounds, PFI attributes the average change in  $\mathcal{L}$  as a feature importance score. PFI can be calculated for individual features or groups of features (Au et al., 2022) and yields a global measure of feature importance.

However, the random permutation of feature values breaks the dependency not only with the target but also with the other features in the data set. Since the dependency structure with other features is removed as well, the analysis of feature importance through PFI does not condition on or account for other features. This circumstance has consequences for interpreting the feature importance measure, which the research question may or may not intend (further discussed in Chapter 3). To account for the dependency structure between features, alternative procedures such as conditional feature importance (Strobl et al., 2008), the conditional subgroup approach (Molnar et al., 2023) and CPI, i.e., conditional predictive impact (Watson and Wright, 2021) have been proposed. Part I of this thesis explores advancements to the CPI, so as a representative of such approaches, the following paragraph discusses the CPI in more detail.

**Conditional Predictive Impact** Watson and Wright (2021) introduced CPI for testing conditional independence in supervised machine learning algorithms. However, in IML, CPI can measure feature importance while incorporating a valid statistical inference procedure to test for nonzero feature importance.

CPI is inspired by PFI, yet with a major modification: For the feature of interest, the change in loss  $\mathcal{L}$  is not evaluated after randomly permuting the feature (as in PFI), but when replacing it with a so-called model-X knockoff (Candès et al., 2018), which this thesis refers to as knockoff for short. Section 2.2.1 introduces the knockoff methodology more formally, yet for a high-level idea of the implications of measuring feature importance with CPI, it is sufficient to understand the following analogy: PFI evaluates the change in  $\mathcal{L}$  when replacing a feature with a permuted version of it, where the random permutation of feature values breaks the dependency between the feature with the target and all other features. On the contrary, CPI evaluates the change in  $\mathcal{L}$  when replacing the feature with a knockoff. The knockoff version of the feature breaks the dependency with only the target but maintains the dependency structure with the other features in the data set. This ensures that the resulting feature importance metric evaluates importance *given* the other features in the model, i.e., CPI results in a measure for conditional feature importance.<sup>5</sup> CPI can evaluate and – using paired t-tests – test conditional feature importance without having to refit the model, which is a major advantage over competing methods, as further discussed for the method presented in the following paragraph. However, the procedure hinges on generating valid and powerful knockoffs, which may be challenging; see further Section 2.2.1.

<sup>4</sup>Readers with a background in statistics may recognize that permutation-based statistical tests rely on the same rationale.

<sup>5</sup>Note that misconception might arise, so it is worth highlighting that including features in a model does not guarantee that quantities are evaluated conditional on the features. For example, PFI will evaluate feature importance irrespective of other features, even though the model, e.g., a random forest, may include all features simultaneously.



## 2.1 Interpretable Machine Learning

---

**Leave-one-covariate-out Importance** The leave-one-covariate-out (LOCO) procedure (Lei et al., 2018) assesses the importance of a feature by examining the consequences of leaving out the respective feature for the prediction task. LOCO as an importance measure follows the rationale that removing a feature from the prediction model will harm the predictive performance if a feature is important (Lei et al., 2018; Rinaldo et al., 2019). In contrast to PFI, LOCO removes the predictive information by dropping the feature altogether. Since machine learning models typically cannot handle missing features directly, LOCO relies on model refits.

In detail, the LOCO procedure works as follows: First, from model class  $\mathcal{M}$ , a supervised machine learning model is fitted, yielding the full model  $\hat{f} = \mathcal{M}(\mathcal{D}_{train})$ . Then again, a model from  $\mathcal{M}$  is fitted, but this time, excluding the feature of interest. In slight abuse of notation, we can formalize the observed feature matrix  $\mathbf{x} = (x_1, \dots, x_p)$  equivalently as  $\mathbf{x}_{\mathcal{P}}$ . Here,  $\mathcal{P} = \{1, \dots, p\}$  denotes the index set of the features in the data set, so when including only a subset of features  $\mathcal{S} \subseteq \mathcal{P}$  in the data set, we can write this as  $\mathbf{x}_{\mathcal{S}}$ . For excluding only the  $j^{th}$  feature (as of interest with LOCO), i.e.,  $\mathcal{S} = \mathcal{P} \setminus \{j\}$ , it is useful to introduce the set  $-j = \mathcal{P} \setminus \{j\}$  which indexes the set of all features except  $j$ . Following this notation, we have the reduced data set  $\mathcal{D}^* = \{(\mathbf{x}_{-j}^{(i)}, y^{(i)})\}_{i=1}^n$ , and analogously  $\mathcal{D}_{train}^*$ , which leads to the reduced model  $\hat{f}^* = \mathcal{M}(\mathcal{D}_{train}^*)$ . The LOCO importance for feature  $j$  is then the difference in predictive performance between the full and reduced model, estimated by the average change in  $\mathcal{L}$  across instances in  $\mathcal{D}_{test}$ , more formally,

$$LOCO_j = \frac{1}{|\mathcal{D}_{test}|} \sum_{(\mathbf{x}^{(i)}, y^{(i)}) \in \mathcal{D}_{test}} \mathcal{L}(\hat{f}^*(\mathbf{x}_{-j}^{(i)}), y^{(i)}) - \mathcal{L}(\hat{f}(\mathbf{x}^{(i)}), y^{(i)}).$$

The LOCO approach appears natural for measuring feature importance and allows for applying statistical inference procedures to test for nonzero importance scores (Lei et al., 2018; Rinaldo et al., 2019), yet the method has its drawbacks. Refitting the model can lead to substantial computational costs, especially when working with complex and tuning intense machine learning algorithms such as neural networks.

Further, the refitted models may learn different structures than the original model. For example, imagine two highly correlated features. LOCO leaves out one feature at a time, so the correlated feature remaining in the model can represent the predictive information of the other feature and capitalize on it. In this way, the refitted model can pick up similar information through a potentially very different model structure. This leads to the question of how comparable refitted models are.

In the case of two highly correlated features, leaving out one feature at a time will result in low importance scores attributed to these features because the predictive information of the left-out feature gets, in parts, represented by the correlated feature. This phenomenon, which occurs similarly with CPI, may or may not be intended by the research question and relates to the discussion on marginal versus conditional feature importance measurement; see further Chapter 3. Notably, leaving out both the correlated features simultaneously may drastically impact the predictive performance, and approaches that remove groups of features can help detect such dynamics (Au et al., 2022). Taking the thought of removing feature groups one step further, we can readily see that it might be of interest to instigate the removal of all possible groups of features, which is an idea Shapley values draw on.

**Shapley Values, SHAP and SAGE** Shapley values originate from game theory as a method to allocate credit among players in a cooperative game (Shapley, 1953). The concept can be adapted to feature attributions by treating the features  $\mathbf{X} = \{X_1, \dots, X_p\}$  as players and defining the game’s payout through a value function  $v$  that is related to the prediction task, such as the predicted value (Lundberg and Lee, 2017) or the loss (Covert et al., 2020).

Intuitively, Shapley values reflect the average change in payout when adding the feature of interest to a subset of features (coalition) that performs the prediction task. More precisely, the Shapley value  $\phi_j$  for a feature  $X_j$  is the weighted average of the change in  $v$  when adding  $X_j$  to all possible subsets of features that exclude it, indexed by  $S \subseteq \mathcal{P} \setminus \{j\}$ . We can write the formal definition of Shapley values as follows:

$$\phi_j = \sum_{S \subseteq \mathcal{P} \setminus \{j\}} \frac{|\mathcal{S}|!(|\mathcal{P}| - |\mathcal{S}| - 1)!}{|\mathcal{P}|!} (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})). \quad (2.1)$$

Shapley values have received considerable attention in IML, which in parts stems from the mathematically sound foundation of Shapley values through axioms. Shapley values are the sole quantity meeting a selection of desirable axioms (Covert et al., 2020; Shapley, 1953; Štrumbelj and Kononenko, 2014; Watson, 2022a):

**Symmetry axiom.** If feature  $j$  and feature  $k$  contribute equally to the payout of all coalitions  $\mathcal{S}$  that exclude  $j$  and  $k$ , i.e., we have  $v(\mathcal{S} \cup j) = v(\mathcal{S} \cup k) \quad \forall \mathcal{S} \subseteq \mathcal{P} \setminus \{j, k\}$ , the attributed Shapley values are equal, i.e.,  $\phi_j = \phi_k$ .

**Efficiency axiom.** The Shapley values of all features  $j \in \mathcal{P}$  add up to the difference in payout of the full coalition  $\mathcal{P}$  and empty coalition  $\emptyset$ , i.e., we have  $v(\mathcal{P}) - v(\emptyset) = \sum_{j \in \mathcal{P}} \phi_j$ .

**Dummy axiom (null player axiom).** If feature  $j$  does not affect the payout in any coalition, that is, if  $v(\mathcal{S} \cup j) = v(\mathcal{S}), \forall \mathcal{S} \subseteq \mathcal{P} \setminus \{j\}$ , then the feature’s Shapley value will equal zero, i.e.,  $\phi_j = 0$ .

**Additivity axiom (linearity axiom).** If the Shapley value of feature  $j$  is evaluated in different games, i.e., concerning different value functions, say,  $\phi_j(v)$  and  $\phi_j(\omega)$  where  $v$  and  $\omega$  is the value function of the respective game, then those Shapley values add up to the same quantity as if the value functions would have been combined, i.e.,  $\phi_j(v) + \phi_j(\omega) = \phi_j(v + \omega)$ .

In practice, there are several challenges arising when calculating Shapley values. In the context of IML, assessing the value function  $v$  on strict subsets  $\mathcal{S} \subset \mathcal{P}$  is not straightforward. That is because supervised machine learning models typically require the same input dimension for reaching predictions as encountered during model training. In other words, a model trained on  $p$  features (that cannot handle missing features) demands  $p$ -dimensional input to yield predictions. The above subsection on LOCO briefly mentioned that model refitting could be an option to evaluate subsets of the feature space; however, this would incorporate immense computational costs and spark a similar debate to that of LOCO on the comparability of refitted models.

An established solution to the problem of machine learning model  $\hat{f}$  requiring  $p$ -dimensional input is to define  $v$  as the expectation of the model for a given coalition, i.e.,  $\mathbb{E}_{\mathbb{R}}[\hat{f}(\mathbf{X} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})]$  (Chen et al., 2023). In application, the out-of-coalition features (feature set  $\mathcal{P} \setminus \mathcal{S}$ ) are imputed with values from sampled the reference distribution  $\mathbb{R}$ , which may be the marginal feature distribution (Lundberg and Lee, 2017), conditional feature distribution (Aas et al., 2021; Olsen et al., 2023)

## 2.1 Interpretable Machine Learning

---

or other distributions (Blesch, Wright, and Watson, 2023; Watson et al., 2023). Part II discusses considerations for choosing an adequate  $\mathbb{R}$  further.

Another obstacle to applying Shapley values for IML is assessing  $v$  for all possible coalitions, which are resembled by the power set of  $\mathcal{P}$ . The calculation of Shapley values hence implies  $2^p$  evaluations of  $v$  and therefore is, even without having to refit the model, computationally expensive in medium or a large of feature spaces. A more suitable concept for applying Shapley values to machine learning is SHAP, i.e., Shapley additive explanations Lundberg and Lee (2017).

SHAP values unify several frameworks related to Shapley values to fit the needs of IML more closely. That is, SHAP values reformulate desirable Shapley-inspired properties for their relevance in IML applications (local accuracy, missingness, and consistency) (Lundberg and Lee, 2017) and allow for computational shortcuts. One can think of SHAP as using local surrogate models with an additive structure to decompose the individual predictions (if the value function  $v$  is defined using the prediction value itself). As a consequence, SHAP values are a local interpretability method. SHAP values may still be expensive to compute; however, several approximation techniques such as KernelSHAP (Lundberg and Lee, 2017), TreeSHAP (Lundberg et al., 2018) or FastSHAP (Jethani et al., 2022) are available.

SHAP values measure feature attributions on a local level of explanation, but extensions to generate global explanations, such as SAGE, i.e., Shapley additive global explanations (Covert et al., 2020), exist. SAGE values yield feature importance scores on a model-wide level which can be approximated by sampling and aggregating local explanations efficiently (Covert et al., 2020).

SAGE follows the notion of feature importance that takes changes in the predictive performance of a model as an indication of relevance. Therefore, SAGE values conceptualize the value function  $v$  through predictive performance, similarly to LossSHAP (Lundberg et al., 2020). In fact, SAGE values are the expectation of LossSHAP values across the data set (Covert et al., 2020).

Explanations based on Shapley values, including SHAP and SAGE values, have received considerable attention in IML and various extensions exist (Sundararajan and Najmi, 2019). Navigating through the rapidly changing, growing literature on Shapley values in IML can be challenging. As of now, I refer interested readers for further reading on the topic and for an overview on how to estimate Shapley value feature attributions to Chen et al. (2023).

The above section summarizes the IML background related to this thesis and presents selected methods for feature importance measurement. From this, it is apparent that most of the methods presented rely (as a subroutine) on sampling feature values. Data sampling can be conducted not only through procedures like feature permutations but also by more advanced techniques from the field of generative modeling. The following section introduces this field of research, concentrating on considerations that are particularly useful in the context of IML. That is, the IML methods discussed in this thesis are conceptualized primarily for tabular data. Related data sampling subroutines hence ought to sample feature values for data tables, which may consist of mixed data types. Further, imposing distinctive statistical properties on the generated data as with knockoffs can be helpful for IML applications. As the generation of newly sampled feature values is only a subroutine in IML methods, this application can benefit particularly from fast and straightforward generative modeling approaches, which leads directly to this thesis's section on the gap in research and contribution.

## 2.2. Generative Modeling

Generative modeling is a subfield of machine learning that centers around generating synthetic data. The goal is to develop models capable of synthesizing data samples  $\mathbf{x}' = \{(x'_1, \dots, x'_p)\}_{i=1}^n$ , which appear similar to some given data  $\mathbf{x} = \{(x_1, \dots, x_p)\}_{i=1}^n$ . More generally, generative models aim to synthesize values of  $\mathbf{X}'$  that follow the same probability distribution as  $\mathbf{X}$ , i.e., aim for  $\mathbf{X}' \stackrel{d}{=} \mathbf{X}$ .

The type of data administered by generative models varies widely and may include image, audio, text or tabular data. Specialized methods for selected data types, for example, large language models for generating text data, have been proposed, yielding tools like ChatGPT (OpenAI, 2023) that allow even laypeople to synthesize convincing text data. Similarly, DALL-E (Ramesh et al., 2022) was introduced as a hands-on tool to generate image data. This thesis, however, focuses on tabular data; hence, this section introduces generative modeling in the context of tabular data.

It is possible to frame generative modeling as a stream of research in unsupervised machine learning, which opposes supervised learning as introduced in Section 2.1.1. Recap that in supervised machine learning, a target  $Y$  guides (supervises) the training of model  $\hat{f} : \mathbf{X} \rightarrow Y$ . In unsupervised machine learning, on the contrary, a target  $Y$  is either non-existent or treated as any other feature in  $\mathbf{X}$ , and the aim is to learn relationships within  $\mathbf{X}$ . Many methods in unsupervised machine learning yield results that focus on detecting patterns in  $\mathbf{X}$  itself, e.g., unsupervised k-means clustering (Sinaga and Yang, 2020). In contrast, generative modeling analyzes patterns in  $\mathbf{X}$  as an intermediate stage for synthesizing data similar to  $\mathbf{X}$ . Even though terminology is inconsistent in the literature (Bishop and Nasrabadi, 2006; Ng and Jordan, 2001), generative modeling may be categorized as a branch of unsupervised learning because generative models synthesize data based on information deduced from only  $\mathbf{X}$ .

Alternative conceptualizations of generative modeling set a more pronounced emphasis on the analyzed probability distribution. Through this lens, generative modeling concerns a different probability distribution than discriminative modeling (Ng and Jordan, 2001): in discriminative modeling, the focus is on the conditional probability distribution  $\mathbb{P}_{Y|\mathbf{X}}$ , whereas in generative modeling, attention directs to the joint probability distribution  $\mathbb{P}_{\mathbf{X},Y}$ , or, if the target  $Y$  is considered a feature,  $\mathbb{P}_{\mathbf{X}}$ , respectively.

From this perspective, it is straightforward to relate generative modeling to density estimation. In density estimation, the aim is to estimate an explicit form of  $\mathbb{P}_{\mathbf{X}}$ . However, density estimation techniques do not automatically provide sampling procedures to generate data from the estimated density, so they hardly constitute a generative model directly. Conversely, generative models aim to synthesize data that follows  $\mathbb{P}_{\mathbf{X}}$  closely but only occasionally estimate  $\mathbb{P}_{\mathbf{X}}$  in an explicit form. Generative models that attempt to learn and give access to  $\mathbb{P}_{\mathbf{X}}$  directly are referred to as *explicit* generative models, whereas *implicit* generative models rely on mappings of a random noise vector  $z$  to the data space of  $\mathbf{X}$  in order to generate synthetic data (Harshvardhan et al., 2020).

Amongst the most widely used methods for generative modeling are variational autoencoders (VAE) (Kingma and Welling, 2014), generative adversarial networks (GAN) (Goodfellow et al., 2014), normalizing flows (NF) (Rezende and Mohamed, 2015), diffusion probabilistic models (DPMs) (Ho et al., 2020) and transformer-based models (Vaswani et al., 2017). An in-depth analysis of these methods is beyond the scope of this section; hence, interested readers are advised to consider Bond-Taylor et al. (2021) and Foster (2022) for further details. Nonetheless, it is

## 2.2 Generative Modeling

---

helpful to delve into GANs as an illustrative example because this procedure will serve as a source of inspiration for the methodology proposed in Part III.

A GAN is an implicit generative model that builds on two neural networks (generator  $\hat{f}_{NN}^G$ , discriminator  $\hat{f}_{NN}^D$ ) and relies on a game-like rationale for model training (Goodfellow et al., 2014): Generator  $\hat{f}_{NN}^G$  takes random noise  $\mathbf{z}$  as input and returns some generated data samples  $\mathbf{x}'$ . Discriminator  $\hat{f}_{NN}^D$  is faced with samples from both real data  $\mathbf{x}$  and generated data  $\mathbf{x}'$  (labeled accordingly), and asked to classify whether a data sample originates from  $\mathbf{x}$  or  $\mathbf{x}'$ . The two neural networks engage in a so-called adversarial training procedure, where  $\hat{f}_{NN}^G$  tries to generate increasingly realistic data samples, and  $\hat{f}_{NN}^D$  aims for improving in the classification task to distinguish  $\mathbf{x}$  from  $\mathbf{x}'$ . Each round updates the network's parameters through backpropagation (Rumelhart et al., 1986), and the procedure continues until  $\hat{f}_{NN}^D$  can no longer distinguish real from generated data samples, i.e., the accuracy of  $\hat{f}_{NN}^D$  is  $\leq 0.5$ . Finally, generator  $\hat{f}_{NN}^G$  with parameters from the last iteration step resembles the generative model.

GANs can yield compelling synthesized data samples, which makes them a popular and widespread candidate model for generative modeling, particularly for image data (Harshvardhan et al., 2020). However, there are also disadvantages attached to the method. In some cases, GANs may fail to capture the diversity of the original data (mode collapse, see further Thanh-Tung and Tran (2020)). Aside from the quality of the synthesized data, it is worth noting that GANs typically involve high efforts in model training. The neural networks require large amounts of data and incorporate the tuning of many parameters and hyperparameters before finding a stable result (Nash equilibrium in the minimax game) for the generator model – if converging at all (Alqahtani et al., 2021). This instability translates to demanding considerable time, expertise, and computational resources to fit GANs (Alqahtani et al., 2021; Harshvardhan et al., 2020).

For tabular data, adaptations such as the conditional tabular GAN (Xu et al., 2019) were introduced, but as with most generative models, the architecture is based on deep learning. While deep learning algorithms frequently outperform other algorithms on image or text data, tree-based methods are strong competitors with tabular data and require remarkably less computational costs and tuning efforts (Borisov et al., 2022; Grinsztajn et al., 2022). That said, using tree-based algorithms for generative modeling with tabular data might be advantageous, and attempts to do so have been proposed recently, e.g., by Correia et al. (2020) and Nock and Guillaume-Bert (2023). In that vein, Paper 3 proposes an explicit generative model based on random forests,<sup>6</sup> further discussed in Section 3.3.

Evaluating the suitability of a generative model is a challenging task, and a gold standard to do so has yet to emerge from the literature. Considerations may take into account factors like computational costs and tuning efforts, yet the aim for 'high-quality' synthetic data may often be a priority. While in supervised machine learning, the true value of  $y^{(i)}$  can be compared to  $\hat{y}^{(i)}$ , there is no ground truth for synthetic data. Therefore, defining evaluation criteria and characteristics of 'high-quality' synthetic tabular data is not straightforward.

Desiderata for synthetic data samples could be that the data synthesized should lie within the support of the original data distribution (high precision) but also reflect the diversity of the data (high recall), which can be inspected by precision-and-recall metrics (Alaa et al., 2022).

---

<sup>6</sup>For an introduction to random forests, see Hastie et al. (2009).

Another approach to evaluating generated data is assessing the performance of a binary classifier in distinguishing real from synthetic data (classifier 2-sample test, see Lopez-Paz and Oquab (2017)). Further, the performance of supervised learning algorithms trained on either original or generated data samples can be compared (machine learning efficacy, see Choi et al. (2017), Vardhan and Kok (2020) and Xu et al. (2019)). The idea behind this approach is that if a supervised learning algorithm achieves similar predictive performance in both cases, the synthetic data reflects the essence of predictive information provided by the real data well. Further, evaluations that rely on visual inspections, such as analyzing similarities in cumulative distribution functions (Chen et al., 2019) or scatter plots (Choi et al., 2017) may enrich evaluation procedures in generative modeling. In brief, the literature proposes several approaches to evaluate the quality of synthetic data created by generative models, but a recognized standard is yet to be established.

High-quality synthetic data can aid in various challenges across academic fields. For example, a use case for synthetic data is data enrichment. There, synthesizing  $\mathbf{X}' \stackrel{d}{\sim} \mathbb{P}_{\mathbf{X}'} \approx \mathbb{P}_{\mathbf{X}}$  can enrich training and test data sets with arbitrary amounts of instances from the same distribution as the empirical data. Data enrichment generally enlarges a data set, which can help to eliminate class imbalances, an obstacle for fitting machine learning models (Ali et al., 2013; Cartella et al., 2021; Engelmann and Lessmann, 2021). Another use case is the respect for data protection rules. Legislation may impose strict regulations on using data sets containing private information on individuals, e.g., the Regulation (EU) 2016/679 (General Data Protection Regulation; GDPR). However, researchers might still want to get insights into relationships on a population level using such data. Specialized generative models that preserve privacy traits can generate realistic data that reflects the data distribution but does not violate the privacy rights of individuals (Choi et al., 2017; Jordon et al., 2018; Vardhan and Kok, 2020).<sup>7</sup> Another promising example of a synthetic data use case is data imputation (Abroshan et al., 2023; Camino et al., 2019). Data imputation aims to fill missing data values, which often relates to few missing values in observed data, but it may be also be a critical subroutine in IML methods, replacing several feature values at once.

As described in Section 2.1.4, several feature importance methods aim to imitate the absence of features by imputing them with other values, e.g., permuted values, as in PFI. Generative modeling opens up the possibility for advanced imputation strategies, yet takes on a very general perspective on sampling new data points. For applications in IML, requiring additional statistical properties on the generated data can be advantageous. A specialized set of generated data, knockoffs, that do so are introduced in the following subsection. In Parts I and II of this thesis, we will see the benefits of using such synthesized data points as a subroutine.

### 2.2.1. Model-X Knockoffs

Candès et al. (2018) introduce model-X knockoffs, or knockoffs for short, in the context of feature selection while controlling the false discovery rate, with knockoffs constituting a set of synthetic features that mimic the statistical characteristics of the original data. In addition to that, knockoffs come with advantageous traits. In that light, we can portray the synthesis of knockoffs as a special case of generative modeling, for which the generated features satisfy certain properties. Knockoffs

<sup>7</sup>Generative models do not automatically have privacy guarantees; such models are rather a further refinement of generative models, for example, PATE-GAN (Jordon et al., 2018) which satisfies privacy guarantees or medGAN (Choi et al., 2017), which the authors claim is associated with reduced privacy risks.

## 2.2 Generative Modeling

---

mirror the structure of the features in the given (original) data set but are known to not incorporate any additional systematic information about the target.<sup>8</sup>

More formally, the definition of knockoffs given by Candès et al. (2018) states that for a set of random features  $\mathbf{X} = \{X_1, \dots, X_p\}$  (original features) a new set of random features  $\tilde{\mathbf{X}} = \{\tilde{X}_1, \dots, \tilde{X}_p\}$  (knockoff features) is generated that satisfies

1. For any subset  $\mathcal{S} \subset \{1, \dots, p\}$ :

$$(\mathbf{X}, \tilde{\mathbf{X}})_{\text{swap}(\mathcal{S})} \stackrel{d}{=} (\mathbf{X}, \tilde{\mathbf{X}}),$$

where  $\stackrel{d}{=}$  represents equality in distribution and  $\text{swap}(\mathcal{S})$  indicates swapping the features in  $\mathcal{S}$  with their knockoff counterparts.

2. Conditional independence:

$$\tilde{\mathbf{X}} \perp\!\!\!\perp Y \mid \mathbf{X}.$$

Property (1) entails that the features  $\mathbf{X}_{j \in \mathcal{S}}$  and their knockoff counterparts  $\tilde{\mathbf{X}}_{j \in \mathcal{S}}$  can be exchanged without affecting the joint distribution of features and knockoffs, i.e., the joint distribution of  $(\mathbf{X}, \tilde{\mathbf{X}})$  is invariant under swaps of  $\mathcal{S}$ . Drawing of an example given by Candès et al. (2018), this implies, for example, with  $p = 3$  and  $\mathcal{S} = \{2, 3\}$  that

$$(X_1, \tilde{X}_2, \tilde{X}_3, \tilde{X}_1, X_2, X_3) \stackrel{d}{=} (X_1, X_2, X_3, \tilde{X}_1, \tilde{X}_2, \tilde{X}_3).$$

Property (2) reflects that the generated knockoffs do not contain further information on the target  $Y$  given the original features  $\mathbf{X}$ .

These properties make knockoffs advantageous in subroutines across a range of tasks. Property (1) allows to swap features with their knockoff counterparts without affecting the joint distribution, and property (2) ensures that the knockoffs  $\tilde{\mathbf{X}}$  carry no systematic information for  $Y$  given  $\mathbf{X}$ . It is straightforward to see that these properties allow methods to remove the predictive information on  $Y$  without affecting the dependency structure within the feature matrix when exchanging features  $\mathbf{X}_{j \in \mathcal{S}}$  with their knockoff counterparts  $\tilde{\mathbf{X}}_{j \in \mathcal{S}}$ . Therefore, a comparison of suitable metrics calculated with respect to either original or knockoff features can reveal whether the features analyzed are important for a prediction task (Blesch, Watson, and Wright, 2023; Watson and Wright, 2021) or should be selected in a model (Candès et al., 2018).

The generation of knockoff features is closely related to generative modeling since the aim is to generate features similar to an original data set. However, the two properties mentioned above must be satisfied for a generative model to qualify as a knockoff sampler.

Property (2) can be easily satisfied by any generative model because it only requires the generated features  $\tilde{\mathbf{X}}$  to not provide further information to the target  $Y$  given the original features  $\mathbf{X}$ . Following Candès et al. (2018),  $\tilde{\mathbf{X}} \perp\!\!\!\perp Y \mid \mathbf{X}$  is guaranteed if “ $\tilde{\mathbf{X}}$  is constructed without looking at  $Y$ .” Hence, property (2) is respected as long as the generative model is trained with only  $\mathbf{X}$  (not including  $Y$ ).

On the contrary, meeting property (1) is more demanding. For second-order knockoffs, which meet equality in the first two moments of the distribution, this can be satisfied, for example, by

---

<sup>8</sup>Note that in the case of knockoffs, a target  $Y$  is – if it exists – treated differently as the features in  $\mathbf{X}$ , which may differ from generative modeling in general.

exploiting characteristics of the Gaussian distribution as in Gaussian knockoffs (Candès et al., 2018). Alternatively, this property can be enforced within the data generator, such as in DeepKnockoffs (Romano et al., 2020). There, the minimized loss function incorporates a penalty term reflecting the maximum mean distribution discrepancy when swapping features.

More generally speaking, the hindrance for generative models to qualify as knockoff samplers is that property (1) requires equality in the distribution of the joint matrix  $(\mathbf{X}, \tilde{\mathbf{X}})$ . Generative models learn an apt representation of  $\mathbb{P}_{\mathbf{X}}$ , though this does not yield guarantees for the original data in conjunction with the synthesized data, i.e.,  $\mathbb{P}_{\mathbf{X}, \tilde{\mathbf{X}}}$ .<sup>9</sup>

Knockoffs are an active field of research, and many methods to generate knockoffs have been proposed. Methods based on generative machine learning include, for example, DeepKnockoffs (Romano et al., 2020), deep direct likelihood knockoffs (Sudarshan et al., 2020) or KnockoffGAN (Jordon et al., 2019). Alternative knockoff sampling procedures work by imposing assumptions on the data through predefined parametric distributions (e.g., Gaussian Knockoffs (Candès et al., 2018)) or stochastic processes (e.g., hidden Markov model knockoffs (Sesia et al., 2018)). More general sampling techniques have been proposed, e.g., metropolized knockoff sampling (Bates et al., 2021) and also Bayesian approaches (Martens et al., 2021). Notably, the research on knockoff generation has also produced specialized procedures for distinct data types, e.g., sequential knockoffs (Kormaksson et al., 2021) for mixed tabular data, which Parts I and II draw on.

From the literature on knockoff generation, it may not be readily apparent to the user how to choose a knockoff sampling algorithm. If a task requires the generation of knockoffs, several procedures may appear reasonable. The data type may give a clue, for example, if an assumption of Gaussianity on the data seems plausible. However, evaluating what makes a knockoff a ‘good’ knockoff is challenging and requires considering other aspects than just the data type.

Imagine a knockoff generation algorithm returning an exact copy of the original features as knockoffs. Such knockoffs fulfill properties (1) and (2), yet clearly, doing so does not help derive valuable insights (Candès et al., 2018). To encourage the generation of powerful knockoffs, a popular procedure is to select knockoffs that minimize the correlation between original features and knockoff counterparts, i.e.,  $\text{corr}(X_j, \tilde{X}_j)$ , which may be encouraged during knockoff sampling (Romano et al., 2020). However, Spector and Janson (2022) argue that knockoffs minimizing reconstructability (which aims to hinder a model faced with  $\mathbf{X}_{-j}$  and  $\tilde{\mathbf{X}}_j$  to reconstruct the information provided by  $X_j$ ) might be more favorable in terms of their ability to generate powerful results over those that minimize  $\text{corr}(X_j, \tilde{X}_j)$ .

Further, knockoffs should strive for robustness. That is, running knockoff-based procedures multiple times, such as feature importance measurement with CPI, should yield consistent outcomes. However, due to the randomness in the sampling of knockoffs, diverging results may occur (Candès et al., 2018). Attempting to overcome this, researchers could generate multiple knockoffs and average results or aggregate multiple knockoffs prior to running the procedure (Gimenez and Zou, 2019; Zhimei Ren and Candès, 2023).

<sup>9</sup>Note that, depending on the application, weaker requirements than property (1) may be sufficient. For example, Part II proposes calculating SHAP values using knockoffs. For this, only the out-of-coalition features in set  $-\mathcal{S} = \mathcal{P} \setminus \mathcal{S}$  are swapped in, which requires that  $\tilde{\mathbf{X}}_{-\mathcal{S}} | \mathbf{X}_{\mathcal{S}} \stackrel{d}{=} \mathbf{X}_{-\mathcal{S}} | \mathbf{X}_{\mathcal{S}}$ , for any subset  $\mathcal{S}$ , which is a weaker condition. Still, knockoffs can be useful in this example because property (1) leverages other favorable traits, such as the opportunity to generate knockoffs upfront and then swap them in where needed for SHAP calculation.



## 2.2 Generative Modeling

---

In sum, the knockoff methodology sets the groundwork to generate synthetic features that obey desirable properties to serve as flexible, modular components in various tasks, such as in deducting IML explanations.



## 3. Gap in Research and Contribution

This chapter details the gap in research and how the three parts of this thesis contribute individually and in conjunction to address it. First, this chapter describes shortcomings in the literature on IML and generative modeling concerning mixed tabular data that exhibits dependency structures. In response, a brief overview of how the contributing parts of this thesis address the gap in research is presented before summarizing the individual contributions of each part in more depth. Subsequently, the connections between the three parts and their combined efforts in effectively conquering the research gap are examined. Following this chapter, Parts I, II and III present the contributing papers of this thesis in full length to provide in-depth insights into the developed methodology.

### 3.1. Gap in Research

A statistical perspective uncovers the need to refine concepts in IML to respect and embrace the unique characteristics of mixed tabular data exhibiting dependency structures. Previous literature insufficiently addresses the challenges real-world data – which often is of this type – poses on machine learning methodology. Further, opportunities to develop beneficial methods for IML and generative modeling that actively take advantage of such data characteristics are yet to be explored. Additionally, the supply of concrete methods and software suitable for practitioners’ needs is critical. A misalignment in methodological concepts and empirical requirements hinders adequate methodology transfer to empirical applications. Given that interpretability strives for meaningful and trustworthy explanations in real-world settings, there is a compulsion to align methodology with real-data needs, however, this is not the case in several respects.

A misalignment readily apparent to practitioners is the mixed data type. Methodological advances are typically derived for either an all-continuous or all-categorical feature space (Aas et al., 2021; Chen et al., 2020; Romano et al., 2020; Sesia et al., 2018). Extensions to the mixed data case are (if acknowledged at all) typically evaluated by follow-up research, such as Luo et al. (2021) or Redelmeier et al. (2020). Methods may claim to generalize to the mixed data case, but concrete methods or software is often missing (Romano et al., 2020; Watson and Wright, 2021). Further, mixed data is barely incorporated when simulating data (Olsen et al., 2023; Watson and Wright, 2021; Williamson and Feng, 2020; Zhang et al., 2019). In real-data benchmark data sets, categorical features may even be removed from benchmark data sets to eliminate the mixed data characteristic of the data (Aas et al., 2021; Molnar et al., 2023). Alternatively, workarounds might be applied to make the methods technically work with mixed data. For example, this includes procedures like dummy encoding categoricals and treating them as continuous, or integer encoding. If methods require an integer encoding of features, then practitioners are forced to find orderings in categorical features (Apley and Zhu, 2020). However, this violates the actual distribution of the data (categoricals are evidently not continuous and may not have an ordering) and hence is

problematic. Implications of mixed data (or workarounds to circumvent it) on a method’s performance remain largely unidentified, and practitioners faced with such data sets may be unable to apply a desired method as intended.

A more subtle misalignment between methodological developments and real-data requirements is the acknowledgment of dependency structures. A coherent understanding of dependency structures’ implications and tools to account for these is fundamental. In statistics, a distinction between marginal as opposed to conditional conceptualizations is well-established, yet methodological development in IML is not up to par. This deficiency can lead to (unintentionally) misguided explanations (Watson, 2022b), which flaws the accurate insights IML strives for and even enables adversarial attackers to manipulate explanations (Slack et al., 2020). Practitioners and IML researchers alike may lack awareness regarding the consequences that dependency structures present in empirical data sets may have. Even in the presence of awareness, users might not have the necessary methods and tools available. In particular, IML methods typically rely on sampling feature values as a subroutine, for which respecting the dependency structure can be pivotal. However, access to and incorporation of user-friendly approaches that can generate mixed tabular data with a dependency structure into IML methods may be challenging. A refined methodology that is both flexible in appropriately respecting dependency structures and ready-made for application is essential to ensure contemplated IML insights and increase robustness against adversarial attacks.

Refining existing methodology represents only one of many approaches to addressing such concerns. From another viewpoint, it sets the groundwork for developing new, advantageous methods. Previous literature has barely recognized opportunities to exploit traits of mixed tabular data with a dependency structure to leverage more straightforward and efficient algorithms. Instead of proposing methods adjusted from other data types, investigating tailor-made algorithms for this kind of data can enhance applicability and user-friendliness for real-data settings.

For sampling feature values in IML procedures and generative modeling more generally, there is considerable room for improvement concerning mixed tabular data. The methodology in generative modeling is developed primarily for other data types. For example, GANs focus mainly on image data (Goodfellow et al., 2014; Harshvardhan et al., 2020), and have been adapted later on to tabular data with only minor adaptations (Park et al., 2018; Xu et al., 2019). The traits of mixed tabular in specific, i.e., the mix of continuous and categorical features “poses a significant challenge” (Borisov et al., 2022) for generative modeling. However, presuming that a method adequate for other data types, such as image data, will also be advantageous for mixed tabular data is a shortsighted perception as this neglects opportunities to take advantage of potentially much simpler methodology that fits the needs of mixed tabular data more closely. The state-of-the-art models in generative modeling mainly draw on deep learning, yet tabular data might not require such tuning-intense and computationally expensive architectures. Instead, lightweight tree-based algorithms, which are established in the field of supervised machine learning for mixed tabular data (Borisov et al., 2022; Grinsztajn et al., 2022; Shwartz-Ziv and Armon, 2022), are largely unrecognized in generative modeling but could offer the potential for improvement. In addition to advancing generative models themselves, tailoring the insertion of generative subroutines in IML procedures – for example, through knockoffs that impose further statistical properties on the generated data – requires improved alignment with mixed tabular data.

