FACULTY OF ENGINEERING AND ARCHITECTURE

The Design of Predictive and Uncertainty-Calibrated Treatment Models for the Intensive Care Unit

Jarne Verhaeghe

Doctoral dissertation submitted to obtain the academic degree of
Doctor of Computer Science Engineering

Supervisors

Prof. Sofie Van Hoecke, PhD* - Prof. Jan De Waele, PhD** - Prof. Femke Ongenae, PhD***

* Department of Electronics and Information Systems
Faculty of Engineering and Architecture, Ghent University

** Department of Internal Medicine and Pediatrics
Faculty of Medicine and Health Sciences, Ghent University

*** Department of Information Technology
Faculty of Engineering and Architecture, Ghent University

September 2025

GHENT UNIVERSITY

ISBN 978-94-93464-19-3
NUR 984, 981
Wettelijk depot: D/2025/10.500/79

Members of the Examination Board

Chair

Prof. Filip De Turck, PhD, Ghent University

Other members entitled to vote

Prof. Kirsten Colpaert, PhD, Ghent University
Prof. Thomas Demeester, PhD, Ghent University
Prof. Dirk Deschrijver, PhD, Ghent University
Prof. Celine Vens, PhD, KU Leuven

Supervisors

Prof. Sofie Van Hoecke, PhD, Ghent University
Prof. Jan De Waele, PhD, Ghent University
Prof. Femke Ongenae, PhD, Ghent University

Preface

„I will not say: do not weep; for not all tears are an evil."

–J.R.R. Tolkien, The Return of the King

Five years. This five-year-long adventure has culminated in the work you see before you, a book of which I am very proud. It was an adventure that began in September 2020, at the height of a pandemic and during a period of great uncertainty. The journey had its share of highs and lows. There were times when motivation was lost, when rejection eroded my confidence, and when quitting seemed like the only option. However, I persevered to the finish line: a doctoral degree in Computer Science Engineering. The experience of conducting this research, learning new domains, and challenging myself has been truly enriching. I would not have reached this point without the many people who pushed me forward, both knowingly and unknowingly.

Pursuing a PhD was not my original plan, not even an honest consideration. As my 2020 graduation in Computer Science Engineering approached, I was uncertain about my next steps. The COVID-19 pandemic was beginning to unfold, and several potential job opportunities had fallen through. It was at that moment that I received the offer to start this PhD under the guidance of my supervisors, Prof. Sofie Van Hoecke, Prof. Femke Ongenae, and Prof. Jan De Waele, on a project called Heroi2C aimed at optimizing antibiotic care in the ICU. This marked the unexpected start of my doctoral adventure. To Sofie and Femke, thank you for this wonderful opportunity, for your exceptional guidance, the countless meetings, the team spirit, the unicorns, the board games, the occasional D&D session, and simply for being there. Thank you as well, Jan, for lending your invaluable medical expertise.

Throughout this journey, I was fortunate to be surrounded by amazing colleagues from the PreDiCT team. A special thanks to Thomas De Corte, my closest colleague for the first few years. You were a fantastic teammate; it was a pleasure to spar with you, to generate far too many ideas than we had time for, and simply to work alongside you. I also want to give a personal thank you to Jef Jonkers for your ideas, expertise, and motivation, which pushed me to go further. I am also grateful to the many other past and present colleagues with whom I shared conversations at the coffee machine, cluster meetings, team days, and much more. Thank you: David Vander Mijnsbrugge, Thomas Kok, Nathan Vandemoortele, Jeroen Van Der Donckt, Jonas Van Der Donckt, Marija Stojchevska, Stephanie Chen, Bram Steenwinckel, Tom Windels, Anahita

Rouze, Thibault Blyau, Mathias De Brouwer, Arne Calleart, Emma Nuyts, Vic Degraeve, Ganjour Mazaev, Dieter De Paepe, Sandeep Ramachandra, Emile Deman, Stef Pletinck, Colin De Soete, Cédric Bruylandt, Diego Nieves Avendano, Warre Avereyn, and Ayko Chevaillier.

I would also like to thank the members of the examination board: Prof. Filip De Turck, Prof. Kirsten Colpaert, Prof. Thomas Demeester, Prof. Dirk Deschrijver, Prof. Celine Vens. Thank you for dedicating your time to reading and evaluating my work, and for your thought-provoking questions during my internal defense.

The adventure would also not have started without the wonderful support of my parents. You two supported me from the start to the finish, and had to endure my explanations of exactly what I did, multiple times.

Finally, to my future wife, Jade. Words cannot fully express what your support has meant in reaching this goal. You were always there to help, encourage, push, and comfort me. You have been one of my pillars over these past years. Thank you for being my soulmate and for always being by my side. And to Maiko, my stupidly large cat, thank you for nearly deleting this manuscript on several occasions and for ensuring I was never alone during the writing process.

At last, for you, the reader, I hope you find this book, or at least parts of it, interesting. Thank you for reading, or at least, a thank you in advance.

Gent, September 2025
Jarne Verhaeghe

Table of Contents

List of Figures

List of Tables

Samenvatting

Op verschillende gelegenheden kunnen we geconfronteerd worden met een situatie waarin we een beslissing moeten nemen over een actie, maar dat we de uitkomst van onze keuzes niet volledig kunnen inschatten. Wat we als mensen dan vaak doen, als we de mogelijkheid hebben natuurlijk, is contact opnemen met een collega of iemand met kennis van het domein om advies te vragen om zo een beter begrip te krijgen van de potentiële uitkomsten. Als de actie die we moeten ondernemen niet binair is, maar continu, zoals het bepalen van een dosering van een medicijn in plaats van het wel of niet uitvoeren van een actie, dan wordt de beslissing nog complexer. Hoeveel moeten we toedienen en waarom? Dit is des te meer het geval op de intensieve zorgen (IZ), waar veel kritisch zieke patiënten verschillende behandelingen en medicatie krijgen en er ook veel onbekenden zijn. Het bepalen van de optimale dosering voor deze patiënten of het optimaliseren van hun behandeling is complex maar essentieel alsook verbetert het de kwaliteit van de zorg en vermindert het de mortaliteit en de verblijfsduur. In dit exacte scenario kan een persoonlijke beslissingsondersteuning voor de behandeling, bijvoorbeeld een doseerassistent, immens nuttig zijn. Met kunstmatige intelligentie (AI) is het mogelijk om zo'n assistent te creëren. Aangezien deze assistent echter door artsen moet worden gebruikt, is betrouwbaarheid in deze AI-assistenten van vitaal belang. Momenteel is de adoptie van AI in de geneeskunde beperkt vanwege het beperkte vertrouwen in deze AI-modellen. Deze observatie wordt versterkt door verschillende enquêtes en vragenlijsten en het beperkte aantal modellen dat van onderzoek naar de patient gaan. Er zijn drie belangrijke aspecten aan betrouwbaarheid voor AI-modellen: causaliteit, onzekerheid en robuustheid. Het model moet correct of nauwkeurig zijn en gebruik maken van de juiste inputs die robuust zijn voor verschillende situaties, zoals vaak voorkomt op de IZ. Alsook moet het model duidelijk communiceren hoe zeker het is om ervoor te zorgen dat de gebruiker begrijpt dat, als een onzekere voorspelling wordt gedaan, de huidige predictie niet volledig moet worden vertrouwd, of toch ten minste wat de mogelijke marges van de predictie zijn. Ten slotte moet het model, bij gebruik, de causale relaties hebben geleerd en gebruikers in staat stellen vragen te stellen als: "Waarom stelt u dit voor?öf "Wat maakt de behandeling zo effectief bij deze patiënt?". Daarom moet een betrouwbare doseerassistent die AI gebruikt 1) de juiste variabelen of gegevens hebben geselecteerd om van te leren voor een robuust model, 2) de onzekerheid kwantificeren en meten, 3) uitleggen welke causale factoren de behandelingsaanbevelingen sturen, en 4) situaties communiceren waarin het geen voorspelling kan doen. Dit proefschrift stelt de tools en methoden voor om te helpen bij het creëren van zo'n assistent die beslissingen over behandelingen kan ondersteunen.

Dit proefschrift zal zich richten op IZ-data en twee specifieke IZ-use cases vanwege het kritieke toepassingsdomein en de complexe omgeving, maar de resultaten zijn niet beperkt tot de IZ en zijn zelfs toepasbaar op andere domeinen buiten de gezondheidszorg. De eerste use case is gericht op het voorspellen van de bloedplasmaconcentratie van antimicrobiële middelen die worden gebruikt om bacteriële infecties op de IZ te behandelen en het voorstellen van de meest optimale dosis voor de patiënt om het risico op complicaties te verminderen, therapeutische concentraties te bereiken en het risico op antimicrobiële resistentie te beperken. De tweede use case voorspelt het risico op nieuw ontstane voorkamerfibrillatie (VKF) op de IZ, een hartritmestoornis die resulteert in onregelmatige en abnormaal snelle hartslagen en die tussen de 4,5% en tot 15% van de patiënten op de IZ treft. Van VKF is bekend dat het het risico op mortaliteit en een langere verblijfsduur op de IZ verhoogt.

Om nauwkeurige en robuuste modellen voor beide use cases te bouwen, moeten we eerst de juiste variabelen of features in de IZ-data identificeren die relevant zijn voor modelontwikkeling. Een featureset die alleen relevante features bevat, helpt de interpreteerbaarheid van het model en verhoogt bovendien de robuustheid door ervoor te zorgen dat het model zich niet richt op irrelevante gegevens. Daarom is de eerste bijdrage van dit proefschrift Powershap, een algoritme voor feature selectie dat is gebaseerd op Shapley-waarden. Deze Shapley-waarden kwantificeren de impact dat een variabele of feature heeft op de voorspelling van een model. Powershap vergelijkt de Shapley-waarden van een bekend willekeurig en irrelevante feature met de Shapley-waarden van alle andere features. Het onderliggende idee is dat een irrelevant feature geen grotere impact mag hebben dan een willekeurig feature. Vervolgens worden statistische testen gebruikt om te bepalen welke features irrelevant en welke relevant zijn voor ons model. Powershap biedt een tijdsefficiënte methode voor feature selectie en is minstens 1,5 keer sneller in vergelijking met state-of-the-art (SOTA) methoden, terwijl het tot 20% meer relevante features vindt. Bovendien is Powershap gepubliceerd als een open-source bibliotheek die momenteel meer dan 1 miljoen downloads heeft.

Gegeven de juiste features kunnen we nu de modellen voor elke use case bouwen. Om de betrouwbaarheid te vergroten, worden de modellen verbeterd met methoden voor onzekerheidskwantificering. In de eerste use case stelde ik drie modellen voor die samenkomen in een Kwantiel Ensemble dat de voorspellingen omzet in een voorspellende verdeling. Toegepast op deze use case, kan deze methode de gebruiker voorzien van een volledige inschatting van de mogelijke bloedplasmaconcentraties, gegeven de variabelen van de patiënt. Deze methode is zeker niet beperkt tot deze specifieke use case. Vervolgens worden deze modellen voor de voorspellingen van antimicrobiële bloedplasmaconcentraties verbeterd door domeinkennis te integreren om ze geschikter te maken voor daadwerkelijk doseeradvies. Dit wordt bereikt door gebruik te maken van farmacokinetische domeinkennis over hoe deze medicijnen uit het lichaam worden geëlimineerd. Dit verbetert zowel de voorspellingsnauwkeurigheid als de bruikbaarheid van het model en transformeert het model tot een doseermodel.

Voor de tweede use case ontwikkelde ik VKF-risicovoorspellingsmodellen die de inherente probabiliteitsinschatting gebruiken om het risico te kwantificeren. Deze modellen zijn geconstrueerd door de tijd van diagnose voor patiënten te matchen om het probleem van onevenwichtigheid aan te pakken en bias te vermijden, aangezien

patiënten met VKF doorgaans een langere verblijfsduur hebben in vergelijking met patiënten zonder VKF, wat de voorspellingen zou kunnen vertekenen. Daarom worden de voorspellingstijden van patiënten zonder VKF gematcht met VKF. Dit creëert in wezen een model dat leert onderscheiden of een patiënt op enig moment tijdens hun IZ-opname risico loopt op VKF, alleen op basis van elektronische patiëntendossiergegevens. Dit model is ook gevalideerd met behulp van gegevens van meerdere IZ's om de robuustheid van de bevindingen te waarborgen.

Beide use cases kwantificeren onzekerheid, die onzekerheid moet bijgevolg ook worden gemeten om de nauwkeurigheid van deze onzekerheidskwantificering te waarborgen. Voor de eerste use case gaf ik een voorspellende verdeling als output, en we kunnen de kalibratie van deze verdeling evalueren door te verifiëren of de voorspelde kansen binnen de verdeling overeenkomen met de waargenomen uitkomsten. In dit proefschrift stel ik twee nieuwe metrieken voor voor dit doel: de Absolute Distribution Coverage Error (ADCE) en de Distribution Coverage Error (DCE). De eerste kwantificeert de algehele miskalibratie, terwijl de tweede beoordeelt of het model consistent onder- of overmoedig is in zijn kansvoorspellingen. Evenzo stelde ik voor de VKF-risicomodellen varianten voor die dezelfde evaluatie uitvoeren, maar dan op de risicokansen: de aangepaste Expected Calibration Error (ECE) en de Expected Signed Calibration Error (ESCE). Deze twee metrieken dienen hetzelfde doel, maar dan voor risicovoorspelling. De gepresenteerde metrieken zijn gebaseerd op bestaande metrieken en dragen bij aan het evalueren en optimaliseren van onzekerheidskwantificeringsmodellen om betrouwbaardere modellen te bereiken.

Gegeven gekalibreerde en gevalideerde modellen met de juiste features voor de doseerassistent, missen we nog steeds een oplossing voor vragen als: "Wat maakt de behandeling zo effectief bij deze patiënt?". De volgende bijdrage van dit proefschrift biedt een methode om deze vragen te beantwoorden: Causalteshap. Causalteshap bouwt voort op het idee van Powershap door te vergelijken met een bekend irrelevant willekeurig feature. Causalteshap is echter gericht op het identificeren van predictieve features, dit zijn features die interageren met de behandeling en bepalen of een behandeling meer of minder succesvol zal zijn voor een individu. Door principes uit causale inferentie of causale AI en statistiek toe te passen, samen met het concept van een vergelijking met een willekeurig bekend irrelevant feature, biedt Causalteshap een methode om deze predictieve features te markeren en te identificeren, die essentieel zijn voor het optimaliseren van behandelingen voor individuele patiënten. Deze methode is rigoureus geëvalueerd door middel van uitgebreide experimenten, die aantonen dat het predictieve features kan vinden zonder veel minder valse positieven te genereren in vergelijking met de huidige methoden, waardoor de betrouwbaarheid van de methode wordt gewaarborgd, aangezien elk gemarkeerd feature waarschijnlijk echt predictief is. Deze methode wordt ook toegepast op de VKF-modellen om te onderzoeken of een specifiek medicijn (Noradrenaline) het risico op VKF verhoogt en welke variabelen bijdragen aan dit verhoogde risico, wat illustreert hoe risicomodellen ook kunnen dienen als doseer- of behandelingsassistenten door deze causale vragen te beantwoorden.

Om de gecreëerde modellen effectief in de praktijk te gebruiken, moeten ze ook robuust en betrouwbaar zijn, vooral omdat gebruikers vragen zullen stellen als: "Welke

dosis is het meest optimaal?öf, omgekeerd, "Wat zou er gebeurd zijn als ik in plaats daarvan deze dosis had gegeven?". Deze tweede vraag stelt een contrafactueel scenario, d.w.z. iets dat niet is gebeurd maar wel had kunnen gebeuren. Het gebeurt vaak dat een model vaak geen toegang heeft tot alle potentiële data waardoor er geen betrouwbaar antwoord kan worden gevormd op deze soorten vragen. Een andere mogelijkheid is dat het model is getraind op data met een specifieke behandeltoewijzing, bijvoorbeeld een medicijn wordt alleen aan gezonde mannelijke patiënten gegeven. Als er dus wordt gevraagd naar het behandelingseffect op kritisch zieke zwangere vrouwen, zal het model waarschijnlijk geen betrouwbare schatting kunnen geven. Dit is analoog aan het vragen naar de uitkomst van een specifieke dosis voor een patiënt wanneer er geen ondersteunende data is, wat betekent dat het model geen vergelijkbare patiënten heeft gezien die die dosis hebben gekregen en daarom geen adequate voorspelling kan doen. De laatste bijdrage van dit proefschrift pakt deze uitdaging aan door gebruik te maken van conformal predictie, een robuuste methode voor onzekerheidskwantificering. In deze bijdrage, kwantificeer ik de waarschijnlijkheid van behandeling met behulp van propensity scores, die vervolgens worden gebruikt om de onzekerheidsschatting te wegen en aan te passen. Deze aanpak compenseert het gebrek aan overlap in behandeling in de data, waardoor onzekerheid direct in de voorspellingen wordt geïntegreerd. Als gevolg hiervan kan het model expliciet aangeven wanneer het onvoldoende basis heeft voor een betrouwbare voorspelling, wat de betrouwbaarheid van doseeradviezen verhoogt.

Met al deze diverse bijdragen beoogt dit proefschrift waardevolle tools toe te voegen aan een gereedschapskist voor het bouwen van een AI-ondersteuningssysteem voor behandelbeslissingen, met een bijzondere focus op medicatiedosering, dat effectief kan worden gebruikt op de IC, en uiteindelijk zorgverleners helpt de kwaliteit van de zorg te verbeteren en de patiëntresultaten te verhogen.

Summary

On various occasions, we can be faced with a situation where we need to decide on an action but cannot fully estimate the outcome of our choices. What we as humans often do, if we have the opportunity, is contact a peer or someone knowledgeable in the domain to ask for advice and gain a better understanding of the potential outcomes. If the action we need to take is not binary but continuous, such as determining a dosage of a medicine instead of performing an action or not, the decision becomes even more complex. How much should we administer and why? This is even more the case in the ICU, where many critically ill patients receive various treatments and medications and there are many unknowns. Determining the optimal dosing for these patients or optimizing their treatment is difficult but beneficial and improves the quality of care and reduces mortality and length of stay. In this exact scenario, a personal treatment decision assistant, e.g. a dosing assistant, can be immensely helpful. With artificial intelligence (AI), it is possible to create such an assistant. However, as this assistant must be used by physicians, trustworthiness in these AI assistants is vital. Currently adoption of AI in medicine is limited due to limited trust in these AI models. This observation was reinforced from various surveys and questionnaires and the limited models going from research to the bedside. There are three important aspects to trustworthiness for AI models: Causality, Uncertainty, and Robustness. The model must be correct or accurate using the correct inputs that are robust to various situations, such as in the ICU. The model must clearly communicate where how sure it is to ensure that the user understands that, if an unsure prediction is made, the current estimation should not be fully relied upon or at least what the possible ranges of the estimation are. Lastly, when using the model, it must have learned the causal relations and enable users to be able to ask questions such as, "Why do you suggest this?" or "What makes the treatment so effective on this patient?". Therefore, to create a trustworthy dosing assistant using AI, it must 1) have selected the correct variables or data to learn from for a robust model, 2) quantify and measure its uncertainty, 3) explain what causal factors drive the treatment recommendations, and 4) communicate situations where it cannot make a prediction. This dissertation proposes and presents the tools and methods to help create such an assistant that can support treatment decision making.

This dissertation will focus on ICU data and two specific ICU use cases because of the critical application domain and complex environment, however, the results are not limited to the ICU and are applicable to other domains even beyond healthcare. The first use case aims to predict the blood plasma concentration of antimicrobials used to treat bacterial infections in the ICU and propose the most optimal dose for

the patient to reduce the risk of complications, achieve therapeutic concentrations, and limit the risk of antimicrobial resistance. The second use case predicts the risk of new-onset atrial fibrillation (AF) in the ICU, a heart rhythm disorder resulting in irregular and abnormally quick heart rates that affects between 4.5% and up to 15% of patients in the ICU. AF is known to increase the risk of mortality and a longer length of stay in the ICU.

To build accurate and robust models for both use cases, we first need to identify the correct variables or features in the ICU data that are relevant for model development. A feature set containing only relevant features only aids the interpretability of the model, additionally it increases robustness by making sure the model will not focus on irrelevant data. Therefore, the first contribution of this dissertation is Powershap, a feature selection algorithm based on Shapley values, which quantify the impact that a variable or feature has on a model's prediction. Powershap compares the Shapley values of a known random and irrelevant feature to the Shapley values of all other features. The underlying idea is that an irrelevant feature should not have a greater impact than a random feature. Statistical testing is then used to determine which features are irrelevant and which are relevant for our model. Powershap offers a time-efficient method for feature selection and is at least 1.5 times faster compared to state-of-the-art (SOTA) methods while finding up to 20% more relevant features. Furthermore Powershap is published as an open-source library having currently received more than 1 million downloads.

Given the correct features, we can now build the models for each use case. To increase the trustworthiness of the solution, the models are enhanced with uncertainty quantification methods. In the first use case, I proposed three models that combine into the Quantile Ensemble that transforms the predictions into a predictive distribution. Particularly for this use case, this method can provide the user with the complete range of possible blood plasma concentration given the patient's variables, however, the method is definitely not limited to this use case. Next, these models for these antimicrobial blood plasma predictions are improved by incorporating domain knowledge to make them more suitable for actual dosing advice. This is achieved by leveraging pharmacokinetic domain knowledge about how these medications are eliminated from the body. This enhances both the prediction accuracy as well as the usability of the model and transforms the model to a dosing model.

For the second use case, I developed AF risk prediction models that utilize the inherent probability output to quantify risk. These models are constructed by matching the time of diagnosis for patients to address the issue of imbalance and avoid bias, as patients with AF tend to have a longer length of stay compared to those without AF, which could skew the predictions. Therefore, the prediction times of patients without AF are matched with AF. This essentially creates a model that learns to discern whether a patient is at risk for AF at any point during their ICU admission, only given electronic health record data. This model is also validated using data from multiple ICUs to ensure the robustness of the findings.

Both use cases quantify uncertainty, which consequently must also be measured to ensure the accuracy of this uncertainty quantification. For the first use case, I outputted a predictive distribution, and we can evaluate the calibration of this distri-

bution by verifying whether the predicted probabilities within the distribution align with observed outcomes. I propose two novel metrics for this purpose: the Absolute Distribution Coverage Error (ADCE) and the Distribution Coverage Error (DCE). The former quantifies the overall miscalibration, while the latter assesses whether the model is consistently under- or overconfident in its probability predictions. Similarly, for the AF risk models, I proposed variations that perform the same evaluation but on the risk probabilities: the adjusted Expected Calibration Error (ECE) and the Expected Signed Calibration Error (ESCE). These two metrics serve the same purpose but for risk prediction. The presented metrics are based on existing metrics and contribute to evaluating and optimizing uncertainty quantification models to achieve more trustworthy models.

Given calibrated and validated models with the correct features for the dosing assistant, we still lack a solution to questions such as, "What makes the treatment so effective on this patient?". The next contribution of this dissertation provides a method to answer these questions: Causalteshap. Causalteshap builds upon the idea of Powershap by comparing against a known irrelevant random feature. However, Causalteshap aims to identify predictive features, which are features that interact with the treatment and determine whether a treatment will be more or less successful for an individual. By applying principles from causal inference or causal AI and statistics, along with the concept of comparison to a random known irrelevant feature, Causalteshap offers a method to flag and identify these predictive features, which are essential for optimizing treatments for individual patients. This method is rigorously evaluated through extensive experiments, demonstrating its ability to find predictive features without generating much less false positives compared to current methods, thereby ensuring the method's trustworthiness, as any flagged feature is likely to be truly predictive. This method is also applied to the AF models to investigate whether a specific medication (Noradrenalin) increases the risk of AF and which variables contribute to this increased risk, illustrating how risk models can also serve as dosing or treatment assistants by answering these causal questions.

To effectively utilize the created models in practice, they must also be robust and trustworthy, especially because users will ask questions like, "What dose will be most optimal?" or, conversely, "What would have happened if I had given this dose instead?". This second question poses a counterfactual scenario, i.e. something that did not occur but might have. Most likely, a model often does not have access to all potential data and thus answering such a question will not result in a trustworthy answer. Another possibility is that the model is trained on data with a specific treatment assignment, for example a medication is given to only healthy male patients. Consequently, if asked about the treatment effect on critically ill pregnant women, the model will likely be unable to provide a reliable estimate. This is analogous to asking for the outcome of a specific dose for a patient when there is no supporting data, meaning the model has not encountered similar patients who received that dose and therefore cannot make an adequate prediction. The final contribution of this dissertation addresses this challenge by leveraging conformal prediction, a robust uncertainty quantification method. In this contribution, I quantify the probability of treatment using propensity scores, which are then used to weight and adjust the uncertainty estimation. This ap-

proach compensates for lack of treatment overlap in the data, integrating uncertainty directly into predictions. As a result, the model can explicitly indicate when it lacks sufficient basis for a reliable prediction—enhancing the trustworthiness of dosing recommendations.

With all these diverse contributions, this dissertation aims to add valuable tools to a toolbox for building an AI treatment decision support assistant, with a particular focus on medication dosing, that can be effectively used in the ICU, ultimately aiding healthcare practitioners in improving the quality of care and enhancing patient outcomes.

1
Introduction

"Even the smallest person can change the course of the future"

–J.R.R. Tolkien, The Fellowship of the Ring

Imagine a personal chef, known for making dishes to individual tastes, guiding you as you prepare a spicy meal for a friend. You want to impress your friend, so you seek advice on the perfect amount of spice to satisfy their taste. However, you have never served them spicy food before, but you know some of their preferences and personality. Suppose this personal chef has years of culinary experience. Thanks to its expertise, the chef can estimate how much any person will enjoy a specific spice level based on their preferences and personality, i.e. the individual spice satisfaction. Then, if you provide details about your friend, such as their palette and personality, the chef can then estimate this individual satisfaction for every spice level, helping you choose the best option [1].

This could already help you immensely when preparing the spicy meal. However, not everyone is the same, even if they share similar tastes or personality; Your friend might just have a naturally higher or lower tolerance for spice [2]. Instead of just estimating satisfaction, the chef can draw on his extensive experience to provide a range of satisfaction, providing minimum and maximum levels. This quantifies the uncertainty in his estimates, adding more credibility to his advice, ultimately making him more trustworthy as a culinary advisor.

Now, suppose your friend is Thai and has lived in Thailand for many years, where exposure to spicy food is common and more likely increasing their spice tolerance.

This increased tolerance could decrease the effect of spiciness and thus limit their satisfaction [3]. The chef, who also followed different cooking courses in Thailand but in a region known for extremely spicy dishes, understands how Thai people handle very spicy food and can provide accurate satisfaction ranges for high spice levels. However, if you ask about lower spice levels, the chef might lack the relevant experience to give reliable advice. The chef may not have explored or learned about the satisfaction of many different spice levels for Thai people, as non-spicy food might not be common in this region. Instead of only providing a guess estimate, the chef can admit these limitations, which is highly valuable in itself.

This hypothetical personal chef can greatly assist you in selecting the right amount of spice for your friend. Although creating a spicy dish for a friend might not be a critical domain, finding the optimal treatment, e.g. the optimal dose of a medicine, for a patient is crucial for a physician in healthcare. As such, having a similar "personal chef" for treatment suggestion, e.g. dosing, in medicine can be invaluable.

Therefore, in this dissertation, you and I will explore my proposed methods and tools that can help create such a "personal chef" with the ultimate goal of helping the decision-making of healthcare professionals and improving patient outcomes.

1.1 Need for trustworthy decision support in healthcare and the ICU

We as humans cannot escape our health. Nowadays, almost anyone eventually needs healthcare, which I define as the activity or business of providing medical services [4]. Patients are treated in hospitals for a plethora of reasons, such as infections, specific diseases, trauma, cancer and ageing-related illnesses [5]. If a patient's condition worsens to the point of requiring intensive care, they are admitted to a specialized department called the Intensive Care Unit (ICU), which is dedicated to providing comprehensive and continuous care for critically ill patients [6]. The first time someone was admitted to a department that resembles a current ICU was, at the time of writing, around 73 years ago; in 1952, during a Polio epidemic in Copenhagen [6]. However, the current ICU is vastly different. The ICU is equipped with specific medical devices and staffed by highly trained healthcare professionals who monitor and treat patients with severe injuries, infections, or organ failures. This ICU environment is highly dynamic and frequently requires rapid life-impacting decision-making with precise interventions in situations with many uncertainties.

Worldwide, there is a growing demand for ICUs due to factors such as an ageing and growing population, the need for advanced medical knowledge to treat various conditions, and increased access to healthcare in general [7]. However, ICUs are complex environments with a relative shortage of medical staff with patients having multiple, interrelated health issues that require a multidisciplinary approach for treat-

ment [7]. Additionally, ICUs must manage limited resources, including staff, equipment, and beds, while maintaining high standards of care [8]. On top of all this, the ICU is a fast-paced environment that requires accurate and rapid decision-making based on available information, as delays can significantly impact patient outcomes [9].

These challenges are significant drivers for innovation in the ICU. While many other healthcare domains also face considerable challenges, the COVID-19 crisis heightened public awareness of the ICU, acting as a catalyst for innovation [10]. Thanks to the considerable number of medical devices in the ICU, it is additionally a large source of information and data [9]. Every day, up to 100GB of data is being collected from a single critically ill patient, such as physiological signals like heart rate, but also other data like imaging and lab data [11]. Considering the rise of artificial intelligence (AI), it was only inevitable that these were also to be applied to the ICU [9]. These methods offer the opportunity to revolutionize the ICU and create "personal chefs" for healthcare professionals, especially given the available data and the challenges currently facing the ICU. These personal AI chefs offer great opportunities to support healthcare ICU staff in making fast-paced and informed decisions about the patients' health.

At the time of writing, many AI solutions or "personal chefs" for the ICU are designed and created for specific targeted use cases, such as disease diagnosis, treatment optimization of a specific medicine, or automatic alert systems [12]. A significant challenge is how to create a trustworthy AI model for these use cases, especially when aiming to build a "personal chef"-like model or a treatment model that provides advice or guidance in the ICU [9]. Despite all the data, creating models for the ICU, or even healthcare in general, comes with specific requirements. These models must be used by physicians, nurses, or other healthcare professionals, as these models will serve as tools to enhance the quality of care. Just as you would not seek advice from a chef you do not trust for cooking spicy food for your friend, healthcare professionals are less likely to use tools they do not trust [13]. This trust or trustworthiness can be defined in many ways, however, the most common view is trust directed toward the technology's capabilities. This trust is described as the willingness to submit to the vulnerability of the technology's capabilities [13]. There can be many paths to trustworthiness, but they all add to the total trust in the model [13]. First, the model must be correct, and to do so, the model should have ideally learned the ground truth or the causality behind the ground truth to give correct treatment advice [13]. If the model is simply incorrect, there is just no justified reason to trust it. Second, especially in the cases where the model might be incorrect, the model must provide its uncertainty to help clinicians understand the reliability of its predictions [13], just like the chef giving ranges of possible spice satisfaction. Third, having a robust and causal model is a necessity to analyze it and ask essential causal questions to the "personal chef", such as "Why do you suggest this amount of dose or treatment?", which in turn strengthens the user's trust [14].

Currently, many AI models, including AI dosing models, have already been published that aim to tackle various use cases in the ICU [13, 15, 16]. Unfortunately, contrary to the number of published models, almost none reach clinical adoption. Many sources refer to a lack of trustworthiness in these systems as one of the main reasons for limited adoption [7, 9, 13, 17, 18], which I also have found in a study of our own that studied the preconditions of AI in the ICU [18]. Therefore, I could phrase this issue into a global overarching challenge that I aim to tackle in this dissertation:

Overarching Challenge: "Can we create trustworthy AI models for the ICU that can assist healthcare professionals in guiding decision making for treatment?"

To approach this overarching challenge, we must explore the individual properties of a trustworthy model. To do so, we can extract three main properties for a trustworthy model from the previously mentioned requirements: Causality, Uncertainty, and Robustness. These three properties are tackled within three distinct domains in AI: 'Causal Inference or Causal AI', 'Uncertainty Quantification', and 'Validated and Robust Machine Learning'. Combining these domains aids in developing validated models from ICU data for treatment advice. The aim of these models is thus to provide trustworthy predictions about possible treatment outcomes with quantified uncertainty, analyze treatments, and determine the optimal treatment to achieve desired or improved patient outcomes in the ICU. These three domains form the technological basis of this whole book to address the overarching challenge. In the next three subsections, each domain will be briefly discussed to flesh out different sub-challenges that form the focus of this dissertation within the overarching challenge. These subsections will provide a high-level overview of the concepts necessary to formulate the specific research goals of this dissertation at the end of this chapter. More details of the State-Of-The-Art (SOTA) and rationale behind each challenge or research goal are presented in the corresponding chapters. I will start this overview with a brief discussion of the Machine Learning domain.

1.2 Machine Learning

Machine learning (ML) is a subdomain of the broad AI field. It is complex to fully define artificial intelligence [19]. On the contrary, for ML, we can simply classify it as algorithms or models that learn from data. Similar to the personal chef mentioned at the beginning of this chapter, who gains expertise through numerous culinary experiences, a ML model improves its performance by learning from data. This interpretation is extremely broad, which also shows the broadness of the ML field. The ML field is generally split into three specific categories: supervised learning, unsupervised learning, and reinforcement learning. Supervised learning refers to all models that train on labeled data and are generally used to predict the label of new, unlabeled

data. There are two types of supervised learning models: regression and classification. Regression models, on the one hand, aim to predict a continuous outcome given data, such as predicting concentrations of a medicine in the blood. Classification models, on the other hand, aim to assign one or multiple classes to a data sample, such as predicting a certain diagnosis using healthcare data. Unsupervised learning models focus on learning patterns from data without access to labels. This class of models often focuses on analyzing, comparing, or clustering the data. A medical example would be to identify abnormal patient states in ICU data, which can then be analyzed to gain insights into worse outcomes or susceptibility to specific treatments. The third category is reinforcement learning, which involves techniques that aim to find an action policy in an environment or data to optimize a specific reward, such as winning a game or improving patient outcomes. In the ICU context, an example could be finding an optimal action policy to enhance care by selecting specific procedures that reduce the length of stay. Although all three types are extremely relevant for the medical context, this dissertation focuses specifically on supervised learning.

1.2.1 Supervised Learning Models

Within supervised learning, there are numerous different models, each with its own assumptions, requirements, advantages, and disadvantages. Some of these models are used in this dissertation, and therefore, I will introduce these models briefly. Let's first discuss a well-known ML model: Neural Networks. Neural networks are the backbone of modern AI, but can also be used for simpler problems, such as cat or dog classification. These models consist of a layered network of perceptrons or neurons that feed information from the input layer to middle layers, and eventually output a result [20]. An example is shown in Figure 1.1. The inspiration for perceptrons came from human neurons, however, they have since changed immensely. Neural networks are fully configurable, where the neural network designer can choose the number of layers, the required number and type of neurons, the optimization method, and even change the whole architecture of the network. This is their strength. Additionally, there are proofs that neural networks can approximate any possible function given enough layers and neurons [21]. If the number of layers becomes large, it is referred to as deep learning. However, there are two big disadvantages of using neural networks. The first disadvantage is that these models are very data-hungry to achieve accurate performance, especially if the network has many layers and neurons [22]. However, reducing the number of inputs, i.e., dimensionality reduction, makes the learning much more efficient and thus reduces this data-hungry aspect if data size is an issue. The second is that these models tend to lack interpretability, as it is hard to understand why and how the model made a prediction. Consider a neural network consisting of k inputs and a single neuron with an activation function f, then the outcome formula is $y = f(w_1x_1 + w_2x_2 ... + w_kx_k + w_0)$. This simple formula can still

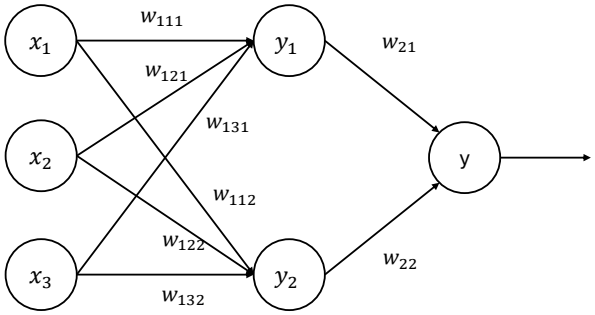


Figure 1.1: Example of a neural network with one hidden layer and three inputs

be somewhat interpreted. However, for a neural network consisting of three layers, as shown in Figure 1.1, the outcome formula already becomes very complicated to interpret: $y = f(w_{21}y_1 + w_{22}y_2 + w_{20})$ with $y_1 = f(\sum_{i=1}^{3} w_{1i1}x_i + w_{101})$ and $y_2 = f(\sum_{i=1}^{3} w_{1i2}x_i + w_{102})$. w_{20}, w_{101}, and w_{102} are the bias terms for y, y_1, and y_2 respectively. Therefore, more complex and time-intensive methods, such as KernelSHAP [23], are required to extract interpretations that can be understood by humans, making NNs less preferable in critical applications such as the ICU.

A second ML model used in this dissertation is called a Gaussian Process (GP). While not as well-known, GPs have a unique advantage: they can measure uncertainty intrinsically in their predictions, which can be very useful in many situations. Imagine you are trying to predict something, like the individual spice satisfaction. A Gaussian Process will not only give you a single prediction, but it also conveys its uncertainty for this prediction. This is because GPs are based on probabilities and can naturally handle uncertainty. A GP has a mean function and a covariance function, also called a kernel. The mean function is simply the average prediction the model makes. The covariance function helps to understand how different inputs relate to each other. For example, it can capture patterns like smoothness or repetition in the data. The kernel can include prior knowledge or assumptions about the data. For instance, if you know that the satisfaction changes smoothly over time, you can build that into the model. In the example shown in Figure 1.2, the GP model starts with some initial beliefs about how the individual spice satisfaction might change given the number of chili peppers or input X (called the GP Prior). After observing a few data points, it updates these beliefs to make more accurate predictions (called the GP Posterior). One of the benefits of using a GP is that it can show where it is uncertain, especially in areas with little data or with a lot of noise, as illustrated in the figure in the area surrounding the zero input. However, GPs have some limitations. They work well with small amounts of data, but can become very slow when dealing with large datasets, typically struggling with more than 10000 samples [24]. This threshold even becomes lower, the more

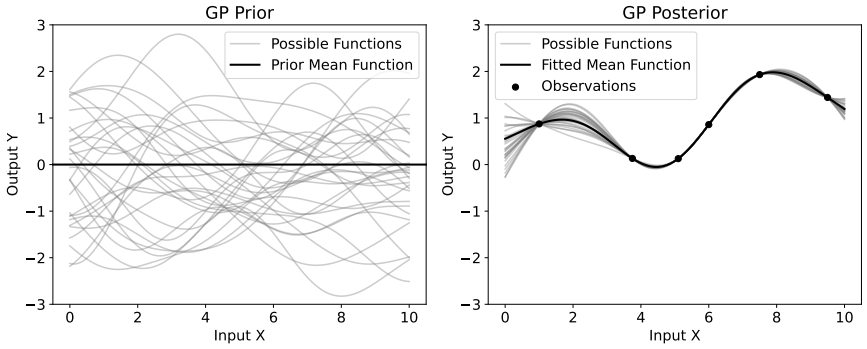


Figure 1.2: The prior and posterior (after fitting) of a Gaussian process Regression model

features there are in the model. Therefore, dimensionality reduction or reducing the number of features to the necessary minimum will also be beneficial and make the model more robust. On the plus side, once a GP is trained, it is considered easier to understand how it makes predictions compared to more complex models like neural networks.

The last model type used in this dissertation is a Gradient boosting model. These gradient boosting models are based on decision trees. A decision tree resembles a decision protocol, where multiple splits are made based on the data to predict an outcome. These splits can be automatically learned from the data to find the best possible splits to divide the data. An example of a decision tree to classify Heart Disease (HD) is shown in Figure 1.3. In this decision tree, if a patient is younger than 80 years, we explore the left part of the tree, otherwise, we explore the right part. In the left part of the tree, we then split based on blood pressure, while in the right part, we split based on heart rate. This process continues until the data is divided into regions that represent similar samples, such as leaves containing only patients with heart disease, while other leaves contain only patients without heart disease. A gradient boosting tree is a model that combines multiple of these decision trees into what we call an ensemble. To train a gradient boosting tree, the process begins with training an initial decision tree. Then a second tree is trained that is trained to correct the prediction errors of the first tree. Subsequently, the model keeps training new trees to correct the prediction errors of the combined previously trained trees, aiming to progressively reduce the prediction errors. This iterative process continues for a predefined number of iterations. After fitting all the trees, the final model then predicts by aggregating the predictions of all individual trees. This approach typically leads to improved predictive performance compared to a single decision tree [25]. Several variants of gradient boosting models exist, with CatBoost currently recognized as the most robust for tabular data, often outperforming other models, even very complex neural networks [25–27]. As ICU and clinical health record data are often organized in tabular formats, such as Excel

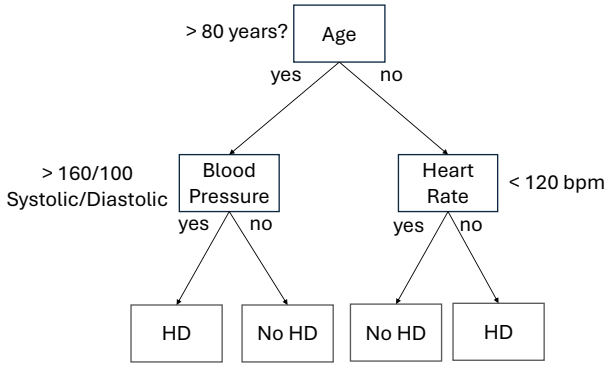


Figure 1.3: A simple Decision tree example to classify Heart Disease (HD)

spreadsheets or SQL databases, these are well-suited for CatBoost. Moreover, compared to NNs or GPs, CatBoost can intrinsically manage missing data, which is abundant in clinical datasets. Nevertheless, CatBoost models can be prone to overfitting, especially with features that include more noise than ground truth, and careful selection of these features is essential for a robust model. While gradient boosting models are not inherently interpretable due to combining a large number of interpretable trees, numerous techniques can be employed to extract post-hoc explainability from these models. Examples include using feature importance [25] and Shapley values through TreeSHAP [23, 28].

1.2.2 Explaining a supervised ML model with Shapley Values

Once an ML model is trained, it can be analyzed, interpreted, and explained. For example, if the model is a Gaussian Process, the interpretation is possible by examining the learned parameters for each feature, although this is also abstract, like the NN. Analyzing a Gradient Boosting model is more complex due to the number of trees, and it is even more challenging for a neural network. For a gradient boosting tree, it is possible to aggregate the feature importances of every decision tree and look at how much the feature contributes to the total prediction in percentage. Another method is to resort to Shapley values. This can be done with TreeSHAP for Gradient Boosting models or KernelSHAP for neural networks or other models [28]. The SHAP method quantifies the impact of each feature on the prediction, represented as Shapley values, and visualizes it in Figure 1.4. The Shapley value can be either positive (red in Figure 1.4), indicating it increases the prediction, or negative (blue in Figure 1.4), indicating it decreases it. We also calculate the baseline Shapley value of the model ($E[f(x)]$ in Figure 1.4), which is simply the mean prediction of the model over a given dataset. As seen in Figure 1.4, a favorable property of Shapley values is local accuracy, which states that for a given data sample, summing the calculated Shapley

values of all the features with the baseline Shapley value $E[f(x)]$ will be the same as the model's prediction $f(x)$. Mathematically, given the baseline Shapley value S_b, the model M, sample X_i with K features, and Shapley value $S(X_i^k, M)$ calculated on model M given the value X_i^k of feature k, the local accuracy is defined as:

$$M(X_i) = S_b + \sum_{j=1}^{K} S(X_i^k, M) \tag{1.1}$$

To calculate the Shapley value of a feature of interest, various random subsets of all features in the model are selected. The prediction is then calculated twice for each subset: once with the feature of interest included and once without it. The differences in these predictions across all subsets are aggregated to determine the final Shapley value for that feature. The results for a whole model can then be visualized, such as in Figure 1.5. In the figure, you can, for example, see that a very high creatinine clearance (CL_{CR}) results in lower predicted values, indicating a reverse relationship between increasing CL_{CR} and the predicted output. There are two main versions used: KernelSHAP and TreeSHAP. KernelSHAP is an approximation method and is model-agnostic, however, it has a higher time complexity compared to TreeSHAP. KernelSHAP for neural networks, for example, is very time-consuming and often not practical at inference time [23]. Compared to KernelSHAP, TreeSHAP is exact and much faster, making it much more feasible at inference as well, but only applicable to tree models [28].

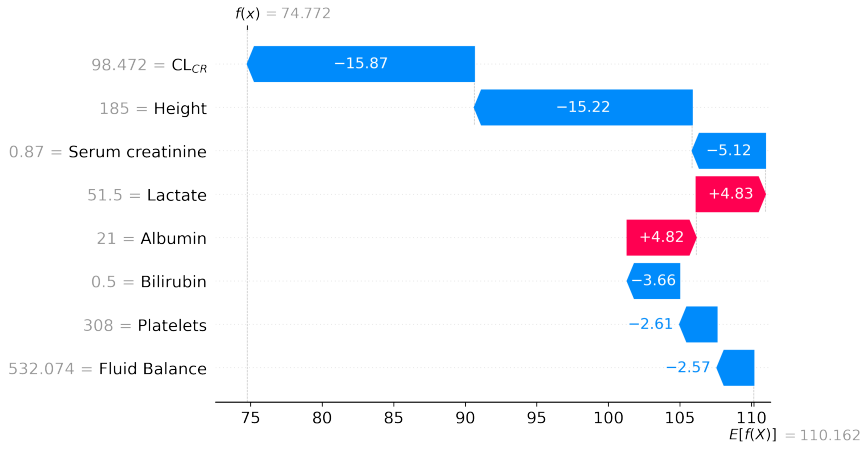


Figure 1.4: A waterfall plot of a single prediction of a model that predicts blood plasma antibiotic concentration (in mg/ml). CL_{CR} = Creatinine Clearance.

If a feature is not important or irrelevant, the Shapley value should ideally be zero. However, in TreeSHAP, this is often not the case, and there will be Shapley

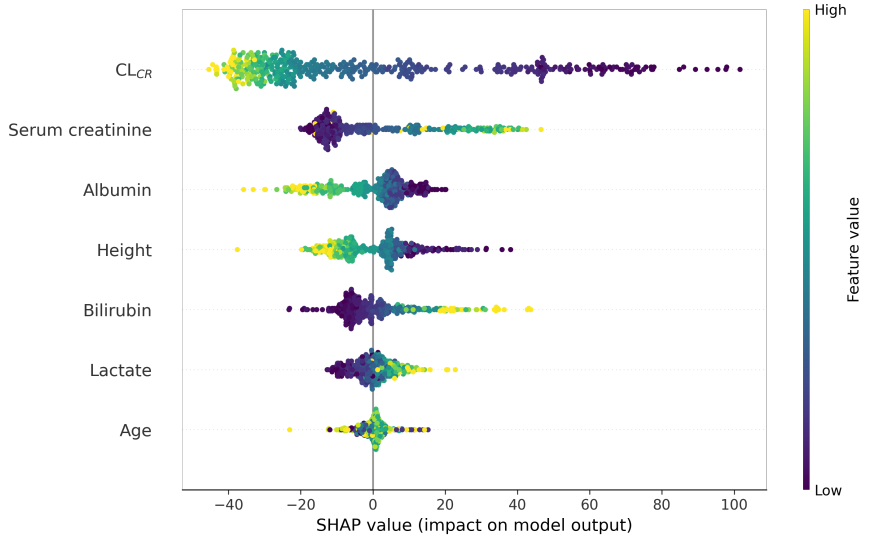


Figure 1.5: A Shap beeswarm plot of a model that predicts blood plasma antibiotic concentration (in mg/ml). CL_{CR} = Creatinine Clearance.

values attributed to noise. As TreeSHAP is exact, it quantifies the effect of every feature in the tree, exactly. However, in random forests or gradient boosting trees, splits are still made on noise features due to the nature of the algorithm; therefore, these noise features will also get Shapley values [29]. This effect is much less severe in KernelSHAP. Consequently, if we add a feature that we know is irrelevant, i.e. a noise feature that contains no information, this feature will then also receive a non-zero Shapley value using TreeSHAP. In general, it is hard to know beforehand if a feature is irrelevant or important without domain knowledge.

1.2.3 Feature Selection for a Robust Supervised ML Model

The end goal is a robust ML model, and building a robust supervised ML model from sample data requires several steps. We need to analyze the data to understand its contents, e.g. different included features and their properties, and the problem's requirements, e.g. dealing with class imbalance or uncertainty. Afterwards, we must preprocess and clean the data, deal with missing data, and extract derived features from the original data (i.e. feature engineering), to make it suitable for extracting relevant information for the model. Subsequently, we optimize the model by fine-tuning its settings, hyperparameter tuning, like for example tuning the number of layers in a NN, and refining the used set of features to identify the most optimal configuration, i.e. performing feature selection. This whole iterative process ensures that the model performs accurately and reliably. However, an essential part of this process are the

included features, because if these are wrong or noisy, your model will not be able to learn the correct relationships. Additionally, the explanations using Shapley values, for example, will mainly give meaningless results. Therefore, I will focus on feature selection, as correct and relevant features are the foundations of a robust model.

In an ICU setting, the features included in or derived from a data set might, for example, be the mean of a lab value over the past 12 hours, the mean blood pressure over the past 6 hours, age, diagnoses, etc. Ideally, features should be engineered to include as much relevant information as possible, also incorporating domain knowledge about the problem if possible. There are currently various software libraries available to extract features automatically [30]. However, this process of feature engineering can result in many possible features, some of which may be irrelevant or redundant. Often, we do not know which ones are actually relevant to our problem and which are not. Additionally, reducing the number of features, especially removing the irrelevant ones, in a model can have various benefits [31]:

- **Faster Training**: Fewer features can speed up training times;
- **Improved Interpretability**: A model with fewer features is easier to interpret, analyze, and explain. Think about having to interpret 1000 different features in a model compared to only 5.;
- **Enhanced Performance**: With limited data, irrelevant features can degrade model performance. Think of it like learning a topic from a single relevant book versus an entire library given only a day.;
- **Lower Operational Costs**: Gathering fewer features can reduce the cost of operating the final model. If you need to calculate 100 different lab values for a patient compared to only the 10 relevant ones, naturally, this is cheaper.;

Therefore, a proper feature set is essential for a robust model, regardless of whether it is a GP, NN, or Gradient Boosting model. From a trustworthy AI perspective, having no noise or irrelevant features reduces the chance of overfitting on noise features and reduces the possibility of making a wrong prediction as well [31].

This process of selecting the correct features or limiting the number of features is known as feature selection, which is a crucial step for optimizing an ML model. Therefore, it is crucial that this is done correctly and can thus also take a lot of time. A common but strong method is forward feature selection, which is an iterative process that first starts with no selected features and trains a model on every feature individually. Then it adds the feature with the highest performance to the selected features. It then retrains models on the selected feature combined with every other feature one by one, and again selects the feature resulting in the highest performance when combined with the already selected feature. This iterative process continues until adding more features no longer enhances performance. Although strong, this method is extremely

slow and can be a huge time-sink when developing a model. There are many improvements or alternatives to this feature selection algorithm, however, they all take a lot of time as they iteratively train many models [32]. This class of algorithms are called wrapper methods. There are also feature selection methods that are quick, however, these do not train a model and solely rely on the data. Common examples are statistical tests, such as the F-test or the χ^2 test, that select features based on thresholds or statistical calculations. These are called filter methods. Although quick, they often lack in performance and are dependent on many assumptions [32].

Considering the disadvantages of these two method types and the requirement of selecting correct features to build a trustworthy and robust model, I can create a first research challenge that aids in tackling the overarching challenge by combining the advantages of both filter and wrapper methods, i.e., quick and accurate:

C1: "Can we design a time-efficient feature selection method to identify relevant features without loss of performance?"

1.3 Causal AI

Going back to the personal chef analogy; Suppose you created a dish for your friend. The dish was fine, but not spectacular. You could ask yourself or the chef "What if I would have added butter instead of olive oil? Would it have been better or not?", or "What caused the dish to be just fine? What if I added more spice?" To quantify or answer these questions, you can use causal inference. If you want to ask these questions to a "personal chef" or an ML model, this requires Causal AI. Causal AI is the intersection of causal inference and ML. The primary goal of causal inference is to address "what-if" or "what caused it" questions, thereby uncovering or using cause-and-effect relationships in the data [33]. Causal inference has been applied across various domains, including medicine, agriculture, policy, and marketing, and has established the Randomized Controlled Trial (RCT) as the gold standard for identifying causal relationships [34].

In medicine, causal inference is crucial for determining which medications are effective for specific diseases, understanding the causes of symptoms, and developing treatment plans for complex patients [14]. Traditionally, when a new medication is developed, it is tested using RCTs. In an ideal RCT, a random population of patients is selected and divided into two groups: the Treatment group and the Control group. The control group receives no treatment, while the treatment group does. After the trial, the results from both groups are compared, and the outcome difference between the two groups is known as the average treatment effect (ATE). This is similar to creating two dishes, one with butter and one with olive oil, while keeping all the other ingredients and techniques the same. A positive treatment effect indicates that

the treatment yielded better outcomes than no treatment. RCTs are effective because the random assignment and selection of patients eliminate biases related to treatment assignment or patient selection, allowing for the identification of the true treatment effect. This treatment assignment bias can be caused by anything that influences the administration of the treatment, such as assigning more medication to older people while analyzing the effect of the medication on mortality. If you do not compensate for the age, it could indicate that this medication increases mortality. However, conducting RCTs is often more complex than it seems. Ethical concerns may prevent conducting RCTs on certain patient populations, such as randomly removing ventilation in the ICU or simply conducting trials on pregnant women. This can limit the options to analyse certain treatments or limit possible conclusions. Additionally, RCTs can be prohibitively expensive and may not always be feasible [35]. With the abundance of data today, new techniques are available to uncover causal relationships from biased RCTs or observational data without even needing to perform RCTs or rely on randomization. This can be achieved by combining causal inference with ML, opening new avenues for understanding and leveraging causal relationships in data. However, many of these techniques require you to know, at least structurally, how the data is generated to account for specific biases resulting from the causal relationships. These causal relationships are traditionally visualized in a causal graph, which we as humans can easily interpret and are therefore an essential part of causal AI.

1.3.1 Causal Graphs

As mentioned, causal relationships can be represented using a Directed Acyclic Graph (DAG) or a Causal Graph, which is a directed graph without cycles. An example of a hypothetical DAG is illustrated in Figure 1.6. In a DAG, an arrow signifies a direct cause-and-effect relationship. In this particular DAG, a bacterial infection leads to the prescription of antibiotic treatment and can also prolong the length of stay in the ICU, especially in cases of resistant strains or hospital-acquired infections. Successful antibiotic treatment can reduce the length of stay by addressing the infection [36]. However, certain antibiotics, such as vancomycin, may cause kidney damage [37]. Additionally, a longer ICU stay increases the risk of complications, potentially affecting kidney function. A DAG serves as a visualization of the assumed ground-truth cause-and-effect relationships within a problem. However, the true DAG is often much more complex. Understanding a DAG helps to identify structures that could introduce bias or negatively influence analysis. Three important structure types could influence results: Confounders, colliders, and mediators. In this scenario, a bacterial infection acts as a confounder. A confounder affects both the treatment (antibiotics) and the outcome (length of stay in the hospital). If we do not account for the infection properly, we might see false connections between the treatment and the length of stay. Kidney problems serve as a collider. A collider is a variable that is influenced

by both the treatment and the outcome. Adjusting or accounting for kidney problems can create false correlations, leading to biased results. Therefore, we should exclude kidney problems from our analysis. Lastly, the antibiotic treatment acts as a mediator between the bacterial infection and kidney problems. A mediator is a variable that explains the process through which one variable affects another. Without considering this mediator, it might seem like the infection directly causes kidney issues. However, it is actually the antibiotic treatment that leads to these problems. By including this mediator in our analysis, we can see that the direct correlation between infection and kidney issues disappears. Consequently, knowing the underlying DAG or causal graph is vital for causal inference problems to correctly estimate causal relationships. Additionally, many problems in ML, especially if the aim is a robust model, can also be represented using a causal graph. For example, a big issue is data drift, where the data is changing due to a shift in the causes. If the model did not properly account for these causes, or confounders even, this can introduce major biases and reduce the robustness and therefore the trustworthiness of the model.

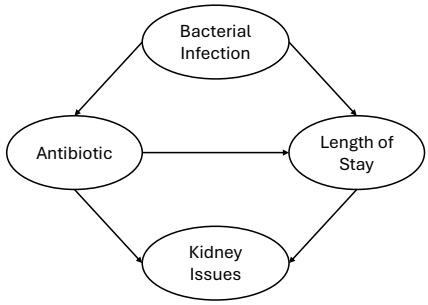


Figure 1.6: Example of a DAG regarding antibiotic treatment in the ICU and its effect on length of stay and kidney issues.

1.3.2 Treatment Effect Estimation

In Causal AI, there are two subdomains: Causal Discovery and Treatment Effect Estimation. Causal Discovery aims to uncover the unknown DAG from data [38], while Treatment Effect Estimation seeks to quantify the impact of a treatment or an action on an outcome [14]. In our analogy, the treatment effect estimation would be quantifying the effect of using butter instead of olive oil in your dish on the satisfaction of your friend. Given the presented ICU use cases and questions, the Treatment Effect Estimation subdomain is of more interest for this dissertation.

The treatment effect is defined as the difference between the outcome under treatment (Y^1) and the outcome under control (Y^0): $Y^1 - Y^0$. In practice, only one of these outcomes can be observed for each individual, as it is impossible to both admin-

ister and withhold treatment simultaneously. The unobserved outcome is referred to as the counterfactual. However, it is possible to calculate the average treatment effect in the data. The average treatment effect can be calculated by subtracting the average outcome under control from the average outcome under treatment. However, this approach does not account for heterogeneity among individuals, such as varying susceptibility to treatment. For more personalized healthcare decisions, it is crucial to understand the treatment effect at the individual level or conditioned on relevant patient parameters. This is known as the Conditional Average Treatment Effect (CATE) [14]. The CATE requires estimating the counterfactual as we still subtract the outcome under control Y^0 from the outcome under treatment Y^1, but we do this per sample where we only observed either Y^0 or Y^1. This counterfactual estimation can be achieved using ML models trained on data from Randomized Controlled Trials (RCTs) or observational studies. There are various models to do this CATE estimation, with meta-learners being among the most popular [14]. Meta-learners, such as S-, T-, R-, and X-learners, are configurations or frameworks where you can substitute various ML models to estimate the CATE [39]. A visualization of how the S- and T-Learner work is shown in Figure 1.7. Additionally, some also work for continuous treatments, such as the S-learner, where the treatment can be viewed as an adjustable dose, such as antibiotic dosing. These models are referred to as dose-response models and try to provide the treatment effect on the outcome for all doses at a particular moment in time.

However, a critical aspect of many treatment effect estimation methods is addressing treatment assignment bias, where treatment groups are influenced by certain decisions or have different distributions [14]. For example, patients with renal dysfunction may receive lower antibiotic doses, or pregnant women may be excluded from treatment unless absolutely necessary. This assignment bias can also occur in an RCT; When there is non-compliance. Non-compliance happens when some were assigned treatment, they, due to various reasons, still did or could not take the treatment. This bias can lead to unfair comparisons, similar to the causal graph example previously, and thus potentially harmful conclusions. For example, we could conclude that the antibiotic reduces kidney damage, however, this is merely because we give fewer antibiotics to those with kidney damage to reach the same target as the kidneys cannot clear the antibiotic anymore. If we were to naively compare the kidney status of those receiving high doses of antibiotics compared to lower doses, it would show that there is more kidney damage in those receiving lower doses. To mitigate this issue, it is essential to adjust for confounders and have a notion of the causal graph of your problem. In this way, we adhere to the unconfoundedness assumption of meta-learners, which states that all possible confounders are identified and accounted for [40]. One way to measure treatment assignment bias is through the propensity score, which represents the probability of a person receiving the treatment [41]. A high propensity score indicates potential bias. In Randomized Controlled Trials (RCTs) with binary treatments

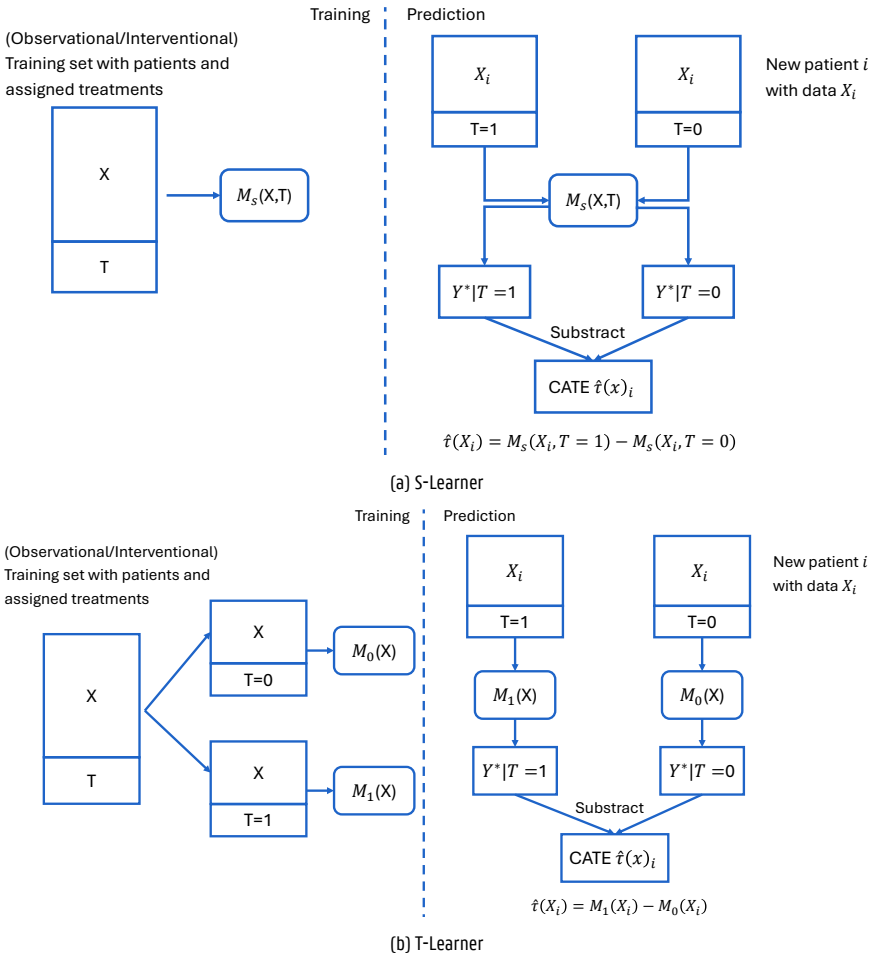


Figure 1.7: Visualization of two meta-learners: The S-Learner and the T-Learner. M represents any ML model. T = Treatment. Y = Outcome.

(control or treatment), the ideal propensity score is 0.5, i.e. 50% chance you will get the treatment or not. Likewise, for continuous treatments, the goal is a uniform distribution where each dose has an equal likelihood of being assigned, e.g. if there are 50 different doses, you have a 2% chance of receiving any dose. Violating these conditions can also undermine the overlap or positivity assumption. This assumption states that patients in all treatment categories have similar characteristics. For instance, if no pregnant women receive the treatment, there is no overlap for this subgroup, and the estimations on pregnant women may not be reliable. Therefore, assessing and compensating for overlap is crucial when using dosing or treatment models, especially

in continuous dosing scenarios with numerous dosing options. Currently, treatment effect or dosing models assume overlap exists and thus make it a requirement [14]. However, when these models are used for dosing or treatment advice, it is vital that they also provide trustworthy predictions in regions with limited overlap. This leads to the next challenge:

C2: "Can we develop methods for treatment models to provide trustworthy predictions that also compensate for regions with limited or no overlap?"

In addition to quantifying treatment effects and knowing whether there is overlap or not, causal inference also aims to uncover causal relationships. In treatment optimization, for example, it is crucial to identify which patients will benefit from a treatment and which may experience harm. If a CATE model is accurately learned, analyzing this model can reveal factors that influence the treatment effect. This is like asking the personal chef: "What makes someone like or dislike spicy food"? These factors, known as predictive variables, provide insights into optimizing treatment strategies and are essential for optimizing care and thus also important for the overarching challenge. Traditional methods to identify these predictive variables mostly use linear models [42, 43], such as target transformation using Ordinary Least Squares Regression. Furthermore, the traditional predictive variables identification methods output many false positives, especially when presented with many possible candidates, making them less reliable [42]. However, if the treatment effect is more complex or not linear, more complex methods such as causal AI are necessary to estimate the treatment effect, as explained earlier in this section with the meta-learners. When estimating the CATE or the treatment heterogeneity of patients, the current state-of-the-art uses meta-learners, which, in turn, can utilize any ML model and thus also non-linear models. To find the predictive features in these meta-learners, they currently simply inspect the models, e.g., using SHAP. However, this method does not discern between features that impact the treatment effect and features that only impact the outcome and is thus prone to false positives. Therefore, we require a method that can identify these predictive variables for more complex treatments that work for causal AI methods, leading to the following challenge:

C3: "Can we identify and analyze variables that influence or interact with treatment effects in a given Causal AI model?"

1.4 Uncertainty Quantification

At the beginning of the chapter, I introduced a hypothetical personal chef who could estimate how satisfied an individual might be with different spice levels. However, this

single estimate provides limited insight. In contrast, if the chef provides a range within which they are certain the individual's satisfaction lies, we gain a deeper understanding of both the chef's expertise and the diverse preferences of individuals. If the range is large, it might indicate that the chef lacks expertise in making accurate guesses for that person, or that this person or likewise persons are less predictable. Similarly, in ML, quantifying uncertainty can significantly enhance the trustworthiness of a model by acknowledging and accounting for possible differences in the prediction compared to the ground truth. Uncertainty Quantification (UQ) in ML is a subdomain focused on measuring the uncertainty associated with predictions made by ML models. This process aims to address potential model errors, biases, assumptions, and natural variations or noise in the data [44]. There are two primary types of uncertainty in this context, which are also visualized in Figure 1.8 [45]:

- **Aleatoric Uncertainty:** This type is also known as data or irreducible uncertainty and is inherent to the data itself. It includes things such as the data noise or the inherent randomness. In the analogy, the natural difference in individuals' spice tolerance can be seen as Aleatoric uncertainty.