Summarizing the shortcomings in current literature with more precision, vital aspects that demand attention and urge methodological advancement include the following:

## 3.2 Contribution Overview

---

1. Conditional versus marginal feature importance measurement is not routinely distinguished in IML, yet neglecting this differentiation yields explanations prone to misinterpretation.
2. Methodology to test for global, conditional feature importance with a post hoc, model-agnostic method that does not require model refits has limited applicability with mixed data.
3. Marginal imputation techniques facilitate adversarial attacks on interpretability methods, which persists not only for feature importance measurement on a local level, such as with SHAP values (Slack et al., 2020), but also extends to global level explanations, such as SAGE values. IML methodology urges practicable advancements to increase robustness.
4. Generative modeling for tabular data relies on methods adapted from other data types, primarily deep-learning-based architectures. Such algorithms typically require intense tuning and computational efforts, which may be unnecessarily complicated for generating mixed tabular data.
5. Practitioners need access to advantageous methods with mixed tabular data through user-friendly, well-documented software implementations in the programming language most widely used by the target audience.

## 3.2. Contribution Overview

This thesis presents four papers, organized in three parts, that contribute toward addressing the shortcomings in current literature as detailed above.

- Part I discusses conditional (as opposed to marginal) feature importance and details a specialized method for measuring and testing conditional feature importance with mixed tabular data to conquer issues 1 and 2.
- Part II highlights that adversarial attack vulnerability persists for both SHAP and SAGE values, and investigates the determinants of this vulnerability. It addresses issue 3 by proposing knockoff imputation for Shapley values to increase robustness.
- Part III introduces a generative model based on random forests that is particularly suitable for mixed tabular data and further provides a software implementation in the *Python* programming language to address issues 4 and 5.

## 3.3. Individual Contributions: Parts I, II and III

**Part I** presents the paper “Conditional feature importance for mixed data” (Blesch, Watson, and Wright, 2023) that explores conditional feature importance measures and proposes a tailor-made procedure for mixed tabular data. Browsing IML methodology for conditional feature importance measurement, Paper 1 reveals that only few methods exist, with limited applicability to mixed data. As a result, Paper 1 proposes a specialized method for testing conditional feature importance measurement with mixed tabular data and demonstrates its effectiveness over competing methods.

More specifically, Paper 1 conquers the misalignment in acknowledging dependency structures by highlighting the connection between the statistical concept of independence testing and feature importance measurement. Features are (statistically) independent if the value of a feature does not affect the probability of realizing the other feature’s value, which independence tests can assess. Analogously, feature importance measurement in supervised machine learning tasks evaluates whether a feature  $X_j$  is relevant (informative) in predicting target  $Y$ . Commonly, feature importance is mainly concerned with the *extent* to which a feature is important. However, from a statistical perspective, testing whether low feature importance scores differ significantly from zero may be a valuable insight for practitioners.

In statistical independence testing, a common distinction is between marginal and conditional independence. The key difference between marginal and conditional testing is whether the evaluation is irrespective of, or accounts for, other features. More formally, a marginal independence test evaluates the null hypothesis  $H_0^M$ , where

$$H_0^M : X_j \perp\!\!\!\perp \{Y, \mathbf{X}_{-j}\}. \quad (3.1)$$

On the contrary, a conditional independence test evaluates the null hypothesis  $H_0^C$ , where

$$H_0^C : X_j \perp\!\!\!\perp Y \mid \mathbf{X}_{-j}. \quad (3.2)$$

In feature importance measurement, however, such a conceptualization of marginal in contrast to conditional evaluation is barely acknowledged even though it has decisive implications for interpretation (Watson and Wright, 2021; Watson, 2022b). Adapting this notion to feature importance measurement, a marginal evaluation corresponds to assessing the importance of a feature for the model’s predictions regardless of the relevance of other features included in the model (marginal feature importance, e.g., PFI (Breiman, 2001a)). On the contrary, a conditional evaluation assesses the importance of a feature for the prediction model given – i.e., in addition to – the other features in the model (conditional feature importance, e.g., CPI (Watson and Wright, 2021)).

In the absence of dependency structures within  $\mathbf{X}$ , the two concepts coincide, yet real data frequently *does* exhibit dependency structures; hence, the distinction becomes crucial as the two concepts address different research questions. For example, marginal importance measures will assess a feature’s predictiveness for the model in a general sense, reflecting also the relevance of predictive information associated with correlated features (for example, in the case of a confounding feature<sup>1</sup>; see Paper 1, Figure 1 for an illustration). When applying a marginal method without acknowledging this effect (and instead, the research question intended to investigate only the direct importance of a feature to the target), the importance scores returned are prone to be misinterpreted. However, it is worth emphasizing that no one-fits-all notion of feature importance exists. Instead, the decision to apply marginal or conditional feature importance metrics has to depend on the research question.

---

<sup>1</sup>The example in Paper 1 works as follows. Imagine a data generating process where the confounding feature  $C$  affects both feature  $X$  and target  $Y$  ( $X \leftarrow C \rightarrow Y$ ), but  $X$  has no direct effect on  $Y$  ( $X \not\rightarrow Y$ ). Fitting a machine learning model, such as a random forest, to predict  $Y$  from  $C$  and  $X$ , a marginal measure of feature importance will yield nonzero importance scores for both  $C$  and  $X$ . That is because  $C$  induces correlation between  $X$  and  $Y$ , hence  $X$  is predictive of – and in that sense, important for the model in predicting –  $Y$ . In a conditional sense, however, a model cannot derive further information from  $X$  on  $Y$  beyond that induced by  $C$ . Hence, there should be no model reliance on  $X$  conditional on  $C$ , and hence, a conditional feature importance measure will in expectation attribute zero feature importance to  $X$ .

### 3.3 Individual Contributions: Parts I, II and III

---

Paper 1 discusses that for measuring and testing conditional feature importance, only few methods exist and an absence of methods suitable for mixed data hinders empirical application. Specifically, the paper demonstrates that workarounds, e.g., using a dummy encoding of categorical features and treating them as continuous, lead to reduced power in testing for non-zero feature importance. Notably, this consequence persists even if the explained supervised machine learning model would have been suitable for mixed data. This finding underpins the necessity that the machine learning algorithm *and* the interpretability method applied must be suitable for mixed data.

For the measurement of conditional feature importance with mixed data, Paper 1 proposes to combine sequential knockoffs (Kormaksson et al., 2021) with CPI (Watson and Wright, 2021). In brief, the procedure exploits that CPI can, in principle, work with any valid knockoff sampler and therefore integrates the sequential knockoff algorithm that is particularly designed for mixed data in the procedure.<sup>2</sup>

In sum, Part I accentuates the need to account for real data characteristics to ensure powerful and adequate feature importance measurement. Dependency structures require nuanced methodology and mixed data types demand suitable algorithms. The main contribution of Paper 1 is to provide a tool for measuring and testing conditional feature importance with mixed tabular data.

**Part II** comprises the paper “Unfooling SHAP and SAGE: Knockoff imputation for Shapley values” (Blesch, Wright, and Watson, 2023), highlighting the need to account for dependency structures in IML. The work demonstrates that failing to respect feature dependencies enables adversaries to manipulate explanations, which replicates and extends findings in previous literature (Slack et al., 2020). At the center of this contribution is the proposition to draw on knockoffs to increase the robustness of Shapley value explanations against adversarial attacks.

Adversarial attacks<sup>3</sup> in IML aim to manipulate the resulting explanations such that they do *not* reflect the model’s actual behavior; see Baniecki and Biecek (2023) for a survey. As an example (see further Blesch, Wright, and Watson, 2023; Slack et al., 2020), imagine a loan company that aims to deploy a model for making credit assessments. Before the model can be used in the real world, it must pass a fairness audit imposed by legal authorities. The loan company is worried because the model bases its decisions solely on the applicants’ **gender**, a feature perceived as discriminatory. If the audit’s IML assessment (which will be carried out by Shapley value explanations, precisely, KernelSHAP (Lundberg and Lee, 2017)) attributes a high feature attribution score to **gender**, model deployment will be prohibited. The loan company, however, is keen to deploy the discriminatory model for real-world application and therefore wants to trick the auditing process by an adversarial attack on the explanations to hide the model’s dependence on **gender**.

As an adversarial strategy to misguide explanations, Slack et al. (2020) propose to deploy an adversarial model  $\alpha$  which predicts real data through some (discriminatory) model  $f$ , yet uses a different (fair) model  $\psi$  with data occurring during IML assessment. As a result, assessments on  $\alpha$  will reflect the predictive behavior of  $\psi$ . This strategy effectively fools the explanation

---

<sup>2</sup>Note that even though the literature on model-X knockoffs is advancing rapidly, recent developments hardly address the mixed data case (Romano et al., 2020; Sesia et al., 2018). This observation mirrors that mixed data is a largely overlooked data type in the literature on knockoffs as well.

<sup>3</sup>Note that adversarial attacks in machine learning more generally may refer to a variety of scenarios (Cartella et al., 2021; Goodfellow et al., 2014; Slack et al., 2020), and the idea of an adversary is revisited in a different context in Part III.

because it does not reveal  $\alpha$ 's actions on real data, which the IML explanation method intended to accomplish. In the example above, this would translate to the loan company passing an adversarial model  $\alpha$  to the auditors, for which KernelSHAP explanations would reflect the behavior of some non-discriminatory model  $\psi$ , even though the loan company would take decisions in the real world based on discriminatory model  $f$ .

Paper 2 replicates the study of Slack et al. (2020), which illustrates the effectiveness of this adversarial attack on KernelSHAP values (Lundberg and Lee, 2017), and further demonstrates that the problem extends to Shapley value explanations on a global level (SAGE values). In addition, Paper 2 investigates determinants for the attack's success, finding that higher levels of correlation in the data set facilitate the adversarial attack. The work highlights that detecting generated data points occurring only during IML assessment is decisive for the attack's success because this indicates to the model that it is under evaluation. Framing this issue through the lens of statistics, again, a marginal in contrast to a conditional conceptualization is essential: adequate respect for dependency structures can effectively defend against the attack.

In detail, IML techniques such as SHAP and SAGE typically rely on a marginal conceptualization for approximating feature attributions, which may evaluate the prediction function at extrapolated data points – the decisive factor for the attack's success. Extrapolated data points are easily identifiable because such data points are implausible to occur naturally for the given data dependency structure, i.e., are off the empirical data manifold. Adversaries can train an algorithm  $\omega$  to distinguish real from generated (extrapolated) data that occurs only during the IML assessment. The classification of whether the data point is real or extrapolated thus indicates to the adversarial model  $\alpha$  to apply either  $f$  or  $\psi$ . The underlying issue is that marginal imputation (as opposed to conditional imputation) during Shapley value calculation will break the dependency structure in the data and hence generates extrapolated data points.<sup>4</sup> Paper 2 highlights that any Shapley value estimation method that prevents extrapolation effectively prevents the attack.

Paper 2 proposes using knockoffs to impute out-of-coalition feature values as a concrete method to prevent adversarial attacks on Shapley value explanations. Knockoffs ensure that the data dependency structure is respected while allowing for flexible adaption to data types such as mixed data. The proposed procedure exploits that knockoffs can be swapped for the out-of-coalition features while maintaining the joint distribution (see Section 2.2.1).

In the context of Shapley value calculation, this directly ensures that the generated data will lie on the same manifold as the original data. Therefore,  $\omega$  will classify data points generated in such a way as real data instances. Hence, the derived explanations on  $\alpha$  will be faithful to the model's behavior on real data. In the example above, an IML audit via knockoff imputed Shapley value explanations would reveal that the model relies on a discriminatory feature. As a result, if the loan company aims for an explanation suggesting that the model is fair, it has to use the fair model for real world data.

An advantage of the procedure is that knockoffs can be calculated upfront and then swapped in for the out-of-coalition features as needed. Alternative procedures that prevent extrapolation,

<sup>4</sup>Recap that the calculation of Shapley values requires the evaluation of predictions with all possible coalitions (feature subsets  $S \subseteq \mathcal{P}$ ), but the prediction model typically demands  $p$ -dimensional input. Values for out-of-coalition features  $\mathbf{x}_{-S}$  can be imputed to meet this requirement, generating new data points  $\mathbf{x}$ . A simple and fast procedure is to randomly sample values from other instances; however, this implicitly assumes independence across features (Lundberg and Lee, 2017) and thus is a marginal conceptualization that ignores dependency structures in the data. Doing so places the generated data points in sparse data regions as well.



### 3.3 Individual Contributions: Parts I, II and III

---

such as conditional Shapley values, have to estimate  $2^P$  conditional densities instead, which is challenging in practice (Kumar et al., 2020). With knockoff imputed Shapley value explanations, model auditors can – thanks to the modular knockoff methodology and model-agnosticism of Shapley values – assess various models and data types with the same procedure, ensuring that adversarial attacks as described above cannot fool the explanation.

In sum, Paper 2 highlights the necessity to respect dependency structures to protect against adversarial attacks and proposes a flexible and modular procedure that is based on knockoffs to do so.

**Part III** focuses on mixed tabular data with a dependency structure in the context of generative modeling. It presents Paper 3 (Watson et al., 2023), which proposes, a fast, lightweight procedure that is particularly suitable for, but not limited to, mixed tabular data is proposed to advance the methodology for density estimation and generative modeling. The method yields competitive results with state-of-the-art competitors and is a promising procedure for synthesizing data, which might also be helpful for applications in IML. To enhance real data applicability for users across disciplines, Paper 4 (Blesch and Wright, 2023) provides a practical guide and introduces a user-friendly implementation in the *Python* programming language to facilitate the accessibility of the method.

Paper 3 addresses the question of adequate methodology for mixed tabular data by tackling the overly complex design of state-of-the-art methods in generative modeling. Instead of applying tuning-intensive and computationally expensive deep learning architectures, the proposed methodology draws on random forests as a base learner. This ensemble learner allows for flexibility in modeling dependency structures with few requirements in tuning and computational resources. Further, it does not require encoding schemes for categorical features, making it particularly suitable for mixed tabular data. Tree-based ensemble methods have been found advantageous over deep learning approaches on tabular data in previous research (Grinsztajn et al., 2022; Shwartz-Ziv and Armon, 2022). Leveraging these favorable traits to generative modeling, the paper introduces the adversarial random forest (ARF) procedure that allows users to generate high-quality data with a ready-made solution that is fast and straightforward to apply.

In detail, ARFs work as follows: First, a random forest is fitted to differentiate real from naively synthesized data, which consists of perturbed feature values.<sup>5</sup> While training the random forest (discriminator step), the dependency structure across features is learned by the splits in the trees. From the observations within the individual leaves of the random forest, a new set of synthetic data can be sampled (generation step). With the newly sampled synthetic data, again, a random forest is fitted to distinguish it from real data (discrimination step). This procedure, inspired by the adversarial training procedure in GANs, is continued until the accuracy of the discriminator random forest is  $\leq 0.5$ , which indicates that the forest has learned all dependency structures (algorithm convergence). From the observations in the leaves of this forest, it is possible to estimate parameters (density estimation) and sample new data points by using the estimated distribution parameters (generative modeling).

From the procedure, it is readily apparent how ARFs effectively address and capitalize on dependency structures in the data. At algorithm convergence, there are no more dependencies to pick up from the observations in the leaves, so it is reasonable to assume that the features in

---

<sup>5</sup>This procedure is also known as fitting an unsupervised random forest; see further Shi and Horvath (2006).

the leaves are mutually independent. These feature independencies substantially simplify the task of joint density estimation because the challenging task of modeling the multivariate density transforms into the much simpler task of performing several univariate density estimations. As a consequence, this allows for sampling new data instances separately from the univariate densities in the respective leaf while maintaining the dependency structure between features overall for the generated data. This simplification makes the ARF a straightforward to use explicit generative model. To fully leverage its potential, it is crucial to provide further a suitable stand-alone software implementation.<sup>6</sup>

Paper 4 aligns the methodological advancement of ARFs with the demands of practitioners regarding software. The contributing paper presents a *Python* implementation of ARFs alongside comprehensible software documentation and a tutorial-style usage guide. This work establishes the crucial link between methodology and application and equips users with adequate software that levels up the impact of the method for a broad spectrum of applications.

Summarizing the contributions of Part III, the methodology proposed in Paper 3 and *Python* software implementation presented in Paper 4 advance the field of generative modeling by providing a fast, straightforward, and user-friendly method that exploits the characteristic traits of tabular (mixed) data.

#### 3.4. Connections between Parts: Joint Contribution

After discussing each part's contributions individually, this subsection explores the connections and synergies between parts in addressing the research gap.

A straightforward connection between Parts I and II is the utilization of knockoffs to account for dependency structures. Both parts draw on this methodology yet illuminate different facets of dependency structure acknowledgment. There are implications for interpreting IML explanations (marginal versus conditional feature importance) and the robustness against adversarial attacks. From the connection between the two parts via the knockoff methodology, it is apparent that methods that address one facet of dependency structures can also help address other aspects related to them. This observation opens up promising lines of future research that consider the implications of accounting for dependency structures in conjunction and encourages the transfer of methods across disciplines.

Further, the implications of an account for dependency structures are not limited to the specific methods discussed: a joint perspective reveals that consequences for interpretation (Part I) and robustness against adversarial attacks (Part II) pertain in reverse for the methods discussed. That is, the marginal feature importance measure PFI (described in Part I) is also susceptible to the adversarial attacks described in Part II because the method calculates the feature attributions through random permutations, which generate extrapolated data points. Replacing the marginal imputation step in PFI with a method that avoids extrapolation turns the measure into a conditional feature importance method, which alters the interpretation. For example, extending PFI with knockoffs to avoid extrapolation directly leads to CPI. Conversely, calculating Shapley values with knockoffs in the background distribution (as in Part II) will not only prevent the adversarial

---

<sup>6</sup>Note that Paper 3 provides code for the methodological experiments and a package implementation in the *R* programming language. However, many machine learning practitioners primarily use the *Python* programming language, which is the key motivation for Paper 4.

### 3.4 Connections between Parts: Joint Contribution

---

attack but also impact the interpretation. In this case, Shapley values reflect the difference to a knockoff version of the instance instead of the average prediction. This consequence may, however, be willingly tolerated in the light of adversarial attacks on legal auditing procedures and brings us directly to the essence of this joint perspective on accounting for dependency structures: an awareness of the various consequences is essential for balancing intended and unintended implications when choosing an IML method for a given research question and context of application.

The opportunities to transfer valuable insights from different methodologies further become clear when combining considerations of Parts I and II with those of Part III. Advances in the fields of IML and generative modeling can be mutually beneficial. Many IML methods rely on the sampling of feature values as a subroutine, and considerations regarding dependency structures can be decisive in this context. Generative models can flexibly generate new data samples that obey the data distribution and hence can be used as a subroutine. Future research could combine various feature attribution measures with generative models, e.g., PFI with ARF imputation for a flexible measure of conditional feature importance. However, generative modeling is a broad field and posing additional statistical desiderata on the generated data can yield further advantageous procedures. For example, Part II discusses the specialized generative procedure of knockoff sampling for imputing values during Shapley value calculation, which actively takes advantage of the swap property satisfied by knockoffs. Note that the transfer of knowledge also applies the other way around, and considerations in IML can help advance algorithms in generative modeling. Part III explored this by leveraging the characteristics of mixed tabular data for generative modeling, which originated from challenges that IML methods like PFI, CPI, or Shapley faced in Parts I and II. The benefits of sourcing inspiration from different fields is reemphasized through this joint perspective on the three parts.

Detecting extrapolated data points, as in Parts II and III, further illuminates a more subtle connection for advancing methods in generative modeling and IML. In Part II, algorithm  $\omega$  attempts to distinguish (real) in-distribution from (generated) out-of-distribution data points to enable the adversarial model to behave differently on generated data. In the ARF procedure proposed in Paper 3, the discriminator step's random forest encounters the exact same task, i.e., to distinguish real from generated data. However, in this context, the detection procedure is embedded in the adversarial procedure<sup>7</sup> to build a generative model. Through this connection, it is straightforward to consider extracting the discriminator random forest from the ARF procedure and use it as a detection algorithm  $\omega$  for elaborated adversarial attacks. Notably, Part II already uses a random forest as  $\omega$ , although without the iterative procedure for model fitting. Hence, a random forest extracted from an ARF procedure may make adversarial attacks more effective, e.g., in cases where data is only weakly correlated. Note, however, that methods that do not extrapolate the data manifold still effectively prevent adversarial attacks, even with advanced versions of  $\omega$ . Still, this line of thought exemplifies another direction of sharing similar concepts in IML and generative modeling.

Another noteworthy observation is that Parts I and III conquer the misalignment in data types from opposed starting points. Part I uncovers that the methodology discussed was previously unsuitable for mixed tabular data, requiring workarounds to make the method technically work for applications. Conversely, Part III shows that even though well-performing algorithms from other data types, e.g., image data, yield acceptable results with tabular data, refinements that align

---

<sup>7</sup>Note that the adversarial idea in the context of Parts II and III is closely linked to veiling or detecting generated data.

the specific requirements of mixed tabular data may still be favorable to facilitate the workflow. In conjunction, the two parts show that the methodological adequacy for data types, ideally, is addressed from both these viewpoints, encouraging powerful results (Part I) without overly complicated algorithms (Part III).

Finally, we can see how the three parts enhance real-world applicability by providing concrete methodology and software implementations that suit the needs of practitioners. Part I details the specifics of using CPI with sequential knockoffs (including software recommendations)<sup>8</sup> such that the method works with mixed data rather than leaving practitioners with claims on how the method may generalize to the mixed data case. Further, Paper 4 presents a stand-alone software implementation of the method proposed in Paper 3 through a tutorial-style paper and comprehensive software documentation. Contributions like this make methodology easily accessible to a wide range of users and promote methodological advancements to users through tailored software implementations. For example, 'arfpv' equips the *Python* based community with software that is easily accessible to them instead of expecting *Python* users to draw on fragile conversion wrappers of the 'arf' *R* package. In sum, the parts of this thesis work together to showcase that explicit, well-evaluated methodology and customized software are crucial to encourage real-data applicability.

To sum up the joint contribution of the three parts, the work highlights the necessity to account for dependency structures and the characteristic traits of mixed tabular data through a broad, interdisciplinary perspective that may include concepts from statistics, IML, and generative modeling. Further, the work underpins the necessity to provide adequate methodology as well as hands-on software tools to enable real-world applications.

---

<sup>8</sup>Note that the *R* package `cpi` includes an argument for specifying the knockoff function and sequential knockoffs have been added as an exemplary case to the vignette, such that the usage is readily apparent for practitioners, see <https://cran.r-project.org/web/packages/cpi/index.html> .

**Part I.**

**Conditional Feature Importance  
Measurement with Mixed Tabular Data**



# Paper 1. Conditional Feature Importance for Mixed Data

**Contributing Article:** Blesch, K., D. S. Watson, and M. N. Wright (2023). Conditional feature importance for mixed data. *ASTA Advances in Statistical Analysis*. <https://doi.org/10.1007/s10182-023-00477-9>.

**Copyright information:** Copyright 2023 by the authors. Creative Commons Attribution 4.0 International License (CC-BY 4.0).

**Author contributions:** The project idea was derived from previous work of David S. Watson and Marvin N. Wright and further developed and refined by all authors. Kristin Blesch conducted all simulations and experiments, drafted the manuscript and lead the revision process. David S. Watson and Marvin N. Wright supervised the project. All authors discussed and interpreted methodological implications, findings from simulations and experiments and contributed to proofreading and revising the paper.







## Conditional feature importance for mixed data

Kristin Blesch<sup>1,2</sup> · David S. Watson<sup>3</sup> · Marvin N. Wright<sup>1,2,4</sup>

Received: 6 October 2022 / Accepted: 28 March 2023  
© The Author(s) 2023

### Abstract

Despite the popularity of feature importance (FI) measures in interpretable machine learning, the statistical adequacy of these methods is rarely discussed. From a statistical perspective, a major distinction is between analysing a variable's importance before and after adjusting for covariates—i.e., between *marginal* and *conditional* measures. Our work draws attention to this rarely acknowledged, yet crucial distinction and showcases its implications. We find that few methods are available for testing conditional FI and practitioners have hitherto been severely restricted in method application due to mismatched data requirements. Most real-world data exhibits complex feature dependencies and incorporates both continuous and categorical features (i.e., mixed data). Both properties are oftentimes neglected by conditional FI measures. To fill this gap, we propose to combine the conditional predictive impact (CPI) framework with sequential knockoff sampling. The CPI enables conditional FI measurement that controls for any feature dependencies by sampling valid knockoffs—hence, generating synthetic data with similar statistical properties—for the data to be analysed. Sequential knockoffs were deliberately designed to handle mixed data and thus allow us to extend the CPI approach to such datasets. We demonstrate through numerous simulations and a real-world example that our proposed workflow controls type I error, achieves high power, and is in-line with results given by other conditional FI measures, whereas marginal FI metrics can result in misleading interpretations. Our findings highlight the necessity of developing statistically adequate, specialized methods for mixed data.

**Keywords** Interpretable machine learning · Feature importance · Knockoffs · Explainable artificial intelligence

---

✉ Kristin Blesch  
blesch@leibniz-bips.de

Extended author information available on the last page of the article

## 1 Introduction

Interpretable machine learning is on the rise as practitioners become interested in not only achieving high prediction accuracy in supervised learning tasks, but also understanding why certain predictions were made. Evaluating the importance of input variables (features) to the target prediction plays a crucial role in facilitating such endeavours. Several feature importance (FI) measures have been proposed by the machine learning community, but differing conceptualizations are spread across the literature.

We identify at least five dichotomies that orient FI methods: (1) global vs. local; (2) model-agnostic vs. model-specific; (3) testing vs. scoring; (4) methods that do and do not accommodate mixed tabular data; and (5) conditional vs. marginal measures. This defines a grid with  $2^5 = 32$  cells that helps categorize FI measures. For example, the popular SHAP algorithm (Lundberg and Lee 2017) produces local, model-agnostic FI scores that can accommodate mixed data and measures marginal FI. We emphasize that there is no “ideal” configuration of these five options—each is the right answer to a different question that is irreducibly context-dependent. However, this grid helps identify a notable lacuna: There are few global, model-agnostic FI methods that accommodate mixed data with error control for conditional FI measurement.

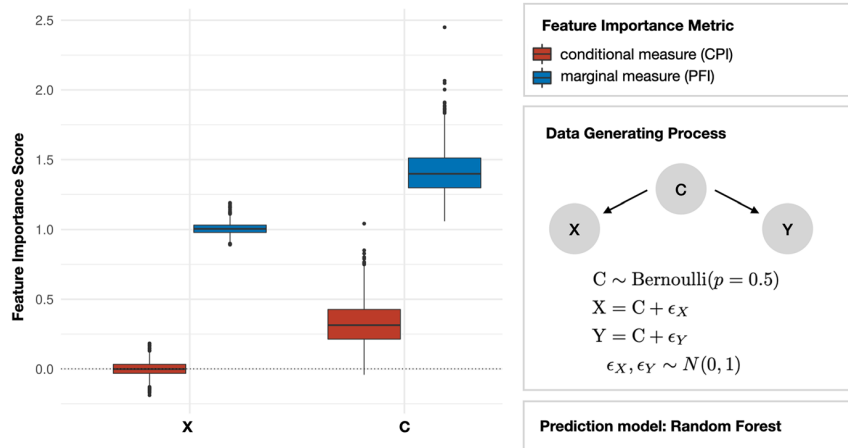
Explaining the dichotomies in more detail, local FI measures (Lundberg and Lee 2017; Ribeiro et al. 2016) are optimized for a particular point or region of the feature space, e.g., a single observation, while global FI scores (Fisher et al. 2019; Friedman 2001) measure a variable’s overall importance. Model-specific measures (Breiman 2001; Kurşa and Rudnicki 2010; Shrikumar et al. 2017) exploit the properties of a particular function class for more efficient or precise FI calculation, while model-agnostic measures (Apley and Zhu 2020; Ribeiro et al. 2018) treat the underlying model as a black box. Testing methods include some inference procedure for error control (Lei et al. 2018), while scoring methods (Covert et al. 2020) do not. Some methods are proposed with limited applicability to certain data types, e.g. only continuous inputs (Watson and Wright 2021), while others are more flexible (Molnar et al. 2023). We discuss a selection of FI methods briefly in Sect. 2, but refer readers to review papers on FI interpretability methods, e.g. Linardatos et al. (2021), for a wider discussion on the topic.

Through the lens of statistics, the division (5), conditional vs. marginal measures, is particularly important, yet insufficiently acknowledged in both literature and practice (Apley and Zhu 2020; Hooker et al. 2021; Molnar et al. 2023; Watson and Wright 2021). The complementary concepts become evident when relating the statistical conception of independence testing to the machine learning view on FI measurement. We can think of the marginal null hypothesis as testing whether the input feature  $X_j$  is independent of other covariates  $X_{-j}$  or the target variable  $Y$ :

$$H_0^M : X_j \perp\!\!\!\perp \{Y, X_{-j}\} \quad (1)$$

On the other hand, testing against (2) accounts for the covariates  $X_{-j}$  and hence corresponds to conditional FI:

### Marginal versus conditional feature importance (FI) measurement



**Fig. 1** Boxplots contrasting marginal and conditional FI metrics for a prediction of  $Y$  with  $C$  and  $X$  ( $N = 200$ ) through a random forest prediction model across 1000 replicates. The conditional FI measure attributes no importance to  $X$ , whereas the marginal measure attributes nonzero importance to  $X$  because (due to induced correlation between  $X$  and  $Y$  by  $C$ ) it is predictive of  $Y$

$$H_0^C : X_j \perp\!\!\!\perp Y \mid X_{-j} \tag{2}$$

These tests clearly target different objectives. In this setup, we have  $H_0^M$  entailing  $H_0^C$ , but not the other way around. However, this strength comes with a certain loss of specificity, because rejecting  $H_0^M$  leaves it unclear whether  $X_j$  is correlated with  $Y$ ,  $X_{-j}$ , or both.

The relationship between FI and independence testing sheds light on another aspect, which may even be considered another dichotomy: does the FI measure aim for investigating the model behaviour or the underlying data structure (Chen et al. 2020)? For example, conditional independence tests that are part of some conditional FI measures (Watson and Wright 2021) may be used for causal structure learning, which often is based on repeated conditional independence testing (Glymour et al. 2019). Therefore, conditional FI measures can help explain the underlying data structure, whereas marginal FI measures differentiate between variables the predictive model relies on, which can be used to evaluate the fairness of a model. This does not preclude practitioners from using marginal and conditional FI measures in conjunction, and since marginal measures are often faster to compute, they might be preferable for quick assessments in large pipelines with many iterations. However, practitioners must be careful to interpret these measures properly and not infer a conditional signal from a marginal test.

In Fig. 1, we illustrate the difference between marginal (permutation feature importance (PFI), Fisher et al. 2019, Breiman 2001) and conditional (conditional predictive impact with Gaussian knockoffs (CPIgauss), Watson and Wright 2021) FI

measures. In this example, the confounding variable  $C$  is a common cause of both  $X$  and  $Y$ . This causal structure induces spurious correlation between  $X$  and  $Y$ , leading the marginal FI measure to attribute nonzero importance values to both  $C$  and  $X$  in predicting  $Y$ . On the contrary, the conditional FI measure attributes nonzero FI only to  $C$ , since  $X$  has no *additional* predictive value for  $Y$  above  $C$ .

This paper explores global, model-agnostic FI methods that accommodate mixed data with error control for conditional FI measurement. This is not a niche problem: mixed tabular data is the norm in many important areas such as health care, economics, and industry, and inference procedures are essential for decision-making in high risk domains to minimize costly errors. With the proliferation of machine learning algorithms, model-agnostic approaches can help standardize FI tasks without recalibrating to a particular function class for each new application. Conditional, global measures are valuable when practitioners seek mechanistic understanding that takes data covariance into account and go beyond individual model outputs.

Even though the empirical relevance of this kind of FI measurement is eminent, specialized methods are lacking. Some FI methods have yet to be evaluated in mixed data settings (Covert et al. 2020; Molnar et al. 2023; Lei et al. 2018), while others are currently inapplicable to mixed data (Watson and Wright 2021). The consequences of neglecting the special nature of mixed data for conditional FI measurement remain unexplored, and therefore practitioners currently have no guidance on how to proceed with conditional FI measurement in such cases, which proves a severe limitation in real-world applications.

We propose to combine the *conditional predictive impact* (CPI) testing framework proposed by Watson and Wright (2021) with the use of *sequential knockoffs* (Kormaksson et al. 2021) in order to enable conditional, global, model-agnostic FI testing for mixed data. CPI is a flexible, model-agnostic tool that relies on the usage of so-called knockoffs (Candès et al. 2018). In short, knockoffs are synthetic variables that carry over the major statistical properties of the original variables, such as the correlation structure among covariates. While Watson and Wright (2021) claim that the CPI should in principle work with any valid set of knockoffs, it has thus far only been applied and evaluated with Gaussian knockoffs (Candès et al. 2018). This currently limits practitioners to using the CPI method only with continuous variables or to disregard the specialities of mixed data. We analyse consequences of such a disregard when using CPI with Gaussian knockoffs (Candès et al. 2018) (CPIgauss) and deep knockoffs (Romano et al. 2020) (CPIdeep) and propose a specialized solution strategy to tackle the mixed data case: using sequential knockoffs (Kormaksson et al. 2021)—a knockoff sampling algorithm explicitly developed for mixed data—within the CPI framework (CPIseq).

The paper will be structured as follows. We present relevant methodology and FI measures in Sect. 2. Section 2.2 reviews several knockoff sampling algorithms, demonstrating the need for specialized procedures with mixed data and motivating our proposed solution CPIseq. Through simulation studies in Sects. 3.1 and 3.2, we will evaluate our newly proposed workflow in more depth and further compare it to other methods. Finally, we illustrate method application to a real-world dataset in Sect. 3.3 before concluding and discussing our findings in Sect. 4.

## 2 Methods

With a focus on the measurement of model-agnostic, global, conditional FI, this section presents related measures proposed by previous literature and discusses their applicability to mixed data. We acknowledge that methods from the statistical literature on conditional independence testing (Shah and Peters 2020; Williamson et al. 2021) might also be utilized for conditional FI measurement; however, a full comparison of such methods is beyond the scope of this paper. Further, it is worth clarifying at this point that we understand FI here as a concept that is tied to the variable's effect on the predictive performance in a supervised learning task.

### 2.1 Feature importance measures

#### 2.1.1 Conditional subgroup approach (CS)

A global, model-agnostic FI measure that acknowledges the crucial distinction between conditional and marginal measures of importance is the *conditional subgroup (CS)* approach proposed by Molnar et al. (2023). CS partitions the data into interpretable subgroups, i.e., groups whose feature distributions are homogeneous within but heterogeneous between groups. The method is promising, as it explicitly specifies the conditioning between subgroups and further allows for an unconditional interpretation within subgroups. This means the method provides both a global conditional and a within-group unconditional interpretation, which sheds light on feature dependence structures.

To determine FI, CS evaluates the change in loss when the variable of interest is permuted within subgroups, which lowers extrapolation to low-density regions of the feature space, thereby mitigating a common problem with permutation-based approaches (Hooker et al. 2021). To decide on a suitable partition, the authors suggest determining subgroups via transformation trees. Using a pre-specified loss function, the average increase in loss is reported for multiple permutations versus the original ordering of variables.

CS is not affected by mixed data other than through the choice of an appropriate prediction algorithm, which is why this method is suspected to work equally well with mixed data. However, for this approach to work, researchers must assume that the data are separable into subgroups. Further, for testing FI, the method would need to rely on computationally expensive permutation tests as no inherent testing procedure is provided.

#### 2.1.2 Leave-one-covariate-out (LOCO)

*Leave-one-covariate-out (LOCO)* is a fairly simple approach to measuring FI, which, as the name suggests, evaluates the change in predictive performance of a model when leaving out a covariate of interest (Lei et al. 2018). This means, FI is

determined by comparing the loss of the model fitted including or excluding the covariate of interest.

While this is a very intuitive approach, it does involve several drawbacks. First, the model has to be retrained with a different set of variables, which not only incurs high computational cost, but also yields an entirely different model, raising concerns about comparability in general. Further, if correlations or other complex dependencies are present within the data, LOCO might give misleading results if only one covariate at a time is excluded, as this neglects potential interaction effects between groups of variables. In the presence of such group-wise structures, the exclusion of multiple covariates at a time is advisable (Au et al. 2022; Rinaldo et al. 2016).

For the speciality of mixed data, we can again see that all reliance is on the level of model choice, hence, as long as the prediction model is able to process mixed data, LOCO is not affected by different data types.

### 2.1.3 Shapley additive global importance (SAGE)

*Shapley additive global importance (SAGE)* (Covert et al. 2020) is a model-agnostic FI measure that aims to take into account feature interactions on a global level. The method is based on Shapley values (Shapley 1953), which have received much attention in interpretable machine learning recently. While Shapley values are widespread in their use for giving local explanations, i.e. explaining the role of features in individual predictions made by the model, Covert et al. (2020) propose a global extension such that the role of features can be understood on a model-wide level. SAGE values are Shapley values for the features with regard to the predictive power of the model. Therefore, SAGE values can also be calculated by directly calculating Shapley values for the model loss, e.g. as proposed in LossSHAP (Lundberg et al. 2020), and then average across all instances to achieve a global measure. However, Covert et al. (2020) propose a fast approximation algorithm.

The SAGE methodology allows for taking feature interaction effects into account, however, in practice, implementations typically use marginal sampling as an approximation to the conditional densities when sampling to replace the respective feature in various coalitions. This results in explanations that are comparable to marginal measures of FI when applied to real-world data.

Mixed data affect SAGE at the variable sampling step to build the coalitions and through the choice of the predictive model. With the use of marginal imputation and a model that is able to process mixed data, SAGE should not be affected by mixed data types.