- **Epistemic Uncertainty:** This type is often referred to as reducible uncertainty, and consists of two parts. The first part is uncertainty introduced due to observation limitations, such as missing data, unobserved variables, or limited data. This uncertainty could be reduced by gathering more data. The second type can be seen as model uncertainty. This type arises from the ML model used, the assumptions made, the trained parameters, model biases, or the selected features. In the analogy, epistemic uncertainty could include the experiences of the chef from which estimations are made.

Some sources categorize missing or limited data as aleatoric uncertainty, as this perspective views the data as something that cannot be changed anymore or that gathering more data is not possible and thus irreducible [45]. This discrepancy in defining the two types of uncertainty complicates the estimation of either, due to differing perspectives [46].

Most UQ methods aim to estimate the total or predictive uncertainty, which combines both aleatoric and epistemic uncertainties. This predictive uncertainty accounts for all uncertainties when making a prediction and is crucial for developing more trustworthy models, particularly in fields like medicine. It quantifies how much a model's prediction might deviate from the ground truth, regardless of the source of uncertainty [47]. If the provided uncertainty is correctly specified, i.e., the estimated probabilities correspond with the observed probabilities, we call the method calibrated, which is necessary to make the uncertainty quantification reliable [47].

There are already efforts to incorporate UQ in ML models for medicine, however, many of the current models do not provide calibrated UQ [15, 48, 49]. If we narrow this down to treatment decision-making, calibrated solutions become even

more sparse [50]. Given that calibrated predictive uncertainty is essential for more trustworthy ML models and that calibrated UQ is lacking in ML models for clinical decision-making [48, 50], this issue can be postulated in the following research challenge:

C4: "Can we integrate calibrated uncertainty quantification methods into the proposed models to enhance their trustworthiness for clinical decision-making?"

1.4.1 Uncertainty Quantification Tasks

The goal of quantifying uncertainty differs between regression and classification tasks [16]. In regression, a ML model provides a point prediction for a given data sample. This single value represents the expected outcome based on the features in the data sample, assuming a deterministic model. When incorporating uncertainty quantification into regression, there are two major approaches: prediction intervals and predictive distributions.

- **Prediction Interval:** This method gives us a range with a lower and upper limit. We expect the true value to be within this range, with a certain level of confidence. For example, if the confidence is 90% for a particular interval, then this means that, for a given set of data, the actual value will fall within this range 90% of the time. Techniques such as quantile regression or conformal prediction can help us achieve this [51].
- **Predictive Distribution:** This method goes further by providing a distribution from which any prediction interval can be derived, regardless of the confidence level. Essentially, it represents the distribution of the model's deviation from the true values. This can be accomplished using Bayesian methods, such as Gaussian processes [24] or Bayesian Neural Networks [52], as well as non-Bayesian methods, such as Conformal Predictive systems [53].

These prediction ranges or distributions can be calculated in two ways: across the entire dataset, called marginally, or for individual samples, called conditionally. When calculated across the entire dataset or marginally, the ranges are generally the same for every sample. However, when calculated for individual samples or conditionally, the ranges can vary based on the specific characteristics of each sample.

In classification tasks, there are also two major methods to present uncertainty: set predictions and calibrated probabilities. Many classification models inherently provide probabilities through functions like softmax or logit. However, these probabilities are often inaccurate [54]. For instance, if a model predicts an 80% probability for a particular class, a natural interpretation is that we might expect the model to be incorrect 20% of the time, which is often not the case [54]. To address this, the first method

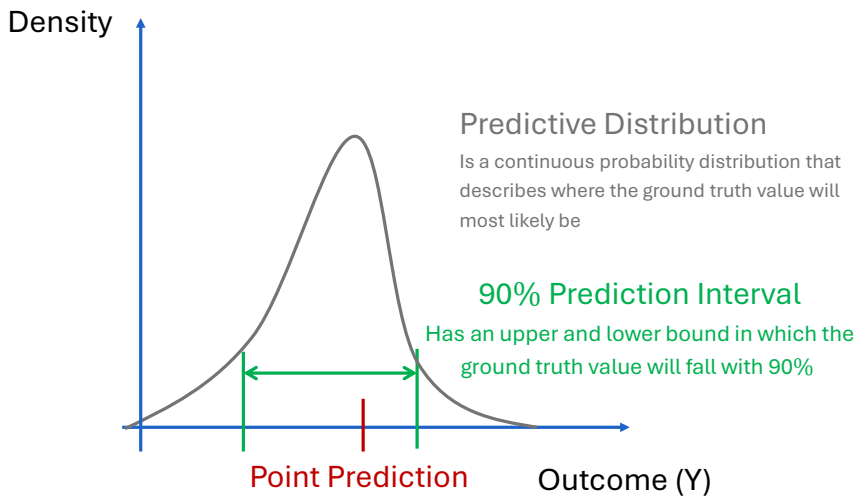


Figure 1.8: A visualization of the difference between a point prediction, prediction interval, and a predictive distribution.

involves recalibrating these probabilities. This process remaps the original probabilities to new values where this natural interpretation is correct. Common recalibration techniques include Platt scaling [55], isotonic regression [56], and Venn-Abers predoctors [57].

The second method, set predictions, presents uncertainty as a set of uncertain outcomes. Instead of predicting a single class with the highest probability, this approach provides a set of potential classes. For example, when diagnosing various diseases, if the model is unsure the model might output a set of possible diseases for the patient rather than a single diagnosis. With set predictions, you can establish a confidence threshold, such as 90%, which means that the true class should be included in the prediction set in 90% of cases. This method can be effectively implemented using conformal prediction for classification, which ensures that the prediction sets have valid coverage guarantees [58].

1.4.2 Uncertainty Quantification Metrics

Regardless of the application or prediction task, each method has its own assumptions and requirements. For example, Bayesian methods assume a prior distribution, which can strongly influence results if incorrectly specified, while conformal prediction requires a calibration set but makes no distributional assumptions. Regardless of the method, to enhance a model with UQ for trustworthiness, the UQ must be accurate, and the probabilities, distributions, or sets must cover the ground truths with the specified uncertainty level.

For each UQ goal in regression and classification, different metrics must be used [59]. For prediction intervals, we can measure the width of the interval, which is in the literature referred to as sharpness or efficiency [59]. A larger interval for the same confidence level indicates higher uncertainty, so the aim is to achieve the smallest interval possible while maintaining correct coverage. This is a balance exercise, because more coverage means larger intervals and vice versa. Similarly, for set predictions in classification, the sizes of the prediction sets should ideally be as small as possible, as this would suggest more certainty. This is referred to as the Set Size.

In addition to sharpness for prediction interval, there is also a coverage metric that calculates whether the uncertainty method correctly covers the uncertainty of the problem: The Prediction Interval Coverage Probability (PICP). The PICP estimates the observed coverage on a dataset, ideally a test set, by counting how many ground truth values fall within the given prediction intervals. If the PICP matches the specified confidence level, the intervals are considered well-calibrated [60]. There is also a similar classification variant for probabilities, which is the Expected Calibration Error (ECE) that quantifies the coverage error of the probabilities of a classification model [54].

For distributions, evaluating the UQ is more complex. The width can be measured using standard deviation, however, this metric can be skewed depending on the distribution and does not quantify the kind of distribution. Additionally, evaluating all possible intervals within the distribution is challenging, especially if we want to condense it into a single value for optimization that directly quantifies the coverage. Besides pure calibration, biases such as overconfidence, where the observed coverage is lower than specified, are often not included and must also be quantified for a proper evaluation. For predictive distributions, there is the Probability Integral Transform (PIT) [61] that is a tool to understand calibration, however, this mainly requires a subjective interpretation from a histogram plot or statistical testing, which are not ideal for optimizing for calibration. A second possible metric for predictive distributions is the Continuous Ranked Probability Score (CRPS) [62]. CRPS is a strictly proper scoring rule that jointly assesses calibration and sharpness (lower is better). While it can be optimized directly, its scale-dependence necessitates normalization for comparisons across datasets with differing units and lacks meaning unless for comparison to other models. In classification, similar challenges arise when quantifying the coverage of probabilities or confidences for set predictions. Calibration metrics like the ECE must bin the probabilities and calculate the mean coverage error across these bins. ECE quantifies calibration error by binning predicted probabilities, but its results depend on the binning strategy; different binning sizes can result in different ECEs [54]. Adaptive binning (e.g., equal-mass bins) reduces arbitrariness but does not eliminate it.

All these issues lead to the following and final research challenge of this dissertation to contribute to the overarching challenge:

C5: "Can we develop a comprehensive metric to quantify the coverage error and biases, such as overconfidence or underconfidence, of uncertainty quantification methods across all confidence levels?"

1.5 The Research Goals

As already stated, the main aim of this work is to research algorithms, methods, and metrics to be able to design trustworthy treatment models, i.e., the "personal chef" for healthcare users, which is formally postulated in the overarching challenge of this dissertation. Below is a summary of the different research challenges this work aims to address, as elaborated in the previous sections:

- **Overarching Challenge:** Can we create trustworthy AI models for the ICU that can assist healthcare professionals in guiding decision making for treatment?
- **C1:** Can we design a time-efficient feature selection method to identify relevant features without loss of performance?
- **C2:** Can we develop methods for treatment models to provide trustworthy predictions that also compensate for regions with limited or no overlap?
- **C3:** Can we identify and analyze variables that influence or interact with treatment effects in a given Causal AI model?
- **C4:** Can we integrate calibrated uncertainty quantification methods into the proposed models to enhance their trustworthiness for clinical decision-making?
- **C5:** Can we develop a comprehensive metric to quantify the coverage error and biases, such as overconfidence or underconfidence, of uncertainty quantification methods across all confidence levels?

The overarching challenge focuses on building models for the ICU with treatment goals or treatment analysis in mind. Each individual research challenge also addresses parts of the overarching research challenge. Therefore, we can combine the different challenges and postulate an overarching research goal that I will aim to address. This leads to the overarching research goal of this dissertation (RG):

Overarching RG: "Build trustworthy machine learning models for the ICU that can be used for treatment decision support while incorporating calibrated uncertainty quantification."

To build these models from ICU data, it is crucial to identify relevant variables or features from the vast amount of possible features. Therefore, addressing C1 first is essential for a more time-efficient feature selection process. We can draw inspiration from the strong performance of wrapper methods and the speed of filter methods to potentially address the disadvantages of both. I can now formalize the first specific research goal to address this challenge:

RG1: "Create an effective and time-efficient method for combining wrapper and filter feature selection techniques to identify relevant features, leveraging the strengths of both approaches."

Once the relevant features are identified, the models must also provide calibrated uncertainty quantification. C5 focuses on ensuring that calibration can be properly measured and biases quantified across all confidence levels. It is not necessary to fully reinvent the wheel, and we can draw inspiration from the available metrics, such as the PICP and the ECE, and modify these to tackle C5. Thus, I can now postulate the second research goal:

RG2: "Adjust current metrics to quantify coverage error and biases, such as overconfidence or underconfidence, in uncertainty quantification methods across all confidence levels for both regression and classification tasks."

The next step in developing the "personal chef" involves analyzing treatment models to identify the predictive variables that are truly influential in determining the treatment effectiveness. This was postulated in C3, where we can specify the causal AI more in detail by focusing on CATE models. It was also mentioned that current methods yield high false positive rates, which must be taken into account when researching a solution. I can now translate this to the third research goal as follows:

RG3: "Identify predictive variables in conditional average treatment effect models while minimizing false positives for higher trustworthiness."

Finally, we need to prepare the treatment models to explore different treatment options or doses, much like asking a personal chef how varying spice levels affect satisfaction. Instead of merely predicting outcomes, we must predict counterfactual scenarios under different doses. This requires addressing the possible limited overlap between treatments, as outlined in C2. There are multiple paths to solve C2, however, considering the overarching RG with UQ, I will address this using UQ, leading to our final research goal focusing on solving C2 with UQ:

RG4: "Enhance dosing models to identify regions with limited or no overlap in

treatment outcomes utilizing uncertainty quantification for reliable dose-response predictions."

There are two specific use cases that I will use in this dissertation to address the overarching RG and validate the specific RGs in this dissertation. Both use cases were selected in collaboration with physicians from the Ghent University Hospital ICU, to ensure medical relevancy and evaluate trustworthiness. For each use case, I will also define a more detailed Use Case Goal (UCG), which also focuses on the use case innovation and explicitly incorporates C4.

The first use case is the diagnosis of atrial fibrillation (AF) in critically ill patients, which affects between 4.5 to 15% of patients [63]. AF is a heart rhythm disorder causing irregular and abnormally quick heart rates [63]. Although curing AF is generally not possible, knowing that a patient has or is developing AF can influence care and treatment decisions to greatly reduce the accompanying risks in the ICU, lowering potential mortality or reducing length of stay (LOS) [63]. In this use case, the ML model or the "chef" could be used to examine a patient admitted to the ICU and assess whether that patient is at risk for developing AF or not at that moment. In turn, the model could enhance the quality of care in the ICU if the risk is properly predicted. This extends the range of possible questions one could ask, just like asking the personal chef for more detailed information. Examples are "Are there other (yet unknown) factors that influence the risk of AF, such as medication?" "Can we lower the risk of AF by changing these factors?", and "What if we would not have given the medication, what would be the effect on the risk of AF?". All these questions require different algorithms and methods and can be investigated, given a proper AF risk model. As trustworthiness is a part of the overarching RG, the model must also be trustworthy in predictions. Given the heterogeneity of ICU patients, developing an unbiased model that does not favor specific populations is particularly challenging and thus must be explicitly added to the research goal to account for it. I can now formalize the first use case goal, creating the following UCG:

UCG1: "Build a trustworthy atrial fibrillation risk prediction model for ICU patients that accounts for the different ICU populations between AF and non-AF patients while providing a calibrated risk probability for any time point."

This use case goal, created by specializing the overarching research goal on the AF use case, involves creating a classification model with calibrated uncertainty quantification for ICU patients. This AF risk model is necessary to ask questions, such as the examples stated above, to analyze treatments.

For the second use case, I address a regression problem with a stronger focus on dosing advice: combating (bacterial) infections in the ICU. These infections can cause severe pneumonia, i.e., lung infection, or in the extreme case sepsis, which is a

critical life-threatening condition that occurs when the body's immune system has an extreme response to an infection [64]. The dose must be sufficient to achieve specific targets, like antibiotic concentration in the blood plasma, to eliminate the bacteria. However, the dose should not be too high, as this can cause toxicity, nor too low, as this may fail to treat the patient and potentially increase the risk of antimicrobial resistance [65]. These antibiotics are often given in a one-size-fits-all approach, not tailored to the patient, risking either overdosing or underdosing the patient with the respective consequences [65]. Here, models can be used to find out what ideal dose is required for an individual patient and provide treatment dosing advice. Many of the questions for the AF use case can be applied to antibiotic treatment as well and are thus more general than previously stated. Similarly, I can formalize a use case goal for this use case as follows:

UCG2: "Model blood plasma concentrations of antimicrobials in critically ill ICU patients for optimizing antimicrobial dosing while providing calibrated uncertainty quantification for trustworthy predictions."

A more in-depth rationale, use case innovation, and impact of these use cases for both UCG1 and UCG2 will be detailed in their respective chapters. These two UCGs mainly address the model building of the overarching research challenge and also incorporate the UQ from C4 for a classification and a regression use case.

By addressing these research goals, I aim to develop key components for a trustworthy treatment model specifically for the ICU to tackle the overarching research challenge and provide guidance for dosing decision-making.

1.6 Chapter Overview

This thesis contains six chapters that address the presented research goals. Each chapter is based on published or submitted work in international conferences or journals.

	UCG1	UCG2	RG1	RG2	RG3	RG4
Chapter 2			■			
Chapter 3		■		■		
Chapter 4		■				
Chapter 5	■			■		
Chapter 6					■	
Chapter 7						■

Figure 1.9: Mapping of research goals (RG) to chapters in the book

The chapters are organized to follow a logical progression from data to final results. The concluding chapter, Chapter 8, synthesizes the findings and discusses potential avenues for future research that build upon the work presented in this thesis. Figure 1.9 shows which chapters tackle which research goals. Below is a brief overview of each chapter.

Chapter 2 introduces Powershap, a novel feature selection method designed to identify relevant features for ML model development. This method utilizes Shapley values to compare the importance of features in a model against a known random feature for determining feature relevance. This chapter addresses the second research goal (RG1) and demonstrates that Powershap is significantly faster than other wrapper methods while providing comparable or superior feature selection performance. Many subsequent chapters also utilize Powershap, and therefore, this chapter is presented first.

Chapter 3 details the development and validation of an ML model for predicting piperacillin blood plasma concentrations in critically ill patients in the ICU. Piperacillin, a commonly administered antibiotic in the ICU, requires adequate dosing to achieve therapeutic concentrations. The proposed models employ quantile regression in an ensemble to provide uncertainty quantification, resulting in a predictive distribution. These models are compared to other ML models, such as Gaussian processes and Multi-Layer Perceptrons (MLPs), as well as traditional Population Pharmacokinetic models. The uncertainty quantification is evaluated using a proposed novel calibration metric for predictive distributions. Additionally, the models are externally validated on an independent dataset and explained through a patient use case with the use Shapley values. Chapter 4 extends this work by building an improved model using new data and incorporating domain knowledge. This enhanced model predicts concentrations of multiple antibiotics and provides individualized dosing advice. These chapters address UCG2, with Chapter 3 also covering the regression aspect of RG2.

Chapter 5 addresses UCG1 and the classification component of RG2. It presents three ML models for predicting the risk of New-Onset Atrial Fibrillation in ICU patients using Electronic Health Record (EHR) data. This chapter also introduces a method to handle class imbalance in ICU data, where different patient populations are associated with varying lengths of stay, potentially introducing bias or even data leakage. The models are evaluated on their uncertainty quantification performance using proposed modifications of metrics for classification calibration and validated on three different internal and external datasets to assess their generalization capabilities. Shapley values are also used to explain the model predictions for explainability.

Chapter 6 introduces Causalteshap, a method for analyzing whether a feature is predictive or not for treatment effects. This method also leverages Shapley values and compares them to a random feature to determine whether the feature is predictive or not. The chapter presents two rigorous benchmarks: a synthetic and a semi-synthetic benchmark. These two benchmarks are used to evaluate the method's ability to discern

predictive features. This chapter addresses RG3 and demonstrates that Causalteshap can identify predictive features with minimal false positives.

Chapter 7 tackles the final research goal, RG4. It assumes the existence of a dose-response model that predicts outcomes given a treatment dose and requires uncertainty quantification to integrate it into a treatment decision support system. Due to the overlap issue, this chapter proposes using propensity-weighted conformal prediction, an extension of standard conformal prediction, to measure overlap and ensure that prediction intervals or distributions adhere to the required coverage. The method provides infinite intervals when data for a specific treatment is lacking, indicating model interpolation or generalization in those regions. The method is evaluated on a robust synthetic benchmark and a semi-synthetic benchmark using an anesthesiology medication. Compared to other methods, the proposed approach demonstrates superior performance and maintains correct coverage, even in regions with limited data support.

1.7 Publications

As mentioned, the research results obtained during this PhD research have been published in scientific journals and presented at international conferences. The following list provides an overview of the publications during this PhD research:

1.7.1 Publications in International Journals

1. **Jarne Verhaeghe**, Sofie Dhaese, Thomas De Corte, David Vander Mijnsbrugge, Heleen Aardema, Jan G. Zijlstra, Alain Verstraete, Veronique Stove, Pieter Colin, Femke Ongenae, Jan De Waele, and Sofie Van Hoecke, *Development and evaluation of uncertainty quantifying machine learning models to predict piperacillin plasma concentrations in critically ill patients*, Published in BMC Medical Informatics and Decision Making, Volume 22, Issue 1, August 2022.

2. Thomas De Corte, **Jarne Verhaeghe**, Sofie Dhaese, Sarah Van Vooren, Jerina Boelens, Alain Verstraete, Veronique Stove, Femke Ongenae, Liesbet De Bus, Pieter Depuydt, Sofie Van Hoecke, and Jan De Waele, *Pathogen-based target attainment of optimized continuous infusion dosing regimens of piperacillin-tazobactam and meropenem in surgical ICU patients: a prospective single center observational study*, Published in Annals of Intensive Care, Volume 13, Issue 1, April 2023.

3. **Jarne Verhaeghe**, Thomas De Corte, Christopher M. Sauer, Tom Hendriks, Olivier W.M. Thijssens, Femke Ongenae, Paul Elbers, Jan De Waele, and Sofie Van Hoecke, *Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in ICU patients*, Published in International Journal of Medical Informatics, Volume 175, July 2023.

4. Thomas De Corte, **Jarne Verhaeghe**, Femke Ongenae, Sofie Van Hoecke, and Jan De Waele, *Towards artificial intelligence as a decision support tool to combat AMR in the ICU*, Published in ICU Management, Volume 23, Issue 4, 2023.

5. **Jarne Verhaeghe**, Femke Ongenae, and Sofie Van Hoecke, *Causalteshap: Discerning Predictive from Prognostic Features for Treatment Effect Analysis*, Published in International Journal of Machine Learning and Cybernetics, June 2025.

1.7.2 Publications in International Conferences

1. **Jarne Verhaeghe**, Jeroen Van Der Donckt, Femke Ongenae, and Sofie Van Hoecke, *Powershap: A Power-Full Shapley Feature Selection Method*, Published in Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2022, PTI, pages 71–87, March 2023.

2. **Jarne Verhaeghe**, Thomas De Corte, Jan J. De Waele, Femke Ongenae, and Sofie Van Hoecke, *Designing a Pharmacokinetic Machine Learning Model for Optimizing Beta-Lactam Antimicrobial Dosing in Critically Ill Patients*, Published in Proceedings of the 2024 8th International Conference on Medical and Health Informatics (ICMHI '24). Association for Computing Machinery, New York, NY, USA, 311–317. Best Presentation Award.

3. Thomas De Corte, Laura Van Maele, Jona Dietvorst, **Jarne Verhaeghe**, Ann Vandendriessche, N De Neve, Sofie Vanderhaeghen, et al. 2023. *Expectations and Readiness of ICU Physicians to Use AI in Clinical Practice: A Multicentre Survey Study.* In ESICM, 36th Annual Congress, Abstracts. European Society of Intensive Care Medicine (ESICM).

1.7.3 Publications Under Review

1. Jef Jonkers, **Jarne Verhaeghe**, Glen Van Wallendael, Luc Duchateau, and Sofie Van Hoecke, *Conformal Convolution and Monte Carlo Meta-learners for Predictive Inference of Individual Treatment Effects*, Submitted to NeurIPS2025. (**Conference**)

2. **Jarne Verhaeghe**, Jef Jonkers, and Sofie Van Hoecke, *Conformal Prediction for Dose-Response Models with Continuous Treatments*, Submitted to International Journal of Approximate Reasoning. (**Journal**)

3. Thomas De Corte, **Jarne Verhaeghe**, Kaat Van der Eycken, Femke Ongenae, Jan De Waele, and Sofie Van Hoecke, *Towards trustful machine learning for antimicrobial therapy using an explainable artificial intelligence dashboard*, Submitted to Journal of Evaluation in Clinical Practice. (**Journal**)

4. De Corte T, Van Maele L, Dietvorst J, **Verhaeghe Jarne**, Vandendriessche A, De Neve N, Vanderhaeghen S, Dumoulin A, Temmerman W, Dewulf B, Van Regenmortel N, Debaveye Y, Ongenae F, Van Hoecke S, De Waele JJ. Exploring Preconditions for the Implementation of Artificial Intelligence-based Clinical Decision Support Systems in the Intensive Care Unit – a Multicentric Mixed Methods Study. Submitted to Intensive Care Medicine Experimental. (**Journal**)

1.7.4 Code Repositories

Most of the results and models created in the works presented in this dissertation are available online in open-source code repositories:

- Chapter 2: Powershap: A power-full Shapley feature selection method, Software library, 206 Github stars, 1M+ downloads, https://github.com/predict-idlab/powershap
- Chapter 3: Piperacillin Plasma Concentrations models and evaluations, urlhttps://github.com/predict-idlab/REACT
- Chapter 5: Atrial Fibrillation models and evaluations https://github.com/predict-idlab/atrial_fibrillation_prediction
- Chapter 6: Causalteshap: Discerning Predictive from Prognostic Features for Treatment Effect Analysis, Software library, https://github.com/predict-idlab/causalteshap
- Chapter 7: Dose Response Conformal Prediction, Software library, https://github.com/predict-idlab/dose-response-conformal-prediction

References

[1] Nadia K. Byrnes and John E. Hayes. *Personality Factors Predict Spicy Food Liking and Intake.* Food Quality and Preference, 28(1):213–221, April 2013.

[2] Nitchara Toontom, Mutita Meenune, Luis Kluwe Aguiar, and Wilatsana Posri. *Exploring Hotness and Pungent Odour Thresholds among Three Groups of Thai Chilli Users.* International Journal of Food Science and Technology, 59(11):8473–8489, November 2024.

[3] Valeeratana K. Sinsawasdi, Holger Y. Toschka, and Nithiya Rattanapanone. *Eating Pleasure of Thai Meal.* In The Science of Thai Cuisine. CRC Press, 2022.

[4] *Healthcare.* https://dictionary.cambridge.org/dictionary/english/healthcare, February 2025.

[5] Shuroug A. Alowais, Sahar S. Alghamdi, Nada Alsuhebany, Tariq Alqahtani, Abdulrahman I. Alshaya, Sumaya N. Almohareb, Atheer Aldairem, Mohammed Alrashed, Khalid Bin Saleh, Hisham A. Badreldin, Majed S. Al Yami, Shmeylan Al Harbi, and Abdulkareem M. Albekairy. *Revolutionizing healthcare: the role of artificial intelligence in clinical practice.* BMC Medical Education, 23(1), September 2023.

[6] Fiona E Kelly, Kevin Fong, Nicholas Hirsch, and Jerry P Nolan. *Intensive care medicine is 60 years old: the history and future of the intensive care unit.* Clinical Medicine, 14(4):376–379, August 2014.

[7] Zhi Mao, Chao Liu, Qinglin Li, Yating Cui, and Feihu Zhou. *Intelligent Intensive Care Unit: Current and Future Trends.* Intensive Care Research, 3(2):182–188, May 2023.

[8] Ivor S. Douglas, Anuj Mehta, and Jason Mansoori. *Policy Proposals for Mitigating Intensive Care Unit Strain: Insights from the COVID-19 Pandemic.* Annals of the American Thoracic Society, 21(12):1633–1642, December 2024.

[9] Michael R. Pinsky, Armando Bedoya, Azra Bihorac, Leo Celi, Matthew Churpek, Nicoleta J. Economou-Zavlanos, Paul Elbers, Suchi Saria, Vincent Liu, Patrick G. Lyons, Benjamin Shickel, Patrick Toral, David Tscholl, and Gilles Clermont. *Use of artificial intelligence in critical care: opportunities and obstacles.* Critical Care, 28(1), April 2024.

[10] Juliane Winkelmann, Dimitra Panteli, Elke Berger, and Reinhard Busse. *HAVE WE LEARNT THE RIGHT LESSONS? INTENSIVE CARE CAPACITIES DURING THE COVID-19 PANDEMIC IN EUROPE.* Eurohealth, 28, 2022.

[11] *Big Data Makes Intensive Care Better.* https://www.snf.ch/en/yXlNpBZ3PZKcJLyt/news/news-191202-press-release-big-data-makes-intensive-care-better.

[12] Vinay Suresh, Kaushal K Singh, Esha Vaish, Mohan Gurjar, Anubuvanan AM, Yashita Khulbe, and Syed Muzaffar. *Artificial Intelligence in the Intensive Care Unit: Current Evidence on an Inevitable Future Tool.* Cureus, May 2024.

[13] Emilie Steerling, Elin Siira, Per Nilsen, Petra Svedberg, and Jens Nygren. *Implementing AI in healthcare—the relevance of trust: a scoping review.* Frontiers in Health Services, 3, August 2023.

[14] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. *Causal machine learning for predicting treatment outcomes.* Nature Medicine, 30(4):958–968, April 2024. Publisher: Nature Publishing Group.

[15] Benjamin Lambert, Florence Forbes, Senan Doyle, Harmonie Dehaene, and Michel Dojat. *Trustworthy Clinical AI Solutions: A Unified Review of Uncertainty Quantification in Deep Learning Models for Medical Image Analysis.* Artificial Intelligence in Medicine, 150:102830, April 2024.

[16] Fahimeh Fakour, Ali Mosleh, and Ramin Ramezani. *A Structured Review of Literature on Uncertainty in Machine Learning & Deep Learning*, June 2024.

[17] Davy van de Sande, Eline Fung Fen Chung, Jacobien Oosterhoff, Jasper van Bommel, Diederik Gommers, and Michel E. van Genderen. *To Warrant Clinical Adoption AI Models Require a Multi-Faceted Implementation Evaluation.* npj Digital Medicine, 7(1):1–5, March 2024.

[18] Tine De Corte, Lore Van Maele, Jelle Dietvorst, Joke Verhaeghe, Annelies Vandendriessche, Nathalie De Neve, Stijn Vanderhaeghen, Annemie Dumoulin, Wim Temmerman, Bram Dewulf, Nele Van Regenmortel, Yves Debaveye, Frederik Ongenae, Sofie Van Hoecke, and Jan J. De Waele. *Exploring Preconditions for the Implementation of Artificial Intelligence-based Clinical Decision Support Systems in the Intensive Care Unit – a Multicentric Mixed Methods Study.* Submitted to Intensive Care Medicine Experimental, 2025. Under Review.

[19] Pei Wang. *On Defining Artificial Intelligence.* Journal of Artificial General Intelligence, 10(2):1–37, January 2019.

[20] Kevin Gurney. *An Introduction to Neural Networks.* Taylor and Francis, Hoboken, 2014.

[21] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. *Multilayer feedforward networks are universal approximators.* Neural Networks, 2(5):359–366, January 1989.

[22] Tjeerd van der Ploeg, Peter C. Austin, and Ewout W. Steyerberg. *Modern Modelling Techniques Are Data Hungry: A Simulation Study for Predicting Dichotomous Endpoints*. BMC Medical Research Methodology, 14(1):137, December 2014.

[23] Scott M Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 4765–4774. Curran Associates, Inc., 2017.

[24] Eric Schulz, Maarten Speekenbrink, and Andreas Krause. *A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions*. Journal of Mathematical Psychology, 85:1–16, August 2018.

[25] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. *CatBoost: Gradient Boosting with Categorical Features Support*. page 7.

[26] Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and De-Chuan Zhan. *A Closer Look at Deep Learning Methods on Tabular Datasets*, January 2025.

[27] Jeroen Van Der Donckt, Jonas Van Der Donckt, Emiel Deprost, Nicolas Vandenbussche, Michael Rademaker, Gilles Vandewiele, and Sofie Van Hoecke. *Do Not Sleep on Traditional Machine Learning: Simple and Interpretable Techniques Are Competitive to Deep Learning for Sleep Scoring*. Biomedical Signal Processing and Control, 81:104429, March 2023.

[28] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. *From local explanations to global understanding with explainable AI for trees*. Nature Machine Intelligence, 2(1):56–67, January 2020.

[29] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. *Explainable AI: A Review of Machine Learning Interpretability Methods*. Entropy, 23(1):18, December 2020.

[30] Jonas Van Der Donckt, Jeroen Van Der Donckt, Emiel Deprost, and Sofie Van Hoecke. *Tsflex: Flexible Time Series Processing & Feature Extraction*. SoftwareX, 17:100971, January 2022.

[31] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P. Trevino, Jiliang Tang, and Huan Liu. *Feature Selection: A Data Perspective*. ACM Computing Surveys, 50(6):94:1–94:45, December 2017.

[32] Savina Colaco, Sujit Kumar, Amrita Tamang, and Vinai George Biju. *A Review on Feature Selection Algorithms*. In Emerging Research in Computing, Information, Communication and Applications, pages 133–153, Singapore, 2019. Springer.

[33] Peter Spirtes. *Introduction to Causal Inference.* J. Mach. Learn. Res., 11:1643–1662, August 2010.

[34] Atul Rawal, Adrienne Raglin, Danda B. Rawat, Brian M. Sadler, and James McCoy. *Causality for Trustworthy Artificial Intelligence: Status, Challenges and Perspectives.* ACM Comput. Surv., 57(6):146:1–146:30, February 2025.

[35] Benjamin Speich, Belinda von Niederhäusern, Nadine Schur, Lars G. Hemkens, Thomas Fürst, Neera Bhatnagar, Reem Alturki, Arnav Agarwal, Benjamin Kasenda, Christiane Pauli-Magnus, Matthias Schwenkglenks, and Matthias Briel. *Systematic review on costs and resource use of randomized clinical trials shows a lack of transparent and comprehensive data.* Journal of Clinical Epidemiology, 96:1–11, April 2018.

[36] Caroline M. A. van den Bosch, Marlies E. J. L. Hulscher, Reinier P. Akkermans, Jan Wille, Suzanne E. Geerlings, and Jan M. Prins. *Appropriate antibiotic use reduces length of hospital stay.* Journal of Antimicrobial Chemotherapy, page dkw469, December 2016.

[37] Shagufta Vora. *Acute Renal Failure Due to Vancomycin Toxicity in the Setting of Unmonitored Vancomycin Infusion.* Baylor University Medical Center Proceedings, 29(4):412–413, October 2016.

[38] Martin Huber. *An Introduction to Causal Discovery.* Swiss Journal of Economics and Statistics, 160(1):14, October 2024.

[39] Jennie E. Brand, Xiang Zhou, and Yu Xie. *Recent Developments in Causal Inference and Machine Learning.* Annual Review of Sociology, 49(1):81–110, July 2023.

[40] Donald B Rubin. *Causal Inference Using Potential Outcomes.* Journal of the American Statistical Association, 100(469):322–331, March 2005.

[41] Peter H. Egger and Maximilian von Ehrlich. *Generalized Propensity Scores for Multiple Continuous Treatment Variables.* Economics Letters, 119(1):32–34, April 2013.

[42] Lu Tian, Ash A Alizadeh, Andrew J Gentles, and Robert Tibshirani. *A Simple Method for Estimating Interactions between a Treatment and a Large Number of Covariates.* Journal of the American Statistical Association, 109(508):1517–1532, October 2014.

[43] Hyung Park, Eva Petkova, Thaddeus Tarpey, and R Todd Ogden. *A Sparse Additive Model for Treatment Effect-Modifier Selection.* Biostatistics, 23(2):412–429, April 2022.

[44] Armen Der Kiureghian and Ove Ditlevsen. *Aleatory or epistemic? Does it matter?* Structural Safety, 31(2):105–112, March 2009.

[45] Eyke Hüllermeier and Willem Waegeman. *Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods*. Machine Learning, 110(3):457–506, March 2021.

[46] Matias Valdenegro-Toro and Daniel Saromo Mori. *A Deeper Look into Aleatoric and Epistemic Uncertainty Disentanglement*. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), page 1508–1516. IEEE, June 2022.

[47] Christopher R. S. Banerji, Tapabrata Chakraborti, Chris Harbron, and Ben D. MacArthur. *Clinical AI tools must convey predictive uncertainty for each individual patient*. Nature Medicine, 29(12):2996–2998, October 2023.

[48] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. *The Need for Uncertainty Quantification in Machine-Assisted Medical Decision Making*. Nature Machine Intelligence, 1(1):20–23, January 2019.

[49] Benjamin Kompa, Jasper Snoek, and Andrew L. Beam. *Second Opinion Needed: Communicating Uncertainty in Medical Machine Learning*. npj Digital Medicine, 4(1):1–6, January 2021.

[50] Silvia Seoni, Vicnesh Jahmunah, Massimo Salvi, Prabal Datta Barua, Filippo Molinari, and U. Rajendra Acharya. *Application of Uncertainty Quantification to Artificial Intelligence in Healthcare: A Review of Last Decade (2013–2023)*. Computers in Biology and Medicine, 165:107441, October 2023.

[51] Yaniv Romano, Evan Patterson, and Emmanuel J. Candès. *Conformalized Quantile Regression*. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, number 318, pages 3543–3553. Curran Associates Inc., Red Hook, NY, USA, December 2019.

[52] Ethan Goan and Clinton Fookes. *Bayesian Neural Networks: An Introduction and Survey*, page 45–87. Springer International Publishing, 2020.

[53] Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. *Nonparametric predictive distributions based on conformal prediction*. Machine Learning, 108(3):445–474, March 2019.

[54] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On calibration of modern neural networks*. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1321–1330. JMLR.org, 2017.

[55] John C. Platt. *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*. In Advances in Large Margin Classifiers, pages 61–74. MIT Press, 1999.

[56] W. B. Wu. *Isotonic regression: Another look at the changepoint problem*. Biometrika, 88(3):793–804, October 2001.

[57] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. *Large-scale probabilistic predictors with and without guarantees of validity*. In Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 892–900, Cambridge, MA, USA, 2015. MIT Press.

[58] Vladimir Vovk, Ivan Petej, Paolo Toccaceli, Alexander Gammerman, Ernst Ahlberg, and Lars Carlsson. *Conformal calibrators*. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov, and Giovanni Cherubin, editors, Proceedings of the Ninth Symposium on Conformal and Probabilistic Prediction and Applications, volume 128 of *Proceedings of Machine Learning Research*, pages 84–99. PMLR, 09–11 Sep 2020.

[59] Soumya Ghosh, Q. Vera Liao, Karthikeyan Natesan Ramamurthy, Jiri Navratil, Prasanna Sattigeri, Kush R. Varshney, and Yunfeng Zhang. *Uncertainty Quantification 360: A Holistic Toolkit for Quantifying and Communicating the Uncertainty of AI*, June 2021.

[60] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. *High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach*. In International Conference on Machine Learning, pages 4075–4084. PMLR, July 2018.

[61] F. N. David and N. L. Johnson. *The Probability Integral Transformation When Parameters Are Estimated from the Sample*. Biometrika, 35(1/2):182–190, 1948.

[62] James E. Matheson and Robert L. Winkler. *Scoring Rules for Continuous Probability Distributions*. Management Science, 22(10):1087–1096, June 1976.

[63] Takuo Yoshida, Tomoko Fujii, Shigehiko Uchino, and Masanori Takinami. *Epidemiology, prevention, and treatment of new-onset atrial fibrillation in critically ill: a systematic review*. Journal of Intensive Care, 3(1), April 2015. ZSCC: 0000062.

[64] Andrew Lever and Iain Mackenzie. *Sepsis: Definition, Epidemiology, and Diagnosis*. BMJ : British Medical Journal, 335(7625):879, October 2007.

[65] Thomas De Corte, Jarne Verhaeghe, Sofie Dhaese, Sarah Van Vooren, Jerina Boelens, Alain G. Verstraete, Veronique Stove, Femke Ongenae, Liesbet De Bus, Pieter Depuydt, Sofie Van Hoecke, and Jan J. De Waele. *Pathogen-based target attainment of optimized continuous infusion dosing regimens of piperacillin-tazobactam and meropenem in surgical ICU patients: a prospective single center observational study*. Annals of Intensive Care, 13(1), April 2023.

2

Powershap: A Power-Full Shapley Feature Selection Method

In many machine learning problems, not only those involving models for the ICU, feature selection is a crucial step in the model development process. This step often consumes significant time, making any efficiency improvements highly beneficial. Given the high-dimensional nature of many ICU-related datasets, effective feature selection is of vital importance in terms of efficiency, costs, speed, interpretability, robustness, and performance. The proposed solution, Powershap, is based on the assumption that irrelevant features should have an equal or lower average impact on predictions compared to known random features and thereby tackles RG1. This hypothesis is tested using Shapley values, with comparisons made across multiple training cycles. To further facilitate model development, Powershap offers an automatic mode that utilizes a heuristic that determines the optimal number of iterations using statistical power calculations. The method's performance and speed are validated through various benchmarks, demonstrating that Powershap is competitive while significantly reducing computational time. My contribution to this chapter is the complete design and implementation of Powershap, designing and performing all the experiments, and writing the chapter.

Powershap: A Power-full Shapley Feature Selection Method

Jarne Verhaeghe, Jeroen Van Der Donckt, Femke Ongenae, Sofie Van Hoecke

Published in Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2022

Abstract Feature selection is a crucial step in developing robust and powerful machine learning models. Feature selection techniques can be divided into two categories: filter and wrapper methods. While wrapper methods commonly result in strong predictive performances, they suffer from a large computational complexity and therefore take a significant amount of time to complete, especially when dealing with high-dimensional feature sets. Alternatively, filter methods are considerably faster, but suffer from several other disadvantages, such as (i) requiring a threshold value, (ii) many filter methods not taking into account intercorrelation between features, and (iii) ignoring feature interactions with the model. To this end, we present *powershap*, a novel wrapper feature selection method, which leverages statistical hypothesis testing and power calculations in combination with Shapley values for quick and intuitive feature selection. *Powershap* is built on the core assumption that an informative feature will have a larger impact on the prediction compared to a known random feature. Benchmarks and simulations show that *powershap* outperforms other filter methods with predictive performances on par with wrapper methods while being significantly faster, often even reaching half or a third of the execution time. As such, *powershap* provides a competitive and quick algorithm that can be used by various models in different domains. Furthermore, *powershap* is implemented as a plug-and-play and open-source *sklearn* component, enabling easy integration in conventional data science pipelines. User experience is even further enhanced by also providing an automatic mode that automatically tunes the hyper-parameters of the *powershap* algorithm, allowing to use the algorithm without any configuration needed.

2.1 Introduction

In many data mining and machine learning problems, the goal is to extract and discover knowledge from data. One of the challenges frequently faced in these problems is the high dimensionality and the unknown relevance of features [1]. Ignoring these challenges will more than often result in modeling obstacles, such as sparse data, overfitting, and the curse of dimensionality. Therefore, feature selection is frequently applied, among other techniques, to effectively reduce the feature dimensionality. The smaller subset of features has the potential to explain the problem better,

reduce overfitting, alleviate the curse of dimensionality, and even facilitate interpretation. Furthermore, feature selection is known to increase model performance, increase computational efficiency, and increase the robustness of many models due to the dimensionality reduction [1].

In this work, we present a novel feature selection method, called *powershap*, that is a faster and easy-to-use wrapper method. The feature selection is realized by using Shapley values, statistical tests, and power calculations.

First, in Section 2.2, a short overview of the related work is given to show how *powershap* improves upon all these methods. Subsequently, in Section 2.3, the method and the design choices are explained as well as the resulting algorithm. Finally, the performance of *powershap* is compared to other state-of-the-art methods in Section 2.4 and 2.5 using both simulation and open-source benchmark datasets and the results are discussed in Section 2.6. Finally, the conclusions are summarized in Section 2.7.

2.2 Related Work

Feature selection approaches can be categorized into filter and wrapper methods. Filter methods select features by measuring the relevance of the feature using model-agnostic measures, such as statistical tests, information gain, distance, similarity, and consistency to the dependent variable (if available). These methods are model independent as this category of feature selection does not rely on training machine learning models [2], resulting in a fast evaluation. However, the disadvantages of filter methods are that they frequently impose assumptions on the data, are limited to a single type of prediction, such as classification or regression, not all methods take inter-correlation between features into account, and often require a cut-off value or hyperparameter tuning [3]. Examples of these filter methods are rank, chi^2 test, f-test, correlation-based feature selection, Markov blanket filter, and Pearson correlation [2].

Wrapper methods measure the relevance of features using a specific evaluation procedure through training supervised models. Depending on the wrapper technique, models are trained on either subsets of the features or on the complete feature set. The trained models are then utilized to select the resulting feature subset, using the aforementioned performance metrics, or by ranking the inferred feature importances. In general, wrapper methods tend to provide smaller and more qualitative feature subsets than filter methods, as they take the interaction between the features, and between the model and the features, into account [4]. A major drawback of wrapper methods is the considerable time complexity associated with the underlying search algorithm, or in the case of feature importance ranking the hyperparameter tuning. Examples of wrapper methods are forward, backward, genetic, or rank-based feature importance feature selection.

In the interpretable machine learning field, one of the emerging and proven techniques to explain model predictions is SHAP [5]. This technique aims at quantifying the im-

pact of features on the output. To do so, SHAP uses a game-theory inspired additive feature-attribution method based on Shapley Regression Values [5]. This method is model-agnostic and implemented for various models, e.g., linear, kernel-based, deep learning, and tree-based models. Although SHAP suffers from shortcomings, such as its TreeExplainer providing non-zero Shapley values to noise features, it is technically strong and very popular [6].

The strength of the SHAP algorithm facilitates the development of new feature selection methods using Shapley values. A simple implementation would be a rank-based feature selection, which ranks the different features based on their Shapley values on which a rank cut-off value determines the final feature set. However, there are more advanced methods available. One of these more advanced techniques is borutashap [7]. Borutashap is based on the Boruta algorithm that makes use of shadow features, i.e. features with randomly shuffled values. Boruta is built on the idea that a feature is only useful if it is doing better than the best-performing shuffled feature. To do so, Boruta compares the feature importance of the best shadow feature to all other features, selecting only features with larger feature importance than the highest shadow feature importance. Statistical interpretation is realized by repeating this algorithm for several iterations, resulting in a binomial distribution which can be used for p-value cut-off selection [8]. Borutashap improves on the underlying Boruta algorithm by using Shapley values and an optimized version of the shap TreeExplainer [7]. As such, implementations of Borutashap are limited to tree-based models only.

Another shap-based feature selection method using statistics is shapicant [9]. This feature selection method is inspired by the permutation-importance method, which first trains a model on the true dataset, and afterward, it shuffles the labels and retrains the model on the shuffled dataset. This process is repeated for a set amount of iterations, from which the average feature importances of both models are compared. If for a specific feature, the feature importance of the true dataset model is consistently larger than the importance of the shuffled dataset model, that feature is considered informative. Using a non-parametric estimation it is possible to assign a p-value to determine a wanted cut-off value [10]. Shapicant improves on this underlying algorithm by using Shapley values. Specifically, it uses both the mean of the negative and positive Shapley values instead of Gini importances, which are only positive and frequently used for tree-based model importances. Furthermore, shapicant uses out-of-sample feature importances for more accurate estimations and an improved non-parametric estimation formula [9].

Powershap draws inspiration from the non-parametric estimation of shapicant and the random feature usage in borutashap and improves upon all these state-of-the-art filter and wrapper algorithms resulting in at least comparable performances while being significantly faster.

2.3 Powershap

Powershap builds upon the idea that a known random feature should have, on average, a lower impact on the predictions than an informative feature. To realize feature selection, the *powershap* algorithm consists of two components: the *Explain* component and the core *powershap* component. First, in the *Explain* part, multiple models are trained using different random seeds, on different subsets of the data. Each of these subsets is comprised of all the original features together with one random feature. Once the models are trained, the average impact of the features (including the random feature) is explained using Shapley values on an out-of-sample dataset. Then, in the core *powershap* component, the impacts of the original features are statistically compared to the random feature, enabling the selection of all informative features.

2.3.1 Powershap Algorithm

In the *Explain* component, a single known random uniform (RandomUniform) feature is added to the feature set for training a machine learning model. Unlike the Boruta algorithm, where all features are duplicated and shuffled, only a single random feature is added. In some models, such as neural networks, duplicating the complete feature set increases the scale and thereby increases the time complexity drastically. Using the Shapley values on an out-of-sample subset of the data allows for quantifying the impact on the output for each feature. The Shapley values are evaluated on unseen data to assess the true unbiased impact [11]. As a final step, the absolute value of all the Shapley values is taken and then averaged (μ) to get the total average impact of each feature. Compared to shapicant, only a single mean value is used here, resulting in easier statistical comparisons. Furthermore, by utilizing the absolute Shapley values, the positive values and the negative values are added to the total impact, which could result in a different distribution compared to the Gini importance. This procedure is then repeated for I iterations, where every iteration retrains the model with a different random feature and uses a different subset of the data to quantify the Shapley values, resulting in an empirical distribution of average impacts that will further be used for the statistical comparison. In the codebase, the procedure explained above is referred to as the *Explain* function. The pseudocode of the *Explain* function is shown in Algorithm 1.

Given the average impact of each feature for each iteration, it is then possible to compare it to the impact of the random feature in the core powershap component. This comparison is quantified using the percentile formula shown in Equation 2.1 where $\mathbf{s}$ depicts an array of average Shapley values for a single feature with the same length as the number of iterations, while x represents a single value, and $\mathbb{I}$ represents the indicator function. This formula calculates the fraction of iterations where x was higher than the average shap-value of that iteration and can therefore be interpreted

Algorithm 1: Powershap Explain algorithm

Function *Explain(*$I \leftarrow$ *Iterations,* $M \leftarrow$ *Model,* $\mathbf{D}^{n \times m} \leftarrow$ *Data,* $rs \leftarrow$ *Random seed)*

- $\mathbf{powershap}_{values} \leftarrow$size $[I, m+1]$
- **for** $i \leftarrow 1, 2, \ldots, I$ **do**
  - $RS \leftarrow i + rs$
  - $D^{n}_{random} \leftarrow \text{RandomUniform}(RS) \in [-1, 1]$ size n
  - $\mathbf{D}^{n \times m+1} \leftarrow \mathbf{D}^{n \times m} \cup D^{n}_{random}$
  - $\mathbf{D}^{0.8n \times m+1}_{train}, \mathbf{D}^{0.2n \times m+1}_{val} \leftarrow$ split $\mathbf{D}$
  - $M \leftarrow \text{Fit } M(\mathbf{D}_{train})$
  - $\mathbf{S}_{values} \leftarrow \text{SHAP}(M, \mathbf{D}_{val})$
  - $\mathbf{S}_{values} \leftarrow |\mathbf{S}_{values}|$
  - **for** $j \leftarrow 1, 2, \ldots, m+1$ **do**
    - $\mathbf{powershap}_{values}[i][j] \leftarrow \mu(\mathbf{S}_{values}[\ldots][j])$
- **return** $\mathbf{powershap}_{values}$

as the p-value.

$$Percentile(\mathbf{s}, x) = \sum_{i}^{n} \frac{\mathbb{I}(x > s_i)}{n} \tag{2.1}$$

Note that this formula provides smaller p-values than what should be observed, the correct empirical formula is $(1 + \sum_i^n \mathbb{I}(x > s_i))/(n+1)$ as explained by North et al. [12]. This issue of smaller p-values mainly persists for lower number of iterations. However, *powershap* implements Equation 2.1 as this anticonservative estimation of the p-value is desired behavior for the automatic mode (see Section 2.3.2). This formula enables setting a static cut-off value for the p-value instead of a varying cut-off value and results in fewer required iterations, while still providing correct results. This will be further explained at the end of Section 2.3.2.

As the hypothesis states that the impact of the random feature should be on average lower than any informative feature, all impacts of the random feature are again averaged, resulting in a single value that can be used in the percentile function. This results in a p-value for every original feature. This p-value represents the fraction of cases where the feature is less important, on average than a random feature. Given the hypothesis and these p-value calculations, a heuristic implementation of a one-sample one-tailed student-t smaller statistic test can be done, where the null hypothesis states that the random feature (H_1-distribution) is not more important than the tested feature (H_0-distribution) [13]. Therefore, the positive class in this statistical test represents a true null hypothesis. This heuristic implementation does not assume a distribution on the tested feature impact scores, in contrast to a standard student-t statistic test where a standard Gaussian distribution is assumed. Then, given a threshold p-value α, it is possible to find and output the set of informative features. The

pseudocode of Algorithm 2 details how the core *powershap* feature selection method is realized.

Algorithm 2: Powershap core algorithm

Function *Powershap (*$I \leftarrow$ *Iterations,* $M \leftarrow$ *Model,* $\mathbf{F}_{set} \leftarrow F_1, \ldots, F_m$, $\mathbf{D} \leftarrow$ *Data size* $[n, m]$, $\alpha \leftarrow$ *required p-value)*

- $\textbf{powershap}_{values} \leftarrow$**Explain**$(I, M, \mathbf{D})$
- $S_{random} \leftarrow \mu(\textbf{powershap}_{values}[...][m+1])$
- $\mathbf{P}^m \leftarrow$ initialize
- **for** $j \leftarrow 1, 2, \ldots, m$ **do**
  - $\mathbf{P}[j] \leftarrow Percentile(\textbf{powershap}_{values}[...][j], S_{random})$
- **return** $\{F_i \mid \forall\, i : \mathbf{P}[i] < \alpha\}$

2.3.2 Automatic Mode

Running the *powershap* algorithm consisting of the *explain* and the *core* components, requires setting two hyperparameters: α the p-value threshold and I the number of iterations. When hyperparameter tuning, one should make a trade-off between runtime and quality. On the one hand, there should be enough iterations to avoid false negatives for a given α, especially with the anticonservative p-values. On the other hand, adding iterations increases the time complexity. To avoid the need for users to manually optimize these two hyperparameters, *powershap* also has an automatic mode. This automatic mode, automatically determines and optimizes the iteration hyperparameter I using statistical power calculation for α, hence the name *powershap*.
The statistical power of a test is $1 - \beta$, where β is the probability of false negatives. In this case, a false negative is a non-informative feature flagged as an informative one. If a statistical test of a tested sample outputs a p-value α, this represents the chance that the tested sample could be flagged as *significant* by chance given the current data. This is calculated using Equation 2.2. If the data in the statistical test is small, it is possible to have a very low α but a large β, resulting in an output that cannot be trusted. Therefore, for a given α, the associated power should be as close to 1 as possible to avoid any false negatives. The power of a statistical test can be calculated using the cumulative distribution function F of the underlying tested distribution H_1 using Equation 2.3. Figure 2.1 explains this visually. In the current context, H_0 could represent the random feature impact distribution and H_1 the tested feature impact distribution.

$$\alpha(x) = F_{H_0}(x) \tag{2.2}$$

$$Power(\alpha) = F_{H_1}\left(F_{H_0}^{-1}(\alpha)\right) \tag{2.3}$$

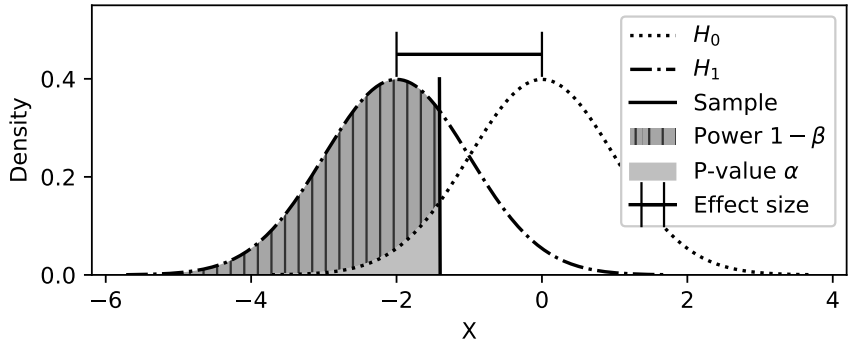


Figure 2.1: Visualization of p-value, effect size, and power for a standard t-test.

The power calculations require the cumulative distribution function F. However, the underlying distributions of the calculated feature impacts are unknown. In addition, calculating F heuristically does not enable calculating the required iteration hyperparameter, which is the goal of the automatic mode. *Powershap* circumvents this by mapping the underlying distributions to two standard student-t distributions as visualized in Figure 2.1. It first calculates the pooled standard deviation, using Equation 2.4, by averaging the standard deviations σ of both distributions. It then calculates the distance d between these two distributions, also called the effect size, in terms of this pooled standard deviation using the Cohen's d effect size as detailed in Equation 2.5 [13]. Now, it is possible to define two standard student-t distributions with distance $\sqrt{I} \cdot d$ apart and $I - 1$ degrees of freedom, where I is the amount of *powershap* iterations. The standard central student-t F_{CT} and non-central student-t F_{NCT} cumulative distribution functions are then used to calculate the power of the statistical test according to Equation 2.6. This equation can in turn be used in a heuristic algorithm to solve for I. *Powershap* uses the *solve_power* implementation of *statsmodels* to determine the required I from the TTestPower equation using brentq expansion for a provided required power [14]. The *powershap* pseudocode for the calculation of the effect size, power, and required iterations is shown in Algorithm 3.

$$PooledStd(\mathbf{s}_1, \mathbf{s}_2) = \frac{\sqrt{(\sigma^2(\mathbf{s}_1) + \sigma^2(\mathbf{s}_2))}}{2} \tag{2.4}$$

$$EffectSize(\mathbf{s}_1, \mathbf{s}_2) = \frac{\mu(\mathbf{s}_1) - \mu(\mathbf{s}_2)}{PooledStd(\mathbf{s}_1, \mathbf{s}_2)} \tag{2.5}$$

$$TTestPower(\alpha, I, d_{calc}) = F_{NCT}\left(F_{CT}^{-1}(\alpha, k = I - 1), k = n - 1, d = \sqrt{I} d_{calc}\right) \tag{2.6}$$

With the calculated required amount of iterations n, the automatic *powershap* algorithm can be executed. The pseudocode to enable the automatic mode is shown in

Algorithm 3: Powershap analysis function

Function *Analysis*($\alpha \leftarrow$ *required p-value,* $\beta \leftarrow$ *required power,* **powershap**$_{values}$)
- $\mathbf{S}_{random} \leftarrow$ **powershap**$_{values}[...][m+1]$
- $\mathbf{P} \leftarrow$ size $[m]$
- $\mathbf{N}_{required} \leftarrow$ size $[m]$
- **for** $j \leftarrow 1, 2, \ldots, m$ **do**
  - $\mathbf{S}_i \leftarrow$ **powershap**$_{values}[...][j]$
  - $\mathbf{P}[j] \leftarrow Percentile(\mathbf{S}_i, \mu(S_{random}))$
  - effectsize $\leftarrow$ EffectSize($\mathbf{S}_i$, $\mathbf{S}_{random}$)
  - $\mathbf{N}_{required} \leftarrow$ **SolveTTestPower**(effectsize, α, β)
- **return** $\mathbf{P}$, $\mathbf{N}_{required}$

Algorithm 4. As can be seen, this is an expansion of the core algorithm (see Algorithm 2) and starts with an initial ten iterations to calculate the initial p-value, effect sizes, power, and required iterations for all features. Then, it searches for the largest required number of iterations I_{max} of all tested features having a p-value below the threshold α. If I_{max} exceeds the already performed number of iterations I_{old}, automatic mode continues *powershap* for the extra required iterations. This process is repeated until the performed iterations exceed the required iterations. For optimization, when the extra required iterations ($I_{max} - I_{old}$) exceed ten iterations, the automatic mode first adds ten iterations and then re-evaluates the required iterations because the required iterations are influenced by the already performed iterations. Furthermore, it is also possible to provide a stopping criterion on the re-execution of *powershap* to avoid an infinite calculation. As a result the time complexity of the algorithm is linear in terms of the underlying model and shap explainer and can be formulated as $O(p[M_{n+1} + S(M_{n+1})])$, with n the amount of features, p the number of powershap iterations, S the shap explainer time, and M_x the model fit time for x features. For the automatic mode, by default, α is set to 0.01 while the required power is set to 0.99. This results in only selecting features that are more important than the random feature for all iterations. Furthermore, this also compensates for the anticonservative p-value and avoids as many false negatives as possible. Realizing the same desired behavior with the more accurate p-value estimation would require a varying α of $1/n$, complicating the power calculations and increasing the likelihood of false negatives. The resulting powershap algorithm is implemented in Python as an open-source plug-and-play *sklearn* compatible component to enables direct usage in conventional machine learning pipelines [15]. The codebase [1] already supports a wide variety of models, such as linear, tree-based, and even deep learning models. To

[1]The code, documentation, and more benchmarks can be found using the following link: https://github.com/predict-idlab/PowerSHAP

assure the quality and correctness of the implementation, we tested the functionality using unit testing.

Algorithm 4: Automatic Powershap algorithm version

Function *Powershap (*$M \leftarrow$ *Model,* $\mathbf{F}_{set} \leftarrow F_1, \ldots, F_m$, $\mathbf{D}^{n \times m} \leftarrow$ *Data,* $\alpha \leftarrow$ *required p-value,* $\beta \leftarrow$ *required power)*

- $\mathbf{powershap}_{values} \leftarrow$**Explain**$(I \leftarrow 10, M, \mathbf{D}, rs \leftarrow 0)$
- $\mathbf{P}, \mathbf{N}_{required} \leftarrow$**Analysis**$(\alpha, \beta, \mathbf{powershap}_{values})$
- $I_{max} \leftarrow \text{ceil}(\mathbf{N}_{required}[\text{MaxArg}(\mathbf{P} < \alpha)])$
- $I_{old} \leftarrow 10$
- **while** $I_{max} > I_{old}$ **do**
  - **if** $I_{max} - I_{old} > 10$ **then**
    - $\mathbf{auto}_{values} \leftarrow$**Explain**$(I \leftarrow 10, M, \mathbf{D}, rs \leftarrow 0)$
    - $I_{old} \leftarrow I_{old} + 10$
  - **else**
    - $\mathbf{auto}_{values} \leftarrow$**Explain**$(I \leftarrow I_{max} - I_{old}, M, \mathbf{D}, rs \leftarrow 0)$
    - $I_{old} \leftarrow I_{max}$
  - $\mathbf{powershap}_{values} \leftarrow \mathbf{powershap}_{values} \cup \mathbf{auto}_{values}$
  - $\mathbf{P}, \mathbf{N}_{required} \leftarrow$**Analysis**$(\alpha, \beta, \mathbf{powershap}_{values})$
  - $I_{max} \leftarrow \text{ceil}(Max(\mathbf{N}_{\mathbf{required}}[i, \forall i : \mathbf{P}[i] < \alpha]))$
- **return** $[F_i, \forall\, i : \mathbf{P}[i] < \alpha]$

2.4 Experiments

2.4.1 Feature Selection Methods

To facilitate a comparison with other feature selection techniques, we benchmark *powershap* together with other frequently used techniques on both synthetic and real-world datasets. In particular, *powershap* is compared with both filter and wrapper methods, and state-of-the-art shap-based wrapper methods. To provide a fair comparison, all methods, including *powershap*, were used in their default out-of-the-box mode without tuning. For *powershap*, this default mode is the automatic mode. Concerning filter methods, two methods were chosen: the chi-squared and f-test feature selection from the *sklearn*-library [15]. The chi-squared test measures the dependence between a feature and the classification outcome and assigns a low p-value to features that are not independent of the outcome. As the chi-squared test only works with positive values, the values are shifted in all chi-squared experiments such that all values are positive. This has no effect on tree-estimators as they are invariant to data scaling [13]. The F-test in *sklearn* is a univariate test that calculates the F-score and p-values on the predictions of a univariate fitted linear regressor with the target [15]. Both filter methods

provide p-values that are set to the same threshold as *powershap*. As wrapper feature selection method, forward feature selection was chosen. This method is a greedy algorithm that starts with an empty set of features and trains a model with each feature separately. In every iteration, forward feature selection then adds the best feature according to a specified metric, often evaluated in cross-validation, until the metric stops improving. This is generally considered a strong method but has a very large time complexity [2]. *Powershap* is also compared to shapicant [9] and borutashap [7], two SHAP-based feature selection methods. The default machine learning model used for all datasets and all feature selection methods, including *powershap*, is a CatBoost gradient boosting tree-based estimator using 250 estimators with the overfitting detector enabled. For classification, the CatBoost model uses adjusted class weights to compensate for any potential class imbalance. The Catboost estimator often results in strong predictive performances out-of-the-box, without any hyper-parameter tuning, making it the perfect candidate for benchmarking and comparison [16]. All experiments are performed on a laptop with a Intel(R) Core(TM) i7-9850H CPU at 2.60GHz processor and 16 GB RAM running at 2667 MHz, with background processes to a minimum.

2.4.2 Simulation Dataset

The methods are first tested on a simulated dataset to assess their ability to discern noise features from informative features. The used simulation dataset is created using the `make_classification` function of *sklearn*. This function creates a classification dataset, however, exactly the same can be done for obtaining a regression dataset (by using `make_regression`). The simulations are run using 20, 100, 250, and 500 total features to understand the performance on varying dimensions of feature sets. The ratio of informative features is varied as 10%, 33%, 50%, and 90% of the total feature set, allowing for assessing the quality of the selected features in terms of this ratio. The resulting simulation datasets each contain 5000 samples. Each simulation experiment was repeated five times with different random seeds. The number of redundant features, which are linear combinations of informative features, and the number of duplicate features were set to zero. Redundant features and duplicate features reduce the performance of models, but they cannot be discerned from true informative features as they are inherently informative. Therefore they are not included in the simulation dataset as the goal of *powershap* is to find informative features. The *powershap* method is compared to shapicant, chi^2, borutashap, and the f-test for feature selection on this simulation dataset. Due to time complexity constraints, forward feature selection was not included in the simulation benchmarking.

Table 2.1: Properties of all datasets

Dataset	Type	Source	# features	train size	test size
Madelon	Classification	OpenML	500	1950	650
Gina priori	Classification	OpenML	784	2601	867
Scene	Classification	OpenML	294	1805	867
CT location	Regression	UCI	384	41347	12153
Appliances	Regression	UCI	30	14801	4934

2.4.3 Benchmark Datasets

In addition to the simulation benchmark, the different methods are also evaluated on five publicly available datasets, i.e. three classification datasets: the Madelon [17], the Gina priori [18], and the Scene dataset [19], and two regression datasets: CT location [20] and Appliances [20]. The details of these datasets are shown in Table 2.1. The Scene dataset is a multi-label dataset, however, a multi-label problem can always be reduced to a one-vs-all classification problem. Therefore only the label "Urban" was chosen here to assess binary classification performance.
Almost all of these datasets have a large feature set, ideal for benchmarking feature selection methods. The datasets are split into a training and test set using a $75/25$ split. All methods are evaluated using both 10-fold cross-validation on the training set and 1000 bootstraps on the test set to assess the robustness of the performance. The test set is utilized to assess generalization beyond the validation set as wrapper methods tend to slightly overfit their validation set [2], while the training set is used for feature selection. The forward feature selection method was performed with 5-fold cross-validation and not 10-fold cross-validation due to the high time complexity. A validation set of 20% of the training set is used for shapicant, using the same validation size as *powershap* in Algorithm 1. The models are evaluated with the AUC metric for classification datasets and with the R^2 metric for regression datasets.

2.5 Results

2.5.1 Simulation Dataset

The results of the simulation benchmarking are shown in Figure 2.2. Each row of subfigures shows the duration, the percentage of informative features found, and the number of selected noise features. These measures are shown for each feature selection method for varying feature set dimensions and varying amounts of informative features. As can be seen, the shapicant method is the slowest wrapper method while *powershap* is, without doubt, the fastest wrapper method. The filter methods

Table 2.2: Benchmarks results for duration and selected features. "default" indicates no feature selection or all features.

	duration (s)						
Dataset	powershap	borutashap	shapicant	forward	chi²	f test	default
Madelon	132s	186s	632s	10483s	$< 1s$	$< 1s$	N/A
Gina priori	184s	299s	812s	68845s	$< 1s$	$< 1s$	N/A
Scene	115s	220s	749s	12496s	$< 1s$	$< 1s$	N/A
CT location	459s	543s	1553s	56879s	N/A	$< 1s$	N/A
Appliances	34s	48s	134s	1913s	N/A	$< 1s$	N/A
	selected features						
Madelon	22	10	30	8	43	18	500
Gina priori	105	37	106	26	328	405	784
Scene	36	14	56	15	93	220	294
CT location	123	162	74	75	N/A	350	384
Appliances	24	24	10	13	N/A	20	30

are substantially faster than any of the wrapper methods, as they do not train models. Furthermore, *powershap* finds all informative features with a limited amount of outputted noise features up to the case with 250 total features with 50% (125) informative features, outperforming every other method. This can be explained by the model underfitting the data. Even with higher dimensional feature sets, *powershap* finds more informative features than the other methods. Interestingly, most methods do not output many noise features, except for shapicant in the experiment with 20 total and 10% informative features.

2.5.2 Benchmark Datasets

Table 2.2 shows the duration of the feature selection methods and the size of the selected feature sets for each method on the different open-source datasets. Chi² does not apply to regression problems and is therefore not included in the results of the CT location and Appliances datasets. The table shows that *powershap* is again the fastest wrapper method, while the number of selected features is in line with the other methods. The filter methods tend to output more features, while forward feature selection outputs a more conservative set of features.

The performance of the selected feature sets for each classification benchmark dataset is shown in Figure 2.3a and in Figure 2.3b for the regression benchmarks. These figures show that *powershap* provides a steady performance on all datasets, consistently achieving the best or equal performance on both the cross-validation and test sets. However, even in cases with equal performance, *powershap* achieves these perfor-

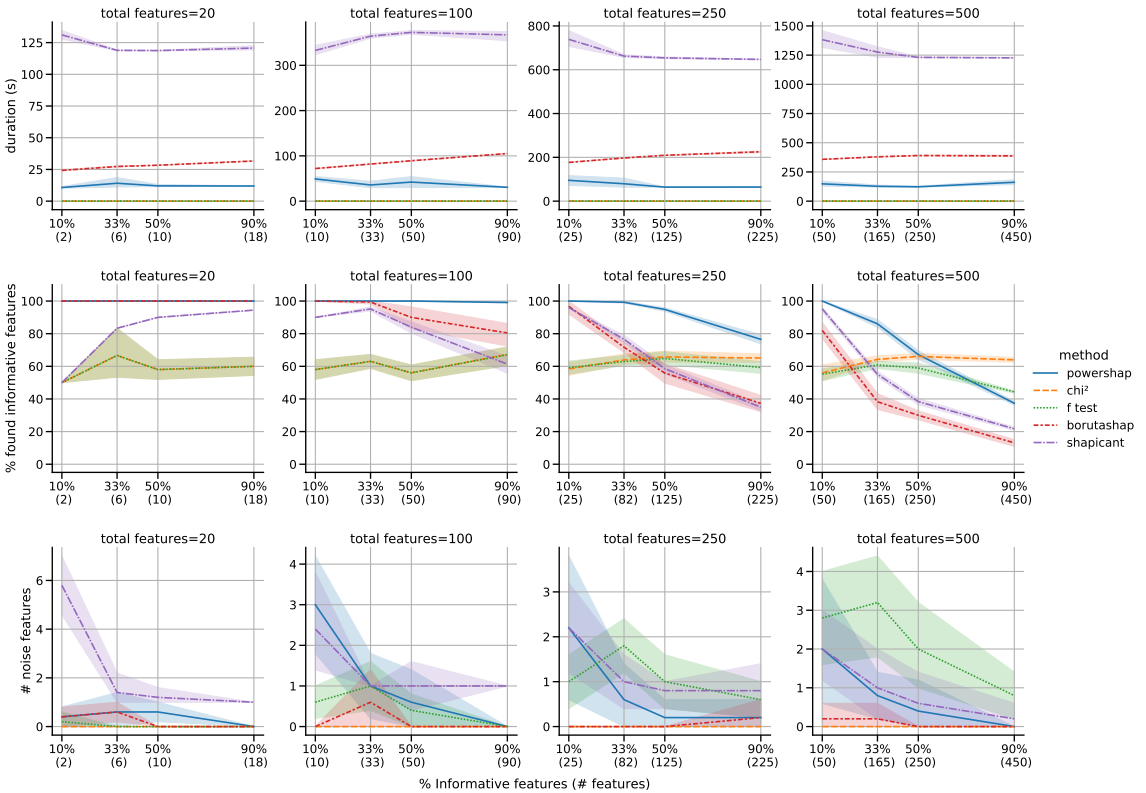
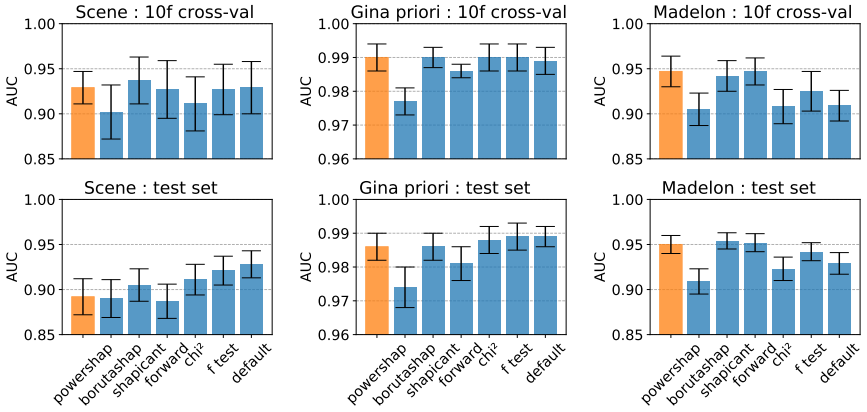
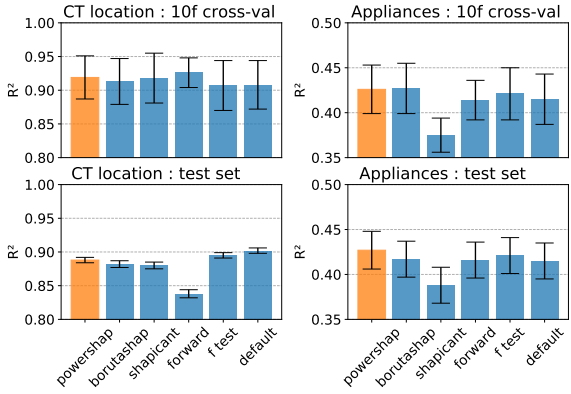


Figure 2.2: Simulation benchmark results using the make_classification sklearn function for 5000 samples with five different make_classification random seeds.



(a) Classification benchmark dataset performances.



(b) Regression benchmark dataset performances.

Figure 2.3: Benchmark performances. The error bars represent the standard deviation.

mances considerably quicker, especially compared to shapicant and forward feature selection. The CT location dataset performances show that forward feature selection tends to overfit on the cross-validation dataset while *powershap* is more robust.

2.6 Discussion

For the above test results, we used the default automatic *powershap* implementation. However, similar to many other feature selection methods, *powershap* can be further optimized or tuned. One of these optimizations is the use of a *convergence* mode to extract as many informative features as possible. In this mode, *powershap* continues recursively in automatic mode where in every recursive iteration, *powershap* re-executes but with any previously found and selected features excluded from the considered

feature set. This process continues, until no more informative features can be found. The convergence mode is especially useful in use-cases with high dimensional feature sets or datasets with a large risk of underfitting as it reduces the feature set dimension each recursive iteration to facilitate finding new informative features. As a basic experiment on the simulation benchmark, using the convergence mode for 500 features and 90% (450) informative features, the percentage of found features increases from around 38% (170) to 73% (330) without adding noise features. However, the duration also increases to the same duration as shapicant.

Other possible optimizations are also applicable to other feature selection methods, such as applying backward feature selection after *powershap* to eliminate any noise features, redundant, or duplicate features. Another possibility is optimizing the used machine learning model to better match the dataset and rerun *powershap*, e.g. by using more CatBoost estimators for datasets with large sample sizes and high dimensional feature sets.

In the benchmarking results, there are datasets where including all features perform equally well or even better, such as in the case of the Gina prior test set. In these cases, the filter methods perform well but output large feature sets, while the forward feature selection performs the worst. Alternatively, *powershap* can be used here as a fast wrapper-based dimensionality reduction method to retain approximately the same performance with a much smaller feature set. As such, there will still be a trade-off for each use-case between filter and wrapper methods based on time and performance.

We are aware that the current design of the benchmarks has some limitations. For the simulation benchmark, the `make_classification` function uses by default a hypercube to create its classification problem, resulting in a linear classification problem, which is inherently easier to classify [15]. The compared filter methods were chosen by their most common usage and availability, however, these are fast and simple methods and are of a much lower complexity than *powershap*. The same argument could be made for our choice of the forward feature selection method (as wrapper method) compared to other methods such as genetic algorithm based solutions. Furthermore, wrapper methods, and thus also *powershap*, are highly dependent on the used model, as the feature selection quality suffers from modeling issues such as for example overfitting and underfitting. Therefore, the true potential achievable performances on the benchmark datasets may differ since every use-case and dataset requires its own tuned model to achieve optimal performance. Additionally, the cut-off values and hyperparameters of none of the methods were optimized and are either set to the same value as in *powershap* or used with their default values. This might impact the performance and could have skewed the benchmark results in both directions. However, choosing the same model and the same values for hyperparameters (if possible) in all experiments, reduces potential performance differences and facilitates a fair enough comparison.

2.7 Conclusion

We proposed *powershap*, a wrapper feature selection method using Shapley values and statistical tests to determine the significance of features. *powershap* uses power calculations to optimize the number of required iterations in an automatic mode to realize fast, strong, and reliable feature selection. Benchmarks indicate that *powershap*'s performance is significantly faster and more reliable than comparable state-of-the-art shap-based wrapper methods. *Powershap* is implemented as an open-source plug-and-play *sklearn* component, increasing its accessibility and ease of use, making it a power-full Shapley feature selection method, ready for your next feature set.

Code. The code, documentation, and more benchmarks can be found using the following link: https://github.com/predict-idlab/PowerSHAP

References

[1] Jundong Li, Kewei Cheng, Suhang Wang, and et al. *Feature Selection: A Data Perspective*. ACM Computing Surveys, 50(6):94:1–94:45, December 2017.

[2] A. Jović, K. Brkić, and N. Bogunović. *A review of feature selection methods with applications*. In 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pages 1200–1205, May 2015.

[3] Binita Kumari and Tripti Swarnkar. *Filter versus Wrapper Feature Subset Selection in Large Dimensionality Micro array: A Review*. International Journal of Computer Science and Information Technologies, 2:6, 2011.

[4] Savina Colaco, Sujit Kumar, Amrita Tamang, and Vinai George Biju. *A Review on Feature Selection Algorithms*. In Emerging Research in Computing, Information, Communication and Applications, pages 133–153, Singapore, 2019. Springer.

[5] Scott M Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. In Advances in Neural Information Processing Systems 30, pages 4765–4774. Curran Associates, Inc., 2017.

[6] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. *Explainable AI: A Review of Machine Learning Interpretability Methods*. Entropy, 23(1):18, December 2020.

[7] Eoghan Keany. *BorutaShap : A wrapper feature selection method which combines the Boruta feature selection algorithm with Shapley values.*, November 2020.

[8] Miron B. Kursa and Witold R. Rudnicki. *Feature Selection with the Boruta Package*. Journal of Statistical Software, 36(11):1–13, 2010.

[9] Manuel Calzolari. *manuel-calzolari/shapicant*, April 2022.

[10] André Altmann, Laura Toloşi, Oliver Sander, and Thomas Lengauer. *Permutation importance: a corrected feature importance measure*. Bioinformatics, 26(10):1340–1347, May 2010.

[11] Leo Breiman. *Random Forests*. Machine Learning, 45(1):5–32, October 2001.

[12] B. V. North, D. Curtis, and P. C. Sham. *A Note on the Calculation of Empirical P Values from Monte Carlo Procedures*. American Journal of Human Genetics, 71(2):439–441, August 2002.

[13] Richard G. Lomax. *An introduction to statistical concepts*. Mahwah, N.J. : Lawrence Erlbaum Associates Publishers, 2007.

[14] Skipper Seabold and Josef Perktold. *statsmodels: Econometric and statistical modeling with python*. In 9th Python in Science Conference, 2010.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, and et al. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12:2825–2830, 2011.

[16] Liudmila Prokhorenkova, Gleb Gusev, and et al. *CatBoost: unbiased boosting with categorical features*. arXiv:1706.09516 [cs], January 2019.

[17] Joaquin Vanschoren. *OpenML: madelon*.

[18] Joaquin Vanschoren. *OpenML: scene*.

[19] Joaquin Vanschoren. *OpenML: gina_priori*.

[20] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*, 2017.

3

Development and evaluation of uncertainty quantifying machine learning models to predict piperacillin plasma concentrations in critically ill patients

This chapter focuses on developing and validating models to predict piperacillin plasma concentrations in ICU patients. Piperacillin, a commonly administered beta-lactam antibiotic, requires a minimum concentration for therapeutic effects, especially against resistant bacterial strains. The proposed model combines three quantile regression models to create a predictive distribution, addressing UCG2. These models are evaluated on both Ghent University Hospital data, as well as on external data provided by another hospital for prediction performance and UQ, outperforming various other models and a traditional Population Pharmacokinetic model. For UQ evaluation, the chapter introduces two adjusted metrics: the Absolute Distribution Coverage Error (ADCE) and the Distribution Coverage Error (DCE). These metrics can be utilized to optimize and select models for UQ, thereby addressing the regression part of RG2. My contributions to this chapter are the development and validation of all the machine learning models, the study design, the experiments, the creation of the quantile ensemble, the creation of the (A)DCE, and the writing of the chapter.

Development and evaluation of uncertainty quantifying machine learning models to predict piperacillin plasma concentrations in critically ill patients

Jarne Verhaeghe, Sofie A.M. Dhaese, Thomas De Corte, David Vander Mijnsbrugge, Heleen Aardema, Jan G Zijlstra, Alain G Verstraete, Veronique Stove, Pieter Colin, Femke Ongenae, Jan J. De Waele, Sofie Van Hoecke

Published in BMC Medical Informatics and Decision Making

Abstract Beta-lactam antimicrobial concentrations are frequently suboptimal in critically ill patients. Population pharmacokinetic (PopPK) modeling is the golden standard to predict drug concentrations. However, currently available PopPK models often lack predictive accuracy, making them less suited to guide dosing regimen adaptations. Furthermore, many currently developed models for clinical applications often lack uncertainty quantification. We, therefore, aimed to develop machine learning (ML) models for the prediction of piperacillin plasma concentrations while also providing uncertainty quantification with the aim of clinical practice. Blood samples for piperacillin analysis were prospectively collected from critically ill patients receiving continuous infusion of piperacillin/tazobactam. Interpretable ML models for the prediction of piperacillin concentrations were designed using CatBoost and Gaussian processes. Distribution-based Uncertainty Quantification was added to the CatBoost model using a proposed Quantile Ensemble method, usable for any model optimizing a quantile function. These models are subsequently evaluated using the distribution coverage error, a proposed interpretable uncertainty quantification calibration metric. Development and internal evaluation of the ML models were performed on the Ghent University Hospital database (752 piperacillin concentrations from 282 patients). Ensuing, ML models were compared with a published PopPK model on a database from the University Medical Centre of Groningen where a different dosing regimen is used (46 piperacillin concentrations from 15 patients.). The best performing model was the Catboost model with an RMSE and R^2 of 31.94 - 0.64 and 33.53 - 0.60 for internal evaluation with and without previous concentration. Furthermore, the results prove the added value of the proposed Quantile Ensemble model in providing clinically useful individualized uncertainty predictions and show the limits of homoscedastic methods like Gaussian Processes in clinical applications. Our results show that ML models can consistently estimate piperacillin concentrations with acceptable and high predictive accuracy when identical dosing regimens as in the training data are used, while providing highly relevant uncertainty predictions. However, generalization ca-

pabilities to other dosing schemes are limited. Notwithstanding, incorporating ML models in therapeutic drug monitoring programs seems definitely promising, and the current work provides a basis for validating the model in clinical practice.

3.1 Introduction

The morbidity, mortality, and healthcare costs associated with infectious diseases in the Intensive Care Unit (ICU) continue to be a major health issue [1]. Antimicrobial therapy remains the mainstay of treatment, with piperacillin/tazobactam (TZP) being one of the most frequently prescribed antimicrobials in the ICU worldwide [2].

Achieving therapeutic antimicrobial concentrations likely improves the clinical outcome, avoids drug toxicity, and reduces the burden of antimicrobial resistance [3, 4]. In the past few years, a wealth of evidence emerged demonstrating possible suboptimal and difficult to predict beta-lactam antimicrobial concentrations in critically ill patients following standard dosing regimens [5]. Several experts have recommended abandoning this 'one-size-fits-all approach' in ICU patients and have moved towards individualized antimicrobial dosing to reach therapeutic windows[6, 7]. This decision is based on early studies indicating that individualized dosing may decrease mortality in ICU patients [3, 8, 9]. An alternative would be to dose up to the point of toxicity to ensure target attainment is reached. For TZP, dose modifications and dosing interval adjustments are usually performed according to renal function. A renal dysfunction suggests dose reduction, and a very good renal function suggests a longer dosing administration time and/or higher dosing [3, 8, 10].

An obstacle limiting the implementation of more advanced individualized therapy is the absence of readily available measured beta-lactam antimicrobial concentrations in daily routine [3, 11]. Measuring plasma concentrations of beta-lactam antimicrobials can be performed using therapeutic drug monitoring (TDM) [12]. However, TDM is not routinely performed as the means and expertise for TDM are not always available and the time interval between sampling and availability of results is often long. An alternative to measuring the concentration is predicting it. Predicting plasma concentrations of beta-lactam antimicrobials is possible with Population pharmacokinetic (PopPK) analysis. PopPK analysis uses non-linear mixed effect modeling to simulate the relationship between the antimicrobial concentrations, the dose, time, and the patient-specific covariates [13]. Once a PopPK model is developed, a typical Pharmacokinetic (PK) profile can be generated and antimicrobial concentrations can be predicted for a given patient [13]. However, PopPK models are frequently based on (small) retrospective datasets from studies that were not primarily aimed towards the development of a PopPK model. This results in context-/subgroup-specific models with poor extrapolation properties to other datasets [14]. Therefore, sample size calculations and simulations to optimize the experimental design for PopPK modeling are not always performed as they require a considerable amount of computational re-

sources [15]. Hence, many of the current beta-lactam PopPK models in ICU patients often have poor predictive accuracy. As a result, dosing recommendations based on these PopPK models are context-specific and vary substantially from one model to another [16, 17]. To overcome these limitations of small, context- and subpopulation-specific PopPK models, large pharmacokinetic data-sharing initiatives are currently underway [18].

Another strategy to overcome these PopPK measuring and prediction limitations is to predict antimicrobial concentrations with machine learning (ML) models [19]. ML uses algorithms to find patterns and relationships in data and is not dependent on many underlying domain-specific assumptions [20]. Therefore, ML models could find new relationships between antimicrobial concentrations and covariates and use many more covariates compared to PopPK models.

However, conventional ML predictions often only provide a single output without any information about this prediction. This all-or-nothing output often limits model acceptance and inhibits risk assessments in clinical practice [21], especially in high-risk environments such as the ICU. It is possible to increase the trust and understanding in these models by providing extra prediction information using uncertainty quantification, which is especially useful for making decisions.

Hence, in this paper, we used three regression ML models to predict total plasma concentrations of piperacillin in critically ill patients. These models will then be compared to a developed and published PopPK model to research the added value of ML with respect to PopPK on an internal and external dataset. Additionally, a general distribution-based uncertainty quantification framework, the quantile ensemble, is proposed to provide uncertainty estimates for the predicted concentrations using the final ML models. At last, we also propose two uncertainty quantification performance metrics, the Absolute Distribution Coverage Error (ADCE) and the DCE (Distribution Coverage Error), usable in model selection and uncertainty quantification performance evaluation.

3.2 Related Work

3.2.1 Drug concentration prediction

Earlier works already explored the idea of using various ML models, such as support vector machines, gradient boosting trees, XGBoost, and neural networks, to predict drug concentrations for tacrolimus, remifentanil, gentamicin, risperidone, teicoplanin, phenytoin, and warfarin [22–29]. A recent study explained and validated the predictions of teicoplanin trough concentrations using Shapley values while combining the best models into a single ensemble [28, 30]. Another study used XGBoost to act as a classifier, trained on virtual patients, to select the best PopPK model to aid TDM-guided dosing [31]. However, none explored the prediction of piperacillin plasma

concentrations directly using machine learning models, nor did they include uncertainty quantification for the concentration prediction while comparing to a published PopPK model on an internal and external evaluation dataset.

3.2.2 Regression Uncertainty Quantification

In regression problems, the observable targets y are theorized to consist of a ground truth function $f(x)$, given the input features x, and additive noise ϵ. When predicting the target variable, we try to find an estimator such that $\hat{y} = \hat{f}(x)$ closely resembles the target y. In contrast, in uncertainty quantification, the goal is to correctly approximate and describe the predictive distribution $P_{\hat{\mathbf{Y}}}$ of the outputs $\hat{y}$ such that it correctly encompasses all sources of uncertainty [32].

Two metrics are important for uncertainty quantification evaluation: calibration and sharpness. The calibration quantifies how well the predictive distribution captures the ground truth uncertainty of the predictions by evaluating and comparing every quantile of the predictive distribution. The sharpness indicates the size of this predictive distribution, which is in this case the standard deviation σ. The sharpness and calibration are both required to effectively evaluate the predictive distribution, and a trade-off exists between these two metrics [33]. The goal is to have a predictive distribution that is as small as possible but still has perfect calibration. Zhao *et al.* discussed various theoretical requirements for regression distribution-based uncertainty quantification, such as different kinds of calibration and sharpness [33].

Various solutions for distribution-based uncertainty quantification for regression problems have already been proposed, such as mean-variance methods and Bayesian-based models. Mean-variance methods output a mean and variance of the Gaussian distribution by optimizing the negative log-likelihood loss. More advanced solutions use Bayesian frameworks to output a Gaussian distribution by e.g. directly optimizing the Kullback-Leibner divergence, such as Bayesian Neural Networks. These methods are often only bound to neural networks or specific models and therefore not model-agnostic [32].

3.3 Methods

3.3.1 Data

3.3.1.1 Ghent University Hospital Patients

Data from patients, included in a prospective observational study conducted between March 2016 and April 2018 in the surgical ICU of the Department of Critical Care of Ghent University Hospital (GUH, Ghent, Belgium), a tertiary university hospital with 52 ICU beds, were used. Ethical approval was obtained from the Ghent University

Hospital Ethics Committee (registration number 2016/0264). Patient agreement was obtained via opting out before participation. Patients admitted to the surgical ICU and receiving both targeted and empirical piperacillin/tazobactam (4g/0.5g powder for solution for infusion; Fresenius Kabi n.v., Schelle, Belgium) for at least 24 hours in continuous infusion were screened for eligibility. Patients younger than 18 years and patients receiving extracorporeal membrane oxygenation or renal replacement therapy (RRT) during antimicrobial therapy were excluded.

Piperacillin antimicrobial concentrations and additional data, such as biochemistry, demographic data, the Sequential Organ Failure Assessment score (SOFA) on the day of sampling [34], and the Acute Physiology and Chronic Health Evaluation (APACHE II) score on admission, were prospectively collected. Biochemical variables such as serum creatinine, albumin, platelets, lactate, white blood cells, and bilirubin were determined from the same blood sample as the antibiotic plasma concentration. Creatinine clearance (CL_{cr}) was determined by measuring urinary creatinine concentrations from an 8-hour urinary collection (mCL_{cr}). If no mCL_{cr} was available, estimated glomerular filtration rate (eGFR) as calculated by the CKD-EPI equation was used [35]. TZP dosing in GUH was as follows: loading dose of 4/0.5g/30 min immediately followed by a continuous TZP infusion depending on mCL_{cr} (eGFR): mCL_{cr} (eGFR) <15mL/min: 8/1g/24h, mCL_{cr} (eGFR) 15-29mL/min: 12/1.5g/24h and for a mCL_{cr} (eGFR) $\geq$ 30 mL/min 16/2g/24h. Measured piperacillin concentrations were not disclosed to the treating physicians.

Remnants of the blood gas syringes (RAPIDLyte; Siemens Healthcare Diagnostics, Deerfield, IL) taken as part of the routine arterial or venous blood sample every morning at 6 a.m. were collected as steady-state study material. Samples were centrifuged, and plasma was frozen at -80°C awaiting batch analysis. Total plasma concentrations of TZP were analyzed by the Laboratory of Clinical Toxicology and Drugs analysis of the Department of Laboratory Medicine of GUH using a validated fast ultra-performance liquid chromatographic method with tandem mass spectrometric detection (UPLC-MS/MS) [36].

The UPLC–MS/MS system consisted of a Waters Acquity UPLC instrument coupled to a TQD triple-quadrupole mass spectrometer (Waters Corp., Milford, MA). Separations were performed on an Acquity UPLC BEH C18 column (100 mm × 2.1 mm, 1.7 μm particle size) equipped with a 0.2 μm precolumn filter unit and a guard column (Waters Corp., Milford, MA). Analytes were measured in the multiple reaction monitoring (MRM) mode. The flow rate was set at 0.4 mL/min. The column and autosampler tray temperatures were set at 50 °C and 4 °C, respectively. 40 μL of the extract was injected into the column. The MS/MS instrument was operated with a capillary voltage of 1.00 kV, a source temperature of 140 °C, and desolvation gas (nitrogen) at 400 °C with a flow of 800 L/h. Analytes were measured in the electrospray positive (ESI+) mode. The deuterated standard D5- piperacillin from Toronto Research Chemicals (Ontario, Canada) was used as an internal standard. Data were

acquired using Masslynx 4.1 software and processed using Quanlynx 4.1 software (Waters Corp., Milford, MA).

3.3.1.2 University Medical Centre of Groningen Patients

For external evaluation, a dataset of ICU patients receiving continuous infusion TZP enrolled by Aardema et al. [37] from the University Medical Centre of Groningen (UMCG) was used. Only UMCG patients fulfilling all inclusion criteria and none of our exclusion criteria were used for evaluation. Some records were duplicates except for the concentration. As these concentrations were close to the other record ($\pm$5 mg/L), only the first occurrence was kept [38]. In UMCG, a 24-hour urine collection was used to measure CL_{cr}. If no such collection was available, the MDRD [39] formula was used to estimate the glomerular filtration rate (eGFR). TZP dosing in UMCG was as follows: loading dose of 4/0.5g/30min immediately followed by a continuous TZP infusion depending on mCL_{cr} or eGFR if no urine collection was available: mCL_{cr} (eGFR) $<$ 20 mL/min: 8/1g/24h, mCL_{cr} (eGFR) 20-39 mL/min: 8/1g/24h for the first 24 hours, followed by 12/1.5g/24h afterward and for a mCL_{cr} (eGFR) $\geq$ 40 mL/min 12/1.5g/24h.

3.3.1.3 Data Cleaning

Missing values of variables with less than 5% missing were either interpolated or replaced by the previous or next value, depending on the data and expert knowledge. Three variables had more than 5% missing values and were handled differently: urine creatinine level, body temperature, and mCL_{cr}. Urine creatinine and temperature missing values were replaced with their mean values for the Gaussian process (GP) and multilayer perceptron (MLP) models for numerical stability and with -999 for the Gradient Boosting Tree (GBT) models to indicate missings. Missing mCL_{cr} values were approximated using the adjusted Cockcroft-Gault [40] and MDRD [39] formulas. An optimized weighted sum of these formulas was determined in the cross-validation phase to be $(CockcroftGault + 2 \cdot MDRD)/3$. The CKD-EPI [35] equation was evaluated but of no additional value. Records still containing missing values after this step were deleted. The number of imputed values per variable can be found in Table 3.1.

Sequential records of patients were linked with a variable that described the previous concentration of the patient. A default previous concentration of 0 mg/L was used in every first record of a patient to indicate that no known previous concentration was available. A second feature was also included to depict the time to the previous concentration.

Table 3.1: The number of missing values for all considered features in both datasets with size N..

Feature	GUH (N=752)	UMCG (N=46)
Albumin (g/dL)	13	7
Bilirubine (mg/dL)	19	20
Creatinin clearance (mL/min)	100	0
Height (cm)	6	0
Hemoglobin (g/dL)	7	36
Lactate (mmol/L)	5	34
Platelets (/mm^3)	7	20
Serum creatinine (mg/dL)	14	0
SOFA	3	21
Temperature (°C)	107	N/A
Urine creatinine (mg/dL)	39	12
White blood cells (/mm^3)	8	36

Features that are not shown in the table contained no missing values.

3.3.1.4 Study Population

After excluding 13 patients with 21 concentrations during data cleaning, 282 patients with 752 piperacillin concentrations were included in the GUH dataset. Patients were split on their patient id into a training set, containing 240 patients and 601 concentrations, and two test sets for model evaluation: a *a priori* test set for evaluating a clinical scenario without a previous piperacillin plasma concentration, and a *a posteriori* test set to mimic the situation where one or more piperacillin plasma concentrations were available. 25 percent of patients with at least two records were used as the test set (same patients for both test sets) for the GUH evaluation. This resulted in a *a priori* test set, containing 42 patients and 151 concentrations, and a *a posteriori* test set with 42 patients and 109 concentrations.
After the exclusion of samples drawn within the first 24h of TZP therapy and data cleaning, the UMCG dataset consists of 15 patients with 46 concentrations and is used for external evaluation. The UMCG dataset is also split into a *a priori* (15 patients, 46 concentrations) and *a posteriori* test set (12 patients, 31 concentrations). Patient demographics and clinical characteristics for both the GUH and UMCG datasets are shown in Tables 3.2 and 3.3.

All statistical analyses were performed using Python (version 3.8.5) and NumPy (version 1.19.1). Continuous variables are presented as median with interquartile range (IQR). Categorical variables are presented as counts and percentages (%). For continuous data with a normal distribution, the independent-samples t-test was used to compare means (p-value). In the case of a non-normal distribution, the Wilcoxon rank-sum test was used to compare distributions between groups. For categorical

data, Pearson's χ^2 or Fisher's Exact Test were used.

3.3.2 Machine Learning Models

3.3.2.1 Models

Three models were selected. Two models were chosen as interpretable models capable of uncertainty quantification to give the clinician insights into the prediction and provide model output confidence: The Quantile regression Gradient Boosting Trees (GBT) (open-source CatBoost library, version 0.25) and Gaussian processes (GP) (GPy library, version 1.9.9). The third model is a Multilayer Perceptron model (MLP) or fully-connected feed-forward neural network (Tensorflow library, Version 2.3.0). The GUH dataset is considered a small dataset, therefore, deep learning approaches are not suitable for this problem. However, to prove this statement, the MLP is included in this study. The MLP model will not be used for uncertainty quantification due to the limited dataset. Therefore, more advanced uncertainty quantifying methods, such as Bayesian Neural Networks, are not considered in this study.

For each ML model, two different sub-models were trained. The first model predicts a concentration when a previous TDM measurement is available and, denoted as the *prev* model, and is used for the a posteriori case. The second model denoted as the *new* model, predicts a concentration when there is no prior TDM measurement available, i.e. a previous concentration of 0, and is used for the a priori case.

For the GP models, the *new* model was built using the Radial Basis Function kernel, while the *prev* model used the Multi-Layer Perceptron kernel to represent the prior knowledge. Each final GBT model is an ensemble consisting of three sub-models: one model dedicated to predicting the concentration, and two models to predict the upper and lower prediction quantile for uncertainty quantification, further referred to as the Quantile Ensemble.

3.3.2.2 Model development strategy

Model development was performed on the GUH training set, using 10-fold cross-validation (CV), where the patients are split using their patient id and the number of measurements per patient was used to stratify the split. The CV was used to select the features, determine parameters, and compare different techniques. Prediction errors were evaluated using the mean error (ME), the mean absolute error (MAE), the root mean square error (RMSE), the coefficient of determination (R^2), median absolute percentage error (MdAPE), and median percentage error (MdPE).

For model development, the RMSE was the preferred metric to determine the best model and feature selection algorithms as it quantifies the average error and quadratically penalizes large errors. Furthermore, to determine the best hyperparameters for the Quantile models in the Quantile Ensemble for uncertainty quantification, the pro-

Table 3.2: Descriptive statistics for the GUH and UMCG dataset (Demographics, admission category, and TZP treatment. The timing of the lab results is from the first piperacillin concentration available for analysis. n is the number of patients included for demographics, admission category, and TZP treatment.

Variable	GUH (n=285)	UMCG (n=15)	*p*-value
Demographics			
Sex (male)	183 (64.9%)	13 (87.0%)	0.718
Age, median (IQR) (yr)	64 (53 – 74)	60 (54 – 66)	0.133
Height, median (IQR) (cm)	170 (165 – 178)	175 (172 – 178)	0.101
Weight, median (IQR) (kg)	75.0 (64.2 – 85.0)	77.0 (70.0 – 90.0)	0.138
Admission APACHE II score median (IQR)	23.0 (3.0 – 29.0)	NA	NA
Admission APACHE IV score median (IQR)	NA	74.0 (65 – 87)	NA
SOFA score on the day of sampling median (IQR)	5 (3 – 8)	12 (9 – 14)	<0.001
ICU mortality (%)	33 (11.7%)	4 (26.7%)	0.329
Admission category (%)			
Medical	118 (41.8%)	4 (26.7%)	0.599
Surgical	135 (47.9%)	8 (53.3%)	0.883
Trauma	29 (10.3%)	2 (13.3%)	0.662
TZP Treatment			
Duration of TZP therapy, median (IQR) (days)	3 (1 – 5)	3 (2 – 6)	0.373
Piperacillin concentration, median (IQR) (mg/L)	81.0 (54.4 –121.4)	50.3 (36.5 – 80.9)	0.260
No. of blood samples per patient, median (IQR)	2 (1 – 3)	4 (1 – 5)	0.350
Blood Sample time relative to treatment median (IQR) (hours)	63 (35 – 115)	30 (12 – 48)	<0.001
Time to the previous concentration median (IQR) (hours)	24 (24 - 48)	24 (24 - 24)	0.023

Table 3.3: Descriptive statistics for the GUH and UMCG dataset (Lab values). n is the number of samples included for lab results.

Variable	GUH (n=752)	UMCG (n=46)	*p*-value
Lab results			
Serum creatinine, median (IQR) (mg/dL)	0.69 (0.51 – 1.04)	0.76 (0.52 – 1.33)	0.255
Measured creatinine clearance, median (IQR) (mL/min)	107.5 (65.5 – 143.7)	96.9 (40.5 – 127.9)	0.125
Albumin, median (IQR) (g/L)	23.0 (20.0 – 26.0)	22.5 (19.0 – 26.0)	0.257
Platelets, median (IQR) (10^9/L)	281 (174 – 414)	160 (123 – 199)	0.729
Lactate, median (IQR) (mg/dL)	10.9 (8.5 – 14.3)	12.0 (9.0 – 18.0)	0.561
White blood cells, median (IQR) (10^9/L)	12.6 (9.9 – 16.9)	11.7 (8.5 – 19.5)	0.689
Bilirubin, median (IQR) (mg/dL)	0.60 (0.40 – 1.10)	0.99 (0.36 – 2.70)	0.468
Previous 24h fluid balance median (IQR) (ml/24h)	421.9 (-359.7 – 1399.3)	816.5 (-210.0 – 2328.2)	<0.001

posed Absolute Distribution Coverage Error metric was the preferred metric, minimizing the metric.

3.3.2.3 Feature selection

The features considered for model building were: age (yrs), height (cm), weight (kg), race, sex, SOFA, lactate (mmol/L), serum creatinine (mg/dL), urine creatinine (mmol/L), creatinine clearance (CL_{cr}, mL/min), hematocrit (%), platelets (/mm3), white blood cells (/mm3), red blood cells (/mm3), bilirubin (mg/dL), hemoglobin (g/dL), albumin (g/dL), fluid balance (mL/24h), piperacillin/tazobactam (TZP) dose per hour (mg/h), temperature (°C), AKI stage (cf. KDIGO definition), cumulative administered dose (mg), previous piperacillin concentration (mg/L), reason for ICU admission (i.e. medical, surgical, trauma admission, neurological trauma), dobutamine usage (yes/no), vasopressor usage (yes/no), epinephrine usage (yes/no), dopamine usage (yes/no), norepinephrine usage (yes/no), milrinone usage (yes/no), and phenylephrine (yes/no). These were all available features, collected on the plausibility of prediction impact as judged by clinicians and permission of collection. Adding features to include changes over time did not result in better performance and were therefore not included. The biochemistry features are determined from the same moment of drawing the piperacillin concentration. Together with the other variables, this creates a feature set to predict the concentration at the moment of drawing the blood sample when the biochemistry variables become available. Hence, our model predicts the antimicrobial concentration at the time of drawing the sample using readily available data, without the need of expensive laboratory equipment or the large turnaround

time required for the concentration determination.

The feature selection for the GBT ensemble and the MLP model used a novel method called PowerShap, a feature selection algorithm that uses statistical hypothesis testing and power calculations on Shapley values [41]. The GP model is not supported by the PowerShap library for feature selection and therefore forward feature selection was used, iteratively adding features providing the best results. Feature selection was performed before optimizing the hyperparameters and re-executed whenever new techniques, models, or loss functions were tried.

3.3.3 Uncertainty Quantification

Providing uncertainty quantification for any prediction is important, especially for high-risk environments such as the ICU [42]. Therefore, for this study, specific attention was given to providing uncertainty quantification. The goal of the uncertainty quantification in this study is to provide a complete predictive distribution together with the regression output to enable calculating the likelihood that the true drug concentration will be between specific bounds. This is especially useful for evaluating whether the predicted concentration attains a therapeutic drug concentration window.

For the current application, the predictive distribution is assumed to be Gaussian, which is characterized by two parameters: the mean μ, corresponding with the regression output, and the standard deviation σ. The Gaussian assumption is inherently incorporated into the Gaussian process model. For the other method, the proposed Quantile Ensemble, the assumption is used to provide a predictive distribution.

3.3.3.1 Gaussian Process

The output of a Gaussian process is a Gaussian Distribution with estimated mean $\tilde{\mu}$ and estimated standard deviation $\tilde{\sigma}$ and therefore requires no further calculations. Furthermore, the Gaussian process is a homoscedastic Bayesian-based uncertainty quantification method, where the standard deviation is approximately equal for all samples, providing a global uncertainty prediction.

3.3.3.2 Quantile Ensemble Model

There are three models in the Quantile Ensemble Model. One for the regression output or $\tilde{\mu}$ and two for estimating an upper $\tilde{y}_U$ and lower quantile $\tilde{y}_L$ of the predictive distribution.

First, a specific coverage p is defined, which is a hyperparameter that specifies the upper and lower quantile:

$$Q_{up} = 0.5 + \frac{p}{2} \tag{3.1}$$

$$Q_{low} = 0.5 - \frac{p}{2} \tag{3.2}$$

Then, given the quantile function of the Gaussian distribution for a coverage p:

$$Q(p;\mu,\sigma) = \mu + \sigma\sqrt{2}erf^{-1}(2p-1) \tag{3.3}$$

With μ and σ parameters of the Gaussian distribution. It is then possible to derive $\tilde{\sigma}$ using the predicted upper $\tilde{y}_U$ and lower $\tilde{y}_L$ quantiles:

$$\tilde{\sigma} = \frac{\tilde{y}_U - \tilde{y}_L}{\sqrt{2}\cdot erf^{-1}(p)} \tag{3.4}$$

3.3.3.3 Distribution Inferences

Given the estimated distribution parameters $\tilde{\mu}$ and $\tilde{\sigma}$ and the Gaussian distribution quantile function, any prediction interval can now be estimated for a given coverage p:

$$[\tilde{y}_{L_p}, \tilde{y}_{U_p}] = \left[Q\left(\frac{1-p}{2}, \tilde{\sigma}, \tilde{\mu}\right), Q\left(\frac{1+p}{2}, \tilde{\sigma}, \tilde{\mu}\right)\right] \tag{3.5}$$

Additionally, the estimated coverage percentage $\tilde{p}$ between any upper y_U and lower bound y_L can then be calculated as follows:

$$\tilde{p} = erf\left(\frac{y_U - \tilde{\mu}}{\tilde{\sigma}\sqrt{2}}\right) - erf\left(\frac{y_L - \tilde{\mu}}{\tilde{\sigma}\sqrt{2}}\right) \tag{3.6}$$

By predicting the quantiles and recalculating the predictive distribution for each individual sample, the Quantile Ensemble becomes a heteroscedastic method. In contrast to a homoscedastic model, a heteroscedastic model provides standard deviations that can differ for each sample, thereby providing an individualized uncertainty prediction. Do note that the provided Quantile Ensemble method can be applied to any model optimizing a quantile loss function, and is not limited to the CatBoost model. Although only two models are required to estimate the distribution, using three models provides higher calibration performance. When using two models, one model predicts the mean and the other predicts a single quantile, which can be converted into the standard deviation using the same method.

3.3.3.4 Uncertainty Quantification Evaluation

To ensure the uncertainty quantification is accurate, the calibration and sharpness of the predictive distribution should be evaluated. To measure the calibration, the (Absolute) Distribution Coverage Error ((A)DCE) is proposed for heuristic calibration calculation in distributions, based on the Prediction Interval Coverage Percentage [32] that calculates the empirical coverage of a prediction interval with upper and lower bounds y_U and y_L :

$$PICP(\mathbf{y}, \mathbf{y}_L, \mathbf{y}_U) = \frac{1}{N}\sum_{i=1}^{N} I\{y_{L_i} \leq y_i \leq y_{U_i}\} \tag{3.7}$$

Where y is the target value vector, I the indicator function, and N the number of included data points.
We then define the coverage function C, which calculates the empirical coverage of estimated centered prediction intervals extracted from the predictive distribution with estimated parameters $\tilde{\sigma}$ and $\tilde{\mu}$ for a specified coverage p:

$$C(\mathbf{y}, p, \tilde{\sigma}, \tilde{\mu}) = \\ PICP\left(\mathbf{y}, Q\left(\frac{1-p}{2}, \tilde{\sigma}, \tilde{\mu}\right), Q\left(\frac{1+p}{2}, \tilde{\sigma}, \tilde{\mu}\right)\right) \tag{3.8}$$

A sampling rate S is defined for the heuristic calculation of the calibration, corresponding to the step size of the percentages, which is set to 1000 in this work. To bound the absolute values of DCE and ADCE to $[0, 1]$ and thereby, both are multiplied by 2:

$$DCE(y, \tilde{\theta}) = \frac{2}{S}\sum_{i=0}^{S}\left(C(y, i/S, \tilde{\sigma}, \tilde{\mu}) - \frac{i}{S}\right) \tag{3.9}$$

$$ADCE(y, \tilde{\theta}) = \frac{2}{S}\sum_{i=0}^{S}\left|C(y, i/S, \tilde{\sigma}, \tilde{\mu}) - \frac{i}{S}\right| \tag{3.10}$$

The ADCE quantifies the average calibration of the complete predictive distribution (for a more elaborate explanation of average calibration, we refer to [33]). The DCE shows any calibration biases, either consistently underestimating (positive) or overestimating its coverage (negative). However, do note that the DCE can be 0 while ADCE can be 1, but not vice versa. Therefore, it is advised to always provide both.

The calibration can then be plotted in calibration plots, plotting the coverage C for each p, for further visual inspection of the calibration performance.

3.3.4 Hyperparameters

3.3.4.1 Gradient Boosting Trees (GBT)

Four hyperparameters of the GBT ensemble were optimized using cross-validation: tree depth, leaf regularization, border count, and the quantile coverage p. The final hyperparameters of all GBT *new* sub-models were chosen to be 4, 1, 250, and 0.80, respectively. For the GBT *prev* model, they were 3, 4, 50, and 0.82, respectively. Therefore, the loss function of the *new* and *prev* sub-model responsible for the regression output was $Quantile : alpha = 0.5$. For the *new* upper and lower quantile models the loss functions were $Quantile : alpha = 0.90$ and $Quantile : alpha = 0.10$ respectively, while they were $Quantile : alpha = 0.91$ and $Quantile : alpha = 0.09$ for the *prev* upper and lower quantile models.

3.3.4.2 Gaussian Processes (GP)

For optimizing the GP, at least one feature is required to determine the kernel, but a kernel is required to perform feature selection. Since the feature with the largest correlation to the concentration has a high chance of being included in the final feature set, and can thus be used for kernel selection. The feature with the largest Pearson correlation to the concentration, CL_{cr}, was chosen to determine the most adequate kernel. The GP *prev* model used the following features (ordered in descending importance): previous concentration (MLP kernel weight variance = 0.085; higher weight indicates larger importance), creatinine clearance (0.080), serum creatinine (0.056), and fluid balance (0.0078). The GP *new* model used the following features: creatinine clearance (RBF kernel lengthscale = 0.95; lower weight indicates larger importance), serum creatinine (1.73), albumin (13.45), and bilirubin (152.7).

3.3.4.3 Multi-Layer Perceptron (MLP)

The developed neural network is a neural network with 3 layers, each with a width of 32 nodes and using the ReLu activation function. The models were optimized using the Tensorflow Adam optimizer with a learning rate of 0.0005, batch size of 32, and 75 epochs.

3.3.5 Population PK Model

A two-compartmental piperacillin PopPK model with parallel linear/Michaelis-Menten elimination was used to predict antimicrobial concentrations for model comparison [43]. In this PopPK model, the mCL_{cr} (mL/min), normalized to 100mL/min, and the body weight, normalized to 70kg, were included for determining the clearance using a power function with 0.75 as an exponent. The volume of distribution had an exponent of 1. This model is described as follows:

$$CL = TVCL \left(\frac{CL_{CR}}{100}\right) \left(\frac{WEIGHT}{70}\right)^{0.75} \tag{3.11}$$

$$V = TVV \left(\frac{WEIGHT}{70}\right) \tag{3.12}$$

$$V_p = TVV_p \left(\frac{WEIGHT}{70}\right) \tag{3.13}$$

$$Q = TVQ \left(\frac{WEIGHT}{70}\right)^{0.75} \tag{3.14}$$

The median and 95% confidence intervals for model parameters drug clearance (CL), volume of the central compartment (V), volume of the peripheral compartment

(V_p), and intercompartmental clearance (Q) were 9 (7.69-11) L/h, 6.18 (4.9-11.2) L, 11.17 (7.26-12) L, and 15.61 (12.66-23.8) L/h. The Michaelis-Menten constant (K_m) and the maximum elimination rate for Michaelis-Menten elimination (V_{max}) were estimated without population variability in the model to avoid overfitting. The population estimates for K_m and V_{max} were 37.09 mg/L and 353.57 mg/h, respectively.

NONMEM®(version 7.5; GloboMax LCC, Hanover, MD, CA, USA) was used to predict antimicrobial concentrations with the published PopPK model. Predictions made with the PopPK model were deterministic, i.e. without residual uncertainty.

3.3.6 Concentration Prediction Evaluation

For *a priori* evaluation, *a priori* PopPK predictions were generated with a parameter distribution equal to the population parameter distribution of the PopPK model (i.e. Bayesian prior) and compared to the *new* ML models. For *a posteriori* evaluation, individual PK parameter estimates, as opposed to population PK parameter estimates, can be used to generate *a posteriori* predictions (the Bayesian posterior), and compared to the *prev* ML models [44–46]. All used ML models provide deterministic predictions.

PopPK and ML predictions for the GUH database were also converted into different categories to assess target attainment, required in clinical practice. The first category, subtherapeutic, contains unbound concentrations below the target attainment value of four times the minimum inhibitory concentration (MIC) of Pseudomonas aeruginosa of 16 mg/L [47]. This breakpoint, the upper limit of piperacillin susceptibility, represents a worst-case scenario for empirical dosing when the MIC of the pathogen is not yet known. However, this can be changed when the MIC of the pathogen is known. The supratherapeutic category is based upon the toxicity risk of TZP, set at an unbound concentration of 112 mg/L [48]. The therapeutic category lies between these two categories. Classification performance was evaluated using the precision (i.e. positive predictive value (PPV)), specificity (i.e. selectivity or true negative rate (TNR)), sensitivity (i.e. recall, hit rate, or true positive rate (TPR)), and F1-score metrics.

As only total plasma concentrations were measured, a protein binding factor of 30% was considered, resulting in a subtherapeutic threshold of 91.43 mg/L and a supratherapeutic threshold of 160 mg/L [49].

3.4 Results

3.4.1 Final features

Features were collected on the plausibility of prediction impact as judged by clinicians and by permission of collection. Adding features to include changes over time did not result in better performance and were hence not included. Features included in

Table 3.4: Features used by each model.

Feature	GBT *prev*	GBT *new*	GP *prev*	GP *new*	MLP	PopPK
Albumine (g/dL)	X	X		X	X	
Bilirubine (mg/dL)	X	X		X		
Creatinine clearance (mL/min)	X	X	X	X	X	X
Fluid balance (mL/24h)		X	X			
Height (cm)	X	X			X	
Lactate (mg/dL)	X	X				
Platelets (/mm^3)	X	X			X	
Red blood cells (/mm^3)	X				X	
Previous concentration (mg/L)	X		X		X	
Sex					X	
Hours since start treatment (h)	X					
Serum creatinine (mg/dL)	X	X	X	X	X	
Weight (kg)						X

the different ML models after feature selection can be seen in Table 3.4; any features not present in the table were not in the final feature set of any model as they did not increase the performance. The GP models did not use many features as they are highly sensitive to high dimensions and therefore prefer small feature sets. The features included in every model are the creatinine clearance and the serum creatinine, and prove their already known predictive capabilities. Albumin return in four of the five ML models as a feature, indicating it as important for predicting the piperacillin plasma concentrations. Weight is seen as an important indicator for the volume of distribution in the PopPK model. However, no ML model included weight, as the addition of weight in the cross-validation phase only decreased performance, always preferring height over weight. As an experiment, when height was not available as a feature, the models included weight as a predictor, indicating that the models found height more informative than weight in this dataset. In the GUH dataset, the weight and the height were normally distributed with the same standard deviation and contained only a few outliers. As a result, model predictions for patients with an outlying weight could be less optimal.

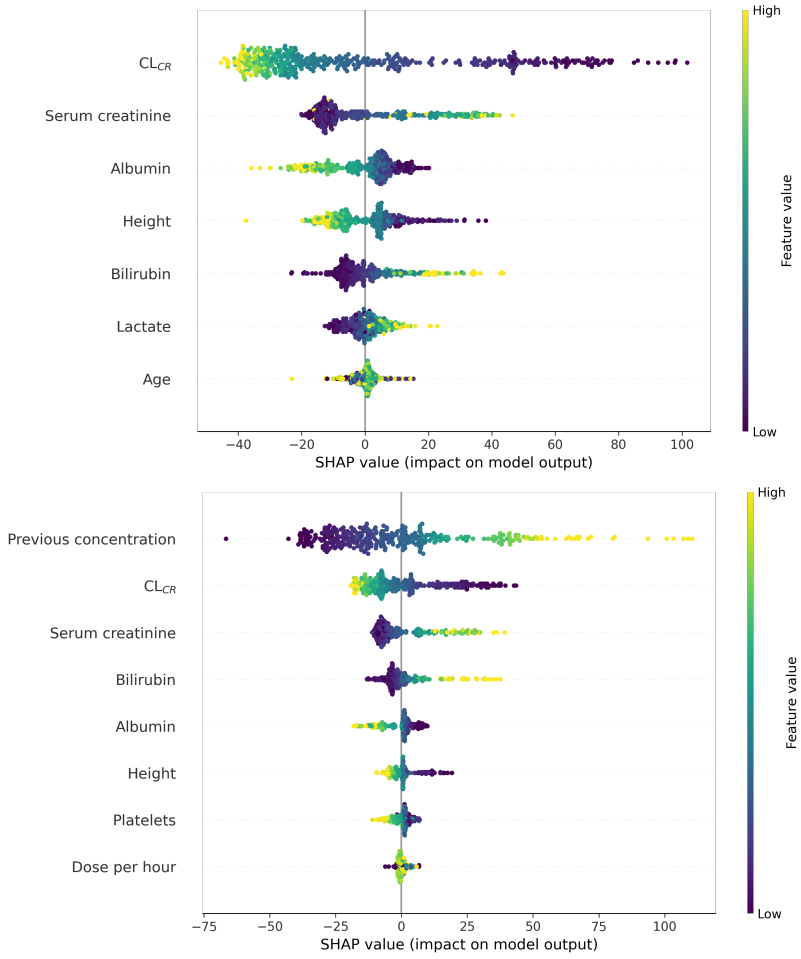


Figure 3.1: SHAP visualization for GBT new (top) and GBT prev (bottom).

The SHAP values are in mg/L.

3.4.2 Visual interpretation of ML Models

Visualization of the shapely additive explanation values (SHAP) [30] for the GBT *prev* and *new* model are shown in Figure 3.1. The SHAP value quantifies the impact of a feature on pushing the concentration output from the baseline prediction to the actual prediction. The baseline is considered the average of all predictions in training [30]. Both figures show the CL_{cr} as the most important predictor, where a low CL_{cr} results in high piperacillin concentrations, in accordance with the literature.

3.4.3 Evaluation

3.4.3.1 Concentration prediction

The concentration prediction performance on the GUH and UMCG datasets is shown in Table 3.5. The metrics are also calculated on the log scale, as many large concentration values might skew the regression results.

3.4.3.2 Target Attainment prediction

A priori and *a posteriori* performance of the ML models and the PopPK model on the GUH database for target attainment prediction is summarized in Table 3.6. This was not performed for the UMCG dataset as the number of samples was insufficient. As the MLP model was solely used to show the limited performance of a deep learning model, the MLP model was not included for target attainment prediction.

3.4.3.3 Compensation of missing Creatinine Clearances.

The most important feature in both models is the creatinine clearance, however, not all ICUs routinely measure this covariate and instead use eGFR formulas. To evaluate these cases, all measured CL_{cr} values in the test set are replaced using the weighted eGFR formula. The evaluation results of the best models, GBT *new* and GBT *prev*, on the GUH dataset for this case were (RMSE / MAE / ME / R^2) 40.71 / 27.84 / -7.24 / 0.41 and 34.51 / 19.37 / -5.75 / 0.58, respectively. As expected, the performance is worse. However, the models are still usable and therefore the weighted formula is an alternative for cases where the creatinine clearance cannot be measured.

3.4.4 Uncertainty Quantification

In Table 3.7, the calibration and sharpness results of both the GBT and the GP model are shown. Figure 3.2 and 3.3 show the visualization of the calibrations for the GUH and the UMCG dataset, respectively.

3.4.5 Patient Case Study

One patient will be discussed in this section to illustrate what information the final ML models can provide to the clinician when predicting the plasma concentration in clinical practice. The discussed patient is a patient with a previously measured concentration and is therefore handled by the GBT *prev* model and *a posteriori* PopPK model for comparison. The features at the time of measuring the discussed observed plasma concentration of this patient are shown in Table 3.8.

The patient suffered a wound infection in the lower legs with amputation of the right leg. The observed plasma concentration was 129.60 mg/L, and the patient had a previous concentration of 173.40 mg/L. The GBT *prev* model predicted a piperacillin

Table 3.5: Evaluation performance of ML and PopPK models. All RMSE, MAE, and ME values are in mg/L. The values in parenthesises are in log scale.

Model	RMSE	MAE	ME	R^2	MdAPE	MdPE
GUH evaluation: *a priori*						
GBT *new*	**34.27** (**0.38**)	**21.55** (**0.25**)	-4.09 (**0.00**)	**0.58** (**0.57**)	**17.29**% (**3.83**%)	**0.06**% (**0.01**%)
GP *new*	37.41 (0.43)	23.54 (0.28)	2.04 (0.07)	0.50 (0.46)	21.39% (4.79%)	-3.83% (-0.84%)
MLP	38.56 (0.47)	27.35 (0.34)	2.58 (0.05)	0.47 (0.36)	23.09% (5.29%)	-5.34% (-1.26%)
PopPK	57.97 (0.64)	39.67 (0.54)	-30.27 (-0.45)	-0.19 (-0.21)	40.79% (11.60%)	38.33% (11.41%)
GUH evaluation: *a posteriori*						
GBT *prev*	**32.93** (**0.27**)	**18.22** (**0.19**)	-6.55 (-0.02)	**0.62** (**0.73**)	**12.75**% (**3.09**%)	**1.77**% (**0.43**%)
GP *prev*	34.03 (0.28)	19.41 (0.21)	**-3.83** (**-0.01**)	0.59 (0.71)	16.48% (3.79%)	-3.76% (-0.92%)
MLP	37.20 (0.36)	23.64 (0.26)	-4.87 (-0.03)	0.51 (0.51)	17.06% (4.14%)	0.73% (0.17%)
PopPK	49.58 (0.43)	31.28 (0.32)	4.91 (0.03)	0.14 (0.32)	26.09% (6.69%)	-1.85% (-0.43%)
UMCG evaluation: *a priori*						
GBT *new*	**43.92** (0.78)	38.67 (0.62)	**30.67** (0.58)	**0.36** (-0.12)	68.38% (12.89%)	-68.38% (-12.89%)
GP *new*	64.99 (0.89)	55.31 (0.74)	50.90 (0.72)	-0.39 (-0.45)	84.88% (15.33%)	-84.88% (-15.33%)
MLP	62.28 (0.85)	51.47 (0.71)	38.52 (0.63)	-0.28 (-0.33)	83.09% (14.97%)	-83.09% (-14.97%)
PopPK	50.46 (**0.67**)	**31.50** (**0.55**)	-23.97 (**-0.30**)	0.16 (**0.18**)	**39.84**% (**12.31**%)	**33.88** % (**9.85**%)
UMCG evaluation: *a posteriori*						
GBT *prev*	28.12 (0.57)	21.11(0.40)	15.01 (0.37)	0.68 (0.25)	37.20% (8.46%)	-37.20% (-8.46%)
GP *prev*	31.58 (**0.57**)	22.73 (**0.39**)	18.05 (0.35)	0.60 (**0.26**)	**25.15**% (**6.90**%)	-25.15% (-6.9%)
MLP	30.35 (0.64)	26.55 (0.50)	22.45 (0.47)	0.63 (0.06)	54.16% (10.48 %)	-54.16% (-10.48%)
PopPK	**25.89** (0.62)	**19.95** (0.45)	**2.15** (**-0.00**)	**0.73** (0.13)	26.69% (7.31%)	**3.31**% (**0.87**%)

Table 3.6: GUH a priori classification performance of the ML and PopPK models.

Model	Range	Precision	Specificity	Sensitivity	F1-score	Support
			A priori			
GBT *new*	Sub.	**0.88**	**0.88**	0.89	**0.88**	99
	Ther.	**0.62**	0.58	**0.69**	**0.65**	35
	Sup.	**0.67**	0.77	0.47	**0.55**	17
GP *new*	Sub.	0.88	0.88	0.85	0.87	99
	Ther.	0.53	0.47	0.60	0.56	35
	Sup.	0.56	0.58	**0.53**	0.55	17
PopPK	Sub.	0.71	0.60	**0.98**	0.82	99
	Ther.	0.50	**0.89**	0.11	0.19	35
	Sup.	0.83	**0.94**	0.29	0.43	17
			A posteriori			
GBT *prev*	Sub.	**0.93**	**0.93**	**0.92**	**0.93**	76
	Ther.	0.63	**0.54**	**0.79**	**0.70**	24
	Sup.	**0.75**	**0.89**	0.33	0.46	9
GP *prev*	Sub.	0.92	0.92	0.92	0.92	76
	Ther.	**0.59**	0.53	0.67	0.63	24
	Sup.	0.50	0.67	0.33	0.40	9
PopPK	Sub.	0.84	0.86	0.75	0.79	76
	Ther.	0.35	0.15	0.46	0.40	24
	Sup.	0.50	0.44	**0.56**	**0.53**	9

Subtherapeutic (Sub.): $<$91.43 mg/L, Therapeutic (Ther.): $\geq$91.43 mg/L and $<$160 mg/L, Supratherapeutic (Sup.): $\geq$160 mg/L. Support indicates the number of samples in that range. Bold indicates the best model for that metric and case.

Table 3.7: Uncertainty quantification performance of the GBT models and the GP models.

Model	ADCE	DCE	Sharpness (std) (mg/L)
GUH evaluation: *a priori*			
GBT *new*	**0.06**	**0.01**	**23.48 (11.41)**
GP *new*	0.29	0.29	41.22 (4.26)
GUH evaluation: *a posteriori*			
GBT *prev*	**0.07**	**0.04**	**17.98 (8.62)**
GP *prev*	0.28	0.28	28.94 (0.86)
UMCG evaluation: *a priori*			
GBT *new*	0.62	-0.62	**25.50 (13.43)**
GP *new*	**0.39**	**-0.39**	42.75 (9.67)
UMCG evaluation: *a posteriori*			
GBT *prev*	0.31	-0.31	**17.61 (8.02)**
GP *prev*	**0.15**	**0.08**	28.22 (0.99)

Bold indicates the best model for that metric and case.

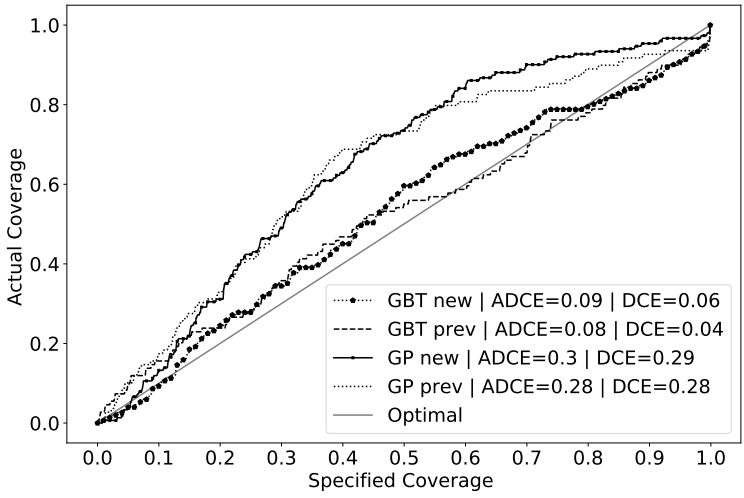


Figure 3.2: Coverage plot of all uncertainty quantification models on the GUH dataset.

The specified coverage is the p to provide the prediction intervals. The actual coverage is the measured coverage C.

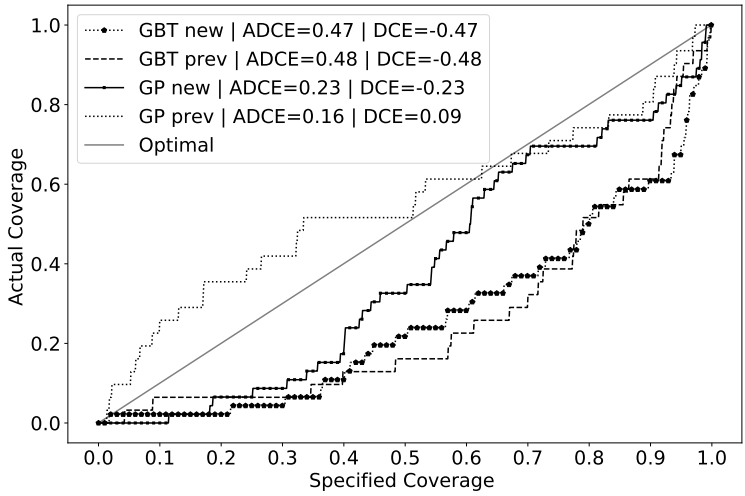


Figure 3.3: Coverage plot of the uncertainty quantification models on the UMCG dataset. The specified coverage is the p to provide the prediction intervals. The actual coverage is the measured coverage C.

Table 3.8: Features of the discussed patient

Height (cm)	**Serum creatinine (mg/dL)**	**Platelets (plt/mm^3)**	**Bilirubin (mg/dL)**	**Lactate (mg/dL)**
170	0.51	248.0	1.7	9.1
Albumin (g/L)	**Hours since start treatment (h)**	**CL_{Cr} (mL/min)**	**Red blood cells (/mm^3)**	
20	78	72.60	3.75	

Table 3.9: A posteriori PopPK prediction.

CL	**V**	**Q**	**V_p**	**K_m**	**V_{max}**	**Pred** (mg/L)
2.36	6.01	15.30	10.90	37.10	354.0	161.0

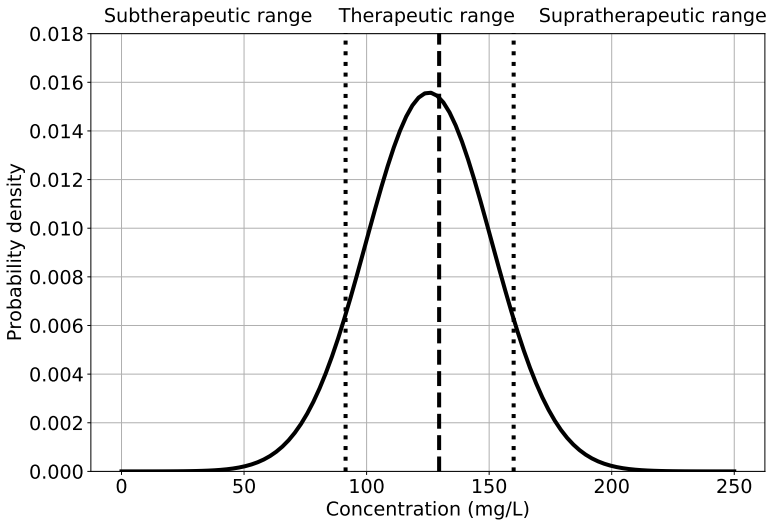


Figure 3.4: Prediction output of the first discussed patient with the GBT prev model.
The dashed (middle) line is the observed concentration, and the dotted (outer) lines indicate the therapeutic range boundaries.

plasma concentration of 123.59 mg/L with an estimated standard deviation of 27.40 mg/L while the *a posteriori* PopPK model predicted 161.0 mg/L. The output distribution of the ML model is visualized in Figure 3.4, and the estimated PopPK parameters are shown in Table 3.9. With this information, the patient was determined by the ML model to be in the therapeutic dosing range with 76.1% certainty, in the subtherapeutic range with 13.4% certainty, and 10.5% certainty for the supratherapeutic range. The influence of each feature is shown in Figure 3.5 using the SHAP-values. Neither the dose per hour nor the height is visible in the SHAP plots, as their SHAP values are too small to visualize. Here we can see that the previous plasma concentration has the highest impact on the output due to its high value of 173.40 mg/L, the low serum creatinine has the second-highest impact and reduces the final prediction. All *a posteriori* PopPK model estimates were: CL = 2.36 L/h, V = 6.01 L, Q = 15.30 L/h, V_p = 10.90 L, K_m = 37.10 mg/L, and V_{max} = 354.0 mg/h. The PopPK model had a noticeably low clearance, while the other parameters are average values, possibly resulting in overprediction of the concentration.