### 2.1.4 Conditional predictive impact (CPI)

A fairly general approach to tackle conditional FI measurement is the *conditional predictive impact (CPI)* proposed by Watson and Wright (2021). To capture conditional FI, a flexible conditional independence test is introduced that works with any supervised learning algorithm, valid knockoff sampler and well-defined loss function. CPI ties FI to predictive performance, arguing that the inclusion of a relevant

variable in the model should improve its predictive performance. Building on this idea, first, a supervised learning algorithm is trained to predict the outcome from given input variables. Then, using a knockoff sampling algorithm, so-called knockoff copies of the input features are generated. These knockoffs retain the covariance structure of the input features,<sup>1</sup> but are (conditional on the input features) independent of the response variable. They therefore serve as a set of negative controls against which to compare the original data. In detail, to compute the CPI statistic, the trained model from the first step is used to predict the target twice: first using the original test data, and again after replacing one or several features of interest in the test data by their knockoff copies. The change in loss is then averaged across samples. Finally, the authors propose to apply inference procedures, such as a paired  $t$  test, to get valid  $p$ -values and confidence intervals for the FI scores.

Given that the prediction algorithm works with mixed data, sampling valid knockoffs for mixed data is the sticking point. As Watson and Wright (2021) claim, the CPI setup is knockoff-agnostic and hence works for any knockoff sampler. However, their simulations are limited to settings of continuous data and Gaussian knockoff sampling, i.e., using CPIgauss, only. Resulting from this, practitioners facing mixed data cannot use CPIgauss directly and are forced to use workarounds that may perform poorly in practice, e.g. dummy encoding variables and treating them as continuous, of which the effects on the method are thus far unknown. The present work sheds light on the consequences of such procedures, see further Sect. 3.1. To propose an efficient way of making CPI applicable to mixed data, we will now delve into the methodology of knockoffs in greater depth.

## 2.2 Model-X knockoffs

The model-X knockoff framework (Candès et al. 2018) was proposed for variable selection while controlling the false discovery rate (FDR). The idea is to use knockoffs as negative controls in the model, which prevents spuriously correlated variables from being detected as important. These knockoffs are a set of variables  $\tilde{\mathbf{X}}$  that mimic the correlational structure between the original input variables  $\mathbf{X}$ , but crucially are known to be irrelevant to the target variable  $Y$ , conditional on the input data. Intuitively, if  $X_j$  does not significantly outperform  $\tilde{X}_j$  by some importance measure, then  $X_j$  can be removed from the model (Candès et al. 2018).

More formally, to construct a valid knockoff matrix  $\tilde{\mathbf{X}}$  for the  $p$ -dimensional feature matrix  $\mathbf{X}$ , two conditions have to be met. The first is pairwise exchangeability, i.e. for any proper subset  $S \subset \{1, \dots, p\}$ :

$$(\mathbf{X}, \tilde{\mathbf{X}})_{\text{swap}(S)} \stackrel{d}{=} (\mathbf{X}, \tilde{\mathbf{X}}), \quad (3)$$

where  $\stackrel{d}{=}$  represents equality in distribution and  $\text{swap}(S)$  indicates swapping the respective variables in  $S$  with their knockoff counterparts. The second condition is conditional independence, i.e.

<sup>1</sup> This holds true for knockoffs that are at least of second-order, i.e. exhibit the same first two moments as the original data.

$$\tilde{\mathbf{X}} \perp\!\!\!\perp Y \mid \mathbf{X}. \quad (4)$$

Knockoff methodology is an active field of research. Numerous approaches to knockoff sampling have been proposed, for example, methods based on distributional assumptions (Bates et al. 2021; Candès et al. 2018; Sesia et al. 2018), Bayesian frameworks (Gu and Yin 2021) or deep learning (Jordon et al. 2019; Liu and Zheng 2018; Romano et al. 2020; Sudarshan et al. 2020). While a comprehensive review of knockoff samplers is beyond the scope of this paper, we will present a selection of knockoff samplers that is particularly interesting for applications on mixed data. Namely, we will investigate Gaussian knockoffs (Candès et al. 2018) because of their widespread use, deep knockoffs (Romano et al. 2020) as a representative of deep learning based knockoff generation, and sequential knockoffs as a specialized approach to tackle mixed data.

### 2.2.1 Gaussian knockoffs

As the name suggests, the *Gaussian knockoff* sampler (Candès et al. 2018) is based on the assumption that the input data matrix  $\mathbf{X} \in \mathbb{R}^{N \times p}$  is multivariate Gaussian, i.e.  $\mathbf{X} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ . For simplicity, we assume  $\boldsymbol{\mu} = \mathbf{0}$  and get for the joint distribution which satisfies Eq. (3)

$$(\mathbf{X}, \tilde{\mathbf{X}}) \sim N(\mathbf{0}, \mathbf{G}), \quad \text{where } \mathbf{G} = \begin{bmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma} - \text{diag}\{s\} \\ \boldsymbol{\Sigma} - \text{diag}\{s\} & \boldsymbol{\Sigma} \end{bmatrix}$$

with diagonal matrix  $\text{diag}\{s\}$  to ensure positive semi-definiteness of the joint covariance matrix  $\mathbf{G}$ . Knockoffs can then be sampled from the conditional distribution  $\tilde{\mathbf{X}} \mid \mathbf{X} \stackrel{d}{=} N(\boldsymbol{\mu}, \mathbf{V})$ , where  $\boldsymbol{\mu}, \mathbf{V}$  can be calculated from regular regression formulas. For details see Candès et al. (2018).

Clearly, it is reasonable to suspect this knockoff sampler to work well with Gaussian data. However, with mixed data types, discrete values can only be handled after encoding, e.g. introducing dummy variables, which are evidently non-Gaussian. The consequences of such transformations, i.e. neglecting the special nature of mixed data, have not yet been evaluated for the Gaussian knockoff sampler. In an attempt to quantify such implications to some extent, we will include this knockoff sampler in our analysis in Sect. 3.1 and compare it to more well-suited alternatives.

### 2.2.2 Deep knockoffs

*Deep knockoffs* as proposed by Romano et al. (2020) rely on a random generator, consisting of a deep neural network, to sample valid knockoffs. For variables  $\mathbf{X}$  sampled independently from an unknown distribution  $P_{\mathbf{X}}$ , the random generator is trained such that the joint distribution of  $(\mathbf{X}, \tilde{\mathbf{X}})$  is invariant under swapping, such that Eq. (3) is satisfied. In detail, the neural network takes variables  $\mathbf{X}$  and i.i.d. sampled noise  $\mathcal{E}$  as input to optimize a scoring function that quantifies the extent to which  $\tilde{\mathbf{X}}$  is a good knockoff copy for  $\mathbf{X}$  by evaluating how well Eq. (3) is approximated. Considering



the neural network architecture, the authors suggest using a width  $h$  that is ten times the dimensionality of the input feature space, i.e.  $h = 10p$  and six hidden layers which they claim should work well for a “wide range of scenarios”, but acknowledge that “more effective designs” might be found (Romano et al. 2020).

Making use of recent deep learning advances, deep knockoffs should—according to the authors—generalize well to the mixed data case. Romano et al. (2020) claim that this framework samples approximate knockoffs for arbitrary distributions. However, it is worth noting that there is little explicit methodology available to the user beyond making general claims about the generalizability of the method. Therefore, an applied user is again left with a knockoff sampler that does not return valid mixed data knockoffs.<sup>2</sup>

### 2.2.3 Sequential knockoffs

*Sequential knockoff* (Kormaksson et al. 2021) sampling is based on the conditional independent pairs algorithm (Candès et al. 2018) given in Supplementary Information A with a specialized strategy to model the conditional distribution  $P(X_j | X_{-j}, \tilde{X}_{1:j-1})$  and sample knockoffs for mixed data.

Sequential knockoffs are synthesized by sampling continuous knockoffs from a Gaussian distribution and categorical knockoffs from a multinomial distribution with distribution parameters that have been sequentially estimated through penalized<sup>3</sup> linear or multinomial logistic regression models. The procedure is given in more detail in Algorithm 1, where  $X_{-j} := (X_1, \dots, X_{j-1}, X_{j+1}, \dots, X_p)$  and  $\tilde{X}_{1:j-1} := (\tilde{X}_1, \dots, \tilde{X}_{j-1})$ .

---

**Algorithm 1** Sequential knockoffs through conditional independent pairs

---

```

j = 1
while j ≤ p do
  if Xj continuous then
    sample  $\tilde{X}_j$  from  $N(\hat{\mu}, \hat{\sigma})$  with  $\hat{\mu}, \hat{\sigma}$  estimated from
    penalized linear regression  $\tilde{X}_j \sim X_{-j}, \tilde{X}_{1:j-1}$ 
  else if Xj categorical then
    sample  $\tilde{X}_j$  from  $Multinom(\hat{\pi})$  with  $\hat{\pi}$  estimated from
    penalized multinomial logistic regression  $\tilde{X}_j \sim X_{-j}, \tilde{X}_{1:j-1}$ 
  end if
  j = j + 1
end while

```

---

Algorithm 1 yields valid knockoff copies for data that may consist of both categorical and continuous covariates. Hence, the present paper puts a special focus on

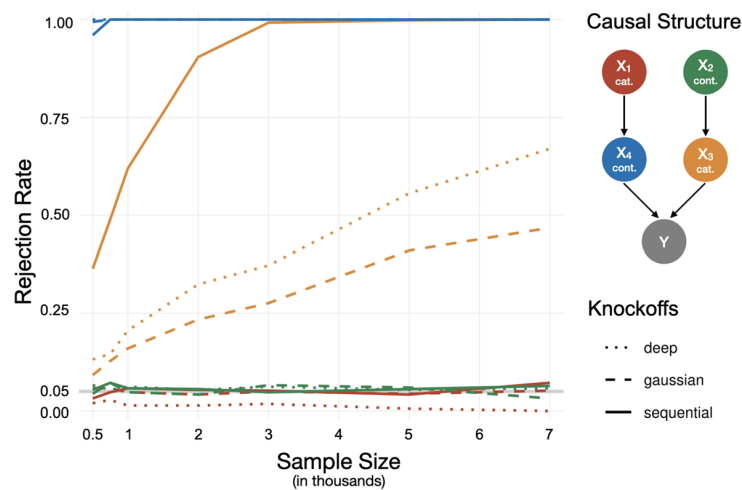
<sup>2</sup> Deep generative models for mixed data is an active and promising area of research (Xu et al. 2019; Watson et al. 2022), although we are unaware of any implementation for knockoff sampling.

<sup>3</sup> In our experiments, we follow the advice of Kormaksson et al. (2021) to use an elastic net (Zou and Hastie 2005). Note that the ordering of variables might be of relevance in finite samples and that the procedure requires the various levels of the categorical variable to occur sufficiently often.

this method and evaluates its suitability for conditional FI measurement with mixed data.

### 2.3 CPI with sequential knockoffs: CPIseq

We propose to combine two frameworks that have, thus far, not been analysed in conjunction, the CPI (Watson and Wright 2021) and sequential knockoffs (Kor-maksson et al. 2021), as a viable solution for conditional FI measurement with mixed data. Section 2 reveals that amongst the limited number of conditional FI measurement methods available, CPI is one of the few conditional FI methods that allows for the direct application of statistical testing procedures. Further, we have seen that the major obstacle of CPI with mixed data is the knockoff generation step. When surveying the literature on knockoffs in Sect. 2.2, the sequential knockoff sampler stands out as a solution that tackles the special nature of mixed data. Algorithm 2 presents details on the procedure we propose here. Note that for calculating CPIseq for several features (or groups)  $j$ , steps 1 and 2 of the algorithm do not have to be recalculated for each  $j$ .



**Fig. 2** Rejection rates of one-sided paired  $t$  tests at  $\alpha = 0.05$  to detect relevant variables, i.e. power and type I error rates, for CPI with various knockoff samplers across 500 simulation runs.  $X_1, X_3$  are 10-level categorical,  $X_2, X_4$  are Gaussian. Effect size  $\beta = 0.5$  and random forest prediction model

---

**Algorithm 2** CPI with sequential knockoffs (CPIseq)

---

**Input:**  $(X^{train}, Y^{train}), (X^{test}, Y^{test})$ , supervised learner  $f$ , feature (or group) of interest  $j$ , sequential knockoff sampler  $s$  (Alg. 1), loss function  $L$ , inference procedure  $h$

- 1: learn  $\hat{f} \leftarrow f(X^{train}, Y^{train})$
- 2: sample knockoffs  $\tilde{X}^{test} \leftarrow s(X^{test})$
- 3: for feature (or group)  $j$  calculate instance-wise loss difference
 
$$\Delta^{(i)} \leftarrow L(\hat{f}, X^{test(i)}) - L(\hat{f}, \{X_{-j}^{test(i)}, \tilde{X}_j^{test(i)}\})$$
- 4: calculate conditional predictive impact
 
$$\widehat{\text{CPI}} \leftarrow \frac{1}{N} \sum_{i=1}^N \Delta^{(i)}$$
- 5: apply inference procedure for  $p$ -value  $p$  and confidence interval  $ci$ 

$$p, ci \leftarrow h(\Delta)$$

**Output:**  $\widehat{\text{CPI}}, p, ci$

---

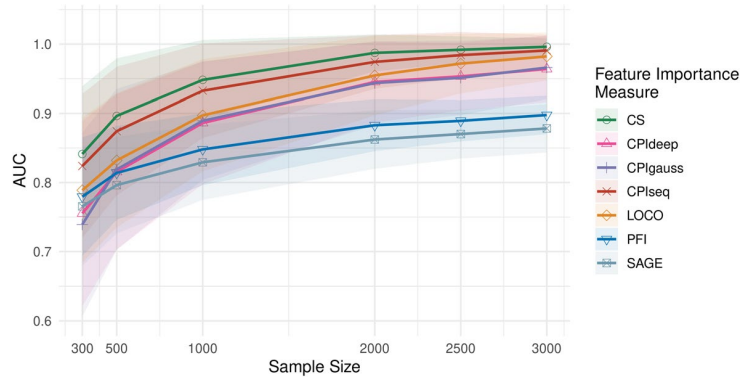
The CPIseq we propose here combines the features of the CPI methodology with ease of applicability to real data, which often consists of mixed data types. Providing frequentist inference procedures without model refitting is the major advantage over other conditional FI methods, such as CS and LOCO. To ensure high power for these testing procedures, adequate handling of mixed data is a prerequisite and CPIseq assures this through the flexible sequential knockoff subroutine.

### 3 Experiments

In this section, we analyse the performance of various FI measures on both simulated and empirical data. Through simulation studies, we evaluate the performance of our newly proposed workflow in comparison to other approaches. First, we investigate how CPIseq compares to CPI with other knockoff samplers, namely CPIgauss and CPIdeep (Sect. 3.1) in terms of power and effective FDR control. Further, we compare feature rankings given by our proposed approach and other conditional FI-related measures that do not use knockoffs (Sect. 3.2). Finally, we use a real-world data example to illustrate method application (Sect. 3.3).

#### 3.1 Comparing knockoffs

Major differences in the performance of CPIgauss, CPIdeep, and CPIseq on mixed data are illustrated using the following simulation setup. Consider a linear system of input variables  $S = \{X_1, X_2, X_3, X_4\}$  and target variable  $Y$ , visualized by the directed acyclic graph (DAG)  $\mathcal{G}$  in Fig. 2. Since the joint distribution is Markov with respect to  $\mathcal{G}$ , it follows by  $d$ -separation (Pearl 2009) that  $X_1 \perp\!\!\!\perp Y \mid S \setminus \{X_1\}$  and  $X_2 \perp\!\!\!\perp Y \mid S \setminus \{X_2\}$ , whereas  $X_3 \not\perp\!\!\!\perp Y \mid S \setminus \{X_3\}$  and  $X_4 \not\perp\!\!\!\perp Y \mid S \setminus \{X_4\}$ . Therefore, a



**Fig. 3** Mean AUC value with  $\pm$  one standard deviation across 500 simulation runs. Categorical variables with  $c = 5$  levels, pairwise correlation  $\rho = 0.8$  and a random forest prediction model for continuous target  $Y$

conditional FI measure should only attribute nonzero importance to variables  $X_3, X_4$ , but not to  $X_1, X_2$ . We consider three scenarios to track consequences of mixed data closely. For the baseline scenario (I),  $S$  will be Gaussian; for scenario (II),  $X_1$  or  $X_3$  will be binary; and in scenario (III),  $X_1$  and/or  $X_3$  will be categorical with  $c \in \{4, 10\}$  levels. Scenarios (II) and (III) further include an all categorical setting, i.e.  $S$  will be categorical, as a point of reference. We carefully select relevant combinations of category levels (2, 4 or 10), type of the target variable (continuous or binary) and fitted model (generalized linear model or random forest). See Supplementary Information B.1 to B.4 for further details on the experimental setup, including details on the prediction models and their validation.

### 3.1.1 Results

For scenario (I), we find CPI achieving high power and effective type I error control with every knockoff sampling algorithm. Naturally, as the data is Gaussian, we see CPIgauss achieving high power in this setting, see Supplementary Information Fig. 3. When transforming  $X_1$  and  $X_3$  into binary variables, (scenario (II)), we still observe high power and type I error control.

For input data consisting of mixed data types where the categorical variables are of high-cardinality (scenario (III)), we can see from Fig. 2 that the sequential knockoff sampler provides greater sensitivity than the deep or Gaussian alternatives across all tested sample sizes. Rejection rates for CPIseq grow quickly with sample size, reaching about 90% power around  $N = 2000$ . By contrast, CPIgauss only reaches about 50% and the deep knockoff sampler about 70% power at the maximal  $N = 7000$ . In terms of type I error control, all methods seem to be robust against the categorical nature of the irrelevant variable  $X_1$ , as the rejection rate in Fig. 2 is kept close to  $\alpha = 0.05$  for all knockoff samplers.

A full presentation of results is given in Supplementary Information B.5, including Figures for the all categorical cases, for which we find similar results as in mixed data settings.

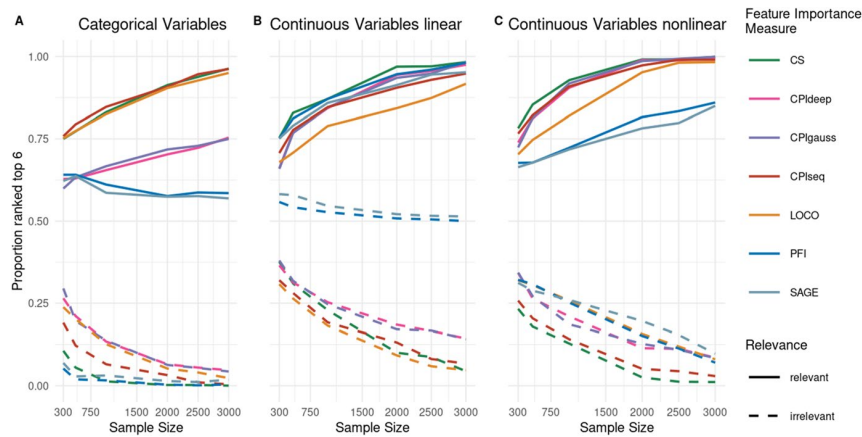
This simulation study demonstrates that the power of CPIgauss and CPIdeep might be severely affected by high-cardinality features. We find CPIseq to provide a powerful solution to conditional FI measurement, i.e. to detect conditionally important categorical features, whereas CPIgauss and CPIdeep are less sensitive with such data. It is worth noting that CPIgauss and CPIdeep perform surprisingly well when mixed data is limited to continuous and binary data types, even though Gaussian and deep knockoffs inevitably generate data outside the support of Boolean variables. Nevertheless, CPIseq appears to be the most powerful solution for conditional FI measurement with high-cardinality categorical data.

### 3.2 Comparing feature importance measures

Through a simulation study, our newly proposed workflow CPIseq will now be set in comparison with LOCO (Lei et al. 2018), CS (Molnar et al. 2023), SAGE (Covert et al. 2020), and permutation feature (PFI) importance (Breiman 2001; Fisher et al. 2019). Even though CPIgauss and CPIdeep have been shown to be outperformed by CPIseq in Sect. 3.1, we add these two methods to the simulation in order to provide a complete picture on how they relate to other measures of FI. Further enriching the picture of FI measure comparison, we discuss a random forest model-specific FI procedure (Kursa and Rudnicki 2010) and its performance in comparison to the other FI measures in the Supplementary Information C.6.

We simulate multivariate normal data with a pre-specified correlation structure to ensure a simple setup while incorporating a larger number of variables than in our toy example in Sect. 3.1. Again, we transform several variables into categoricals, such that we end up with mixed data. We distinguish between variables having zero, weak, or strong effect on the outcome  $Y$ , and for the continuous variables we further separate variables with a linear or nonlinear effect on  $Y$ . Further, we ensure that there is an equal number of relevant and irrelevant variables, such that each relevant variable is correlated with exactly one irrelevant variable of the same type, yielding a total of  $p = 12$  variables. In sum, we analyse a total of 24 settings by varying the correlation strength ( $\rho = 0.5$  or  $0.8$ ), type of target variable  $Y$  (continuous or binary), varying number of category levels ( $c = 2$  or  $5$ ) and fitting various machine learning prediction models (generalized linear model, random forest or neural network), see Supplementary Information C.1 and C.2 for further details.

Some of the methods included in the comparison do not provide statistical testing procedures. Therefore, we will compare methods by their tendency to rank relevant features higher than irrelevant alternatives. By construction,  $p = 6$  variables are relevant to the outcome, whereas the other  $p = 6$  variables are not. Hence, when we ask the methods to rank the variables according to their importance, ideally, the 6 relevant variables are ranked amongst the top 6. We will use the area under the receiver operating characteristic curve (AUC) as a measure of



**Fig. 4** Proportion of features ranked amongst the top 6 of 12 by variable type across 500 simulation runs. Solid lines (relevant variables) correspond to sensitivity, dashed lines (irrelevant variables) correspond to 1-specificity. Categorical variables with  $c = 5$  levels, pairwise correlation  $\rho = 0.8$  and a random forest prediction model for continuous target  $Y$

performance and will further report sensitivity and 1-specificity for each of the methods. See further Supplementary Information C.3.

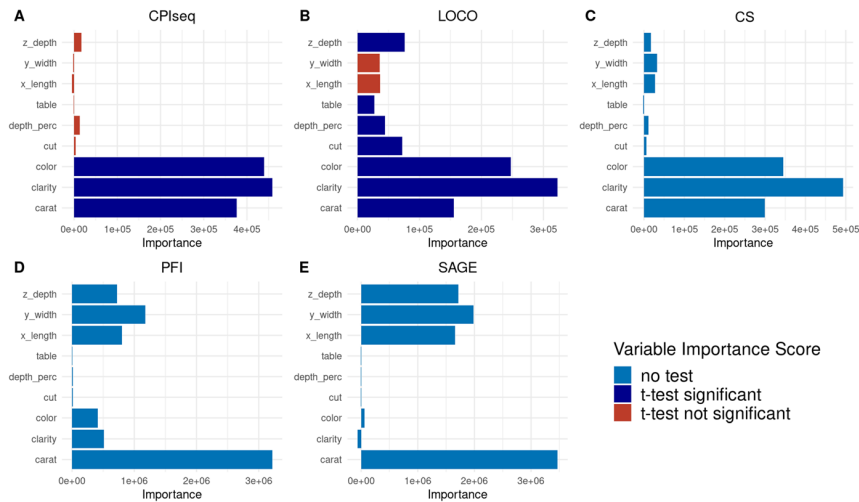
### 3.2.1 Results

We find CS, CPIseq, CPIgauss, CPIdeep and LOCO outperforming PFI and SAGE in ranking the relevant variables amongst the top 6 variables in terms of AUC scores (Fig. 3). AUC scores rise with increasing sample size, however, while the conditional measures form a group that gets close to the optimal score of 1, the performance of marginal measures<sup>4</sup> flattens out. This behaviour stems from the phenomenon of marginal methods to attribute nonzero importance to correlated, but irrelevant variables, affecting the methods ability to separate the top 6 from the bottom 6 variables, as can be further investigated from Fig. 4.

Figure 4 depicts the proportion of the respective variable types being ranked amongst the top 6 variables. Ideally, this proportion should be high for relevant variables (solid lines) and low for irrelevant variables (dashed lines). Panel (B) shows that both PFI and SAGE mistakenly rank the irrelevant continuous variables with a linear effect, which are correlated to the relevant continuous linear variables, amongst the top 6 variables. This is unsurprising, because relevant continuous variables with a linear effect on the target are the easiest to detect, and hence, irrelevant variables correlated to these variables are most likely to be mixed up by marginal measures in the full ranking. Note that because each of the methods has to assign ranks 1–12, an irrelevant variable being mistakenly ranked amongst the

<sup>4</sup> Note that SAGE here is closer to a marginal measure because of the marginal imputation subroutine.

### Conditional feature importance for mixed data



**Fig. 5** Feature importance scores for predicting the selling price of diamonds using a random forest model. For the CPIseq and LOCO, *t*-tests are at  $\alpha = 5\%$ , using the Holm procedure to adjust for multiple testing

top 6 variables in return leads to a relevant variable being ranked within the bottom 6 ranks. For example, due to the marginal measurement of FI, the PFI measure is ranking correlated yet irrelevant variables amongst the most important predictors (dashed line in Fig. 4, Panel B), which in turn forces PFI to mistakenly rank some relevant variables low (solid line in Fig. 4, Panel A).

Regarding the comparison of CPI-based methods, we find CPIseq outperforming CPIgauss and CPIdeep in detecting relevant categorical variables in the mixed data setting, see Fig. 4, Panel A, which underpins the findings of simulations in Sect. 3.1.

To check for robustness, we used several predictive models (generalized linear model, random forest, and neural network), varied the type of the target variable (regression or classification task) and the number of categories for the categorical variables (2, 5), and found similar results. Further, we analysed the fit of the prediction models on test data to ensure reliable FI measurement. See Supplementary Information C.4 and C.5 for details on the robustness analyses.

In sum, this simulation demonstrates both that CPIseq is competitive with other conditional FI measures, and illustrates the importance of distinguishing between marginal and conditional measures. It is worth emphasizing again that the CPIseq workflow not only ranks features, but also enables powerful conditional FI testing. We will see the practical relevance of this in the following section.

### 3.3 Real-world data

We conclude the section on experiments with a real-world data application to illustrate our proposed workflow on empirical mixed data. As an example, we use the

diamonds dataset which is publicly available on OpenML<sup>5</sup> (Vanschoren et al. 2014). Consisting of 9 covariates (6 numerical, 3 categorical) which relate to characteristics of diamonds such as length, depth and colour. We predict the selling price of the diamond in USD (`price`) using a random forest prediction model. Similar to the experiments in Sect. 3.2, the importance of the covariates for the prediction model will be determined by CPIseq, CS, LOCO, PFI and SAGE. For further details on the dataset and the procedure, as well as a comparison to results given by another prediction model (neural network), see Supplementary Information D.

Figure 5 illustrates the difference between conditional and marginal measures of feature importance. The marginal measures (Fig. 5, Panels D, E) attribute high importance scores to the covariates `x_length`, `y_width`, `z_depth` and `carat`, whereas the conditional measures (Fig. 5, Panels A, B, C) attribute high importance scores to the covariates `colour`, `clarity` and `carat`. Note that the scale of the FI measures in Fig. 5 differs, since marginal measures also incorporate the importances of correlated variables and hence, by construction, exhibit much larger values than conditional FI measures.

With some background knowledge on the physical characteristics of diamonds, we can understand the causal relationships that lead to this result. Carat is a measure of weight, and with round diamonds, this weight can be approximated by the formula  $carat = length \times width \times depth \times 0.0061$  (Miller 1988). Note that to ensure this formula holds, we only considered diamonds with a deviation  $< 0.02$  mm from a perfect round shape, yielding a subset of  $N = 4463$  observations. The covariates `x_length`, `y_width` and `z_depth` therefore determine the weight (`carat`), which all the importance measures suggest as an important predictor variable for `price`. Conditional FI measures then suggest that `x_length`, `y_width` and `z_depth` do not carry further information on the price, given the other covariates, including `carat`. Marginal measures, however, attribute importance irrespective of other covariates and hence do not condition on the information given by `carat`, which leads to high importance values for `x_length`, `y_width`, `z_depth` as well as `carat`, even though it is reasonable to assume that `carat` absorbs all relevant information given by `x_length`, `y_width` and `z_depth` on the price of diamonds.

The conditional FI measures further detect the variables `colour` and `clarity` to be relevant for the prediction of `price`. Note that we here again have to see this in a conditional sense. Given the other covariates, the variables `colour` and `clarity` do provide additional information on the price, whereas marginal measures estimate a rather low importance of these variables.

This real-world example emphasizes the difference between conditional and marginal FI measures and its implications. Again, it is worth repeating that out of the conditional measures, CPIseq facilitates the interpretation through inference procedures providing a clear indication of the relevant variables, whereas this indication is less clear with the LOCO testing procedure and CS not providing the user with testing procedures at all.

<sup>5</sup> <https://www.openml.org/search?type=data &sort=runs &id=42225>.



## 4 Conclusion and discussion

In this work, we highlight the importance of taking statistical considerations into account when measuring FI in interpretable machine learning. Specifically, we focus on conditional versus marginal perspectives on FI measurement, and analyse conditional FI methods with special regard for mixed data. We introduce the combination of CPI and sequential knockoffs (CPIseq) as a strategy that enables testing of conditional, model-agnostic, global FI with mixed data. Through simulation studies, we show that CPIseq achieves high power, whereas CPIgauss and CPIdeep are less sensitive for categorical features. Further, we benchmark this method against other conditional FI measures, finding competitive performance, and use a real-world data example to illustrate empirical implications. In sum, we demonstrate that the CPIseq provides researchers with a powerful test for conditional FI while working on a global, model-agnostic level.

Our analyses are limited by the availability of specialized knockoff sampling algorithms for the generation of mixed data knockoffs. Astonishingly, the case of mixed data has not received much attention in the knockoff literature so far and even if some methods were claimed to generalize to the mixed data case (Romano et al. 2020), there is a lack of concrete methodology and software implementation. Also, the scarce availability of conditional FI measures that allow for effective statistical testing impedes efficient comparison between FI metrics, forcing the evaluation to rely on rankings. While rankings are oftentimes used in the literature on FI for illustrative purposes, a systematic gold standard for comparing rankings between methods has not emerged. We hypothesize that this might be due to the fact that in the machine learning community, simulation studies—a standard procedure in the statistics community—are relatively rare, and hence evaluations involving, e.g. ground truth variable rankings are not in the focus. In particular, with mixed data, a ground truth ranking of simulated variables is not straightforward since it is unclear how the categorical nature should be respected and challenging disagreements across methods are likely to occur (Krishna et al. 2022). Methodological development that bridges evaluation strategies commonly applied in statistics with the setting faced in interpretable machine learning, e.g. FI rankings, is highly desirable.

This work highlights the necessity for procedures that respect data-specific requirements, such as respecting the categorical nature of variables in mixed datasets. Our simulations show that a neglect of such requirements and the application of workarounds might lead to undesirable consequences. We encourage researchers to develop methods that are specifically designed for realistic (mixed) data, instead of leaving practitioners with broad claims of the generalizability of their method. While some generalizations are indeed effortless, e.g. for conditional independence testing with all categorical data exact  $p$ -values can be computed through permutations (Tsamardinos and Borboudakis 2010), whereas conditional independence testing in general, including mixed data cases, is severely more challenging (Shah and Peters 2020). Moreover, other data type specific adjustments such as the presence of ordinal data might be of interest for future research, for example, random forest regression models yield the

same results with ordinal as with numeric data (Hastie et al. 2009) and hence FI methods that exploit model-specific advantages for ordinal data might be proposed.

Further, the present work raises awareness of the fact that even though the concept of FI might sound intuitive at first, statistical perspectives on the problem reveal that, for example, the question of marginal in contrast to conditional measurement is of fundamental relevance. We hope this paper elucidates the potential of advancing interpretable machine learning methodology through statistical considerations, which might in turn be mutually beneficial for the future development of the field of explainable artificial intelligence and statistics.

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1007/s10182-023-00477-9>.

**Acknowledgements** MNW and KB received funding for this project from the German Research Foundation (DFG), Emmy Noether Grant 437611051.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

**Data and code availability** Data and code to reproduce all the results presented in this paper is available at [https://github.com/bips-hb/CFI\\_mixedData](https://github.com/bips-hb/CFI_mixedData).

## Declarations

**Conflict of interest** MNW is associate editor of AStA—Advances in Statistical Analysis. KB and DSW declare no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References


- Apley, D.W., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models. *J. R. Stat. Soc. Ser. B* **82**(4), 1059–1086 (2020). <https://doi.org/10.1111/rssb.12377>
- Au, Q., Herbringer, J., Stachl, C., Bischl, B., Casalicchio, G.: Grouped feature importance and combined features effect plot. *Data Min. Knowl. Disc.* **36**(4), 1401–1450 (2022). <https://doi.org/10.1007/s10618-022-00840-5>
- Bates, S., Candès, E., Janson, L., Wang, W.: Metropolized knockoff sampling. *J. Am. Stat. Assoc.* **116**(535), 1413–1427 (2021). <https://doi.org/10.1080/01621459.2020.1729163>
- Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
- Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: Model-free knockoffs for high-dimensional controlled variable selection. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **80**, 551–577 (2018). <https://doi.org/10.1111/rssb.12265>
- Chen, H., Janizek, J.D., Lundberg, S., Lee, S.-I.: True to the model or true to the data? ArXiv preprint (2020). <https://doi.org/10.48550/arXiv.2006.16234>

- Covert, I., Lundberg, S.M., Lee, S.-I.: Understanding global feature contributions with additive importance measures. *Adv. Neural Inf. Process. Syst.* **33**, 17212–17223 (2020)
- Fisher, A., Rudin, C., Dominici, F.: All models are wrong, but many are useful: learning a variable's importance by studying an entire class of prediction models simultaneously. *J. Mach. Learn. Res.* **20**(177), 1–81 (2019)
- Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Ann. Stat.* 1189–1232 (2001)
- Glymour, C., Zhang, K., Spirtes, P.: Review of causal discovery methods based on graphical models. *Front. Genetics* (2019). <https://doi.org/10.3389/fgene.2019.00524>
- Gu, J., Yin, G.: Bayesian knockoff filter using gibbs sampler. ArXiv preprint (2021). <https://doi.org/10.48550/arXiv.2102.05223>
- Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H.: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, vol. 2. Springer, New York (2009)
- Hooker, G., Mentch, L., Zhou, S.: Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Stat. Comput.* **31**(6), 1–16 (2021). <https://doi.org/10.1007/s11222-021-10057-z>
- Jordon, J., Yoon, J., van der Schaar, M.: Knockoffgan: generating knockoffs for feature selection using generative adversarial networks. In: *International Conference on Learning Representations* (2019)
- Kormaksson, M., Kelly, L.J., Zhu, X., Haemmerle, S., Pricop, L., Ohlssen, D.: Sequential knockoffs for continuous and categorical predictors: with application to a large psoriatic arthritis clinical trial pool. *Stat. Med.* **40**(14), 3313–3328 (2021). <https://doi.org/10.1002/sim.8955>
- Krishna, S., Han, T., Gu, A., Pombra, J., Jabbari, S., Wu, S., Lakkaraju, H.: The disagreement problem in explainable machine learning: a practitioner's perspective. ArXiv preprint (2022). <https://doi.org/10.48550/arXiv.2202.01602>
- Kursa, M.B., Rudnicki, W.R.: Feature selection with the Boruta package. *J. Stat. Softw.* **36**(11), 1–13 (2010). <https://doi.org/10.18637/jss.v036.i11>
- Lei, J., G'Sell, M., Rinaldo, A., Tibshirani, R.J., Wasserman, L.: Distribution-free predictive inference for regression. *J. Am. Stat. Assoc.* **113**(523), 1094–1111 (2018). <https://doi.org/10.1080/01621459.2017.1307116>
- Linardatos, P., Papastefanopoulos, V., Kotsiantis, S.: Explainable AI: a review of machine learning interpretability methods. *Entropy* (2021). <https://doi.org/10.3390/e23010018>
- Liu, Y., Zheng, C.: Auto-encoding knockoff generator for FDR controlled variable selection. ArXiv preprint (2018). <https://doi.org/10.48550/ARXIV.1809.10765>
- Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **30** (2017)
- Lundberg, S.M., Erion, G., Chen, H., DeGrave, A., Prutkin, J.M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., Lee, S.-I.: From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2**(1), 56–67 (2020). <https://doi.org/10.1038/s42256-019-0138-9>
- Miller, A.M.: *Jewelry from antiquity to the modern era*. In: *Gems and Jewelry Appraising*. Springer, Boston (1988). [https://doi.org/10.1007/978-1-4684-1404-2\\_5](https://doi.org/10.1007/978-1-4684-1404-2_5)
- Molnar, C., König, G., Bischl, B., Casalicchio, G.: Model-agnostic feature importance and effects with dependent features: a conditional subgroup approach. In: *Data Mining and Knowledge Discovery*, pp. 1–39 (2023). <https://doi.org/10.1007/s10618-022-00901-9>
- Pearl, J.: *Causality*. Cambridge University Press, Cambridge (2009). <https://doi.org/10.1017/CBO9780511803161>
- Ribeiro, M.T., Singh, S., Guestrin, C.: “Why should I trust you?” Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016). <https://doi.org/10.1145/2939672.2939778>
- Ribeiro, M.T., Singh, S., Guestrin, C.: Anchors: high-precision model-agnostic explanations. In: *Proceedings of the AAAI Conference on Artificial Intelligence* (2018)
- Rinaldo, A., Wasserman, L., G'Sell, M., Lei, J.: Bootstrapping and sample splitting for high-dimensional, assumption-free inference (2016). <https://doi.org/10.48550/ARXIV.1611.05401>
- Romano, Y., Sesia, M., Candès, E.: Deep knockoffs. *J. Am. Stat. Assoc.* **115**(532), 1861–1872 (2020). <https://doi.org/10.1080/01621459.2019.1660174>
- Sesia, M., Sabatti, C., Candès, E.J.: Gene hunting with hidden Markov model knockoffs. *Biometrika* **106**(1), 1–18 (2018). <https://doi.org/10.1093/biomet/asy033>
- Shah, R.D., Peters, J.: The hardness of conditional independence testing and the generalised covariance measure. *Ann. Stat.* **48**(3), 1514–1538 (2020). <https://doi.org/10.1214/19-AOS1857>

- Shapley, L.: A value for n-Person games. In: Kuhn, H., Tucker, A. (eds.) *Contributions to the Theory of Games II*. Princeton University Press, Princeton (1953). <https://doi.org/10.1515/9781400881970-018>
- Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: *International Conference on Machine Learning*. PMLR (2017)
- Sudarshan, M., Tansey, W., Ranganath, R.: Deep direct likelihood knockoffs. *Adv. Neural Inf. Process. Syst.* **33** (2020)
- Tsamardinos, I., Borboudakis, G.: Permutation testing improves Bayesian network learning. In: *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD* (2010). [https://doi.org/10.1007/978-3-642-15939-8\\_21](https://doi.org/10.1007/978-3-642-15939-8_21)
- Vanschoren, J., Van Rijn, J.N., Bischl, B., Torgo, L.: Openml: networked science in machine learning. *ACM SIGKDD Explorations Newsl.* **15**(2), 49–60 (2014)
- Watson, D.S., Wright, M.N.: Testing conditional independence in supervised learning algorithms. *Mach. Learn.* **110**(8), 2107–2129 (2021). <https://doi.org/10.1007/s10994-021-06030-6>
- Watson, D.S., Blesch, K., Kapar, J., Wright, M. N.: Adversarial random forests for density estimation and generative modeling. In: *Proceedings of the 26th international conference on artificial intelligence and statistics*, PMLR **206** (2023)
- Williamson, B.D., Gilbert, P.B., Carone, M., Simon, N.: Nonparametric variable importance assessment using machine learning techniques. *Biometrics* **77**(1), 9–22 (2021). <https://doi.org/10.1111/biom.13392>
- Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional GAN. *Adv. Neural Inf. Process. Syst.* **32** (2019)
- Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **67**(2), 301–320 (2005). <https://doi.org/10.1111/j.1467-9868.2005.00503.x>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## Authors and Affiliations

Kristin Blesch<sup>1,2</sup>  · David S. Watson<sup>3</sup> · Marvin N. Wright<sup>1,2,4</sup>

David S. Watson  
david.watson@kcl.ac.uk

Marvin N. Wright  
wright@leibniz-bips.de

<sup>1</sup> Leibniz Institute for Prevention Research & Epidemiology - BIPS, Bremen, Germany

<sup>2</sup> Faculty of Mathematics and Computer Science, University of Bremen, Bremen, Germany

<sup>3</sup> Department of Informatics, King's College London, London, UK

<sup>4</sup> Department of Public Health, University of Copenhagen, Copenhagen, Denmark

**Part II.**

**Adversarial Attack Robustness in IML**



## Paper 2. Unfooling SHAP and SAGE: Knockoff Imputation for Shapley Values

**Contributing Article:** Blesch, K., M. N. Wright, and D. S. Watson (2023). Unfooling SHAP and SAGE: Knockoff imputation for Shapley values. In Longo, L. (Ed.) *Explainable Artificial Intelligence. xAI 2023. Communications in Computer and Information Science*, Volume 1901, pp. 131–146. Springer, Cham. [https://doi.org/10.1007/978-3-031-44064-9\\_8](https://doi.org/10.1007/978-3-031-44064-9_8).

**Copyright information:** Copyright 2023 by the authors. Creative Commons Attribution 4.0 International License (CC-BY 4.0).




**Author contributions:** Kristin Blesch developed the project idea, conducted all simulations and experiments, drafted the manuscript and lead the revision process. David S. Watson and Marvin N. Wright supervised the project. All authors interpreted and discussed methodological implications, findings from simulations and experiments and contributed to proofreading and revising the paper.







# Unfooling SHAP and SAGE: Knockoff Imputation for Shapley Values

Kristin Blesch<sup>1,2</sup> , Marvin N. Wright<sup>1,2,3</sup> , and David Watson<sup>4</sup> 

<sup>1</sup> Leibniz Institute for Prevention Research and Epidemiology – BIPS,  
Bremen, Germany  
[blesch@leibniz-bips.de](mailto:blesch@leibniz-bips.de)

<sup>2</sup> Faculty of Mathematics and Computer Science, University of Bremen,  
Bremen, Germany

<sup>3</sup> Department of Public Health, University of Copenhagen, Copenhagen, Denmark

<sup>4</sup> Department of Informatics, King’s College London, London, UK

**Abstract.** Shapley values have achieved great popularity in explainable artificial intelligence. However, with standard sampling methods, resulting feature attributions are susceptible to adversarial attacks. This originates from target function evaluations at extrapolated data points, which are easily detectable and hence, enable models to behave accordingly. In this paper, we introduce a novel strategy for increased robustness against adversarial attacks of both local and global explanations: Knockoff imputed Shapley values. Our approach builds on the model-X knockoff methodology, which generates synthetic data that preserves statistical properties of the original samples. This enables researchers to flexibly choose an appropriate model to generate on-manifold data for the calculation of Shapley values upfront, instead of having to estimate a large number of conditional densities or make strong parametric assumptions. Through real and simulated data experiments, we demonstrate the effectiveness of knockoff imputation against adversarial attacks.

**Keywords:** XAI · Shapley Values · Adversarial Attacks · Knockoffs

## 1 Introduction

Explainable artificial intelligence (XAI) oftentimes strives to deliver insights on the underlying mechanisms of black-box machine learning models in order to generate trust in these algorithms. To do so, XAI methods themselves must be trustworthy.

Several popular XAI tools, such as SHAP [17] and LIME [19], are vulnerable to adversarial attacks [23]. The issue stems from how these approaches generate new data during the explanation process – typically by independently permuting feature values. Permute-and-predict methods force models to extrapolate beyond their training data, yielding off-manifold samples. This results in potentially misleading assessments [13] and enables adversaries to pass fairness audits

© The Author(s) 2023  
L. Longo (Ed.): xAI 2023, CCIS 1901, pp. 131–146, 2023.  
[https://doi.org/10.1007/978-3-031-44064-9\\_8](https://doi.org/10.1007/978-3-031-44064-9_8)

even with discriminatory models. For example, an algorithm could fool the XAI method by using a fair model for queries on synthetic, extrapolated data during XAI evaluation in order to suggest the model would be fair even though it may produce discriminatory outcomes for non-synthetic, i.e. real data [23].

Robustness against such adversarial attacks can be achieved by reducing extrapolation during data generation. Ideally, conditional sampling procedures should be used, which ensures that the generated data is indistinguishable from the original data. Figure 1 visualizes data points generated through marginal in contrast to a conditional sampling method.



**Fig. 1.** Sampling of out-of-coalition features for a digit from  $\{28 \times 28\}$  mnist data. The first 14 columns from the left are in-coalition, whereas the remaining 14 columns are sampled either from marginals (as in Kernel SHAP [17]) or deep knockoffs [21].

For Shapley values [22] – one of the most prominent XAI methods – conditional variants and their properties have been widely discussed in the literature [6, 8, 10, 25, 29]. Conditional Shapley values sample out-of-coalition features from a distribution conditioned on the in-coalition features. However, this requires knowledge about conditional distributions for all possible feature coalitions and, since estimating conditional distributions is generally challenging, there remains considerable room for improvement. However, to prevent adversarial attacks, calculating conditional Shapley values may be unnecessarily challenging. It suffices to minimize extrapolation, which is a strictly simpler task.

In that vein, we propose the model-X knockoff framework [5] in its full generality to sample out-of-coalition features for protection against adversarial attacks on Shapley value explanations. Knockoffs are characterized by two properties, formally defined below: (1) pairwise exchangeability with the original features; and (2) conditional independence of the response, given the true data. We argue that this makes them well-suited to serve as reference data in Shapley value pipelines. For example, property (1) allows us to estimate knockoffs upfront and use them to impute out-of-coalition features, which effectively avoids extrapolation and does not require the separate estimation of conditional distributions for any feature coalition. Knockoff imputed Shapley values balance on-manifold data sampling with maintaining utmost generality and flexibility in application.

The paper is structured as follows. First, we present the relevant background on Shapley values and model-X knockoffs in Sect. 2. We combine these approaches and study the theoretical properties of the resulting algorithm in Sect. 3. In Sect. 4, we present a series of experiments to demonstrate the effec-

tiveness of our approach against adversarial attacks. We present a comprehensive discussion and directions for future research in Sects. 5 and 6, respectively.

## 2 Background and Related Work

### 2.1 Shapley Values

Originating from cooperative game theory, Shapley values [22] aim to attribute payouts fairly amongst a game’s players. The basic idea is to evaluate the average change in output when a player is added to a coalition.

In XAI, we can think of the features  $\mathbf{X} = \{X_1, \dots, X_d\}$ , where each  $X_j$  denotes a random variable, as a set of players  $\mathcal{D} = \{1, \dots, d\}$  who may or may not participate in a coalition of players  $\mathcal{S} \subseteq \mathcal{D}$ , i.e.  $\mathcal{S}$  is a subset of  $\mathcal{D}$ . The value function  $v$  assigns a real-valued payout to each possible coalition, i.e. to every element of the power set of  $\mathcal{D}$ , which consists of  $2^{|\mathcal{D}|} = 2^d$  elements, to a real value. This may be the expected output of a machine learning model  $f$  [17], or other quantities related to the model’s prediction, such as the expected loss [8]. To compute the Shapley value  $\phi_j$  for player  $j$ , we take a weighted average of  $j$ ’s marginal contributions to all subsets that exclude it:

$$\phi_j = \sum_{\mathcal{S} \subseteq \mathcal{D} \setminus \{j\}} \frac{|\mathcal{S}|!(|\mathcal{D}| - |\mathcal{S}| - 1)!}{|\mathcal{D}|!} (v(\mathcal{S} \cup \{j\}) - v(\mathcal{S})). \quad (1)$$

It is not immediately obvious how to evaluate  $v$  on strict subsets of  $\mathcal{D}$ , since  $f$  requires  $d$ -dimensional input. One common solution is to use an expectation with respect to some reference distribution  $\mathcal{R}$ :

$$v(\mathcal{S}) = \mathbb{E}_{\mathcal{R}} [f(\mathbf{x}_{\mathcal{S}}, \mathbf{X}_{\bar{\mathcal{S}}})]. \quad (2)$$

In other words, for the random variables  $\mathbf{X}_{\mathcal{S}}$ , which are the in-coalition features, we take the realized values  $\mathbf{x}_{\mathcal{S}}$  as fixed and sample values for out-of-coalition features  $\mathbf{X}_{\bar{\mathcal{S}}}$  according to  $\mathcal{R}$ . Common choices for  $\mathcal{R}$  include the marginal distribution  $P(\mathbf{X}_{\bar{\mathcal{S}}})$  and the conditional distribution  $P(\mathbf{X}_{\bar{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$ .

*Adversarial Attack Vulnerability.* Taking the marginal distribution  $\mathcal{R} = P(\mathbf{X}_{\bar{\mathcal{S}}})$  typically serves as an approximation to the conditional distribution  $P(\mathbf{X}_{\bar{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$  in order to facilitate computation, e.g. as in KernelSHAP [17]. However, marginal and conditional distributions only coincide when features are jointly independent, which is scarcely ever the case in empirical applications. A consequence from a violation of feature independence is that sampling a set of  $\mathbf{x}'_{\bar{\mathcal{S}}}$  from marginals instead of conditional distributions will lead to generated instances  $\mathbf{x}' = (\mathbf{x}_{\mathcal{S}}, \mathbf{x}'_{\bar{\mathcal{S}}})$  that are off the data manifold of original, i.e. real data observations  $\mathbf{x} = (\mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\bar{\mathcal{S}}})$ . In such cases, it is possible to train a prediction model  $\omega$  that successfully distinguishes real from generated data. In adversarial explanations, e.g. the strategy described by [23], such an out-of-distribution (OOD) detector  $\omega$  that exposes synthetic data is the primary workhorse. If the data

is synthetic, the adversary deploys a different model as with real data, which effectively fools the explanation.

We want to highlight that even though this fooling strategy was introduced and is typically discussed for local Shapley values [23, 28], it can also be applied to global aggregates such as Shapley additive global importance (SAGE) [8].

*Achieving Adversarial Attack Robustness.* Avoiding the generation of extrapolated data protects against adversarial attacks by preventing  $\omega$  from distinguishing real from generated data during Shapley value calculation.

Some approaches naturally circumvent the task of generating synthetic data altogether, for example by using surrogate models [10], retraining the model such that it adapts to missing features [8] or fitting a separate model for each coalition [25, 29]. However, these approaches come at a high computational costs, since repeated model refitting is required.

Another approach is to calculate conditional Shapley values, for which we will give a brief overview of methods in the following paragraph. Working with conditional Shapley values, i.e. using  $\mathcal{R} = P(\mathbf{X}_{\mathcal{S}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$ , is clearly the most rigorous way of enforcing on-manifold sampling of synthetic data, even though prior literature merely acknowledges the potential for preventing adversarial attacks. Several conditional Shapley value estimation procedures have been proposed, yet conditional feature sampling remains a challenging task and improvements are highly desirable.

A straightforward, empirical approach is to simply use the observed data that naturally satisfies the conditioning on the selected in-coalition features by using data points in close proximity to the instance to be explained [1, 11]. For example, in Fig. 1, one could also sample the out-of-coalition features from other observations of digit zero in the data set. This approach, however, has the downside that the number of observations fulfilling the conditioning event might be very small, leading to only very few or even no appropriate observations available. Another approach to calculating conditional Shapley values is assuming a specific data distribution, e.g. a Gaussian distribution [1, 7], for which conditional distributions are easy to derive, but this approach has the drawback of strong assumptions on the data generating process. Further, conditional generative models might be used [10, 20], however, these models might be challenging to train and it is unclear whether they truly approximate the data well. In sum, conditional Shapley values are challenging to access and hence have limited applicability.

For the goal of preventing adversarial attacks, conditional Shapley values are sufficient but not necessary, since any method that avoids extrapolation will prevent the attack and hence related, but less strict frameworks provide another suite of promising methods. Such an idea is pursued by [28], where generative models use ‘focused sampling’ of new instances that are close to the original observations. However, this approach lacks theoretical guarantees and may fail depending on the fit of the generative models. We acknowledge that [28] investigate Gaussian knockoffs in conjunction with the so-called Interactions-based Method for Explanation (IME, [24]). However, the authors do not use model-X

knockoffs for imputation in full generality, nor do they apply the strategy to SHAP or SAGE values directly. The authors even mention that the knockoff imputation idea merits further investigation as an approach, which is what the present paper contributes to.

## 2.2 Model-X Knockoffs

The model-X knockoff framework [5] is a theoretically sound concept to characterize synthetic variables with specific statistical properties. Intuitively speaking, knockoffs are synthetic variables that aim to copy the statistical properties of a given set of original variables, e.g. the covariance structure, such that they are indistinguishable from the original variables when the target variable  $Y$  is not looked at. Crucially, valid knockoffs ensure that original variables can be swapped with their knockoff counterparts without affecting the joint distribution.

Formally, in order for  $\tilde{\mathbf{X}}$  to be a valid knockoff matrix for  $\mathbf{X}$ , two conditions have to be met:

1. Pairwise exchangeability: For any proper subset  $\mathcal{S} \subset \{1, \dots, d\}$ :

$$(\mathbf{X}, \tilde{\mathbf{X}})_{\text{swap}(\mathcal{S})} \stackrel{d}{=} (\mathbf{X}, \tilde{\mathbf{X}}), \quad (3)$$

where  $\stackrel{d}{=}$  represents equality in distribution and  $\text{swap}(\mathcal{S})$  indicates swapping the variables in  $\mathcal{S}$  with their knockoff counterparts.

2. Conditional independence:

$$\tilde{\mathbf{X}} \perp\!\!\!\perp Y \mid \mathbf{X}. \quad (4)$$

Generating valid knockoffs is an active field of research and various sampling algorithms have been proposed, which ensures that practitioners can flexibly choose appropriate algorithms. For example, there are algorithms based on distributional assumptions [3, 5, 21], Bayesian statistics [12] or deep learning [14, 16, 18, 26].

## 3 Combining Model-X Knockoffs with Shapley Values

This paper proposes to impute out-of-coalition features with model-X knockoffs for the calculation of Shapley value based quantities. Knockoffs come with strong theoretical guarantees that ensure avoiding extrapolation. Moreover, they provide a major computational boost, since knockoffs can be sampled upfront for the full data matrix instead of requiring separate models for each possible coalition. Since many methods are available for knockoff generation—including some that are essentially tuning-free—practitioners have a large collection of tools available for valid, flexible and convenient sampling of the out-of-coalition space that ensures robustness against adversarial attacks.

In detail, we propose Algorithm 1 to impute out-of-coalition features with knockoffs for Shapley values and Algorithm 2 (see Appendix A) for knockoff imputation with SAGE [8] values. In brief, the algorithms use  $N_{ko}$  knockoffs as the background distribution in the calculation of Shapley values. Note that for  $N_{ko} = 1$ , the Shapley values are with respect to a single knockoff baseline value, while for larger values of  $N_{ko}$ , Shapley values explain the difference between the selected instance and the expected value of the knockoff distribution.

---

**Algorithm 1.** Knockoff Imputed Shapley Values
 

---

**Input:** data matrix  $(\mathbf{X}, Y)$ , knockoff sampler  $ko()$ , model  $f$ , explanation instance  $\mathbf{x}^0 = \{x_1^0, \dots, x_d^0\}$ , number of knockoffs  $N_{ko}$ , power set  $\pi$  of  $\mathcal{D} \setminus \{j\}$

- 1: train knockoff sampler  $ko(\mathbf{X})$
- 2: **for** feature  $j$  in  $\mathcal{D}$  **do**
- 3:   initialize  $\phi_j = 0$
- 4:   **for**  $i$  in  $N_{ko}$  **do**
- 5:     draw  $\tilde{\mathbf{x}}^i = \{\tilde{x}_1^i, \dots, \tilde{x}_d^i\}$  from  $ko(\mathbf{X})$
- 6:     initialize  $\Delta_j^i = 0$
- 7:     **for**  $\mathcal{S}$  in  $\pi$  **do**
- 8:       out-of-coalition set  $\bar{\mathcal{S}} = \mathcal{D} \setminus \mathcal{S}$
- 9:        $v(\mathcal{S}) = f(\mathbf{x}_{\mathcal{S}}^0, \tilde{\mathbf{x}}_{\bar{\mathcal{S}}}^i)$
- 10:        $\mathcal{S}' = \mathcal{S} \cup \{j\}$
- 11:        $\mathcal{S}' = \mathcal{S} \setminus \{j\}$
- 12:        $v(\mathcal{S}') = f(\mathbf{x}_{\mathcal{S}'}^0, \tilde{\mathbf{x}}_{\bar{\mathcal{S}'}}^i)$
- 13:        $\Delta_j^i = \Delta_j^i + \frac{|\mathcal{S}'|!(|\mathcal{D}|-|\mathcal{S}'|-1)!}{|\mathcal{D}|!} \cdot (v(\mathcal{S}') - v(\mathcal{S}))$
- 14:     **end for**
- 15:   **end for**
- 16:    $\phi_j = \frac{1}{N_{ko}} \sum_{i=1}^{N_{ko}} \Delta_j^i$
- 17: **end for**
- 18: **return** Shapley values  $\phi = \{\phi_1, \dots, \phi_d\}$

---

To understand the advantages of knockoff imputed Shapley values on a theoretical level, let us investigate the implications of the exchangeability property (Eq. 3) in more depth. This property ensures that we can swap *any* set  $\mathcal{S} \subseteq \mathcal{D}$  of original variables  $\mathbf{X}$  with knockoffs  $\tilde{\mathbf{X}}$ , while maintaining the same joint distribution. The same joint distribution guarantees that any generated data is indeed on the same data manifold, so for the prevention of adversarial attacks, it is both necessary and sufficient that  $\mathbf{x}'_{\bar{\mathcal{S}}}$  is generated such that  $P(\mathbf{X}_{\mathcal{S}}, \mathbf{X}_{\bar{\mathcal{S}}}) = P(\mathbf{X}_{\mathcal{S}}, \mathbf{X}'_{\bar{\mathcal{S}}})$ . Conditional Shapley values ensure this by sampling  $\mathbf{x}'_{\bar{\mathcal{S}}}$  from  $P(\mathbf{X}_{\bar{\mathcal{S}}}|\mathbf{X}_{\mathcal{S}})$ . Doing so, the original joint distribution is maintained by factorizing through  $P(\mathbf{X}_{\mathcal{S}}) \cdot P(\mathbf{X}_{\bar{\mathcal{S}}}|\mathbf{X}_{\mathcal{S}}) = P(\mathbf{X}_{\mathcal{S}}, \mathbf{X}_{\bar{\mathcal{S}}})$ , whereas knockoffs directly guarantee  $P(\mathbf{X}_{\mathcal{S}}, \mathbf{X}_{\bar{\mathcal{S}}}) = P(\mathbf{X}_{\mathcal{S}}, \mathbf{X}'_{\bar{\mathcal{S}}})$  by exchangeability.

That said, it becomes obvious that we can generate knockoff copies for  $\mathbf{X}$  upfront and then swap in knockoffs for the out-of-coalition features  $\mathbf{X}_{\bar{\mathcal{S}}}$  where needed. This is a clear advantage in contrast to conditional Shapley value methods that need access to  $P(\mathbf{X}_{\bar{\mathcal{S}}}|\mathbf{X}_{\mathcal{S}})$  for all possible coalitions  $2^{|\mathcal{D}|}$ . Note that

the pairwise exchangeability fulfilled by knockoffs is needed to guarantee on-manifold data in the imputation step, which is why other conditional sampling methods cannot be calculated upfront. This suggests a lower computational complexity for the knockoff imputed Shapley values in comparison to conditional Shapley values, however, the exact complexity will depend on the knockoff generating algorithm used. Further, even though we may want to sample  $N_{ko}$  knockoffs in advance to reduce bias, a reasonable number for  $N_{ko}$  is typically  $N_{ko} \ll 2^{|D|}$ .

However, the benefit of being able to sample knockoffs upfront comes at the cost of enforcing a restrictive set of conditioning events. At a first glance, knockoff imputation and calculating conditional Shapley values, i.e. using  $\mathcal{R} = P(\mathbf{X}_{\mathcal{S}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}})$ , may appear interchangeable. However, knockoffs implicitly condition on all the feature values of the observation, which is inevitable since the exchangeability property must hold for *any* set of variables. This subtle difference yields the following expression for the game that uses knockoffs  $\tilde{\mathbf{X}}_{\mathcal{S}}$  as imputation for the out-of-coalition features in set  $\bar{\mathcal{S}}$ :

$$v_{ko}(\mathcal{S}) = \mathbb{E}_{p(\tilde{\mathbf{X}}_{\bar{\mathcal{S}}} | \mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\bar{\mathcal{S}}})} \left[ f(\mathbf{x}_{\mathcal{S}}, \tilde{\mathbf{X}}_{\bar{\mathcal{S}}}) \right]. \quad (5)$$

To elaborate on the consequences of the expectation taken w.r.t.  $P(\tilde{\mathbf{X}}_{\bar{\mathcal{S}}} | \mathbf{X}_{\mathcal{S}} = \mathbf{x}_{\mathcal{S}}, \mathbf{X}_{\bar{\mathcal{S}}} = \mathbf{x}_{\bar{\mathcal{S}}})$ , imagine a dataset with three variables, i.e.  $X_1, X_2, X_3$ , where  $X_1$  is in-coalition and the task is to impute values for the out-of-coalition features  $X_2$  and  $X_3$ . When using knockoff  $\tilde{X}_2$  for imputation, this knockoff has been generated from a knockoff sampler that was fitted on the observed values of all three variables in the dataset. For the Shapley value calculation however, the data for imputation is required to condition on the observed value of  $X_1$  only. Hence, the procedure leaks information from the out-of-coalition feature  $X_3$  during the imputation of  $X_2$ . As a consequence, the range of values sampled for imputing out-of-coalition values will be too narrow, i.e. conditioned on more features than necessary, which reduces the entropy of the predicted values in  $f(\mathbf{x}_{\mathcal{S}}, \mathbf{x}_{\bar{\mathcal{S}}})$ . To be clear, the generated data throughout Shapley value calculation will still be on the same data manifold as the original samples, so this procedure effectively defends against adversarial attacks, which is what we focus on in this paper. We can interpret the restrictive conditioning as a form of regularization imposed through the data sampling mechanism. We therefore expect estimated Shapley values of lower magnitude when using knockoff imputation. As a result, on the one hand, conditioning on variables in the out-of-coalition set may introduce bias due to information leakage from other covariates; on the other hand, this will also lead to a reduction in variance of Shapley values that are estimated by approximation instead of exact calculation, which may be advantageous. We encourage future research to investigate potential trade-offs.

## 4 Experiments

### 4.1 Unfooling SHAP

We start off the section on experiments by illustrating that knockoff imputed Shapley values indeed are able to prevent adversarial attacks that make use of extrapolation. We replicate and extend the German Credit [9] experiments conducted by [23], where the task is to determine whether clients will be good customers (`GoodCustomer` = 1) or not (`GoodCustomer` = -1). We demonstrate that with knockoff imputation, the adversarial attack is no longer successful.

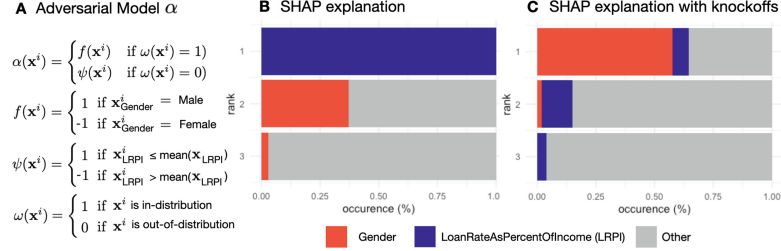
The fooling strategy of [23] can be illustrated in an intuitive manner: Imagine an unethical bank that aims to exploit sensitive information, such as the `Gender` of the credit applicant, to assess creditworthiness with a prediction model  $f$ . However, in the country the bank operates in, credit assessment tools are legally required to pass a so-called fairness audit prior to model deployment. During the audit, black-box models will be evaluated via SHAP [17]. If high importance scores are attributed to sensitive features, the model fails the audit and its deployment is legally prohibited. In spite of this danger, the bank is determined to use the discriminatory model  $f$  in its daily business, and comes up with a strategy to fool the auditors and make them believe that the bank uses an innocent prediction algorithm  $\psi$ , which utilises the non-discriminatory feature `LoanRateAsPercentOfIncome` only.

To fool the explanation, a supervised classification algorithm  $\omega$ , in this case a random forest [4], is trained to distinguish in-distribution from OOD data. For in-distribution data, the adversarial model  $\alpha$  uses the discriminatory prediction algorithm  $f$ , but for OOD data,  $\alpha$  makes predictions according to  $\psi$ . For simplicity, the prediction model  $f$  here makes use of only the discriminatory variable `Gender`, whereas prediction model  $\psi$  determines the predictions according to the non-discriminatory variable `LoanRateAsPercentOfIncome`, see further Fig. 2, Panel A.

Extending the experiments of [23], we evaluate the adversarial model  $\alpha$  not only with SHAP using marginal imputation with `kmeans(10)` to summarize the background distribution, but also with knockoff imputed SHAP. We use sequential knockoffs [15] due to the presence of both categorical and continuous features in the dataset and generate  $N_{ko} = 10$  knockoffs for each test data point. In brief, sequential knockoffs [15] are generated through an iterative procedure that samples values from a Gaussian (for continuous features) or multinomial (for categorical features) distribution. The distribution parameters are estimated from a penalized linear or multinomial logistic regression of the feature for which a knockoff copy should be generated on the other features and already sampled knockoffs from earlier iterations.

Figure 2 Panel B replicates results of [23], where we can see the aggregated explanations across test data instances suggesting `LoanRateAsPercentOfIncome` as most relevant for the model, hence the adversarial attack is successful. On the contrary, we can see from Panel C that with knockoff imputation, the explanation reveals the true importance of `Gender` for the model’s prediction in the majority of test data points.





**Fig. 2.** Occurrences of features ranked amongst the top 3 features in SHAP explanations of test data,  $N = 99$ . Fidelity to detect real data as in-distribution with  $\omega$  is 1.

Clearly,  $\omega$  plays a crucial role for the success of the adversarial attack. In this experiment, we see that with knockoff imputation,  $\omega$  is not able to detect OOD data as such, which is why the adversarial attack cannot unfold. At the same time, we have that  $\omega$  always detects in-distribution data as such (fidelity = 1), so real world data will be reliably predicted with discriminatory model  $f$ . With this knowledge, advanced adversarial attackers might try to improve the model training of  $\omega$  and optimize the decision threshold to reinstate a successful adversarial attack. However, with knockoff imputation – and conditional out-of-coalition-sampling in general – it is not possible to train an effective OOD classifier  $\omega$  because the resulting data will lie on the same data manifold as the original data, hindering its detection as synthetic data. We discuss this issue further in Appendix B.

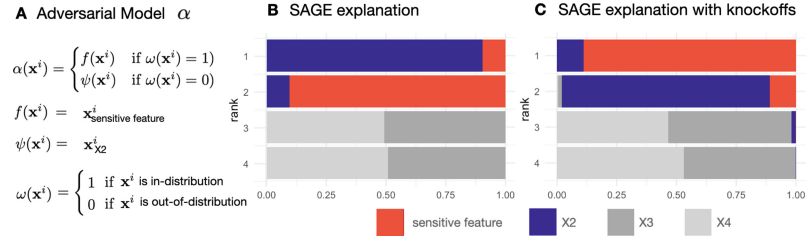
## 4.2 Unfooling SAGE

We now illustrate that global aggregates of Shapley values, SAGE values [8], suffer from the same vulnerability as local Shapley values and that knockoff imputation again can increase robustness. In this experiment, we simulate data, which further allows us to analyze key drivers in the data characteristics that affect the robustness against adversarial attacks.

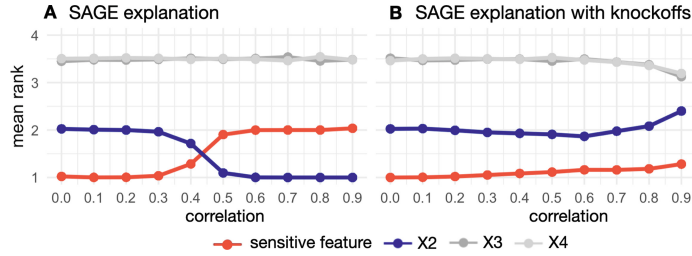
We simulate data from  $\mathbf{X}^{1000 \times 4} \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ , with  $\boldsymbol{\mu} = [0, \dots, 0]$  and correlation matrix  $\boldsymbol{\Sigma}$  with correlation coefficient  $\rho \in [0, 0.9]$  in all off-diagonal elements. The target variable  $Y$  is defined by  $Y = \mathbf{X}\boldsymbol{\beta} + \epsilon$ , where  $\boldsymbol{\beta} = [1, \dots, 1]$  and  $\epsilon \sim N(0, \sigma^2)$ , such that  $\sigma^2$  guarantees a signal to noise ratio  $SNR = \frac{\text{Var}(Y)}{\sigma^2} = 2$ . The adversarial model  $\alpha$  is given in Fig. 3 Panel A.

SAGE with marginal imputation uses a background distribution summarized by  $kmeans(10)$  and SAGE with knockoff imputation uses  $N_{ko} = 10$  Gaussian knockoffs for the background distribution of each observation. Gaussian knockoffs are generated through sampling from a multivariate Gaussian distribution with parameters estimated from the observed features, see [5] for further details.

Figure 3 reveals that SAGE can be fooled just as SHAP and that knockoff imputation again prevents the adversarial attack, i.e. unfools the explanation.



**Fig. 3.** Occurrences of features ranked amongst the top 4 in SAGE explanations with adversarial model  $\alpha$  across 500 replicates. Correlation  $\rho = 0.5$ , fidelity of  $\omega = 0.88$ .



**Fig. 4.** Mean rank of features in SAGE explanations with adversarial classifier  $\alpha$  across 500 replicates. We consider the adversarial attack successful if  $X_2$  is ranked first, and unsuccessful if the sensitive feature is ranked first.

We can see from Fig. 4 that correlation strength plays a crucial role in the success of the attack. For marginal imputation (Fig. 4, Panel A), the adversarial attack is successful and hides the true rank of the sensitive feature (true rank is 1) for correlation levels greater than or equal to 0.5. Note that lower levels of correlation do not allow the adversarial attack to unfold regardless, because extrapolation is more pronounced and hence more likely to be detected by  $\omega$  with medium or strongly correlated data (recall that  $\rho = 0$  corresponds to feature independence). For SAGE with knockoffs (Fig. 4, Panel B), the model’s usage of the sensitive feature is exposed for all levels of correlation. Hence, knockoff imputation effectively prevents the adversarial attack on SAGE explanations.

## 5 Discussion

We demonstrate that knockoff imputed Shapley values are robust against adversarial attacks that exploit extrapolated data. However, other adversarial attacks might be proposed. For example, because Shapley values are spread out across correlated features, the true importance of a sensitive feature could be toned down by adding correlated features to the model.

Further, the special characteristics of knockoffs may open up new trajectories in Shapley value research. One such example is SHAPLIT, which proposes con-

ditional independence testing with FDR control for Shapley values [27]. Another promising approach could be to leverage the overly restrictive conditioning of knockoff imputed Shapley values for approximation tasks, where Shapley values are calculated with just a small fraction of all possible coalitions as opposed to exact Shapley value calculation. It is common in Shapley value software to optionally include some form of  $L_1$  penalty on feature attributions to encourage sparse explanations, even when the underlying model  $f$  is not itself sparse [17]. Like many regularization methods, this effectively introduces bias in exchange for a decrease in variance. Knockoff imputed Shapley values may give a similar regularizing effect through the data sampling method rather than directly on the parameter estimation technique. This does not zero out feature attributions as the  $L_1$  penalty does, but may serve to improve predictions for practitioners with limited computational budgets.

We want to emphasize that the use case for knockoff imputed Shapley values should be carefully chosen, since the method narrows down entropy of the target function, which may be disadvantageous in comparison to other methods when the computational capacity suffices to calculate exact conditional Shapley values.

Further, we want to highlight that a comparative benchmark study that analyzes variants for Shapley value calculation, including conditional Shapley value calculation, may be of great value for future research. For example, the knockoff-based approach proposed here could be compared with other conditional variants [1, 2, 20] both in terms of theory, e.g. analyzing the variance, and in empirical application, e.g. investigating the computational efficiency of the proposed algorithms and accuracy of estimates for different datasets. Such endeavors may further include novel methods that combine ideas from existing approaches. For example, one could use an overly strict conditioning set, as it is the case with knockoffs, for the conditional distribution based approaches to cut down the computational complexity of those approaches.

## 6 Conclusion

The paper presents an innovative approach to make Shapley explanations, such as SHAP [17] and SAGE [8], more robust against adversarial attacks by using model-X knockoffs. The discussion on theoretical guarantees and implications reveals that knockoffs can serve as a flexible and off-the-shelf methodology that effectively prevents extrapolation during Shapley value calculation. Through both real data and simulated data experiments, the paper demonstrates that vulnerability to adversarial attacks can be successfully reduced. It is worth emphasizing that the presented methodology can be used in conjunction with any valid knockoff sampling procedure and not only the deep [18], sequential [15] and Gaussian knockoffs [5] used in this paper, which further highlights the flexibility of the proposed approach. This, and the possibility to sample knockoffs upfront, which drastically reduces computational complexity, is a major advantage over conditional Shapley value calculation approaches that may otherwise be used for the prevention of adversarial attacks.

**Acknowledgements.** MNW and KB received funding for this project from the German Research Foundation (DFG), Emmy Noether Grant 437611051.

**Data and Code availability.** Reproducible code for the results presented in this paper is available at [https://github.com/bips-hb/unfooling\\_shapley](https://github.com/bips-hb/unfooling_shapley).

## A Knockoff Imputed SAGE Values

---

**Algorithm 2.** Sampling-based approximation for SAGE values [8] with knockoff imputation

---

**Input:** data  $(\mathbf{X}, Y)$ , model  $f$ , loss function  $l$ , outer samples  $n$ , number of knockoffs  $N_{ko}$ , knockoff sampler  $ko()$

- 1: Initialize  $\hat{\phi}_1 = 0, \hat{\phi}_2 = 0, \dots, \hat{\phi}_d = 0$
- 2:  $\hat{y}_{init} = \frac{1}{N} \sum_{i=1}^N f(\mathbf{x}_i)$
- 3: train knockoff sampler  $k(\mathbf{X})$
- 4: **for**  $i = 1$  to  $n$  **do**
- 5:   Sample a data instance  $(\mathbf{x}_i, y_i)$
- 6:   Sample instance  $\pi$ , a permutation of  $\mathcal{D}$
- 7:    $\mathcal{S} = \emptyset$
- 8:    $L_{prev} = l(\hat{y}_{init}, y)$
- 9:   **for**  $j$  in  $\mathcal{D}$  **do**
- 10:      $\mathcal{S} = \mathcal{S} \cup \{\pi[j]\}$
- 11:      $\hat{y} = 0$
- 12:     **for**  $k = 1$  to  $N_{ko}$  **do**
- 13:       Sample  $\tilde{\mathbf{x}}^k = \tilde{x}_1^k, \dots, \tilde{x}_d^k$  from  $ko(\mathbf{X})$
- 14:        $\hat{y} = \hat{y} + f(\mathbf{x}_{\mathcal{S}}, \tilde{\mathbf{x}}_{\mathcal{S}}^k)$
- 15:     **end for**
- 16:      $\tilde{\hat{y}} = \frac{\hat{y}}{N_{ko}}$
- 17:      $L = l(\tilde{\hat{y}}, y)$
- 18:      $\Delta = L_{prev} - L$
- 19:      $\hat{\phi}_{\pi[j]} = \hat{\phi}_{\pi[j]} + \Delta$
- 20:      $L_{prev} = L$
- 21:   **end for**
- 22: **end for**
- 23: **return** SAGE values  $\frac{\hat{\phi}_1}{n}, \dots, \frac{\hat{\phi}_d}{n}$

---

## B The Role of $\omega$

The detection of OOD data is the backbone of adversarial attacks which is why the performance of  $\omega$  is crucial for the success of the attack. Let us take a closer look onto how  $\omega$  is derived: In a first step, we take the training data and add the label  $y = 1$  to it, because it is real world data. Then, we mimic the

imputation procedure that occurs during Shapley value calculation. This means we treat some feature values as in-coalition, i.e. we take the actual value of these features for the respective observation, and impute the remaining out-of-coalition feature values with other quantities, for example, by sampling values from the marginal distribution. The resulting 'new' observations will then be labeled with  $y = 0$ , because this is generated data. In a second step, we fit a random forest classifier to distinguish in-distribution from OOD data, or technically speaking, to distinguish  $y = 1$  from  $y = 0$  data.