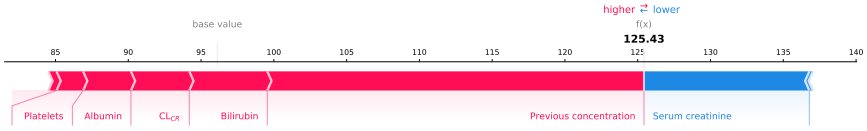


Figure 3.5: SHAP visualization for a given patient with the GBT prev model.

The red values increase the output while the blue values decrease the output. The mentioned values are piperacillin plasma concentrations (mg/L).

3.5 Discussion

The ML models were developed with piperacillin plasma concentrations from critically ill patients receiving a continuous TZP infusion. The model with the smallest bias and imprecision for the piperacillin concentration predictions on the GUH evaluation sets was the GBT model. The PopPK model showed better performance on the UMCG evaluation *a posteriori* set in the natural scale. In log scale, the GP *prev* and the GBT *prev* models performed better, while this is reversed for the *a priori* UMCG test set. All models tend to lose performance for higher concentrations (i.e. the supratherapeutic range), which can be explained by the lack of data in this range.

Predicted drug concentrations are point estimates and reporting the degree of uncertainty is important for clinical decision-making [21]. To this end, we proposed the Quantile Ensemble method and compared it to a Gaussian process model to provide and evaluate reliable prediction intervals and predictive distributions. Looking at the results, the GP model's predictive distribution calibration and sharpness are much worse than the GBT model. The size of the predictive distribution of the GBT models is around 40% smaller than the GP models while achieving much better calibration metrics. Furthermore, the GP model's sharpness standard deviation is low, showing the homoscedastic nature of the model and its negative impact on uncertainty quantification. The GP model has to increase the size of the predictive distribution to account for the global uncertainty prediction, as a result, the individual uncertainty predictions have very large bounds and are the same for every patient. Subsequently, the GP model uncertainty prediction has no real clinical value as a more individualized prediction is preferred for each patient. Looking at the GUH test set ADCE and DCE values and the calibration plots, the large positive GP model DCE values indicate a large conservative uncertainty prediction. This behavior proves that the size of the distribution can be much smaller on average, further supported by the large GP model sharpness and the better GBT model performances as mentioned previously. Furthermore, we can also conclude that the GBT model still provides slightly conservative uncertainty predictions, given the small positive DCE. These results show the added value and the strength of the heteroscedastic approach of the Quantile En-

semble method, even on a small dataset, and the interpretability and strength of the ADCE and DCE metrics, even in model selection.

Together with the regression and uncertainty quantification performance, the preferred model is the GBT model. For clinical practice, Struys *et al.* defined the threshold for drug prediction model acceptance as 30% for MdAPE and $[-20\%, 20\%]$ for MdPE on the natural concentration scale [50]. If we were to apply this to all models, then only the *a priori* PopPK model does not achieve this minimum requirement. For external evaluation, only the *a posteriori* PopPK model achieves the requirement.

ML models are often perceived as 'black-box' models, and, when it comes to ML for the prediction of drug concentrations, it may be difficult to understand how concentration 'X' is predicted, and dose 'Y' is suggested. End-user interpretability is largely determined by the choice of specific ML techniques such as GBT, which can provide insight into the model output ('white-box'). As shown, visualization libraries, like SHAP, may further increase the understanding of end-users and thereby lower the threshold for ML adaptation in clinical practice.

This study is limited by only using plasma concentrations from ICU patients receiving continuous infusion TZP. Therefore, the findings of this study cannot yet be extrapolated to other antimicrobial drugs or alternative modes of infusion instead of continuous infusion. Additionally, the used piperacillin concentrations are total plasma concentrations and not tissue concentrations. Only the unbound drug fraction at the site of infection can exert its antimicrobial effect. Furthermore, tissue perfusion of critically ill patients is unpredictable, therapeutic plasma concentrations may not necessarily predict therapeutic tissue concentrations. However, attaining sufficiently high plasma concentrations is required for achieving therapeutic tissue concentrations [5].

Renal replacement therapy (RRT) patients were not included. Therefore, there was no training on these patients, and the performance will likely be worse when using the model on RRT patients. As a backup, the weighted CG and MDRD formula can be used when the patient is on RRT, however, this solution is not validated. The main challenge of modeling RRT patients is creating a surrogate for the CL_{cr} that can be used as a feature in the model.

The dataset for external model evaluation (UMCG dataset) is small. Therefore, reliably extrapolating these results to other hospitals is not yet possible. As the UMCG patients received lower doses and the ML models assume the GUH dosing scheme, they, therefore, overestimate the external validation concentrations which is visible in the high ME. If we compensate for this bias, i.e. by subtracting the ME from the predictions, the R^2 (log scale) becomes 0.77 (0.56) and the RMSE becomes 23.78 (0.44) for the GBT *prev* model and 0.67 (0.50) and 31.44 (0.52) for the *new* model, further proving this is a bias introduced due to the different dosing as this model then outperforms the PopPK *a posteriori* model. The MdAPE and MdPE of the compensated *prev* GBT model also reduce to 17.92% and -1.55%, making it an acceptable

model by the criteria of Struys *et al.*, [50]. The same conclusion can be made for the *new* GBT model (MdAPE=20.69%, MdPE=0.39%). As a result, the model shows generalization capabilities if adjusting for a different dosing range is possible, however, these bias-compensated models should then be further validated on a new dataset before acceptance. As the external dataset was too small, this was not possible.

The dosing bias also explains the worse uncertainty quantification performance on the UMCG dataset, as the uncertainty quantification method is not robust against this kind of data bias, resulting in a highly negative biased calibration (i.e., the predictive distribution is not wide enough). If we also calculate the calibration error on the compensated results, the ADCE and DCE become 0.13 and 0.11 for the GBT *new* model, and 0.15 and 0.14 for the *prev* model. Interestingly, the positive DCE on the compensated results proves that the predictive distribution is even slightly too wide for the external evaluation, proving the generalization capabilities of the Quantile Ensemble method.

Further work should try to adjust for different dosing regimens. Since the GUH only contained a single dosing scheme, training on different dosings was not possible. This will also enable dosing suggestions when the estimated plasma concentration is not in the therapeutic dosing range.

Lastly, the ML models do not currently take time into account. Therefore, calculating time above the MIC (%fT>MIC), the pharmacokinetic/pharmacodynamic index for beta-lactam antimicrobials, is not entirely possible with the current ML model. However, the models still do indicate the plasma concentration which can be used for initial dose optimization. True dose optimization based on PK/PD target attainment is an area needing further research and can combine the PopPK modeling techniques with the predictive power of ML [5].

Overall, the proposed models demonstrate that this can be considered as an alternative strategy to guide antibiotic therapy, in addition to PopPK methods, by predicting plasma antibiotic concentration while also providing uncertainty estimation. As a result, this opens the path to incorporating machine learning models in decision support systems for more individualized and targeted antibiotic therapy. Furthermore, both the uncertainty framework and the ACDE and DCE metrics can be applied to many more use cases by following the same approach to enable uncertainty quantification and uncertainty evaluation. The presented piperacillin models presented in the paper are based on retrospective analysis. The next step, in future work, is performing a prospective study together with dose suggestion. The final aim is deployment in clinical practice in the intensive care unit by integration it with the electronic health records for real-time concentration predictions.

3.6 Conclusion

Our results show that ML models can consistently estimate piperacillin concentrations with high predictive accuracy, especially when no previous concentration is available, and special emphasis was placed on the interpretability of ML model output using SHAP visualization. Furthermore, the method of generating a predictive distribution using the Quantile Ensemble model can be translated into many other regression problems using any ML model and optimizing a quantile loss function. Additionally, this work also proposed the (Absolute) Distribution Coverage Error, an interpretable uncertainty quantification evaluation metric, usable for any distribution-based uncertainty quantification method.
As such, incorporating ML models in therapeutic drug monitoring programs is definitely promising. Furthermore, these results create a model that is ready to be validated in clinical practice, or at least, in the locally developed hospital.

Declarations

Ethics approval and consent to participate

Ethical approval was obtained from the Ghent University Hospital Ethics Committee (registration number 2016/0264). All methods were performed in accordance with the relevant guidelines and regulations as stated by the Ghent University Hospital Ethics Committee. Informed consent was obtained for all participants and their guardians via opting out before participation.

Availability of data and materials

The data that supports the findings of this study are available from Ghent University Hospital but restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. The data is however available from the authors upon reasonable request and with permission of Ghent University Hospital Ethics Committee. The code supporting the conclusions of this article is available in the predict-idlab/REACT repository, https://github.com/predict-idlab/REACT.

Funding

Part of the research was funded by the FWO Junior Research project HEROI2C that investigates hybrid machine learning for improved infection management in critically ill patients (Ref: 1881020N). Sofie A.M. Dhaese is funded by a Centre of Research Excellence Grant (APP1099452) from the Australian National Health and Medical

Research Council awarded to Jason A. Roberts. Jan J. De Waele is senior clinical investigator funded by the Research Foundation Flanders (FWO, Ref. 1881020N).

Acknowledgements

The authors wish to thank the laboratory technicians of the Laboratory for Toxicology from the GUH for analyzing the piperacillin plasma concentrations.

References

[1] Matthew J. Neidell, Bevin Cohen, Yoko Furuya, Jennifer Hill, Christie Y. Jeon, Sherry Glied, and Elaine L. Larson. *Costs of Healthcare- and Community-Associated Infections With Antimicrobial-Resistant Versus Antimicrobial-Susceptible Organisms*. Clinical Infectious Diseases, 55(6):807–815, September 2012.

[2] Yasser Sakr, Ulrich Jaschinski, Xavier Wittebole, Tamas Szakmany, Jeffrey Lipman, Silvio A Ñamendys Silva, Ignacio Martin-Loeches, Marc Leone, Mary-Nicoleta Lupu, and Jean-Louis Vincent. *Sepsis in Intensive Care Unit Patients: Worldwide Data From the Intensive Care over Nations Audit*. Open Forum Infectious Diseases, 5(12), November 2018.

[3] Jason A. Roberts, Claire Roger, and Jan J. De Waele. *Personalized antibiotic dosing for the critically ill*. Intensive Care Medicine, 45(5):715–718, May 2019.

[4] Jan J. De Waele, Murat Akova, Massimo Antonelli, Rafael Canton, Jean Carlet, Daniel De Backer, George Dimopoulos, José Garnacho-Montero, Jozef Kesecioglu, Jeffrey Lipman, Mervyn Mer, José-Artur Paiva, Mario Poljak, Jason A. Roberts, Jesus Rodriguez Bano, Jean-François Timsit, Jean-Ralph Zahar, and Matteo Bassetti. *Antimicrobial resistance and antibiotic stewardship programs in the ICU: insistence and persistence in the fight against resistance. A position statement from ESICM/ESCMID/WAAAR round table on multi-drug resistance*. Intensive Care Medicine, 44(2):189–196, February 2018.

[5] Jason A. Roberts, Mohd H. Abdul-Aziz, Jeffrey Lipman, Johan W. Mouton, Alexander A. Vinks, Timothy W. Felton, William W. Hope, Andras Farkas, Michael N. Neely, Jerome J. Schentag, George Drusano, Otto R. Frey, Ursula Theuretzbacher, Joseph L. Kuti, and International Society of Anti-Infective Pharmacology and the Pharmacokinetics and Pharmacodynamics Study Group of the European Society of Clinical Microbiology and Infectious Diseases. *Individualised antibiotic dosing for patients who are critically ill: challenges and potential solutions*. The Lancet. Infectious Diseases, 14(6):498–509, June 2014.

[6] T. Tängdén, V. Ramos Martín, T. W. Felton, E. I. Nielsen, S. Marchand, R. J. Brüggemann, J. B. Bulitta, M. Bassetti, U. Theuretzbacher, B. T. Tsuji, D. W. Wareham, L. E. Friberg, J. J. De Waele, V. H. Tam, Jason A. Roberts, and the Pharmacokinetics and Pharmacodynamics Study Group of the European Society of Clinical Microbiology and Infectious Diseases on behalf of the Infection Section for the European Society of Intensive Care Medicine, the International Society of Anti-Infective Pharmacology and the Critically Ill Patients Study Group of European Society of Clinical Microbiology and Infectious Diseases. *The role of infection models and PK/PD modelling for optimising care of critically ill patients with severe infections*. Intensive Care Medicine, 43(7):1021–1032, July 2017.

[7] Alexis Tabah, Jan De Waele, Jeffrey Lipman, Jean Ralph Zahar, Menino Osbert Cotta, Greg Barton, Jean-Francois Timsit, Jason A. Roberts, and on behalf of the Working Group for Antimicrobial Use in the ICU within the Infection Section of the European Society of Intensive Care Medicine (ESICM). *The ADMIN-ICU survey: a survey on antimicrobial dosing and monitoring in ICUs*. Journal of Antimicrobial Chemotherapy, 70(9):2671–2677, September 2015.

[8] Jason A. Roberts, Anand Kumar, and Jeffrey Lipman. *Right Dose, Right Now: Customized Drug Dosing in the Critically Ill*. Critical Care Medicine, 45(2):331–336, February 2017.

[9] Jason A. Roberts, Mohd-Hafiz Abdul-Aziz, Joshua S. Davis, Joel M. Dulhunty, Menino O. Cotta, John Myburgh, Rinaldo Bellomo, and Jeffrey Lipman. *Continuous versus Intermittent beta-Lactam Infusion in Severe Sepsis. A Meta-analysis of Individual Patient Data from Randomized Trials*. American Journal of Respiratory and Critical Care Medicine, 194(6):681–691, September 2016. Publisher: American Thoracic Society - AJRCCM.

[10] Daniel C. Richter, Otto Frey, Anka Röhr, Jason A. Roberts, Andreas Köberer, Thomas Fuchs, Nikolaos Papadimas, Monika Heinzel-Gutenbrunner, Thorsten Brenner, Christoph Lichtenstern, Markus A. Weigand, and Alexander Brinkmann. *Therapeutic drug monitoring-guided continuous infusion of piperacillin/-tazobactam significantly improves pharmacokinetic target attainment in critically ill patients: a retrospective analysis of four years of clinical experience*. Infection, 47(6):1001–1011, December 2019.

[11] Gloria Wong, Alexander Brinkman, Russell J. Benefield, Mieke Carlier, Jan J. De Waele, Najoua El Helali, Otto Frey, Stephan Harbarth, Angela Huttner, Brett McWhinney, Benoit Misset, Federico Pea, Judit Preisenberger, Michael S. Roberts, Thomas A. Robertson, Anka Roehr, Fekade Bruck Sime, Fabio Silvio Taccone, Jacobus P. J. Ungerer, Jeffrey Lipman, and Jason A. Roberts. *An international, multicentre survey of beta-lactam antibiotic therapeutic drug monitoring practice*

in intensive care units. Journal of Antimicrobial Chemotherapy, 69(5):1416–1423, May 2014.

[12] Mieke Carlier, Veronique Stove, Steven C. Wallis, Jan J. De Waele, Alain G. Verstraete, Jeffrey Lipman, and Jason A. Roberts. *Assays for therapeutic drug monitoring of beta-lactam antibiotics: A structured review*. International Journal of Antimicrobial Agents, 46(4):367–375, October 2015.

[13] Catherine M. T. Sherwin, Tony K. L. Kiang, Michael G. Spigarelli, and Mary H. H. Ensom. *Fundamentals of population pharmacokinetic modelling: validation methods*. Clinical Pharmacokinetics, 51(9):573–590, September 2012.

[14] Joao Gonçalves-Pereira and Pedro Póvoa. *Antibiotics in critically ill patients: a systematic review of the pharmacokinetics of beta-lactams*. Critical Care (London, England), 15(5):R206, 2011.

[15] Peter L. Bonate. *Pharmacokinetic-Pharmacodynamic Modeling and Simulation*. Springer US, 2 edition, 2011.

[16] Gloria Wong, Andras Farkas, Rachel Sussman, Gergely Daroczi, William W. Hope, Jeffrey Lipman, and Jason A. Roberts. *Comparison of the accuracy and precision of pharmacokinetic equations to predict free meropenem concentrations in critically ill patients*. Antimicrobial Agents and Chemotherapy, 59(3):1411–1417, March 2015.

[17] Sofie A. M. Dhaese, Andras Farkas, Pieter Colin, Jeffrey Lipman, Veronique Stove, Alain G. Verstraete, Jason A. Roberts, and Jan J. De Waele. *Population pharmacokinetics and evaluation of the predictive performance of pharmacokinetic models in critically ill patients receiving continuous infusion meropenem: a comparison of eight pharmacokinetic models*. The Journal of Antimicrobial Chemotherapy, 74(2):432–441, February 2019.

[18] Pieter J. Colin, Karel Allegaert, Alison H. Thomson, Daan J. Touw, Michael Dolton, Matthijs de Hoog, Jason A. Roberts, Eyob D. Adane, Masato Yamamoto, Dolores Santos-Buelga, Ana Martín-Suarez, Nicolas Simon, Fabio S. Taccone, Yoke-Lin Lo, Emilia Barcia, Michel M. R. F. Struys, and Douglas J. Eleveld. *Vancomycin Pharmacokinetics Throughout Life: Results from a Pooled Population Analysis and Evaluation of Current Dosing Recommendations*. Clinical Pharmacokinetics, 58(6):767–780, June 2019.

[19] Thomas De Corte, Paul Elbers, and Jan De Waele. *The future of antimicrobial dosing in the ICU: an opportunity for data science*. Intensive Care Medicine, 47(12):1481–1483, December 2021.

[20] The Lancet. *Artificial intelligence in health care: within touching distance*. The Lancet, 390(10114):2739, December 2017. Publisher: Elsevier.

[21] Anne Kümmel, Peter L. Bonate, Jasper Dingemanse, and Andreas Krause. *Confidence and Prediction Intervals for Pharmacometric Models*. CPT: pharmacometrics & systems pharmacology, 7(6):360–373, June 2018.

[22] Jie Tang, Rong Liu, Yue-Li Zhang, Mou-Ze Liu, Yong-Fang Hu, Ming-Jie Shao, Li-Jun Zhu, Hua-Wen Xin, Gui-Wen Feng, Wen-Jun Shang, Xiang-Guang Meng, Li-Rong Zhang, Ying-Zi Ming, and Wei Zhang. *Application of Machine-Learning Models to Predict Tacrolimus Stable Dose in Renal Transplant Recipients*. Scientific Reports, 7, February 2017.

[23] Rong Liu, Xi Li, Wei Zhang, and Hong-Hao Zhou. *Comparison of Nine Statistical Model Based Warfarin Pharmacogenetic Dosing Algorithms Using the Racially Diverse International Warfarin Pharmacogenetic Consortium Cohort Database*. PLOS ONE, 10(8):e0135784, August 2015. Publisher: Public Library of Science.

[24] M. R. Poynton, B. M. Choi, Y. M. Kim, I. S. Park, G. J. Noh, S. O. Hong, Y. K. Boo, and S. H. Kang. *Machine learning methods applied to pharmacokinetic modelling of remifentanil in healthy volunteers: a multi-method comparison*. The Journal of International Medical Research, 37(6):1680–1691, December 2009.

[25] Wei Guo, Ze Yu, Ya Gao, Xiaoqian Lan, Yannan Zang, Peng Yu, Zeyuan Wang, Wenzhuo Sun, Xin Hao, and Fei Gao. *A Machine Learning Model to Predict Risperidone Active Moiety Concentration Based on Initial Therapeutic Drug Monitoring*. Frontiers in Psychiatry, 12:711868, November 2021.

[26] Danish Shakeel and Shakeel Ahmad Mir. *Personalized drug concentration predictions with machine learning: an exploratory study*. International Journal of Basic & Clinical Pharmacology, 9(6):980, May 2020.

[27] Xiaolan Mo, Xiujuan Chen, Xianggui Wang, Xiaoli Zhong, Huiying Liang, Yuanyi Wei, Houliang Deng, Rong Hu, Tao Zhang, Yilu Chen, Xia Gao, Min Huang, and Jiali Li. *Prediction of Tacrolimus Dose/Weight-Adjusted Trough Concentration in Pediatric Refractory Nephrotic Syndrome: A Machine Learning Approach*. Pharmacogenomics and Personalized Medicine, 15:143–155, February 2022.

[28] Pan Ma, Ruixiang Liu, Wenrui Gu, Qing Dai, Yu Gan, Jing Cen, Shenglan Shang, Fang Liu, and Yongchuan Chen. *Construction and Interpretation of Prediction Model of Teicoplanin Trough Concentration via Machine Learning*. Frontiers in Medicine, 9:808969, March 2022.

[29] Michael E. Brier, Jacek M. Zurada, and George R. Aronoff. *Neural Network Predicted Peak and Trough Gentamicin Concentrations*. Pharmaceutical Research, 12(3):406–412, March 1995.

[30] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. arXiv:1705.07874 [cs, stat], November 2017. arXiv: 1705.07874.

[31] Sooyoung Lee, Moonsik Song, Jongdae Han, Donghwan Lee, and Bo-Hyung Kim. *Application of Machine Learning Classification to Improve the Performance of Vancomycin Therapeutic Drug Monitoring.* Pharmaceutics, 14(5):1023, May 2022. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[32] Tim Pearce, Alexandra Brintrup, Mohamed Zaki, and Andy Neely. *High-Quality Prediction Intervals for Deep Learning: A Distribution-Free, Ensembled Approach.* In International Conference on Machine Learning, pages 4075–4084. PMLR, July 2018. ISSN: 2640-3498.

[33] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. *Individual Calibration with Randomized Forecasting.* arXiv:2006.10288 [cs, stat], September 2020. arXiv: 2006.10288.

[34] J. L. Vincent, R. Moreno, J. Takala, S. Willatts, A. De Mendonça, H. Bruining, C. K. Reinhart, P. M. Suter, and L. G. Thijs. *The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure. On behalf of the Working Group on Sepsis-Related Problems of the European Society of Intensive Care Medicine.* Intensive Care Medicine, 22(7):707–710, July 1996.

[35] Andrew S. Levey, Lesley A. Stevens, Christopher H. Schmid, Yaping (Lucy) Zhang, Alejandro F. Castro, Harold I. Feldman, John W. Kusek, Paul Eggers, Frederick Van Lente, Tom Greene, and Josef Coresh. *A New Equation to Estimate Glomerular Filtration Rate.* Annals of internal medicine, 150(9):604–612, May 2009.

[36] Mieke Carlier, Veronique Stove, Jan J. De Waele, and Alain G. Verstraete. *Ultrafast quantification of beta-lactam antibiotics in human plasma using UPLC-MS/MS*. Journal of Chromatography. B, Analytical Technologies in the Biomedical and Life Sciences, 978-979:89–94, January 2015.

[37] Heleen Aardema, Prashant Nannan Panday, Mireille Wessels, Kay van Hateren, Willem Dieperink, Jos G. W. Kosterink, Jan-Willem Alffenaar, and Jan G. Zijlstra. *Target attainment with continuous dosing of piperacillin/tazobactam in critical illness: a prospective observational study.* International Journal of Antimicrobial Agents, 50(1):68–73, July 2017.

[38] Jason Osborne. *Best practices in data cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data.* January 2013. Journal Abbreviation: Thousand Oaks, CA: Sage Publications Publication Title: Thousand Oaks, CA: Sage Publications.

[39] Andrew S. Levey, Josef Coresh, Tom Greene, Lesley A. Stevens, Yaping Lucy Zhang, Stephen Hendriksen, John W. Kusek, Frederick Van Lente, and Chronic Kidney Disease Epidemiology Collaboration. *Using standardized serum creatinine*

values in the modification of diet in renal disease study equation for estimating glomerular filtration rate. Annals of Internal Medicine, 145(4):247–254, August 2006.

[40] D. W. Cockcroft and M. H. Gault. *Prediction of creatinine clearance from serum creatinine*. Nephron, 16(1):31–41, 1976.

[41] Jarne Verhaeghe, Jeroen Van Der Donckt, Femke Ongenae, and Sofie Van Hoecke. *Powershap: A Power-full Shapley Feature Selection Method*, 2022.

[42] Danilo Bzdok, Naomi Altman, and Martin Krzywinski. *Statistics versus machine learning*. Nature Methods, 15(4):233–234, April 2018. Number: 4 Publisher: Nature Publishing Group.

[43] S. a. M. Dhaese, P. Colin, H. Willems, A. Heffernan, B. Gadeyne, S. Van Vooren, P. Depuydt, E. Hoste, V. Stove, A. G. Verstraete, J. Lipman, J. A. Roberts, and J. J. De Waele. *Saturable elimination of piperacillin in critically ill patients: implications for continuous infusion*. International Journal of Antimicrobial Agents, 54(6):741–749, December 2019.

[44] Tom C. Zwart, Dirk Jan A. R. Moes, Paul J. M. van der Boog, Nielka P. van Erp, Johan W. de Fijter, Henk-Jan Guchelaar, Ron J. Keizer, and Rob Ter Heine. *Model-Informed Precision Dosing of Everolimus: External Validation in Adult Renal Transplant Recipients*. Clinical Pharmacokinetics, 60(2):191–203, February 2021.

[45] Pieter J. Colin, Douglas J. Eleveld, Andrew Hart, and Alison H. Thomson. *Do Vancomycin Pharmacokinetics Differ Between Obese and Non-obese Patients? Comparison of a General-Purpose and Four Obesity-Specific Pharmacokinetic Models*. Therapeutic Drug Monitoring, 43(1):126–130, February 2021.

[46] A. Broeker, M. Nardecchia, K. P. Klinker, H. Derendorf, R. O. Day, D. J. Marriott, J. E. Carland, S. L. Stocker, and S. G. Wicha. *Towards precision dosing of vancomycin: a systematic evaluation of pharmacometric models for Bayesian forecasting*. Clinical Microbiology and Infection: The Official Publication of the European Society of Clinical Microbiology and Infectious Diseases, 25(10):1286.e1–1286.e7, October 2019.

[47] *EUCAST: EUCAST*.

[48] Marie-Charlotte Quinton, Sandra Bodeau, Loay Kontar, Yoann Zerbib, Julien Maizel, Michel Slama, Kamel Masmoudi, Anne-Sophie Lemaire-Hurtel, and Youssef Bennis. *Neurotoxic Concentration of Piperacillin during Continuous Infusion in Critically Ill Patients*. Antimicrobial Agents and Chemotherapy, 61(9), September 2017.

[49] Yoshiro Hayashi, Jason A. Roberts, David L. Paterson, and Jeffrey Lipman. *Pharmacokinetic evaluation of piperacillin-tazobactam*. Expert Opinion on Drug Metabolism & Toxicology, 6(8):1017–1031, August 2010.

[50] Michel Struys, Anthony Absalom, and Steven L. Shafer. *Intravenous Drug Delivery Devices*. In Miller's Anesthesia. Elsevier, 9 edition, 2019.

4

Designing a Pharmacokinetic Machine Learning Model for Optimizing Beta-Lactam Antimicrobial Dosing in Critically Ill Patients

The models presented in Chapter 3 focus on predicting antibiotic concentrations for a single antimicrobial and are not fully equipped to provide treatment dosing suggestions. This chapter builds upon that work by introducing a new model trained on new data to predict the Pharmacokinetic Constant, inspired by domain knowledge. This constant quantifies the drug's clearance rate in the patient, rather than absolute concentrations, enabling the calculation of the required dose to achieve a specific blood plasma concentration for optimized treatment. Furthermore, the model predicts this constant for multiple beta-lactam antimicrobials, Piperacillin and Meropenem, providing a single model for various antimicrobials. Thus, this chapter enhances the ideas presented in Chapter 3 by incorporating treatment dosing advice. As such, it is a further elaboration on addressing UCG2. My contributions to this chapter are the development and validation of all the machine learning models, the study design, the experiments, the implementation of the Pharmacokinetic Constant, and the writing of the chapter.

Designing a Pharmacokinetic Machine Learning Model for Optimizing Beta-Lactam Antimicrobial Dosing in Critically Ill Patients

Jarne Verhaeghe, Thomas De Corte, Jan J. De Waele, Femke Ongenae, Sofie Van Hoecke

Published in the Proceedings of the 2024 8th International Conference on Medical and Health Informatics

Abstract In the intensive care unit (ICU), accurate dosing of antimicrobials is crucial to not only benefit the patient but also society by reducing the burden of antimicrobial resistance, considered a major global health threat. Piperacillin-tazobactam (TZP) and meropenem (MEM), two beta-lactam antimicrobials, are frequently prescribed in ICUs worldwide. Current dosing of these antimicrobials has been shown to not reach therapeutic plasma concentrations in many patients. Furthermore, it is believed that appropriate antimicrobial dosing decreases patient morbidity, decreases mortality, and avoids drug toxicity. Therapeutic drug monitoring guided dosing is advocated to optimize the usage of these antimicrobials. However, adoption in clinical practice is limited due to its time-consuming and costly nature. We, therefore, propose a hybrid machine-learning model that incorporates pharmacokinetics into a Catboost regressor model for personalized dosing of TZP and MEM in ICU patients. The resulting model is a single model for both antimicrobials that learns the patient's drug clearance rate for determining their plasma concentrations and suggesting doses. The model was trained and temporally validated on two different datasets from different periods at the Ghent University Hospital ICU each with/without having access to a previous plasma concentration. The model reached R^2 test scores of $0.869/0.807$ on the internal dataset and $0.727/0.752$ on the time validation dataset.

This work established the feasibility of predicting plasma concentrations for an antibiotic group enabling personalized antimicrobial therapy in critically ill patients using machine learning. As such this work contributes to facilitating truly personalized antimicrobial therapy for critically ill patients.

4.1 Introduction

Appropriate dosing of antimicrobials in the intensive care unit (ICU) is a crucial part of treating infections. On the one hand, for an individual patient, it is believed that appropriate antimicrobial dosing decreases patient morbidity, decreases mortality, and avoids drug toxicity [1, 2]. On the other hand, antimicrobial resistance is considered one of the biggest threats to global health according to the World Health Organiza-

tion [3]. As appropriate dosing is believed to reduce antimicrobial resistance it is not only crucial for the individual patient, but also for society [3]. Historically, dosing regimens of antimicrobials have been developed based on data from healthy volunteers. In recent years, however, a substantial body of evidence has emerged that shows that ICU patients have an altered physiology compared to non-critically ill patients. This altered physiology impacts the way their bodies absorb, distribute and eliminate antimicrobials [4]. For piperacillin (TZP) and meropenem (MEM), two frequently prescribed antimicrobials from the beta-lactam family, research has shown that current dosing regimens achieve inadequate antimicrobial plasma concentrations in a substantial proportion of ICU patients [5, 6]. Therapeutic drug monitoring (TDM) guided dosing is advocated as a way to optimize the usage of these antimicrobials [7]. In TDM, the adjustment of the dosing regimen is based on the measured concentration of the antimicrobial in the plasma of the patient. However, adoption of TDM in clinical practice is limited [8]. Often cited reasons for this low TDM adoption rate are unavailable means, unavailable expertise, and a too-long time interval between sampling and availability of results to allow for timely dose adjustments [8].

Instead of measuring the antimicrobial concentration for personalized antimicrobial dosing, one can also try to predict the concentration. Historically, population pharmacokinetic models (PopPK), which are statistical mixed-effect models, have been used for this purpose. However, these models are frequently based on small non-representative datasets resulting in generalizability issues [9, 10]. In previous work, we showed that machine learning (ML) models are a suitable alternative to these PopPK models for predicting plasma concentrations, specifically for TZP plasma concentrations [9]. However, by only focusing on predicting the concentration, the model is less suited for giving dosing recommendations. Published models for other drug dosing use cases similarly only focus on a single antimicrobial and on predicting the concentration without providing dosing advice [11–16].

In this work, we expand upon our previous work by presenting a new model that is capable of not only predicting the antimicrobial concentration for TZP but also for MEM. Furthermore, the new model is developed such that dosing advice for both TZP and MEM can be proposed at the same time. As a novelty to previous work, the proposed model incorporates pharmacokinetic principles to quantify the drug clearance of a patient. Once this drug clearance rate is identified, proposing dosing advice becomes straightforward [17].

The remainder of this paper is as follows. In the next section, we discuss related work and provide some background knowledge on pharmacokinetics. The method section explains the used data, the data preprocessing, the model development, and the evaluation strategy. The results section then presents the validation results of two models. One model has access to the previous concentration, the *a posteriori* model, to mimic complementing TDM with ML models, and the other model has no access to it, the *a priori* model, to mimic the clinical scenario where no TDM is available.

The results section also presents a small example to demonstrate how the model can provide dosing adjustments. A discussion and conclusion finish up the paper.

4.2 Background and related work

4.2.1 Background on pharmacokinetics

Pharmacokinetics (PK) is the study of how the body interacts with administered substances [18]. PK tries to quantify the uptake of a drug, its distribution throughout the body, and its elimination. The PK of drugs are generally described using compartment models characterized by differential equations [17]. A single compartment mimics a part of the body that has the same PK properties. These can be derived into analytical formulas which are sometimes easier to model [17]. A 1-compartment continuous intravenous (IV) infusion model describes the concentration $C(t)$ in the plasma of the patient at time t by [17]:

$$C(t) = \frac{D}{kV}(1 - e^{-kt} + Ae^{-kt}) \tag{4.1}$$

with D the drug infusion rate, k the elimination rate constant of the patient, V the volume of distribution of the drug in the patient, and A the previous concentration in that compartment. If you take a large t, denoted as t_{SS}, or take the limit to $t = \infty$, you get the following simplification of that formula:

$$C(t_{SS}) = \frac{D}{kV} \tag{4.2}$$

Higher compartment models have the same simplification for large t_{SS}, so this is considered the steady-state formula for IV infusions. When t is small, it has not reached the steady-state yet and is then considered as being in a non-steady state. However, although the infusion rate D is often known, both the clearance rate k and the volume of distribution V are unknown for a specific patient. We can define these unknowns into a new constant, the $PK\ constant$, or sometimes also defined as the C/D (Concentration/Dose) ratio [19]:

$$PK_C = \frac{1}{kV} = \frac{C(t_{SS})}{D} \tag{4.3}$$

This PK_C constant represents the ratio between the clearance and distribution of a drug in a patient. The absorption is not taken into account here, as the drug is directly administered in the bloodstream. Beware that both k and V can change over time as these parameters represent biological states, which can vary, especially in critically ill patients. Although every drug has its own PK behaviour, drugs within the same drug class, e.g. beta-lactam antimicrobials, mostly behave the same in terms of PK [20].

4.2.2 Related Work

Earlier works, including ours, already explored the idea of using ML models, such as support vector machines, gradient boosting trees, XGBoost, and neural networks, to predict drug concentrations for various drugs and antimicrobials. These works have verified that predicting drug and antimicrobial concentration with machine learning is possible although challenging for tacrolimus, piperacillin, gentamicin, remifentanil, lamotrigine, phenytoin, risperidone, warfarin, and teicoplanin [9, 11–16, 19, 21–23]. These models are often limited to only predicting concentrations of a single antimicrobial instead of providing dosing recommendations, including our previous work [9]. Some works go further than simply applying machine learning to the antibiotic concentration prediction problem and try to combine PK with machine learning. Janssen et al. proposed deep compartment models that combine PK compartment modelling with neural networks [23]. Zhu et al. tried to predict the PK constant for lamotrigine drug dosing. However, both studies had issues with large relative errors, limited generalizability, and/or focused on a single drug [19]. Therefore, combining multiple antibiotics into a single model while providing dosing recommendations has been unsolved to this day.

4.3 Methods

4.3.1 Data

Two different datasets are used in this paper. The first dataset, which we consider the *internal* dataset, will be used for both training and testing the model. The second dataset, which we denote as the *time validation* dataset, will solely be used for testing purposes to evaluate the generalizability of the model. Consequently, there is an internal and external validation possible for the model. Each dataset is converted into an *a priori* and an *a posteriori* dataset. The *a priori* dataset mimics a situation where a previous measurement of the antimicrobial plasma concentration is not available. The *a posteriori* dataset then mimics a situation where this is available, e.g. by performing TDM, and therefore only includes data samples where a previous concentration is available.

4.3.1.1 Internal Dataset

For the internal dataset, patients admitted to the surgical ICU of Ghent University Hospital between March 2016 and April 2019, receiving either piperacillin-tazobactam (4 g/0.5 g powder for solution for infusion; Fresenius Kabi n.v., Schelle, Belgium) or meropenem (0.5 g or 1 g powder for solution for infusion; Fresenius Kabi n.v., Schelle, Belgium) in a continuous infusion, in need of routine blood sampling as TDM requires blood sampling, and above the age of 18 years old were included. Initial dos-

ing regimens and any subsequent dosing modifications were based on the creatinine clearance (ClCr), calculated from creatinine measurements on a once-daily 8h urinary collection and a plasma sample. The TZP dosing was as follows: a loading dose of 4/0.5g/30min, immediately followed by a continuous TZP infusion depending on the ClCr: ClCr $<$ 15 mL/min: 8/1g/24 h, ClCr $\geq$ 15 and $<$ 30 mL/min: 12/1.5g/24 h and for a ClCr $\geq$ 30 mL/min 16/2g/24h. The MEM dosing was as follows: a loading dose of 1g/30min, immediately followed by a continuous MEM infusion depending on the ClCr: ClCr $<$ 15 mL/min: 1g/24 h, ClCr $\geq$ 15 and $<$ 30 mL/min: 2g/24 h and for a ClCr $\geq$ 30 mL/min 3g/24h.

Any additional data, such as biochemistry and demographic data were gathered on the same and the previous day of blood plasma sampling. Biochemical variables, such as serum creatinine, albumin, platelets, lactate, white blood cells, and bilirubin were determined from the same drawn blood sample as the antibiotic plasma concentration. When no ClCr was available, it was substituted using $ClCr = (CG + 2 * MDRD)/3$ with the Cockcroft-Gault (CG) formula [24] and the MDRD formula [25] in the dataset in accordance to previous work [9]. Other missing data was not interpolated and left as is.

For TZP, a protein binding of 30% was assumed, for which the measured concentration was corrected [26]. For MEM, the influence of protein binding was considered to be negligible [20]. Samples taken within a time frame of 12h after the start of therapy or a dose change were considered non-steady state samples, following Section 4.2.1, while the other samples were considered to be in a steady state.

A total of 795 unique patients were in the dataset resulting in 2396 plasma concentration measurements (1668 TZP, 728 MEM). Of these, 82 patients received both TZP and MEM during the same ICU stay. A total of 452 samples were taken within a time frame of 12h after a dose change or the start of therapy and hence were considered to be non-steady state samples (329 TZP, 123 MEM). The exclusion of non-steady state samples and other exclusion criteria, mentioned in Section 4.3.2.1, resulted in 630 patients and 1813 samples (1216 TZP, 597 MEM). The final distribution of the plasma concentrations is shown in Figure 4.1.

4.3.1.2 Time validation dataset

The time validation dataset comprises 149 patients admitted between May 2019 and July 2021 to the same surgical ICU of Ghent University Hospital, receiving the same antimicrobials and initial dosing regimen as in the internal dataset in continuous infusion, in need of routine blood sampling, and above the age of 18 years. The time validation dataset comprises 149 unique ICU admissions with 13 patients having received both TZP and MEM during their ICU stay. The complete dataset contains 313 measurements (170 TZP, 143 MEM) of which 68 are in non-steady state (33 TZP, 35 MEM). The same biochemistry and demographic variables were collected as in the internal dataset. The preprocessing of all variables was carried out analogous

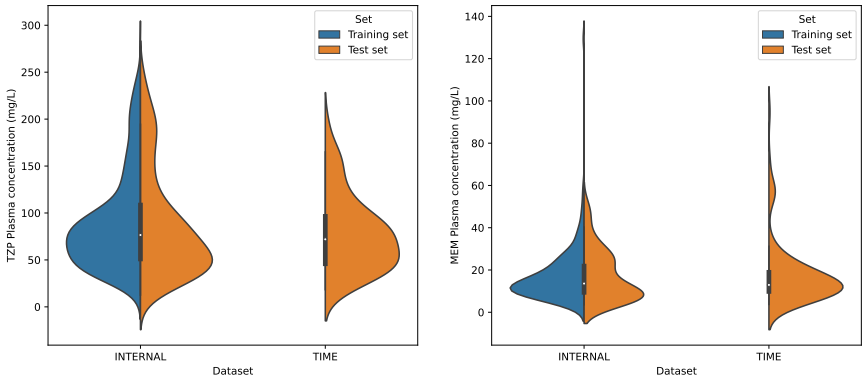


Figure 4.1: Violin plot distribution of the TZP and MEM plasma concentrations for both the internal and Time validation datasets. The white dot represents the mean, the black bar represents the Q1 and Q3 quartiles.

to the internal dataset. Six samples corresponded with having received less than 0.5g over 24h and were deemed anomalous and removed. The exclusion of non-steady state samples and other exclusion criteria, mentioned in Section 4.3.2.1, resulted in 184 samples (82 TZP, 102 MEM). The distribution of the plasma concentrations is shown in Figure 4.1.

The main difference with the internal dataset is the timing of the data collection and the reason for plasma concentration determination. The data from the internal dataset was collected for study purposes and the measured antimicrobial concentrations were not disclosed to the treating physicians. On the contrary, the time validation dataset encompasses real-world data of plasma concentrations that were determined at the request of the treating physicians to evaluate the dosing adequacy in patients who were suspected to be at risk for inadequate dosing. Furthermore, part of the time validation dataset was collected during the COVID-19 pandemic which had a large impact on therapy in the ICU resulting in a different patient population.

4.3.2 Machine Learning Models

4.3.2.1 Data preprocessing

The internal dataset was split into a training set and a test set using an 80/20 patient split, splitting the dataset on the patient level instead of the concentration level to avoid data leakage. To perform the split, the number of concentration measurements per patient was counted for both antimicrobials and added together. If the count exceeded 7 for TZP or 5 for MEM, they were rounded down to 7 or 5 respectively to enable stratification. The split was then stratified on both the number of measurements and the distribution of antimicrobials. As a result, the test set has approximately the same distribution of the number of TZP and MEM measurements per patient as the training

set. The time validation set was only used for testing purposes.

Only steady-state samples were used for training and evaluating the models as the relation between the concentration and the dose has not stabilized yet in non-steady-state samples, which would introduce noise. Specific measurements were excluded as these were considered outliers in consultation with physicians. TZP concentrations below 3mg/L and above 250mg/L were excluded as the number of samples in these ranges was too sparse for learning (for $> 250mg/L$) or the relation between the dose, the concentration, and the clearance was deemed unlikely. MEM concentrations below 3mg/L were also excluded as all three samples were deemed outliers by physicians. For the *a priori* training dataset, TZP samples that have a previous measurement were excluded if their concentration did not differ by more than 10mg/L compared to their previous measurement. This was to ensure that the model would not learn any duplicates that could introduce more noise. The same threshold was applied to MEM measurements, however with a 5mg/L difference threshold instead. This was not done for the test set. By default, a previous concentration of 0 mg/L was used in every first record of a patient to indicate no available previous concentration. Additionally, if the previous concentration was collected more than 48 hours before the current measurement, the previous concentration was also set to 0 mg/L to avoid over-reliance on the previous concentration as the patient's situation can change drastically in 48 hours. This was suggested by physicians and experimentally validated using 5-fold cross-validation using the same validation sets for the cases where the previous concentration was set to 0 as mentioned previously and where it was not. Samples that did not have a previous concentration (or previous concentration = 0 mg/L) were automatically included in the *a priori* dataset, as the a priori model dataset does not take this variable into account. This resulted in an *a priori* training set of 910 samples (651 TZP, 259 MEM) and an *a priori* test set of 121 samples (82 TZP, 39 MEM). The *a posteriori* dataset only includes samples where a previous concentration measurement was available. This resulted in an *a posteriori* training set of 952 samples (626 TZP, 326 MEM) and an *a posteriori* test set of 213 samples (173 TZP, 40 MEM).

All features were extracted from data gathered up to the moment of antimicrobial concentration sampling to avoid data leakage. The lab variables' features consisted of the most recent values collected within 24 hours preceding antimicrobial concentration sampling, including the values analyzed on the same sample on which the antimicrobial concentration was determined. Medication features include the mean, min, max, and sum of the dose of the respective medication up to 12 hours before sampling. The same aggregates, together with the standard deviation and the slope, were also taken from continuous variables, such as heart rate and temperature, from a data window up to 12 hours before sampling of the concentration. This resulted in 926 possible features in total.

As the true goal of the model is to allow for dose optimization, the prediction target was set to the PK constant PK_C. If the model can truly predict PK_C, one

can easily convert any concentration to a specific infusion rate using $\frac{C_{target}}{PK_C} = D$. However, for numerical purposes, the machine learning model was trained on the natural logarithm (ln) of the PK constant (in h/L) as this reduces skewness:

$$PK_C = ln\left(\frac{C}{D}\right) \tag{4.4}$$

4.3.2.2 Model Development

The used model is a gradient boosting tree model, namely a Catboost regressor, following previous work as it has proven ideal for the antibiotic concentration use case [9, 27]. The model was trained with the Mean Absolute Error (MAE) loss function. Furthermore, the given antibiotic was set as a categorical feature for the CatBoost regressor, which has inherent support for categorical features. Feature selection was performed using Powershap, a feature selection method that utilizes Shapley values, statistical power and significance tests to automatically determine informative features by testing each feature to a random known uninformative feature [28]. An initial set of features was selected using a single execution of Powershap. This feature set was then used in five-fold cross-validation to optimise hyperparameters. Afterwards, Powershap was again executed utilizing the new Catboost model with the new hyperparameters in Powershap convergence mode and convergence limitations set to 1. The resulting feature set was then again used for a last Powershap execution, with convergence mode set to off, to reduce the size of the feature set.

4.4 Results

4.4.1 Final models

The final *a priori* model used 300 iterations while the *a posteriori* model used 400 iterations, determined in the hyperparameter optimization. Other hyperparameters were kept at their default values. Both models share the following features: [*ClCr, AB, Creatinine serum previous day, Creatinine serum, Bilirubin serum, Protein total serum, pH arterial, Length, Age, Mean central temperature, Urine creatinine, Platelets, Urine Urea Nitrogen, C-reactive protein, Gender, Creatinine Kinase serum, Urea Nitrogen serum, Albumin serum, Base excess arterial, Lactate arterial, Weight, aPTT, SAO2 all std, INR, and Lipase serum*]. The additional *a priori* specific features are: [*IBP Systolic sum, Calcium serum, Urine volume sum, and Magnesium serum*]. The additional *a posteriori* specific features are: [*previous concentration, Sodium urine, Difference ClCr previous day, GCS ER sum, Fibrinogen, Chloride serum, Phosphate serum, GGT serum, GCS MR sum, and IBP Diastolic mean*]. As can be seen, there is a large overlap in features for both situations where the selected features follow the literature and previous work [9]. Excluding the previous concentration and difference ClCr previous day features of the *a posteriori* model, the model-specific features

Evaluation	Model	Target	AB	Samples	RMSE	MAE	R^2	ME	MdAPE
Internal	Priori	PK_C	COMBO	334	0.296	0.231	0.807	0.004	8.74
		PK_C	TZP	255	0.294	0.231	0.799	-0.010	8.64
		C			27.8	19.7	0.756	-7.60	19.0
		PK_C	MEM	79	0.305	0.232	0.826	0.049	9.60
		C			5.51	3.69	0.786	-0.704	19.4
	Posteriori	PK_C	COMBO	213	0.249	0.188	0.869	0.021	7.47
		PK_C	TZP	173	0.247	0.182	0.869	0.031	6.91
		C			20.5	14.2	0.880	-2.15	13.9
		PK_C	MEM	40	0.254	0.217	0.870	-0.023	9.50
		C			5.09	3.70	0.795	-1.73	21.0
Time	Priori	PK_C	COMBO	135	0.318	0.242	0.752	0.110	8.67
		PK_C	TZP	68	0.247	0.195	0.796	0.132	7.76
		C			16.3	13.2	0.831	7.32	17.4
		PK_C	MEM	67	0.377	0.290	0.726	0.087	11.1
		C			6.86	4.53	0.711	-0.187	21.0
	Posteriori	PK_C	COMBO	49	0.323	0.247	0.727	-0.043	7.86
		PK_C	TZP	14	0.274	0.191	0.613	0.031	6.67
		C			27.17	16.8	0.532	0.044	15.7
		PK_C	MEM	35	0.340	0.270	0.746	-0.073	9.27
		C			11.0	6.52	0.630	-4.25	20.0

Table 4.1: Test results of both the posteriori and priori model on the internal and the time validation set. PK_C = PK constant (h/L), C = Plasma concentration (mg/L), TZP = piperacillin/tazobactam, MEM = Meropenem, MdAPE = Median Absolute Percentage Error, AB = antimicrobial, RMSE = Root Mean Square Error, MAE = Mean Absolute Error.

only contribute marginally to the prediction (10.63% contribution for the *a posteriori model* and 7.18% for the *a priori model*). The difference ClCr is only relevant when comparing the previous concentration to the current concentration to determine the change between the previous concentration and is, therefore, less relevant for the *a priori* model.

4.4.2 Validation results

The test results on the internal and the time validation test set are shown in Table 4.1. The results are shown in both the log scale (PK_C) and the concentration (C) scale for interpretation, both scales are the performances of the same model. The results are also split per antibiotic (TZP or MEM) or together (COMBO) to understand the performance for each antibiotic separately and globally. For the internal testing, the models achieve high R^2 while still reaching Median Average Percentage Errors (MdAPE) of around 20% which is considered the state-of-the-art threshold [29]. Interestingly the performance does not change much between antimicrobials. As expected, the results of the *a posteriori* model are considerably better compared to the *a priori* model, however, the *a priori* model still achieves acceptable results. Compared to the results of the internal test set, there is a significant decrease in performance when comparing the R^2 scores for the time validation set showing a larger error variance and a less ideal fit. However, the $MdAPE$ scores are better for TZP or in the same range for MEM as the internal results indicating that the median error of the model is comparable. The *a posteriori* scores for the time validation set are worse, which could be explained by over-reliance on the previous concentration instead of other patient parameters, where the previous concentration has a different distribution compared to the internal dataset because of the different measurement biases.

4.4.3 Dosing Case Study

Consider a test patient with a creatinine clearance of 166 ml/min, which is considered augmented renal clearance [30]. The patient received 664mg/h or around 16g/24h TZP and the actual measured concentration was 35.4mg/L. This would be below a worst-case scenario target, which would be 4 times the 16mg/L TZP minimum inhibitory concentration (MIC) of *Pseudomonas Aeruginosa* [31], resulting in a target of 64mg/L. Compensating for a 30% protein binding would result in a target of 92.42 mg/L. The predicted $ln(PK_C)$ is -2.84 which equals to a $PK_C = 0.05864\ h/L$. The predicted concentration is 38.89mg/L given a TZP dose of 16g/24h. To reach the target of 92.42 mg/L would require a dose of 37.87g/24h, which is considerably higher than current standard dosing regimens. For comparison, we can also estimate the PK_C for MEM resulting in $PK_C = 0.05362\ h/L$, which is a 9% difference in clearance. The *Pseudomonas Aeruginosa* MIC for MEM is 2mg/L [31], resulting in a target of 8mg/L or the required dose of 149mg/h or 3.58g/24h. This small exam-

ple shows the potential of the model for dose adjustment and the flexibility of using the PK_C instead of the concentration in compensating for different scenarios and different antibiotics in contrast to previous work.

4.5 Discussion

In this paper, we presented two models that provide beta-lactam plasma concentration predictions for critically ill patients in the ICU, one when a previous concentration is available and one when this is not available, respectively the *a posteriori* and *a priori* models. Furthermore, the models are trained to predict the PK constant of a patient, which allows for individualized dosing advice as illustrated in the dosing case study. The validation results on the internal dataset show acceptable error margins for both the *a priori* and *a posteriori* models, regardless of the antimicrobial. Interestingly, the performance on the time validation set is also within the acceptable error margins for TZP concentrations and close for the MEM concentrations even though the time validation includes a different patient population. The time validation dataset represents patients that were believed to be at risk for inadequate dosing and the measured concentration was used to perform dosing adjustments. Furthermore, this dataset was partially gathered during the COVID-19 pandemic, which had a significant influence on the ICU and its patient population resulting in a larger patient distribution shift. Ultimately, we can establish that these models can complement TDM in clinical practice or provide concentrations in situations where TDM is not available to provide real-time dose optimization at a time when adequate antimicrobial dosing is crucial.

Both TZP and MEM are beta-lactam antimicrobials that have comparable pharmacokinetic behaviour in terms of clearance [20]. Although some differences, such as a different protein binding, can affect clearance rates, the clearance mechanisms are comparable [31]. The shared clearance mechanism is also represented by the approximately equal performances of both models for both antimicrobials in our dataset. A SHAP-analysis [32] on the antibiotic feature reveals an average difference between MEM and TZP of 0.0779 in the log scale which corresponds to a factor of 1.081 to PK_C, which would be an 8% difference on average. For example, a PK_C of $0.05\ h/L$ for MEM then corresponds to a PK constant of $0.050 * 1.081 = 0.05405$ h/L for TZP. Although there is a difference in the PK_C between the two antimicrobials, this does not imply that the order of magnitude of dosing adaptations would significantly differ. If for example, a change of 25% in the predicted PK_C of TZP would have led to a dosing adjustment of TZP and the physician wants to switch to meropenem, the order of magnitude with which meropenem would need to be adjusted would be comparable. Consequently, the model indicates how fast beta-lactams will be cleared in the patient and therefore provides a prediction for more personalized therapy with beta-lactams and not specifically for a single antimicrobial.

Future work would include adding explainability methods and uncertainty quan-

tification to make the model ready for integration into decision support systems for clinical use in the ICU. Furthermore, the current results are only validated on retrospective datasets and therefore do not mimic a real clinical situation. Therefore, future work should validate the models in vivo, for example by running in the background and prospectively providing predictions in real-time. The current data is also limited to only two beta-lactams and to critically ill patients. Incorporating and evaluating more beta-lactams could provide more insights into the generalizability of the models across multiple beta-lactam concentrations to potentially create a single model for all beta-lactam antimicrobials. Including data from multiple ICUs with potentially multiple dosing schemes and different standards can also increase the generalizability of the model and increase its potential value for healthcare.

4.6 Conclusion

We presented a model for multiple antibiotic plasma concentration prediction and dosing recommendation. Our results revealed that a single model can predict plasma concentrations with acceptable errors for both TZP and MEM. Consequently, machine learning models for antimicrobial plasma concentration predictions could expand from single antimicrobial to models for complete antimicrobial classes instead. Furthermore, if steady-state concentrations are available, the machine learning model can even learn to predict the PK constant instead of the concentration to facilitate personalized antimicrobial dosing for critically ill patients in the ICU. This work therefore has the potential to facilitate truly personalized antimicrobial therapy for critically ill patients.

References

[1] Jason A. Roberts, Claire Roger, and Jan J. De Waele. *Personalized antibiotic dosing for the critically ill.* Intensive Care Medicine, 45(5):715–718, May 2019.

[2] Jan J. De Waele, Murat Akova, Massimo Antonelli, Rafael Canton, Jean Carlet, and Daniel et al. De Backer. *Antimicrobial resistance and antibiotic stewardship programs in the ICU: insistence and persistence in the fight against resistance. A position statement from ESICM/ESCMID/WAAAR round table on multi-drug resistance.* Intensive Care Medicine, 44(2):189–196, February 2018.

[3] World Health Organization. *Ten health issues WHO will tackle this year*, 2023. Accessed: 17-10-2023.

[4] Jason A. Roberts, Mohd H. Abdul-Aziz, Jeffrey Lipman, Johan W. Mouton, and Alexander A. et al. Vinks. *Individualised antibiotic dosing for patients who are critically*

ill: challenges and potential solutions. The Lancet. Infectious Diseases, 14(6):498–509, June 2014.

[5] Jean-Louis Vincent, Yasser Sakr, Mervyn Singer, Ignacio Martin-Loeches, Flavia R. Machado, John C. Marshall, Simon Finfer, and Paolo et al. Pelosi. *Prevalence and Outcomes of Infection Among Patients in Intensive Care Units in 2017*. JAMA, 323(15):1478–1487, April 2020.

[6] Jason A. Roberts, Sanjoy K. Paul, Murat Akova, Matteo Bassetti, Jan J. De Waele, and George et al. Dimopoulos. *DALI: Defining Antibiotic Levels in Intensive Care Unit Patients: Are Current beta-Lactam Antibiotic Doses Sufficient for Critically Ill Patients?* Clinical Infectious Diseases, 58(8):1072–1083, April 2014.

[7] Mohd H. Abdul-Aziz, Jan-Willem C. Alffenaar, Matteo Bassetti, Hendrik Bracht, and George et al. Dimopoulos. *Antimicrobial therapeutic drug monitoring in critically ill adult patients: a Position Paper#*. Intensive Care Medicine, 46(6):1127–1153, June 2020.

[8] Rekha Pai Mangalore, Aadith Ashok, Sue J Lee, Lorena Romero, Trisha N Peel, Andrew A Udy, and Anton Y Peleg. *Beta-Lactam Antibiotic Therapeutic Drug Monitoring in Critically Ill Patients: A Systematic Review and Meta-Analysis*. Clinical Infectious Diseases, 75(10):1848–1860, November 2022.

[9] Jarne Verhaeghe, Sofie A. M. Dhaese, Thomas De Corte, David Vander Mijnsbrugge, Heleen Aardema, Jan G. Zijlstra, Alain G. Verstraete, Veronique Stove, Pieter Colin, Femke Ongenae, Jan J. De Waele, and Sofie Van Hoecke. *Development and evaluation of uncertainty quantifying machine learning models to predict piperacillin plasma concentrations in critically ill patients*. BMC Medical Informatics and Decision Making, 22(1):224, August 2022.

[10] Catherine M. T. Sherwin, Tony K. L. Kiang, Michael G. Spigarelli, and Mary H. H. Ensom. *Fundamentals of population pharmacokinetic modelling: validation methods*. Clinical Pharmacokinetics, 51(9):573–590, September 2012.

[11] Jie Tang, Rong Liu, Yue-Li Zhang, Mou-Ze Liu, Yong-Fang Hu, and Ming-Jie et al. Shao. *Application of Machine-Learning Models to Predict Tacrolimus Stable Dose in Renal Transplant Recipients*. Scientific Reports, 7, February 2017.

[12] Rong Liu, Xi Li, Wei Zhang, and Hong-Hao Zhou. *Comparison of Nine Statistical Model Based Warfarin Pharmacogenetic Dosing Algorithms Using the Racially Diverse International Warfarin Pharmacogenetic Consortium Cohort Database*. PLOS ONE, 10(8):e0135784, August 2015. Publisher: Public Library of Science.

[13] M. R. Poynton, B. M. Choi, Y. M. Kim, I. S. Park, G. J. Noh, S. O. Hong, Y. K. Boo, and S. H. Kang. *Machine learning methods applied to pharmacokinetic modelling of*

remifentanil in healthy volunteers: a multi-method comparison. The Journal of International Medical Research, 37(6):1680–1691, December 2009.

[14] Wei Guo, Ze Yu, Ya Gao, Xiaoqian Lan, Yannan Zang, Peng Yu, Zeyuan Wang, Wenzhuo Sun, Xin Hao, and Fei Gao. *A Machine Learning Model to Predict Risperidone Active Moiety Concentration Based on Initial Therapeutic Drug Monitoring*. Frontiers in Psychiatry, 12:711868, November 2021.

[15] Danish Shakeel and Shakeel Ahmad Mir. *Personalized drug concentration predictions with machine learning: an exploratory study*. International Journal of Basic & Clinical Pharmacology, 9(6):980, May 2020.

[16] Xiaolan Mo, Xiujuan Chen, Xianggui Wang, Xiaoli Zhong, and Huiying et al. Liang. *Prediction of Tacrolimus Dose/Weight-Adjusted Trough Concentration in Pediatric Refractory Nephrotic Syndrome: A Machine Learning Approach*. Pharmacogenomics and Personalized Medicine, 15:143–155, February 2022.

[17] Ahmad Y. Abuhelwa, David J.R. Foster, and Richard N. Upton. *ADVAN-style analytical solutions for common pharmacokinetic models*. Journal of Pharmacological and Toxicological Methods, 73:42–48, May 2015.

[18] Sean Grogan and Charles V. Preuss. *Pharmacokinetics*. In StatPearls. StatPearls Publishing, Treasure Island (FL), 2023.

[19] Xiuqing Zhu, Wencan Huang, Haoyang Lu, Zhanzhang Wang, Xiaojia Ni, Jinqing Hu, Shuhua Deng, Yaqian Tan, Lu Li, Ming Zhang, Chang Qiu, Yayan Luo, Hongzhen Chen, Shanqing Huang, Tao Xiao, Dewei Shang, and Yuguan Wen. *A machine learning approach to personalized dose adjustment of lamotrigine using noninvasive clinical parameters*. Scientific Reports, 11(1):5568, March 2021.

[20] William A. Craig. *The Pharmacology of Meropenem, A New Carbapenem Antibiotic*. Clinical Infectious Diseases, 24(Supplement_2):S266–S275, February 1997.

[21] Pan Ma, Ruixiang Liu, Wenrui Gu, Qing Dai, Yu Gan, Jing Cen, Shenglan Shang, Fang Liu, and Yongchuan Chen. *Construction and Interpretation of Prediction Model of Teicoplanin Trough Concentration via Machine Learning*. Frontiers in Medicine, 9:808969, March 2022.

[22] Michael E. Brier, Jacek M. Zurada, and George R. Aronoff. *Neural Network Predicted Peak and Trough Gentamicin Concentrations*. Pharmaceutical Research, 12(3):406–412, March 1995.

[23] Alexander Janssen, Frank W. G. Leebeek, Marjon H. Cnossen, Ron A. A. Mathôt, and for the OPTI-CLOT study group and SYMPHONY consortium. *Deep compartment models: A deep learning approach for the reliable prediction of time-series data*

in pharmacokinetic modeling. CPT: Pharmacometrics & Systems Pharmacology, 11(7):934–945, 2022.

[24] D. W. Cockcroft and M. H. Gault. *Prediction of creatinine clearance from serum creatinine*. Nephron, 16(1):31–41, 1976.

[25] Andrew S. Levey, Josef Coresh, Tom Greene, Lesley A. Stevens, Yaping Lucy Zhang, Stephen Hendriksen, John W. Kusek, Frederick Van Lente, and Chronic Kidney Disease Epidemiology Collaboration. *Using standardized serum creatinine values in the modification of diet in renal disease study equation for estimating glomerular filtration rate*. Annals of Internal Medicine, 145(4):247–254, August 2006.

[26] a subsidiary of Pfizer Wyeth Piperacillin Division of Wyeth Holdings Corporation. *ZOSYN- tazobactam sodium and piperacillin sodium injection, powder, lyophilized, for solution*. Accessed 4 Dec 2023.

[27] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. *CatBoost: gradient boosting with categorical features support*. arXiv:1810.11363 [cs, stat], October 2018. arXiv: 1810.11363.

[28] Jarne Verhaeghe, Jeroen Van Der Donckt, Femke Ongenae, and Sofie Van Hoecke. *Powershap: A Power-Full Shapley Feature Selection Method*. In Machine Learning and Knowledge Discovery in Databases, Lecture Notes in Computer Science, pages 71–87, Cham, 2023. Springer International Publishing.

[29] Michel Struys, Anthony Absalom, and Steven L. Shafer. *Intravenous Drug Delivery Devices*. In Miller's Anesthesia. Elsevier, 9 edition, 2019.

[30] Idoia Bilbao-Meseguer, Alicia Rodríguez-Gascón, Helena Barrasa, Arantxazu Isla, and María Ángeles Solinís. *Augmented Renal Clearance in Critically Ill Patients: A Systematic Review*. Clinical Pharmacokinetics, 57(9):1107–1121, September 2018.

[31] Thomas De Corte, Jarne Verhaeghe, Sofie Dhaese, Sarah Van Vooren, Jerina Boelens, Alain G. Verstraete, Veronique Stove, Femke Ongenae, Liesbet De Bus, Pieter Depuydt, Sofie Van Hoecke, and Jan J. De Waele. *Pathogen-based target attainment of optimized continuous infusion dosing regimens of piperacillin-tazobactam and meropenem in surgical ICU patients: a prospective single center observational study*. Annals of Intensive Care, 13(1):35, April 2023.

[32] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. arXiv:1705.07874 [cs, stat], November 2017. arXiv: 1705.07874.

5

Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in ICU patients

In the ICU, various comorbidities like Atrial Fibrillation (AF) significantly influence patient outcomes. Atrial Fibrillation (AF) is characterized by irregular and rapid heart rates and can increase mortality and prolong ICU stays. Therefore, addressing any modifiable risk factors can reduce this impact. This chapter proposes and validates a calibrated AF risk estimation model using multiple ICU datasets to asses both predictive and UQ performance, as such adressing UCG1. Patients with AF often have longer ICU stays, creating distinct patient populations between those with AF and those without. This difference can introduce a potential time bias. To address this, the chapter demonstrates the use of patient-matching and class balancing to mitigate the time bias, which can negatively impact performance if left unaddressed. As the model aims to provide risk assessments, it outputs probabilities that are evaluated using novel adjusted metrics for UQ evaluation: the Expected Calibration Error (ECE) and the Expected Signed Calibration Error (ESCE). As such, this chapter also contributes towards adressing RG2. My contributions to this chapter are the validation and development of all the machine learning models, the study design, the experiments, the creation of the E(S)CE, and the writing of the chapter.

Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in ICU patients

Jarne Verhaeghe, Thomas De Corte, Christopher M Sauer, Tom Hendriks, Olivier WM Thijssens, Femke Ongenae, Paul Elbers, Jan De Waele, Sofie Van Hoecke

Published in International Journal of Medical Informatics

Abstract **Background:** Atrial Fibrillation (AF) is the most common arrhythmia in the intensive care unit (ICU) and is associated with increased morbidity and mortality. Identification of patients at risk for AF is not routinely performed as AF prediction models are almost solely developed for the general population or for particular ICU populations. However, early AF risk identification could help to take targeted preemptive actions and possibly reduce morbidity and mortality. Predictive models need to be validated across hospitals with different standards of care and convey their predictions in a clinically useful manner. Therefore, we designed AF risk models for ICU patients using uncertainty quantification to provide a risk score and evaluated them on multiple ICU datasets.

Methods: Three CatBoost models, utilizing feature windows comprising data 1.5-13.5, 6-18, or 12-24 hours before AF occurrence, were built using 2-repeat-10-fold cross-validation on AmsterdamUMCdb, the first freely available European ICU database. Furthermore, AF Patients were matched with no-AF patients for training. Transferability was validated using a direct and a recalibration evaluation on two independent external datasets, MIMIC-IV and GUH. The calibration of the predicted probability, used as an AF risk score, was measured using the Expected Calibration Error (ECE) and the presented Expected Signed Calibration Error (ESCE). Additionally, all models were evaluated across time during the ICU stay.

Results: The model performance reached Areas Under the Curve (AUCs) of 0.81 at internal validation. Direct external validation showed partial generalizability with AUCs reaching 0.77. However, recalibration resulted in performances matching or exceeding that of the internal validation. All models furthermore showed calibration capabilities demonstrating adequate risk prediction competence.

Conclusion: Ultimately, recalibrating models reduces the challenge of generalization to unseen datasets. Moreover, utilizing the patient-matching methodology together with the assessment of uncertainty calibration can serve as a step toward the development of clinical AF prediction models.

5.1 Introduction

Atrial fibrillation (AF) is a heart rhythm disorder that causes an irregular and often abnormally fast heart rate. It affects between 4.5 to 15% of patients admitted to the intensive care unit (ICU). The incidence is even higher in specific patient populations, i.e. patients admitted after cardiac surgery (35%) or patients with septic shock (40%-46%) [1–3]. Several studies have indicated that the occurrence of AF in critically ill patients is associated with poorer outcomes, including prolonged length of stay (LOS) and increased hospital mortality [1, 4]. Although several risk factors for AF are non-modifiable (e.g., age), identifying patients at high risk of developing AF could allow clinicians to preemptively address modifiable risk factors (e.g., electrolyte imbalances or medication). However, clinical identification of patients at risk for AF is not routinely performed in the ICU as developed clinical risk prediction models are often limited to either the general population or to selected ICU patient populations [3, 5, 6]. Therefore, we aimed to build a machine learning (ML) model to predict the risk of AF occurrence in real-time for any ICU patient using calibrated uncertainty predictions. Special attention was given to developing a model that generates a meaningful, interpretable risk output for the bedside clinician by using Shapley values. Additionally, the models were validated on data from multiple ICUs across the globe to determine their generalizability. Finally, the models were evaluated in a simulated clinical situation across time to understand their behavior in clinical practice, while fully explaining the model using interpretability libraries.

5.2 Materials and methods

A complete overview of all conducted experiments and methods in this study is visualized in Figure 5.1.

5.2.1 Prediction model development

5.2.1.1 Study population and prediction window

The AmsterdamUMCdb database (v1.02) was used for model development and internal validation [7]. The outcome, i.e. the occurrence of AF, was operationalized as the first AF registration by the nursing staff after at least one registration of a sinus rhythm. The study cohort was limited to patients with a minimal LOS of 13.5 hours. These 13.5 hours are based on a feature aggregation window of 12 hours and a 1.5-hour time window between feature aggregation and event occurrence to avoid data leakage. The database and the study population characteristics are summarized in Table 5.1.

Three models with different feature windows, but all with an aggregation window of 12 hours, were designed. The different windows reflect the period from which collected data is used to make a risk prediction. The models respectively have a window

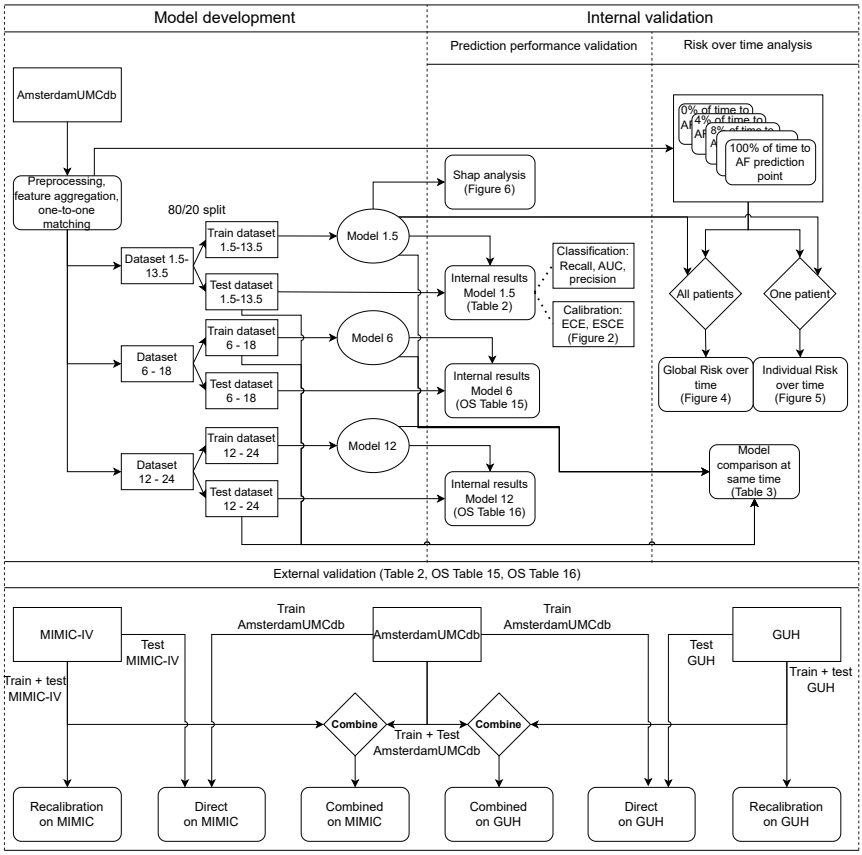


Figure 5.1: Diagram of the complete methodology of this study.

of 1.5 – 13.5, 6 - 18, and 12 - 24 hours before AF occurrence (Model-1.5, Model-6, and Model-12).

As there were more ICU admissions without AF (16,163) than with AF (2,000), and as AF patients tend to have a longer LOS [1, 4] the training set was balanced by one-to-one matching every AF admission to a no-AF admission. This was achieved by fixing the surrogate AF prediction point for the no-AF admission to the time after admission of the AF admission AF diagnosis following a case-control study design, as visualized in Figure 5.2. Only the relative time after ICU admission was considered. The AF admission AF diagnosis time should be before the ICU discharge of the candidate no-AF admission. Furthermore, the no-AF admission should not have been matched already to an AF admission. Once these requirements were fulfilled, the two admissions were matched and shared their relative prediction points. This procedure was performed before train-test splitting. Afterward, all AF admissions were split into

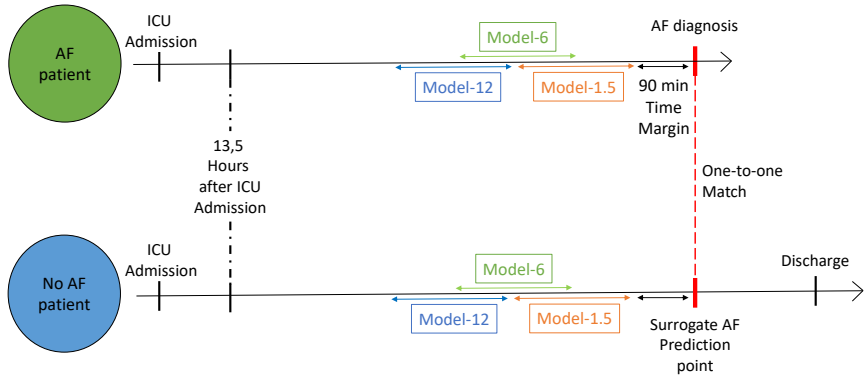


Figure 5.2: Illustration of the model development and the case-control matching procedure used. All information between the event and 90 minutes prior to the event was excluded. Every AF patient (green circle) was matched with a no-AF patient (blue circle). For the no-AF patient, the surrogate AF prediction point was defined as the same time point, relative to ICU admission, as AF occurrence in the AF patient (red line). Model-1.5 is built on a time window of 1.5-13.5 hours before AF occurrence. Model-6 is built on a time window of 6-18 hours. Model-12 is built on a time window of 12-24 hours.

a training and a test set with an 80/20 ratio. Their respective no-AF match was then also put in their respective training or test set to avoid splitting a match. The remaining no-AF admissions without a match were given a random prediction moment between 13.5 hours (resp. 18 or 24 hours, depending on the model) after admission and before discharge. These no-AF admissions were only used in a separate imbalanced test set for evaluation on the complete cohort. Ensuing, the models were trained on the training set and optimized using 2-repeat-10-fold cross-validation as there was enough data where two repeats were sufficiently stable. Finally, the models were evaluated on two different test sets: a balanced test set containing only matched admissions, and an imbalanced test set containing the matched AF admissions and all unmatched no-AF admissions.

5.2.1.2 Model building and feature selection

The models were built using Python (v3.8.10). CatBoost classification models were chosen because of their interpretability, missing value handling, strong prediction performances, and automatic categorical feature processing [8]. Furthermore, the model probability output was used for uncertainty quantification (UQ) to provide trustworthiness and an AF risk score. The classification performance was measured using the precision, recall, and Area Under the Curve (AUC) metrics.

Based on expert opinion and literature review, 282 unique potential variables were selected in AmsterdamUMCdb (listed in the Appendix 5.A). Variables with a count $<$ 250 or above 99% missingness in the whole dataset were excluded, resulting in 194

	AmsterdamUMC		MIMIC- IV*		GUH	
Time period	2003 – 2016		2008 – 2019		2013 - 2020	
Total admissions	23,106		69,211		25,297	
Study cohort size	18,163		59,492		23,459	
	AF	no-AF	AF	no-AF	AF	no-AF
Study cohort size	2,000	16,163	5,350	54,142	1,005	22,454
Age (group)	71 (69–75)	64 (59–75)	71 (63–80)	61 (50–73)	71 (66-79)	60 (50-72)
Gender (Male)	62%	65%	59%	56%	/	/
BMI	26.2 (23.9 – 27.8)	26.1 (23.9 – 27.8)	29 (22.1 - 34.8)	29.0 (22.0 - 33.6)	28.20 (24.3 - 30.6)	26.1 (22.5 - 28.4)
SOFA first 24h	8.8 (6–11)	5.7 (3–8)	5.3 (3-7)	3.4 (1-5)	7.1 (3-10)	4.3 (2-6)
APACHE II	22.0 (17–26)	16.6 (12–20)	/	/	27.8 (24-32)	20.2 (14-26)
Prediction point (Hours)	91.7 (31.3 – 91.3)	49.2 (16 – 41.0)	64.7 (27.2 - 71.06)	46.6 (19 - 48.0)	80.03 (33.9 - 82.4)	57.1 (18.4 - 51.7)
ICU LOS (days)	15.47 (3.9 - 20.4)	3.7 (0.9 - 2.9)	7.8 (2.9 - 9.0)	3.3 (1.2 - 3.5)	10.3 (3.7 - 11.7)	4.2 (1.0 - 4.0)
ICU survival	46.1%	71.8%	88.9%	94.6%	85.7%	92.8%

Table 5.1: The study population characteristics. Mean (Q25 - Q75) for continuous variables, percentages for categorical variables. BMI = Body Mass Index. SOFA = Sequential Organ Failure Assessment score. APACHE = Acute Physiology and Chronic Health Evaluation. LOS = Length Of Stay. *Only patients admitted to the ICU were evaluated for inclusion.

remaining variables. Subsequently, aggregate features were constructed to quantify the trend over time, such as min, mean, max, and slope. Powershap [9] was used as the feature selection method to determine the final feature set.

An initial feature selection with PowerShap was performed with the automatic mode on and with 500 CatBoost iterations. With these initially selected features, hyperparameter optimization of the CatBoost model was performed using 2-repeat-10-fold cross-validation as there was enough data where two repeats were sufficiently stable. The explored hyperparameters of the CatBoost model were: *depth* = [1, 2, 3, 4, 5, 6], *iterations* = [100, 200, 300, 400, 500], *l2 regularization* = [1, 2, 3, 4, 5, 6], with the *LogLoss* loss-function. There could be minor indirect data leakage because the feature selection is not performed in the loop of the cross-validation. However, as PowerShap also uses random validation splits to determine important features, this effect is mostly mitigated. Furthermore, performing PowerShap inside the 2-repeat-10-fold cross-validation would increase the time complexity by a factor of around 40 minutes per loop. These new hyperparameters were then used in PowerShap with the following arguments: *force_convergence* = *True*, *limit_convergence_its* = *1*, *automatic* = *True*. After this execution, multiple aggregates of the same feature were removed, and only the most important aggregate was kept. Then, with these new features, new hyperparameters were sought. This process continues until the cross-validation results stop improving based on a combination of the AUC, precision, and recall. The final hyperparameters are shown in Table 5.2.

Model	Depth	Iterations	L2 regularization
Model-1.5	3	300	3
Model-6	3	400	3
Model-12	3	300	3

Table 5.2: The final hyperparameters for all three models.

The final features for Model-1, Model-6, and Model-12 are shown in Table 5.3. Eleven features are correlated with the occurrence of AF in every model, although for some features different types of aggregates were retained. Features that are not consistent over all models also do not have an obvious clinical relationship to the occurrence of AF. Age is one of the most important and well-known risk factors for AF occurrence. All other consistent features besides ICU admission urgency either have a known link or influence the hemodynamics of the patients and/or are associated with tissue/cardiac oxygenation.

Features	Model-1.5	Model-6	Model-12
Age	X	X	X
PEEP setting (mmHg)	Mean	Mean	Mean
ICU admission Urgency	X	X	X
Heart frequency (bpm)	Max	Max	Max
Has received noradrenalin	X	X	X
CVP (mmHg)	Mean	Min	Mean
Ventilator administered FiO2 (%)	Mean	Mean	Mean
Calculated O2 saturation on ABG (%)	Mean	Mean	Mean
Has received loop diuretics	X	X	X
Hourly urinary volume (mL/h)	Mean	Mean	Max
Systolic ABP (mmHg)	Slope	Max	Max
Fluid balance (mL)	X	X	
Thrombocytes ($10^3/\mu L$)	Min		Max
Phosphate (mmol/L)		Max	Min
Has received Propofol (Diprivan)			X
Blood ureum (mmol/L)	Mean		
Arterial pH on blood gas			Min
Base excess (mmol/L)	Slope		
Lactate (mmol/L)		Min	
Mean ABP (mmHg)		Slope	

Table 5.3: Definitive feature set for all models. X indicates that the feature does not have an aggregate associated with it. The unit of the feature (if relevant) is indicated between brackets. Age = the upper of the following age groups: 18 – 39 yo, 40 – 49 yo, 50 – 59 yo, 60 – 69 yo, 70 – 79 yo. 80 for the group: $>$ 80 yo. PEEP = Peak End Expiratory Pressure. Max = maximum. Min = minimum. CVP = central venous pressure. ABP = Arterial Blood Pressure. Sec = seconds. ABG = arterial bood gas. Bpm = beats per minute. FiO2 = fraction of inspired oxygen.

5.2.2 External Validation

Two datasets were used for external validation. The first one is the publicly available Medical Information Mart for Intensive Care (MIMIC-IV) [10]. The second dataset comprised all patients admitted between 2013 and 2020 to the ICU of Ghent University Hospital (GUH), a tertiary Belgian hospital with a total of 52 surgical and medical ICU beds. The study population in both datasets was also defined as described in Section 5.2.1.1. Data descriptions of all datasets and missing values are present in Appendix 5.B.

Several approaches were used to evaluate different hypotheses during external validation. To test 'out-of-the-box-readiness' and robustness, the model trained on the AmsterdamUMCdb data was directly applied to the external datasets (denoted as Direct - MIMIC-IV and Direct – GUH).

The second hypothesis was that the prediction models could be transferred between hospitals and adapted to local customs without a detrimental drop in performance or the need for complete redevelopment. To test this, the model framework developed on AmsterdamUMCdb (type of model, hyperparameters, definitive feature set, etc.) was retrained using the same one-to-one matching training approach on each external dataset individually to recalibrate the model (denoted as Recalibrated – MIMIC-IV and Recalibrated – GUH, respectively). These recalibrated models were then evaluated against an unseen test set derived from this same dataset. The same approach as explained in Section 5.2.1.1 for AmsterdamUMCdb was used to construct the balanced training and test set and model development.

Finally, to determine if adding more data would increase performance, two datasets were created: AmsterdamUMCdb with GUH and AmsterdamUMCdb with MIMIC-IV. These datasets were then used to develop a model using the previously mentioned method.

5.2.3 Uncertainty calibration

To assess the UQ performance, the uncertainty prediction calibration was measured using the Expected Calibration Error (ECE) to quantify the average absolute calibration error [11]. We also propose a variant, called the Expected Signed Calibration Error (ESCE), to quantify the uncertainty bias. These metrics are the classification variant of the distribution coverage error and absolute distribution coverage error used for regression UQ evaluation [12]. These metrics are used instead of the Brier score because the Brier score does not evaluate the clinical value of diagnostic tests or prediction models [13]. To calculate the ECE and ESCE, the probability output $\hat{p}_i$, which is bounded between 0 and 1, is binned. This will result in M bins in total with a bin size equal to $1/M$. The errors are then calculated by subtracting the average confidence per bin from the accuracy within that bin using the following formulae:

$$acc(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i) \tag{5.1}$$

$$conf(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i \tag{5.2}$$

$$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)| \tag{5.3}$$

$$ESCE = \sum_{m=1}^{M} \frac{|B_m|}{n} (acc(B_m) - conf(B_m)) \tag{5.4}$$

B_m is the bin with the samples whose probability output falls into the half-open interval $((1-m)/M, m/M]$. $\hat{y}_i$ and y_i are the predicted resp. true class label for

sample i, and n represents the number of samples. These formulae are defined for a single bin size of $1/M$. However, by only using a single bin size these metrics are susceptible to local variance and can therefore only give a limited view of the uncertainty calibration. To understand the total model calibration, the bin size was varied from 0.005 to 0.05 with 0.001 intervals. The mean of the ECE and ESCE over all these varying bin sizes are reported in this manuscript. Furthermore, this method also facilitates proper graphical visualization in calibration plots.

5.2.4 Prediction over time analysis

As the goal is clinical practice, it is beneficial to understand how the proposed models will work in this setting. Therefore, the models were evaluated throughout the ICU admission on all test patients to understand the model's behavior across time. Hence, every test patient received multiple prediction points throughout their ICU admission instead of one. These moments were defined as percentages of time from the original diagnostic/surrogate AF prediction point (T_AF_Event) determined after one-to-one matching: $f * T\_AF\_Event$ with f a fraction between 0 and 1. These fractions started at admission and were incremented by 0.04 until the original prediction point was reached. This resulted in 26 evaluations across time for every test patient. The label was the same across all these prediction points, i.e., AF or no-AF. With this method, it is possible to verify whether or not the models provide a risk estimate, characterized by a steady or slightly increasing performance across time. Furthermore, to evaluate and compare the performance of each model at the same time instant, every model was evaluated on an unbalanced dataset 24 to 36 hours before the prediction point.

5.2.5 Shapley analysis

To interpret the final models, the SHAP [14] library was used to understand the predictions and find potential correlated risk factors using Shapley values [14]. A Shapley value is always tied to a feature value of a data sample and represents the impact that feature has on the prediction, compared to the average prediction of the model across all samples.

5.3 Results

5.3.1 Internal and external validation

The Model-1.5 internal and external validation metrics are shown in Table 5.4. The tables for Model-6 and Model-12 can be found in Table 5.5 and Table 5.6 respectively. The internal validation reached an AUC of 0.81 showing prediction capabilities, even

Method	External dataset	Validation patient group	no-AF patients	AF patients	no-AF recall	AF recall	no-AF precision	AF precision	AUC	ECE	ESCE
Internal validation	/	A	14563	400	0.75	0.71	0.99	0.07	0.81	0.04	0.04
		B	400	400	0.74	0.71	0.72	0.73	0.81	0.05	0.02
Direct	MIMIC -IV	A	49863	1070	0.90	0.38	0.99	0.07	0.77	0.12	0.12
		B	1070	1070	0.90	0.38	0.59	0.79	0.76	0.09	-0.08
	GUH	A	19244	183	0.57	0.91	1.00	0.02	0.84	0.15	-0.15
		B	182	183	0.57	0.91	0.87	0.68	0.83	0.08	0.04
Re-calibration	MIMIC -IV	A	49863	1070	0.69	0.73	0.99	0.05	0.79	0.02	-0.01
		B	1070	1070	0.67	0.73	0.71	0.69	0.78	0.05	0.02
	GUH	A	19244	183	0.74	0.84	1.00	0.03	0.85	0.04	-0.04
		B	182	183	0.74	0.84	0.82	0.77	0.85	0.09	0.07
Combining Amsterdam UMCdb with external dataset	MIMIC -IV	A - MIMIC-IV	49863	1070	0.70	0.72	0.99	0.05	0.79	0.02	0.00
		B - MIMIC-IV	1070	1070	0.68	0.72	0.71	0.69	0.78	0.05	0.02
		A - UMCdb	14563	400	0.70	0.76	0.99	0.06	0.81	0.02	0.01
		B - UMCdb	400	400	0.69	0.76	0.74	0.71	0.81	0.06	0.04
	GUH	A - GUH	19244	183	0.71	0.83	1.00	0.03	0.84	0.04	-0.03
		B - GUH	182	183	0.71	0.83	0.81	0.74	0.83	0.10	0.08
		A - UMCdb	14563	400	0.73	0.70	0.99	0.07	0.80	0.02	0.02
		B - UMCdb	400	400	0.75	0.70	0.71	0.73	0.81	0.06	0.02

Table 5.4: The results of Model-1.5 . A = All patients, B = Balanced test set

Method	External dataset	Validation patient group	NO AF patients	AF patients	NO AF recall	AF recall	NO AF precision	AF precision	AUC	ECE	ESCE
Internal validation	/	A	14578	371	0.70	0.70	0.99	0.06	0.78	0.02	0.01
		B	371	371	0.72	0.70	0.71	0.72	0.79	0.07	0.03
Direct	MIMIC -IV	A	50004	981	0.86	0.4	0.99	0.05	0.73	0.09	0.09
		B	981	981	0.85	0.4	0.58	0.73	0.74	0.09	-0.09
	GUH	A	12241	158	0.78	0.68	0.99	0.04	0.80	0.06	0.05
		B	158	158	0.81	0.68	0.72	0.78	0.82	0.10	0.09
Re-calibration	MIMIC -IV	A	50004	981	0.67	0.72	0.99	0.04	0.77	0.04	-0.03
		B	981	981	0.7	0.72	0.72	0.71	0.78	0.04	0.03
	GUH	A	12241	158	0.75	0.84	1.00	0.04	0.87	0.04	-0.03
		B	158	158	0.75	0.84	0.83	0.77	0.88	0.10	0.08
Combining Amsterdam UMCdb with external dataset	MIMIC -IV	A - MIMIC-IV	50004	981	0.68	0.71	0.99	0.04	0.77	0.03	-0.01
		B - MIMIC-IV	981	981	0.7	0.71	0.71	0.7	0.78	0.05	0.03
		A - UMCdb	14578	371	0.64	0.73	0.99	0.05	0.76	0.05	-0.04
		B - UMCdb	371	371	0.67	0.73	0.71	0.69	0.79	0.07	0.02
	GUH	A - GUH	12241	158	0.74	0.78	1.00	0.04	0.85	0.04	-0.01
		B - GUH	158	158	0.76	0.78	0.77	0.76	0.87	0.13	0.09
		A - UMCdb	14578	371	0.67	0.69	0.99	0.05	0.76	0.04	-0.02
		B - UMCdb	371	371	0.72	0.69	0.70	0.71	0.79	0.05	0.02

Table 5.5: The results of Model-6. A = All patients, B = Balanced test set

Method	External dataset	Validation patient group	NO AF patients	AF patients	NO AF recall	AF recall	NO AF precision	AF precision	AUC	ECE	ESCE
Internal validation	/	A	8623	334	0.67	0.68	0.98	0.08	0.75	0.04	-0.03
		B	334	334	0.73	0.68	0.70	0.71	0.78	0.06	0.03
Direct	MIMIC -IV	A	43566	858	0.74	0.55	0.99	0.04	0.72	0.04	0.03
		B	858	858	0.75	0.55	0.62	0.68	0.71	0.05	-0.03
	GUH	A	9835	158	0.67	0.73	0.99	0.03	0.77	0.06	-0.04
		B	157	158	0.68	0.73	0.71	0.69	0.75	0.14	0.05
Re-calibration	MIMIC -IV	A	43566	858	0.68	0.68	0.99	0.04	0.75	0.05	-0.01
		B	858	858	0.70	0.68	0.69	0.69	0.76	0.06	0.02
	GUH	A	9835	158	0.74	0.77	1.00	0.05	0.83	0.03	-0.02
		B	157	158	0.71	0.77	0.76	0.73	0.82	0.11	0.05
Combining Amsterdam UMCdb with external dataset	MIMIC -IV	A - MIMIC-IV	43566	858	0.7	0.66	0.99	0.04	0.75	0.04	0.01
		B - MIMIC-IV	858	858	0.72	0.66	0.68	0.7	0.76	0.05	0.02
		A - UMCdb	8623	334	0.60	0.75	0.98	0.07	0.75	0.09	-0.08
		B - UMCdb	334	334	0.66	0.75	0.73	0.69	0.78	0.07	0.05
	GUH	A - GUH	9835	158	0.70	0.77	0.99	0.04	0.81	0.04	-0.03
		B - GUH	157	158	0.71	0.77	0.75	0.73	0.79	0.12	0.08
		A - UMCdb	8623	334	0.66	0.68	0.98	0.07	0.75	0.06	-0.04
		B - UMCdb	334	334	0.73	0.68	0.69	0.72	0.77	0.06	0.02

Table 5.6: The results of Model-12. A = All patients, B = Balanced test set

when tested on all patients. Additionally, models with a time window closer to the AF occurrence performed better than models with a more distant time window (Model-1.5 AUC = 0.81 (Table 5.4), Model-6 AUC = 0.79 (Table 5.5), Model-12 AUC = 0.78 (Table 5.6)). Directly applying the designed models to the unseen datasets resulted in an expected performance drop, represented by the imbalanced recall metrics. The recalibrated models, however, showed comparable performance to the models developed on AmsterdamUMCdb for the MIMIC-IV dataset, and even better performance on the GUH dataset. Looking at the ECE and ESCE, the models showed sufficient uncertainty calibration with at most an average 5% error (ECE) in UQ for the internal results without a high bias (ESCE). However, the UQ performance on the GUH dataset was worse reaching up to 9% error for Recalibration-GUH. Figure 5.3 visualizes this calibration performance of Model-1.5 across all bins on the internal dataset. This figure shows that the calibration is accurate but slightly conservative. For example, the figure suggests that a prediction of 90% probability for a certain class is in 95% of the predictions correct.

The feature selection algorithm considered all variables in the AmsterdamUMCdb that passed the clinical selection. As the development of the model is done without regarding any external datasets, there could be some features that were available in AmsterdamUMCdb but not in the available external datasets. However, these selected features might contain significant information for predicting the occurrence of AF. Therefore, this effect on performance is quantified using 2-repeat-10-fold cross-validation by including and excluding these features and measuring their impact. The dropped features for each model are reported in Table 5.7 and the performance difference for dropping the features in Table 5.8. Interestingly, the impact of the features is limited with even almost no effect on Model-1.5. Therefore it can be guaranteed that no important features are dropped that severely impact the performance of the models.

Dropped features	Model-1.5	Model-6	Model-12
Fluid in (ml)	X		
O2 (l/min)	Max		Mean
Has received Enoximon (Perfan)	X	X	X
APTT in blood (sec)		Slope	
Anion-Gap in blood (mmol/l)	Max		
Creatinine in blood (μmol/l)	Mean		

Table 5.7: Dropped feature set for all models. The unit of the feature (if relevant) is indicated between brackets. Max = maximum. Min = minimum. APTT = Activated Partial Thromboplastin Time. Sec = seconds.

Model	Dropped Features	AUC (± std)	Precision (± std)	Recall (± std)
1.5	without	0.80 (± 0.04)	0.73 (± 0.04)	0.73 (± 0.04)
	included	0.81 (± 0.04)	0.73 (± 0.04)	0.73 (± 0.04)
6	without	0.79 (± 0.04)	0.72 (± 0.05)	0.71 (± 0.05)
	included	0.79 (± 0.05)	0.71 (± 0.06)	0.71 (± 0.06)
12	without	0.79 (± 0.06)	0.72 (± 0.05)	0.72 (± 0.05)
	included	0.79 (± 0.06)	0.72 (± 0.06)	0.72 (± 0.06)

Table 5.8: The results of dropping the features that are not available in the external datasets evaluated on the 2-repeat-10-fold cross-validation. std = standard deviation.

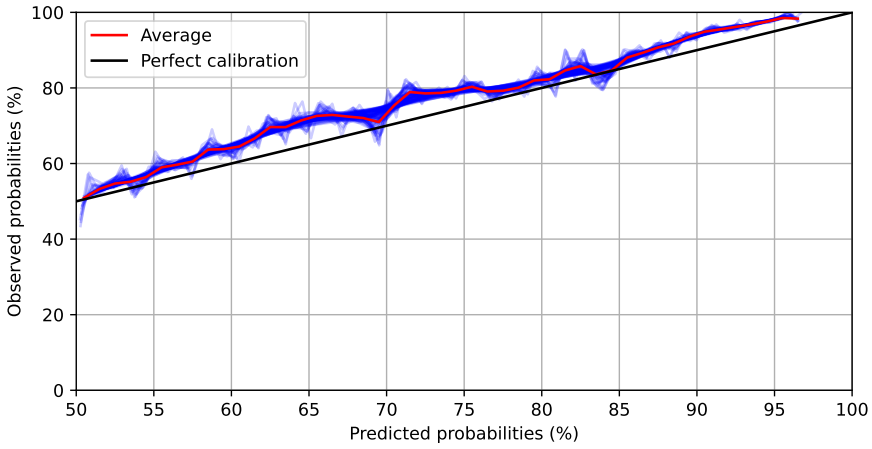


Figure 5.3: The Calibration plot for Model-1.5 on all patients evaluated on the AmsterdamUMCdb. The blue lines represent the varying bin sizes from 0.005 to 0.05. The red line is the average calibration of all these bin sizes. The probabilities represent the probability for the predicted class.

5.3.2 Evaluation over time

Figure 5.4 and Figure 5.5 display how Model-1.5 performs over the entire admission period. Figure 5.4 demonstrates that the model can achieve a class-weighted average recall of 65% and an AUC of around 0.7 at 10% of the time to the original prediction point, which is close to ICU admission. These results improve when the prediction point reaches the original AF prediction point, verifying the prediction of risks.

Looking at the individual level, shown in Figure 5.5, these trends continue. Here, the predicted probability of AF changes across time in response to the features in the dataset. Additionally, the Shapley values visualize the impact on the prediction resulting from these features and their changes across time, enabling a total image of the

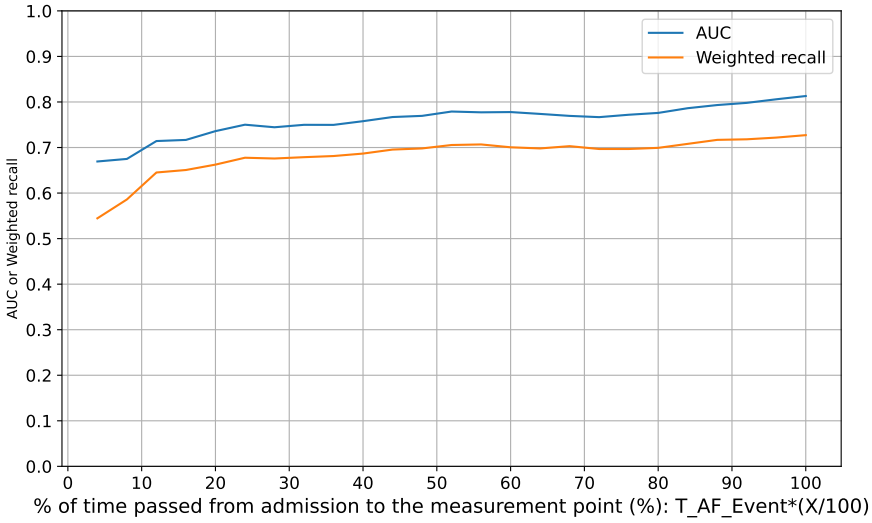


Figure 5.4: Evaluation over time for Model-1.5. The weighted recall is the weighted average of the recalls of both classes. The results are calculated using the balanced test set.

model's behavior across time. As an example, consider the time frame between 20% and 40% in Figure 5.5. The maximum heart frequency rises to 180 bpm resulting in a significantly higher AF risk prediction for that period, indicated by increased Shapley values from 0.0 to 1.0. This is also visible in the AF probability sub-graph, where the risk increases to 90%, and drops to 60% when the heart frequency drops again. Accordingly, the changes in the other features and their respective Shapley values can be analyzed using the same interpretation.

The results of the comparison of all models at the same time instances are shown in Table 5.9. Model-1.5 has the best AF recall performance, demonstrating Model-1.5 can best find patients at risk for AF compared to the other models (Wilcoxon signed rank test p-value < 0.001).

5.3.3 Shapley Analysis

Figure 5.6 visualizes the Shapley analysis for the features in the model. This figure demonstrates, e.g., that giving noradrenalin is correlated with a higher risk of AF according to the model. The most important feature here is age, where the model discovered that higher age (red) is a considerable predictor for AF (positive Shapley value). This method also works on the sample prediction level and enables physicians to evaluate both input and output to make an informed decision when using the model.

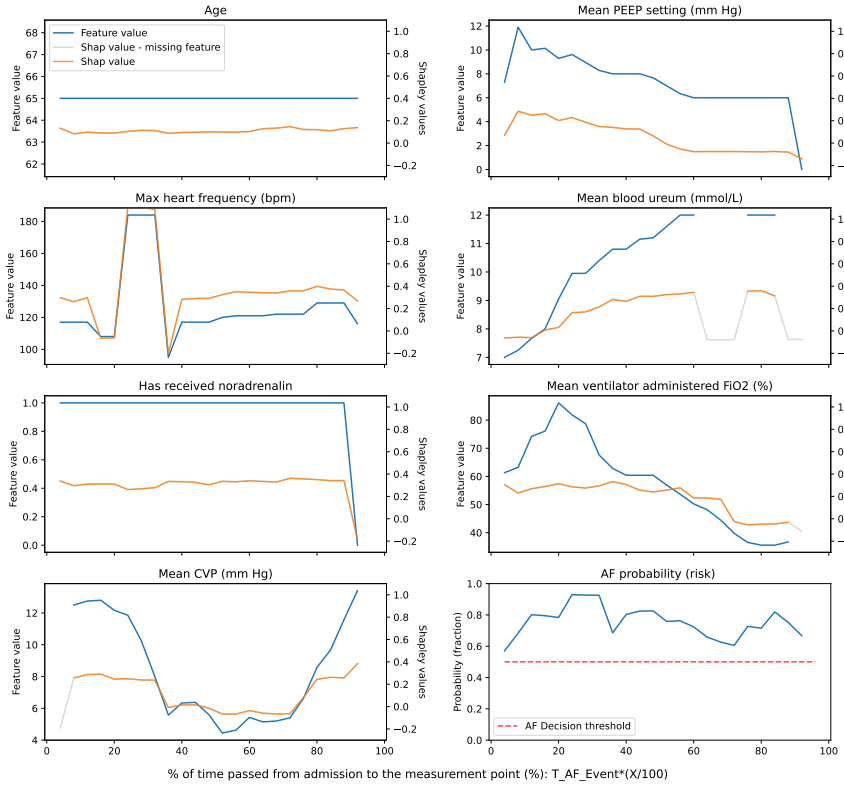


Figure 5.5: Evaluation over time for a single AF patient with Model-1.5. Blue = the feature value, orange = the Shapley value, grey = the Shapley value corresponding to the missing feature value, and Red = the decision boundary. The seven most important features are visualized together with their Shapley values. The prediction probability of AF over time is also visualized, where AF is predicted when this probability is above 0.5.

5.4 Discussion

We developed risk prediction models for AF occurrence in critically ill patients achieving adequate internal validation performances. Furthermore, retraining the models on combined multi-centric datasets did not improve the performance metrics indicating that the maximum performance was reached with the current features, models, and methods. Previous papers have also aimed to predict the occurrence of AF using ML. For instance, Ortega-Martorell et al. used a similar approach and achieved a comparable AUC (0.836), though without verified risk prediction and external validation [15]. Others tried to leverage electrocardiography (ECG) signals in ML with promising results [16, 17]. However, these models focus on diagnosing of AF a short time before its occurrence and use ECG data instead of routinely collected healthcare data.

Model	no-AF recall	AF recall	no-AF precision	AF precision	AUC
Model-1.5	0.68 [0.66-0.69]	0.74 [0.67-0.81]	0.98 [0.98-0.99]	0.10 [0.09-0.11]	0.77 [0.74-0.80]
Model-6	0.66 [0.65-0.68]	0.73 [0.67-0.80]	0.98 [0.98-0.98]	0.10 [0.09-0.10]	0.76 [0.73-0.80]
Model-12	0.68 [0.67-0.70]	0.71 [0.65-0.78]	0.98 [0.97-0.98]	0.10 [0.09-0.11]	0.78 [0.74-0.81]

Table 5.9: The results of evaluating each model on the unbalanced test set but created with data 24 to 36 hours before the prediction point to compare and evaluate the performance of the models at the same time instant, while being trained on their original dataset. These results are created by bootstrapping this dataset 1000 times and reported in the following format: mean [95% confidence interval]

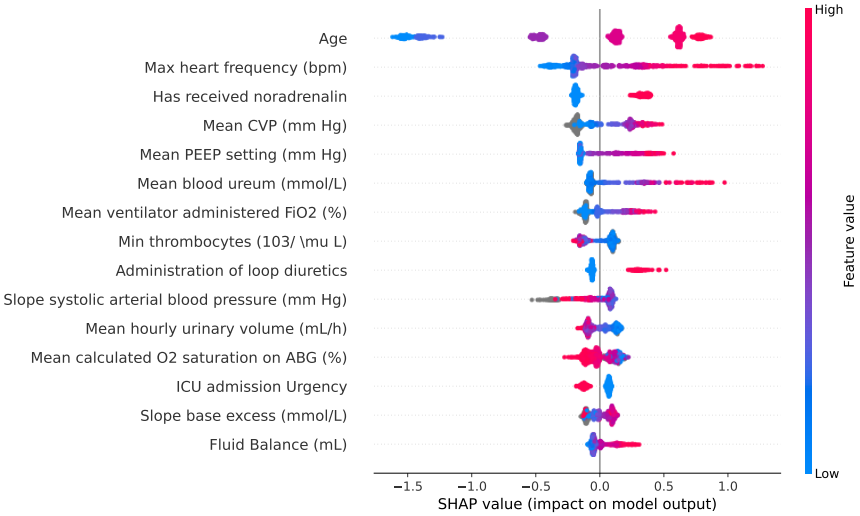


Figure 5.6: Shapley analysis of Model-1.5. The grey values are NaNs (missing feature values).

In high-stakes environments, such as the ICU, it is argued that model interpretability and transparency are paramount [18]. The use of 'white-box' models and the Shapley analysis are key factors for both interpretability and transparency. Furthermore, the transparency of the models is additionally reinforced by providing a calibrated risk score between 0 and 1 with low E(S)CE calibration errors. The uncertainty can resonate with the thought process clinicians use when making decisions and therefore contribute to the interpretability and transparency of the models. Combined, all these properties transform the output of the model from a binary occurrence prediction to a continuous risk prediction that facilitates the development of decision support sys-

tems. The Shapley analysis facilitates hypothesizing which predictors are risk factors for AF and whether clinicians can influence these to lower the risk of AF. Additionally, given the calibrated UQ results, the analysis of the effects of adjustment of care delivery can be used as a steppingstone for future causal (machine learning) studies.

Although several ML models have been developed in recent years, many fail to be translated into clinical practice [19]. One of the main issues hampering their widespread integration is the lack of external validity when directly applying developed models to unseen data. This also applies to this study, as directly applying our internally validated models to an external dataset was accompanied by a drop in overall performance mainly attributable to a data shift

Figure 5.7 and 5.8 show the data reconstruction drift between the AmsterdamUMCdb and the MIMIC-IV or GUH dataset respectively. The figures are created using the multivariate drift calculations incorporated in the Python library NannyML [20]. The reconstruction error of the analysis datasets is higher than the determined drift thresholds indicating a significant data shift or data difference between the datasets. The analysis uses a multivariate calculation to incorporate all features and ensure that the calculations also count potential interdependency between features. This method only considers the data and not the learned model and is therefore not dependent on the performance of Model-1.5, Model-6, or Model-12. However, the significant data difference explains why the direct external validation lacks generalizability and why recalibration is necessary to mitigate this issue. By applying this recalibration method, the external validation AUC remained consistent with the internal validation AUC. This demonstrates a suitable strategy to overcome a considerable barrier between AI research and patient care and provide generalizability for many settings.

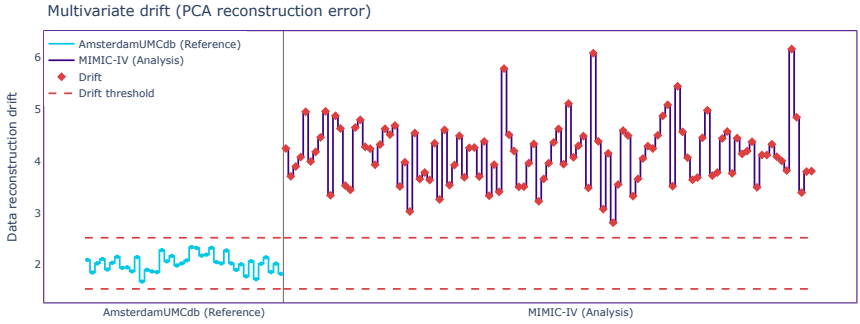


Figure 5.7: Data shift analysis between AmsterdamUMCdb and MIMIC-IV for the Model-1.5 test set.

Our study also has limitations. AF diagnosis was based on nursing charts as electrocardiograms were not available in AmsterdamUMCdb. Therefore, the study depends on the accuracy and timely recording of the diagnosis by the nurses to avoid data leakage. However, this accuracy and timely recording of AF registrations has

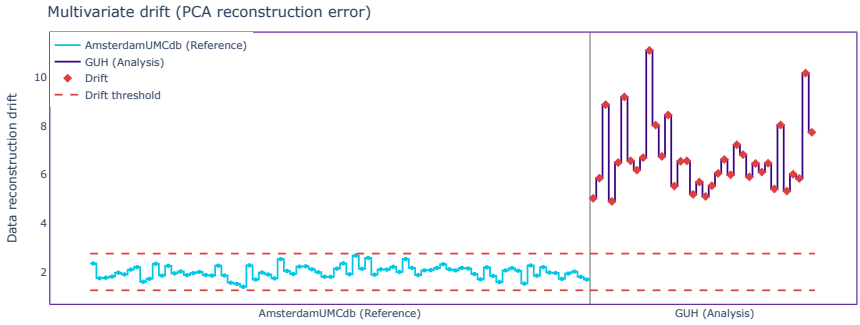


Figure 5.8: Data shift analysis between AmsterdamUMCdb and GUH for the Model-1.5 test set.

been studied and found to be adequate [21]. Additionally, Moss et al. found that new-onset AF diagnosed by both clinicians and an ECG model was associated with increased LOS and hospital mortality, whereas new-onset AF diagnosed by only the algorithm was not [4]. Hence, using only nursing chart notes is likely sufficient to capture clinically relevant AF episodes. Classifying ML models that use ECG waveforms have already been developed to improve AF detection and timely diagnosis registration [22, 23]. Although adding ECG data could increase the performance, our method enables the clinician to get a risk prediction using only routinely collected data. Finally, as medical history is not recorded in the AmsterdamUMCdb database, the focus was not to discriminate new-onset AF from an AF event in a patient already known with AF.

5.5 Conclusion

We proposed AF Risk models that provide a calibrated risk score between 0 and 1 for ICU patients. The calibration of these risk scores was verified using the ECE and the ESCE metrics. These models were built using a case-control design for training to facilitate discriminating between AF and no-AF patients to achieve meaningful results of 0.81 AUC. Furthermore, these models were validated on multiple datasets using various validation methods. Among these, the recalibration method was identified to be a reliable method with minimal effort to achieve generalization across ICUs globally. Ultimately, the used methodologies in this article can serve as a step toward the development of clinical AF prediction models across multiple ICUs.

Data Availability

The code is available on GitHub using the following link: https://github.com/predict-idlab/atrial_fibrillation_prediction. The AmsterdamUMCdb dataset is available by request. More information can be found here: https://github.com/AmsterdamUMC/AmsterdamUMCdb. The GUH dataset can only be shared after the ethical approval of the GUH ethical board. The MIMIC-IV dataset is available by request on the following link: https://physionet.org/content/mimiciv/2.1/.

Appendix

Appendix 5.A List of considered variables from AmsterdamUMCdb

Itemid	Item	Itemid	Item
9466	.CVVH starten	8492	AnGap*
13092	A_Ademfrequentie*	9559	Anion-Gap (bloed)
13080	A_Arteriële_PH*	18588	Apache II Hoofdgroep
13075	A_Hartfrequentie*	16997	APACHE IV Groepen
13070	A_Serum_Kalium*	11944	APTT (bloed)
13067	A_Temperatuur*	17982	APTT (bloed)
6642	ABP gemiddeld	11948	APTT Gecorrigeerd (bloed)*
6641	ABP systolisch	6806	ASAT
9992	Act.HCO3 (bloed)	11990	ASAT (bloed)
10195	ACTH (bloed)	6862	Atenolol (Tenormin)
7624	Actrapid (Insuline)	6864	Atropine sulfaat
6816	Adenosine*	6807	B.E.
	Admissioncount	9994	B.E. (bloed)
	admissionyeargroup	6808	Bezinking*
10197	Adrenaline (bloed)*	11902	Bezinking (bloed)
10199	Adrenaline (bloed)*	6813	Bili Totaal
6818	Adrenaline (Epinefrine)	9945	Bilirubine (bloed)
	agegroup	6812	Bilirubine geconjugeerd
6800	ALAT	19368	Bisoprolol*
11978	ALAT (bloed)	12460	Bloedflow
9937	Alb.Chem (bloed)	10736	Bloed-flow
6803	Alk. Fosfatase	12444	Bloedflow ingesteld

Table 5.A.1: Considered variables available in the AmsterdamUMCdb ordered alphabetically. $*$ features with less than 250 samples.

Itemid	Item	Itemid	Item
11984	Alk.Fosf. (bloed)	12087	Buikomvang
16113	Amiodaron	6882	Bumetanide (Burinex)*
9696	Amiodaron (bloed)*	21241	Bupivacaine HCL*
9015	Amiodaron Onderhoudsdosis	21242	Bupivacaïne/Sufentanil*
6844	Amiodaron Oplaaddosis	6815	Ca. geioniseerd*
9697	Amiodaron, desethyl (bloed)	9560	Ca++ Astrup
9561	CA++(7.4) Astrup	9721	DIGOXINE (bloed)*
10267	Ca-ion (7.4) (bloed)	18323	Digoxine (bloed)*
6817	Calcium	9087	Digoxine (lanoxin pg)
7412	Calcium Glubionaat (Calcium Sandoz)	7173	Digoxine (Lanoxin)*
9933	Calcium totaal (bloed)	7205	Digoxine spiegel*
19164	Calciumgluconaat*	7174	Diltiazem (Tildiem)
18783	Calciumgluconaat 10%	7178	Dobutamine (Dobutrex)
6656	Cardiac Output	15640	Dopamine (bloed)*
12938	Carvedilol (Eucardic)	7179	Dopamine (Inotropin)
6819	Chloor	19227	DOS score
9930	Chloor (bloed)	6826	Eiwit totaal*
9558	Chloor Astrup	7196	Enoximon (Perfan)
6820	Cholesterol*	9577	Ery/3ul*
9954	Cholesterol (bloed)	9578	Ery/ul*
6822	CK	9962	Ery's (bloed)
11998	CK (bloed)	7213	Esmolol (Brevibloc)*
6824	CK-MB	7214	Etomidaat (Hypnomidate)
14413	Cl (onv.ISE) (bloed)	6984	FCOHb*
6949	Cortisol	7219	Fentanyl
10238	Cortisol (bloed)	19929	Fenylefrine*
10751	Cortisol in bloed*	9126	Fenylefrine (phenylephrine)*
15142	CPAP PEEP (cmH2O)	6776	Fibrinogeen*
6825	CRP	10175	Fibrinogeen (bloed)
10079	CRP (bloed)	9989	Fibrinogeen (bloed)
18854	CRP (overig)*	9732	Flecaïnide (bloed)*
9553	CtHB Astrup	7224	Flecaïnide (Tambocor)*
6655	CVD	6828	Fosfaat
10393	D-dimeren (bloed)	9935	Fosfaat (bloed)
6992	Desmopressine (Minrin)*	7244	Furosemide (Lasix)
9561	CA++(7.4) Astrup	9721	DIGOXINE (bloed)*
10267	Ca-ion (7.4) (bloed)	18323	Digoxine (bloed)*
6817	Calcium	9087	Digoxine (lanoxin pg)
7412	Calcium Glubionaat (Calcium Sandoz)	7173	Digoxine (Lanoxin)*

Itemid	Item	Itemid	Item
9933	Calcium totaal (bloed)	7205	Digoxine spiegel*
19164	Calciumgluconaat*	7174	Diltiazem (Tildiem)
18783	Calciumgluconaat 10%	7178	Dobutamine (Dobutrex)
6656	Cardiac Output	15640	Dopamine (bloed)*
12938	Carvedilol (Eucardic)	7179	Dopamine (Inotropin)
6819	Chloor	19227	DOS score
9930	Chloor (bloed)	6826	Eiwit totaal*
9558	Chloor Astrup	7196	Enoximon (Perfan)
6820	Cholesterol*	9577	Ery/3ul*
9954	Cholesterol (bloed)	9578	Ery/ul*
6822	CK	9962	Ery's (bloed)
11998	CK (bloed)	7213	Esmolol (Brevibloc)*
6824	CK-MB	7214	Etomidaat (Hypnomidate)
14413	Cl (onv.ISE) (bloed)	6984	FCOHb*
6949	Cortisol	7219	Fentanyl
10238	Cortisol (bloed)	19929	Fenylefrine*
10751	Cortisol in bloed*	9126	Fenylefrine (phenylephrine)*
15142	CPAP PEEP (cmH2O)	6776	Fibrinogeen*
6825	CRP	10175	Fibrinogeen (bloed)
10079	CRP (bloed)	9989	Fibrinogeen (bloed)
18854	CRP (overig)*	9732	Flecaïnide (bloed)*
9553	CtHB Astrup	7224	Flecaïnide (Tambocor)*
6655	CVD	6828	Fosfaat
10393	D-dimeren (bloed)	9935	Fosfaat (bloed)
6992	Desmopressine (Minrin)*	7244	Furosemide (Lasix)
10258	Metanefrine (verz. urine)*	10282	O2-Content (bloed)
7184	Metoprolol (Selokeen)*	12311	O2-Saturatie (bloed)
8470	Metoprolol (ZOC Selokeen)	15808	Opname Sepsis
20078	MFT_Filtraatvolume_huidig	12128	Patiënt Specialisme
20079	MFT_Filtraatvolume_totaal	10469	Patiënt Geslacht

Itemid	Item	Itemid	Item
14849	MFT_Ultrafiltratie (ingesteld)	10468	PatiëntNationaliteit*
7194	Midazolam (Dormicum)	12335	PC boven PEEP (Set) (2)*
7225	Morfine	6846	PCO2
10739	NaCL 2,9%	9990	pCO2 (bloed)
6840	Natrium	21213	PCO2 (bloed) - kPa
9924	Natrium (bloed)	8879	PEEP (gemeten)
9555	Natrium Astrup	9666	PEEP (gemeten)(2)*
19932	Natrium bicarbonaat 1.4%	12284	PEEP (Set)
8936	Natrium bicarbonaat 4,2 %*	12336	PEEP (Set) (2)*
7295	Natrium bicarbonaat 8,4 %	8862	PEEP/CPAP
8998	Natriumbicarbonaat 8,4%	6848	PH
8999	Natrium-kalium-fosfaat	12310	pH (bloed)
19138	Nebivolol*	14058	PiCCO APm
10743	Nefrodrain li Uit	13150	PiCCO CI (Cardiac Index)X*
10745	Nefrodrain re Uit	13151	PiCCO CO (Cardiac Output)
19750	NICE Apache III Score	14055	PiCCO HF Hartfrequentie
19500	NICE Apache IV Score	14047	PiCCO Tb blood temperature
7229	Noradrenaline (Norepinefrine)	7433	PO2
10198	Noradrenaline (plasma)*	9996	PO2 (bloed)
10196	Noradrenaline (serum)*	21214	PO2 (bloed) - kPa
10259	Normetanefrine (verz. urine)*	6927	Procainamide (Pronestyl)*
14249	NT-proBNP (bloed)	7480	Propofol (Diprivan)
12282	O2 concentratie (Set)	6933	Propranolol (Hydrochloride)

Itemid	Item	Itemid	Item
8845	O2 l/min	11894	Prothrombinetijd (bloed)
11893	Prothrombinetijd (bloed)	10407	TroponineT (bloed)
6789	Protrombinetijd	7071	TSH*
12337	PS boven P hoog (Set) (2)*	11925	TSH (bloed)
12338	PS boven PEEP (Set) (2)*	6850	Ureum
12106	Risicofactor Roken Aantal sig/dag	9943	Ureum (bloed)
12107	Risiocofactor Roken Pack Years		Urgency
6982	Salbutamol (Ventolin)*	8794	UrineCAD
12312	Serotonine (bloed)*	8800	UrineIncontinentie
8908	Serum Eiwit*	8798	UrineSpontaan
10051	Serum Totaal Eiwit (bloed)	8796	UrineSupraPubis
7006	Sotalol (Sotacor)	8803	UrineUP
7056	T4*	9014	Velosuline (Insuline)
13952	Temp Blaas	7139	Verapamil (Isoptin)*
8658	Temp Bloed	11941	Vrij Cortisol (verz. urine)*
13063	Temp Huid	7079	Vrij T4*
8662	Temperatuur Perifeer 1	19702	Vrij-T3 (bloed)*
8659	Temperatuur Perifeer 2	10201	Vrij-T3 (bloed)
12467	Terlipressine (Glypressin)*	10187	Vrij-T4 (bloed)
7027	Thiamine (Vitamine B1)		weightgroup
6798	Thrombinetijd*	12071	Y-GT (bloed)
11950	Thrombinetijd (bloed)*	18587	Zuurstof toediening*
6797	Thrombocyten	6849	Triglyceride*
9964	Thrombo's (bloed)	9956	Triglyceriden (bloed)
10409	Thrombo's citr. bloed (bloed)*	11951	Trombinetijd (bloed)*
18558	Tot.Amiodaron + Amiodaron desethyl (bloed)*	8115	Troponine

Appendix 5.B Feature descriptions and missing values

Feature - AmsterdamUMCdb	Model-1.5	Model-6	Model-12
Age	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Has received noradrenalin	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Max heart frequency (bpm)	0.12 - 0.75	0.2 - 0.34	
Mean PEEP setting (mmHg)	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean CVP (mmHg)	35.06 - 58.31		32.36 - 54.53
Mean ventilator administered FiO2 (%)	35.56 - 58.5	32.59 - 51.62	30.41 - 48.61
Min thrombocytes ($10^3/\mu L$)	40.5 - 42.12		
ICU admission Urgency	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean hourly urinary volume (mL/h)	4.62 - 6.19	4.32 - 5.61	
Has received loop diuretics	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean blood ureum (mmol/L)	64.12 - 68.69		
Slope base excess (mmol/L)	16.44 - 33.38		
Slope systolic ABP (mmHg)	1.81 - 9.19		
Mean calc. O2 saturation on ABG (%)	7.81 - 15.38	9.45 - 16.76	7.79 - 15.66
Fluid Balance (mL)	0.56 - 1.5	0.34 - 1.22	
Min CVP (mmHg)		34.35 - 56.01	
Slope mean ABP (mmHg)		1.82 - 9.39	
Max Phosphate (mmol/L)		50.2 - 61.69	
Max systolic ABP (mmHg)		1.82 - 9.12	1.72 - 7.72
Min Lactate (mmol/L)		62.08 - 74.12	
Min arterial pH on blood gas			7.57 - 14.91
Max hourly urinary volume (mL/h)			4.57 - 5.32
Mean heart frequency (bpm)			0.15 - 0.22
Min Phosphate (mmol/L)			52.28 - 61.8
Has received Propofol (Diprivan)			0.0 - 0.0
Max thrombocytes ($10^3/\mu L$)			45.02 - 52.51

Table 5.B.1: Percentage missing values for each feature for each model on the training data for AmsterdamUMCdb. Structured as no-AF - AF data. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - no-AF - AmsterdamUMCdb	Model-1.5	Model-6	Model-12
Age	62.2 (49.0 - 75.0)	62.37 (49.0 - 75.0)	61.74 (49.0 - 75.0)
Has received noradrenalin	22%	26%	30%
Max heart frequency (bpm)	99.85 (86.0 - 112.0)	99.93 (87.0 - 112.0)	
Mean PEEP setting (mmHg)	3.34 (0.0 - 6.0)	3.88 (0.0 - 7.45)	4.14 (0.0 - 8.0)
Mean CVP (mmHg)	8.2 (5.37 - 10.67)		8.31 (5.24 - 10.78)
Mean ventilator administered FiO2 (%)	43.51 (40.0 - 45.49)	43.8 (40.0 - 46.36)	43.44 (40.0 - 45.91)
Min thrombocytes ($10^3/\mu L$)	212.13 (135.0 - 259.0)		
ICU admission Urgency	40%	41%	43%
Mean hourly urinary volume (mL/h)	439.44 (77.0 - 188.0)	142.97 (74.44 - 190.0)	
Has received loop diuretics	13%	13%	14%
Mean blood ureum (mmol/L)	8.91 (4.5 - 10.7)		
Slope base excess (mmol/L)	0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	0.0 (-0.02 - 0.03)		
Mean calc. O2 saturation on ABG (%)	0.96 (0.96 - 0.99)	0.96 (0.96 - 0.99)	0.96 (0.96 - 0.99)
Fluid Balance (mL)	-413.63 (178.95 - 1643.27)	1171.3 (316.9 - 1802.77)	
Min CVP (mm Hg)		7.1 (4.0 - 10.0)	
Slope mean ABP (mmHg)		0.0 (-0.01 - 0.02)	
Max Phosphate (mmol/L)		1.09 (0.83 - 1.3)	
Max systolic ABP (mmHg)		418.56 (140.0 - 188.0)	168.44 (140.0 - 187.0)
Min Lactate (mmol/L)		1.67 (0.8 - 1.8)	
Min arterial pH on blood gas			7.39 (7.35 - 7.44)
Max hourly urinary volume (mL/h)			258.81 (125.0 - 360.0)
Mean heart frequency (bpm)			84.9 (72.73 - 96.27)
Min Phosphate (mmol/L)			1.05 (0.79 - 1.25)
Has received Propofol (Diprivan)			29%
Max thrombocytes ($10^3/\mu L$)			222.62 (133.0 - 278.0)

Table 5.B.2: Training data description of the data for no-AF patients on AmsterdamUMCdb. ABP = Arterial blood pressure. ABG = Arterial blood gas.