In the experiments of Sect. 4.1, we train  $\omega$  exactly like [23], where we impute the out-of-coalition features with values from a background distribution that is characterised by the center points of a  $kmeans(c = 10)$  clustering of the data. Note that this is the exact same background distribution which is used for the marginal imputation in the SHAP explanations later on. With this  $\omega$ , the adversarial attack is successful for marginal imputation, but with the very same  $\omega$ , SHAP explanations using knockoff imputation prevents the attack.

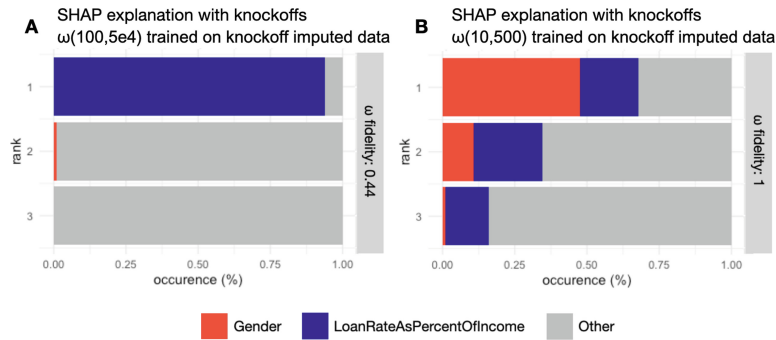
To some readers, it might have been clear from the beginning that an  $\omega$  which is trained to detect OOD samples that are generated with  $kmeans(c = 10)$  will not perform well in detecting OOD samples that occur when calculated knockoff imputed SHAP, where the background distribution consists of knockoffs and not  $kmeans(c = 10)$  cluster centers. An advanced adversarial attacker strategy could be to train  $\omega$  for detecting OOD samples from many different background distributions, including a knockoff background distribution. However, the crucial point is that with conditional out-of-coalition-sampling, for example knockoffs, it is not possible to train an effective OOD classifier  $\omega$  because the  $y = 1$  and  $y = 0$  data points will lie on the same data manifold. In other words, there is no difference in in-distribution and OOD data, which hinders its detection as synthetic data. When training  $\omega$  on such data, the classifier cannot learn reasonable information from the data.

The implications of this are illustrated in Fig. 5. There, we train  $\omega$  on data that is generated by knockoff imputation. We vary the hyperparameters of the random forest classifier to force model  $\omega$  to overfit, i.e. be less (Fig. 5, Panel **A**) or more (Fig. 5, Panel **B**) conservative in predicting data as OOD. This can be achieved by varying the number of trees in the random forest classifier, and the number of  $y = 0$  training samples we generate. We denote the hyperparameters with  $\omega(\text{number of trees, number of samples generated})$ .

For an adversarial attacker, the aim is high fidelity, i.e. a high percentage of true in-distribution classifications by  $\omega$  and a high rank of the innocent feature `LoanRateAsPercentOfIncome` in the SHAP explanation. Different hyperparameter settings reveal that there is a trade-off between fidelity and the degree to which the innocent feature `LoanRateAsPercentOfIncome` is ranked up high. If the adversarial attacker is keen to predict real-world data with the discriminatory model, i.e. uses an  $\omega$  that is conservative in classifying data as OOD, then knockoff imputed SHAP reveals the sensitive feature `Gender` as highly important (Fig. 5, Panel **B**). On the contrary, if the adversarial attacker prioritizes that the explanation should pretend that `LoanRateAsPercentOfIncome` is important, i.e.

uses an  $\omega$  that is liberal in predicting data as OOD, then the fidelity of  $\omega$  drops drastically (Fig. 5, Panel A). This is clearly in contrast to the overarching goal of adversarial attackers to use the discriminatory model for in-distribution (real world) applications, but fool the SHAP explanation such that the model appears innocent.

Consequently, when using knockoff imputed SHAP, the adversarial attacker is forced to use the fair model if the SHAP evaluation should suggest that the model is fair – in other words and recollecting the example stated in the main text before: The only way to pass a fairness audit that uses knockoff imputed SHAP explanations is using a fair model.



**Fig. 5.** Occurrences of features ranked amongst the top 3 features in SHAP explanations of  $N = 99$  test data points.

## References

1. Aas, K., Jullum, M., Løland, A.: Explaining individual predictions when features are dependent: more accurate approximations to Shapley values. *Artif. Intell.* **298**, 103502 (2021)
2. Aas, K., Nagler, T., Jullum, M., Løland, A.: Explaining predictive models using Shapley values and non-parametric vine copulas. *Depend. Model.* **9**(1), 62–81 (2021)
3. Bates, S., Candès, E., Janson, L., Wang, W.: Metropolized knockoff sampling. *J. Am. Stat. Assoc.* **116**(535), 1413–1427 (2021)
4. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001)
5. Candès, E., Fan, Y., Janson, L., Lv, J.: Panning for gold: model-free knockoffs for high-dimensional controlled variable selection. *J. Roy. Stat. Soc. Ser. B (Stat. Methodol.)* **80**(3), 551–577 (2018)
6. Chen, H., Covert, I.C., Lundberg, S.M., Lee, S.I.: Algorithms to estimate Shapley value feature attributions. [arXiv:2207.07605](https://arxiv.org/abs/2207.07605) (2022)
7. Chen, H., Janizek, J.D., Lundberg, S., Lee, S.I.: True to the model or true to the data? [arXiv:2006.16234](https://arxiv.org/abs/2006.16234) (2020)

8. Covert, I., Lundberg, S.M., Lee, S.I.: Understanding global feature contributions with additive importance measures. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 17212–17223 (2020)
9. Dua, D., Graff, C.: UCI machine learning repository (2017)
10. Frye, C., de Mijolla, D., Begley, T., Cowton, L., Stanley, M., Feige, I.: Shapley explainability on the data manifold. [arXiv:2006.01272](https://arxiv.org/abs/2006.01272) (2020)
11. Ghalebikesabi, S., Ter-Minassian, L., DiazOrdaz, K., Holmes, C.C.: On locality of local explanation models. In: *Advances in Neural Information Processing Systems*, vol. 34, pp. 18395–18407 (2021)
12. Gu, J., Yin, G.: Bayesian knockoff filter using Gibbs sampler. [arXiv:2102.05223](https://arxiv.org/abs/2102.05223) (2021)
13. Hooker, G., Mentch, L., Zhou, S.: Unrestricted permutation forces extrapolation: variable importance requires at least one more model, or there is no free variable importance. *Stat. Comput.* **31**(6), 1–16 (2021)
14. Jordon, J., Yoon, J., van der Schaar, M.: KnockoffGAN: generating knockoffs for feature selection using generative adversarial networks. In: *International Conference on Learning Representations* (2019)
15. Kormaksson, M., Kelly, L.J., Zhu, X., Haemmerle, S., Pricop, L., Ohlssen, D.: Sequential knockoffs for continuous and categorical predictors: With application to a large psoriatic arthritis clinical trial pool. *Stat. Med.* **40**(14), 3313–3328 (2021)
16. Liu, Y., Zheng, C.: Auto-encoding knockoff generator for FDR controlled variable selection. [arXiv:1809.10765](https://arxiv.org/abs/1809.10765) (2018)
17. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
18. Romano, Y., Sesia, M., Candès, E.: Deep knockoffs. *J. Am. Stat. Assoc.* **115**(532), 1861–1872 (2020)
19. Ribeiro, M. T., Singh, S., Guestrin, C.: Why should I trust you? Explaining the predictions of any classifier. In: *Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining ACM SIGKDD 22*, pp. 1135–1144 (2016)
20. Redelmeier, A., Jullum, M., Aas, K.: Explaining predictive models with mixed features using Shapley values and conditional inference trees. In: *Proceedings of the 4th International Cross-Domain Conference for Machine Learning and Knowledge Extraction CD-MAKE*, pp. 117–137 (2020)
21. Sesia, M., Sabatti, C., Candès, E.J.: Gene hunting with hidden Markov model knockoffs. *Biometrika* **106**(1), 1–18 (2018)
22. Shapley, L.: A value for n-person games. In: Kuhn, H., Tucker, A. (eds.) *Contributions to the Theory of Games II*. Princeton University Press, Princeton (1953)
23. Slack, D., Hilgard, S., Jia, E., Singh, S., Lakkaraju, H.: Fooling LIME and SHAP: adversarial attacks on post hoc explanation methods. In: *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186 (2020)
24. Štrumbelj, E., Kononenko, I.: An efficient explanation of individual classifications using game theory. *J. Mach. Learn. Res.* **11**, 1–18 (2010)
25. Štrumbelj, E., Kononenko, I.: Explaining prediction models and individual predictions with feature contributions. *Knowl. Inf. Syst.* **41**(3), 647–665 (2014)
26. Sudarshan, M., Tansey, W., Ranganath, R.: Deep direct likelihood knockoffs. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 5036–5046 (2020)

27. Teneggi, J., Bharti, B., Romano, Y., Sulam, J.: From Shapley back to Pearson: hypothesis testing via the Shapley value. [arXiv:2207.07038](https://arxiv.org/abs/2207.07038) (2022)
28. Vreš, D., Robnik-Šikonja, M.: Preventing deception with explanation methods using focused sampling. *Data Mining Knowl. Discov.* (2022)
29. Williamson, B., Feng, J.: Efficient nonparametric statistical inference on population feature importance using Shapley values. In: *International Conference on Machine Learning*, pp. 10282–10291. PMLR (2020)

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.





**Part III.**

**Generative Modeling with Mixed  
Tabular Data**



# Paper 3. Adversarial Random Forests for Density Estimation and Generative Modeling

**Contributing Article:** Watson, D. S., Blesch, K., Kapar, J., and Wright, M. N. (2023) Adversarial random forests for density estimation and generative modeling. In *Proceedings of the 26<sup>th</sup> International Conference on Artificial Intelligence and Statistics*, pp. 5357-5375. PMLR. <https://proceedings.mlr.press/v206/watson23a.html>.

**Copyright information:** Copyright 2023 by the authors and PMLR. Creative Commons Attribution 4.0 International License (CC-BY 4.0).

**Author contributions:** The project idea was derived from previous work of Kristin Blesch, David S. Watson and Marvin N. Wright and further developed and refined by all authors. David S. Watson lead the project and its methodological development and drafted the manuscript. Kristin Blesch conducted the experiments for FORGE on real data (Section 5.2) and runtime analyses (Section 5.3), contributing to writing the Sections 5.2 and 5.3 related to this and producing the respective Figures and Tables. David S. Watson and Marvin N. Wright conducted simulations for FORDE (Section 5.1 and 5.2). David S. Watson and Jan Kapar derived theoretical contributions of the method, i.e., proofs. All authors discussed and interpreted methodological implications, findings from simulations and experiments and contributed to proofreading and revising the paper.



---

# Adversarial Random Forests for Density Estimation and Generative Modeling

---

**David S. Watson**  
King's College London

**Kristin Blesch**  
Leibniz Institute for Prevention  
Research and Epidemiology – BIPS,  
University of Bremen

**Jan Kapar**  
Leibniz Institute for Prevention  
Research and Epidemiology – BIPS,  
University of Bremen

**Marvin N. Wright**  
Leibniz Institute for Prevention  
Research and Epidemiology – BIPS,  
University of Bremen,  
University of Copenhagen

## Abstract

We propose methods for density estimation and data synthesis using a novel form of unsupervised random forests. Inspired by generative adversarial networks, we implement a recursive procedure in which trees gradually learn structural properties of the data through alternating rounds of generation and discrimination. The method is provably consistent under minimal assumptions. Unlike classic tree-based alternatives, our approach provides smooth (un)conditional densities and allows for fully synthetic data generation. We achieve comparable or superior performance to state-of-the-art probabilistic circuits and deep learning models on various tabular data benchmarks while executing about two orders of magnitude faster on average. An accompanying R package, `arf`, is available on CRAN.

## 1 INTRODUCTION

Density estimation is a fundamental unsupervised learning task, an essential subroutine in various methods for data imputation (Efron, 1994; Rubin, 1996), clustering (Bramer, 2007; Rokach and Maimon, 2005), anomaly detection (Chandola et al., 2009; Pang et al., 2021), and classification (Lugosi and Nobel, 1996; Vincent and Bengio,

2002). One important application for density estimators is generative modeling, where we aim to create synthetic samples that mimic the characteristics of real data. These simulations can be used to test the robustness of classifiers (Song et al., 2018; Buzhinsky et al., 2021), augment training sets (Ravuri and Vinyals, 2019; Lopez et al., 2018), or study complex systems without compromising the privacy of data subjects (Augenstein et al., 2020; Yelmen et al., 2021).

The current state of the art in generative modeling relies on deep neural networks, which have proven remarkably adept at synthesizing images, audio, and even video data. Architectures built on variational autoencoders (VAEs) (Kingma and Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014) have dominated the field for the last decade. Recent advances in normalizing flows (Papamakarios et al., 2021) and diffusion models (Ramesh et al., 2022) have sparked considerable interest. While these algorithms are highly effective with structured data, they can struggle in tabular settings with continuous and categorical covariates. Even when successful, deep learning models are notoriously data-hungry and require extensive tuning.

Another major drawback of these deep learning methods is that they do not generally permit tractable inference for tasks such as marginalization and conditioning, which are essential for coherent probabilistic reasoning. A family of hierarchical mixture models known as probabilistic circuits (PCs) (Vergari et al., 2020; Choi et al., 2020) are better suited to such problems. Despite their attractive theoretical properties, existing PCs can also be slow to train and are often far less expressive than unconstrained neural networks.

We introduce an adversarial random forest algorithm that vastly simplifies the task of density estimation and data synthesis. Our method naturally accommodates mixed data

---

Proceedings of the 26<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2023, Valencia, Spain. PMLR: Volume 206. Copyright 2023 by the author(s).

in tabular settings, and performs well on small and large datasets using the computational resources of a standard laptop. It compares favorably with deep learning models while executing some 100 times faster on average. It can be compiled into a PC for efficient and exact probabilistic inference.

Following a brief discussion of related work (Sect. 2), we review relevant notation and background on random forests (Sect. 3). We motivate our method with theoretical results that guarantee convergence under reasonable assumptions (Sect. 4), and illustrate its performance on a range of benchmark tasks (Sect. 5). We conclude with a discussion (Sect. 6) and directions for future work (Sect. 7).

## 2 RELATED WORK

A random forest (RF) is a bootstrap-aggregated (bagged) ensemble of independently randomized trees (Breiman, 2001), typically built using the greedy classification and regression tree (CART) algorithm (Breiman et al., 1984). RFs are extremely popular and effective, widely used in areas like bioinformatics (Chen and Ishwaran, 2012), remote sensing (Belgiu and Drăguț, 2016), and ecology (Cutler et al., 2007), as well as more generic prediction tasks (Fernández-Delgado et al., 2014). Advantages include their efficiency (RFs are embarrassingly parallelizable), ease of use (they require minimal tuning), and ability to adapt to sparse signals (uninformative features are rarely selected for splits).

It is well-known that tree-based models can approximate joint distributions. Several authors advocate using leaf nodes of CART trees as piecewise constant density estimators (Ram and Gray, 2011; Wu et al., 2014; Wen and Hang, 2022). While this method provably converges on the true density in the limit of infinite data, finite sample performance is inevitably rough and discontinuous. Smooth results can be obtained by fitting a distribution within each leaf, e.g. via kernel density estimation (KDE) or maximum likelihood estimation (MLE) (Smyth et al., 1995; Gray and Moore, 2003; Loh, 2009; Ram and Gray, 2011; Criminisi et al., 2012), a version of which we develop further below. Existing methods have mostly been limited to supervised trees rather than unsupervised forests, and are often inefficient in high dimensions.

Another strategy, better suited to high-dimensional settings, uses Chow-Liu trees (Chow and Liu, 1968) to learn a second-order approximation to the underlying joint distribution (Bach and Jordan, 2003; Liu et al., 2011; Rahman et al., 2014). Whereas these methods estimate a series of bivariate densities over the full support of the data, we attempt to solve a larger number of simpler tasks, modeling univariate densities in relatively small subregions.

Despite the popularity of tree-based density estimators, they are rarely if ever used for fully synthetic data generation.

Instead, they are commonly used for *conditional* density estimation and data imputation (Stekhoven and Bühlmann, 2011; Tang and Ishwaran, 2017; Correia et al., 2020; Lundberg et al., 2020; Hothorn and Zeileis, 2021; Ćević et al., 2022). We highlight that methods optimized for this task are often ill-suited to generative modeling, since their reliance on supervised signals limits their ability to capture dependencies between features with little predictive value for the selected outcome variable(s).

Another family of methods for density estimation and data synthesis is based on probabilistic graphical models (PGMs) (Lauritzen, 1996; Koller and Friedman, 2009), e.g. Bayesian networks (Pearl and Russell, 2003; Darwiche, 2009). Learning graph structure is difficult in practice, which is why most methods impose restrictive parametric assumptions for tractability (Heckerman et al., 1995; Drton and Maathuis, 2017). PCs replace the representational semantics of PGMs with an operational semantics, encoding answers to probabilistic queries in the structural alignment of sum and product nodes. This class of computational graphs subsumes sum-product networks (Poon and Domingos, 2011), cutset networks (Rahman et al., 2014), and probabilistic sentential decision diagrams (Kisa et al., 2014), among others. Correia et al. (2020) show that RFs instantiate smooth, decomposable, deterministic PCs, thereby enabling efficient marginalization and maximization.

Deep learning approaches to generative modeling became popular with the introduction of VAEs (Kingma and Welling, 2013) and GANs (Goodfellow et al., 2014), which jointly optimize parameters for network pairs—encoder-decoder and generator-discriminator, respectively—via stochastic gradient descent. Various extensions of these approaches have been developed (Higgins et al., 2017; Arjovsky et al., 2017), including some designed for mixed data in the tabular setting (Choi et al., 2017; Jordon et al., 2019; Xu et al., 2019). While the evidence lower bound of a VAE approximates the data likelihood, there is no straightforward way to compute this quantity with GANs. More recent work in neural density estimation includes autoregressive networks (van den Oord et al., 2016; Ramesh et al., 2021; Roy et al., 2021), normalizing flows (Kobyzev et al., 2021; Papamakarios et al., 2021; Lee et al., 2022), and diffusion models (Kingma et al., 2021; Song et al., 2021; Ramesh et al., 2022). These methods are generally optimized for structured data such as images or audio, where they often attain state-of-the-art results.

## 3 BACKGROUND

Consider the binary classification setting with training data  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ , where  $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$  and  $y_i \in \mathcal{Y} = \{0, 1\}$ . Samples are independent and identically distributed according to some fixed but unknown distribution  $P$  with density  $p$ . The classic RF algorithm takes  $B$  bootstrap sam-

ples of size  $n$  from  $\mathcal{D}$  and fits a binary decision tree for each, in which observations are recursively partitioned according to some optimization target (e.g., Gini index) evaluated on a random subset of features at each node. The size of this subset is controlled by the `max` parameter, conventionally set at  $\lfloor \sqrt{d} \rfloor$  for classification. Resulting splits are literals of the form  $X_j \bowtie x$  for some  $X_j, j \in [d] = \{1, \dots, d\}$ , and value  $x \in \mathcal{X}_j$ , where  $\bowtie \in \{=, <\}$  (the former for categorical, the latter for continuous variables). Data pass to left or right child nodes depending on whether they satisfy the literal. Splits continue until some stopping criterion is met (e.g., purity). Terminal nodes, a.k.a. *leaves*, describe hyperrectangles in feature space with boundaries given by the learned splits. These disjoint cells collectively cover all of  $\mathcal{X}$ . Each leaf is associated with a label  $\hat{y} \in [0, 1]$ , representing either the frequency of positive outcomes (soft labels) or the majority class (hard labels) within that cell. Because trees are grown on independent bootstraps, an average of  $n/e$  samples are excluded from each tree. This so-called “out-of-bag” (OOB) data can be used to estimate empirical risk without need for cross-validation.

Each new datapoint  $\mathbf{x}$  falls into exactly one leaf in each tree. Predictions are computed by aggregating over the trees, e.g. by tallying votes across all  $B$  basis functions of the ensemble. Let  $\theta_b^\ell$  denote the conjunction of literals that characterize membership in leaf  $\ell \in [L_b]$ , where  $L_b$  is the number of leaves in tree  $b \in [B]$ , with corresponding hyperrectangular subspace  $\mathcal{X}_b^\ell \subset \mathcal{X}$ . Each leaf has some nonnegative volume and diameter, denoted  $\text{vol}(\mathcal{X}_b^\ell)$  and  $\text{diam}(\mathcal{X}_b^\ell)$ , where the latter represents the longest line segment contained in  $\mathcal{X}_b^\ell$ . Let  $n_b$  be the number of training samples for tree  $b$  (not necessarily equal to  $n$ ) and  $n_b^\ell$  the number of samples that fall into leaf  $\ell$  of  $b$ . The ratio  $n_b^\ell/n_b$  represents an empirical estimate of the leaf’s coverage  $p(\theta_b^\ell)$ , i.e. the probability that a random  $\mathbf{x}$  falls within  $\mathcal{X}_b^\ell$ . A tree is fully parametrized by  $\theta_b = \bigcup_{\ell=1}^{L_b} \theta_b^\ell$ , and the complete forest by  $\Theta = \bigcup_{b=1}^B \theta_b$ .

Many variations of the classic algorithm exist, including a number of simplified versions designed to be more amenable to statistical analysis. See (Biau and Scornet, 2016) for an overview. Common sources of variation include how observations are randomized across trees (e.g., by subsampling or bootstrapping) and how splits are selected (e.g., uniformly or according to some adaptive procedure).

Our method builds on the *unsupervised* random forest (URF) algorithm (Shi and Horvath, 2006).<sup>1</sup> This procedure creates a synthetic dataset  $\tilde{\mathbf{X}}$  of  $n$  observations by independently sampling from the marginals of  $\mathbf{X}$ , i.e.  $\tilde{\mathbf{x}} \sim \prod_{j=1}^d P(X_j)$ . A RF classifier is trained to distinguish between  $\mathbf{X}$  and  $\tilde{\mathbf{X}}$ , with labels indicating whether samples are original ( $Y = 1$ )

<sup>1</sup>Not to be confused with RF variants that employ non-adaptive splits, which are sometimes also referred to as unsupervised, since they ignore the response variable. See, e.g., Genuer (2012).

or synthetic ( $Y = 0$ ). The method has expected accuracy  $1/2$  in the worst case, corresponding to a dataset in which all features are mutually independent. However, if dependencies are present, then a consistent learning procedure will converge on expected accuracy  $1/2 + \delta$  for some  $\delta > 0$  as  $n$  grows (Kim et al., 2021).

## 4 ADVERSARIAL RANDOM FORESTS

We introduce a recursive variant of URFs, which we call *adversarial random forest* (ARF). The goal of this algorithm is to render data jointly independent within each leaf. We achieve this by first fitting an ordinary URF  $f^{(0)}$  with synthetic data  $\tilde{\mathbf{X}}^{(0)}$ . We compute the coverage of each leaf w.r.t. original data, then generate a new synthetic dataset by sampling from marginals within random leaves selected with probability proportional to this coverage. Call the resulting  $n \times d$  matrix  $\tilde{\mathbf{X}}^{(1)}$ . A new classifier  $f^{(1)}$  is trained to distinguish  $\mathbf{X}$  from  $\tilde{\mathbf{X}}^{(1)}$ . If OOB accuracy for this model is sufficiently low, then the ARF has converged, and we move forward with splits from  $f^{(0)}$ . Otherwise, we iterate the procedure, drawing a new synthetic dataset from the splits learned by  $f^{(1)}$  and evaluating performance via a new classifier. The loop repeats until convergence (see Alg. 1).

ARFs bear some obvious resemblance to GANs. The “generator” is a simple sampling scheme that draws from the marginals in adaptively selected subregions; the “discriminator” is a RF classifier. The result can be understood as a zero-sum game in which adversaries take turns increasing and decreasing label uncertainty at each round. However, beyond this conceptual link between our method and GANs lie some important differences. Both generator and discriminator share the same parameters in our algorithm. Indeed, our generator does not, strictly speaking, *learn* anything; it merely exploits what the discriminator has learned. This means that ARFs cannot be used for adversarial attacks of the sort made famous by GANs, which involve separately parametrized networks for each model. Moreover, the synthetic data generated by ARFs is relatively naïve, consisting of bootstrap samples drawn from subsets of the original observations. That is because our goal is not (yet) to generate new data, but merely to learn an independence-inducing partition. Empirically, we find that this is often achieved in just a single round even with the tolerance  $\delta$  set to 0.

Formally, we seek a set of splits  $\Theta$  such that, for all trees  $b$ , leaves  $\ell$ , and samples  $\mathbf{x}$ , we have  $p(\mathbf{x}|\theta_b^\ell) = \prod_{j=1}^d p(x_j|\theta_b^\ell)$ . Call this the *local independence criterion*. Our first result states that ARFs converge on this identity in the limit. We assume:

- (A1) The feature domain is limited to  $\mathcal{X} = [0, 1]^d$ , with joint density  $p$  bounded away from 0 and  $\infty$ .
- (A2) At each round, the target function  $P(Y = 1|\mathbf{x})$  is Lipschitz-continuous. The Lipschitz constant may

---

**Algorithm 1** ADVERSARIAL RANDOM FOREST

---

**Input:** Training data  $\mathbf{X}$ , tolerance  $\delta$   
**Output:** Random forest classifier  $f^{(0)}$

Sample  $\tilde{\mathbf{X}}^{(0)} \sim \prod_{j=1}^d P(X_j)$   
 $\mathbf{X}^+ \leftarrow \text{row.append}(\mathbf{X}, \tilde{\mathbf{X}}^{(0)})$   
 $Y \leftarrow \text{row.append}(\mathbf{1}_n, \mathbf{0}_n)$   
 $f^{(0)} \leftarrow \text{RANDOMFOREST}(\mathbf{X}^+, Y)$   
**if**  $\text{ACC}(f^{(0)}) > 1/2 + \delta$  **then**  
  converged  $\leftarrow$  FALSE  
  **while not** converged **do**  
    **for all**  $b \in [B^{(0)}], \ell \in [L_b^{(0)}]$  **do**  
       $q(\theta_b^\ell) \leftarrow \frac{2}{n_b} \sum_{i: \mathbf{x}_i \in \mathcal{X}_b^\ell} y_i$   
    **end for**  
    **for**  $i \in [n]$  **do**  
      Sample tree  $b \in [B^{(0)}]$  uniformly  
      Sample leaf  $\ell \in [L_b^{(0)}]$  w.p.  $q(\theta_b^\ell)$   
      Sample  $\tilde{\mathbf{x}}_i^{(1)} \sim \prod_{j=1}^d P(X_j | \theta_b^\ell)$   
    **end for**  
     $\mathbf{X}^+ \leftarrow \text{row.append}(\mathbf{X}, \tilde{\mathbf{X}}^{(1)})$   
     $f^{(1)} \leftarrow \text{RANDOMFOREST}(\mathbf{X}^+, Y)$   
    **if**  $\text{ACC}(f^{(1)}) \leq 1/2 + \delta$  **then**  
      converged  $\leftarrow$  TRUE  
    **else**  
       $f^{(0)} \leftarrow f^{(1)}$   
    **end if**  
  **end while**  
**end if**

---

vary with from one round to the next, but it does not increase faster than  $1/\max_{\ell, b} (\text{diam}(\mathcal{X}_b^\ell))$ .

- (A3) Trees satisfy the following conditions: (i) training data for each tree is split into two subsets: one to learn split parameters, the other to assign leaf labels; (ii) trees are grown on subsamples rather than bootstraps, with subsample size  $n_b$  satisfying  $n_b \rightarrow \infty, n_b/n \rightarrow 0$  as  $n \rightarrow \infty$ ; (iii) at each internal node, the probability that a tree splits on any given  $X_j$  is bounded from below by some  $\pi > 0$ ; (iv) every split puts at least a fraction  $\gamma \in (0, 0.5]$  of the available observations into each child node; (v) for each tree  $b$ , the total number of leaves  $L_b$  satisfies  $L_b \rightarrow \infty, L_b/n \rightarrow 0$  as  $n \rightarrow \infty$ ; and (vi) predictions are made with soft labels both within leaves and across trees, i.e. by averaging rather than voting.

(A1) is simply for notational convenience, and can be replaced w.l.o.g. by bounding the feature domain with arbitrary constants. Lipschitz continuity is a common learning theoretic assumption widely used in the analysis of RFs. (A2)'s extra condition regarding the Lipschitz constant controls the variation in smoothness over adversarial training rounds. (A3) imposes standard regularity conditions for RFs (Meinshausen, 2006; Biau, 2012; Denil et al., 2014; Scornet, 2016; Wager and Athey, 2018). With these assumptions in place, we have the following result (see Appx. A for all proofs).

---

**Algorithm 2** FORDE

---

**Input:** ARF classifier  $f$ , training data  $\mathbf{X} \in \mathbb{R}^{n \times d}$   
**Output:** Estimated density  $q$

**for all**  $b \in [B], \ell \in [L_b]$  **do**  
   $q(\theta_b^\ell) \leftarrow \frac{2}{n_b} \sum_{i: \mathbf{x}_i \in \mathcal{X}_b^\ell} y_i$   
  **for**  $j \in [d]$  **do**  
     $\psi_{b,j}^\ell \leftarrow$  estimated parameter(s) for  $p(x_j | \theta_b^\ell)$   
     $q(\cdot; \psi_{b,j}^\ell) \leftarrow$  corresponding pdf/pmf  
  **end for**  
**end for**

---



---

**Algorithm 3** FORGE

---

**Input:** FORDE model  $q$ , target sample size  $m$   
**Output:** Synthetic dataset  $\tilde{\mathbf{X}} \in \mathbb{R}^{m \times d}$

**for**  $i \in [m]$  **do**  
  Sample tree  $b \in [B]$  uniformly  
  Sample leaf  $\ell \in [L_b]$  w.p.  $q(\theta_b^\ell)$   
  **for**  $j \in [d]$  **do**  
    Sample data  $\tilde{x}_{ij} \sim q(\cdot; \psi_{b,j}^\ell)$   
  **end for**  
**end for**

---

**Theorem 1** (Convergence). *Under (A1)-(A3), ARFs converge in probability on the local independence criterion. Let  $\Theta_n$  be the parameters of an ARF trained on a sample of size  $n$ . Then for all  $\mathbf{x} \in \mathcal{X}$ ,  $\theta_b^\ell \in \Theta_n$ , and  $\epsilon > 0$ :*

$$\lim_{n \rightarrow \infty} \mathbb{P} \left[ \left| p(\mathbf{x} | \theta_b^\ell) - \prod_{j=1}^d p(x_j | \theta_b^\ell) \right| \geq \epsilon \right] = 0.$$

#### 4.1 Density Estimation and Data Synthesis

ARFs are the basis for two further algorithms, FORests for Density Estimation (FORDE) and FORests for GENERative modeling (FORGE). We present pseudocode for both (see Algs. 2 and 3). The key point to recognize is that, under the local independence criterion, joint densities can be learned by running  $d$  separate univariate estimators within each leaf. This is exponentially easier than multivariate density estimation, which suffers from the notorious *curse of dimensionality*. Summarizing the challenges with estimating joint densities, one recent textbook on KDE concludes that “nonparametric methods for kernel density problems should not be used for high-dimensional data and it seems that a feasible dimensionality should not exceed five or six...” (Gramacki, 2018, p. 60). By contrast, our method scales much better with data dimensionality, exploiting the flexibility of ARFs to learn an independence-inducing partition that renders density estimation relatively straightforward.

Of course, this does not *escape* the curse of dimensionality so much as relocate it. The cost for this move is potentially deep trees and/or many ARF training rounds, especially when dependencies between covariates are strong or com-



plex. However, deep forests are generally more efficient than deep neural networks in terms of data and computation, and our experiments suggest that ARF convergence is usually fast even for  $\delta = 0$  (see Sect. 5).

With our ARF in hand, the algorithm proceeds as follows. For each tree  $b$ , we record the split criteria  $\theta_b^\ell$  and empirical coverage  $q(\theta_b^\ell)$  of each leaf  $\ell$ . Call these the *leaf parameters*. Then we estimate *distribution parameters*  $\psi_{b,j}^\ell$  independently for each (original)  $X_j$  within  $\mathcal{X}_b^\ell$ , e.g. the kernel bandwidth for KDE or class probabilities for MLE with categorical data. In the continuous case,  $\psi_{b,j}^\ell$  must either encode leaf bounds (e.g., via a truncated normal distribution with extrema given by  $\theta_b^\ell$ ) or include a normalization constant to ensure integration to unity. The generative model then follows a simple two-step procedure. First, sample a tree uniformly from  $[B]$  and a leaf from that tree with probability  $q(\theta_b^\ell)$ , just as we do to construct synthetic data within the recursive loop of the ARF algorithm. Next, sample data for each feature  $X_j$  according to the density/mass function parametrized by  $\psi_{b,j}^\ell$ . We repeat this procedure until the target number of synthetic samples has been generated.

We are deliberately agnostic about how distribution parameters  $\psi_{b,j}^\ell$  should be learned, as this will tend to vary across features. In our theoretical analysis, we restrict focus to continuous variables and consider a flexible family of KDE methods. In our experiments, we use MLE for continuous data, effectively implementing a truncated Gaussian mixture model, and Bayesian inference for categorical variables, to avoid extreme probabilities when values are unobserved but not beyond the support of a given leaf. Under local independence, distribution learning is completely modular, so different methods can coexist without issue. We revisit this topic in Sect. 6.

Our estimated density takes the following form:

$$q(\mathbf{x}) = \frac{1}{B} \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} q(\theta_b^\ell) \prod_{j=1}^d q(x_j; \psi_{b,j}^\ell). \quad (1)$$

Compare this with the true density:

$$p(\mathbf{x}) = \frac{1}{B} \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} p(\theta_b^\ell) p(\mathbf{x} | \theta_b^\ell). \quad (2)$$

In both cases, the density evaluated at a given point is just a coverage-weighted average of its density in all leaves whose split criteria it satisfies.

Because we are concerned with  $L_2$ -consistency, our loss function is the mean integrated squared error (MISE)<sup>2</sup>, defined as:

$$\text{MISE}(p, q) := \mathbb{E} \left[ \int_{\mathcal{X}} (p(\mathbf{x}) - q(\mathbf{x}))^2 d\mathbf{x} \right].$$

<sup>2</sup>Alternative loss functions may also be suitable, e.g. the Kullback-Leibler divergence or the Wasserstein distance.

We require one extra assumption, imposing standard conditions for KDE consistency (Silverman, 1986):

- (A4) The true density function  $p$  is smooth. Specifically, its second derivative  $p''$  is finite, continuous, square integrable, and ultimately monotone.

Our method admits three potential sources of error, quantified by the following residuals:

$$\epsilon_1 := \epsilon_1(\ell, b) := p(\theta_b^\ell) - q(\theta_b^\ell) \quad (3)$$

$$\epsilon_2 := \epsilon_2(\ell, b, \mathbf{x}) := \prod_{j=1}^d p(x_j | \theta_b^\ell) - \prod_{j=1}^d q(x_j; \psi_{b,j}^\ell) \quad (4)$$

$$\epsilon_3 := \epsilon_3(\ell, b, \mathbf{x}) := p(\mathbf{x} | \theta_b^\ell) - \prod_{j=1}^d p(x_j | \theta_b^\ell) \quad (5)$$

We refer to these as errors of *coverage*, *density*, and *convergence*, respectively. Observe that  $\epsilon_1$  is a random variable that depends on  $\ell$  and  $b$ , while  $\epsilon_2, \epsilon_3$  are random variables depending on  $\ell, b$  and  $\mathbf{x}$ . We suppress the dependencies for ease of notation.

**Lemma 1.** The error of our estimator satisfies the following bound:

$$\text{MISE}(p, q) \leq 2B^{-2} \mathbb{E} \left[ \int_{\mathcal{X}} \alpha^2 + \beta^2 d\mathbf{x} \right],$$

where

$$\alpha := \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} p(\theta_b^\ell) \epsilon_3 \quad \text{and}$$

$$\beta := \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} \left( p(\theta_b^\ell) \epsilon_2 + \epsilon_1 \prod_{j=1}^d p(x_j | \theta_b^\ell) - \epsilon_1 \epsilon_2 \right).$$

This lemma establishes that total error is bounded by a quadratic function of  $\epsilon_1, \epsilon_2, \epsilon_3$ . We know by Thm. 1 that errors of convergence vanish in the limit. Our next result states that the same holds for errors of coverage and density.

**Theorem 2 (Consistency).** *Under assumptions (A1)-(A4), FORDE is  $L_2$ -consistent. Let  $q_n$  denote the joint density estimated on a training sample of size  $n$ . Then we have:*

$$\lim_{n \rightarrow \infty} \text{MISE}(p, q_n) = 0.$$

Our consistency proof is fundamentally unlike those of piecewise constant density estimators with CART trees (Ram and Gray, 2011; Wu et al., 2014; Correia et al., 2020), which essentially treat base learners as adaptive histograms and rely on tree-wise convergence when leaf volume goes to zero (Devroye et al., 1996; Lugosi and Nobel, 1996). Alternative methods that perform KDE or MLE within each leaf do not come with theoretical guarantees (Smyth et al., 1995;

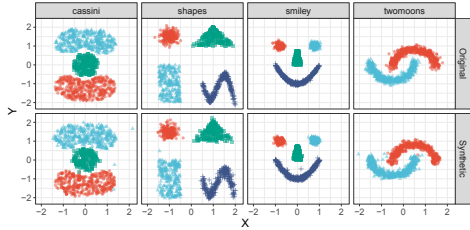


Figure 1: Visual examples. Original (top) and synthetic (bottom) data are presented for four three-dimensional problems with two continuous covariates and one categorical feature.