Feature - AF - AmsterdamUMCdb	Model-1.5	Model-6	Model-12
Age	70.89 (69.0 - 75.0)	70.73 (69.0 - 75.0)	70.57 (69.0 - 75.0)
Has received noradrenalin	50%	51%	52%
Max heart frequency (bpm)	107.17 (90.0 - 120.0)	104.44 (89.0 - 117.0)	
Mean PEEP setting (mmHg)	6.18 (0.0 - 10.0)	6.4 (0.0 - 10.0)	6.72 (0.0 - 10.0)
Mean CVP (mmHg)	9.5 (6.67 - 12.0)		-1355.04 (7.0 - 12.13)
Mean ventilator administered FiO2 (%)	47.96 (40.0 - 53.64)	47.41 (40.0 - 51.61)	48.05 (40.0 - 53.33)
Min thrombocytes ($10^3/\mu L$)	180.73 (97.0 - 230.0)		
ICU admission Urgency	30%	30%	31%
Mean hourly urinary volume (mL/h)	105.99 (46.82 - 145.34)	104.29 (45.65 - 141.12)	
Has received loop diuretics	23%	22%	22%
Mean blood ureum (mmol/L)	13.8 (7.9 - 16.75)		
Slope base excess (mmol/L)	-0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	0.0 (-0.02 - 0.02)		
Mean calc. O2 saturation on ABG (%)	0.95 (0.94 - 0.98)	0.95 (0.94 - 0.98)	1.02 (0.94 - 0.98)
Fluid Balance (mL)	1551.97 (549.9 - 2410.1)	1586.96 (623.3 - 2416.8)	
Min CVP (mm Hg)		8.24 (5.0 - 11.0)	
Slope mean ABP (mm Hg)		0.0 (-0.01 - 0.02)	
Max Phosphate (mmol/L)		1.24 (0.9 - 1.52)	
Max systolic ABP (mmHg)		166.66 (140.0 - 183.0)	167.06 (138.0 - 183.0)
Min Lactate (mmol/L)		2.12 (1.1 - 2.4)	
Min arterial pH on blood gas			7.35 (7.3 - 7.41)
Max hourly urinary volume (mL/h)			190.59 (80.0 - 270.0)
Mean heart frequency (bpm)			87.24 (74.64 - 98.91)
Min Phosphate (mmol/L)			1.24 (0.88 - 1.52)
Has received Propofol (Diprivan)			32%
Max thrombocytes ($10^3/\mu L$)			184.34 (100.25 - 234.75)

Table 5.B.3: Training data description of the data for AF patients on AmsterdamUMCdb as mean (IQR). ABP = Arterial blood pressure. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - GUH	Model-1.5	Model-6	Model-12
Age	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Has received noradrenalin	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Max heart frequency (bpm)	0.0 - 0.0	0.0 - 0.0	
Mean PEEP setting (mmHg)	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean CVP (mmHg)	10.84 - 40.88		10.95 - 43.33
Mean ventilator administered FiO2 (%)	6.04 - 4.66	6.83 - 3.97	6.19 - 4.6
Min thrombocytes ($10^3/\mu L$)	22.22 - 33.47		
ICU admission Urgency	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean hourly urinary volume (mL/h)	0.14 - 0.69	0.32 - 0.16	
Has received loop diuretics	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean blood ureum (mmol/L)	23.18 - 33.74		
Slope base excess (mmol/L)	4.66 - 17.83		
Slope systolic ABP (mmHg)	4.12 - 13.03		
Mean calc. O2 saturation on ABG (%)	3.98 - 14.68	3.02 - 14.29	3.33 - 13.81
Fluid Balance (mL)	0.0 - 0.0	0.0 - 0.0	
Min CVP (mmHg)		10.16 - 41.59	
Slope mean ABP (mmHg)		2.22 - 12.06	
Max Phosphate (mmol/L)		25.71 - 39.52	
Max systolic ABP (mmHg)		2.06 - 11.27	2.38 - 10.48
Min Lactate (mmol/L)		1.11 - 7.3	
Min arterial pH on blood gas			2.38 - 10.63
Max hourly urinary volume (mL/h)			0.48 - 0.79
Mean heart frequency (bpm)			0.0 - 0.16
Min Phosphate (mmol/L)			22.22 - 41.9
Has received Propofol (Diprivan)			0.0 - 0.0
Max thrombocytes ($10^3/\mu L$)			21.75 - 42.06

Table 5.B.4: Percentage missing values for each feature for each model on the training data for GUH as mean (IQR). Structured as for no-AF - AF data. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - no-AF - GUH	Model-1.5	Model-6	Model-12
Age	59.54 (50.81 - 70.94)	61.72 (52.33 - 73.28)	60.02 (51.3 - 72.17)
Has received noradrenalin	21%	23%	26%
Max heart frequency (bpm)	92.2 (80.0 - 102.0)	92.53 (80.0 - 103.0)	
Mean PEEP setting (mmHg)	1.51 (0.0 - 0.0)	1.35 (0.0 - 0.0)	1.7 (0.0 - 5.0)
Mean CVP (mmHg)	10.39 (7.14 - 13.13)		10.0 (6.6 - 13.0)
Mean ventilator administered FiO2 (%)	33.36 (24.78 - 36.0)	33.77 (25.33 - 38.33)	34.13 (24.33 - 39.2)
Min thrombocytes ($10^3/\mu L$)	193.97 (124.0 - 239.0)		
ICU admission Urgency	38%	34%	33%
Mean hourly urinary volume (mL/h)	147.78 (65.73 - 192.17)	141.25 (67.5 - 183.33)	
Has received loop diuretics	18%	20%	14%
Mean blood ureum (mmol/L)	47.21 (24.0 - 58.0)		
Slope base excess (mmol/L)	0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	0.25 (-0.02 - 0.03)		
Mean calc. O2 saturation on ABG (%)	94.15 (93.87 - 96.2)	94.14 (93.61 - 96.05)	94.42 (93.8 - 96.1)
Fluid Balance (mL)	353.9 (-129.64 - 775.83)	416.37 (-136.52 - 856.89)	
Min CVP (mm Hg)		7.26 (3.0 - 10.0)	
Slope mean ABP (mmHg)		0.0 (-0.01 - 0.01)	
Max Phosphate (mmol/L)		1.14 (0.86 - 1.33)	
Max systolic ABP (mmHg)		399.11 (132.0 - 163.5)	382.94 (133.0 - 165.0)
Min Lactate (mmol/L)		8.81 (5.6 - 11.0)	
Min arterial pH on blood gas			7.41 (7.37 - 7.45)
Max hourly urinary volume (mL/h)			272.55 (120.0 - 400.0)
Mean heart frequency (bpm)			83.75 (71.5 - 92.92)
Min Phosphate (mmol/L)			1.15 (0.84 - 1.36)
Has received Propofol (Diprivan)			13%
Max thrombocytes ($10^3/\mu L$)			198.53 (111.0 - 252.0)

Table 5.B.5: Training data description of the data for no-AF patients on GUH as mean (IQR). ABP = Arterial blood pressure. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - AF - GUH	Model-1.5	Model-6	Model-12
Age	71.31 (65.52 - 78.95)	71.25 (65.48 - 79.03)	71.56 (65.72 - 79.3)
Has received noradrenalin	46%	45%	48%
Max heart frequency (bpm)	102.16 (84.0 - 114.0)	98.2 (82.0 - 109.0)	
Mean PEEP setting (mmHg)	2.43 (0.0 - 5.0)	2.36 (0.0 - 5.0)	2.75 (0.0 - 5.0)
Mean CVP (mmHg)	11.16 (8.09 - 13.5)		11.41 (8.4 - 13.55)
Mean ventilator administered FiO2 (%)	44.5 (32.0 - 54.29)	44.67 (32.0 - 54.29)	45.79 (33.25 - 56.25)
Min thrombocytes ($10^3/\mu L$)	174.91 (116.5 - 206.0)		
ICU admission Urgency	63%	63%	63%
Mean hourly urinary volume (mL/h)	98.89 (50.0 - 132.98)	96.35 (50.31 - 125.1)	
Has received loop diuretics	29%	31%	29%
Mean blood ureum (mmol/L)	64.47 (35.88 - 80.0)		
Slope base excess (mmol/L)	0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	-0.01 (-0.02 - 0.02)		
Mean calc. O2 saturation on ABG (%)	94.36 (93.69 - 96.2)	94.72 (93.8 - 96.26)	94.55 (93.72 - 96.35)
Fluid Balance (mL)	412.07 (-63.98 - 822.74)	426.62 (-54.95 - 831.44)	
Min CVP (mm Hg)		7.17 (4.0 - 10.0)	
Slope mean ABP (mmHg)		0.01 (-0.01 - 0.01)	
Max Phosphate (mmol/L)		1.37 (1.04 - 1.65)	
Max systolic ABP (mmHg)		356.33 (128.0 - 156.0)	142.61 (127.0 - 155.0)
Min Lactate (mmol/L)		8.65 (1.68 - 11.8)	
Min arterial pH on blood gas			7.39 (7.35 - 7.43)
Max hourly urinary volume (mL/h)			198.33 (80.0 - 280.0)
Mean heart frequency (bpm)			85.88 (76.0 - 95.0)
Min Phosphate (mmol/L)			1.34 (1.0 - 1.63)
Has received Propofol (Diprivan)			26%
Max thrombocytes ($10^3/\mu L$)			172.71 (122.0 - 209.0)

Table 5.B.6: Training data description of the data for AF patients on GUH as mean (IQR). ABP = Arterial blood pressure. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - MIMIC-IV	Model-1.5	Model-6	Model-12
Age	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Has received noradrenalin	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Max heart frequency (bpm)	1.12 - 0.3	1.28 - 0.28	
Mean PEEP setting (mmHg)	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean CVP (mmHg)	68.74 - 85.53		63.22 - 82.98
Mean ventilator administered FiO2 (%)	48.69 - 66.79	44.69 - 63.7	40.98 - 61.18
Min thrombocytes ($10^3/\mu L$)	38.81 - 40.27		
ICU admission Urgency	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean hourly urinary volume (mL/h)	18.41 - 31.99	17.6 - 31.71	
Has received loop diuretics	0.0 - 0.0	0.0 - 0.0	0.0 - 0.0
Mean blood ureum (mmol/L)	29.42 - 35.08		
Slope base excess (mmol/L)	77.57 - 91.98		
Slope systolic ABP (mmHg)	51.45 - 71.4		
Mean calc. O2 saturation on ABG (%)	76.14 - 93.36	74.26 - 91.79	71.82 - 89.71
Fluid Balance (mL)	98.93 - 99.04	98.85 - 99.26	
Min CVP (mmHg)		66.25 - 84.34	
Slope mean ABP (mmHg)		47.68 - 68.78	
Max Phosphate (mmol/L)		39.85 - 36.73	
Max systolic ABP (mmHg)		46.89 - 67.93	44.13 - 67.01
Min Lactate (mmol/L)		64.9 - 82.09	
Min arterial pH on blood gas			42.06 - 67.44
Max hourly urinary volume (mL/h)			16.32 - 30.2
Mean heart frequency (bpm)			1.02 - 0.38
Min Phosphate (mmol/L)			39.9 - 44.04
Has received Propofol (Diprivan)			0.0 - 0.0
Max thrombocytes ($10^3/\mu L$)			37.45 - 44.39

Table 5.B.7: Percentage missing values for each feature for each model on the training data for MIMIC-IV. Structured as no-AF - AF data. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - no-AF - MIMIC-IV	Model-1.5	Model-6	Model-12
Age	60.91 (51.0 - 73.0)	60.49 (50.0 - 73.0)	60.97 (51.0 - 73.0)
Has received noradrenalin	4%	4%	5%
Max heart frequency (bpm)	94.23 (82.0 - 105.75)	94.22 (82.0 - 105.0)	
Mean PEEP setting (mmHg)	6.21 (5.0 - 8.0)	6.53 (5.0 - 8.0)	6.4 (5.0 - 8.0)
Mean CVP (mmHg)	12.39 (6.44 - 12.78)		15.46 (7.14 - 13.56)
Mean ventilator administered FiO2 (%)	47.87 (40.0 - 50.0)	47.51 (40.0 - 50.0)	47.4 (40.0 - 50.0)
Min thrombocytes ($10^3/\mu L$)	199.15 (124.0 - 250.0)		
ICU admission Urgency	75%	74%	75%
Mean hourly urinary volume (mL/h)	130.27 (61.67 - 165.93)	131.18 (61.36 - 166.67)	
Has received loop diuretics	7%	7%	7%
Mean blood ureum (mmol/L)	25.25 (12.0 - 30.0)		
Slope base excess (mmol/L)	0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	0.0 (-0.02 - 0.02)		
Mean calc. O2 saturation on ABG (%)	85.14 (76.0 - 97.0)	84.87 (74.75 - 97.0)	87.45 (79.8 - 97.0)
Fluid Balance (mL)	460.97 (-585.39 - 1097.44)	74.08 (-622.51 - 533.5)	
Min CVP (mm Hg)		7.37 (3.0 - 9.0)	
Slope mean ABP (mmHg)		0.01 (-0.01 - 0.02)	
Max Phosphate (mmol/L)		3.5 (2.7 - 4.0)	
Max systolic ABP (mmHg)		139.35 (123.0 - 153.0)	139.83 (125.0 - 154.25)
Min Lactate (mmol/L)		2.01 (1.1 - 2.2)	
Min arterial pH on blood gas			7.38 (7.34 - 7.43)
Max hourly urinary volume (mL/h)			248.56 (110.0 - 350.0)
Mean heart frequency (bpm)			83.89 (72.53 - 94.36)
Min Phosphate (mmol/L)			3.4 (2.6 - 4.0)
Has received Propofol (Diprivan)			7%
Max thrombocytes ($10^3/\mu L$)			197.72 (122.0 - 248.0)

Table 5.B.8: Training data description of the data for no-AF patients on MIMIC-IV as mean (IQR). ABP = Arterial blood pressure. ABG = Arterial blood gas. calc. O2 = Calculated O2

Feature - AF - MIMIC-IV	Model-1.5	Model-6	Model-12
Age	71.01 (63.0 - 80.0)	70.84 (63.0 - 80.0)	70.68 (63.0 - 80.0)
Has received noradrenalin	10%	11%	11%
Max heart frequency (bpm)	99.29 (85.0 - 110.0)	96.93 (83.0 - 107.0)	
Mean PEEP setting (mmHg)	6.97 (5.0 - 8.5)	6.86 (5.0 - 8.19)	6.97 (5.0 - 8.67)
Mean CVP (mmHg)	15.52 (8.14 - 15.0)		15.09 (8.42 - 15.43)
Mean ventilator administered FiO2 (%)	50.66 (40.0 - 55.0)	50.12 (40.0 - 55.0)	50.62 (40.0 - 56.67)
Min thrombocytes ($10^3/\mu L$)	168.82 (102.0 - 215.0)		
ICU admission Urgency	69%	69%	68
Mean hourly urinary volume (mL/h)	102.73 (45.91 - 134.13)	99.83 (46.24 - 131.25)	
Has received loop diuretics	15%	14%	14%
Mean blood ureum (mmol/L)	34.12 (17.0 - 43.0)		
Slope base excess (mmol/L)	-0.0 (-0.0 - 0.0)		
Slope systolic ABP (mmHg)	-0.01 (-0.02 - 0.02)		
Mean calc. O2 saturation on ABG (%)	85.03 (77.0 - 96.5)	85.2 (77.0 - 96.5)	85.22 (78.0 - 96.37)
Fluid Balance (mL)	856.34 (-480.72 - 1649.23)	976.4 (-202.58 - 1610.47)	
Min CVP (mm Hg)		8.76 (4.0 - 11.0)	
Slope mean ABP (mmHg)		0.07 (-0.01 - 0.01)	
Max Phosphate (mmol/L)		3.96 (2.9 - 4.7)	
Max systolic ABP (mmHg)		136.26 (122.0 - 147.0)	135.97 (121.0 - 148.0)
Min Lactate (mmol/L)		2.07 (1.2 - 2.3)	
Min arterial pH on blood gas			7.37 (7.32 - 7.42)
Max hourly urinary volume (mL/h)			207.42 (80.0 - 300.0)
Mean heart frequency (bpm)			85.14 (74.49 - 94.4)
Min Phosphate (mmol/L)			3.84 (2.8 - 4.5)
Has received Propofol (Diprivan)			12%
Max thrombocytes ($10^3/\mu L$)			173.95 (107.0 - 217.0)

Table 5.B.9: Training data description of the data for AF patients on MIMIC-IV as mean (IQR). ABP = Arterial blood pressure. ABG = Arterial blood gas. calc. O2 = Calculated O2

References

[1] Takuo Yoshida, Tomoko Fujii, Shigehiko Uchino, and Masanori Takinami. *Epidemiology, prevention, and treatment of new-onset atrial fibrillation in critically ill: a systematic review*. Journal of Intensive Care, 3(1), April 2015. ZSCC: 0000062.

[2] Jason W. Greenberg, Timothy S. Lancaster, Richard B. Schuessler, and Spencer J. Melby. *Postoperative atrial fibrillation following cardiac surgery: a persistent complication*. European Journal of Cardio-Thoracic Surgery, 52(4):665–672, October 2017. ZSCC: 0000138.

[3] Peter M. C. Klein Klouwenberg, Jos F. Frencken, Sanne Kuipers, David S. Y. Ong, Linda M. Peelen, Lonneke A. van Vught, Marcus J. Schultz, Tom van der Poll, Marc J. Bonten, and Olaf L. Cremer. *Incidence, Predictors, and Outcomes of New-Onset Atrial Fibrillation in Critically Ill Patients with Sepsis. A Cohort Study*. American Journal of Respiratory and Critical Care Medicine, 195(2):205–211, January 2017.

[4] Travis J. Moss, James Forrest Calland, Kyle B. Enfield, Diana C. Gomez-Manjarres, Caroline Ruminski, John P. DiMarco, Douglas E. Lake, and J. Randall Moorman. *New-Onset Atrial Fibrillation in the Critically Ill*. Critical Care Medicine, 45(5):790–797, May 2017. ZSCC: 0000096.

[5] Wei Zhang, Weiling Liu, Sophia T. H. Chew, Liang Shen, and Lian Kah Ti. *A Clinical Prediction Model for Postcardiac Surgery Atrial Fibrillation in an Asian Population*. Anesthesia and Analgesia, 123(2):283–289, August 2016.

[6] Alvaro Alonso, Bouwe P. Krijthe, Thor Aspelund, Katherine A. Stepas, Michael J. Pencina, Carlee B. Moser, Moritz F. Sinner, Nona Sotoodehnia, João D. Fontes, A. Cecile J. W. Janssens, Richard A. Kronmal, Jared W. Magnani, Jacqueline C. Witteman, Alanna M. Chamberlain, Steven A. Lubitz, Renate B. Schnabel, Sunil K. Agarwal, David D. McManus, Patrick T. Ellinor, Martin G. Larson, Gregory L. Burke, Lenore J. Launer, Albert Hofman, Daniel Levy, John S. Gottdiener, Stefan Kääb, David Couper, Tamara B. Harris, Elsayed Z. Soliman, Bruno H. C. Stricker, Vilmundur Gudnason, Susan R. Heckbert, and Emelia J. Benjamin. *Simple Risk Model Predicts Incidence of Atrial Fibrillation in a Racially and Geographically Diverse Population: the CHARGE AF Consortium*. Journal of the American Heart Association, 2(2):e000102, March 2013. Publisher: American Heart Association.

[7] Patrick J. Thoral, Jan M. Peppink, Ronald H. Driessen, Eric J. G. Sijbrands, Erwin J. O. Kompanje, Lewis Kaplan, Heatherlee Bailey, Jozef Kesecioglu, Maurizio Cecconi, Matthew Churpek, Gilles Clermont, Mihaela van der Schaar, Ari Ercole, Armand R. J. Girbes, and Paul W. G. Elbers. *Sharing ICU Patient Data Responsibly Under the Society of Critical Care Medicine/European Society of Intensive Care Medicine*

*Joint Data Science Collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) Example**. Critical Care Medicine, 49(6):e563, June 2021. ZSCC: NoCitationData[s0].

[8] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. *CatBoost: gradient boosting with categorical features support*. arXiv:1810.11363 [cs, stat], October 2018. ZSCC: 0000342 arXiv: 1810.11363.

[9] Jarne Verhaeghe, Jeroen Van Der Donckt, Femke Ongenae, and Sofie Van Hoecke. *Powershap: A Power-full Shapley Feature Selection Method*, June 2022. arXiv:2206.08394 [cs, stat].

[10] Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. *MIMIC-IV*, 2022. Version Number: 0.4 Type: dataset.

[11] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On Calibration of Modern Neural Networks*, August 2017. arXiv:1706.04599 [cs].

[12] Jarne Verhaeghe, Sofie A. M. Dhaese, Thomas De Corte, David Vander Mijnsbrugge, Heleen Aardema, Jan G. Zijlstra, Alain G. Verstraete, Veronique Stove, Pieter Colin, Femke Ongenae, Jan J. De Waele, and Sofie Van Hoecke. *Development and evaluation of uncertainty quantifying machine learning models to predict piperacillin plasma concentrations in critically ill patients*. BMC Medical Informatics and Decision Making, 22(1):224, August 2022.

[13] Melissa Assel, Daniel D. Sjoberg, and Andrew J. Vickers. *The Brier score does not evaluate the clinical utility of diagnostic tests or prediction models*. Diagnostic and Prognostic Research, 1(1):19, December 2017.

[14] Scott M Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

[15] Sandra Ortega-Martorell, Mark Pieroni, Brian W. Johnston, Ivan Olier, and Ingeborg D. Welters. *Development of a Risk Prediction Model for New Episodes of Atrial Fibrillation in Medical-Surgical Critically Ill Patients Using the AmsterdamUMCdb*. Frontiers in Cardiovascular Medicine, 9, 2022.

[16] Syed Khairul Bashar, Eric Y. Ding, Allan J. Walkey, David D. McManus, and Ki H. Chon. *Atrial Fibrillation Prediction from Critically Ill Sepsis Patients*. Biosensors, 11(8):269, August 2021.

[17] Ali Narin, Yalcin Isler, Mahmut Ozer, and Matjaž Perc. *Early prediction of paroxysmal atrial fibrillation based on short-term heart rate variability*. Physica A: Statistical Mechanics and its Applications, 509:56–65, November 2018.

[18] Cynthia Rudin. *Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead.* Nature Machine Intelligence, 1(5):206–215, May 2019. Number: 5 Publisher: Nature Publishing Group.

[19] Lucas M. Fleuren, Patrick Thoral, Duncan Shillan, Ari Ercole, Paul W. G. Elbers, Mark Hoogendoorn, Ben Gibbison, Thomas L. T. Klausch, Tingjie Guo, Luca F. Roggeveen, Eleonora L. Swart, Armand R. J. Girbes, and Right Data Right Now Collaborators. *Machine learning in intensive care medicine: ready for take-off?* Intensive Care Medicine, 46(7):1486–1488, July 2020.

[20] *NannyML (release 0.12.1).* https://github.com/NannyML/nannyml, March 2023. NannyML, Belgium, OHL.

[21] Eric Y. Ding, Daniella Albuquerque, Michael Winter, Sophia Binici, Jaclyn Piche, Syed Khairul Bashar, Ki Chon, Allan J. Walkey, and David D. McManus. *Novel Method of Atrial Fibrillation Case Identification and Burden Estimation Using the MIMIC-III Electronic Health Data Set.* Journal of Intensive Care Medicine, 34(10):851–857, October 2019.

[22] César A. Millán, Nathalia A. Girón, and Diego M. Lopez. *Analysis of Relevant Features from Photoplethysmographic Signals for Atrial Fibrillation Classification.* International Journal of Environmental Research and Public Health, 17(2), January 2020. ZSCC: 0000005.

[23] Thilo Rieg, Janek Frick, Hermann Baumgartl, and Ricardo Buettner. *Demonstration of the potential of white-box machine learning approaches to gain insights from cardiovascular disease electrocardiograms.* PLoS ONE, 15(12), December 2020.

6

Causalteshap: Discerning Predictive from Prognostic Features for Treatment Effect Analysis.

Understanding the factors that influence the success or failure of treatment is crucial for providing effective treatment advice. A treatment effect model, which predicts outcomes with or without treatment, is a valuable tool for analyzing these effects. This chapters draws inspiration from Powershap in Chapter 3, by introducing a method, called Causalteshap, that incorporates a random feature into a treatment effect model, such as an S-learner, and compares the Shapley values of this random feature with those of other features using statistical testing. Causalteshap, is rigorously evaluated on a synthetic benchmark designed to differentiate between predictive and prognostic variables, as well as on multiple semi-synthetic benchmarks. The results demonstrate that Causalteshap effectively identifies predictive variables without generating false positives, outperforming other methods and proving to be a potentially valuable tool for informing treatment advice. As such, this chapter focuses on tackling RG3. To further validate this, Causalteshap is applied to the atrial fibrillation (AF) models presented in Chapter 5 to analyze which variables increase or decrease the effect of Noradrenaline on the predicted risk of AF, given its importance as a predictor for AF. My contributions to this chapter are the design and development of the complete method, the experiments, the writing, and the design.

Causalteshap: Discerning Predictive from Prognostic Features for Treatment Effect Analysis

Jarne Verhaeghe, Femke Ongenae, Sofie Van Hoecke

Published in International Journal of Machine Learning and Cybernetics

Abstract Treatment effect analysis investigates the effect of a treatment or intervention. The variables that will determine the treatment effect are called, predictive variables, while prognostic variables determine the outcome regardless of treatment, based on existing conditions on characteristics. The identification of these predictive factors facilitates understanding the treatment effect and even allows for improving its success. However, in many cases, the predictive factors of a treatment or intervention are unknown. Furthermore, methods to find these predictive factors are limited and only focus on quantifying the predictive performance of a CATE estimator instead of discerning predictive from prognostic variables. Therefore, to find these predictive variables we present *Causalteshap*. *Causalteshap* is a Shapley-based method that leverages multiple statistical tests and treatment effect estimators to discern prognostic from predictive features. The method is benchmarked on multiple fully synthetic datasets and four semi-synthetic datasets. In most of these benchmarks, *Causalteshap* demonstrates high precision and recall performances above 0.9. Subsequently, *Causalteshap* is applied to a real-world ICU use case using the AmsterdamUMCdb dataset. We analyzed the effect of Noradrenaline on Atrial Fibrillation in the ICU to display the potential of *Causalteshap* as a tool for treatment effect analysis. Our results demonstrate that *Causalteshap* has the potential of combining treatment effect estimators with Shapley values and statistical tests to provide a novel method for discerning predictive from prognostic features in treatment effect analysis and making understanding treatment effects more accessible.

6.1 Introduction

Causal thinking is a vital part of what makes humans inventive and creative. Reasoning about hypothetical worlds and understanding the effects of actions is what makes us unique [1]. However, this cannot yet be said about Artificial Intelligence (AI). Although, efforts are being made to reach this status using causal AI and causal inference, both emerging subdomains within AI and machine learning [2]. Their significance for AI lies in their potential to explain, interpret, analyze, and pave the way for groundbreaking research and applications. The primary goal of causal inference is to unravel causal relationships between variables. In contrast to traditional AI or

machine learning (ML), causal AI or causal ML focuses on leveraging machine learning models to estimate and understand causal relationships. This approach enables informed decision-making, accurate predictions, and the ability to answer 'what-if' questions by identifying the underlying mechanisms that drive outcomes [3]. Causal AI is paramount for clinical trials, policy formulation, and marketing strategies by harnessing causal knowledge to generalize predictions, provide insights, and guide decision-making processes [1].

The estimation and analysis of the impact of treatments or actions on a specific outcome is a branch of causal AI, defined as treatment effect analysis [4]. Treatment effect analysis has large importance in healthcare, policy, and marketing, to understand which actions or treatments will be the most beneficial. This involves quantifying the impact of an action, such as the introduction of a policy by a government or the administration of a medication to a patient, on a particular outcome [2]. Fully understanding treatment effects can facilitate providing personalized healthcare, personalized marketing, or fully targeted policy-making.

A key component of understanding and interpreting the treatment effect is identifying predictive features (also called predictive covariates) that can influence it [5]. Predictive features are variables that increase or decrease the treatment effect, and their identification can provide valuable insights into how the system operates and can even be improved. In contrast, prognostic features are features that contribute to the outcome but their contribution is not influenced by treatment. This distinction can be formally described as $y = y_{prog} + T \cdot y_{pred}$, where prognostic features y_{prog} fully characterize the treatment-independent outcome, and predictive features y_{pred} determine the treatment effect, activated only when treatment is present. Predictive features are modelled as parents of mediators that transmit treatment effects, while prognostic features influence outcomes directly, independent of treatment. However, a feature can be both as well. Recognizing this distinction is crucial for effective treatment strategies in various domains.

For example, in clinical trials, predictive features could include a patient's age, gender, and medical history, which influence the effectiveness of a treatment. Prognostic features in the same context might be demographic factors like baseline health status or genetic predispositions that affect outcomes regardless of treatment. It is also possible for features to be both predictive and prognostic, such as Age. An example Directed Acyclic Graph (DAG) or Causal graph is presented in Figure 6.1.1. In this example, it might be, for example, more beneficial to give the treatment to patients with a specific medical history as the treatment will be more effective compared to those who do not have this medical history. Similarly, in marketing campaigns, predictive features might encompass customer demographics, buying habits, and prefer-

ences that determine the success of a promotion, while prognostic features relate to consistent purchasing behaviour unaffected by marketing efforts. A feature can be both predictive and prognostic, therefore, it is important to know if there is a predictive component as only the predictive features can help tailor treatments to individual characteristics, in turn optimizing the chances of favourable outcomes. Additionally, access to predictive features in causal AI can enhance the interpretation and impact of causal AI models.

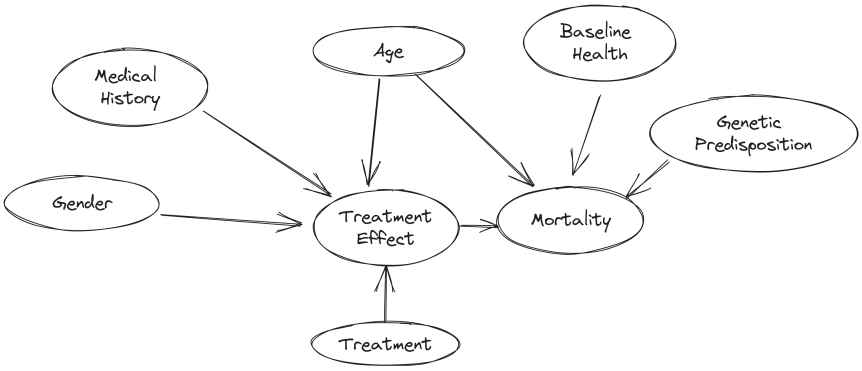


Figure 6.1.1: Directed Acyclic graph or causal graph of the example. The predictive features are Gender, Medical History, and Age. The prognostic features are Age, Baseline Health, and Genetic Predisposition. This is merely an illustrative example and does not necessarily reflect the real world.

Therefore, in this paper, we present a method, named *Causalteshap* (Causal Treatment Effect Shapley values), that fully focuses on identifying predictive features for treatment effect models. Our contributions are as follows:

- We propose a novel method leveraging Shapley values, based on meta-learners and combined statistical tests to differentiate predictive features from purely prognostic ones, providing a model-agnostic method that applies to any gradient-boosting model while generalizing to other ML models supporting Shapley values.
- We demonstrate the effectiveness of *Causalteshap* through extensive experiments and compare them to other methods on newly proposed and expanded synthetic and semi-synthetic datasets for identifying predictive features.
- We showcase a real-world use case using *Causalteshap*, highlighting the benefit of discerning predictive features from prognostic and how this can impact future research.

With this work, we aim to position *Causalteshap* as a valuable method for distinguishing predictive features from purely prognostic ones in causal AI and treatment effect analysis.

6.2 Background

6.2.1 The Potential Outcome Framework for Treatment Effects

The potential outcomes framework, introduced by Neyman and Rubin, provides a formalized framework for inference of treatment effects in causal analysis [6]. The framework is based on the idea of potential outcomes, which are the outcomes that would be observed under each possible treatment assignment. Therefore, each individual has two possible or potential outcomes (can be binary or continuous), one under treatment $Y(1)$ and one without treatment $Y(0)$. We can never observe both at the same time as only the performed action can be observed. Given the framework, it is possible to estimate the Conditional Average Treatment Effect (CATE) for an individual with covariates X = x [5]:

$$CATE(x) = \tau(x) = E[Y(1) - Y(0)|X = x] = \mu_1(x) - \mu_0(x) \tag{6.1}$$

With $\mu_T(x) = E[Y(T)|X = x]$ the expected potential outcome for treatment T, or in this case, treatment ($T = 1$) or control ($T = 0$). Setting T to either 0 or 1 is defined as intervening and tries to mimic random assignment because T is changed while other variables are not. Furthermore, only one of the outcomes can be observed, therefore the other outcome should be inferred for a given patient. Here is where machine learning can fit in. This CATE estimation requires three standard assumptions [6]:

- **Unconfoundedness**: $Y(t) \perp\!\!\!\perp T|X, \forall t \in T$. This assumption implies that, given the observed covariates, the treatment assignment is independent of the potential outcomes. In simpler terms, all confounding factors affecting both the treatment and the outcome have been accounted for, with no hidden variables influencing both.

- **Overlap** or **positivity**: $0 < P(T = t|X = x) < 1, \forall t \in T$ with $x \in X$. The overlap assumption guarantees that, for every covariate value x, there is a non-zero chance of receiving each treatment option. This is essential for reliably estimating treatment effects across all treatment groups.

- **Consistency**: $Y = Y(t)$ with probability 1. This assumption connects the observed outcomes to the potential outcomes, stating that the observed outcome matches the potential outcome associated with the treatment actually received.

Meta-learners in causal AI are frameworks in which machine learning models are used to perform CATE estimation [4, 7]. These meta-learners are popular and already widely available for CATE estimation use cases [4]. Examples are T-learners, S-learners, X-learners, and R-learners. The simplest of these meta-learners is the S-learner or Shared-model. This meta-learner only trains a single model M and predicts the outcome y, however, the treatment is added as a feature. After training, a patient

is put through the single model twice: once with $T = 1$ and once with $T = 0$. The CATE can then be estimated as: $CATE(x) = M(X, T = 1) - M(X, T = 0)$. However, if the treatment effect is small or weak, the S-learner could learn to disregard the treatment variable as it is not informative enough during training. Each meta-learner has its advantages and disadvantages and varying complexity [7].

6.2.2 Predictive and prognostic features

In classic machine learning problems, some features are informative to predict the outcome. Likewise, as there are variables that are informative for a specific outcome, there are variables that are informative to explain the treatment effect. Variables that model the prediction outcome without the influence of any treatment are called prognostic variables. On the other hand, variables that model the treatment effect of a specific treatment are called predictive variables. This is modeled as follows [8]:

$$y = y_{prog} + T \cdot y_{pred} \tag{6.2}$$

Here, the predictive features fully characterize the treatment effect. The prognostic variables fully characterize the treatment-independent outcome. These predictive features can be interpreted as determinants of the treatment susceptibility that only get activated when the treatment is given. This interpretation is visualized using a Directed Acyclic Graph, called a causal graph, in Figure 6.2.1. In the graph, only the *prognostic*, *predictive*, Y, and *treatment* variables are observable. Do note that there can also be a direct path from *predictive* to Y as variables can be both predictive and prognostic, this causal graph simply visualizes the prognostic versus predictive variable theory. Another interpretation of the graph is that giving the treatment or not is separated from its treatment effect. This treatment effect can be seen as the mediator through which the treatment achieves its effect. This effect is also dependent on variables that determine the treatment susceptibility, which can be deemed the predictive variables. Hence, predictive variables are thus defined as parents of mediators.

Knowing these predictive features facilitates explaining and understanding the success or failure of treatment. Furthermore, if these variables are modifiable, these can also increase the chance of success for specific treatments, enable targeting a sub-population that benefits more from it, or avoid a sub-population that suffers from the treatment.

6.2.3 Shapley values

Determining whether a feature is predictive or prognostic requires quantifying its feature contribution when $T = 1$ and $T = 0$. There are various methods to find and explain the contributions of variables to the predictions of machine learning models. A popular technique to explain model predictions is SHAP [9]. SHAP aims at

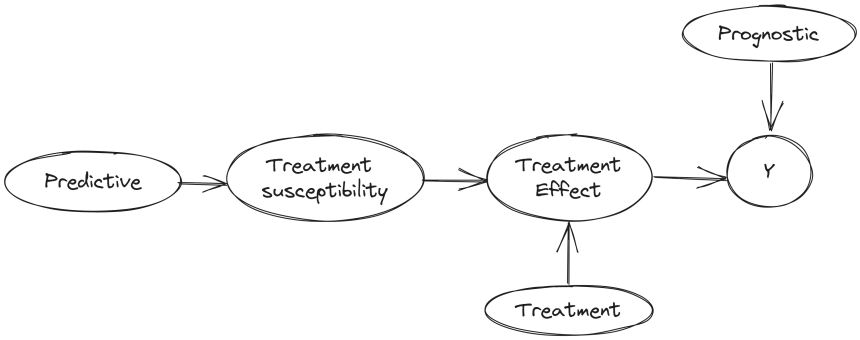


Figure 6.2.1: Theoretic causal graph of the predictive and prognostic feature theory. Y is the measured outcome

quantifying the marginal contribution of each feature to the output prediction. This is defined as the average marginal contribution of a feature to all possible combinations of features. The so-called Shapley values measure how much each feature contributes to the prediction when considered in combination with other features [9]. Mathematically Shapley values try to reconstruct the prediction using their contributions. Given a trained estimator M on data X, with K features and N samples, and $M(X_i)$ being the prediction of M on X_i, we first define the baseline Shapley value S_b as the mean prediction over X:

$$S_b = \frac{1}{N} \sum_{i=1}^{N} M(X_i) \tag{6.3}$$

We then define the Shapley values for each sample and each feature $S(X_i^k, M)$ as the marginal additive contribution of feature k in sample X_i in the prediction $M(X_i)$ compared to the mean prediction of baseline Shapley value S_b. Now, given S_b and $S(X_i, M)$ we can define the local accuracy of Shap:

$$M(X_i) = S_b + \sum_{j=1}^{K} S(X_i^k, M) \tag{6.4}$$

The method is model-agnostic and available for various models, e.g., linear, kernel-based, deep learning, and tree-based models. Although SHAP suffers from shortcomings, such as the TreeExplainer variant (i.e. the SHAP method to explain ensemble tree models) providing non-zero Shapley values to noise features, it is technically strong and very popular [10].

6.2.4 Related Work

In the causal machine learning field, only limited work is available to provide interpretations to treatment effect estimators [5]. Even though the literature on finding predictive variables is limited in the causal machine learning field, the concept is more explored in the statistical field under interaction effects and subgroup analysis [11–13]. Several studies have attempted to identify subgroups with differential treatment effects, mentioning distinguishing between prognostic and predictive variables.

In early work, Bonetti and Gelber [14] introduced the STEPP (Subpopulation Treatment Effect Pattern Plot) method, which identifies subgroups with distinct treatment effects on the outcome. Their approach is mainly for clinical trials, requiring predefined subpopulation regions, making it heavily parameter-dependent. Similarly, Lipkovich et al. [15] proposed the SIDES (Subgroup Identification based on Differential Effect Search) method, which recursively partitions data to determine subgroups with significant treatment differences, extending ideas from interaction trees. The features used in splits could be seen as predictive features. However, their method does not scale well with the number of features and data.

Foster et al. [16] explicitly mentioned the distinction between prognostic and predictive roles in subgroup analysis and tried to find them using a Virtual Twins framework. This method, an early form of a meta-learner, estimates treatment effects via a two-step modelling approach. In the first step, they train an S-learner with either a random forest or a linear regression model and then fit a new model on the pseudo outcomes. However, their focus was mainly on finding subgroups, not on finding the predictive variables. Cai et al. [17] also mentioned that finding predictive features is an important issue for covariate selection, however, the conclusions depend on the underlying model (mis)specification. Doove et al. [18] developed QUINT, a tree-based method that partitions patients into three groups: those who benefit more from treatment A, those who benefit more from treatment B, and those with no significant difference. However, QUINT is computationally expensive and only applicable to randomized controlled trials (RCTs) without handling confounders. A solution for high-dimensional settings was proposed by Guo et al. [19] which applied bootstrapped sparse logistic regression models to identify vulnerable subgroups in electronic health record data.

Instead of directly searching for subgroups, several other studies have focused on statistical and machine-learning methods for identifying treatment-interaction effects. For example, Tian et al. [20] proposed a modified covariate method (MCM) that transforms the features to better estimate the treatment-interaction effects in linear models. Consequently, the coefficients of these models can then be interpreted to understand which features influence the treatment. Park et al. [21] proposed a solution with sparse additive models (SAM) for high-dimensional problems to find treatment-interaction effects. The estimated nonzero components in the SAM model can then be seen as

predictive features.

Within causal machine learning, Hermansson et al. [8] analyzed causal trees and virtual twins for ranking predictive features using gradient boosting trees but did not propose a method to separate predictive from prognostic variables. Crabbe et al [5] introduced the ITErpretability framework, evaluating treatment effect models on their accuracy and ability to attribute prognostic and predictive contributions. However, they focused on attribution measurement rather than directly separating predictive variables. Besides these works, the literature for finding predictive variables is limited in causal machine learning.

While the above methods focus on distinguishing prognostic and predictive features, another line of work has focused on selecting the minimal adjustment set required for unbiased treatment effect estimation. These methods aim to control for confounding rather than identifying features that modify treatment response. For example, De Luna and Waernbaum [22] addressed covariate selection for nonparametric treatment effect estimation, providing minimal adjustment sets in the Neyman-Rubin framework. Similarly, Cheng et al. [23] refined treatment effect estimation by using local search algorithms to determine optimal adjustment sets even in the presence of hidden variables. However, these methods primarily target valid treatment effect estimation rather than identifying predictive variables for treatment heterogeneity. Unlike these approaches, our work focuses on discovering predictive features that influence treatment response, independent of their role as confounders.

Building upon these works, *Causalteshap* introduces a novel approach that explicitly separates predictive and prognostic variables by leveraging Shapley values in combination with statistical testing. Unlike prior methods that focus on either interaction-based subgroup discovery or tree-based importance attribution, *Causalteshap* provides a model-agnostic framework that applies to any gradient-boosting model while generalizing to other ML models supporting Shapley values. This enables more precise identification of predictive features while avoiding the scalability and parametric limitations of prior approaches. Compared to treatment subgroup methods, *Causalteshap* focuses solely on identifying predictive features in a computationally efficient manner. Unlike previous approaches that either rely on linear models, tree-based importance measures, or strong parametric assumptions, our method remains flexible and scalable, making it applicable to a wider range of machine-learning-based meta-learners.

6.3 Methods

6.3.1 Causalteshap

Causalteshap leverages meta-learners, such as an S-learner, to find predictive features. The current algorithm employs an S-learner and is extendable to other meta-learners

with the requirement that the baseline Shapley value should be the same. First, we train an S-learner M on the data. Given Equation 6.2, the T^1-model models the potential outcome as $y_{T^1} = y_{prog} + y_{pred}$ while the T^0 models it as $y_{T^0} = y_{prog}$. Therefore, subtracting $y_{T^1} - y_{T^0}$ leaves us only with y_{pred} in the ideal case. To find these features we leverage Shapley values using the SHAP library [9] to explain the attribution of these features. For this, we define the Predictive Shapley features $S_{pred}(X)$ as follows, with S representing the function for Shapley value calculation:

$$S_{T^0}(X) = S(X, T = 0, M) \tag{6.5}$$

$$S_{T^1}(X) = S(X, T = 1, M) \tag{6.6}$$

$$S_{pred}(X) = S_{T^1}(X) - S_{T^0}(X) \tag{6.7}$$

The $T = 1$ or $T = 0$ represents performing an intervention on the data or not respectively. To avoid bias, these Shapley values are calculated on a validation set of unseen data, i.e., data that was not used to train the model M.

Ideally, a purely prognostic feature X will have $S_{pred}(X) = 0$. However, the SHAP library, especially treeSHAP, tends to attribute non-zero importance to noise, as noted in prior studies [10]. This occurs due to model-induced randomness, overfitting tendencies in complex models (e.g., gradient boosting), and the inherent variance in Shapley value estimation. Consequently, simply comparing $S_{pred}(X)$ to zero is not reliable, as random fluctuations can lead to false positives even when no true predictive effect exists.

To address this, *Causalteshap* employs a two-part approach designed to mitigate the impact of such noise. The first part tests whether the variable's effect differs between treatment groups, a necessary condition for a feature to be predictive. The second part further controls for false positives by comparing the variable's Shapley contributions to those of a known random (non-informative) feature. This composite strategy ensures that identified predictive variables are robust to both model-related noise and the stochastic nature of SHAP explanations. The two parts are as follows:

1. If the feature is purely prognostic, then the $S_{T^0}(X)$ and $S_{T^1}(X)$ distribution should have the same variance and same mean.

2. When these distributions are different and the feature X is truly prognostic, then $|S_{pred}(X_{noisy})|$ of a known noise variable X_{noisy} that contains no information should be larger or equal compared to $|S_{prog}(X)|$. This covers the cases where these differences would be caused by noise.

The following subsections explain how each part is addressed by *Causalteshap*.

6.3.1.1 Part 1: Test whether the prognostic features have the same variance & mean

To address the first part we use Welch's t-test, i.e. student t-test with unequal variance, to check for different means and the Fligner test to check for different variances. The combination of these tests compares $S_{T^1}(X)$ with $S_{T^0}(X)$ to test the following null-hypothesis:

"The Shapley values $S_{T^1}(X_{prog})$ of a prognostic feature X_{prog} when $T = 1$ should have the same mean and same variance as the Shapley values $S_{T^0}(X_{prog})$ of the feature when $T = 0$."

If the p-value of either Welch's t-test or the Fligner test is below our predefined threshold α, the hypothesis is rejected. A mathematical explanation of the Welch's t-test can be found in the appendix 6.A.1. Welch's t-test assumes that the mean of the sampled distribution is normally distributed. Given the central limit theorem and the size of the tested datasets ($N > 100$), this can be assumed to be true [24]. Therefore, the test can be applied.

The second test is the Fligner test, or the Fligner-Killeen test of homogeneity of variances, which is a distribution-free test of variances [25]. A mathematical explanation of Fligner test can be found in the appendix 6.A.2.

6.3.1.2 Part 2: Test whether the difference in distributions of a predictive feature is due to noise

To check whether the difference is caused by noise, we add a random feature sampled from a uniform distribution to the dataset. This feature provides a baseline of what $S_{pred}(X)$ should be for a pure prognostic feature and can therefore be used as a comparison. Shapley values are signed, therefore, we take the absolute value of the difference as we are only interested in the amplitude of the difference. Now, given the predictive Shapley values of the features and those of the random feature, we can statistically test whether they are predictive. This part is built on the following null hypothesis:

"The Cumulative Distribution Function (CDF) of the absolute predictive Shapley values $|S_{pred}(X_p)|$ of a prognostic feature X_p should be lower or equal to the CDF of the absolute predictive Shapley values $|S_{pred}(X_r)|$ of the random feature X_r".

To verify this hypothesis we use the Kolmogorov-Smirnov test (KS-test). As already stated, this KS-test is mainly a method to compensate for the importance attribution to the noise of TreeShap. A mathematical explanation of the Kolmogorov-Smirnov test can be found in the appendix 6.A.3.

If a feature passes both parts, i.e. significant result on both the KS-test and ei-

ther the t-test or Fligner test (that tests whether either the mean and or variance is different), we determine the feature to be predictive. In the case any of the parts fail, the feature is flagged as prognostic. Additionally, while these specific tests are well-suited to our hypotheses, they are not the only options: Any test that fulfils similar requirements, e.g. testing comparable null hypotheses with fewer assumptions, could of course be substituted.

6.3.1.3 Main algorithm

We can now present the *Causalteshap* algorithm. The complete algorithm can be found in Algorithm 5. The function returns the predictive features, given an S-Learner M, treatments $\mathbf{T}$ for n samples, data $D^{n\times m}$ with n samples and m features, all feature names $\mathbf{F}_{set}$, and the significance threshold α. *Causalteshap* is implemented in Python as an open-source plug-and-play *sklearn* compatible component[1] to enable direct usage in meta-learner machine learning pipelines [26].

This section describes the *Causalteshap* algorithm. The algorithm begins by generating an array D^n_{random} of length n, sampled from a uniform distribution over the interval $[-1, 1]$. Next, we construct the dataset $\mathbf{D}^{n\times(m+2)}$, which includes the original m features, the treatment variable T, and the random variable D^n_{random}. The dataset $\mathbf{D}$ is then split into a training set $\mathbf{D}_{train}$ and a validation set $\mathbf{D}_{val}$ based on the specified ratio ω. This results in $(1-\omega)n$ samples in $\mathbf{D}_{train}$ and ωn samples in $\mathbf{D}_{val}$. The S-learner model M is subsequently trained on $\mathbf{D}_{train}$. After training, we create two interventional datasets, $\mathbf{D}^{T^1}_{val}$ and $\mathbf{D}^{T^0}_{val}$, by setting the treatment covariate to 1 and 0, respectively. The Shapley values, $\mathbf{S}_{T^1}$ and $\mathbf{S}_{T^0}$, are then computed using the SHAP library [9] for both interventional datasets. For each feature i in $\mathbf{F}_{set}$, we then calculate the test p-values. First, representing Part 1, we calculate the Kolmogorov-Smirnov test statistic D^+ to compare the distribution of $\mathbf{S}_{pred}[i]$ with that of the random variable $\mathbf{S}_{pred}[m+2]$. Then, representing Part 2, we compute the p-values for both Welch's t-test (p_{Wt}) and the Fligner test (p_{Fl}), comparing $\mathbf{S}_{T^1}[i]$ and $\mathbf{S}_{T^0}[i]$. A feature is marked as predictive ($\mathbf{P}[i] = 1$) if the Kolmogorov-Smirnov test statistic D^+ exceeds the threshold value $Q_K(\alpha)$ (Appendix 6.A.3) and at least one of the p-values (p_{Wt} or p_{Fl}) is less than or equal to the significance threshold α. Otherwise, the feature is marked as prognostic ($\mathbf{P}[i] = 0$). Finally, the algorithm returns all features F_i for which $\mathbf{P}[i] = 1$, indicating their predictive relevance.

[1]The code, documentation, and more benchmarks can be found using the following link: https://github.com/predict-idlab/causalteshap

Algorithm 5: *Causalteshap* loop

Function *Causalteshap*($M \leftarrow$ *S-Learner,* $\mathbf{T} \leftarrow t_1, ..., t_n$, $\mathbf{D}^{n \times m} \leftarrow$ *Data,* $\mathbf{F}_{set} \leftarrow F_1, \ldots, F_m$, $\alpha \leftarrow$ *threshold, split size* ω)

- $D^n_{random} \leftarrow$ Sample n times from RandomUniform $\in [-1, 1]$
- $\mathbf{D}^{n \times m+2} \leftarrow \mathbf{D}^{n \times m} \cup T^n \cup D^n_{random}$
- $\mathbf{D}^{(1-\omega)n \times m+2}_{train}, \mathbf{D}^{\omega n \times m+2}_{val} \leftarrow$ split $(\mathbf{D}, \omega)$
- $M \leftarrow$ Fit $M(\mathbf{D}_{train})$
- $\mathbf{D}^{T^1}_{val} \leftarrow T = 1$
- $\mathbf{D}^{T^0}_{val} \leftarrow T = 0$
- $\mathbf{S}_{T^1} \leftarrow \text{SHAP}(M, \mathbf{D}^{T^1}_{val})$
- $\mathbf{S}_{T^0} \leftarrow \text{SHAP}(M, \mathbf{D}^{T^0}_{val})$
- $\mathbf{S}_{pred} \leftarrow S_{T^1} - S_{T^0}$
- $\mathbf{P} \leftarrow$ size m
- **for** $i \leftarrow 1, 2, \ldots, m$ **do**
  - $D^+ \leftarrow \sqrt{n} \cdot sup_x(F_i(\mathbf{S_{pred}}[...][\mathbf{i}]) - F_R(\mathbf{S_{pred}}[...][\mathbf{m+2}]))$
  - $p_{Wt} \leftarrow WelchsTTest(\mathbf{S_{T^1}}[...][i], \mathbf{S_{T^0}}[...][i])$
  - $p_{Fl} \leftarrow Fligner(\mathbf{S_{T^1}}[...][i], \mathbf{S_{T^0}}[...][i])$
  - **if** $D^+ > Q_K(\alpha)$ ***AND*** *(*$p_{Wt} \leq \alpha$ ***OR*** $p_{Fl} \leq \alpha$*)* **then**
    - $\mathbf{P}[i] \leftarrow 1$
  - **else**
    - $\mathbf{P}[i] \leftarrow 0$
- **return** $[F_i, \forall\, i : \mathbf{P}[i] = 1]$

6.3.2 Experiments

To objectively evaluate *Causalteshap* it is not possible to benchmark the method on real-world data as it is hard to know the true relationships of the prognostic or predictive features in many of these datasets. Therefore, *Causalteshap* is first benchmarked on fully synthetic data with predefined cases each testing different possible treatment effect relations. Afterwards, the methods are benchmarked on semi-synthetic data using a randomly generated Data Generating Process (DGP) inspired by the ITErpretability Benchmark [5]. Only predictive or prognostic features will be simulated. Non-informative features will not be included as finding which variables are informative and which ones are not is a feature selection problem and not in the scope of this paper. As predictive and prognostic features are both informative features to predict the outcome, we advise first using some high-sensitivity feature selection algorithms such as PowerShap [27] to eliminate non-informative features. Afterwards, we demonstrate how *Causalteshap* can be applied to a real-world use-case where we try to find the predictive variables of the treatment effect of Noradrenaline on the occurrence of Atrial Fibrillation (AF) in the Intensive Care Unit (ICU) using published

risk outcome models [28]. All experiments are performed using a Catboost S-learner using 1000 iterations, categorical feature T, use_best_model set to True, a *Causalteshap* split size of 0.3, and a data 80%/20% train-test split for fitting and evaluating *Causalteshap*. In all these experiments the random feature is sampled from a uniform distribution with lower and upper bounds of -1 and 1, respectively. This sampling can be changed according to knowledge of the data distribution. These experiments are evaluated on the precision and recall of finding the true predictive features.

6.3.2.1 Synthetic Benchmarks

The synthetic experiment setup is an expansion of the paper by Hermansson and Svensson [8]. In the original paper, there were 11 cases (M1 to M11), however, we expanded these to include 7 more difficult cases incorporating small and large treatment effects, as well as complex relations with predictive and prognostic features. All cases can be found in Table 6.3.1. ϵ represents the Gaussian noise with a mean of 0. Each case covers a specific scenario. Case M4 is a case that does not have any predictive variables while cases M14 and M15 cover the scenarios with very large treatment effects. Other cases such as M2, M6, M8, M9, M11, M12, M14, and M16 are scenarios with features having both a prognostic and predictive component. M9 on the other hand covers a non-linear treatment-effect relationship. In Table 6.3.2, different setups for confounding which influences treatment assignment are also presented. The T0 case is for experiments without confounding resulting in balanced treatment groups. T1 has some small confounding where treatment assignment is dependent on features x_1 and x_2 while T2 has stronger confounding and is more influenced by x_1. All variables x_i are drawn from independent normal distributions with mean 0 and standard deviation 1.

All synthetic experiments are performed with $100, 250, 500, 1000, 2500, 5000,$ and 10000 generated samples to investigate the impact of data availability on the performance. The experiments are repeated 10 times, each with different random seeds to quantify the variance of the method. The standard deviation of the added Gaussian noise is also varied from $0.001, 0.01, 0.1, 0.5, 1.0, 2.5, 5,$ to 10 to check the robustness against noise. A significance threshold of $\alpha = 0.02$ is chosen as a setting in *Causalteshap* to represent a false positive rate of 2%. *Causalteshap* will also be compared to other models capable of finding predictive variables: the modified covariate method by Tian et al. [20] and the Sparse Additive Models by Park et al. [21]. For the MCM method, we fitted an Ordinary Least Squares regression on the modified features and picked the coefficients that had an $\alpha < 0.05$. For the SAM model we took all nonzero components and determined them as predictive as mentioned in their work [21].

Case	Relationship
M1	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4)$ $+T \cdot (I(x_6 > 0) + I(x_7 > 0)) + \epsilon$
M2	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot I(x_1 > 0) + \epsilon$
M3	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot I(x_{19} > 0) + \epsilon$
M4	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + \epsilon$
M5	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4)$ $+T \cdot (x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14}) + \epsilon$
M6	$y = -1$ $+3 \cdot (I(x_0 > 0) + I(x_1 > 0) + I(x_2 > 0) + I(x_3 > 0) + I(x_4 > 0))$ $+T \cdot I(x_4 > 0) + \epsilon$
M7	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9)$ $+T \cdot I(x_{19} > 0) + \epsilon$
M8	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9) +$ $+T \cdot I(x_0 > 0) + \epsilon$
M9	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot sin(x_0) + \epsilon$
M10	$y = -1 + 10 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot I(x_0 > 0) + \epsilon$
M11	$y = -1 + (x_0 + x_1 + x_2 + x_3 + x_4)^3 + T \cdot x_0^3 + \epsilon$
M12	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot x_0 + \epsilon$
M13	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot x_{19} + \epsilon$
M14	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4)$ $+50 \cdot T \cdot (I(x_6 > 0) + I(x_7 > 0)) + \epsilon$
M15	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + 50 \cdot T \cdot x_0 + \epsilon$
M16	$y = -1 + 3 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) +$ $T \cdot (I(x_0 > 0) + 10 \cdot I(x_1 > 0) + 5 \cdot I(x_7 > 0)) + \epsilon$
M17	$y = -1 + 10 \cdot (x_0 + x_1 + x_2 + x_3 + x_4) + T \cdot I(x_5 > 0) + \epsilon$
M18	$y = -1 + (x_0 + x_1 + x_2 + x_3 + x_4)^3 + T \cdot I(x_5 > 0)^3 + \epsilon$

Table 6.3.1: The different cases for the evaluation on synthetic data

6.3.2.2 Semi-synthetic Benchmarks

Synthetic data is optimal for testing ideal performance, however, it is difficult to model realistic interactions between features, such as unobserved interdependency and confoundness. Therefore, to expand the experiments we also included a DGP evaluation using semi-synthetic data. Here, we use available features in real-world datasets and

Case	Relationship
T0	$Random(0, 2, N)$
T1	$Binomial(1, \frac{exp(0.1+0.5\cdot x_1-0.25\cdot(2+x_2))}{1+exp(0.1+0.5\cdot x_1-0.25\cdot(2+x_2))}, N)$
T2	$Binomial(1, \frac{exp(0.1+1.5\cdot x_1-0.25\cdot(2+x_2))}{1+exp(0.1+1.5\cdot x_1-0.25\cdot(2+x_2))}, N)$

Table 6.3.2: The different cases for the evaluation on synthetic data for treatment assignment generation

use these features to model an outcome where we can specify the amount of prognostic and predictive features. In this way, we have more realistic complex data while still having a ground truth for benchmarking. The DGP algorithm is shown in Algorithm 6 and follows the ITErpretability benchmark method presented by Crabbe et al. [5].

Algorithm 6: Semi-Synthetic Data generating process

Function *DGP($T^n \leftarrow$ Treatment Array, $X^{n\times m} \leftarrow X_1, ..., X_m$, $m_{prog} \leftarrow$ Amount Prognostic Features, $m_{pred} \leftarrow$ Amount Predictive Features, $\beta \leftarrow$ weight threshold, $w_{pred} \leftarrow$ Treatment Weight)*

$\alpha_{prog}^{m\times 1} \leftarrow Uniform(-1, 1, m)$
$\alpha_{pred}^{m\times 1} \leftarrow Uniform(-1, 1, m)$
$\alpha_{prog} \leftarrow [\alpha_{i,prog}$ if $|\alpha_{i,prog}| > \beta$ else 0 for $i..m]$
$\alpha_{pred} \leftarrow [\alpha_{i,pred}$ if $|\alpha_{i,pred}| > \beta$ else 0 for $i..m]$
$i_{prog}^{m_{prog}\times 1} \leftarrow$ select m_{prog} non-zero weights from α_{prog}
$i_{pred}^{m_{pred}\times 1} \leftarrow$ select m_{pred} non-zero weights from α_{pred}
$\mu_{prog}^{n\times 1} = \sum_{i\in i_{prog}}(\alpha_{i,prog} \cdot Xi)$
$\mu_{pred}^{n\times 1} = \sum_{i\in i_{pred}}(\alpha_{i,pred} \cdot Xi)$
return $\mu_{prog} + T \cdot w_{pred} \cdot \mu_{pred}$

Four datasets were used: the TCGA [29], Twins [30], News [31], and ACIC2016 [32] datasets. The number of samples is set to be equal to available data points in the dataset. For the News and ACIC datasets the 100 most varying features were selected for the DGP as these datasets have large amounts of features. The resulting datasets have 38, 64, 100, and 100 features respectively. The experiments are performed using multiple amounts of prognostic and predictive variables; 10%, 25%, 40%, and 65% of all features being prognostic and 3%, 12.5%, 28%, and 50% being predictive to test increasing data complexity. These features can overlap and are selected independently of each other. The experiments are repeated 10 times, each with different random seeds. The strength of the treatment effect w_{pred} is also varied to investigate the sensitivity of the method using four settings: $0.01, 0.1, 0.5$, and 1.0. There

is no added Gaussian noise for the DGP as the added noise should be adjusted according to the relative size of the output, w_{pred}, the number of features, and every dataset for realistic evaluation. However, as an added experiment, to understand the impact of noise on the semi-synthetic data, the News dataset is also benchmarked using $0.001, 0.01, 0.1, 1, 5$, and 10 Gaussian noise standard deviations. The News dataset has lower absolute feature values, making it the most sensitive dataset of the four for the chosen noise standard deviations.

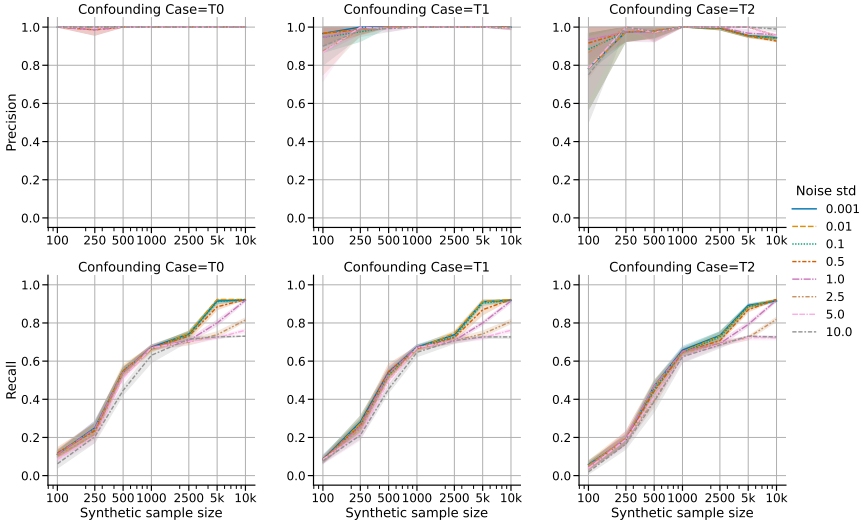
6.3.3 The effect of Noradrenaline on Atrial Fibrillation

Although benchmarking is not feasible on real-world datasets, the method can be applied to one. To test the method on a real-world use case, *Causalteshap* is applied to a treatment effect analysis of noradrenaline (= the treatment) on the occurrence of Atrial fibrillation (AF) (= the outcome) in the ICU using the AmsterdamUMCdb dataset [33]. AF is a heart rhythm disorder that causes an irregular and often abnormally fast heart rate. It affects between 4.5 to 15% of patients admitted to the intensive care unit (ICU). Several studies have indicated that the occurrence of AF in critically ill patients is associated with poorer outcomes, including prolonged length of stay (LOS) and increased hospital mortality [34, 35]. Although several risk factors for AF are non-modifiable (e.g., age), identifying patients at high risk of developing AF could allow clinicians to preemptively address modifiable risk factors (e.g., electrolyte imbalances or medication). In the study of Verhaeghe et al. [28] noradrenaline was an important predictor for the occurrence of AF in the final model and is, therefore, a candidate for causal analysis of the medication. Model-6 from the same study was used as the outcome model for the S-learner and applied to the AmsterdamUMCdb dataset [33]. This model used features aggregated from routinely collected time series data 18 hours to 6 hours before the diagnosis of AF or from the same time point as an AF patient for matched non-AF samples. The final features of the S-learner are determined using double feature selection which selects the union of the selected features of the propensity model and the selected features of the outcome model. The feature selection of both the outcome and the propensity model was performed using PowerShap [27]. Both the outcome and the propensity model use the same preprocessing method as described in the original study.

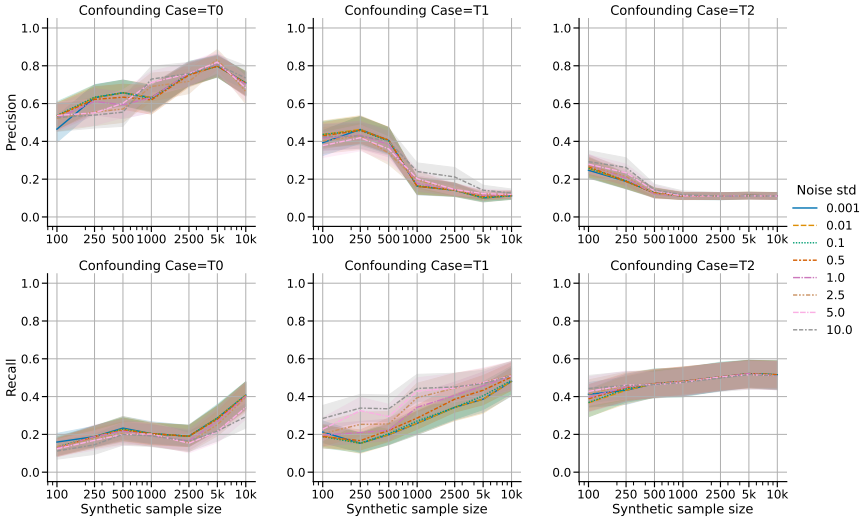
6.4 Results

6.4.1 Synthetic

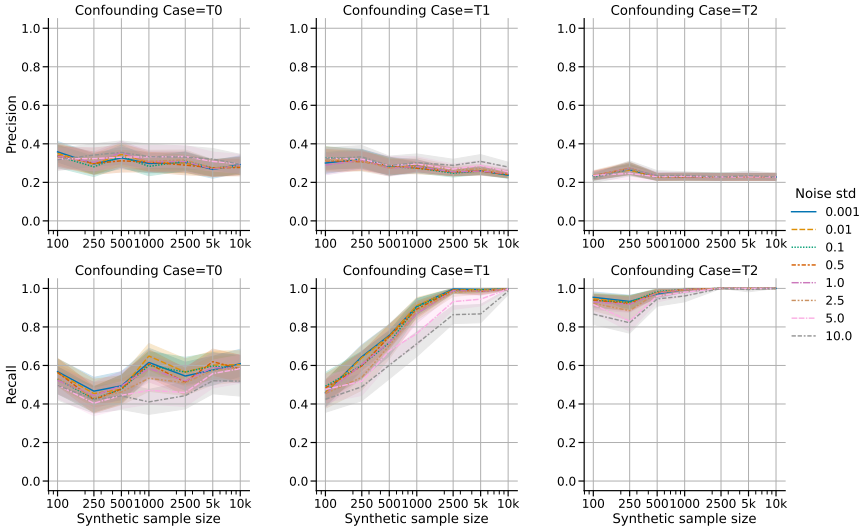
The results for the synthetic benchmarking results are shown in Figure 6.4.1. For *Causalteshap*, for all sample sizes, noise levels, and confounding levels, the precision mostly stays above 0.95. However, we see that the recall, or the capability of finding



(a) Causalteshap



(b) Modified Covariate Method (MCM) by Tian et al. [20]



(c) Sparse Additive Models (SAM) by Park et al. [21]

Figure 6.4.1: Benchmarking results synthetic datasets

predictive features, is related to the number of available samples. This is a result of using a wrapper-based method that has all the limitations of model-driven approaches, e.g. limited data limiting the overall performance. Only when the number of samples exceeds 2500, does the noise start affecting the recall as the results are less limited by the sample size. For the synthetic benchmarks, given an adequate-sized dataset, *Causalteshap* finds most predictive variables without outputting false positives, even in more high-noise situations. In contrast, for the MCM approach both the precision and recall are much lower for all confounding cases, indicating difficulty in finding all predictive variables while preventing false positives. For the SAM method, interestingly the recall drastically increases with more confounding, however, the method is not robust against false positives. In contrast, the MCM approach exhibits substantially lower precision and recall across all confounding scenarios, suggesting difficulties in identifying predictive variables in these various scenarios. Interestingly, the SAM method demonstrates a significant increase in recall with higher levels of confounding, however, it lacks robustness in controlling false positives.

On the case level for *Causalteshap*, there are difficulties with finding predictive features for cases M10 and M11 even with more than 10000 samples finding no predictive features. This is mainly attributed to the strength of the prognostic features that drown out the much smaller predictive component of x_0. For cases M2, M9, and M7, only when the number of samples exceeds 2500, does *Causalteshap* also find the predictive features. On the other hand, for cases M13, M14, M15, M3, and M5, *Causalteshap*

already finds all predictive features with only 500 samples. The relation between the contribution of the predictive features and the prognostic features determines how difficult finding all predictive features will be. If there is a shared prognostic and predictive component, where the predictive component is relatively small, the chance of detecting the specific predictive component is smaller, such as in case M10 or case M11.

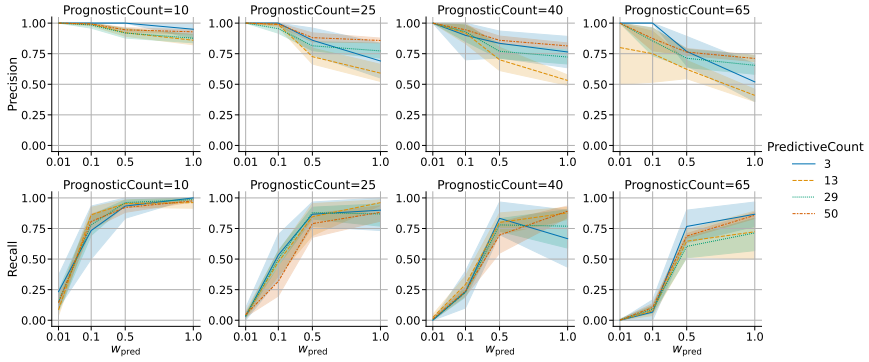
6.4.2 Semi-Synthetic

The results for all semi-synthetic benchmarks are shown in Figure 6.4.2. For all four datasets, the strength of the predictive effect greatly influences whether the predictive features are found or not. This effect is worsened when the number of prognostic features grows, increasing the overall data complexity of the benchmark. In lower dimensionalities, i.e. a low number of prognostic features, the recall reaches levels of 0.8 to 1.0 with a precision above 0.95 showing that *Causalteshap* finds most predictive features without outputting many false positives. However, when the number of prognostic features grows, the performance reduces as the data complexity increases. Even in those situations recalls above 0.8 are still reached while retaining precisions above 0.8, depending on the dataset. Counterintuitively, if the strength of the treatment effect increases, the false positive rate also increases. This is not the case for the synthetic benchmarks and is attributed to an inherent contradiction in the way semi-synthetic datasets are built. This will be further specified in the discussion section.

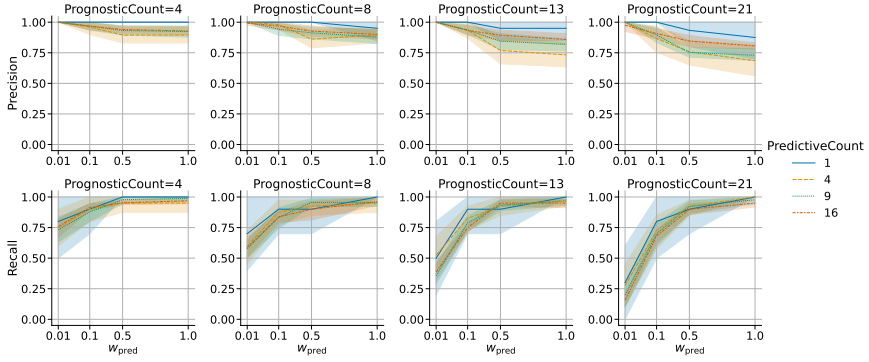
The effect of noise on the News semi-synthetic dataset is shown in Figure 6.4.3. Unlike the synthetic benchmarks, the noise here does have a much larger effect. With increased noise, both recall and precision are reduced. However, even with higher noise levels, *Causalteshap* retains precision levels of around 0.7 while having a lower recall. If the treatment effect is masked by noise, it is intrinsically harder to detect. Because the approach is wrapper-based, *Causalteshap* is not immune to this phenomenon. However, in reduced noise situations *Causalteshap* manages to achieve low false positive rates while finding on average 80% of all predictive features.