Gray and Moore, 2003; Loh, 2009; Ram and Gray, 2011). Recently, consistency has been shown for some RF-based conditional density estimators (Hothorn and Zeileis, 2021; Čevič et al., 2022). However, these results do not extend to the unconditional case, since features with little predictive value for the outcome variable(s) are unlikely to be selected for splits. The resulting models will therefore fail to detect dependencies between features deemed uninformative for the given prediction task. In the rare case that authors use some form of unsupervised splits, they make no effort to factorize the distribution and are therefore subject to the curse of dimensionality (Criminisi et al., 2012; Feng and Zhou, 2018). By contrast, our method exploits ARFs to find regions of local independence, and univariate density estimation to compute marginals within each leaf. Though our consistency result comes at the cost of some extra assumptions, we argue that this is a fair price to pay for improved performance in finite samples.

## 5 EXPERIMENTS

In this section, we present results from a wide range of experiments conducted on simulated and real-world datasets. We use 100 trees for density estimation tasks and 20 for data synthesis. Increasing this parameter tends to improve performance for FORDE, but appears to have less of an impact on FORGE. Trees are grown until purity or a minimum node size of two (with just a single sample, variance is undefined). In all cases, we set the slack parameter  $\delta = 0$  and use the default `mtry` =  $\lfloor \sqrt{d} \rfloor$ . For more details on hyperparameters and datasets see Appx. B. Code for reproducing all results is available online at [https://github.com/bips-hb/arf\\_paper](https://github.com/bips-hb/arf_paper).

### 5.1 Simulation

**FORGE recreates visual patterns.** We begin with a simple proof of concept experiment, illustrating our method on a handful of low-dimensional datasets that allow for easy visual assessment. The *cassini*, *shapes*, *smiley*, and *twomoons* problems are all three-dimensional examples

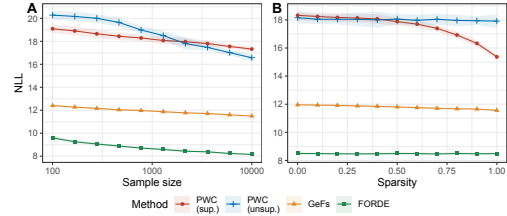


Figure 2: Negative log-likelihood (NLL) measured in nats on a test set for varying sample size (A) and sparsity (B). Lower is better. Shading represents standard errors.

that combine two continuous covariates with a categorical class label. We simulate  $n = 2000$  samples from each data generating process (see Fig. 1, top row) and estimate densities using FORDE. We proceed to FORGE a synthetic dataset of  $n = 1000$  samples (Fig. 1, bottom row) and compare results. We find that the model consistently approximates its target distribution with high fidelity. Classes are clearly distinguished in all cases, and the visual form of the original data is immediately recognizable. A few stray samples are evident on close inspection. Such anomalies can be mitigated with a larger training set.

### FORDE outperforms alternative CART-based methods.

We simulate data from a multivariate Gaussian distribution  $\mathbf{X} \sim \mathcal{N}(0, \Sigma)$ , with Toeplitz covariance matrix  $\Sigma_{ij} = 0.9^{|i-j|}$  and fixed  $d = 10$ . To compare against supervised methods, we also simulate a binary target  $Y \sim \text{Bern}([1 + \exp(-\mathbf{X}\beta)]^{-1})$ , where the coefficient vector  $\beta$  contains a varying proportion of 0's (non-informative features) and 1's (informative features). Performance is evaluated by the negative log-likelihood (NLL) on a test set of  $n_{\text{tst}} = 1000$ . We compare our method to piecewise constant (PWC) estimators with supervised and unsupervised split criteria,<sup>3</sup> as well as generative forests (GeFs), a RF-based smooth density estimation procedure (Correia et al., 2020).

Fig. 2 shows the average NLL over 20 replicates for varying sample sizes (A) and levels of sparsity (B). For the former, we fix the proportion of informative features at 0.5; for the latter, we fix  $n_{\text{trn}} = 2000$ . We find that PWC methods fare poorly, with much greater NLL in all settings. This is likely due to the unrealistic uniformity assumption, according to which the corner of a hyperrectangle is no less probable than the center. GeFs, which also use a Gaussian mixture model to estimate densities, perform better in this experiment. However, FORDE dominates throughout.

<sup>3</sup>Supervised PWC is simply an ensemble version of the classic method (Gray and Moore, 2003; Ram and Gray, 2011; Wu et al., 2014). To the best of our knowledge, no one has previously proposed *unsupervised* PWC density estimation with CART trees. This can be understood as a variant of our approach in which all marginals are uniform within each leaf.

Table 1: Average NLL on the Twenty Datasets benchmark for five PC models and FORDE. Winning results in bold.

Dataset	EiNet	RAT-SPN	PGC	Strudel	CMCLT	FORDE
nitcs	6.02	6.01	6.05	6.07	<b>5.99</b>	6.01
msnbc	6.12	<b>6.04</b>	6.06	<b>6.04</b>	6.05	6.10
kdd	2.18	2.13	2.14	2.14	<b>2.12</b>	2.13
plants	13.68	13.44	13.52	13.22	<b>12.26</b>	<b>12.26</b>
audio	39.88	39.96	40.21	42.40	<b>39.02</b>	39.74
jester	52.56	52.97	53.54	54.24	<b>51.94</b>	52.8
netflix	56.64	56.85	57.42	57.93	<b>55.31</b>	56.67
accidents	35.59	35.49	30.46	29.05	<b>28.69</b>	33.85
retail	10.92	10.91	10.84	10.83	<b>10.82</b>	10.93
pumsb	31.95	31.53	29.56	24.39	<b>23.71</b>	28.4
dna	96.09	97.23	<b>80.82</b>	87.15	84.91	91.85
kosarek	11.03	10.89	10.72	10.70	<b>10.56</b>	10.84
msweb	10.03	10.12	9.98	9.74	<b>9.62</b>	9.72
book	34.74	34.68	34.11	34.49	<b>33.75</b>	34.85
movie	51.71	53.63	53.15	53.72	<b>49.23</b>	50.86
webkb	157.28	157.53	155.23	154.83	<b>147.77</b>	153.45
reuters	87.37	87.37	87.65	86.35	<b>81.17</b>	84.15
20ng	153.94	152.06	154.03	153.87	<b>148.17</b>	155.51
bbc	248.33	252.14	254.81	256.53	242.83	<b>240.31</b>
ad	26.27	48.47	21.65	16.52	<b>14.76</b>	21.80
Avg. rank	4.5	4.2	4	3.6	1.2	3.3

Panel (B) clearly illustrates that unsupervised methods are unaffected by changes in signal sparsity, since their splits are independent of the outcome variable  $Y$ . By contrast, sparsity appears to benefit the supervised methods. This can be explained by the fact that splits are random when features are uninformative, a strategy that is known to work well in noisy settings (Geurts et al., 2006; Genuer, 2012).

## 5.2 Real Data

**FORDE is competitive with alternative PCs.** Building on Correia et al. (2020)’s observation that RFs can be compiled into probabilistic circuits, we compare the performance of FORDE to that of five leading PCs on the Twenty Datasets benchmark (Van Haaren and Davis, 2012), a heterogeneous collection of tasks ranging from retail to biology that is widely used to evaluate tractable probabilistic models. Each dataset is randomly split into training (70%), validation (10%), and test sets (20%). Competitors include Einsum networks (EiNet) (Peharz et al., 2020a), random sum-product networks (RAT-SPN) (Peharz et al., 2020b), probabilistic generating circuits (PGC) (Zhang et al., 2021), Strudel (Dang et al., 2022), and continuous mixtures of Chow-Liu trees (CMCLT) (Correia et al., 2023). We report the average NLL on the test set for each model in Table 1. Though the recently proposed CMCLT algorithm generally dominates in this experiment, FORDE attains top performance on two datasets and is never far behind the state of the art. Its average rank of 3.3 places it second overall.

**FORGE generates realistic tabular data.** To evaluate the performance of FORGE on real-world datasets, we recreate a benchmarking pipeline originally proposed by Xu et al. (2019). They introduce the conditional tabular GAN (CTGAN) and tabular VAE (TVAE), two deep learning algorithms for generative modeling with mixed continuous and categorical features. We include three additional state-of-

the-art tabular GAN architectures for comparison: invertible tabular GAN (IT-GAN) (Lee et al., 2021), regularized compound conditional GAN (RCC-GAN) (Esmailpour et al., 2022), and a differentially private conditional tabular GAN (CTAB-GAN+) (Zhao et al., 2022).

A complete summary of the experimental setup is presented in Appx. B. Briefly, we take five benchmark datasets for classification and partition the samples into training and test sets, which we denote by  $\mathbf{Z}_{\text{tm}} = (\mathbf{X}_{\text{tm}}, Y_{\text{tm}})$  and  $\mathbf{Z}_{\text{tst}} = (\mathbf{X}_{\text{tst}}, Y_{\text{tst}})$ , respectively.  $\mathbf{Z}_{\text{tm}}$  is used as input to a series of generative models, each of which creates a synthetic training set  $\tilde{\mathbf{Z}}_{\text{tm}}$  of the same sample size as the original. Several classifiers are then trained on  $\tilde{\mathbf{Z}}_{\text{tm}}$  and evaluated on  $\mathbf{Z}_{\text{tst}}$ , with performance metrics averaged across learners. Results are benchmarked against the same set of algorithms, now trained on the original data  $\mathbf{Z}_{\text{tm}}$ . We refer to this model as the *oracle*, since it should perform no worse in expectation than any classifier trained on synthetic data. However, if the generative model approximates its target with high fidelity, then differences between the oracle and its competitors should be negligible.<sup>4</sup> Similar approaches are widely used in the evaluation of GANs (Yang et al., 2017; Shmelkov et al., 2018; Santurkar et al., 2018); for a critical discussion, see Ravuri and Vinyals (2019).

Results are reported in Table 2, where we average over five trials of data synthesis and subsequent supervised learning. We include information on each dataset, including the cardinality of the response variable, the training/test sample size, and dimensionality of the feature space. Performance is evaluated via accuracy and F1-score (or F1 macro-score for multiclass problems), as well as wall time. FORGE fares well in this experiment, attaining the top accuracy and F1-score in three out of five tasks. On a fourth, the highly imbalanced `credit` dataset, the only models that do better in terms of accuracy receive F1-scores of 0, suggesting that they entirely ignore the minority class. Only FORGE and RCC-GAN strike a reasonable balance between sensitivity and specificity on this task. Perhaps most impressive, FORGE executes over 60 times faster than its nearest competitor on average, and over 100 times faster than the second fastest method. (We omit results for algorithms that fail to converge in 24 hours of training time.) Differences in compute time would be even more dramatic if these deep learning algorithms were configured with a CPU backend (we used GPUs here), or if FORGE were run using more extensive parallelization (we distribute the job across 10 cores). This comparison also obscures the extra time required to tune hyperparameters for these complex models, whereas our method is an off-the-shelf solution that works

<sup>4</sup>Note that the so-called “oracle” is not necessarily optimal w.r.t. the true data generating process—other models may have lower risk—but it should be optimal w.r.t. a given function class-dataset pair. If logistic regression attains 60% test accuracy training on  $\mathbf{Z}_{\text{tm}}$ , then it should do about the same training on  $\tilde{\mathbf{Z}}_{\text{tm}}$ , regardless of how much better a well-tuned MLP may perform.

Table 2: Performance on the Xu et al. (2019) benchmark for five deep learning models and FORGE. We report average results across five replicates  $\pm$  standard errors. Winning results in bold.

Dataset	Model	Accuracy $\pm$ SE	F1 $\pm$ SE	Time (sec)
adult classes = 2 $n_{\text{trn}} = 23\text{k}$ $n_{\text{tst}} = 10\text{k}$ $d = 14$	Oracle	$0.828 \pm 0.006$	$0.884 \pm 0.004$	
	FORGE	<b><math>0.819 \pm 0.006</math></b>	<b><math>0.877 \pm 0.005</math></b>	<b>2.9</b>
	CTGAN	$0.786 \pm 0.020$	$0.853 \pm 0.019$	263.3
	CTAB-GAN+	$0.808 \pm 0.008$	$0.869 \pm 0.006$	561.6
	IT-GAN	$0.794 \pm 0.005$	$0.853 \pm 0.005$	3435.6
	RCC-GAN	$0.770 \pm 0.015$	$0.841 \pm 0.015$	8823.0
	TVAE	$0.804 \pm 0.007$	$0.865 \pm 0.006$	115.1
census classes = 2 $n_{\text{trn}} = 200\text{k}$ $n_{\text{tst}} = 100\text{k}$ $d = 40$	Oracle	$0.922 \pm 0.002$	$0.957 \pm 0.001$	
	FORGE	$0.903 \pm 0.019$	$0.946 \pm 0.012$	<b>53.2</b>
	CTGAN	$0.916 \pm 0.015$	$0.954 \pm 0.009$	4287.8
	CTAB-GAN+	$0.912 \pm 0.026$	$0.952 \pm 0.016$	10182.1
	IT-GAN	NA	NA	>24hr
	RCC-GAN	$0.900 \pm 0.016$	$0.944 \pm 0.011$	8908.6
	TVAE	<b><math>0.928 \pm 0.007</math></b>	<b><math>0.961 \pm 0.004</math></b>	1814.9
covertype classes = 7 $n_{\text{trn}} = 481\text{k}$ $n_{\text{tst}} = 100\text{k}$ $d = 54$	Oracle	$0.895 \pm 0.000$	$0.838 \pm 0.000$	
	FORGE	<b><math>0.707 \pm 0.006</math></b>	<b><math>0.549 \pm 0.006</math></b>	<b>103.5</b>
	CTGAN	$0.633 \pm 0.009$	$0.400 \pm 0.009$	13387.2
	CTAB-GAN+	NA	NA	>24hr
	IT-GAN	NA	NA	>24hr
	RCC-GAN	NA	NA	>24hr
	TVAE	$0.698 \pm 0.013$	$0.459 \pm 0.013$	4882.0
credit classes = 2 $n_{\text{trn}} = 264\text{k}$ $n_{\text{tst}} = 20\text{k}$ $d = 30$	Oracle	$0.997 \pm 0.001$	$0.607 \pm 0.029$	
	FORGE	$0.995 \pm 0.001$	$0.527 \pm 0.036$	<b>32.2</b>
	CTGAN	$0.881 \pm 0.099$	$0.047 \pm 0.031$	4898.0
	CTAB-GAN+	<b><math>0.998 \pm 0.000</math></b>	$0.000 \pm 0.000$	7497.3
	IT-GAN	NA	NA	>24hr
	RCC-GAN	$0.993 \pm 0.003$	<b><math>0.569 \pm 0.056</math></b>	10608.4
	TVAE	<b><math>0.998 \pm 0.000</math></b>	$0.000 \pm 0.000$	3847.6
intrusion classes = 5 $n_{\text{trn}} = 394\text{k}$ $n_{\text{tst}} = 100\text{k}$ $d = 40$	Oracle	$0.998 \pm 0.001$	$0.833 \pm 0.001$	
	FORGE	<b><math>0.993 \pm 0.001</math></b>	<b><math>0.656 \pm 0.001</math></b>	<b>68.2</b>
	CTGAN	$0.944 \pm 0.088$	$0.645 \pm 0.088$	8749.3
	CTAB-GAN+	NA	NA	>24hr
	IT-GAN	NA	NA	>24hr
	RCC-GAN	NA	NA	>24hr
	TVAE	$0.990 \pm 0.002$	$0.598 \pm 0.002$	4306.0

well with default settings.

### 5.3 Runtime

To further demonstrate the computational efficiency of our pipeline relative to deep learning methods, we conduct a runtime experiment using the smallest dataset above, `adult`. By repeatedly sampling stratified subsets—varying both sample size  $n$  and dimensionality  $d$ —and measuring the time needed to train a generative model and synthesize data from it, we illustrate how complexity scales with  $n$  and  $d$ . For this experiment, we ran the three fastest deep learning competitors—CTGAN, TVAE, and CTAB-GAN+—with both CPU and GPU backends. We use default parameters for all algorithms, which include automated parallelization over all available cores (24 in this experiment).

Fig. 3 shows the results. FORGE clearly dominates in training time (see panels A and C), executing orders of magnitude faster than the competition (note the log scale). For those with limited access to GPUs, deep learning methods may be completely infeasible for large datasets. Even when GPUs are available, FORGE still scales far better, completing the full pipeline about 35 times faster than TVAE, 85 times faster than CTGAN, and nearly 200 times faster than CTAB-GAN+ in this example. Other methods appear to gen-

erate samples more quickly than FORGE (see panels B and D), but this computation is trivial compared to training. Interestingly, our method is a faster sampler when measured in processing time (see Fig. 4, Appx. B.4), suggesting that it could outperform competitors here too with more efficient parallelization. Note that FORGE attains the highest accuracy and F1-score of all methods for the `adult` dataset, so this speedup need not come at the cost of performance.

## 6 DISCUSSION

ARFs enable fast, accurate density estimation and data synthesis. However, the method is not without its limitations. First, it is not tailored to structured data such as images or text, for which deep learning models have proven especially effective. See Appx. B.5 for a comparison to state-of-the-art models on the MNIST dataset, where convolutional GANs clearly outperform FORGE, as expected. Where our method excels, by contrast, is in speed and flexibility.

We caution that our convergence guarantees have no implications for finite sample performance. Though ARFs only required a few rounds of training in most of our experiments, it is entirely possible that discriminator accuracy increase from one round to the next, or that Alg. 1 fail to terminate altogether for some datasets. (In practice, this behavior is mitigated by increasing  $\delta$  or setting some maximum number of iterations.) For instance, on MNIST, we generally find accuracy plateauing around 65% after five rounds with little improvement thereafter. Of course, the same caveats apply to any asymptotic guarantee. Finite sample results are rare in the RF literature, although there has been some recent work in this area (Gao et al., 2022).

Another potential difficulty for our approach is selecting an optimal density estimation subroutine. KDE relies on a smoothness assumption (A4), while MLE requires a (local) parametric model. Bayesian inference imposes a prior distribution, which may bias results. All three methods will struggle when their assumptions are violated. Resampling alternatives such as permutations or bootstrapping do not produce any data that was not observed in the training set and may therefore raise privacy concerns. No approach is generally guaranteed to strike the optimal balance between efficiency, accuracy, and privacy, and so the choice of which combination of methods to employ is irreducibly context-dependent.

We emphasize that our method performs well in a range of settings without any model tuning. However, we acknowledge that optimal performance likely depends on RF parameters (Scornet, 2017; Probst et al., 2019). In particular, there is an inherent trade-off between the goals of minimizing errors of density ( $\epsilon_2$ ) and errors of convergence ( $\epsilon_3$ ) in finite samples. Grow trees too deep, and leaves will not contain enough data to accurately estimate marginal densities; grow trees too shallow, and ARFs may not satisfy

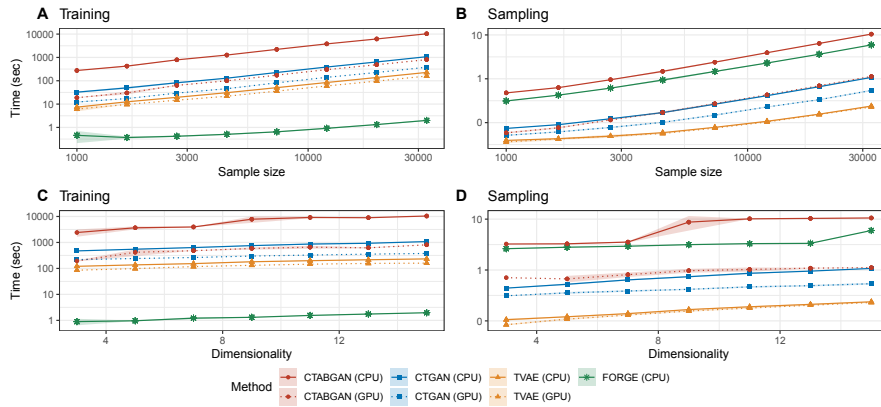


Figure 3: Complexity curves, evaluated using stratified subsamples of the `adult` dataset. (A): Training time as a function of sample size. (B): Sampling time as a function of sample size. (C): Training time as a function of dimensionality. (D): Sampling time as a function of dimensionality.

the local independence criterion. Meanwhile, the `mtry` parameter has been shown to control sparsity in low signal-to-noise regimes (Mentch and Zhou, 2020). Smaller values may therefore be appropriate when  $\epsilon_3$  is large, in order to regularize the forest. Adding more trees tends to improve density estimates, though this incurs extra computational cost in both time and memory (Probst and Boulesteix, 2017). Despite these considerations, we reiterate that ARFs do remarkably well with default parameters.

The ethical implications of generative models are potentially fraught. Deepfakes have attracted particular attention in this regard (de Ruiter, 2021; Öhman, 2020; Diakopoulos and Johnson, 2021), as they can deceive their audience into believing that people said or did things they never in fact said or did. These dangers are most acute with convolutional neural networks or other architectures optimized for visual and audio data. Despite and in full awareness of these concerns, we point out that generative models also present a valuable ethical opportunity, since they may preserve the privacy of data subjects by creating datasets that preserve statistical signals without exposing the personal information of individuals. However, the privacy-utility trade-off can be unpredictable with synthetic data (Stadler et al., 2022). As with all powerful technologies, caution is advised and regulatory frameworks are welcome.

## 7 CONCLUSION

We have introduced a novel procedure for learning joint densities and generating synthetic data using a recursive, adversarial variant of unsupervised random forests. The method is provably consistent under reasonable assumptions, and performs well in experiments on simulated and real-world examples. Our FORDE algorithm is more accurate

than other CART-based density estimators and compares favorably to leading PC algorithms. Our FORGE algorithm is competitive with deep learning models for data generation on tabular data benchmarks, and routinely executes some 100 times faster. An R package, `arf`, is available on CRAN. A Python implementation is forthcoming.

Future work will explore further applications for these methods, such as anomaly detection, clustering, and classification, as well as potential connections with differential privacy (Dwork, 2008). Though we have focused in this work on unconditional density estimation tasks, it is straightforward to compute arbitrary conditional probabilities with ARFs by reducing the event space to just those leaves that satisfy some logical constraint(s). More complex functionals may be estimated with just a few additional steps—e.g. (conditional) quantiles, CDFs, and copulas—thereby linking these methods with recent work on functional regression with random forests (Hothorn and Zeileis, 2021; Fu et al., 2021; Čevič et al., 2022). Alternative tree-based solutions based on gradient boosting also warrant further exploration, especially given promising recent developments in this area (Friedman, 2020; Gao and Hastie, 2022).

## Acknowledgments

MNW and KB received funding from the German Research Foundation (DFG), Emmy Noether Grant 437611051. MNW and JK received funding from the U Bremen Research Alliance/AI Center for Health Care, financially supported by the Federal State of Bremen. We are grateful to Cassio de Campos, Gennaro Gala, Robert Peharz, and Alvaro H.C. Correia for their feedback on an earlier draft of this manuscript.

## References

- Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, page 214–223.
- Athey, S., Tibshirani, J., and Wager, S. (2019). Generalized random forests. *Ann. Statist.*, 47(2):1148–1178.
- Augenstein, S., McMahan, H. B., Ramage, D., Ramaswamy, S., Kairouz, P., Chen, M., Mathews, R., and y Arcas, B. A. (2020). Generative models for effective ML on private, decentralized datasets. In *International Conference on Learning Representations*.
- Bach, F. R. and Jordan, M. I. (2003). Beyond independent components: Trees and clusters. *J. Mach. Learn. Res.*, 4:1205–1233.
- Belgiu, M. and Drăguț, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.*, 114:24–31.
- Biau, G. (2012). Analysis of a random forests model. *J. Mach. Learn. Res.*, 13:1063–1095.
- Biau, G. and Devroye, L. (2010). On the layered nearest neighbour estimate, the bagged nearest neighbour estimate and the random forest method in regression and classification. *J. Multivar. Anal.*, 101(10):2499–2518.
- Biau, G., Devroye, L., and Lugosi, G. (2008). Consistency of random forests and other averaging classifiers. *J. Mach. Learn. Res.*, 9(66):2015–2033.
- Biau, G. and Scornet, E. (2016). A random forest guided tour. *TEST*, 25(2):197–227.
- Blackard, J. (1998). Coverttype. UCI Machine Learning Repository.
- Bramer, M. (2007). *Clustering*, pages 221–238. Springer, London, UK.
- Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):1–33.
- Breiman, L. (2004). Consistency for a simple model of random forests. Technical Report 670, Statistics Department, UC Berkeley, Berkeley, CA.
- Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees*. Taylor & Francis, Boca Raton, FL.
- Buzhinsky, I., Nerinovsky, A., and Tripakis, S. (2021). Metrics and methods for robustness evaluation of neural networks with generative models. *Mach. Learn.*
- Čevič, D., Michel, L., Näf, J., Bühlmann, P., and Meinhäuser, N. (2022). Distributional random forests: Heterogeneity adjustment and multivariate distributional regression. *J. Mach. Learn. Res.*, 23(333).
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3).
- Chen, X. and Ishwaran, H. (2012). Random forests for genomic data analysis. *Genomics*, 99(6):323–329.
- Choi, E., Biswal, S., Malin, B., Duke, J., Stewart, W. F., and Sun, J. (2017). Generating multi-label discrete patient records using generative adversarial networks. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, pages 286–305.
- Choi, Y., Vergari, A., and Van den Broeck, G. (2020). Probabilistic circuits: A unifying framework for tractable probabilistic models. Technical Report, University of California, Los Angeles.
- Chow, C. and Liu, C. (1968). Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theory*, 14(3):462–467.
- Correia, A., Gala, G., Quaegebeur, E., de Campos, C., and Peharz, R. (2023). Continuous mixtures of tractable probabilistic models. Proceedings of the 37th AAAI Conference.
- Correia, A., Peharz, R., and de Campos, C. P. (2020). Joints in random forests. In *Advances in Neural Information Processing Systems*, volume 33, pages 11404–11415.
- Criminisi, A., Shotton, J., and Konukoglu, E. (2012). *Decision Forests: A Unified Framework for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning*, volume 7, pages 81–227. NOW Publishers, Norwell, MA.
- Cutler, D. R., Edwards Jr., T. C., Beard, K. H., Cutler, A., Hess, K. T., Gibson, J., and Lawler, J. J. (2007). Random forests for classification in ecology. *Ecology*, 88(11):2783–2792.
- Dang, M., Vergari, A., and Van den Broeck, G. (2022). Strudel: A fast and accurate learner of structured-decomposable probabilistic circuits. *Int. J. Approx. Reason.*, 140:92–115.
- Darwiche, A. (2009). *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, New York.
- de Ruiter, A. (2021). The distinct wrong of deepfakes. *Philos. Technol.*, 34(4):1311–1332.
- Denil, M., Matheson, D., and De Freitas, N. (2014). Narrowing the gap: Random forests in theory and in practice. In *Proceedings of the 31st International Conference on Machine Learning*, pages 665–673.
- Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York.
- Diakopoulos, N. and Johnson, D. (2021). Anticipating and addressing the ethical implications of deepfakes in the context of elections. *New Media Soc.*, 23(7):2072–2098.
- Drton, M. and Maathuis, M. H. (2017). Structure learning in graphical modeling. *Annu. Rev. Stat. Appl.*, 4(1):365–393.

- Dua, D. and Graff, C. (2019). UCI machine learning repository.
- Dwork, C. (2008). Differential privacy: A survey of results. In *Theory and Applications of Models of Computation*, volume 4978, pages 1–19, Berlin, Heidelberg. Springer.
- Efron, B. (1994). Missing data, imputation, and the bootstrap. *J. Am. Stat. Assoc.*, 89(426):463–475.
- Esmailpour, M., Chaalia, N., Abusitta, A., Devailly, F.-X., Maazoun, W., and Cardinal, P. (2022). RCC-GAN: Regularized compound conditional gan for large-scale tabular data synthesis. *arXiv preprint*, 2205.11693.
- Feng, J. and Zhou, Z.-H. (2018). Autoencoder by forest. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*.
- Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(90):3133–3181.
- Friedman, J. H. (2020). Contrast trees and distribution boosting. *Proc. Natl. Acad. Sci.*, 117(35):21175–21184.
- Fu, G., Dai, X., and Liang, Y. (2021). Functional random forests for curve response. *Sci. Rep.*, 11(1):24159.
- Gao, W., Xu, F., and Zhou, Z.-H. (2022). Towards convergence rate analysis of random forests for classification. *Artif. Intell.*, 313(C).
- Gao, Z. and Hastie, T. (2022). LinCDE: Conditional density estimation via Lindsey’s method. *J. Mach. Learn. Res.*, 23(52):1–55.
- Genuer, R. (2012). Variance reduction in purely random forests. *J. Nonparametr. Stat.*, 24(3):543–562.
- Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Mach. Learn.*, 63(1):3–42.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, page 2672–2680.
- Gramacki, A. (2018). *Nonparametric Kernel Density Estimation and Its Computational Aspects*. Springer, Cham.
- Gray, A. G. and Moore, A. W. (2003). Nonparametric density estimation: Toward computational tractability. In *Proceedings of the 2003 SIAM International Conference on Data Mining (SDM)*, pages 203–211.
- Györfi, L., Kohler, M., Krzyżak, A., and Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer-Verlag, New York.
- Heckerman, D., Geiger, D., and Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Mach. Learn.*, 20(3):197–243.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017).  $\beta$ -VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.
- Hothorn, T. and Zeileis, A. (2021). Predictive distribution modeling using transformation forests. *J. Comput. Graph. Stat.*, 30(4):1181–1196.
- Jordon, J., Yoon, J., and van der Schaar, M. (2019). PATE-GAN: generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*.
- Kim, I., Ramdas, A., Singh, A., and Wasserman, L. (2021). Classification accuracy as a proxy for two-sample testing. *Ann. Stat.*, 49(1):411 – 434.
- Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. In *Advances in Neural Information Processing Systems*, volume 34, pages 21696–21707.
- Kingma, D. and Welling, M. (2013). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.
- Kisa, D., Van den Broeck, G., Choi, A., and Darwiche, A. (2014). Probabilistic sentential decision diagrams. In *14th International Conference on the Principles of Knowledge Representation and Reasoning*.
- Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979.
- Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models*. The MIT Press, Cambridge, MA.
- Lauritzen, S. L. (1996). *Graphical Models*. Oxford Statistical Science Series. Clarendon Press, Oxford.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.
- Lee, J., Hyeong, J., Jeon, J., Park, N., and Cho, J. (2021). Invertible tabular GANs: Killing two birds with one stone for tabular data synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 4263–4273.
- Lee, J., Kim, M., Jeong, Y., and Ro, Y. (2022). Differentially private normalizing flows for synthetic tabular data generation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7345–7353.
- Liu, H., Xu, M., Gu, H., Gupta, A., Lafferty, J., and Wasserman, L. (2011). Forest density estimation. *J. Mach. Learn. Res.*, 12(25):907–951.
- Loh, W.-Y. (2009). Improving the precision of classification trees. *Ann. Appl. Stat.*, 3(4):1710 – 1737.
- Lopez, R., Regier, J., Cole, M. B., Jordan, M. I., and Yosef, N. (2018). Deep generative modeling for single-cell transcriptomics. *Nat. Methods*, 15(12):1053–1058.

- Lugosi, G. and Nobel, A. (1996). Consistency of data-driven histogram methods for density estimation and classification. *Ann. Stat.*, 24(2):687 – 706.
- Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Lee, S.-I. (2020). From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.*, 2(1):56–67.
- Malley, J., Kruppa, J., Dasgupta, A., Malley, K., and Ziegler, A. (2012). Probability machines: Consistent probability estimation using nonparametric learning machines. *Methods Inf. Med.*, 51(1):74–81.
- Mandelbrot, B. B. (1982). *The Fractal Geometry of Nature*. W.H. Freeman & Co., New York.
- Meinshausen, N. (2006). Quantile regression forests. *J. Mach. Learn. Res.*, 7:983–999.
- Mentch, L. and Hooker, G. (2016). Quantifying uncertainty in random forests via confidence intervals and hypothesis tests. *J. Mach. Learn. Res.*, 17(26).
- Mentch, L. and Zhou, S. (2020). Randomization as regularization: A degrees of freedom explanation for random forest success. *J. Mach. Learn. Res.*, 21(171).
- Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint*, 1411.1784.
- Öhman, C. (2020). Introducing the pervert’s dilemma: a contribution to the critique of deepfake pornography. *Ethics Inf. Technol.*, 22(2):133–140.
- Pang, G., Shen, C., Cao, L., and Hengel, A. V. D. (2021). Deep learning for anomaly detection: A review. *ACM Comput. Surv.*, 54(2).
- Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *J. Mach. Learn. Res.*, 22(57):1–64.
- Pearl, J. and Russell, S. (2003). Bayesian networks. In Arbib, M. A., editor, *Handbook of Brain Theory and Neural Networks*, pages 157–160. The MIT Press, Cambridge, MA.
- Peharz, R., Lang, S., Vergari, A., Stelzner, K., Molina, A., Trapp, M., Van Den Broeck, G., Kersting, K., and Ghahramani, Z. (2020a). Einsum networks: Fast and scalable learning of tractable probabilistic circuits. In *Proceedings of the 37th International Conference on Machine Learning*, pages 7563–7574.
- Peharz, R., Vergari, A., Stelzner, K., Molina, A., Shao, X., Trapp, M., Kersting, K., and Ghahramani, Z. (2020b). Random sum-product networks: A simple and effective approach to probabilistic deep learning. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, volume 115, pages 334–344.
- Peng, W., Coleman, T., and Mentch, L. (2022). Rates of convergence for random forests via generalized U-statistics. *Electron. J. Stat.*, 16(1):232 – 292.
- Poon, H. and Domingos, P. (2011). Sum-product networks: A new deep architecture. In *Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence*, page 337–346.
- Probst, P. and Boulesteix, A.-L. (2017). To tune or not to tune the number of trees in random forest. *J. Mach. Learn. Res.*, 18(1):6673–6690.
- Probst, P., Wright, M. N., and Boulesteix, A.-L. (2019). Hyperparameters and tuning strategies for random forest. *WIREs Data Mining and Knowledge Discovery*, 9(3):e1301.
- Rahman, T., Kothalkar, P., and Gogate, V. (2014). Cutset networks: A simple, tractable, and scalable approach for improving the accuracy of Chow-Liu trees. In *Machine Learning and Knowledge Discovery in Databases*, pages 630–645, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Ram, P. and Gray, A. G. (2011). Density estimation trees. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 627–635.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with CLIP latents. *arXiv preprint*, 2204.06125.
- Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *arXiv preprint*, 2102.12092.
- Ravuri, S. and Vinyals, O. (2019). Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems*, volume 32.
- Rokach, L. and Maimon, O. (2005). *Clustering Methods*, pages 321–352. Springer US, Boston, MA.
- Roy, A., Saffar, M., Vaswani, A., and Grangier, D. (2021). Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.
- Rubin, D. B. (1996). Multiple imputation after 18+ years. *J. Am. Stat. Assoc.*, 91(434):473–489.
- Santurkar, S., Schmidt, L., and Madry, A. (2018). A classification-based study of covariate shift in GAN distributions. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 4480–4489.
- Scornet, E. (2016). On the asymptotics of random forests. *J. Multivar. Anal.*, 146:72–83.
- Scornet, E. (2017). Tuning parameters in random forests. *ESAIM: Procs*, 60:144–162.
- Scornet, E., Biau, G., and Vert, J. P. (2015). Consistency of random forests. *Ann. Statist.*, 43(4):1716–1741.



- Shi, T. and Horvath, S. (2006). Unsupervised learning with random forest predictors. *J. Comput. Graph. Stat.*, 15(1):118–138.
- Shmelkov, K., Schmid, C., and Alahari, K. (2018). How good is my GAN? In *Computer Vision – ECCV 2018*, pages 218–234, Cham. Springer International Publishing.
- Silverman, B. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.
- Smyth, P., Gray, A. G., and Fayyad, U. M. (1995). Retrofitting decision tree classifiers using kernel density estimation. In *Proceedings of the 12th International Conference on International Conference on Machine Learning*, page 506–514.
- Song, Y., Shu, R., Kushman, N., and Ermon, S. (2018). Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, volume 31.
- Song, Y., Sohl-Dickstein, J., Kingma, D., Kumar, A., Erman, S., and Poole, B. (2021). Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.
- Stadler, T., Oprisanu, B., and Troncoso, C. (2022). Synthetic data – anonymisation groundhog day. In *31st USENIX Security Symposium*, pages 1451–1468.
- Stekhoven, D. J. and Bühlmann, P. (2011). MissForest—non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118.
- Stone, C. J. (1977). Consistent nonparametric regression. *Ann. Statist.*, 5(4):595 – 620.
- Tang, C., Garreau, D., and von Luxburg, U. (2018). When do random forests fail? In *Advances in Neural Information Processing Systems*, volume 31.
- Tang, F. and Ishwaran, H. (2017). Random forest missing data algorithms. *Stat. Anal. Data Min.*, 10(6):363–377.
- van den Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. (2016). Pixel recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1747–1756.
- Van Haaren, J. and Davis, J. (2012). Markov network structure learning: A randomized feature generation approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 26, pages 1148–1154.
- Vapnik, V. and Chervonenkis, A. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264 – 280.
- Vergari, A., Choi, Y., Peharz, R., and Van den Broeck, G. (2020). Probabilistic circuits: Representations, inference, learning and applications. In *Tutorial at the 34th AAAI Conference on Artificial Intelligence*.
- Vincent, P. and Bengio, Y. (2002). Manifold Parzen windows. In *Advances in Neural Information Processing Systems*, volume 15.
- Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *J. Am. Stat. Assoc.*, 113(523):1228–1242.
- Wager, S. and Walther, G. (2015). Adaptive concentration of regression trees, with application to random forests. *arXiv preprint*, 1503.06388.
- Wand, M. and Jones, M. (1994). *Kernel Smoothing*. Chapman & Hall, Boca Raton, FL.
- Weierstrass, K. (1895). Über continuirliche Functionen eines reellen Arguments, die für keinen Werth des letzteren einen bestimmten Differentialquotienten besitzen. In *Mathematische Werke von Karl Weierstrass*, pages 71–74. Mayer & Mueller, Berlin.
- Wen, H. and Hang, H. (2022). Random forest density estimation. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23701–23722.
- Worldline and the ML Group of ULB (2013). Credit card fraud detection data. license: Open database.
- Wu, K., Zhang, K., Fan, W., Edwards, A., and Yu, P. S. (2014). RS-Forest: A rapid density estimator for streaming anomaly detection. In *2014 IEEE International Conference on Data Mining*, pages 600–609.
- Xu, L., Skoularidou, M., Cuesta-Infante, A., and Veeramachaneni, K. (2019). Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*, volume 32.
- Yang, J., Kannan, A., Batra, D., and Parikh, D. (2017). LR-GAN: Layered recursive generative adversarial networks for image generation. In *International Conference on Learning Representations*.
- Yelmen, B., Decelle, A., Ongaro, L., Marnetto, D., Tallec, C., Montinaro, F., Furtlehner, C., Pagani, L., and Jay, F. (2021). Creating artificial human genomes using generative neural networks. *PLOS Genetics*, 17(2):1–22.
- Zhang, H., Juba, B., and Van Den Broeck, G. (2021). Probabilistic generating circuits. In *Proceedings of the 38th International Conference on Machine Learning*, pages 12447–12457.
- Zhao, Z., Kunar, A., Birke, R., and Chen, L. Y. (2022). CTAB-GAN+: Enhancing tabular data synthesis. *arXiv preprint*, 2204.00401.

## A PROOFS

### A.1 Proof of Thm. 1

To secure the result, we must show that (a) the discriminator reliably converges on the Bayes risk at each iteration  $t$ ; and (b) the generator’s sampling strategy drives original and synthetic data closer together, ultimately taking the Bayes risk to  $1/2$  as  $n, t \rightarrow \infty$ . (For the purposes of this proof, we set the tolerance parameter  $\delta$  to 0.)

Take (a) first. This amounts to a consistency requirement for RFs. The consistency of RF classifiers has been demonstrated under various assumptions about splitting rules and stopping criteria (Breiman, 2004; Biau et al., 2008; Biau and Devroye, 2010; Gao et al., 2022), but these results generally require trees to be grown to purity or even completion (i.e.,  $n_b^\ell = 1$  for all  $\ell, b$ ). However, this would turn the generator’s sampling strategy into a simple copy-paste operation and make intra-leaf density estimation impossible. We therefore follow Malley et al. (2012) in observing that regression procedures constitute probability machines, since  $P(Y = 1|\mathbf{x}) = \mathbb{E}[Y|\mathbf{x}]$  for  $Y \in \{0, 1\}$ .

For simplicity, we focus on the single tree case, as the consistency of the ensemble follows from the consistency of the base method (Biau et al., 2008). We define  $\eta^{(t)}(\mathbf{x}) := P(Y = 1|\mathbf{x}, t)$  as the target function for fixed  $t$ . Let  $f_n^{(t)}(\mathbf{x})$  be a tree trained according to (A1)-(A3) on a sample of size  $n$  at iteration  $t$ . Since  $L_2$ -consistency entails classifier consistency using the soft labeling approach of (A3).(vi), our goal in this section is to show that, for all  $t \in \mathbb{N}$ , we have:

$$\lim_{n \rightarrow \infty} \mathbb{E} \left[ \left( f_n^{(t)}(\mathbf{x}) - \eta^{(t)}(\mathbf{x}) \right)^2 \right] = 0.$$

Consistency for RF regression has been established for several variants of the algorithm, occasionally under some constraints on the data generating process (Genuer, 2012; Scornet et al., 2015; Wager and Walther, 2015; Biau and Scornet, 2016). Recent work in this area has tended to focus on asymptotic normality (Mentch and Hooker, 2016; Wager and Athey, 2018; Athey et al., 2019; Peng et al., 2022), which requires additional assumptions. These often include an upper bound on leaf sample size, which would complicate our analysis in Thm. 2. To avoid unnecessary difficulties, we borrow selectively from Meinshausen (2006), Biau (2012), Denil et al. (2014), Scornet (2016), and Wager and Athey (2018), striking a delicate balance between theoretical parsimony and fidelity to the classic RF algorithm.<sup>5</sup>

For full details, we refer readers to the original texts. The main point to recognize is that, under assumptions (A1)-(A3), RFs satisfy the conditions of Stone’s theorem (Stone, 1977), which guarantees the universal consistency of a large family of local averaging methods. Devroye et al. (1996, Thm. 6.1) and Györfi et al. (2002, Thm. 4.2) show that partitioning estimators such as decision trees qualify provided that (1)  $\text{diam}(\mathcal{X}_\ell) \rightarrow_p 0$  and (2)  $n_\ell \rightarrow_p \infty$  for all  $\ell$  as  $n \rightarrow \infty$ , effectively creating leaves of infinite density. The former is derived by Meinshausen (2006, Lemma 2) under (A3).(iii) and (A3).(iv); the latter follows trivially from (A3).(v). Thus RF discriminators weakly converge on the Bayes risk in the large sample limit, completing part (a) of the proof.

Desideratum (b) effectively says that original and synthetic data become indistinguishable as  $n$  and  $t$  increase. Recall that at  $t = 0$ , we generate synthetic data  $\tilde{\mathbf{X}}^{(0)} \sim \prod_{j=1}^d P(X_j)$ , which becomes input to the discriminator  $f_n^{(0)}$ . Let  $\theta^{(0)}$  denote the resulting splits once the discriminator has converged. In subsequent rounds, synthetic data are sampled according to  $\tilde{\mathbf{X}}^{(t+1)} \sim \prod_{j=1}^d P(X_j|\theta_\ell^{(t)})P(\theta_\ell^{(t)})$ . (The consistency of coverage estimates is treated separately in Appx. A.3.) We proceed to train a new discriminator and repeat the process.

Let  $P^*$  be the target distribution and  $P^{(t)}$  the synthetic distribution at round  $t$ . For all  $t \geq 1$ , the input data to the discriminator  $f_n^{(t)}$  is the dataset  $\mathcal{D}_n^{(t)} \sim 0.5P^* + 0.5P^{(t-1)}$ . Our goal in this section is to show that, as  $n, t \rightarrow \infty$ :

$$\sup_{\mathbf{x} \in \mathcal{D}_n^{(t)}} |\eta^{(t)}(\mathbf{x}) - 1/2| \rightarrow_p 0.$$

An apparent challenge to our recursive strategy for generating synthetic data is posed by self-similar distributions, in which dependencies replicate at ever finer resolutions, as in some fractal equations (Mandelbrot, 1982). For instance, let  $g$  be the Weierstrass function (Weierstrass, 1895), and say that  $X_2 = g(X_1)$ . Then the generative model will tend to produce

<sup>5</sup>Several authors have conjectured that RF consistency may not require honesty (A3).(i) or subsampling (A3).(ii) after all. Empirical performance certainly seems unencumbered by these requirements. However, both come with major theoretical advantages—the former by making predictions conditionally independent of the training data while preserving some form of adaptive splits, the latter by avoiding thorny issues arising from duplicated samples when bootstrapping. See Biau (2012, Rmk. 8), Wager and Athey (2018, Appx. B), and Tang et al. (2018) for a discussion.

off-manifold data at each iteration  $t$ , no matter how small  $\text{vol}(\mathcal{X}_\ell)$  becomes. However, this only shows that convergence can fail for finite  $t$ . Since the discriminator is consistent, it will accurately identify synthetic points in round  $t + 1$ , pruning the space still further.

Let  $[L^{(t)}]$  be the leaves of the discriminator  $f_n^{(t)}$ , and define the maximum leaf diameter  $m_t := \max_{\ell \in [L^{(t)}]} \text{diam}(\mathcal{X}_\ell)$ . We say that two samples are *neighbors* in  $f_n^{(t)}$  if the model places them in the same leaf. We show that, as  $n, t \rightarrow \infty$ , conditional probabilities for neighboring samples converge—including, crucially, original and synthetic counterparts. Our Lipschitz condition (A2) states that for all  $\mathbf{x}, \mathbf{x}'$ , we have:

$$|\eta^{(t)}(\mathbf{x}) - \eta^{(t)}(\mathbf{x}')| \leq c_t \|\mathbf{x} - \mathbf{x}'\|_2,$$

where  $c_t$  denotes the Lipschitz constant at round  $t$ . Suppose that  $\mathbf{x}$  and  $\mathbf{x}'$  are neighbors. Then we can replace the second factor on the rhs with  $m_t$ , since the  $L_2$  distance between neighbors cannot exceed the maximum leaf diameter at round  $t$ . Meinshausen (2006)'s aforementioned Lemma 2 ensures that this value goes to zero in probability as rounds increase. This could in principle be offset by a sufficient increase in  $c_t$  over training rounds, but the second condition of (A2) prevents this, imposing the constraint that  $c_t = o(m_t^{-1})$ . Thus, for observations in the same leaf,  $c_t m_t \rightarrow_p 0$  as  $t \rightarrow \infty$ . Because original and synthetic samples are equinumerous in all leaves following the generative step, each original sample has a synthetic counterpart to which it is arbitrarily close in  $L_2$  space as  $t$  grows large. Since no feature values are sufficient to distinguish between the two classes in any region, all conditional probabilities go to  $1/2$ , and Bayes risk therefore also converges to  $1/2$  in probability. This concludes the proof.

## A.2 Proof of Lemma 1

Define the first-order approximation to  $p$  satisfying local independence:

$$\hat{p}(\mathbf{x}) := \frac{1}{B} \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} p(\theta_b^\ell) \prod_{j=1}^d p(x_j | \theta_b^\ell).$$

We also define the root integrated squared error (RISE), i.e. the Euclidean distance between probability densities:

$$\text{RISE}(p, q) := \left( \int_{\mathcal{X}} (p(\mathbf{x}) - q(\mathbf{x}))^2 d\mathbf{x} \right)^{1/2}.$$

By the triangle inequality, we have:

$$\text{RISE}(p, q) \leq \text{RISE}(p, \hat{p}) + \text{RISE}(\hat{p}, q).$$

Squaring both sides, we get:

$$\text{ISE}(p, q) \leq \text{ISE}(p, \hat{p}) + \text{ISE}(\hat{p}, q) + 2 \text{RISE}(p, \hat{p}) \text{RISE}(\hat{p}, q).$$

Adding a nonnegative value to the rhs, we can reduce the expression:

$$\begin{aligned} \text{ISE}(p, q) &\leq \text{ISE}(p, \hat{p}) + \text{ISE}(\hat{p}, q) + 2 \text{RISE}(p, \hat{p}) \text{RISE}(\hat{p}, q) + (\text{RISE}(p, \hat{p}) - \text{RISE}(\hat{p}, q))^2 \\ &= 2(\text{ISE}(p, \hat{p}) + \text{ISE}(\hat{p}, q)). \end{aligned}$$

Now observe that we can rewrite both ISE formulae in terms of our predefined residuals (Eqs. 3-5):

$$\begin{aligned} \text{ISE}(p, \hat{p}) &= \int_{\mathcal{X}} \left( \frac{1}{B} \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} (p(\mathbf{x} | \theta_b^\ell) p(\theta_b^\ell) - \prod_{j=1}^d p(x_j | \theta_b^\ell) p(\theta_b^\ell)) \right)^2 d\mathbf{x} \\ &= \frac{1}{B^2} \int_{\mathcal{X}} \left( \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} p(\theta_b^\ell) \epsilon_3 \right)^2 d\mathbf{x}. \\ \text{ISE}(\hat{p}, q) &= \int_{\mathcal{X}} \left( \frac{1}{B} \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} \left( \prod_{j=1}^d p(x_j | \theta_b^\ell) p(\theta_b^\ell) - \prod_{j=1}^d q(x_j; \theta_{b,j}^\ell) q(\theta_b^\ell) \right) \right)^2 d\mathbf{x} \\ &= \frac{1}{B^2} \int_{\mathcal{X}} \left( \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} (p(\theta_b^\ell) \epsilon_2 + \epsilon_1 \prod_{j=1}^d p(x_j | \theta_b^\ell) - \epsilon_1 \epsilon_2) \right)^2 d\mathbf{x}. \end{aligned}$$

We replace the interior squared terms for ease of presentation:

$$\alpha := \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} p(\theta_b^\ell) \epsilon_3$$

$$\beta := \sum_{\ell, b: \mathbf{x} \in \mathcal{X}_b^\ell} \left( p(\theta_b^\ell) \epsilon_2 + \epsilon_1 \prod_{j=1}^d p(x_j | \theta_b^\ell) - \epsilon_1 \epsilon_2 \right).$$

Finally, we take expectations on both sides:

$$\text{MISE}(p, q) \leq 2B^{-2} \mathbb{E} \left[ \int_{\mathcal{X}} \alpha^2 + \beta^2 d\mathbf{x} \right],$$

where we have exploited the linearity of expectation to pull the factor outside of the bracketed term, and the monotonicity of expectation to preserve the inequality.

### A.3 Proof of Theorem 2

Lemma 1 states that error is bounded by a quadratic function of  $\epsilon_1, \epsilon_2, \epsilon_3$ . Thus for  $L_2$ -consistency, it suffices to show that  $\mathbb{E}[\epsilon_j^2] \rightarrow 0$ , for  $j \in \{1, 2, 3\}$ . Since this is already established by Thm. 1 for  $j = 3$ , we focus here on errors of coverage and density. Start with  $\epsilon_1$ . A general version of the Glivenko-Cantelli theorem (Vapnik and Chervonenkis, 1971) guarantees uniform convergence of empirical proportions to population proportions. Let  $\mathcal{L}$  denote the set of all possible hyperrectangular subspaces induced by axis-aligned splits on  $\mathcal{X}$ . Then the following holds with probability 1:

$$\lim_{n \rightarrow \infty} \sup_{\ell \in \mathcal{L}} |p(\theta^\ell) - q_n(\theta^\ell)| = 0.$$

Next, take  $\epsilon_2$ . (A4) guarantees that  $p$  satisfies the consistency conditions for univariate KDE (Silverman, 1986; Wand and Jones, 1994; Gramacki, 2018), while condition (v) of (A3) ensures that within-leaf sample size increases even as leaf volume goes to zero (Meinshausen, 2006, Lemma 2). Our kernel is a nonnegative function  $K : \mathbb{R}^d \rightarrow \mathbb{R}$  that integrates to 1, parametrized by the bandwidth  $h$ :

$$p_h(\mathbf{x}) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right).$$

Using standard arguments, we take a Taylor series expansion of the MISE and minimize the *asymptotic* MISE (AMISE):

$$\text{AMISE}(p, p_h) = \frac{1}{nh} R(K) + \frac{1}{4} h^4 \mu_2(K)^2 R(p''),$$

where

$$R(K) = \int K(x)^2 dx,$$

$$\mu_2(K) = \int x^2 K(x) dx, \text{ and}$$

$$R(p'') = \int p''(x)^2 dx.$$

For example values of these variables under specific kernels, see (Wand and Jones, 1994, Appx. B). Under (A4), it can be shown that

$$\text{MISE}(p, p_h) = \text{AMISE}(p, p_h) + o((nh)^{-1} + h^4).$$

Thus if  $(nh)^{-1} \rightarrow 0$  and  $h \rightarrow 0$  as  $n \rightarrow \infty$ , the asymptotic approximation is exact and  $\mathbb{E}[\epsilon_2^2] \rightarrow 0$ .

These results, combined with the proof of Thm. 1 (see Appx. A.1), establish that errors of coverage, density, and convergence all vanish in the limit. Thus  $\mathbb{E}[\epsilon_j^2] \rightarrow 0$  for  $j \in \{1, 2, 3\}$ , and the proof is complete.

## B EXPERIMENTS

Our experiments do not include any personal data, as defined in Article 4(1) of the European Union’s General Data Protection Regulation. All data are either simulated or from publicly available resources. We performed all experiments on a dedicated 64-bit Linux platform running Ubuntu 20.04 with an AMD Ryzen Threadripper 3960X (24 cores, 48 threads) CPU, 256 gigabyte RAM and two NVIDIA Titan RTX GPUs. We used R version 4.1.2 and Python version 3.7.12. Further details on the environment setup are provided in the supplemental code.

### B.1 Simulations

The `cassini`, `shapes`, and `smiley` simulations are all available in the `mlbench` R package; the `twomoons` problem is available in the `fdm2id` R package. Default parameters were used throughout, with fixed sample size  $n = 2000$ .

### B.2 Twenty Datasets

The Twenty Datasets benchmark was originally proposed by Van Haaren and Davis (2012). A conventional training/validation/test split is widely used in the PC literature. Because our method does not include any hyperparameter search, we combine training and validation sets into a single training set. We downloaded the data from <https://github.com/joshuacnf/Probabilistic-Generating-Circuits/tree/main/data> and include the directory in our project GitHub repository for completeness. All datasets are Boolean, with sample size and dimensionality given in Table 3.

Table 3: Summary of datasets included in the Twenty Datasets benchmark.

Dataset	Train	Validation	Test	Dimensions
<code>nltns</code>	16181	2157	3236	16
<code>msnbc</code>	291326	38843	58265	17
<code>kdd</code>	180092	19907	34955	64
<code>plants</code>	17412	2321	3482	69
<code>audio</code>	15000	2000	3000	100
<code>jester</code>	9000	1000	4116	100
<code>netflix</code>	15000	2000	3000	100
<code>accidents</code>	12758	1700	2551	111
<code>retail</code>	22041	2938	4408	135
<code>pumsb</code>	12262	1635	2452	163
<code>dna</code>	1600	400	1186	180
<code>kosarek</code>	33375	4450	6675	190
<code>msweb</code>	29441	3270	5000	294
<code>book</code>	8700	1159	1739	500
<code>movie</code>	4524	1002	591	500
<code>webkb</code>	2803	558	838	839
<code>reuters</code>	6532	1028	1540	889
<code>20ng</code>	11293	3764	3764	910
<code>bbc</code>	1670	225	330	1058
<code>ad</code>	2461	327	491	1556

Results for competitors are reported in the cited papers:

- Einsum networks (Peharz et al., 2020a)
- Random sum-product networks (Peharz et al., 2020b)
- Probabilistic generating circuits (Zhang et al., 2021)
- Strudel (Dang et al., 2022)
- Continuous mixtures of Chow-Liu trees (Correia et al., 2023).

### B.3 Tabular GANs

For benchmarking generative models on real-world data, we use the benchmarking pipeline proposed by Xu et al. (2019). In detail, the workflow is as follows:

1. Load classification datasets used in Xu et al. (2019), namely `adult`, `census`, `credit`, `covertype`, `intrusion`, `mnist12`, and `mnist28`. Note that the type of prediction task does not affect the process of synthetic data generation, so we omit the single regression example (`news`) for greater consistency.

---

### Adversarial Random Forests

---

2. Split the data into training and test sets (see Table 4 for details).
3. Train the generative models FORGE (number of trees = 10, minimum node size = 5), CTGAN<sup>6</sup> (batch size = 500, epochs = 300), TVAE<sup>7</sup> (batch size = 500, epochs = 300), CTAB-GAN+<sup>8</sup> (batch size = 500, epochs = 150), IT-GAN<sup>9</sup> (batch size = 2000, epochs = 300) and RCC-GAN<sup>10</sup> (batch size = 500, epochs = 300).
4. Generate a synthetic dataset of the same size as the training set using each of the generative models trained in step (3), measuring the wall time needed to execute this task.
5. Train a set of supervised learning algorithms (see Table 4 for details): (a) on the real training data set (i.e., the *Oracle*); and (b) on the synthetic training datasets generated by FORGE, CTGAN, TVAE, CTAB-GAN+ and RCC-GAN.
6. Evaluate the performance of the learning algorithms from step (5) on the test set.
7. For each dataset, average performance metrics (accuracy, F1-scores) across learners. We report F1-scores for the positive class, e.g. '>50k' for *adult*, '+50000' for *census* and '1' for *credit*.

Table 4: Benchmark Setup. Supervised learning algorithms for prediction: (A) Adaboost, estimators = 50, (B) Decision Tree, tree depth for binary/multiclass target = 15/30, (C) Logistic Regression, (D) MLP, hidden layers for binary/multiclass target = 50/100

Dataset	Train/Test	Learner	Link to dataset
<i>adult</i> (Dua and Graff, 2019)	23k/10k	A,B,C,D	<a href="http://archive.ics.uci.edu/ml/datasets/adult">http://archive.ics.uci.edu/ml/datasets/adult</a>
<i>census</i> (Dua and Graff, 2019)	200k/100k	A,B,D	<a href="https://archive.ics.uci.edu/ml/datasets/census+income">https://archive.ics.uci.edu/ml/datasets/census+income</a>
<i>covertype</i> (Blackard, 1998)	481k/100k	A,D	<a href="https://archive.ics.uci.edu/ml/datasets/covertype">https://archive.ics.uci.edu/ml/datasets/covertype</a>
<i>credit</i> (Worldline and the ML Group of ULB, 2013)	264k/20k	A,B,D	<a href="https://www.kaggle.com/mlg-ulb/creditcardfraud">https://www.kaggle.com/mlg-ulb/creditcardfraud</a>
<i>intrusion</i> (Dua and Graff, 2019)	394k/100k	A,D	<a href="http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data">http://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data</a>
<i>mnist12</i> (LeCun et al., 1998)	60k/10k	A,D	<a href="http://yann.lecun.com/exdb/mnist/index.html">http://yann.lecun.com/exdb/mnist/index.html</a>
<i>mnist28</i> (LeCun et al., 1998)	60k/10k	A,D	<a href="http://yann.lecun.com/exdb/mnist/index.html">http://yann.lecun.com/exdb/mnist/index.html</a>

#### B.4 Run Time

In order to evaluate the run time efficiency of FORGE, we chose to focus on the smallest dataset of the benchmark study in Sect. 5.2, namely *adult*. We (A) drew stratified subsamples and (B) drew covariate subsets. For step (B), the target variable is always included. We select an equal number of continuous/categorical covariates when possible and use all  $n = 32,561$  instances. Results in terms of processing time are visualized in Fig. 4.

#### B.5 Image Data

We include results on the *mnist12* and *mnist28* datasets here, both included in the original Xu et al. (2019) pipeline. Benchmarking against CTGAN and TVAE (other methods proved too slow to test), we find that FORGE outperforms both competitors in accuracy, F1-score, and speed (see Table 5).

However, since MNIST is not a tabular data problem, perhaps a more relevant comparison would be against convolutional networks specifically designed for image data. We train a conditional GAN with convolutional layers (Mirza and Osindero, 2014) and find that the resulting cGAN clearly outperforms FORGE (see Fig. 5). This result is expected, given that our method is not optimized for image data. It also illustrates a limitation of our approach, which excels in speed and flexibility but is no match for deep learning methods on structured datasets.

<sup>6</sup>[https://sdv.dev/SDV/api\\_reference/tabular/ctgan.html](https://sdv.dev/SDV/api_reference/tabular/ctgan.html). MIT License.

<sup>7</sup>[https://sdv.dev/SDV/api\\_reference/tabular/tvae.html](https://sdv.dev/SDV/api_reference/tabular/tvae.html). MIT License.

<sup>8</sup><https://github.com/Team-TUD/CTAB-GAN-Plus>

<sup>9</sup><https://github.com/leejaehoon2016/ITGAN>. Samsung SDS Public License V1.0.

<sup>10</sup><https://github.com/EsmaeilpourMohammad/RccGAN>

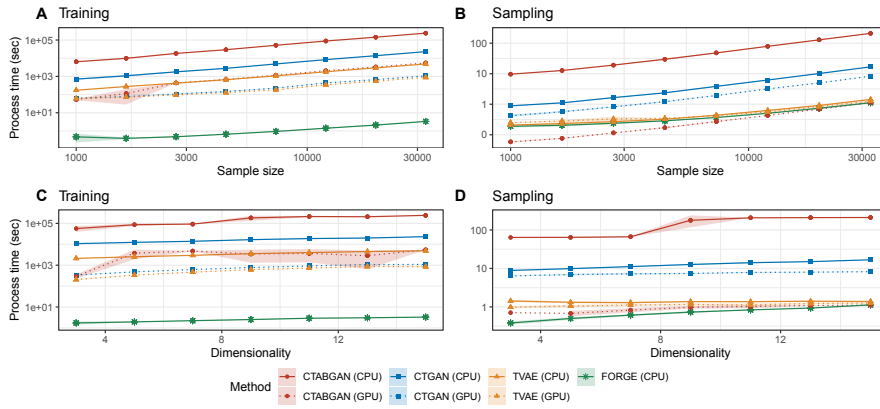


Figure 4: Complexity curves. (A): Processing time as a function of sample size, using stratified subsamples of the `adult` dataset. (B): Processing time as a function of dimensionality, using random features from the `adult` dataset.

Table 5: Performance on `mnist` datasets from the Xu et al. (2019) benchmark for CTGAN and TVAE vs. FORGE We report average results across five replicates  $\pm$  the associated standard error. Winning results in bold.

Dataset	Model	Accuracy $\pm$ SE	F1 $\pm$ SE	Time (sec)
<code>mnist12</code> classes = 10 $n = 70,000$ $d = 144$	Oracle	$0.892 \pm 0.003$	$0.891 \pm 0.003$	
	FORGE	<b><math>0.799 \pm 0.007</math></b>	<b><math>0.795 \pm 0.007</math></b>	<b>32.3</b>
	CTGAN	$0.172 \pm 0.032$	$0.138 \pm 0.032$	2737.4
	TVAE	$0.763 \pm 0.002$	$0.761 \pm 0.002$	1143.8
<code>mnist28</code> classes = 10 $n = 70,000$ $d = 784$	Oracle	$0.918 \pm 0.002$	$0.917 \pm 0.002$	
	FORGE	<b><math>0.729 \pm 0.008</math></b>	<b><math>0.723 \pm 0.008</math></b>	<b>169.5</b>
	CTGAN	$0.197 \pm 0.051$	$0.167 \pm 0.051$	14415.4
	TVAE	$0.698 \pm 0.016$	$0.697 \pm 0.016$	5056.0

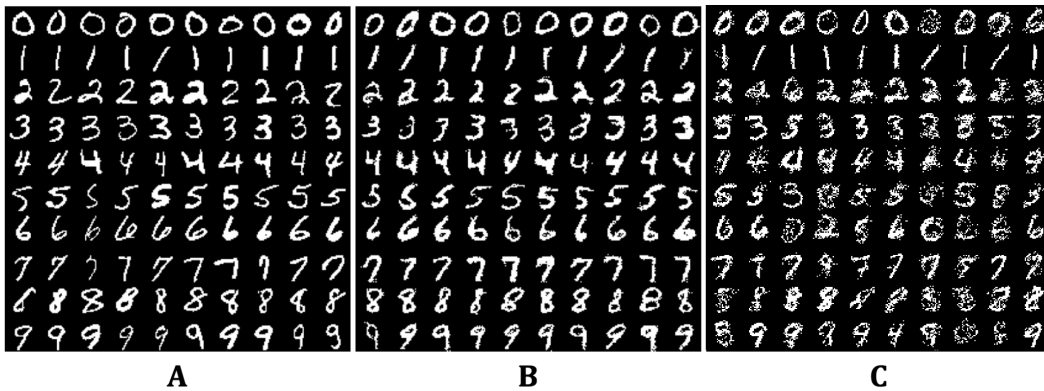


Figure 5: Results from `mnist28` experiment. (A): Original samples. (B): Samples generated by cGAN. (C): Samples generated by FORGE.





# Paper 4. Arfpy: A Python Package for Density Estimation and Generative Modeling with Adversarial Random Forests

**Contributing Article:** Blesch, K., and Wright, M. N. (2023) Arfpy: A python package for density estimation and generative modeling with adversarial random forests. *ArXiv Preprint arXiv:2311.07366*. <https://arxiv.org/abs/2311.07366>.

**Copyright information:** Copyright 2023 by the authors. Creative Commons Attribution 4.0 International License (CC-BY 4.0). International License (CC-BY 4.0).

**Author contributions:** Kristin Blesch initiated the project, lead the software development of the *Python* package, its deployment and documentation and drafted the manuscript. Marvin N. Wright supervised the project and contributed to the *Python* code. All authors contributed to proofreading and revising the paper.



---

# arfpv: A python package for density estimation and generative modeling with adversarial random forests

Kristin Blesch<sup>\*1,2</sup>      Marvin N. Wright<sup>1,2,3</sup>

<sup>\*</sup> corresponding author; blesch@leibniz-bips.de

<sup>1</sup>Leibniz Institute for Prevention Research and Epidemiology – BIPS, Germany;

<sup>2</sup>Faculty of Mathematics and Computer Science, University of Bremen, Germany;

<sup>3</sup>Department of Public Health, University of Copenhagen, Denmark

## Abstract

This paper introduces `arfpv`, a python implementation of Adversarial Random Forests (ARF) (Watson et al., 2023), which is a lightweight procedure for synthesizing new data that resembles some given data. The software `arfpv` equips practitioners with straightforward functionalities for both density estimation and generative modeling. The method is particularly useful for tabular data and its competitive performance is demonstrated in previous literature. As a major advantage over the mostly deep learning based alternatives, `arfpv` combines the method’s reduced requirements in tuning efforts and computational resources with a user-friendly python interface. This supplies audiences across scientific fields with software to generate data effortlessly.

<https://github.com/bips-hb/arfpv>

## Keywords

Generative Modeling; Density Estimation; Machine Learning

## Introduction

Generative modeling is a challenging task in machine learning that aims to synthesize new data which is similar to a set of given data. State of the art are computationally intense and tuning-heavy algorithms such as generative adversarial networks (GANs) [1, 2], variational autoencoders [3], normalizing flows [4], diffusion models [5] or transformer-based models [6]. A much more lightweight procedure is to use an Adversarial Random Forest (ARF) [7]. ARFs achieve competitive performance in generative modeling with much faster runtime [7], yet they do not require the practitioner to have extensive knowledge of generative modeling.

Further, ARFs are especially useful for data that comes in a table format, i.e., tabular data. That is because ARFs are based on random forests [8] which leverage the advantages that tree-based methods have over neural networks on tabular data [9] for generative modeling. Further, as part of the procedure, ARFs give access to the estimated joint density, which is useful for several other fields of research, e.g., unsupervised machine learning. For the task of density estimation, ARFs have

---

been demonstrated to yield remarkable results as well [7]. In brief, ARFs are a promising methodological contribution to the field of generative modeling and density estimation, providing a ready-made solution to generate data for practitioners across fields.

ARFs have already gained some attention in the scientific community [10], however, the paper by [7] provides the audience with a R software package only. The machine learning and generative modeling community, however, is mostly using python as a programming language and to reach a broad audience more generally, a fast and user-friendly implementation of ARFs in python is highly desirable. We aim to fill this gap with the presented python implementation of ARFs.

`arfp` is inspired by the R implementation called `arf` [11], but transfers the algorithmic structure to match the class-based structure of python code and takes advantage of computationally efficient python functions. Similar to the R implementation, separate functions for first fitting the density (FORDE algorithm [7]) and then generating new data samples (FORGE algorithm [7]) exist. However, in `arfp`, the functions are called for an initialized ARF class object, which is showcased in the usage example below.

Crucially, for practitioners working with python as programming language, the direct python implementation is more robust and convenient to users than calling fragile wrappers like `rpy2` [12] that aim to make R code running in python. The benefits of a direct python implementation of ARFs for the generative modeling community have already been recognized by now. For example, `arfp` is integrated in the data synthesizing framework `synthcity` [13].

### Implementation and architecture

**Module Design** The general workflow of generating data with `arfp` is (1) to initialize an object of class `arf` with real data, (2) estimate the density and (3) sample new data. This procedure is visualized in Figure 1.

The architecture of `arfp` reflects this workflow and we have class `arf` building the backbone of the procedure. An instance of class `arf` takes the real data set as input and trains an ARF, i.e., learns the actual data's structure. To this object, functions to estimate the density (FORDE algorithm [7], function `forde()`) and generate data (FORGE algorithm [7], function `forge()`) can be applied. This architecture allows users to learn the structure of the real data once (when initializing the `arf` class object) and then flexibly adapt density estimation, e.g., using different parameters, or repeatedly sampling new data without having to refit the model.

**Methodology Overview** For interested readers, we want to briefly describe the methodological foundations of ARFs, but refer to [7] for further details. From a given real data set, first, naive synthetic data is generated (initial generation step) by sampling from the marginal distributions of the features. Then, a random forest [8] is fit to distinguish this synthetic from the real data (initial discrimination step). This procedure, also known as fitting an unsupervised random forest [14], guides the random forest to learn the dependency structure in the data. Using this forest, we can sample observations from the leaves of the trees to generate updated synthetic data (generation step). Subsequently, a new random forest is

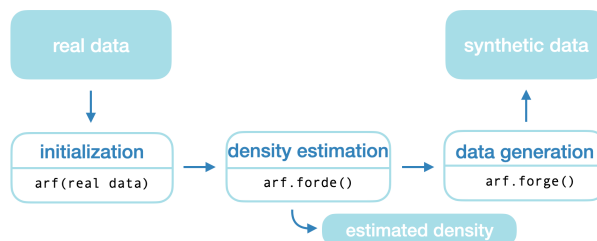


Figure 1: Workflow of `arfpy`'s core functionalities.

fit to differentiate between synthetic and real data (discrimination step). Drawing on the adversarial idea of GANs, this iterative procedure of data generation and discrimination will be repeated until the discriminator cannot distinguish between generated and real data anymore. At this stage, the accuracy of the forest will be  $\leq 0.5$  and the forest is assumed to have converged, which implies mutually independent features in the terminal nodes. This drastically simplifies density estimation and generative modeling, as it allows us to formulate the univariate density for each feature separately with data in the leaves of the fitted ARF (FORDE algorithm) and then combine them to the joint density, instead of having to model multivariate densities. For data generation, we can use this trait to sample a new observation by drawing a leaf from the forest of the last iteration step and use the data distributions with parameters estimated from that leaf to sample each feature separately (FORGE algorithm).

**Example Usage** Let us illustrate the usage of `arfpy` with a visually intuitive example: We create data using `make_moons` from `sklearn.datasets`, which results in data along two continuous axes that looks like two moons from different categories. Statistically speaking, this is a tabular dataset, consisting of both continuous and categorical features that exhibit a dependency structure. For a more intuitive understanding of the data, see Figure 2, Panel **A**. The task of `arfpy` is to learn the structure of this given (real) data and generate new data instances that appear similar.

To initialize the workflow, we need to run relevant imports, including the import of class `arf` from the `arfpy` module, and create the real dataset. The `arf` class takes a `pandas DataFrame` as input, so the real data is pre-processed to match this requirement. This incorporates setting unique column names (`'dim_1'`, `'dim_2'`, `'label'`) and ensuring that `'label'` is stored in the correct data type `'category'`.

```

1 import pandas as pd
2 from sklearn.datasets import make_moons
3 from arfpy import arf
4
5 moons_X, moons_y = make_moons(n_samples = 3000, noise=0.1)
6 df = pd.DataFrame({"dim_1" : moons_X[:,0], "dim_2" : moons_X[:,1],
7                   "label" : moons_y})
8 df['label'] = df['label'].astype('category')
```

---

```

8
9 df.head()
10
11 #>   dim_1      dim_2      label
12 #>  1.782717  0.099124         1
13 #>  1.087497  0.298744         0
14 #> -0.576695  0.801675         0
15 #>  0.623931 -0.506896         1

```

With the real dataset preprocessed as needed, we can proceed with training the ARF to learn the data's structure. Creating an object of class `arf` will trigger ARF model fitting using the data provided.

```

1 my_arf = arf.arf(x = df)
2
3 #> Initial accuracy is 0.82
4 #> Iteration number 1 reached accuracy of 0.36

```

Because we have used the parameter default `verbose = True`, the training of `my_arf` prints out some interesting information: The initial accuracy, which corresponds to the accuracy of the random forest in distinguishing real data from naive synthetic data, is 0.82. This implies that the random forest is doing very well in distinguishing real from naive synthetic data and therefore, we can assume the model to have learned relevant dependencies that allow the model to make this distinction. Using this forest to sample updated synthetic data, and fitting a new random forest to distinguish this data from real data leads to an accuracy of only 0.36. This accuracy is below the default threshold of 0.5, so loosely speaking, the synthetic data generated with the forest cannot be accurately distinguished from real data, i.e., the generated data looks like real data, which is the goal the algorithm was aiming for. In other words, the relevant dependency structures of the real data have been learned by the forest in the first round of iteration already, so the algorithm has converged and no further iterations need to be conducted.

After the ARF has converged, we can proceed to estimating the joint density. Recap that in a converged ARF, the features are mutually independent in the leaves, which simplifies the challenging multivariate density estimation task into many simple univariate density estimation tasks. The joint density is then a factorization of the individual density estimates across leaves in the ARF. We can call function `forde()` on the `my_arf` object to estimate the density and store the returned dictionary to explore the parameters. The FORDE dictionary contains the estimated parameters for continuous (key `'cnt'`) and categorical features (key `'cat'`). As mentioned in the above paragraph, the parameters are estimated using the data points in the forest's leaves, so we will get estimates for each leaf individually. The parameters for the categorical features simply correspond to the empirical frequency of categories in the leaves, so for a more complex example, we can take a look at the continuous feature's parameter estimates in `FORDE['cnt']`. We have used the default distribution (truncated normal distribution) to model continuous features, so the output will reflect estimates for the mean and standard deviation for each feature (`'dim_1'`, `'dim_2'`) in each leaf, which is uniquely identified by `'tree'` and `'nodeid'`:

```

1 FORDE = my_arf.forde()
2
3 FORDE['cnt'].iloc[:, :5].head()
4
5 #>   tree nodeid   variable   mean   sd
6 #>   0     3     dim_1     0.961437 0.214925
7 #>   0     3     dim_2    -0.671571 0.028193
8 #>   0    11     dim_1     1.040565 0.185581
9 #>   0    11     dim_2    -0.621924 0.003328

```

With the parameters estimated, we can move on to the final step of the generative modeling task and sample new data instances with the function `forge()`.