6.4.3 Noradrenaline

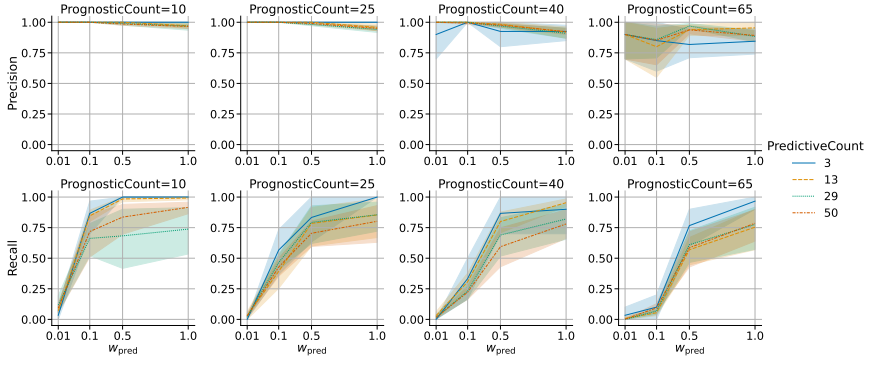
The added value of *Causalteshap* shines through when applied to real-world datasets to facilitate mediator and treatment effect analysis. As mentioned in Section 6.2.2, predictive values are defined as parents of unobserved mediators that influence the treatment, hence these features might provide pointers to the underlying mechanism through which the treatment works, i.e. the mediator(s). Applying *Causalteshap* to the Noradrenaline treatment analysis on AF is shown in Table 6.4.1. The Noradrenaline treatment variable (T) is considered predictive, which means that there is a significant



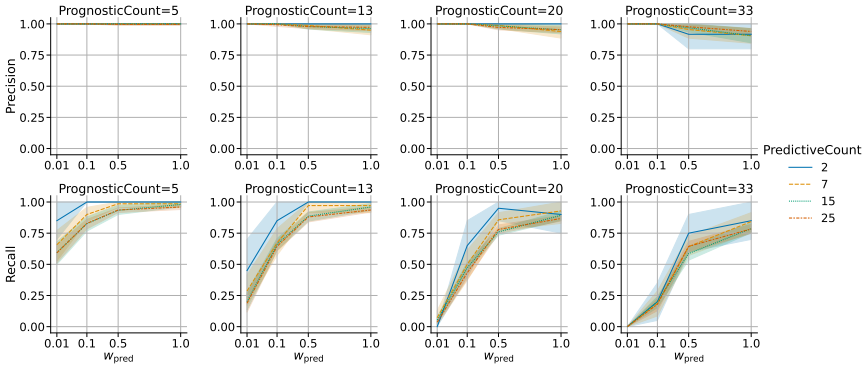
(a) TCGA



(b) TWINS



(c) NEWS



(d) ACIC2016

Figure 6.4.2: Benchmarking results on semi-synthetic datasets. PrognosticCount represents the number of prognostic features, likewise for PredictiveCount. w_pred represents the strength of the predictive features

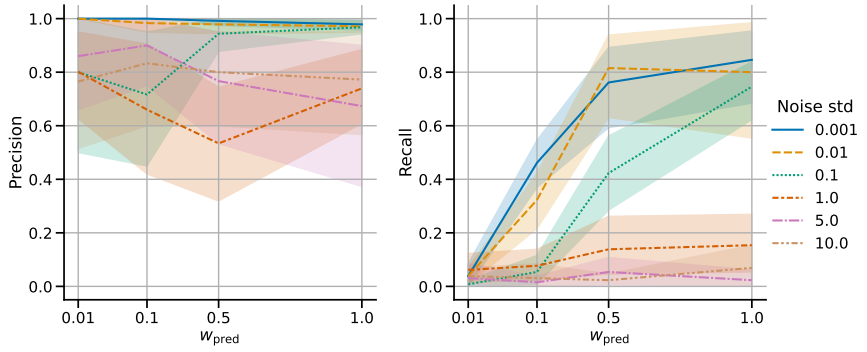


Figure 6.4.3: Varying noise benchmarking results on the News semi-synthetic dataset with N_pred = 25 and N_prog = 13

treatment effect. The $T^1 - T^1$ and $|T^1| - |T^0|$ columns, which are positive, suggest that Noradrenaline increases the risk for AF. The Central Venous Pressure (CVP) feature is considered predictive, determined by the Fligner test and the KS statistic. The $|T^1| - |T^0|$ column displays that the variance of the CVP Shapley values when $T = 0$ is lower than when $T = 1$. In other words, Noradrenaline increases the effect CVP has on the risk of AF. On the contrary, Noradrenaline decreases the effect Fluid balance and Creatinin have on AF. Lactate and pH have different variances for the T^1 and T^0 distribution, however, their predictive Shapley values are not always larger than the noise feature Shapley values. As a result, they are not considered to be predictive, however, they can still provide hypothetical medical insights. Additionally, the Age feature does have a larger predictive difference than the random feature, but the T^1 and T^0 distributions do not differ significantly. This is explained by the

strong prognostic effect of Age on AF, as it is considered a risk factor for AF by the literature [36].

Feature	KS	ttest	Fligner	$T^1 - T^0$	$\lvert T^1\rvert - \lvert T^0\rvert$	Pred
CVP	**0.03**	0.64	**0.00**	-0.01	0.05	**1**
Fluid balance	**0.03**	0.87	**0.00**	0.00	-0.07	**1**
Creatinin (blood)	**0.09**	0.98	**0.00**	0.00	-0.03	**1**
T	**0.00**	**0.00**	**0.00**	0.62	0.43	**1**
Lactate (blood)	2.10	0.44	**0.02**	0.00	0.01	0
pH (blood)	0.21	0.95	**0.02**	0.00	0.02	0
Age	**0.00**	0.69	0.82	0.00	0.00	0
Heart frequency	9.44	0.91	0.76	0.00	0.00	0
O2 concentration (Set)	1.47	0.34	0.72	-0.01	-0.01	0
urgency	13.61	0.99	0.58	0.00	-0.01	0
UrineCAD	0.21	0.90	0.26	0.00	-0.01	0
Furosemide (Lasix)	7.01	0.96	0.64	0.00	0.01	0
ABP mean	6.26	0.88	0.62	0.00	-0.01	0
O2-Saturation (blood)	5.19	0.99	0.04	0.00	0.02	0
Fosfate (blood)	19.87	0.96	0.91	0.00	0.00	0
ABP systolic	2.28	0.91	0.12	0.00	-0.01	0
PEEP (Set)	3.57	0.92	0.52	0.00	-0.01	0
Fluid in	3.66	0.55	0.11	0.00	-0.01	0
Active HCO3 (blood)	8.15	0.50	0.97	0.00	0.00	0
Anion-Gap (blood)	26.11	0.98	0.60	0.00	0.00	0
Enoximon (Perfan)	20.44	0.90	0.85	0.00	0.00	0
Fentanyl	8.15	0.97	0.47	0.00	-0.01	0
Glucose (blood)	10.76	0.68	0.18	0.00	-0.01	0
Hydrocortisone	19.54	0.72	0.78	0.00	0.00	0
Magnesiumsulfate	18.97	0.76	0.96	0.00	0.00	0
Midazolam (Dormicum)	27.13	0.93	0.91	0.00	0.00	0
Propofol (Diprivan)	15.05	0.94	0.38	0.00	0.01	0

Table 6.4.1: Causalteshap results for the Noradrenaline treatment analysis. CVP = Central Venous pressure. ABP = Arterial Blood pressure. Pred = Predictive Flag. KS statistic condition is 0.07 for $\alpha = 0.02$

6.5 Discussion

In this work, we introduced *Causalteshap*, a Shapley-based method for finding predictive features for treatment effect analysis. As presented with the Noradrenaline treatment analysis, *Causalteshap* facilitates testing and finding insights in treatment effect models to further research or understand the treatment. These predictive features

can offer valuable insights into potential (unobserved) mediators as they are the parents of these mediators by definition. The method provides explainability built upon previous research for treatment effect analysis using well-known Shapley values in a plug-and-play open-source library to facilitate causal analysis of treatments.

Leveraging the additive properties of Shapley values provides the opportunity for predictive variable analysis and interpretation. However, Shapley values simply look at encountered associations in the model and therefore do not leverage causal structures. Furthermore, ML models try to optimize these associations to optimize their predictions. Therefore, the outputted Shapley values are devoid of causal direction. There are implications for *Causalteshap* because of this. First, suppose we have $T \rightarrow Y \rightarrow Z$ with Z an effect of Y. If Z is included, it will be an important variable to predict Y because of their high association. Furthermore, T will determine Z indirectly. Therefore, the Shapley value distribution of feature Z for $T = 0$ and $T = 1$ can differ and will be considered predictive. Even stronger, Z can even mask the true predictive features as the model might optimize the strongest associations to Y. Second, consider the case with strong confounding and limited predictive features. Suppose $Y \leftarrow C \rightarrow T$ and no true predictive features. Given the following: if $C < -1$ then $T = 0$, if $C > 1$ then $T = 1$, else T is randomly determined. C is uniformly distributed between -10 and 10. If C is included in *Causalteshap*, the Shapley distributions for feature C when $T = 0$ and $T = 1$ can be considered predictive because of its confounding effect and the absence of causal direction in ML models and Shapley. Do note that this effect is less when considering direct meta-learners, such as the X- and R-learner, which are more robust to this confounding effect and can also be integrated in *Causalteshap* as extensions. These are simple examples, however, they can produce false positives. Therefore, when using *Causalteshap* and interpreting its results it is advised to not blindly apply it to data and think about the causal data-generating process, before and after applying the method. Further, when applying *Causalteshap*, verify the dataset to the original Causal diagram of *Causalteshap*, shown in Figure 6.2.1. If it violates the diagram, *Causalteshap* can still be applied, although with caution.

The synthetic and semi-synthetic datasets do provide a closed environment in which the method can be developed, evaluated, and tested. These datasets try to mimic real-world datasets, while still having access to the ground truth behind these datasets. However, there is a large issue with the DGP of the semi-synthetic datasets for predictive and prognostic features. The predictive and prognostic features are chosen at random to determine the outcome. However, the values of the features themselves are not necessarily independent. Suppose we have two features X and Z, where in its true DGP $X \rightarrow Z$. For the semi-synthetic dataset, X is selected to be solemnly prognostic, while Z is selected to be predictive. X will then be the cause of the predictive feature. Therefore, X could then also be interpreted as a predictive feature, which is a contradiction to its label. Subsequently, giving the label *predictive* to X is intrinsically not wrong, but will be considered wrong in the semi-synthetic

benchmark. This biases the precision, making the true precision higher than it seems. This factor has to be taken into account when working with semi-synthetic datasets as the semi-synthetic benchmark results can be worse than their true results.

Causalteshap is developed on the assumption that all included features for analysis are informative features relevant to either the treatment or the outcome. Incorporating numerous non-informative features, particularly in high-dimensional settings, can increase the complexity of the data analysis, potentially degrade the performance of *Causalteshap*, and consequently elevate the risk of false positives. Therefore, before applying *Causalteshap*, make sure to do proper feature selection beforehand, e.g. using PowerShap [27], or test the relevance of every feature to the hypothesis that you want to evaluate. For feature selection methods in machine learning, we refer to the survey of Dhal et al [37].

The presented method and code library are currently limited to an S-learner for CATE estimation. Future work can expand the method to other meta-learners, such as T-, X-, or R-learners. For the T-learner, the Shapley baseline values can be different for each estimator. Therefore, comparing them using the current algorithm solution will not yield correct results, which requires changing the main algorithm to solve the problem. For direct meta-learners (such as an R-learner or X-learner), the direct CATE estimator in the meta-learner can be used to calculate the predictive Shapley values, however, the same problem as with the T-learner and incompatible Shapley values exists as well. A more detailed analysis can be found in the appendix 6.C. As *Causalteshap* relies on meta-learners, it inherits their limitations. Violations of the unconfoundedness assumption (i.e., the presence of hidden confounders) can bias the estimated effects in unpredictable ways, as the underlying meta-learner would produce unreliable treatment effect estimates. Similarly, when the overlap (positivity) assumption is violated, i.e. meaning there is limited or no overlap in covariate distributions between treated and control groups, the model may generalize poorly. This can lead to exaggerated or underestimated Shapley values for predictive features, depending on the extent of distributional mismatch between $T = 0$ and $T = 1$ groups.

All benchmarks were currently performed using CatBoost and can differ when using different models [38]. Using another feature attribution method besides SHAP, that solves the causal direction problem, will also improve the results. *Causalteshap* can also be extended to multi-output regression or classification, i.e., scenarios with multivariate outcomes, as SHAP inherently supports such settings by providing Shapley values for each outcome with respect to every variable. By aggregating these Shapley values across outcomes for each feature, e.g. through summation, the standard *Causalteshap* framework remains applicable, as the underlying hypotheses hold for the aggregated values. Finally, *Causalteshap* is currently developed for binary treatments or situations that can be reduced to binary treatments. Further work can include finding predictive features for continuous treatments to widen the application potential of the method.

6.6 Conclusion

We aimed to create an open-source plug-and-play sklearn-compatible method, named *Causalteshap*, for finding predictive features in treatment effect estimators by leveraging Shapley values and statistical tests. The benchmark results reveal the high precision of *Causalteshap*, while still achieving high recall rates, making it a tool for better treatment effect analysis, research, and interpretability. With *Causalteshap*, the step to truly discern predictive from prognostic features hereby comes closer.

Declarations

Data and code availability The analysed and used datasets and code, except for the AF dataset are available on GitHub using the following link: https://github.com/predict-idlab/causalteshap. The AF dataset is based on the AmsterdamUMCdb dataset, which requires approval. More information can be found here: https://github.com/AmsterdamUMC/AmsterdamUMCdb.

Appendix

Appendix 6.A Detailed Formulations of Statistical Tests

6.A.1 Welch's t-test

Welch's t-test is calculated using the following formulae, with μ being the mean, X, A, and B arrays, and N the length of the respective array [39]:

$$s_X = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(X_i - \mu(X))^2} \tag{6.8}$$

$$t(A,B) = \frac{\mu(A) - \mu(B)}{\sqrt{(\frac{s_A}{\sqrt{N_A}})^2 + (\frac{s_B}{\sqrt{N_B}})^2}} \tag{6.9}$$

The degrees of freedom ν for the test is then calculated using the Welch-Satterthwaite equation as:

$$\nu(A,B) = \frac{s_A^4 + s_B^4}{\nu_A^{-1}s_A^4 + \nu_B^{-1}s_B^4} \tag{6.10}$$

The equality only holds if the sample sizes of the two distributions are the same, i.e. $N_A = N_B$, which is always the case in this work.

With $\nu_X = N_X - 1$ the degrees of freedom for X. Then, given ν, we can define the Cumulative Distribution Function (CDF) of the t-distribution $T(t)$ and estimate the two-tailed p-value using:

$$WelchsTTest(A,B) = 2(1 - T(|t(A,B)|, \nu(A,B))) \tag{6.11}$$

6.A.2 The Fligner test

The test is calculated as follows given two arrays, A and B, each with a length N_A and N_B, medians η_A and η_B, and $N_{tot} = N_A + N_B$ the total length of A and B together.

First, define $V = |A - \eta_A|$ and $W = |B - \eta_B|$. Then, calculate the rank of each element of the combined array $[V, W]$ as $R_V = rank(V, [V, W])$ and $R_W =$

$rank(W, [V, W])$. Given these ranks, the standard deviations away from the mean of the total array $[V, W]$ are calculated for every element using the percentile point function (i.e. the inverse of the CDF) of the Normal distribution $NormPercentile$ as follows:

$$S_A = NormPercentile\left(\frac{R_V}{2(N_{tot}+1)} + 0.5\right) \tag{6.12}$$

$$S_B = NormPercentile\left(\frac{R_B}{2(N_{tot}+1)} + 0.5\right) \tag{6.13}$$

With S being S_A and S_B appended. This now allows us to calculate the p-value of the Fligner test which is calculated using the Chi-squared CDF function $Chi(\chi)$ with estimated chi-square statistic over S (i.e. Fligner statistic) χ_{est}:

$$\sigma(S) = \frac{\sum_i (S_i - mean(S))^2}{len(S) - 1} \tag{6.14}$$

$$\chi_{est} = \frac{N_A(\sum_i (S_{A,i}) - mean(S))^2 + N_B(\sum_i (S_{B,i}) - mean(S))^2}{\sigma(S)} \tag{6.15}$$

$$Fligner(A, B) = 1 - Chi(\chi_{est}, 1) \tag{6.16}$$

This results in the Fligner-Killeen p-value of A and B.

6.A.3 Kolmogorov-Smirnov Test

To calculate the KS-test we first need to calculate the two CDFs F_i and F_R:

$$F_i = CDF(|S_{pred}(X_i[1..N])|) \tag{6.17}$$

$$F_R = CDF(|S_{pred}(X_r[1..N])|) \tag{6.18}$$

Given these CDFs, the null hypothesis for the KS statistic is defined as:

$$\forall x : F_i(x) <= F_R(x) \tag{6.19}$$

The KS statistic for this case is then calculated as

$$D^+ = \sqrt{n} \cdot sup_x(F_i(x) - F_R(x)) \tag{6.20}$$

To verify whether the KS statistic rejects the null hypothesis or not we need to define a threshold. Suppose α represents the probability of the value being lower than the random variable K (Kolmogorov distribution). Then, given the approximate CDF of this function (which holds for $N > 30$) [40] we get:

$$\alpha = F_K(x) = Pr[K \leq x] = 1 - e^{-2x^2} \tag{6.21}$$

We can calculate the quantile function by inversing $F_K(x)$:

$$Q_K(\alpha) = \sqrt{-0.5 * ln(1 - \alpha)} \tag{6.22}$$

Then given a predefined α, we can reject the hypothesis if $D^+ > Q_K(\alpha)$.

Appendix 6.B Multiple Testing Corrections and Bounds for Causalteshap

Causalteshap performs multiple statistical testing for every single feature. Therefore, we need to discuss the multiple testing problems for a single feature. For Part 1, we have a hypothesis that is the union of the Fligner test T_f and of Welch's t-test T_w. For clarity, we will call the first part, part A or test T_A. Part 2 consists of a single Kolmogorov-Smirnov test, T_{KS}. We will denote the second part as part B or test $T_B = T_{KS}$. Part A and Part B must both be true for *Causalteshap* to flag a feature as predictive. We will denote this composite test as T_p, which always applies to a single feature.

$$T_A = T_f \vee T_w \tag{6.23}$$

$$T_p = T_A \wedge T_B = (T_f \vee T_w) \wedge T_B \tag{6.24}$$

We will assume a single α for T_f, T_w, and T_{KS}. So T_A is positive if either T_f or T_w is positive. Supposing all tests are independent we then have the following α_A for part A:

$$\alpha_A = 1 - (1 - \alpha)^2 = 2\alpha - \alpha^2 \tag{6.25}$$

For part B we have $\alpha_B = \alpha$ because we are only performing a single test. Given independence between part A and part B, the total α_p is:

$$\alpha_p = \alpha_A \alpha_B = 2\alpha^2 - \alpha^3 \tag{6.26}$$

However, independence between all these tests is hard to guarantee. Therefore, given the setup, a positive dependency is highly likely. Thus, we also will assume positive dependency to assess the robustness of the composite test. Given positive dependency between T_f and T_w the lower and upper bound of T_A becomes:

$$\alpha \leq \alpha_A \leq 2\alpha - \alpha^2 \tag{6.27}$$

Likewise, if there is positive dependency between T_A and T_B the upper and lower bound of T_p becomes:

$$\alpha_A \alpha_B \leq \alpha_p \leq min(\alpha_A, \alpha_B) \tag{6.28}$$

Combining these bounds, we get the following lower and upper bounds on α_p:

$$\alpha^2 \leq \alpha_p \leq \alpha \tag{6.29}$$

Therefore, given even positive dependency between any of the tests controlling for false positives or multiple testing for T_p is not necessary as the actual false positive rate will be most likely lower than the specified α. Therefore, by setting the significance threshold for all base tests (T_f, T_w, T_{KS}) to α guarantees at least a significance threshold for T_p of α as well.

As the hypotheses of *Causalteshap* are formulated for a single feature we are not making any conclusions over a set of features. Therefore, there is no need to control for multiple comparisons across these features. However, if there are conclusions that are being made on the model, such as a set of features being predictive, then of course a multiple comparison adjustment must incorporated.

Appendix 6.C Extension of Causalteshap to Other Meta-Learners

6.C.1 T-learner

A T- or Twin-learner is a meta-learner that fits two estimators: One for $T = 1$ and one for $T = 0$:

$$\hat{\mu}_0(X) = \hat{E}[Y|X, T = 0] \tag{6.30}$$

$$\hat{\mu}_1(X) = \hat{E}[Y|X, T = 1] \tag{6.31}$$

The extension to *Causalteshap* is then trivial with S_{μ_0} simply the Shapley values of $\hat{\mu}_0(X)$ and vice versa. However, the main issue here lies with the baseline values of Shapley that are different which makes comparing these Shapley values with each other more complex and more prone to false positives.

6.C.2 X-learner

An X-learner is considered a direct or A-learner and consists of three estimators [41]: the treatment outcome estimator μ_1, the control outcome estimator μ_0, and the CATE model τ. The first estimators are trained as follows, analogous to a T-learner:

$$\hat{\mu}_0(X) = \hat{E}[Y|X, T = 0] \tag{6.32}$$

$$\hat{\mu}_1(X) = \hat{E}[Y|X, T = 1] \tag{6.33}$$

We then define the pseudo-outcomes CATE estimators τ_0 and τ_1:

$$\hat{\tau}_0(X) = \hat{E}[\hat{\mu}_1(X) - Y|X, T = 0] \tag{6.34}$$

$$\hat{\tau}_1(X) := \hat{E}[Y - \hat{\mu}_0(X)|X, T = 1] \tag{6.35}$$

We then define a new function $g(x)$ that weights τ_0 and τ_1 according to a function, often also a propensity estimator. This results in the final CATE estimator:

$$\hat{\tau}(X) = g(x)\hat{\tau}_0(X) + (1 - g(x))\hat{\tau}_1(x) \tag{6.36}$$

The first testing procedure of *Causalteshap* can be applied to the T-learner part $\hat{\mu}_0(X)$ and $\hat{\mu}_1(X)$, but then we have the same issue of having different baseline values. The

second part of *Causalteshap* is easy to apply to an X-learner, simply calculate the Shapley values of $\hat{\tau}(X)$ with an added random variable and perform the KS-test. An alternative would be to perform the KS-test on both $\hat{\tau}_0(X)$ and $\hat{\tau}_1(X)$ and do a Bonferroni correction and take the features that are predictive in both.

6.C.3 R-learner

An R-learner is considered a direct or A-learner and consists of three base learners: a nuisance estimator $\hat{\mu}$, propensity estimator $\hat{\pi}$, and pseudo-outcome estimator $\hat{\tau}$. Given the features X, treatment T, and outcome Y, the learners are trained as follows [42]:

$$\hat{\mu} = \hat{E}[Y|X] \tag{6.37}$$

$$\hat{\pi} = \hat{E}[T|X] \tag{6.38}$$

If you are training a gradient-boosting pseudo-outcome estimator you fit the following function:

$$\hat{\tau} = \hat{E}\left[\frac{Y - \hat{\mu}(X)}{T - \hat{\pi}(X)}|X\right] \tag{6.39}$$

You do add weighted learning where every sample is weighted with weights $w = (T - \hat{\pi}(X))^2$. On the contrary, if you train a linear model such as lasso you regress $Y - \hat{\mu}\, T - \pi(\hat{X})$.

Given these models you can define the predicted potential outcome under control $\hat{\mu}_0$ and outcome under treatment $\hat{\mu}_1$ as follows:

$$\hat{\mu}_0(X) = \hat{\mu}(X) - \hat{\pi}(X)\hat{\tau}(X) \tag{6.40}$$

$$\hat{\mu}_1(X) = \hat{\mu}(X) + (1 - \hat{\pi}(X))\hat{\tau}(X) \tag{6.41}$$

Consequently, we can define the Shapley values for each model and the potential outcomes. $S_\mu(X)$ are the shapley values of model $\hat{\mu}$ on features X, and likewise for all other models.

$$S_{\mu_0}(X) = S_\mu(X) - S_\pi(X)S_\tau(X) \tag{6.42}$$

$$S_{\mu_1}(X) = S_\mu(X) + (1 - S_\pi(X))S_\tau(X) \tag{6.43}$$

In theory, all variables included in the pseudo-outcome model $\hat{\tau}$ should be predictive as only these will explain the treatment outcome. However, in practice, it could be that other variables are included in $\hat{\tau}$ as this model trains on the whole covariate set, especially if trained on a high-dimensional covariate set or with a complex treatment effect. If the treatment effect is simple, an easily interpretable model such as LASSO regression will suffice, however, if the treatment effect is more complex a stronger non-linear model might be required where the predictive variables might not be easily interpreted. To directly apply *Causalteshap* to an R-learner, the first part (Fligner and Welch's test) will be applied to $S_{\mu_0}(X)$ for $T = 0$ and $S_{\mu_1}(X)$ for T=1. Likewise,

the second part (KS-test) can be applied on S_τ which must be fitted on $[X, X_r]$ with X_r a random feature.

However, $S_{\mu_1}(X) - S_{\mu_0}(X) = S_\tau(X)$. Therefore, the Shapley values of $\hat{\tau}$ are the only difference between $T = 0$ and $T = 1$. Compared to an S-learner, where you compare the model to a slightly different version of itself, this R-learner will be much less robust to random noise in Shapley values. If we have considerable Shapley attribution to noise in $\hat{\tau}$, the Fligner test or Welch's t-test can generate more false positives due to this noise attribution. A perfectly viable alternative would be to perform feature selection on the $\hat{\tau}$ estimator.

References

[1] Judea Pearl and Dana Mackenzie. *The Book of Why*. Basic Books, New York, 2018.

[2] Andrew Forney and Scott Mueller. *Causal inference in AI education: A primer*. Journal of Causal Inference, 10(1):141–173, January 2022. Publisher: De Gruyter.

[3] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. *Causal machine learning for predicting treatment outcomes*. Nature Medicine, 30(4):958–968, April 2024.

[4] Yaobin Ling, Pulakesh Upadhyaya, Luyao Chen, Xiaoqian Jiang, and Yejin Kim. *Emulate randomized clinical trials using heterogeneous treatment effect estimation for personalized treatments: Methodology review and benchmark*. Journal of Biomedical Informatics, 137:104256, January 2023.

[5] Jonathan Crabbé, Alicia Curth, Ioana Bica, and Mihaela van der Schaar. *Benchmarking Heterogeneous Treatment Effect Models through the Lens of Interpretability*, June 2022. arXiv:2206.08363 [cs, stat].

[6] Donald B Rubin. *Causal Inference Using Potential Outcomes*. Journal of the American Statistical Association, 100(469):322–331, March 2005. Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/016214504000001880.

[7] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. *Metalearners for estimating heterogeneous treatment effects using machine learning*. Proceedings of the National Academy of Sciences, 116(10):4156–4165, March 2019. Publisher: Proceedings of the National Academy of Sciences.

[8] Erik Hermansson and David Svensson. *On Discovering Treatment-Effect Modifiers Using Virtual Twins and Causal Forest ML in the Presence of Prognostic Biomarkers*. In Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić,

David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre, editors, Computational Science and Its Applications – ICCSA 2021, pages 624–640, Cham, 2021. Springer International Publishing.

[9] Scott M Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions*. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 4765–4774. Curran Associates, Inc., 2017.

[10] Pantelis Linardatos, Vasilis Papastefanopoulos, and Sotiris Kotsiantis. *Explainable AI: A Review of Machine Learning Interpretability Methods*. Entropy, 23(1):18, December 2020.

[11] Zhongheng Zhang, Heidi Seibold, Mario V. Vettore, Woo-Jung Song, and Vieille François. *Subgroup identification in clinical trials: an overview of available methods and their implementations with R.* Annals of Translational Medicine, 6(7), 2018.

[12] Yang Liu, Xiwen Ma, Donghui Zhang, Lijiang Geng, Xiaojing Wang, Wei Zheng, and Ming-Hui Chen. *Look before you leap: systematic evaluation of tree-based statistical methods in subgroup identification.* Journal of Biopharmaceutical Statistics, 29(6):1082–1102, 2019.

[13] Demissie Alemayehu, Yang Chen, and Marianthi Markatou. *A comparative study of subgroup identification methods for differential treatment effect: Performance metrics and recommendations.* Statistical Methods in Medical Research, 27(12):3658–3678, December 2018.

[14] Marco Bonetti and Richard D. Gelber. *Patterns of Treatment Effects in Subsets of Patients in Clinical Trials*. Biostatistics (Oxford, England), 5(3):465–481, July 2004.

[15] Ilya Lipkovich, Alex Dmitrienko, Jonathan Denne, and Gregory Enas. *Subgroup Identification Based on Differential Effect Search—A Recursive Partitioning Method for Establishing Response to Treatment in Patient Subpopulations.* Statistics in Medicine, 30(21):2601–2621, 2011.

[16] Jared C. Foster, Jeremy M.G. Taylor, and Stephen J. Ruberg. *Subgroup Identification from Randomized Clinical Trial Data.* Statistics in medicine, 30(24):10.1002/sim.4322, October 2011.

[17] T. Cai, L. Tian, P. H. Wong, and L. J. Wei. *Analysis of Randomized Comparative Clinical Trial Data for Personalized Treatment Selections*. Biostatistics, 12(2):270–282, April 2011.

[18] Lisa L. Doove, Katrijn Van Deun, Elise Dusseldorp, and Iven Van Mechelen. *QUINT: A Tool to Detect Qualitative Treatment–Subgroup Interactions in Randomized Controlled Trials*. Psychotherapy Research, 26(5):612–622, September 2016.

[19] Xinzhou Guo, Waverly Wei, Molei Liu, Tianxi Cai, Chong Wu, and Jingshen Wang. *Assessing the Most Vulnerable Subgroup to Type II Diabetes Associated with Statin Usage: Evidence from Electronic Health Record Data.* Journal of the American Statistical Association, 118(543):1488–1499, July 2023.

[20] Lu Tian, Ash A Alizadeh, Andrew J Gentles, and Robert Tibshirani. *A Simple Method for Estimating Interactions between a Treatment and a Large Number of Covariates*. Journal of the American Statistical Association, 109(508):1517–1532, October 2014.

[21] Hyung Park, Eva Petkova, Thaddeus Tarpey, and R Todd Ogden. *A Sparse Additive Model for Treatment Effect-Modifier Selection.* Biostatistics, 23(2):412–429, April 2022.

[22] X. De Luna, I. Waernbaum, and T. S. Richardson. *Covariate Selection for the Nonparametric Estimation of an Average Treatment Effect.* Biometrika, 98(4):861–875, December 2011.

[23] Debo Cheng, Jiuyong Li, Lin Liu, Jiji Zhang, Jixue Liu, and Thuc Duy Le. *Local Search for Efficient Causal Effect Estimation.* IEEE Transactions on Knowledge and Data Engineering, 35(9):8823–8837, 2023.

[24] Thomas Lumley, Paula Diehr, Scott Emerson, and Lu Chen. *The Importance of the Normality Assumption in Large Public Health Data Sets*. Annual Review of Public Health, 23(1):151–169, May 2002.

[25] Michael A. Fligner and Timothy J. Killeen. *Distribution-Free Two-Sample Tests for Scale*. Journal of the American Statistical Association, 71(353):210–213, March 1976.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, and et al. *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 12:2825–2830, 2011.

[27] Jarne Verhaeghe, Jeroen Van Der Donckt, Femke Ongenae, and Sofie Van Hoecke. *Powershap: A Power-Full Shapley Feature Selection Method.* In Massih-Reza Amini, Stéphane Canu, Asja Fischer, Tias Guns, Petra Kralj Novak, and Grigorios Tsoumakas, editors, Machine Learning and Knowledge Discovery in Databases, pages 71–87. Springer International Publishing, March 2023.

[28] Jarne Verhaeghe, Thomas De Corte, Christopher M. Sauer, Tom Hendriks, Olivier W. M. Thijssens, Femke Ongenae, Paul Elbers, Jan De Waele, and Sofie Van Hoecke. *Generalizable calibrated machine learning models for real-time atrial fibrillation risk prediction in ICU patients.* International Journal of Medical Informatics, 175:105086, July 2023.

[29] John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M. Stuart. *The Cancer Genome Atlas Pan-Cancer analysis project*. Nature Genetics, 45(10):1113–1120, October 2013. Number: 10 Publisher: Nature Publishing Group.

[30] Douglas Almond, Kenneth Y Chay, and David S Lee. *THE COSTS OF LOW BIRTH WEIGHT*. QUARTERLY JOURNAL OF ECONOMICS, 2005.

[31] David Newman. *Bag of Words*, 2008.

[32] Vincent Dorie, Jennifer Hill, Uri Shalit, Marc Scott, and Dan Cervone. *Automated versus Do-It-Yourself Methods for Causal Inference: Lessons Learned from a Data Analysis Competition*. Statistical Science, 34(1), February 2019.

[33] Patrick J. Thoral, Jan M. Peppink, Ronald H. Driessen, Eric J. G. Sijbrands, Erwin J. O. Kompanje, Lewis Kaplan, Heatherlee Bailey, Jozef Kesecioglu, Maurizio Cecconi, Matthew Churpek, Gilles Clermont, Mihaela van der Schaar, Ari Ercole, Armand R. J. Girbes, and Paul W. G. Elbers. *Sharing ICU Patient Data Responsibly Under the Society of Critical Care Medicine/European Society of Intensive Care Medicine Joint Data Science Collaboration: The Amsterdam University Medical Centers Database (AmsterdamUMCdb) Example**. Critical Care Medicine, 49(6):e563, June 2021. ZSCC: NoCitationData[s0].

[34] Takuo Yoshida, Tomoko Fujii, Shigehiko Uchino, and Masanori Takinami. *Epidemiology, prevention, and treatment of new-onset atrial fibrillation in critically ill: a systematic review*. Journal of Intensive Care, 3(1), April 2015. ZSCC: 0000062.

[35] Travis J. Moss, James Forrest Calland, Kyle B. Enfield, Diana C. Gomez-Manjarres, Caroline Ruminski, John P. DiMarco, Douglas E. Lake, and J. Randall Moorman. *New-Onset Atrial Fibrillation in the Critically Ill*. Critical Care Medicine, 45(5):790–797, May 2017. ZSCC: 0000096.

[36] Kristina Wasmer, Lars Eckardt, and Günter Breithardt. *Predisposing factors for atrial fibrillation in the elderly*. Journal of Geriatric Cardiology : JGC, 14(3):179–184, March 2017.

[37] Pradip Dhal and Chandrashekhar Azad. *A comprehensive survey on feature selection in the various fields of machine learning*. Applied Intelligence, 52(4):4543–4581, July 2021.

[38] Liudmila Prokhorenkova, Gleb Gusev, and et al. *CatBoost: unbiased boosting with categorical features*. arXiv:1706.09516 [cs], January 2019.

[39] B. L. Welch. *THE GENERALIZATION OF 'STUDENT'S' PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARLANCES ARE INVOLVED*. Biometrika, 34(1-2):28–35, January 1947.

[40] Donald Ervin Knuth. *The art of computer programming*, chapter 3.3.1, page 52. Addison-Wesley, Reading, Mass, 3rd ed edition, 1997.

[41] Sören R. Künzel, Jasjeet S. Sekhon, Peter J. Bickel, and Bin Yu. *Metalearners for Estimating Heterogeneous Treatment Effects Using Machine Learning*. Proceedings of the National Academy of Sciences, 116(10):4156–4165, March 2019.

[42] X Nie and S Wager. *Quasi-oracle estimation of heterogeneous treatment effects*. Biometrika, 108(2):299–319, 09 2020.

7

Conformal Prediction for Dose-Response Models with Continuous Treatments

To provide effective treatment advice using a treatment dosing model for continuous treatments like antimicrobials, the model should accurately predict outcomes for any given dose. Additionally, it should clearly indicate when a prediction is not supported, ensuring trustworthiness. We aim to tackle this using uncertainty quantification. Therefore, this chapter explores the use of conformal prediction to provide uncertainty quantification for dose-response models to address RG4. The proposed method, propensity-weighted conformal prediction, generates infinite intervals in regions with limited or no support and compensates for the lack of overlap in other regions. Additionally, it provides coverage guarantees for any dose in the dose-response model. The method is rigorously evaluated on a synthetic benchmark specifically designed to test overlap issues and on a semi-synthetic benchmark generated using an anesthesiology simulator for propofol, AMICAS. The results demonstrate that incorporating propensity in weighted conformal prediction effectively addresses overlap issues, providing a reliable method for trustworthy dose-response models. My contributions to this chapter are the development and conception of the method, the writing, and the design and validation of the experiments.

Conformal Prediction for Dose-Response Models with Continuous Treatments

Jarne Verhaeghe, Jef Jonkers, Sofie Van Hoecke

Submitted to International Journal of Approximate Reasoning

Abstract Understanding the dose-response relation between a continuous treatment and the outcome for an individual can greatly drive decision-making, particularly in areas like personalized drug dosing and personalized healthcare interventions. Point estimates are often insufficient in these high-risk environments, highlighting the need for uncertainty quantification to support informed decisions. Conformal prediction, a distribution-free and model-agnostic method for uncertainty quantification, has seen limited application in continuous treatments or dose-response models. To address this gap, we propose a novel methodology that frames the causal dose-response problem as a covariate shift, leveraging weighted conformal prediction. By incorporating propensity estimation, conformal predictive systems, and likelihood ratios, we present a practical solution for generating prediction intervals for dose-response models. Additionally, our method approximates local coverage for every treatment value by applying kernel functions as weights in weighted conformal prediction. Finally, we use a new synthetic and semi-synthetic benchmark dataset to demonstrate the significance of covariate shift assumptions in achieving robust prediction intervals for counterfactual dose-response models.

7.1 Introduction

How can we determine the optimal dose for a patient to ensure the best therapeutic outcome? What is the impact of discounts in an online store on sales? What impact does CO_2 concentration have on local climates? At the core of each of these questions lies a shared causal idea: understanding the dose-response relation under continuous treatments to inform decision-making. In many cases, these decisions bear significant consequences, where relying solely on point estimates may be insufficient [1]. Particularly in high-stakes situations, augmenting predictions with uncertainty quantification (UQ) can significantly improve decision-making processes [1]. For instance, while the estimated causal effect of a continuous treatment may appear positive, prediction intervals could suggest a largely negative outcome for a specific individual. Such insights are crucial for deciding interventions. To tackle this, conformal prediction (CP) offers a robust solution for UQ, being both distribution-free and model-agnostic, with formal coverage guarantees [2].

In this work, we seek to extend CP to UQ in dose-response models, aiming to aid decision-makers with more informed estimates to tackle such questions. We intro-

duce a novel approach for deriving prediction intervals in the continuous treatment setting using weighted conformal prediction by combining propensity estimation with weighted conformal predictive systems. Furthermore, with the aid of a novel synthetic and semi-synthetic benchmark, we show how viewing the problem as a covariate shift approach provides coverage across all treatment values to help create more individualized dose-response curves.

7.2 Background

In this paper we expand upon the potential outcomes framework introduced in Rubin et al. [3], otherwise known as the Rubin framework to accommodate continuous treatments. Consider a continuous treatment variable $T \in [t_L, t_U]$ with a lower bound t_L and upper bound t_U, observed covariates X, and potential outcomes $Y(t) \in \mathbb{R}$ representing the outcome that would be observed under treatment level t. The Conditional Average Dose-Response Function (CADRF) is defined as $\nu(x,t) = E[Y(t)|X = x]$, the expected value over the Individual Dose-Response Functions (IDRF) for all individuals with observed X. Similar to Conditional Average Treatment Effects (CATE), to estimate the CADRF we make the following standard assumptions [3, 4]:

- Unconfoundedness: $Y(t) \perp\!\!\!\perp T|X, \forall t \in T$. This assumption states that, conditional on the observed covariates, the treatment assignment is independent of the potential outcomes. In other words, there are no unobserved confounders that influence both the treatment assignment and the outcome.
- Overlap or positivity: $0 < P(T = t|X = x) < 1, \forall t \in T$ with $x \in X$. The overlap assumption ensures that for every covariate value x, there is a positive probability of receiving any treatment level. This is crucial for estimating treatment effects across the entire range of treatment levels.
- Consistency: $Y = Y(t)$ with probability 1. This assumption links the observed outcomes to the potential outcomes, stating that the observed outcome is equal to the potential outcome corresponding to the treatment received.

Quantifying the IDRF requires observing the $Y(t)$ for all possible treatment values. These treatment values are all counterfactuals and thus impossible to observe as we only can observe Y for a single treatment value t at a time. Furthermore for estimating the CADRF, likewise with CATE estimation, the distribution of the treatment assignment can bias the estimation [4]. This distribution of the treatment assignment is called the propensity distribution, which was initially defined for binary treatments. Hirano and Imbens [4] introduced the generalized propensity score (GPS) for continuous treatments that aims to unbias the CATE estimation for continuous treatments. The GPS is defined as $\pi(t_i|x) = f_{T|X}(T = t_i|X = x)$, which is the evaluation of $T = t_i$ on the conditional probability density function $T|X$ [4]. If the treatment

is independent of X, i.e. there are no confounders that influence treatment assignment, then $f_{T|X}$ is equal for all possible X. Furthermore, the treatment assignment is considered uniformly assigned between lower t_L and upper t_U possible treatment if $f_{T|X}$ represents the density function of the uniform distribution between t_L and t_U. The GPS can then be used to mimic the randomly assigned treatment to estimate the unbiased CADRF [5].

The simplest method to estimate the CADRF is using an S-learner where a single learner is fit on both the covariates X and the treatment T to estimate Y. This approach provides a CADRF for each specific sample by keeping the covariates X constant and changing T to all different treatment values. However, if the treatment in the data is not uniformly assigned then the epistemic error can increase for specific treatment values t_i and $X = x$ in low overlap regions or where $\pi(t_i|x)$ becomes very small. Consequently inferring $T = t_i$ in these regions would yield unreliable model estimates which should be communicated to ensure correct usage of a CADRF model.

The estimated $\widehat{IDRF}$ can also be seen as follows: $\widehat{IDRF} = \nu(x,t) + \epsilon_{a,IDRF}(x,t) + \epsilon_{e,IDRF}(x,t)$. The aleatoric uncertainty is symbolized by $\epsilon_{a,IDRF}(x,t)$ created by the inherent variability between individuals having the same covariates. $\epsilon_{e,IDRF}(x,t)$ symbolises the epistemic uncertainty coming from model specification and finite samples. Estimating both uncertainties creates the opportunity to estimate the ranges of the $\widehat{IDRF}$:

Problem Definition: To accurately estimate the $\widehat{IDRF}$ for all possible treatment values we require correctly estimating both uncertainties for all treatment values equally, or more formally; for a specific significance level α, lower treatment bound t_L, upper treatment bound t_U, and covariates X, we require prediction intervals $C(t, X)$ such that

$$\mathbb{P}(Y(t) \in C(X,t)) \geq 1 - \alpha, \quad \forall t \in [t_L, t_U] \tag{7.1}$$

This requirement necessitates prediction intervals that guarantee coverage for each possible treatment value individually.

7.3 Related Work

Our proposed solution combines three different domains: propensity score methods, conformal prediction, and treatment effect or dose-response modelling.

Propensity score methods, introduced by Rosenbaum and Rubin [6], have become widespread in causal inference, especially in observational studies. These methods aim to balance confounders across treatment groups, reducing bias in treatment effect estimates. Hirano and Imbens [4] generalized this propensity score to continuous instead of binary treatments, introducing the generalized propensity score and building the foundation for causal inference with continuous exposures. Wu et al. [5] used the gen-

eralized propensity score for matching continuous treatments to debias the treatment assignment and more accurately estimate the average dose-response curve for all treatment values. Other approaches adapt machine learning techniques to dose-response modelling. For instance, Athey et al. [7] developed generalized random forests for heterogeneous treatment effect estimation, adaptable to continuous treatments.

To provide UQ, this work adapts conformal prediction. Conformal prediction is a model-agnostic method introduced by Vovk et al. [2] that constructs prediction intervals with guaranteed finite-sample coverage under distribution-free assumptions. Conformal prediction uses conformity scores to assess uncertainty. Various improvements, such as the adaptive version by Romano et al. [8], have increased the flexibility and applicability to even heteroscedastic settings. Additionally, Lei et al. [9] and Papadopoulos et al. [10] introduced split conformal prediction, significantly improving computational efficiency. For scenarios involving covariate or distribution shifts, Tibshirani et al. [11] introduced weighted conformal prediction to ensure coverage under mismatched training and testing data distributions, with additional work by Gibbs and Candès [12, 13] and Barber et al. [14]. By reweighting the calibration samples similar to weighted conformal prediction, Guan [15] introduced localized conformal prediction where the prediction intervals are determined by calibration samples localized around the test sample. Vovk et al. [16] also introduced conformal predictive systems (CPS); an extension of full conformal prediction that allows extracting predictive distributions instead of prediction intervals. More recently, Jonkers et al. [17] combined previous concepts, introducing weighted conformal predictive systems to also account for covariate shifts.

In causal inference, conformal prediction has mainly been applied to binary treatments. For instance, Lei and Candès [18] were among the first to apply conformal prediction to treatment effects estimation in randomized experiments and confounded or observational data. Jonkers et al. [19] and Alaa and Ahmad [20] extended this approach to the potential outcomes framework, providing uncertainty to quantify individual treatment effects. However, the use of conformal prediction in continuous treatment settings remains largely unexplored. Schröder et al. [21] proposed a conformal prediction framework for prediction intervals of treatment effects for continuous treatment interventions. However, their approach mainly covers single-treatment interventions and is computationally intensive, requiring optimization per confidence level, treatment, and sample where they provide prediction intervals for a single treatment value. For a more in-depth analysis of Schröder et al. [21], see Appendix 7.E.

Our goal is to achieve predictive coverage across the entire range of the treatment variable in estimating the dose-response curve. To our knowledge, no existing UQ methods offer conformal prediction guarantees for dose-response models with continuous treatments. To address this gap, we propose a novel methodology that seeks to provide this coverage by integrating weighted conformal prediction with propensity score weighting thereby guaranteeing coverage for any treatment value in continuous

treatment dose-response models.

7.4 Method

7.4.1 Introduction to Conformal Prediction

Before delving into our proposed method, we provide a formal introduction to conformal prediction [11, 17]. Conformal prediction offers a powerful method for constructing prediction intervals with guaranteed finite-sample coverage under distribution-free assumptions [2]. The key insight of conformal prediction lies in its use of a nonconformity measure to quantify the degree to which a new observation differs from previously observed data.

Let us consider a regression problem with the training data being n independent and identically distributed (i.i.d.) data pairs $Z_1 = (X_1, y_1), ..., Z_n = (X_n, y_n)$, where $X_i \in \mathbb{R}^d$ represents a vector of d features and $y_i \in \mathbb{R}$ the corresponding label. Consider $Z_{n+1} = (X_{n+1}, y_{n+1})$ a new exchangeable point being the test observation to evaluate and provide prediction intervals. Conformal prediction aims to construct a prediction interval $\hat{C}(X_{n+1})$ such that

$$\mathbb{P}\{y_{n+1} \in \hat{C}(X_{n+1})\} \geq 1 - \alpha \tag{7.2}$$

for a pre-specified significance level $\alpha \in (0, 1)$ where the probability is calculated over the points $Z_i, i = 1, ..., n$.

To achieve this, we first define a nonconformity measure $S((X, y), Z_{1:n})$ that quantifies how different the pair (X, y) is from a multiset $Z_{1:n} = \{Z_1, ..., Z_n\}$ of data points. The lower the nonconformity measure, the more the pair conforms to the multiset $Z_{1:n}$. The most commonly used nonconformity measure is the absolute error $S((X, y), Z_{1:n}) = |y - \hat{\mu}(X)|$ with $\hat{\mu}$ an estimator fitted on $Z_{1:n}$.

Next, for each possible value $y \in \mathbb{R}$ that y_{n+1} could be, we compute the nonconformity scores:

$$\begin{aligned} R_i^y := S((X_i, y_i), \{(X_1, y_1), ..., (X_{i-1}, y_{i-1}), \\ (X_{i+1}, y_{i+1}), ..., (X_n, y_n), (X_{n+1}, y)\}), i = 1, ..., n \end{aligned} \tag{7.3}$$

$$R_{n+1}^y := S((X_{n+1}, y), \{(X_1, y_1), ..., (X_n, y_n)\}) \tag{7.4}$$

Finally, we construct the prediction interval containing all y where [17]

$$\hat{C}(X_{n+1}) = \left\{ y \in \mathbb{R} : \frac{\#\{i = 1, ..., n+1 : R_i^y \geq R_{n+1}^y\}}{n+1} \geq 1 - \alpha \right\} \tag{7.5}$$

Tibshirani et al. [11] presented conformal prediction slightly differently by using quantile functions instead, which will be more convenient for weighted conformal prediction later on. Tibshirani et al. [11] defines the $1-\alpha$ quantile function as follows,

where $F_R(y)$ represents the distribution of nonconformity scores R_i^y consisting of a sum of point masses δ_a with mass at a where $R^y \sim F_R(y)$ [11]. $F_R(y)$ can then be used to calculate probabilities:

$$Quantile(1-\alpha; F_R(y)) = inf\{R_i^y : \mathbb{P}\{R^y \leq R_i^y\} \geq 1-\alpha\} \tag{7.6}$$

$$F_R(y) = \frac{1}{n+1}\sum_{i=1}^{n}\delta_{R_i^y} + \frac{1}{n+1}\delta_\infty \tag{7.7}$$

Finally, we construct the prediction interval containing all y where

$$\hat{C}(X_{n+1}) = \{y \in \mathbb{R} : R_{n+1}^y \leq Quantile\,(1-\alpha; F_R(y))\} \tag{7.8}$$

This procedure guarantees that $P(y_{n+1} \in \hat{C}(X_{n+1})) \geq 1-\alpha$ for any exchangeable distribution of the data and any choice of nonconformity measure [11].

7.4.1.1 Inductive Conformal Prediction

The previously mentioned conformal prediction approach is computationally heavy as it requires fitting $n \cdot \#\{\mathbb{R}\} + 1$ estimators $\hat{\mu}$. Inductive or split conformal prediction (ICP), introduced by Papadopoulos et al. [10], tackles this computation issue by splitting the training sequence $Z_{1:n} = \{Z_1, ..., Z_n\}$ into two sets: the proper training set $Z_{1:m} = \{Z_1, ..., Z_m\}$ and the calibration set $Z_{m+1:n} = \{Z_{m+1}, ..., Z_n\}$. A single regression model $\hat{\mu}$ is fit on the proper training set while the nonconformity scores (e.g., $R_i = |y_i - \hat{\mu}(X_i)|, i = m+1, ..., n$) are generated from the calibration set. These scores are sorted in descending order denoted as $R_1^*, ..., R_{n-m}^*$. Then, for a new sample with features X_{n+1}, a point prediction is made $\hat{y}_{n+1} = \hat{\mu}(X_{n+1})$. Finally, given a target coverage of $1-\alpha$, the prediction interval becomes

$$\hat{C}(X_{n+1}) = [\hat{y}_{n+1} - R_s^*, \hat{y}_{n+1} + R_s^*] \tag{7.9}$$

where $s = \lfloor \alpha(n-m+1) \rfloor$ represents the $1-\alpha$ quantile of the ordered nonconformity set with size $n-m$ [17].

7.4.1.2 Weighted Conformal Prediction

Evaluating and requiring coverage guarantees for the dose-response model at all possible treatment values changes the test distribution compared to the training distribution. In the training data, all treatment values are sampled according to their (conditional) training distribution, which can be determined by other variables in the case of confounding. However, every treatment value is possible in testing, and thus, every treatment sample can be sampled. This mimics sampling a new test sample with the treatment value from a uniform distribution, which can be vastly different from the treatment distribution in the training data. Standard conformal prediction only guarantees coverage if the joint distribution of the new sample Z_{n+1} and $Z_{1:n}$ remains

the same under permutations, which is called the exchangeability assumption [2, 11]. This issue is called covariate shift; The features X_{n+1} come from a different distribution compared to $X_{1:n}$, while the relation between X and y remains the same. More formally: $X_i \sim P_X,\ i = 1, ..., n$ and $X_{n+1} \sim \tilde{P}_X$ where $\tilde{P}_X \neq P_X$ while $y_i \sim P_{Y|X},\ i = 1, ..., n$.

Weighted conformal prediction provides a solution to tackle this issue [11]. However, their main assumption is that the likelihood ratio between the training P_X and the test covariate distribution $\tilde{P}_X$ is known, defined as

$$w(x) = \frac{d\tilde{P}(x)}{dP(x)} \tag{7.10}$$

The rationale is that they reweight the distribution of nonconformity scores $F_R(y)$ to make the nonconformity scores more exchangeable with the test population by using the following weights in equation 7.7 [11]:

$$\begin{aligned} p_i^w(X_{n+1}) &= \frac{w(X_i)}{\sum_{j=1}^{n} w(X_j) + w(X_{n+1})} \\ p_{n+1}^w(X_{n+1}) &= \frac{w(X_{n+1})}{\sum_{j=1}^{n} w(X_j) + w(X_{n+1})} \end{aligned} \tag{7.11}$$

$$F_R(y) = \sum_{i=1}^{n} p_i^w(X_{n+1})\delta_{R_i^y} + p_{n+1}^w(X_{n+1})\delta_\infty \tag{7.12}$$

Consequently, these weights adjust the distribution of nonconformity scores to give more weight to nonconformity scores that are more likely in the test set and vice versa while in standard conformal prediction, every R_i has equal weight. Also, note that the weights $p^w(x)$ are normalized, cancelling out any constant terms resulting in $w(x)$ being proportional to $w(x) \propto \frac{d\tilde{P}(x)}{dP(x)}$. An extension to split weighted conformal prediction can be done similarly as in section 7.4.1.1 [11].

7.4.1.3 Conformal Predictive Systems

In some cases, providing a prediction interval often does not suffice and a complete predictive distribution is required. The extension proposed by Vovk et al. [16] produces a predictive distribution by arranging p-values, created using specific conformity measures, into a probability distribution function. A requirement to create a Conformal Predictive System (CPS) is to use a specific type of conformity measures [1] which include monotonic measures. Then, given the training data $Z_{1:n}$ and observed test sample X_{n+1}, we define an example of this specific conformity measure S and conformity scores R_i^y similar as in equations 7.3 and 7.4:

$$S((X, y), Z_{1:n}) = y - \hat{\mu}(X) \tag{7.13}$$

[1] For the specific definition see Vovk et al. [22]

With $\hat{\mu}$ an estimator fitted on the training set $Z_{1:n}$. R_i^y and R_{n+1}^y are then similarly defined as in equation 7.3 for a CPS. Then, as defined in Vovk et al. [2] we can define a predictive distribution Q for value y, using a distribution of nonconformity scores $F_R(y)$ of y to calculate $\mathbb{P}$, similarly to the quantile function in equation 7.6 as follows:

$$Q_R(y, \phi) = \mathbb{P}_{F_R(y)}\{R^y < R_{n+1}^y\} + \phi \cdot \mathbb{P}_{F_R(y)}\{R^y = R_{n+1}^y\} \tag{7.14}$$

Where ϕ is a random number sampled from a uniform distribution between 0 and 1 to ensure a smooth predictive distribution. Using the same approach as section 7.4.1.2, these conformal predictive systems can be expanded to weighted conformal predictive systems by adjusting $F_R(y)$ to account for the covariate shift [17].

Additionally, conformal predictive systems also suffer from computational issues, therefore Vovk et al. [22] introduced split conformal predictive systems to tackle the same issues in a way analogous to section 7.4.1.1.

7.4.2 Proposed methodology: Propensity Weighted Conformal Prediction

Taking into account the background knowledge of conformal prediction, we first need to formally define the target distribution to tackle our problem definition. A CADRF model $\hat{\nu}(X, T)$ is trained on triples (X, T, Y) with X d-dimensional observed covariates $X \in \mathbb{R}^d \sim P_X$ and continuous treatment variables $T \in [t_L, t_U] \sim P_{T|X}$ to predict responses $Y \in \mathbb{R} \sim P_{Y|T,X}$. P_X represents the covariate distribution, $P_{T|X}$ represents the observational conditional treatment distribution given confounders X, and $P_{Y|T,X}$ represents the outcome distribution. $P_{T|X} = P_T$ if there are no confounders for T. A CADRF model will be used to query the dose-response for all $T \in [t_L, t_U]$, creating an interventional distribution $\tilde{P}_T$. As every treatment value t is equally likely in this query we can define $\tilde{P}_T = \tilde{P}_{T|X} = Uniform(t_L, t_U)$.

To attain marginal coverage across the interventional test set for a CADRF we can use weighted conformal prediction [11]. This requires defining the weights w for X_i and treatment value t using equation 7.11, which we will call the global (g) propensity (p) weights $w_{g,p}$:

$$\begin{aligned} w_{g,p}(X_i, T_i) &= \frac{d\tilde{P}_{X,T}(X_i, T_i)}{dP_{X,T}(X_i, T_i)} = \frac{d\tilde{P}_{T|X}(X_i, T_i)d\tilde{P}_X(X_i)}{dP_{T|X}(X_i, T_i)dP_X(X_i)} \\ &= \frac{d\tilde{P}_{T|X}(X_i, T_i)dP_X(X_i)}{dP_{T|X}(X_i, T_i)dP_X(X_i)} = \frac{d\tilde{P}_{T|X}(X_i, T_i)}{dP_{T|X}(X_i, T_i)} \\ &= \frac{f_{U(t_L,t_U)}(T_i)}{\pi(T_i|X_i)} = \frac{\frac{\mathbb{1}_{[t_L,t_U]}(T_i)}{t_U - t_L}}{\pi(T_i|X_i)} \propto \frac{\mathbb{1}_{[t_L,t_U]}(T_i)}{\pi(T_i|X_i)} \end{aligned} \tag{7.15}$$

with $\mathbb{1}_{[t_L,t_U]}(T_i)$ the indicator function for $T_i \in [t_L, t_U]$.

For simplicity, we assume that there is no distribution shift for X and thus $\tilde{P}_X(X_i) = P_X(X_i)$ (The covariate shift approach for X is detailed in Appendix 7.D.1). Additionally, $f_{U(t_L,t_U)}$ is the probability density function for the uniform distribution. We also define the propensity function $\pi(T_i|X_i)$ as the probability density function for $P_{T|X}(T_i)$ as specified in Section 7.2. To generate the prediction intervals at treatment value t for a new sample X_{n+1} the weights change to $w_{g,p}(X_{n+1}, t) = \frac{1}{\pi(t|X_{n+1})}$. According to the weighted exchangeability defined in [11], this guarantees marginal coverage over the interventional distribution, for all $T \in [t_L, t_U]$, and $X \sim P_X$. Tibshirani et al. [11] also suggested a method to attain local coverage around a predetermined target point x_0 using weighted conformal prediction. Consequently, this can provide varying prediction intervals for different values of x_0 providing another heteroscedastic approach. The proposed weights, which we call the local (l) weights w_l, utilize kernel functions with bandwidth parameter h:

$$w_l^{x_0}(X_i) \propto K\left(\frac{X_i - x_0}{h}\right) \tag{7.16}$$

These weights then guarantee

$$\mathbb{P}_{x_0}\{Y_{n+1} \in \hat{C}(X_{n+1}; x_0)\} \geq 1 - \alpha \tag{7.17}$$

This assures coverage *around* x_0, but x_0 must be determined beforehand. Additionally, if a new x_0 must be evaluated, a new calibration procedure must be performed which should be considered when applying it to general regression use cases. However, for this work, the target interventional treatment distribution is known in advance and can all be computed before deployment. Consequently, for a target treatment value t we can define $w_l^t(T_i) \propto K(\frac{T_i - t}{h})$ instead.

The local weights guarantee coverage where $d\tilde{P}_T(T_i)/dP_T(T_i) \propto K(\frac{T_i - t}{h})$. To adjust the local weights for a CADRF model we need to be aware of the covariate shift introduced by evaluating the interventional distribution and thus must combine $w_{g,p}$ with w_{local} to achieve weighted exchangeability. These new weights are defined as $w_{l,p}$ for target treatment t:

$$w_{l,p}^t(X_i, T_i) \propto \frac{\mathbb{1}_{[t_L,t_U]}(T_i) K\left(\frac{T_i - t}{h}\right)}{\pi(T_i|X_i)} \tag{7.18}$$

To generate the prediction intervals for target treatment t for a new sample X_{n+1} the weights are then $w_{l,p}^t(X_{n+1}, t) = \frac{\mathbb{1}_{[t_L,t_U]}(T_i) K((t-t/h))}{\pi(t|X_i)} = \frac{\mathbb{1}_{[t_L,t_U]}(T_i)}{\pi(t|X_i)}$, which is equal to $w_{g,p}^t(X_{n+1}, t)$. By using these weights in a weighted conformal prediction framework, we provide a solution to the problem definition in Section 7.2. Theoretical coverage results of our approach are shown and discussed in Appendix 7.A.

7.5 Experiments

7.5.1 Synthetic Data

We evaluate the proposed approach on synthetic and semi-synthetic data as evaluating the true individual dose-response curve requires knowing the counterfactuals which is simply not possible in real-world data. Therefore, to evaluate the method we are forced to use (semi-)synthetic data. For the synthetic benchmarking, we used three experimental setups using synthetic data, each having different scenarios that change specific parameters. Setup 1 is inspired by Wu et al. [5] and Setup 2 follows the experimental setup of Schröder et al [21]. Both Setup 1 and 2 are clarified in Appendix 7.B. Setup 3 is novel, proposed by us, which mimics a situation where, for every scenario, two different possible dose-response functions are possible that each depends on the covariates, resulting in heavy confounding and thus limited overlap. For each scenario (over the different setups), 5000 samples were generated using 50 different random seeds resulting in 50 datasets for each scenario. These datasets were split into 25% test (1250), 25% calibration (1250), and 50% training (2500) samples. For each scenario, two different α (significance values) were evaluated (i.e., 0.1 and 0.05 for a confidence of 90% and 95% resp.). Each sample in the test set is evaluated using 40 treatment values t_0 at equal intervals between the 2% and 98% training treatment value quantile to include varying treatment overlap regions and to mimic the uniform treatment sampling. In the results, the coverage of all treatment values and all samples in the test set are aggregated to a single mean coverage for each experiment, resulting in 50 mean coverage results for every method and scenario.

7.5.1.1 Setup 3

Setup 3 is a new experimental setup proposed in this work to underline the importance of compensating for confounding in UQ for CADRF. The covariates are independently sampled from a normal distribution. The treatment T is confounded by two variables, determining the mean of the treatment assignment distribution:

$$X_1, X_2, X_3 \sim \text{Normal}(0, 5) \qquad T \sim \text{Normal}(X_2 + 0.1 \cdot X_1, 4)$$

The two scenarios have slightly different outcome distributions, as shown in Table 7.5.1. The idea is the same for both scenarios; The individual dose-response function is truly conditional and thus equal treatment values between different individuals or samples do not necessarily translate to each other. In total, there are four different possible dose-response functions depending on the covariates. Furthermore, there is heavy confounding resulting in limited samples where $T - X_2$ yields high values that in turn create large outcome values. This creates an opportunity for high epistemic uncertainty and limited overlap. For scenario two, the aleatoric uncertainty is also

heteroscedastic based on X_3 forcing solutions to look beyond the treatment value to quantify uncertainty.

Scenario	Outcome Distribution
1	$Y \sim sign(X_3) \cdot (2(T - X_2))^2 + 33T \cdot sign(X_1) + \text{Normal}(0, 2)$
2	$Y \sim sign(X_3) \cdot (2(T - X_2))^2 + 33T \cdot sign(X_1)$ $+ \frac{(sign(X_3)+1)}{2} \cdot \text{Normal}(0, 30) + \text{Normal}(0, 2)$

Table 7.5.1: The outcome distributions for setup 3

7.5.1.2 Implementation

In the case of synthetic data, the true propensity distribution, also known as the oracle distribution, is available. However, in real-world applications, the true propensity distribution is mostly unknown. As a result, any method that relies on propensity is evaluated using both the oracle propensity distribution and an estimated propensity distribution in the experiments, denoted as "Oracle" and "Propensity" in the results respectively. The estimated distribution in this work is obtained using the Conformal Prediction System (CPS), leveraging conformal prediction, though other propensity estimators could also be used. Do note that CPS quantifies total uncertainty and thus also includes the epistemic uncertainty while ideally only the aleatoric uncertainty is included. Additionally, this propensity distribution estimate is not completely guaranteed to be equal to the true conditional propensity distribution, which we theoretically need to get complete finite sample guarantees of validity. Although, in practice, this can still be a valid approximation. A learner is trained on the covariates X to predict the treatment assignment T, deemed the propensity learner. Subsequently, a CPS is calibrated for this learner using the calibration set as it is more practical to extract an empirical density distribution compared to standard conformal prediction. Since CPS produces an empirical density distribution being a sum of Dirac delta distribution similar to F_R, kernel density estimation (KDE) is applied to derive a continuous propensity density function for a treatment value t, given covariates X_i. Do note that KDE interpolates the density and depending on the KDE parameters may introduce additional epistemic error, which is a drawback of estimating the propensity in this manner. The implementation and computational discussion for Global and Local Propensity WCP is presented in Appendix 7.C.1 and our propensity estimation in Appendix 7.C.2.

For the evaluation, several baseline methods were tested and compared, including Gaussian Process, CatBoost with Uncertainty [23], Standard Conformal Prediction, and Locally Weighted Conformal Prediction (WCP Local, using weights w_l). For the proposed propensity methods we included both variations, using their respective

weights: Global Propensity-Weighted Conformal Prediction (WCP Global Oracle and WCP Global Propensity using $w_{g,p}$) and Local Propensity-Weighted Conformal Prediction (WCP Local Oracle and WCP Local Propensity, using $w_{l,p}$). The Gaussian Process was included in the comparison due to its widespread use for UQ in regression problems assuming a normal error distribution [24]. All other approaches used a CatBoost model for the base CADRF learner, chosen for its strong out-of-the-box performance [25]. As a result, the "CatBoost with Uncertainty" method was incorporated as a baseline for comparison of UQ. The propensity learner employed in the propensity-weighted approaches was a `CatboostRegressor` with 4000 iterations and default hyperparameters. Similarly, the CADRF models were a CatBoost model with 5000 iterations and default hyperparameters. The CatBoost with Uncertainty approach used the same underlying CatBoost model as the other methods to ensure consistency. For the locally weighted conformal approaches, a Gaussian kernel [26] was employed to represent local coverage. The bandwidth parameter for the kernel was set as $h = 2 \cdot (0.2 \cdot \sigma_{\hat{\pi}})^2$, where $\sigma_{\hat{\pi}}$ denotes the standard deviation of the estimated propensity distribution [2].