For each instance to be generated, the function randomly samples a leaf from the forest with weighted probability according to the coverage of real data in the leaves of the ARF and then uses the parameters estimated through `forde()` to sample a new data instance.

```

1 df_syn = my_arf.forge(n = 1000)
2
3 df_syn.head()
4
5 #>   dim_1   dim_2   label
6 #> -0.018004  0.283963    1
7 #>  1.734200 -0.085115    1
8 #> -0.009840  1.046872    0
9 #>  0.868400 -0.352692    1

```

Calling `forge()` completes the task of generating synthetic data that mimics real data. From the generated data table itself, the similarity is hard to grasp, but we can visually inspect the quality of the synthetic data in Figure 2.

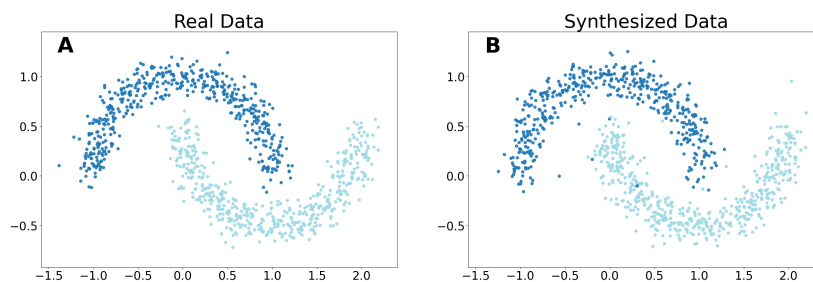


Figure 2: Comparison of real and synthesized data.

### Quality control

The software has been tested through unit tests, which includes testing of relevant functionalities with various input data sets. The workflow of running these tests is automated on GitHub actions, but can be run locally and with customized data sets using the instructions provided in the software repository. Further, the repository

---

allows to publicly raise questions or report bugs and gives clear guidelines on how to contribute to the open source software project are lined out.

## (2) Availability

### Operating system

Platform Independent

### Programming language

Python  $\geq$  3.8

### Additional system requirements

No specific requirements

### Dependencies

numpy  $\geq$  1.20.3, pandas  $\geq$  1.5, scikit-learn  $\geq$  0.24, scipy  $\geq$  1.4

### List of contributors

Blesch, Kristin<sup>a,b</sup>;

Wright, Marvin N.<sup>a,b,c</sup>;

(a) Leibniz Institute for Prevention Research and Epidemiology – BIPS, Germany;

(b) Faculty of Mathematics and Computer Science, University of Bremen, Germany;

(c) Department of Public Health, University of Copenhagen, Denmark

### Software location:

#### Archive

**Name:** arfpy

**Persistent identifier:** <https://pypi.org/project/arfpy/>

**Licence:** MIT

**Publisher:** Kristin Blesch

**Version published:** v0.1.1

**Date published:** 22/09/2023

#### Code repository

**Name:** arfpy

**Persistent identifier:** <https://github.com/bips-hb/arfpy>

**Licence:** MIT

**Date published:** 06/09/2023

### Language

English

## (3) Reuse potential

ARFs have been introduced with a solid theoretical background, yet do not have to compromise on a complex algorithmic structure and instead are a low-key algorithm that does not require extensive hyperparameter tuning [7]. In contrast to the typically deep learning based alternatives, ARF does not require background



---

knowledge of generative modeling, intense tuning efforts or large computational resources. Given the theoretical foundation and straightforward implementation with `arfp`, the methodology is attractive for both scholars conducting rather theoretical research in statistics, e.g., density estimation, as well as practitioners from other fields that need to generate new data samples.

Typical use cases of such synthesized data samples are, for example, the imputation of missing values, data augmentation or the conduct of analyses that respect data protection rules. With the specialty of ARFs being particularly suitable for tabular data, including a natural incorporation of both continuous and categorical features, the straightforward python implementation of ARFs provides a convenient algorithm to a broad audience from different fields.

With the python programming language being widespread, `arfp` can smoothly integrate in the code of various applications. Further, usability is enhanced by the intuitive documentation provided at <https://bips-hb.github.io/arfp/>, making `arfp` an easily accessible tool to generate data.

In sum, `arfp` introduces density estimation and generative modeling with ARFs to python, which enables practitioners from a wide variety of fields to generate fast and reliable synthetic data and density estimates with python as a programming language.

#### **Acknowledgements**

We thank David S. Watson and Jan Kapar for their contributions to establishing the theoretical groundwork of adversarial random forests.

#### **Funding statement**

This work was supported by the German Research Foundation (DFG), Emmy Noether Grant 437611051.

#### **Competing interests**

The authors declare that they have no competing interests.

#### **References**

- [1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014.
- [2] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachani. Modeling tabular data using conditional GAN. *Advances in Neural Information Processing Systems*, 32, 2019.
- [3] DP Kingma and M Welling. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.
- [4] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. *Proceedings of the 32nd International Conference on Machine Learning*, 37:1530–1538, 07–09 Jul 2015.

- 
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [7] David S Watson, Kristin Blesch, Jan Kapar, and Marvin N Wright. Adversarial random forests for density estimation and generative modeling. *International Conference on Artificial Intelligence and Statistics. PMLR*, 206:5357–5375, 2023.
- [8] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001. doi: 10.1023/A:1010933404324.
- [9] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022.
- [10] Richard Nock and Mathieu Guillaume-Bert. Generative forests. *arXiv Preprint 2308.03648*, 2023. doi: 10.48550/arXiv.2308.03648.
- [11] Marvin N. Wright and David S. Watson. *arf: Adversarial Random Forests*, 2023. URL <https://CRAN.R-project.org/package=arf>. R package version 0.1.3.
- [12] Laurent Gautier et al. rpy2: Python-r bridge. *GitHub repository*, 2023. URL <https://github.com/rpy2/rpy2>.
- [13] van der Schaar Lab. synthcity: A library for generating and evaluating synthetic tabular data. *GitHub repository*, 2023. URL <https://github.com/vanderschaarlab/synthcity>.
- [14] Tao Shi and Steve Horvath. Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics*, 15(1):118–138, 2006. doi: 10.1198/106186006X94072.

**Part IV.**

**Conclusion and Discussion**



## 4. Conclusion

This thesis elucidates obstacles in applying IML methods and proposes suitable methodological advancements to mitigate this. In particular, the work uses statistical considerations to illuminate barriers and contributes towards overcoming the hurdles mixed tabular data with a dependency structure imposes (Paper 1 and Paper 2). Expanding such considerations to the closely related field of generative modeling, actively taking advantage of this data type yields straightforward methodology (Paper 3) and software (Paper 4) that can reach results more efficiently.

**Mixed Tabular Data** In real-world problems, tabular data frequently consists of both continuous and categorical features. Previous literature in IML scarcely uses specialized methods for this kind of data, forcing practitioners to use workarounds such as dummy encoding data and treating those features as continuous. Paper 1 illustrates that specialized methods lead to more powerful results than such workarounds and proposes a specialized method for measuring conditional feature importance with mixed tabular data. In generative modeling, which may be a subroutine in IML techniques, the nature of mixed tabular data has also received little attention. There, complex algorithmic procedures adapted from other data types form the state-of-the-art. However, Paper 3 demonstrates that it is possible to leverage algorithms particularly suitable for mixed tabular data, such as random forests, to improve efficiency and user-friendliness substantially. Paper 3 introduces ARF, i.e., adversarial random forests, for generative modeling and density estimation alongside a *Python* software implementation (Paper 4). The contributions of this thesis embrace the mixed data type and equip users with methods suitable for the data they encounter in practice.

**Dependency Structures** Accounting for dependency structures is crucial to ensure that methods perform as intended. In statistics, the distinction between conditional and marginal approaches is well-established but thus far less frequently acknowledged in IML. However, Paper 1 demonstrates that disregarding dependency structures yields divergent explanations. Further, Paper 2 replicates results from previous literature, which illustrate that adversarial attacks are feasible on methods that fail to account for dependency structures. As a remedy, Paper 2 proposes knock-off imputation for Shapley value explanations to defend attacks on both a local and global level by taking dependency structures into account. For Paper 3 and Paper 4, appropriately learning data dependency structures is vital for synthesizing data that mimics the given data. Evidently, dependency structures are essential in method development and deployment.

**Real Data Applicability** This thesis advances methodology to align closely with the demands of real data, that is, tabular data that is both of mixed data types and exhibits dependency structures. This thesis moves methodology towards an improvement in terms of real-world applicability by considering these aspects in conjunction. Notably, the barrier to application may – besides a

lack of suitable methods – be an insufficient understanding of potential consequences and missing software implementations. In order to streamline real-world application, Paper 1 explicitly discusses guidelines on when to use which feature importance measure, Paper 2 delivers a method that is robust against adversarial attacks and ready-to-use for model auditors, and Paper 3 proposes a low-key algorithm that requires little tuning efforts for density estimation and generative modeling. The supply of open-source software is crucial, hence, Paper 4 is dedicated to providing straightforward software to elevate real-world applicability.

In sum, this thesis sheds light on relevant statistical considerations in IML and advances methodology to yield adequate, powerful, and user-friendly methods required for real data applications. The contributing parts of this thesis illustrate the need for statistically adequate methods to enable meaningful and robust IML insights. Parts I, II and III provide users with methods that suit real-world data, which frequently is mixed tabular data that exhibits dependency structures. For approaches that assess the importance of features and, more generally, algorithms that synthesize mixed tabular data, this thesis details analytically and demonstrates empirically the relevance of statistical considerations for adequate method choice. To equip users with suitable algorithms for their work on real-world applications, the methods proposed are modular, flexible, and easily accessible through open-source software.

## 5. Discussion and Future Work

This section critically analyzes this thesis’s findings and examines them in a broader context. It sheds light on more general implications of the findings, highlighting that further circumstances and aspects of real data requirements may hinder the statistical adequacy of IML method application in the real world. Moreover, this section addresses overarching challenges within the IML and generative modeling field and outlines potential directions for future research.

Real data exhibits a multiplicity of data types. Other than the distinction between continuous and categorical features, further data types, such as ordinal, or temporally structured data, may occur in real-world applications. The focus on mixed tabular data limits the scope of this thesis, yet specialized procedures for other data types also demand consideration in application and method development by future research. Additionally, extending methodology for the conjunction of different data types may be a promising field of future research. Multimodal machine learning is a vibrant field of research (Liang et al., 2023), and explainability methods tailored to this could provide valuable insight. For example, a multimodal analysis of tabular and image data might benefit from apt aggregation procedures across IML methods.

Empirical data is characterized not only by the data types included but also by domain specifics. IML method development has to balance domain-specific requirements with general applicability across fields. Matching the needs of IML stakeholders from various domains and the methods’ properties is challenging (Vermeire et al., 2021). Future research demands multi-disciplinary research collaborations that develop methods incorporating the perspectives of both methodological disciplines, such as computer science or statistics, as well as domain experts (Saeed and Omlin, 2023). This raises the question of the extent to which IML methodology developed by only computer scientists or statisticians, in fact, assists in solving real-world problems. In that light, the claims of this thesis for being relevant for real data applications have a grain of salt attached. Nonetheless, being a methodological rather than empirical work, this thesis contributes to providing methods that approach characteristics frequently occurring in real data sets across fields such as data dependency structures<sup>1</sup> and, therefore, refrains from focusing on a specific domain. Still, future research in IML may benefit from collaborating closely with domain experts.

Instead of real-world applications, machine learning literature typically relies on publicly available benchmark data sets, which may be a shortcoming deserving further consideration. This aspect relates to the above-mentioned absence of real-world demonstrations in method propositions and further highlights the potentially problematic concentration on very few benchmark data sets. In defense of method evaluations on benchmark data sets, this procedure makes methods easily comparable and facilitates highlighting the benefits of newly proposed methods. This interest often

---

<sup>1</sup>On a side note, it is remarkable to realize that if there were no dependency structures in the data, most phenomena discussed in this thesis would vanish: Marginal and conditional feature importance discussed in Paper 1 would coincide, the adversarial attacks in Paper 2 would not unfold, and instead of advanced generative modeling approaches like the one proposed in Paper 3, data generation could be conducted simply by sampling from the marginals. The crucial point, however, is that real data sets often do exhibit dependency structures.

outweighs the desire for diversity (in terms of domains) and real-world problem-solving. However, doing so may also lead method development to overfit on few selected problems. Ultimately, method development is not a self-purposed ambition but should facilitate conducting empirical findings. However, human-centric studies have found mixed findings on whether IML explanations available thus far are helpful for human understanding (Krishna et al., 2022; Weerts et al., 2019). Hence, method proposals are encouraged to demonstrate their effectiveness beyond the evaluation of only benchmarking tasks like credit assessment or digit recognition which may not reflect the actual complexity of empirical tasks.

For applicability to real-world problems, this thesis contributes to overcoming built-in limitations of IML methods regarding mixed data applications, yet other pitfalls may persist. This potential shortcoming is particularly apparent when considering the entire machine learning pipeline: the explained machine learning model itself must be suitable, IML methods could be applied incorrectly, or further intrinsic limitations of IML methods may persist; see Molnar et al. (2022) for further discussion and Rudin et al. (2022) for a characterization of related challenges in IML. The awareness of potential pitfalls from a broad perspective is crucial to ensure meaningful explanations through IML.

Further, practical issues in method application demand enhanced attention in machine learning research. Method developers working in academia or tech companies may be better equipped with computational resources than domain-specific practitioners working in other fields, e.g., in finance or medicine. In this thesis, Paper 3 briefly addresses this issue by demonstrating that ARF performs reasonably fast on CPU units. In contrast, deep learning-based methods typically require GPU resources – which may be challenging to access for users – to be feasible in a reasonable time. An awareness of the resources available to users may help develop methods that suit the equipment of practitioners more closely and hence increase the real-world impact of methods.

Practical considerations are not limited to computational hardware but, more crucially, also software. Users are, typically, neither method developers nor software engineers who can quickly adapt code provided on a conceptual level to their use case. Academic papers that introduce novel methods, however, often provide only code for their experiments, which does not encourage the transfer of the method to other users. A key issue might be that the academic community thus far does not sufficiently incentivize the provision of stand-alone software and tutorials, e.g., as in Paper 4. Academic recognition is widely achieved for publishing papers proposing novel methods rather than for providing (and maintaining) user-friendly software. A change in academic culture to better acknowledge such contributions may positively impact bridging the gap between method development and application.

That said, future research should focus on leveraging efficient tools for challenging tasks. For example, generative modeling subroutines beyond IML applications may use advantageous algorithms like ARF, e.g., for missing value imputation or even (through an adaptation from the CPI procedure) conditional independence testing for causal structure learning. Focusing on the adaptation of easily applicable procedures may be especially beneficial for users.

Through a statistical perspective, this thesis contributes to several aspects related to mixed tabular data with dependency structures in IML and generative modeling. Nonetheless, future research directions demand further advancements in methodology. Matching methodological assumptions with real-world requirements, including the evaluation of real problems, remains an open challenge in the field of IML and generative modeling.







## References

- Aas, K., M. Jullum, and A. Løland (2021). Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *Artificial Intelligence* 298, Article 103502.
- Aas, K., T. Nagler, M. Jullum, and A. Løland (2021). Explaining predictive models using Shapley values and non-parametric vine copulas. *Dependence Modeling* 9(1), 62–81.
- Abroshan, M., K. H. Yip, C. Tekin, and M. van der Schaar (2023). Conservative policy construction using variational autoencoders for logged data with missing values. *IEEE Transactions on Neural Networks and Learning Systems* 34(9), 6368–6378.
- Adadi, A. and M. Berrada (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access* 6, 52138–52160.
- Aivodji, U., H. Arai, O. Fortineau, S. Gambs, S. Hara, and A. Tapp (2019). Fairwashing: The risk of rationalization. In *Proceedings of the 36<sup>th</sup> International Conference on Machine Learning*, pp. 161–170. PMLR.
- Alaa, A., B. Van Breugel, E. S. Saveliev, and M. van der Schaar (2022). How faithful is your synthetic data? Sample-level metrics for evaluating and auditing generative models. In *Proceedings of the 39<sup>th</sup> International Conference on Machine Learning*, pp. 290–306. PMLR.
- Ali, A., S. M. Shamsuddin, and A. L. Ralescu (2013). Classification with class imbalance problem. *International Journal of Advances in Soft Computing and its Applications* 5(3), 176–204.
- Alikhademi, K., B. Richardson, E. Drobina, and J. E. Gilbert (2021). Can explainable AI explain unfairness? A framework for evaluating explainable AI. *ArXiv Preprint arXiv:2106.07483*.
- Alqahtani, H., M. Kavakli-Thorne, and G. Kumar (2021). Applications of generative adversarial networks (GANs): An updated review. *Archives of Computational Methods in Engineering* 28, 525–552.
- Apley, D. W. and J. Zhu (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 82(4), 1059–1086.
- Au, Q., J. Herbringer, C. Stachl, B. Bischl, and G. Casalicchio (2022). Grouped feature importance and combined features effect plot. *Data Mining and Knowledge Discovery* 36(4), 1401–1450.
- Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS One* 10(7), Article e0130140.

- Baniecki, H. and P. Biecek (2023). Adversarial attacks and defenses in explainable artificial intelligence: A survey. *ArXiv Preprint arXiv:2306.06123*.
- Bargagli Stoffi, F. J., G. Cevolani, and G. Gnecco (2022). Simple models in complex worlds: Occam’s razor and statistical learning theory. *Minds and Machines* 32(1), 13–42.
- Bates, S., E. Candès, L. Janson, and W. Wang (2021). Metropolized knockoff sampling. *Journal of the American Statistical Association* 116(535), 1413–1427.
- Bishop, C. M. and N. M. Nasrabadi (2006). *Pattern Recognition and Machine Learning* (1<sup>st</sup> ed.). Springer.
- Blesch, K., D. S. Watson, and M. N. Wright (2023). Conditional feature importance for mixed data. *ASTA Advances in Statistical Analysis*.
- Blesch, K. and M. N. Wright (2023). Arfpy: A python package for density estimation and generative modeling with adversarial random forests. *ArXiv Preprint arXiv:2311.07366*.
- Blesch, K., M. N. Wright, and D. S. Watson (2023). Unfooling SHAP and SAGE: Knockoff imputation for Shapley values. In L. Longo (Ed.), *Explainable Artificial Intelligence. xAI 2023. Communications in Computer and Information Science*, Volume 1901, pp. 131–146. Springer, Cham.
- Bond-Taylor, S., A. Leach, Y. Long, and C. G. Willcocks (2021). Deep generative modelling: A comparative review of VAEs, GANs, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(11), 7327–7347.
- Borisov, V., T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci (2022). Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.
- Breiman, L. (2001a). Random forests. *Machine Learning* 45(1), 5–32.
- Breiman, L. (2001b). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical Science* 16(3), 199–231.
- Camino, R. D., C. A. Hammerschmidt, and R. State (2019). Improving missing data imputation with deep generative models. *ArXiv Preprint arXiv:1902.10666*.
- Candès, E., Y. Fan, L. Janson, and J. Lv (2018). Panning for gold: Model-free knockoffs for high-dimensional controlled variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology* 80(3), 551 – 577.
- Cartella, F., O. Anunciacao, Y. Funabiki, D. Yamaguchi, T. Akishita, and O. Elshocht (2021). Adversarial attacks for tabular data: Application to fraud detection and imbalanced data. *ArXiv Preprint arXiv:2101.08030*.
- Cave, S. and K. Dihal (2019). Hopes and fears for intelligent machines in fiction and reality. *Nature machine intelligence* 1(2), 74–78.
- Chen, H., I. C. Covert, S. M. Lundberg, and S.-I. Lee (2023). Algorithms to estimate Shapley value feature attributions. *Nature Machine Intelligence* 5(6), 590–601.

## References

---

- Chen, H., S. Jajodia, J. Liu, N. Park, V. Sokolov, and V. Subrahmanian (2019). FakeTables: Using GANs to generate functional dependency preserving tables with bounded real data. In *Proceedings of the 28<sup>th</sup> International Joint Conference on Artificial Intelligence*, pp. 2074–2080.
- Chen, H., J. D. Janizek, S. Lundberg, and S.-I. Lee (2020). True to the model or true to the data? *ArXiv Preprint arXiv:2006.16234*.
- Choi, E., S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun (2017). Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for Healthcare Conference*, pp. 286–305. PMLR.
- Correia, A., R. Peharz, and C. P. de Campos (2020). Joints in random forests. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 11404–11415.
- Covert, I., S. M. Lundberg, and S.-I. Lee (2020). Understanding global feature contributions with additive importance measures. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 17212–17223.
- Deeks, A. (2019). The judicial demand for explainable artificial intelligence. *Columbia Law Review* 119(7), 1829–1850.
- Ding, J., V. Tarokh, and Y. Yang (2018). Model selection techniques: An overview. *IEEE Signal Processing Magazine* 35(6), 16–34.
- Dixon, M., I. Halperin, and P. Bilokon (2020). *Machine Learning in Finance: From Theory to Practice* (1<sup>st</sup> ed.), Volume 1170. Springer, Cham.
- Doshi-Velez, F. and B. Kim (2017). Towards a rigorous science of interpretable machine learning. *ArXiv Preprint arXiv:1702.08608*.
- Du, M., N. Liu, and X. Hu (2019). Techniques for interpretable machine learning. *Communications of the ACM* 63(1), 68–77.
- Engelmann, J. and S. Lessmann (2021). Conditional wasserstein GAN-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications* 174, Article 114582.
- Fisher, A., C. Rudin, and F. Dominici (2019). All models are wrong, but many are useful: Learning a variable’s importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research* 20(177), 1–81.
- Foster, D. (2022). *Generative deep learning: Teaching Machines To Paint, Write, Compose, and Play* (2<sup>nd</sup> ed.). O’Reilly Media, Inc.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5), 1189–1232.
- Gilpin, L. H., D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal (2018). Explaining explanations: An overview of interpretability of machine learning. In *IEEE 5<sup>th</sup> International Conference on Data Science and Advanced Analytics*, pp. 80–89. IEEE.
- Gimenez, J. R. and J. Zou (2019). Improving the stability of the knockoff procedure: Multiple simultaneous knockoffs and entropy maximization. In *Proceedings of the 22<sup>nd</sup> International Conference on Artificial Intelligence and Statistics*, pp. 2184–2192. PMLR.

- Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, Volume 27.
- Goodman, B. and S. Flaxman (2017). European union regulations on algorithmic decision-making and a “right to explanation”. *AI Magazine* 38(3), 50–57.
- Grinsztajn, L., E. Oyallon, and G. Varoquaux (2022). Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems*, Volume 35, pp. 507–520.
- Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi (2018). A survey of methods for explaining black box models. *ACM Computing Surveys* 51(5), 1–42.
- Hancock, J. T. and T. M. Khoshgoftaar (2020). Survey on categorical data for neural networks. *Journal of Big Data* 7(1), 1–41.
- Harshvardhan, G., M. K. Gourisaria, M. Pandey, and S. S. Rautaray (2020). A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review* 38, Article 100285.
- Hastie, T., R. Tibshirani, J. H. Friedman, and J. H. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.*, Volume 2. Springer.
- Hastie, T., R. Tibshirani, and R. J. Tibshirani (2017). Extended comparisons of best subset selection, forward stepwise selection, and the Lasso. *ArXiv Preprint arXiv:1707.08692*.
- Herbinger, J., B. Bischl, and G. Casalicchio (2023). Decomposing global feature effects based on feature interactions. *ArXiv Preprint arXiv:2306.00541*.
- Ho, J., A. Jain, and P. Abbeel (2020). Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 6840–6851.
- Jethani, N., M. Sudarshan, I. Covert, S.-I. Lee, and R. Ranganath (2022). FastSHAP: Real-time Shapley value estimation. In *Proceedings of the 10<sup>th</sup> International Conference on Learning Representations*.
- Jordon, J., J. Yoon, and M. Van Der Schaar (2018). PATE-GAN: Generating synthetic data with differential privacy guarantees. In *Proceedings of the 6<sup>th</sup> International Conference on Learning Representations*.
- Jordon, J., J. Yoon, and M. van der Schaar (2019). KnockoffGAN: Generating knockoffs for feature selection using generative adversarial networks. In *Proceedings of the 7<sup>th</sup> International Conference on Learning Representations*.
- Jullum, M., J. Sjødin, R. Prabhu, and A. Løland (2023). eXplego: An interactive tool that helps you select appropriate XAI-methods for your explainability needs. In L. Longo (Ed.), *Joint Proceedings of the xAI-2023 Late-breaking Work, Demos and Doctoral Consortium co-located with the 1<sup>st</sup> World Conference on Explainable Artificial Intelligence (xAI-2023)*, pp. 146–151.
- Kingma, D. P. and M. Welling (2014). Auto-encoding variational bayes. In *Proceedings of the 2<sup>nd</sup> International Conference on Learning Representations*.

## References

---

- Kormaksson, M., L. J. Kelly, X. Zhu, S. Haemmerle, L. Pricop, and D. Ohlssen (2021). Sequential knockoffs for continuous and categorical predictors: With application to a large psoriatic arthritis clinical trial pool. *Statistics in Medicine* 40(14), 3313–3328.
- Krishna, S., T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu, and H. Lakkaraju (2022). The disagreement problem in explainable machine learning: A practitioner’s perspective. *ArXiv Preprint arXiv:2202.01602*.
- Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, Volume 25.
- Kumar, I. E., S. Venkatasubramanian, C. Scheidegger, and S. Friedler (2020). Problems with Shapley-value-based explanations as feature importance measures. In *Proceedings of the 37<sup>th</sup> International Conference on Machine Learning*, pp. 5491–5500. PMLR.
- Lapuschkin, S., S. Waldchen, A. Binder, G. Montavon, W. Samek, and K.-R. Muller (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications* 10(1), Article 1096.
- Leblanc, B. and P. Germain (2023). Interpretability in machine learning: On the interplay with explainability, predictive performances and models. *ArXiv Preprint arXiv:2311.11491*.
- Lei, J., M. G’Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman (2018). Distribution-free predictive inference for regression. *Journal of the American Statistical Association* 113(523), 1094–1111.
- Li, J. and J.-S. Huang (2020). Dimensions of artificial intelligence anxiety based on the integrated fear acquisition theory. *Technology in Society* 63, Article 101410.
- Liang, P. P., A. Zadeh, and L.-P. Morency (2023). Foundations and trends in multimodal machine learning: Principles, challenges, and open questions. *ArXiv Preprint arXiv:2209.03430*.
- Lipton, Z. C. (2017). The mythos of model interpretability. *Queue* 16(3), 31–57.
- Lopez-Paz, D. and M. Oquab (2017). Revisiting classifier two-sample tests. In *Proceedings of the 5<sup>th</sup> International Conference on Learning Representations*.
- Lou, Y., R. Caruana, J. Gehrke, and G. Hooker (2013). Accurate intelligible models with pairwise interactions. In *Proceedings of the 19<sup>th</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 623–631. ACM.
- Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* 2(1), 56–67.
- Lundberg, S. M., G. G. Erion, and S.-I. Lee (2018). Consistent individualized feature attribution for tree ensembles. *ArXiv Preprint arXiv:1802.03888*.
- Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, Volume 30.
- Luo, H., F. Cheng, H. Yu, and Y. Yi (2021). SDTR: Soft decision tree regressor for tabular data. *IEEE Access* 9, 55999–56011.

- Marcílio, W. E. and D. M. Eler (2020). From explanations to feature selection: Assessing SHAP values as feature selection mechanism. In *33<sup>rd</sup> Brazilian Symposium on Computer Graphics and Image Processing Conference on Graphics, Patterns and Images*, pp. 340–347. IEEE.
- Martens, M. J., A. Banerjee, X. Qi, and Y. Shi (2021). Bayesian knockoff generators for robust inference under complex data structure. *ArXiv Preprint arXiv:2111.06985*.
- Miao, J. and L. Niu (2016). A survey on feature selection. *Procedia Computer Science* 91, 919–926.
- Miller, T. (2019). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence* 267, 1–38.
- Mohri, M., A. Rostamizadeh, and A. Talwalkar (2018). *Foundations of Machine Learning* (2<sup>nd</sup> ed.). MIT Press.
- Mohseni, S., N. Zarei, and E. D. Ragan (2021). A multidisciplinary survey and framework for design and evaluation of explainable ai systems. *ACM Transactions on Interactive Intelligent Systems* 11(3-4), 1–45.
- Molnar, C. (2020). Interpretable Machine Learning. Available at: <https://christophm.github.io/interpretable-ml-book/> (Accessed: December 28, 2023).
- Molnar, C., G. Casalicchio, and B. Bischl (2020). Interpretable machine learning – a brief history, state-of-the-art and challenges. In Koprinska, I. *et al.* (Ed.), *ECML PKDD 2020 Workshops*, pp. 417–431. Springer International Publishing.
- Molnar, C., G. Casalicchio, M. Grosse-Wentrup, and B. Bischl (2022). General pitfalls of model-agnostic interpretation methods for machine learning models. In *International Workshop on Extending Explainable AI Beyond Deep Models and Classifiers*, pp. 39–68. Springer Nature.
- Molnar, C., G. König, B. Bischl, and G. Casalicchio (2023). Model-agnostic feature importance and effects with dependent features: A conditional subgroup approach. *Data Mining and Knowledge Discovery*.
- Müller, S., V. Toborek, K. Beckh, M. Jakobs, C. Bauckhage, and P. Welke (2023). An empirical evaluation of the rashomon effect in explainable machine learning. In D. Koutra, C. Plant, M. Gomez Rodriguez, E. Baralis, and F. Bonchi (Eds.), *Machine Learning and Knowledge Discovery in Databases: Research Track*, pp. 462–478. Springer Nature Switzerland.
- Murdoch, W. J., C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu (2019). Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences* 116(44), 22071–22080.
- Ng, A. and M. Jordan (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *Advances in Neural Information Processing Systems*, Volume 14.
- Nock, R. and M. Guillame-Bert (2023). Generative forests. *ArXiv Preprint arXiv:2308.03648*.
- Olsen, L. H. B., I. K. Glad, M. Jullum, and K. Aas (2023). A comparative study of methods for estimating conditional Shapley values and when to use them. *ArXiv Preprint arXiv:2305.09536*.
- OpenAI (2023). Gpt-4 technical report. *ArXiv Preprint arXiv:2303.08774*.



## References

---

- Pargent, F., F. Pfisterer, J. Thomas, and B. Bischl (2022). Regularized target encoding outperforms traditional methods in supervised machine learning with high cardinality features. *Computational Statistics* 37(5), 2671–2692.
- Park, N., M. Mohammadi, K. Gorde, S. Jajodia, H. Park, and Y. Kim (2018). Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment* 11(10), 1071–1083.
- Pearl, J. (2009). *Causality*. Cambridge University Press.
- Peters, J., D. Janzing, and B. Schölkopf (2017). *Elements of Causal Inference: Foundations and Learning Algorithms*. MIT Press.
- Rajkomar, A., J. Dean, and I. Kohane (2019). Machine learning in medicine. *New England Journal of Medicine* 380(14), 1347–1358.
- Ramesh, A., P. Dhariwal, A. Nichol, C. Chu, and M. Chen (2022). Hierarchical text-conditional image generation with CLIP latents. *ArXiv Preprint arXiv:2204.06125*.
- Redelmeier, A., M. Jullum, and K. Aas (2020). Explaining predictive models with mixed features using Shapley values and conditional inference trees. In A. Holzinger, P. Kieseberg, A. Tjoa, and E. Weippl (Eds.), *Machine Learning and Knowledge Extraction*, pp. 117–137. Springer, Cham.
- Rezende, D. and S. Mohamed (2015). Variational inference with normalizing flows. In *Proceedings of the 32<sup>th</sup> International Conference on Machine Learning*, pp. 1530–1538. PMLR.
- Ribeiro, M. T., S. Singh, and C. Guestrin (2016). "Why should I trust you?" Explaining the predictions of any classifier. In *Proceedings of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Rinaldo, A., L. Wasserman, and M. G'Sell (2019). Bootstrapping and sample splitting for high-dimensional, assumption-lean inference. *The Annals of Statistics* 47(6), 3438 – 3469.
- Romano, Y., M. Sesia, and E. Candès (2020). Deep knockoffs. *Journal of the American Statistical Association* 115(532), 1861–1872.
- Ru, B., A. Alvi, V. Nguyen, M. A. Osborne, and S. Roberts (2020). Bayesian optimisation over multiple continuous and categorical inputs. In *Proceedings of the 37<sup>th</sup> International Conference on Machine Learning*, pp. 8276–8285. PMLR.
- Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1(5), 206–215.
- Rudin, C., C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong (2022). Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys* 16, 1 – 85.
- Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). Learning representations by back-propagating errors. *Nature* 323(6088), 533–536.
- Saeed, W. and C. Omlin (2023). Explainable AI: A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems* 263, Article 110273.
- Sesia, M., C. Sabatti, and E. J. Candès (2018, 08). Gene hunting with hidden Markov model knockoffs. *Biometrika* 106(1), 1–18.

- Shapley, L. (1953). A value for n-person games. In H. Kuhn and A. Tucker (Eds.), *Contributions to the Theory of Games II*, pp. 307–317. Princeton University Press.
- Shi, T. and S. Horvath (2006). Unsupervised learning with random forest predictors. *Journal of Computational and Graphical Statistics* 15(1), 118–138.
- Shwartz-Ziv, R. and A. Armon (2022). Tabular data: Deep learning is not all you need. *Information Fusion* 81, 84–90.
- Sinaga, K. P. and M.-S. Yang (2020). Unsupervised k-means clustering algorithm. *IEEE Access* 8, 80716–80727.
- Slack, D., S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju (2020). Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.
- Spector, A. and L. Janson (2022). Powerful knockoffs via minimizing reconstructability. *The Annals of Statistics* 50(1), 252 – 276.
- Strobl, C., A.-L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis (2008). Conditional variable importance for random forests. *BMC Bioinformatics* 9, 1–11.
- Štrumbelj, E. and I. Kononenko (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems* 41(3), 647–665.
- Sudarshan, M., W. Tansey, and R. Ranganath (2020). Deep direct likelihood knockoffs. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 5036–5046.
- Sundararajan, M. and A. Najmi (2019). The many Shapley values for model explanation. In *Proceedings of the 37<sup>th</sup> International Conference on Machine Learning*, New York. PMLR.
- Thai, H.-T. (2022). Machine learning for structural engineering: A state-of-the-art review. *Structures* 38, 448–491.
- Thanh-Tung, H. and T. Tran (2020). Catastrophic forgetting and mode collapse in GANs. In *Proceedings of the 2020 International Joint Conference on Neural Networks*, pp. 1–10. IEEE.
- Tsang, M., S. Rambhatla, and Y. Liu (2020). How does this interaction affect me? Interpretable attribution for feature interactions. In *Advances in Neural Information Processing Systems*, Volume 33, pp. 6147–6159.
- Ustun, B. and C. Rudin (2014). Methods and models for interpretable linear classification. *ArXiv Preprint arXiv:1405.4047*.
- Vardhan, L. V. H. and S. Kok (2020). Generating privacy-preserving synthetic tabular data using oblivious variational autoencoders. In *Proceedings of the Workshop on Economics of Privacy and Data Labor at the 37<sup>th</sup> International Conference on Machine Learning*.
- Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, Volume 30.

## References

---

- Vermeire, T., T. Laugel, X. Renard, D. Martens, and M. Detyniecki (2021). How to choose an explainability method? Towards a methodical implementation of XAI in practice. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 521–533. Springer.
- Waltl, B. and R. Vogl (2018). Increasing transparency in algorithmic-decision-making with explainable AI. *Datenschutz und Datensicherheit-DuD* 42(10), 613–617.
- Watson, D. (2022a). Rational Shapley values. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pp. 1083–1094.
- Watson, D. S. (2022b). Conceptual challenges for interpretable machine learning. *Synthese* 200(2), Article 65.
- Watson, D. S., K. Blesch, J. Kapar, and M. N. Wright (2023). Adversarial random forests for density estimation and generative modeling. In *Proceedings of the 26<sup>th</sup> International Conference on Artificial Intelligence and Statistics*, pp. 5357–5375. PMLR.
- Watson, D. S., J. O’Hara, N. Tax, R. Mudd, and I. Guy (2023). Explaining predictive uncertainty with information theoretic Shapley values. *ArXiv Preprint arXiv:2306.05724*.
- Watson, D. S. and M. N. Wright (2021). Testing conditional independence in supervised learning algorithms. *Machine Learning* 110(8), 2107–2129.
- Weerts, H. J., W. van Ipenburg, and M. Pechenizkiy (2019). A human-grounded evaluation of SHAP for alert processing. *ArXiv Preprint arXiv:1907.03324*.
- Williamson, B. and J. Feng (2020). Efficient nonparametric statistical inference on population feature importance using Shapley values. In *Proceedings of the 37<sup>th</sup> International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 10282–10291. PMLR.
- Xu, L., M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni (2019). Modeling tabular data using conditional GAN. In *Advances in Neural Information Processing Systems*, Volume 32.
- Zhang, Y., K. Song, Y. Sun, S. Tan, and M. Udell (2019). Why should you trust my explanation? Understanding uncertainty in LIME explanations. *ArXiv Preprint arXiv:1904.12991*.
- Zhimei Ren, Y. W. and E. Candès (2023). Derandomizing knockoffs. *Journal of the American Statistical Association* 118(542), 948–958.