7.5.2 Semi-synthetic

In addition to the fully synthetic dataset, we evaluate the proposed method on a semi-synthetic dataset derived from AMICAS, an open-source patient simulator for anaesthesia drug administration, designed for multi-drug dosing control in surgical patients [27]. We used this simulator to simulate the bispectral index (BIS) of the patients, a commonly used measure of anaesthetic depth. The advantage of using this simulator is that we can also query the counterfactual using the simulator and thus evaluate the methods on the ground truth. This dataset consists of 1000 randomly generated patients with physiologically plausible characteristics sampled from the following distributions with a slightly biased dataset having more males (Gender = 0):

$$
\begin{aligned}
Age \sim Lognormal(\mu = 3.5, \sigma = 0.6) \qquad & Gender \sim Binomial(p = 0.3) \\
Height &\sim (1 - Gender) * Normal(178, 8) + Gender * Normal(164, 7) \\
Weight &\sim (1 - Gender) * Normal(75, 10) + Gender * Normal(70, 10) \\
BMI &= \frac{Weight}{(Height/100)^2} \\
LBM &= (1.1 * (1 - Gender) + 1.07 * Gender)) * Weight \\
&\quad - (128 * (1 - Gender) + 148 * Gender) * \frac{Weight^2}{Height}
\end{aligned}
$$

Each patient was administered a dose of Propofol $T_{propofol}$, an intravenous anaesthetic, drawn from a normal distribution representing the conditional propensity dis-

[2] The code of the proposed methodology and the experiments are available open-source at https://github.com/predict-idlab/dose-response-conformal-prediction

tribution incorporating confounding effects from body mass index (BMI) and lean body mass (LBM):

$$Propofol \sim \mathrm{N}\left(\mu_{prop} = 0.05 + \frac{0.0005\ age^2 + 0.3\frac{BMI}{40}^3 + 0.4\ e^{\frac{LBM}{40}}}{25}, \sigma = 0.1\right) \tag{7.19}$$

$$T_{Propofol} = max(0.05, Propofol)$$

Due to simulator constraints, a minimum propofol dose of 0.05 was enforced. Additionally, we incorporated three commonly co-administered anaesthesia-related drugs: Atracurium, Dopamine, and Sodium Nitroprusside (SNP). These can influence the effect of propofol on the BIS and are sampled as follows:

$$Atracurium \sim Binomial(0.3) * Uniform(0, 29.5)$$
$$Dopamine \sim Binomial(0.3) * Uniform(0, 20)$$
$$SNP \sim Binomial(0.3) * Uniform(0, 10)$$

Given these patient characteristics and drug administrations, the AMICAS simulator then calculated the BIS (BIS_{AMICAS}) using a Minto pharmacokinetic model, an anesthesiologist-in-the-loop setting, and no disturbance profile. The AMICAS simulator simulates the BIS across time, starting at 0 minutes to a maximum of 300 minutes representing the simulation times.

Since the BIS_{AMICAS} values are derived from a simulation rather than real-world measurements, we introduced heteroscedastic noise to better approximate more complex uncertainty which can occur in reality. This noise accounts for both age-dependent measurement variability and increased uncertainty when the administered propofol deviates from the expected dose for a given patient. Specifically, the noise follows a normal distribution with a standard deviation modulated by age and dose deviation from the mean conditional propensity μ_{prop} for that patient (see Equation 7.19):

$$\epsilon \sim \mathrm{N}\left(0, \left(\frac{|T - \mu_{prop}|}{0.1}\right)^2 * ((1 - I(age < 69)) * 3 + (I(age < 69)) * 6)\right) \tag{7.20}$$

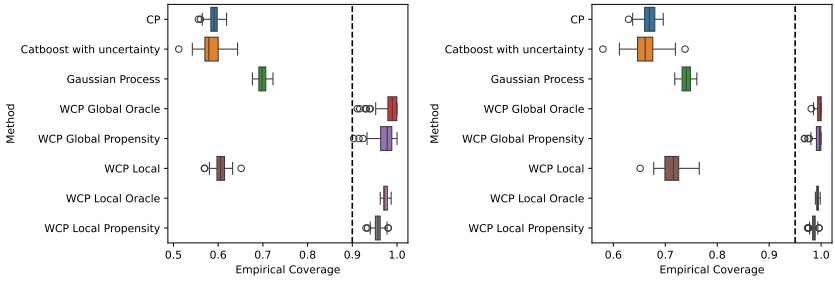
$$BIS = BIS_{AMICAS} + \epsilon \tag{7.21}$$

This standard deviation increases variability in the BIS values for older patients and for cases where the administered dose substantially deviates from the expected treatment value, introducing additional confounding in the form of added noise. This noise is also indedepently sampled for every time point and every patient.

The experimental evaluation largely follows the synthetic benchmarking with some modifications: Model training was performed using 1000 iterations for all CatBoost models and every model is trained to predict the BIS given the following features:

time, administered propofol $T_{propofol}$, age, gender, LBM, BMI, height, weight, and co-administered medication such as Dopamine, Atracurium, and SNP. The dataset was split on patient IDs into training, calibration, and test sets with proportions of 53.3%, 26.6%, and 20%, respectively. The significance levels were also evaluated at $\alpha = [0.1, 0.05]$ and the bandwidth of the KDE is set to 0.02 instead of 1. We included 12 different time points in each model, with 10 minutes being the first BIS measurement: 10, 30, 60, 80, 100, 120, 150, 170, 200, 220, 250, and 300 minutes resulting in a complete semi-synthetic dataset of 12000 samples. All models were included in the evaluation except for Gaussian Processes (GP) due to the computational constraints of Gaussian Processes. The propofol treatments t_0 of test patients were assessed over a dose range from 0.05 to 0.5, with increments of 0.01 resulting in 45 evaluations per time point per patient. The counterfactual is evaluated using the AMICAS simulator and a new noise value is generated to ensure independent noise across counterfactuals. This is repeated using 100 different random seeds for splitting aggregated, similar to the synthetic benchmarking[3].

7.6 Results

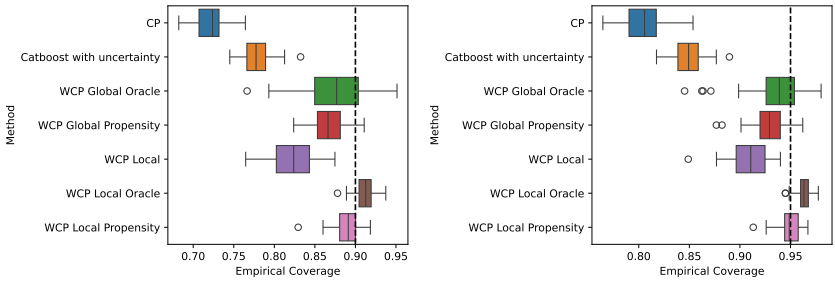


(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.6.1: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 3 scenario 1. The black dotted line is the ideal coverage.

Figure 7.6.1 presents the coverage bar plots across all methods for Setup 3 Scenario 1 on the test set. More evaluations and CADRF RMSE on all synthetic setups and scenarios can be found in Appendix 7.F. The bar plots in Figure 7.6.1 clearly illustrate the impact of covariate shift in the treatment on coverage guarantees for methods that did not account for this shift. All propensity-weighting methods assumed uniform treatment sampling during evaluation, mimicking the interpretation of a dose-response curve for decision-making for all treatment values, keeping their coverage guarantees.

[3]The code and data are also available on GitHub



(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.6.2: Barplot of the mean coverage calculated over 45 treatment values in 100 experiments for the AMICAS semi-synthetic evaluation. The black dotted line is the ideal coverage.

As can be seen in Figure 7.6.1, the global propensity-weighting method shows a high variance in coverage across different experiments. This variance arises due to the calibration process, which considers all possible treatment values between t_L and t_U, including those with minimal or no overlap. Depending on the calibration and test set split, certain samples may receive a significantly large likelihood ratio, thereby assigning considerable weight to those values according to Equation 7.12. This inflates the size of the prediction intervals, leading to conservative estimates. The oracle estimates are also notably more conservative, as they tend to provide narrower propensity distributions. This increases the frequency of large likelihood ratios when compared to the estimated propensity distribution, where the epistemic uncertainty of the propensity learner is also taken into account by the CPS procedure. On the contrary, for a new sample, the local propensity method uses calibration samples with treatment values close to the predefined value t_0 and weighting the propensities as well. Our presented approach uses more comparable calibration samples rather than the entire dataset, resulting in more conditional prediction intervals, provided there are enough calibration samples. Our method thus combines the strengths of both the local and the propensity weighting techniques. The same conclusions for the local propensity calibration can be made on the results of the semi-synthetic benchmarking, presented in Figure 7.6.2b. In the semi-synthetic benchmarking, the global (oracle) model has a high variance in performance attributed to the severe heteroscedastic noise in the semi-synthetic dataset that becomes worse in regions with less treatment overlap.

These trends are additionally supported by Figure 7.6.3, which visually shows the prediction intervals for all weighting methods alongside the treatment assignment distribution for a specific test observation in the synthetic dataset Setup 3 Scenario 1. This example highlights the necessity of the uniform treatment sampling assumption for the evaluation of dose-response curves, as both the local weighting method and standard conformal prediction produce inaccurate prediction intervals in regions with low

treatment overlap. In these regions, there is insufficient data to support predictions for the model, making these predictions unreliable. Consequently, propensity-weighted methods produce much larger prediction intervals in these areas to compensate for this lack of data support. If there is almost no support or extremely low propensity values, then the propensity-weighted methods provide intervals with an infinite width to show that there is no support in these regions. It is important to note, however, that these intervals may be overly conservative if the model has indeed generalized effectively in such regions. The only way to validate this is through additional data collection in these areas to confirm the model's performance.

Note that Schröder et al. [21] also introduced a conformal prediction method to provide prediction intervals in the continuous treatment setting. However, we did not include a direct comparison in this study due to the high computational complexity of their approach, which would require several years to complete the same experiments we executed in a matter of hours. For a more detailed comparison, including a discussion of the difference in assumptions and methodologies, see Appendix 7.E.

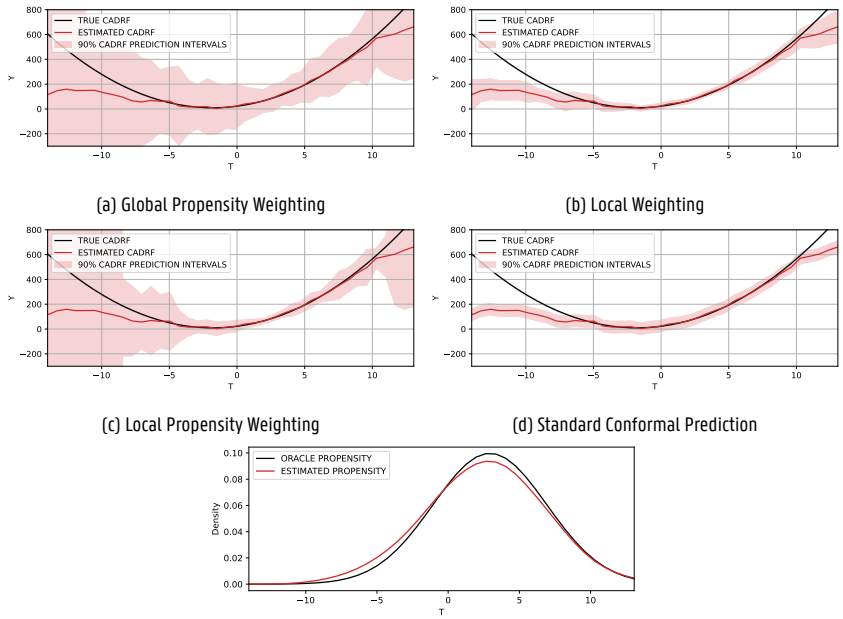


Figure 7.6.3: CADRF UQ Example on Setup 3 Scenario 1 using estimated propensity

Implementing local propensity weighting in practice is less straightforward as it involves calibrating for a set of predefined treatment values and either storing these models for later use during inference or performing this action in parallel. This has the advantage that it allows conditional prediction intervals to be calculated more quickly

during inference. However, a drawback is that evaluating a treatment value not included in the predefined set requires recalibration, and must be considered for inference. Still, this approach is particularly useful in fields like drug dosing, where treatment ranges are often predefined and personalized CADRF is highly relevant or where inference of new treatment values is not time-critical. Additionally, an important factor to consider is the effective sample size $\hat{n}$ in local propensity weighting [11, 17]. Reweighting $F_R(y)$ can significantly reduce the effective sample size, which increases variability in empirical coverage compared to standard conformal prediction. This issue is especially pronounced in regions with low treatment overlap, where the effective sample size can become extremely small. However, as prediction intervals with infinite length are possible using weighted conformal prediction, these infinite intervals additionally provide information to the user where the model cannot be trusted adding an interpretability layer to the UQ. In the current work, only an S-learner was used as a CADRF estimator which could influence the epistemic error, so in future work, more specialised dose-response models can be used to reduce the interval widths and provide even more informative prediction intervals.

Our current approach can be readily extended by incorporating other conformal prediction frameworks that support weighted conformal prediction, such as adaptive conformal prediction [8] or weighted conformal predictive systems [17]. Additionally, the weighting can be further expanded or changed to account for other types of covariate shifts in a similar manner or serve different purposes such as evaluating interventions of causal effects, thus broadening the applicability of the proposed method, as detailed in Appendix 7.D.

7.7 Conclusion

In this work, we have introduced a novel approach to weighted conformal prediction for UQ in dose-response models, utilizing propensity estimation and kernel functions as weights for the likelihood ratio. Alongside a newly proposed synthetic dataset, our approach highlights the necessity of compensating for the covariate shift in the treatment assignment when evaluating dose-response models across all possible treatment values. This is achieved by assuming uniform treatment sampling during testing, similar to methods used in discrete treatment effect estimation. Additionally, by leveraging conformal predictive systems to estimate propensity distributions, we offer a practical solution to implement UQ in continuous dose-response estimation for various practical use cases.

Our contribution not only adds to the field of dose-response modelling but also facilitates delivering reliable, individualized dose-response functions. Our approach has the potential to aid decision-making for personalized dosing in fields such as marketing, policy-making, and healthcare. With this UQ for continuous treatments, we are one step closer to achieving truly personalized interventions that optimize outcomes

for individuals.

Statements and Declarations

Data and code availability

All code, data, and additional results are available open-source on https://github.com/predict-idlab/dose-response-conformal-prediction.

Appendix

Appendix 7.A Finite Sample Coverage Guarantees

For counterfactual prediction intervals, the ideal goal is to achieve the following general conditional coverage guarantee:

$$\mathbb{P}_{Y\sim P_{Y|T=t,X=x}}(Y(t) \in \hat{C}(x,t)|X=x) \geq 1-\alpha, \text{where } t \in [t_L, t_U] \tag{7.22}$$

which, under the *strong ignorability assumption*, is equivalent to:

$$\mathbb{P}_{Y\sim P_{Y|T=t,X=x}}(Y \in \hat{C}(x,t)|X=x, T=t) \geq 1-\alpha. \tag{7.23}$$

However, constructing non-trivial prediction intervals with such conditional guarantees is generally impossible without additional modelling assumptions, as shown in Foygel Barber et al [28]. Even under the relaxed conditional guarantee, where conditioning is only on the treatment value, as in binary treatment settings [18]:

$$\mathbb{P}_{Y\sim P_X \times P_{Y|T=t,X}}(Y \in \hat{C}(X,t)|T=t) \geq 1-\alpha, \tag{7.24}$$

the problem persists when the treatment variable t is continuous.

7.A.1 Proposed Framework

To address this challenge, we introduce a distribution shift in the treatment variable by moving from the generalized propensity distribution to a user-specified interventional distribution, $T_{n+1} \sim \tilde{P}_{T|X}$. We then leverage the weighted conformal prediction (WCP) framework to construct prediction intervals. This approach allows us to build on prior theoretical coverage results under both oracle and estimated likelihood functions[11, 18].

Table 7.A.1 outlines the two interventional distributions utilized in this work: global propensity, local propensity, and δ-propensity (Dirac delta). The latter corresponds to a hard intervention. Relaxing the δ-propensity to the local propensity enables the construction of non-trivial prediction intervals (see Remark 4). Notably, when $T \in \{0,1\}$, our approach under δ-propensity aligns with the counterfactual inference framework for binary treatments proposed in Lei and Candès [18].

General	Global propensity	Local propensity	δ-propensity
$\tilde{P}_{T\|X}$	$Uniform(t_L, t_U)$	$\frac{\mathbb{1}_{[t_L,t_U]}(T)K(\frac{T-t}{h})}{\int_{t_L}^{t_U}\mathbb{1}_{[t_L,t_U]}(T)K(\frac{T-t}{h})dT}$	$\delta(T-t)$
$w(X,T)$	$\frac{\mathbb{1}_{[t_L,t_U]}(T)}{\pi(T\|X)}$	$\frac{\mathbb{1}_{[t_L,t_U]}(T)K\left(\frac{T-t}{h}\right)}{\pi(T\|X)}$	$\frac{\delta(T-t)}{\pi(T\|X)}$
$\hat{w}(X,T)$	$\frac{\mathbb{1}_{[t_L,t_U]}(T)}{\hat{\pi}(T\|X)}$	$\frac{\mathbb{1}_{[t_L,t_U]}(T)K\left(\frac{T-t}{h}\right)}{\hat{\pi}(T\|X)}$	$\frac{\delta(T-t)}{\hat{\pi}(T\|X)}$

Table 7.A.1: Translation of general interventional distribution framework to WCP global, local, and δ-propensity.

7.A.2 Propostion: Finite-Sample Guarantees

Proposition 1 (following Tibshirani et al.[11]; Lei and Candès [18]). *Assume* $(X_i, T_i, Y_i) \overset{i.i.d.}{\sim} P_X \times P_{T|X} \times P_{Y|T,X}$, $i = 1, ..., n$; *the likelihood ratio* $w(X,T) \propto \frac{d\tilde{P}_{T|X}}{dP_{T|X}}$; *and the estimated likelihood ratio* $\hat{w}(X,T)$. *Using WCP to construct* $\hat{C}(X,T)$, *the following finite-sample bounds apply:*

S1. Oracle Likelihood Ratio

If $\hat{w}(\cdot,\cdot) = w(\cdot,\cdot)$, *i.e. oracle likelihood ratio function; then,*

$$1 - \alpha \leq \mathbb{P}_{(X,T,Y)\sim P_X \times \tilde{P}_{T|X} \times P_{Y|T,X}}\{Y \in \hat{C}(X,T)\} \tag{7.25}$$

S2. Finite Sample with Regularity Conditions

If $\hat{w}(\cdot,\cdot) = w(\cdot,\cdot)$; *the non-conformity scores* S_i *have no ties almost surely;* $\tilde{P}_{T|X} \times P_X$ *is absolutely continuous with respect to* $P_{T|X} \times P_X$; *and* $(\mathbb{E}_{(X,T)\sim P_X \times P_{T|X}}[w(X,T)^r])^{\frac{1}{r}} \leq M_r < \infty$ *where* $r > 0$ *and* M_r *denotes the upper bound of the* r*-th moment of the likelihood ratio; then,*

$$1 - \alpha \leq \mathbb{P}_{(X,T,Y)\sim P_X \times \tilde{P}_{T|X} \times P_{Y|T,X}}\{Y \in \hat{C}(X,T)\} \leq 1 - \alpha + cn^{\frac{1}{r-1}} \tag{7.26}$$

where c *is an arbitrary positive constant depending on* M_r *and* r.

S3. Estimated Likelihood Ratio

If $\hat{w}(\cdot,\cdot) \neq w(\cdot,\cdot)$; $\Delta_w = \frac{1}{2}\mathbb{E}_{(X,T)\sim P_X \times P_{T|X}}[|\hat{w}(X,T) - w(X,T)|]$; $(\mathbb{E}_{(X,T)\sim P_X \times P_{T|X}}[\hat{w}(X,T)^r])^{\frac{1}{r}} \leq M_r < \infty$; *and further assuming the same assumptions as in* ***S2.****; then,*

$$1-\alpha-\Delta_w \leq \mathbb{P}_{(X,T,Y)\sim P_X \times \tilde{P}_{T|X} \times P_{Y|T,X}}\{Y \in \hat{C}(X,T)\} \leq 1-\alpha+\Delta_w+cn^{\frac{1}{r-1}} \tag{7.27}$$

Proof. We can reformulate our problem as a covariate shift scenario by treating the treatment variable as part of the covariates, i.e., defining $X^* = [X,T]$. Under this transformation:

- The proof for setting ***S.1*** follows directly from Theorem 2 in Tibshirani et al. [11].

- The proof for setting ***S.2*** aligns with Proposition 1 in Lei and Candès [18]. While their work focuses explicitly on split-weighted conformalized quantile regression (CQR) [8], the argument extends to WCP because it only depends on the weighted exchangeability of nonconformity scores and the boundedness of the likelihood ratio function.

- Similarly, the proof for setting ***S.3*** follows from Theorem 3 in Lei and Candès [18], along with its corresponding derivation.

□

Remark 1. *r specifies which moment of the likelihood ratio $w(X,T)$ is being considered. Larger r corresponds to stricter regularity conditions on $w(X,T)$. M_r defines the upper bound on the r-th moment of $w(X,T)$, ensuring the likelihood ratio does not grow too large and remains well-behaved.*

Remark 2. *Note that the term $cn^{\frac{1}{r-1}}$, represents the upper bound of the expectation of maximum weight (probability), i.e., $\mathbb{E}\left[\max_{i\in[1,\dots,n]\cup\{\infty\}} p_i^w(X_{n+1})\right]$, which under no covariate shift is equal to $\frac{1}{n+1}$ the upper bound of unweighted conformal prediction.*

Remark 3. *The bounding condition assumed in **S.2** and **S.3** in Proposition 1, $(\mathbb{E}[w(X,T)^r])^{\frac{1}{r}} \leq M_r < \infty$, that $\mathbb{E}[w(X,T)^r] < \infty$ implies that $\mathbb{P}_{(X,T)\sim P_X \times P_{T|X}}(w(X) < \infty) = 1$ and $\mathbb{E}[w(X)] < \infty$ [18], i.e. $P_X \times \tilde{P}_{T|X}$ is absolutely continuous with respect to $P_X \times P_{T|X}$.*

Remark 4. *For setting **S.1**, the overlap or positivity assumption can be violated, i.e., $\frac{d\tilde{P}_{T|X}}{dP_{T|X}} = \infty$ in terms of the interventional distribution. However, this results in the trivial interval $(-\infty,\infty)$, since $w(X_i) = 0, \forall i \in [1,\dots,n]$ and $w(X_{n+1}) = \infty$ resulting in $p_i^w(X_{n+1}) = 0, \forall i \in [1,\dots,n]$ and $p_{n+1}^w = 1$.*

Remark 5. *Since inductive (or split) conformal prediction is a special case of conformal prediction, Proposition 1 also applies to inductive conformal prediction, which we use in our experiments.*

Remark 6. *With an estimated likelihood ratio under weighted CQR, our approach also follows the asymptotic double robustness result (see Theorem 1 [18]).*

Appendix 7.B Synthetic Data

7.B.1 Setup 1

For setup 1, inspired by Wu et al. [5], six independent covariates are sampled from various distributions representing both continuous and discrete values:

$$\begin{aligned} X_1, X_2, X_3, X_4 &\sim \text{Normal}(0, 1) \\ X_5 &\sim \text{Uniform}[-2, 2] \text{ (Integer)} \\ X_6 &\sim \text{Uniform}(-3, 3) \end{aligned}$$

The treatment value is confounded by all variables in this setup and thus determined by a treatment function T_μ. All scenarios share the same treatment function except for scenario 3, where a quadratic term was added. The treatment functions are shown in Table 7.B.1.

Scenario	Treatment function
$1, 2, 4, 5, 6, 7, 8$	$T_\mu = -0.8 + X_1 + 0.1X_2 - 0.1X_3$ $+0.2X_4 + 0.1X_5 + 0.1X_6$
3	$T_\mu = -0.8 + X_1 + 0.1X_2 - 0.1X_3$ $+0.2X_4 + 0.1X_5 + 0.1X_6 + \frac{3}{2}X_3^2$

Table 7.B.1: The treatment functions for all scenarios in setup 1.

The true assigned treatment value T is then sampled from a treatment assignment distribution to add randomness and ensure some overlap in the simulated data. This treatment assignment distribution is different for various scenarios to evaluate the differences in the assumed distributions. The various functions are shown in Table 7.B.2

Now, given both the covariates X and the assigned treatment T the outcome function is defined as a random variable sampled from a normal distribution with a variance of 5, with the mean a function dependent on both the treatment and the covariates:

$$\begin{aligned} Y \sim &-1 - (2X_1 + 2X_2 + 3X_3^3 - 20X_4 - 2X_5 + 20X_6) \\ &- 0.1T(1 - X_1 + X_4 + X_5 + X_3^2) + 0.13^2|T|^3 sin(X_4) + \text{Normal}(0, 5) \end{aligned}$$

7.B.2 Setup 2

Setup 2 tests the different treatment assignment distributions in the two different scenarios, which is the same experimental setup as proposed by Schröder et al [21]. The

Scenario	Treatment T	Treatment Assignment Distribution
1	$9T_\mu + 17$	$\text{Normal}(0, 5)$
2	$15T_\mu + 22$	$\text{StudentT}(df = 2)$
3	$9T_\mu + 15$	$\text{Normal}(0, 5)$
4	$49\frac{e^{T_\mu}}{1+e^{T_\mu}} - 6$	$\text{Normal}(0, 5)$
5	$42\frac{1}{1+e^{T_\mu}} + 18$	$\text{Normal}(0, 5)$
6	$7log(\lvert T_\mu \rvert + 0.001) + 13$	$\text{Normal}(0, 4)$
7	$7T_\mu + 16$	$\text{Normal}(0, 1)$
8	$7T_\mu + 16$	$20 \cdot \text{Beta}(\alpha = 2, \beta = 8)$

Table 7.B.2: The propensity functions per scenario for Setup 1

covariates are sampled from a discrete uniform distribution. The treatment is sampled from the treatment assignment distributions shown in Table 7.B.3. The outcome function is sampled from a normal distribution with a mean determined by a sinus function based on both X and T:

$$X \sim \text{Uniform}[1, 4] \text{ (Integer)}$$
$$Y \sim sin\left((0.05\pi)(T - X)\right) + \text{Normal}(0, 0.1)$$

Scenario	Treatment Assignment Distribution
1	$T \sim p \cdot \text{Uniform}(0, 5X) + (1 - p)\text{Uniform}(5X, 40)$
2	$T \sim \text{Normal}(5X, 10)$

Table 7.B.3: The propensity functions per scenario for Setup 2 with $p \sim \text{Bernoulli}(0.3)$

Appendix 7.C Algorithm pseudocode and computational analysis

7.C.1 Propensity-based Weighted Conformal Prediction Pseudocode

Algorithm 7 presents the fit procedure for both the Local and the Global Propensity WCP, using their respective weights $w^t_{l,p}$ and $w^t_{g,p}$ for an array of treatment values we want to evaluate t_{eval}. The pseudocode is written for any Kernel, although in the experiments, we used the Gaussian kernel as presented in the methodology section. The pseudocode assumes either a pre-fitted propensity estimator $\hat{\pi}$ or having access to

an Oracle estimator. The method used to fit the propensity estimator in this paper is presented in Appendix 7.C.2. Algorithm 8 then presents how the prediction intervals for a significance level α are generated using both Local and Global Propensity WCP as the implementation is the same for both methods. The get_interval function is the prediction interval function of the WCP method.

Algorithm 7: Fit and calibrate Local or Global Propensity WCP

Function *Calibrate WCP (Training covariates X_{tr}, calibration covariates X_{cal}, training outcome y_{tr}, calibration outcome y_{cal}, training treatment values T_{tr}, calibration treatment values T_{cal}, calibrated PropensityEstimator or oracle $\hat{\pi}$, to evaluate treatments in array t_{eval}, kernel K, CADRF learner $\hat{\mu}$)*

- Fit CADRF $\hat{\mu}$ on (X_{tr}, T_{tr}) to predict y_{tr};
- Calculate propensities $\pi_{cal} = \hat{\pi}(X_{cal})$;
- **if** *Global Propensity WCP* **then**
  - Calculate weights: $w_{g,p} = 1/\pi_{cal}$;
  - Define WCP as Weighted Conformal Prediction with learner $\hat{\mu}$ and weights $w_{g,p}$ on $(X_{cal}, T_{cal}, y_{cal})$;
  - Calibrate WCP;
- **else if** *Local Propensity WCP* **then**
  - **for** t ***in*** t_{eval} **do**
    - Calculate weights: $w^t_{l,p} = K(T_{cal}, t)/\pi_{cal}$;
    - Define WCP_t as Weighted Conformal Prediction with learner $\hat{\mu}$ and weights $w^t_{l,p}$ on $(X_{cal}, T_{cal}, y_{cal})$;
    - Calibrate WCP_t;
- **Output:** Calibrated models $\{WCP_t : t \in t_{\text{eval}}\}$ for Local Propensity WCP or WCP for Global Propensity WCP;

Algorithm 8: Provide uncertainty estimates Local and Global Propensity WCP

Function ***Input:*** *Test sample X_{n+1}, calibrated PropensityEstimator or oracle $\hat{\pi}$, k to evaluate treatments in array t_{eval}, kernel K, CADRF learner $\hat{\mu}$, calibrated WCP_t for all t in t_{eval}, significance α)*

- Calculate $\pi_{n+1} = \hat{\pi}(X_{n+1})$
- Calculate weights $w = 1/\pi_{cal}$
- **for** t ***in*** t_{eval} **do**
  - Predict outcome: $\hat{\mu}(X_{n+1}, t)$
  - Obtain prediction interval: $\hat{C}^t_{n+1} = \text{get\_interval}(WCP_t, (X_{n+1}, t), \alpha, w^t)$
- **Output:** Prediction intervals $\left[\hat{C}^{t_{\text{eval},1}}_{n+1,\alpha}, \ldots, \hat{C}^{t_{\text{eval},k}}_{n+1,\alpha}\right]$

7.C.2 Propensity Distribution Estimation Pseudocode

Algorithm 9 presents the propensity distribution estimation using Conformal Predictive Systems (CPS). This results in a propensity distribution array π_{arr} with the calculated propensity density for each sample in X_{cal}. exp is the exponential function and $len(X)$ denotes the length of the array X.

Algorithm 9: Estimating the Propensity Distribution

Function *Propensity Distribution Estimation(Training covariates X_{tr}, calibration covariates X_{cal}, training treatment values T_{tr}, calibration treatment values T_{cal}, Kernel Density Estimator KD)*

Fit propensity learner on X_{tr} to predict T_{tr}
Calibrate CPS using X_{cal} and T_{cal}
Initialize π_{arr} as an array of length $len(X_{cal})$
for $i = 1$ ***to*** $len(X_{cal})$ **do**
    Fit KD on CPS($X_{cal,i}$)
    Set $\pi_{arr}[i] = exp(KD(T_{cal,i}))$

Output: Propensity array π_{arr}

7.C.3 Computational Overhead

The computational overhead is greatest for Local Propensity WCP due to the evaluation over multiple treatment values, so we will focus on this version. Let m denote the number of treatment values in the evaluation array t_{eval}. In this case, the computational overhead compared to standard weighted conformal prediction (WCP) scales linearly with the number of treatment values, i.e., $O(m \cdot \text{WCP})$, where WCP refers to the cost of standard weighted conformal prediction. In addition, calculating the propensities π_{cal} on the calibration set incurs an additional computational cost, which depends on the size of the calibration set and the chosen propensity estimator. This step can be done once beforehand, so it does not need to be repeated during each evaluation.

If the treatment values in t_{eval} are known and fixed, the calibration for each treatment value can be precomputed and stored, resulting in saved WCP_t models. This means that, during inference, the computational overhead is reduced to calculating the propensity for a single new sample once and performing m predictions using the CADRF, followed by retrieving the prediction intervals for each treatment value using the pre-calibrated WCP_t. Thus, the inference overhead is $O(m)$ for a single inference, consisting of a propensity calculation and m predictions and interval retrievals. In the case of a non-static or on-demand t_{eval}, the overhead is additive as we need $O(m \cdot WCP)$ calibrations and directly afterward $O(m)$ for the inference.

If there is no Oracle propensity estimator, we need to fit the propensity estimator, which, in our case, also involves fitting the Kernel Density Estimator (KDE) for each sample in X_{cal}, as detailed in Algorithm 9. This introduces an extra layer of computa-

tional overhead, which depends on the size of the calibration set and the output of the CPS, which is an empirical distribution of the treatment values for x_{cal}. The KDE fitting step needs to be performed for each element of X_{cal}, resulting in a complexity of $O(\text{len}(X_{\text{cal}}) \cdot \text{KDE})$, where $\text{len}(X_{\text{cal}})$ is the number of calibration samples and KDE denotes the cost of fitting the KDE.

Appendix 7.D Extensions and Applications of weighted conformal dose-response curves

Here, we discuss possible extensions and how the proposed method can be applied to various applications.

7.D.1 Extensions

The paper's current setup assumes no covariate shift in the features X between the training, calibration, and test set, i.e., $P_X = \tilde{P}_X$, to simplify the derivation of the propensity-based weights. However, in real-world applications, covariate shifts are much more common and can hamper the coverage guarantee of conformal prediction, and also thus our proposed method ([11]). If we assume $P_X \neq \tilde{P}_X$ in equation 7.15, we observe that this results in adding a multiplicative term that represents the likelihood term for the covariate shift in X. As such, both $w^t_{l,p}$ and $w_{g,p}$ can be easily adjusted to cover a covariate shift in the test set if the covariate shift is known or can be calculated, analogous to Tibshirani et al. [11], resulting in the following new weights:

$$w_{g,p}(X_i, T_i) \propto \frac{\mathbb{1}_{[t_L, t_U]}(T_i)}{\pi(T_i|X_i)} \frac{dP_X}{d\tilde{P}_X} \tag{7.28}$$

and

$$w^t_{l,p}(X_i, T_i) \propto \frac{\mathbb{1}_{[t_L, t_U]}(T_i) K\left(\frac{T_i - t}{h}\right)}{\pi(T_i|X_i)} \frac{dP_X}{d\tilde{P}_X} \tag{7.29}$$

Furthermore, because the method is built using conformal prediction, the whole approach is model-agnostic. As such any possible CADRF model that provides a dose-response curve given features and treatment can be used and thus is not limited to the presented CADRF approach in this paper.

7.D.2 Applications

The classic application is in drug dosing, where the goal is to construct a dose-response curve for every individual to facilitate decision-making when determining an optimal dose for a new patient. In a clinical trial, especially phase 1 and phase 2 where the optimal dose is being determined, the weighted conformal dose-response curve can

also act as a tool to analyse the results individually while having an estimate of the uncertainty estimates that is not biased by the treatment assignment distribution. It quantifies uncertainty for individual predictions, compensating for any treatment distribution bias. Furthermore, it highlights areas with insufficient data support with infinite prediction intervals, guiding decisions about whether further trials or treatments are necessary for specific patient subgroups. In the regions where there is support, the model predictions provide the CADRF estimate for this patient and the uncertainty regions show how the outcome would vary.

Treatment is not limited to healthcare. Treatment can be generalized as any intervention or action which opens applications in other domains. For example, in predictive maintenance, the model can optimize decisions by estimating the effect of operating pressure on the remaining useful life of equipment like valves. Similarly, in sales, it can help determine the ideal discount for specific clients to maximize the sold units, demonstrating flexibility in various domains.

7.D.3 Explainability

The application potential is also not limited to actual treatments and interventions. The method can also be used for the explainability of a model. Suppose we fitted a regression model, regressing $X = [X_1, ..., X_m]$ on Y. X is observed data; thus, any feature can be confounded or biased. By considering a feature X_i as a treatment to "intervene" in a model, this method then provides uncertainty quantification on a Ceteris Paribus curve of a model in a similar manner to a dose-response curve [4]. This curve can then give unbiased uncertainty estimates of the "true" outcome for an individual sample if that sample would have had other values for this particular feature.

An example is shown in Figure 7.D.1 using Local Propensity WCP. This example is generated using the Boston Housing data available native in sklearn [29], split into a training and calibration set using a $75/25$ split. A CatBoostRegressor using 300 iterations is fitted on a training set, and a propensity CatBoostRegressor with the same number of iterations is fitted on the training set. A CPS is used and calibrated on the calibration set for the propensity distribution estimate, similar to the experimental setup in this paper. No hyperparameter tuning is applied for simplicity, so note that the epistemic uncertainty could be further reduced. The chosen feature for generating a ceteris paribus curve is MedInc, the median income, an important variable in predicting the median house value in this dataset. The figure is for a single data sample where all other variables of this sample are kept constant except for our "treatment" MedInc. In Figure 7.D.1, it is apparent that the prediction intervals go to infinity for MedInc values below 1 and above 6.5. This indicates that there is insufficient overlap

[4] A Ceteris Paribus curve visualizes a model's predictions while keeping all features constant except for one explanatory variable. The x-axis represents the explanatory variable, and the y-axis shows the corresponding predictions.

to evaluate this sample for these values of MedInc, clearly showing a bias in the data distribution of MedInc, given the other features. Consequently, the predictions for a sample with these features but with a MedInc of, e.g., 8 cannot be trusted as the model is simply doing an interpolation in an out-of-bounds region. In the regions with support, i.e., around $1.5 < MedInc < 6.5$, we see that the model shows a linear relation with the median house value with relatively small uncertainty bounds. This analysis can be done for any other regression model in a likewise manner.

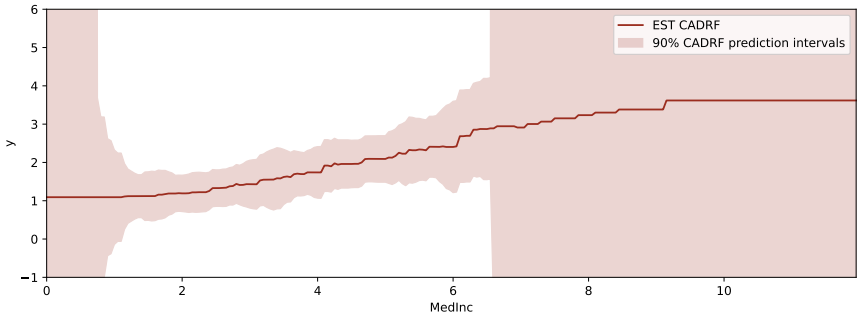


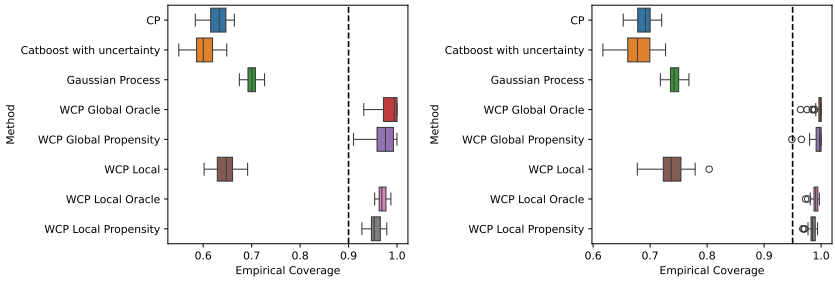
Figure 7.D.1: A Ceteris Paribus curve generated with Local Propensity WCP.

Appendix 7.E Comparison to Schröder et al.

In comparison to the work of Schröder et al. [21], our approach differs in several key aspects. First, the aim of their work is different from ours. The aim of their work [21] is to provide prediction intervals for the causal effect of treatment interventions where the treatment value is continuous. In our work, the goal is to provide prediction intervals for dose-response models instead of treatment interventions, answering a different causal question. However, adjusting our work to interventions is possible; In the case of soft interventions, the target distribution propensity changes and thus substituting the current uniform distribution in the weights $w(x)$ with the new target propensity distribution covers the soft intervention case. For hard interventions, this is an evaluation for a single treatment value which is similar to the local propensity method, but for only that target treatment value. Secondly, their approach differs in their conformal prediction approach where they want to provide correct prediction intervals for a single sample, single α value, and single treatment using a mathematical solver based on the proposed weighted conformal prediction by Gibbs and Candes [12]. Thirdly, they frame the propensity or covariate shift differently as either a Dirac distribution for a hard intervention, or a different propensity distribution in the case of a soft intervention. This is a direct consequence of their aim to quantify the causal effect of a single intervention, compared to providing a dose-response

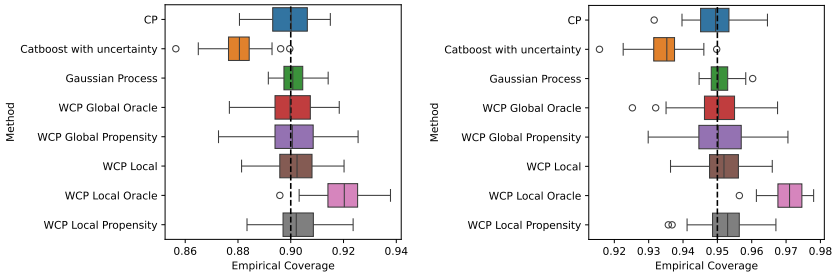
model in our case which requires a uniform assumption. Fourthly, the experimental setup of Schröder et al. [21] does not address the impact of a treatment covariate shift as shown by Figure 7.F.2 and Figure 7.F.3 where even standard conformal prediction (CP) achieves the required empirical coverage. Lastly, we also approach the propensity estimation in cases with unknown propensity as an uncertainty quantification problem and tackle it with conformal predictive systems. In the end, our approach offers a different solution on continuous treatment effects through dose-response modelling.

Appendix 7.F Additional Results



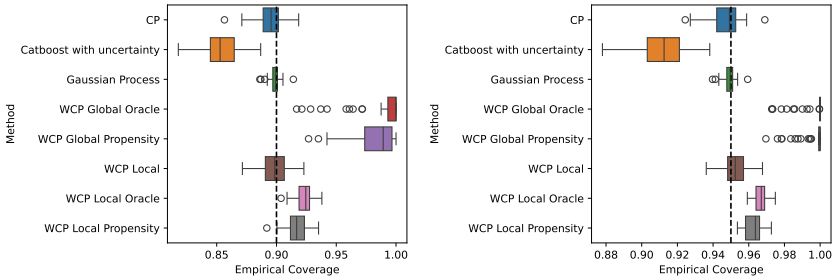
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.1: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 3 scenario 2. Black dotted line is the ideal coverage.



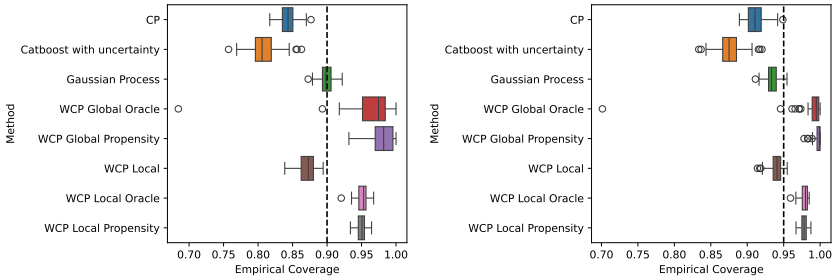
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.2: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 2 scenario 1. Black dotted line is the ideal coverage.



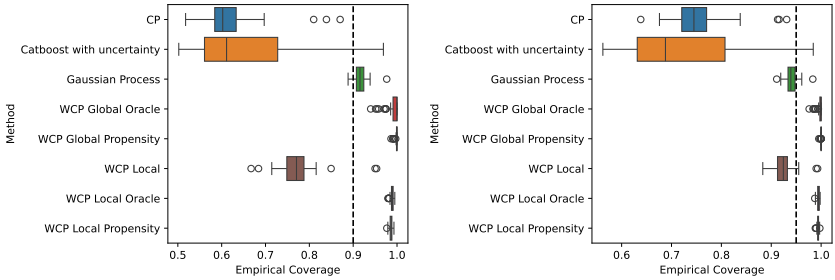
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.3: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 2 scenario 2. Black dotted line is the ideal coverage.



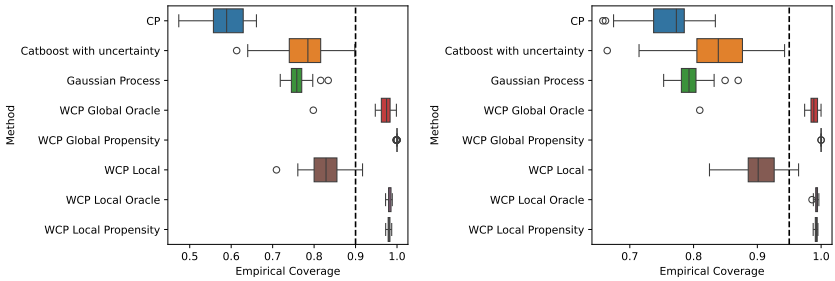
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.4: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 1. Black dotted line is the ideal coverage.



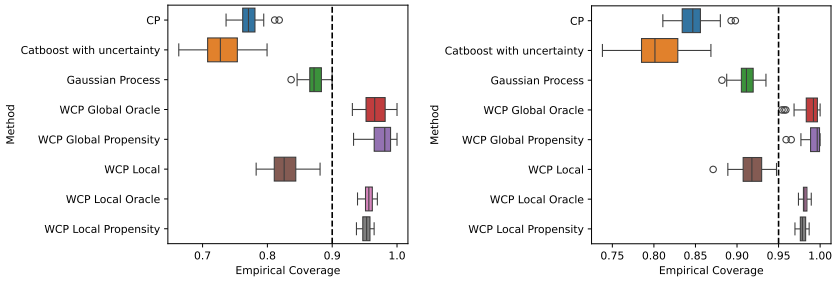
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.5: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 2. Black dotted line is the ideal coverage.



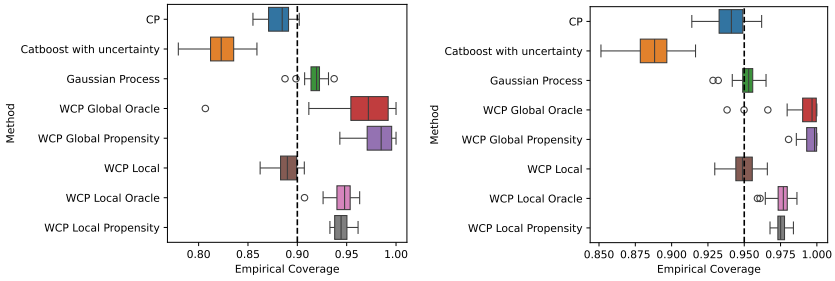
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.6: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 3. Black dotted line is the ideal coverage.



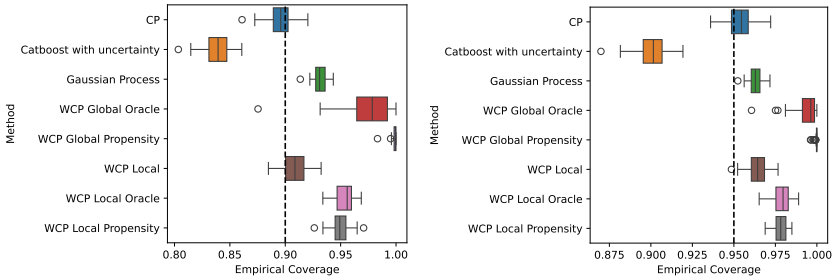
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.7: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 4. Black dotted line is the ideal coverage.



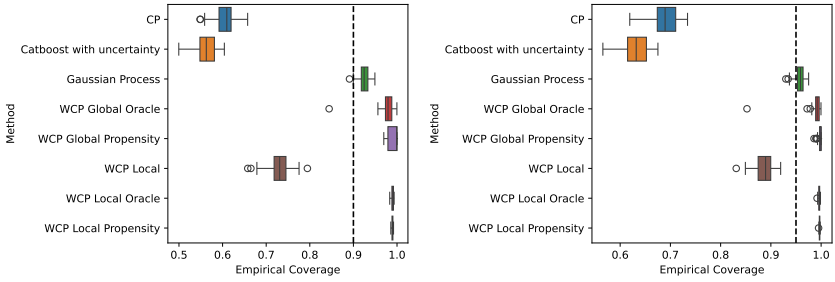
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.8: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 5. Black dotted line is the ideal coverage.



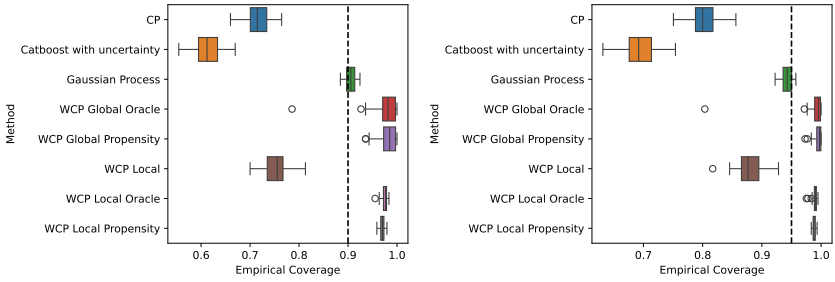
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.9: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 6. Black dotted line is the ideal coverage.



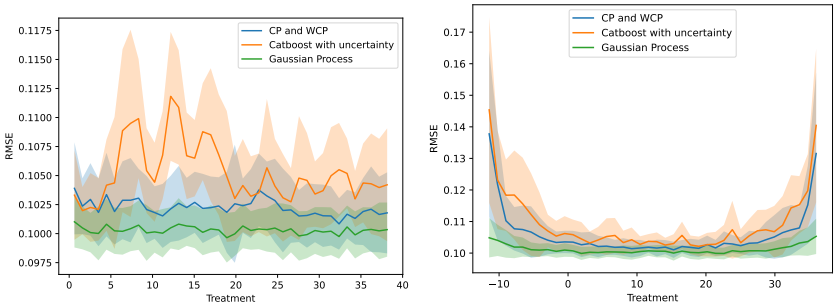
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.10: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 7. Black dotted line is the ideal coverage.



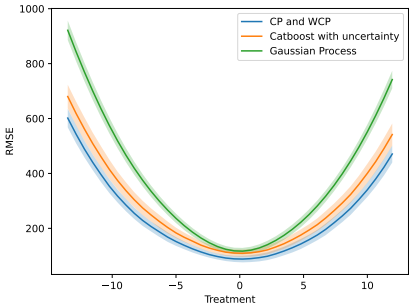
(a) $\alpha = 0.1$ (b) $\alpha = 0.05$

Figure 7.F.11: Barplot of the mean coverage calculated over 40 treatment values in 50 experiments for setup 1 scenario 8. Black dotted line is the ideal coverage.

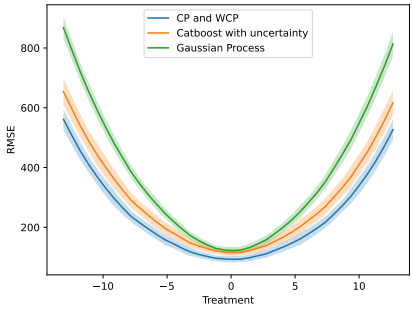


(a) Setup = 2, Scenario = 1

(b) Setup = 2, Scenario = 2

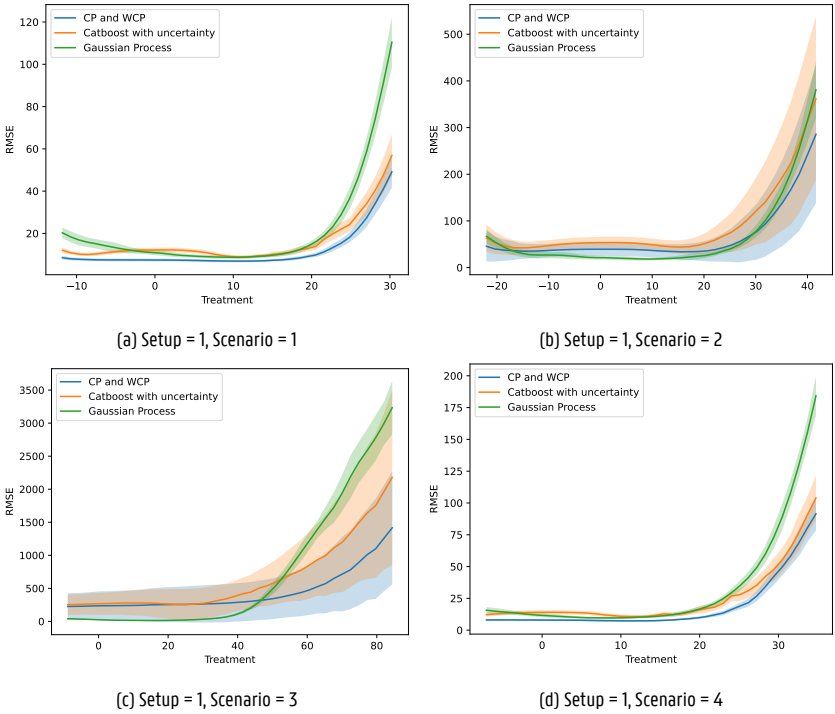


(c) Setup = 3, Scenario = 1



(d) Setup = 3, Scenario = 2

Figure 7.F.12: Plot of the CADRF RMSE with $\pm$ RMSE standard deviation across all repeated experiments for the considered treatment values for setup 2 and setup 3. As All WCP and CP methods use the same fitted base CatBoost CADRF learner they are represented by "CP and WCP".



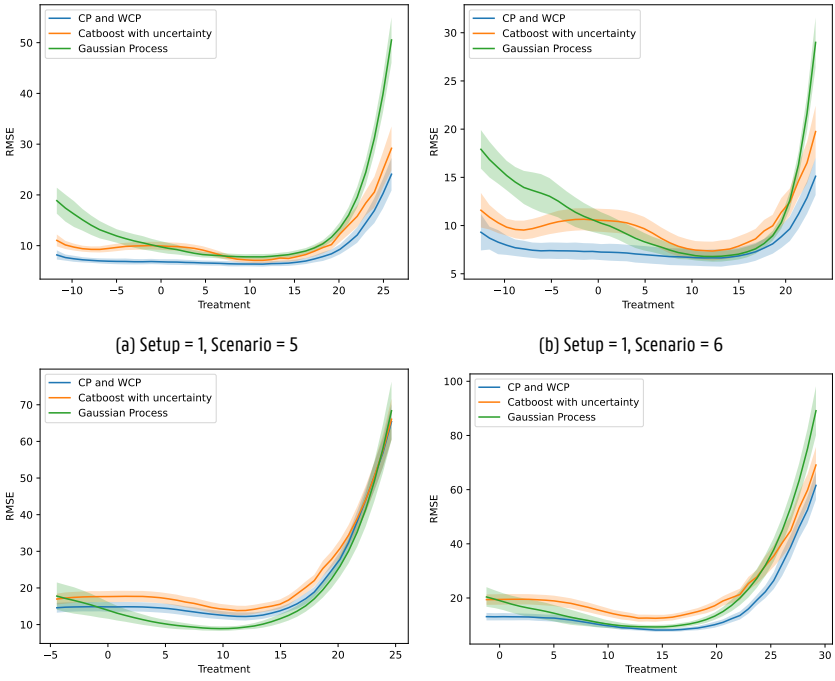
(a) Setup = 1, Scenario = 1

(b) Setup = 1, Scenario = 2

(c) Setup = 1, Scenario = 3

(d) Setup = 1, Scenario = 4

Figure 7.F.13: Plot of the CADRF RMSE with $\pm$ RMSE standard deviation across all repeated experiments for the considered treatment values for setup 1, scenarios 1 to 4. As All WCP and CP methods use the same fitted base CatBoost CADRF learner they are represented by “CP and WCP”.



(a) Setup = 1, Scenario = 5

(b) Setup = 1, Scenario = 6

(c) Setup = 1, Scenario = 7

(d) Setup = 1, Scenario = 8

Figure 7.F.14: Plot of the CADRF RMSE with $\pm$ RMSE standard deviation across all repeated experiments for the considered treatment values for setup 1, scenarios 5 to 8. As All WCP and CP methods use the same fitted base CatBoost CADRF learner they are represented by "CP and WCP".

References

[1] Stefan Feuerriegel, Dennis Frauen, Valentyn Melnychuk, Jonas Schweisthal, Konstantin Hess, Alicia Curth, Stefan Bauer, Niki Kilbertus, Isaac S. Kohane, and Mihaela van der Schaar. *Causal machine learning for predicting treatment outcomes*. Nature Medicine, 30(4):958–968, April 2024. Publisher: Nature Publishing Group.

[2] Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer International Publishing, Cham, 2022.

[3] Donald B Rubin. *Causal Inference Using Potential Outcomes*. Journal of the American Statistical Association, 100(469):322–331, March 2005.

[4] Keisuke Hirano and Guido W. Imbens. *The Propensity Score with Continuous Treatments*. In Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives, pages 73–84. John Wiley & Sons, Ltd, 2004.

[5] Xiao Wu, Fabrizia Mealli, Marianthi-Anna Kioumourtzoglou, Francesca Dominici, and Danielle Braun. *Matching on Generalized Propensity Scores with Continuous Exposures*. Journal of the American Statistical Association, 119(545):757–772, January 2024. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2022.2144737.

[6] Paul R. Rosenbaum and Donald B. Rubin. *The central role of the propensity score in observational studies for causal effects*. Biometrika, 70(1):41–55, April 1983.

[7] Susan Athey, Julie Tibshirani, and Stefan Wager. *Generalized random forests*. The Annals of Statistics, 47(2):1148–1178, April 2019. Publisher: Institute of Mathematical Statistics.

[8] Yaniv Romano, Evan Patterson, and Emmanuel Candes. *Conformalized Quantile Regression*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[9] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. *Distribution-Free Predictive Inference for Regression*. Journal of the American Statistical Association, 113(523):1094–1111, July 2018. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2017.1307116.

[10] Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. *Inductive Confidence Machines for Regression*. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, Machine Learning: ECML 2002, pages 345–356, Berlin, Heidelberg, 2002. Springer.

[11] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. *Conformal Prediction Under Covariate Shift*. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

[12] Isaac Gibbs and Emmanuel Candes. *Adaptive Conformal Inference Under Distribution Shift*. In Advances in Neural Information Processing Systems, volume 34, pages 1660–1672. Curran Associates, Inc., 2021.

[13] Isaac Gibbs and Emmanuel Candes. *Conformal Inference for Online Prediction with Arbitrary Distribution Shifts*. Journal of Machine Learning Research, 2024.

[14] Rina Foygel Barber, Emmanuel J. Candès, Aaditya Ramdas, and Ryan J. Tibshirani. *Conformal prediction beyond exchangeability*. The Annals of Statistics, 51(2):816–845, April 2023. Publisher: Institute of Mathematical Statistics.

[15] Leying Guan. *Localized conformal prediction: a generalized inference framework for conformal prediction*. Biometrika, 110(1):33–50, March 2023.

[16] Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. *Nonparametric predictive distributions based on conformal prediction*. Machine Learning, 108(3):445–474, March 2019.

[17] Jef Jonkers, Glenn Van Wallendael, Luc Duchateau, and Sofie Van Hoecke. *Conformal Predictive Systems Under Covariate Shift*, April 2024. arXiv:2404.15018 [cs, stat].

[18] Lihua Lei and Emmanuel J. Candès. *Conformal Inference of Counterfactuals and Individual Treatment Effects*. Journal of the Royal Statistical Society Series B: Statistical Methodology, 83(5):911–938, November 2021.

[19] Jef Jonkers, Jarne Verhaeghe, Glenn Van Wallendael, Luc Duchateau, and Sofie Van Hoecke. *Conformal Convolution and Monte Carlo Meta-learners for Predictive Inference of Individual Treatment Effects*, June 2024. arXiv:2402.04906 [cs, stat].

[20] Ahmed M Alaa and Zaid Ahmad. *Conformal Meta-learners for Predictive Inference of Individual Treatment Effects*. In Advances in Neural Information Processing Systems. Curran Associates, Inc., 2024.

[21] Maresa Schröder, Dennis Frauen, Jonas Schweisthal, Konstantin Heß, Valentyn Melnychuk, and Stefan Feuerriegel. *Conformal Prediction for Causal Effects of Continuous Treatments*, July 2024. arXiv:2407.03094 [cs, stat].

[22] Vladimir Vovk, Ivan Petej, Ilia Nouretdinov, Valery Manokhin, and Alexander Gammerman. *Computationally efficient versions of conformal predictive distributions*. Neurocomputing, 397:292–308, 2020.

[23] Tony Duan, Anand Avati, Daisy Yi Ding, Khanh K. Thai, Sanjay Basu, Andrew Y. Ng, and Alejandro Schuler. *NGBoost: Natural Gradient Boosting for Probabilistic Prediction*, October 2019.

[24] Christian Fiedler, Carsten W. Scherer, and Sebastian Trimpe. *Practical and Rigorous Uncertainty Bounds for Gaussian Process Regression*. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8):7439–7447, May 2021. Number: 8.

[25] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. *CatBoost: gradient boosting with categorical features support*. arXiv:1810.11363 [cs, stat], October 2018. arXiv: 1810.11363.

[26] Sergios Theodoridis. *Chapter 11 - Learning in Reproducing Kernel Hilbert Spaces*. In Sergios Theodoridis, editor, Machine Learning, pages 509–583. Academic Press, Oxford, January 2015.

[27] Clara M. Ionescu, Martine Neckebroek, Mihaela Ghita, and Dana Copot. *An Open Source Patient Simulator for Design and Evaluation of Computer Based Multiple Drug Dosing Control for Anesthetic and Hemodynamic Variables*. IEEE Access, 9:8680–8694, 2021.

[28] Rina Foygel Barber, Emmanuel J Candès, Aaditya Ramdas, and Ryan J Tibshirani. *The limits of distribution-free conditional predictive inference*. Information and Inference: A Journal of the IMA, 10(2):455–482, 08 2020.

[29] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. *API design for machine learning software: experiences from the scikit-learn project*. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning, pages 108–122, 2013.

8

Concluding discussion and future work

"Don't adventures ever have an end? I suppose not. Someone else always has to carry on the story."

–J.R.R. Tolkien, The Fellowship of the Ring

The development of a trustworthy personal assistant or "personal chef" for healthcare practitioners using machine learning and artificial intelligence is an ambitious task. Focusing this effort on dosing and continuous treatment advice narrows the scope, yet the task remains substantial and complex. Achieving this goal requires overcoming several significant challenges, some of which are explored in this dissertation, however, there are many more challenges to tackle to reach this end goal. These include the need for high-quality data, the complexity of medical decision-making, and the ethical considerations of integrating AI into healthcare, to name only a few. Addressing these challenges demands not only innovative technological solutions, but also a deep understanding of the practical needs of healthcare providers and a commitment to rigorous evaluation and continuous improvement. While the journey to developing such a system is long, the potential to enhance patient care and support practitioners is immense.

This dissertation aimed to advance the development of trustworthy ML models for dosing and treatment advice in healthcare. I introduced Powershap, a time-efficient method for identifying relevant features from data, which significantly enhances model building and prototyping. I then constructed models for two example use cases, integrating uncertainty quantification (UQ) to bolster trustworthiness and proposing adjusted UQ metrics for evaluation. Using Causalteshap, I identified predictive features for the treatment effect, providing insights that can guide clinical strategies and

enhance patient outcomes. Finally, our propensity-weighted conformal prediction ensures reliable uncertainty estimates for 'what-if' scenarios, enabling clinicians to explore different treatment options confidently. Together, these contributions fill a technological toolbox to work towards the development of a personal treatment assistant for healthcare providers. By addressing key challenges in feature selection, uncertainty quantification, and causal inference, this work paves the way for more trustworthy treatment models in healthcare.

8.1 A review of the Research Goals

In the introduction chapter, we formalized four specific research goals aimed at tackling the overarching research goal. Having presented the chapters, we can now revisit these goals and discuss how each chapter contributes to them. Let us begin by reintroducing the overarching research goal that could be seen as the mission statement of this dissertation:

Overarching RG: "Build trustworthy machine learning models for the ICU that can be used for treatment decision support while incorporating calibrated uncertainty quantification."

This overarching research goal serves as a guiding principle to keep in mind while formulating the more specific research goals and tackling them. However, to build these models, we required feature selection, especially in medical data with numerous possible feature candidates. Chapters 3, 4, and 5 used Powershap for their feature selection, which is the solution to RG1:

RG1: "Create an effective and time-efficient method for combining wrapper and filter feature selection techniques to identify relevant features, leveraging the strengths of both approaches."

Powershap is based on the assumption that irrelevant features should have an equal or lower impact compared to a known random feature. By adding a random feature to a machine learning model and comparing the absolute Shapley values of each feature to those of the random feature, it is possible to evaluate this assumption or null hypothesis. Common wrapper methods, such as genetic, forward, or backwards feature selection, scale poorly in time complexity as the number of features increases. Filter methods, on the other hand, only consider the data and do not account for model interactions, often relying on assumptions, leading to less accurate feature selection. Powershap combines the strengths of both approaches: it trains a model, quantifies feature impacts, and then filters relevant features by statistically comparing the mean impact of each feature to that of the random feature. The various benchmark experi-

ments show that this method is leagues faster compared to forward feature selection and also faster compared to other Shapley-based feature selection methods. Additionally, the recall of finding the relevant features stayed comparatively high, even with a high number of features, while keeping the execution time low. The "automatic" setting in Powershap additionally makes it ideal for a plug-and-play limited effort solution for many ML models, speeding up model building and prototyping considerably if compared to using only wrapper feature selection methods. Powershap was published as an open-source library with at the time of writing more than 1 million downloads and 200+ stars and is used in companies and various research papers [1–3]. With RG1 tackled, we have the feature selection method to build trustworthy models.

I illustrated our overarching research goal by introducing two use cases. These use cases also provided a means to validate our other research goals. In addition, each use case had its own specific use case goal (UCG1 and UCG2). These use case goals additionally focused on the innovation within each context and translated the overarching goal into practical applications. For reference, UCG1 and UCG2 are:

UCG1: "Build a trustworthy atrial fibrillation risk prediction model for ICU patients that accounts for the different ICU populations between AF and non-AF patients while providing a calibrated risk probability for any time point."

UCG2: "Model blood plasma concentrations of antimicrobials in critically ill ICU patients for optimizing antimicrobial dosing while providing calibrated uncertainty quantification for trustworthy predictions."

The overarching research goal focuses on building trustworthy machine learning models, with UQ playing a crucial role in enhancing trustworthiness. There were three relevant characteristics of trustworthiness that I introduced at the beginning of this dissertation: Causality, Uncertainty, and Robustness. This UQ component was integrated into both UCG1 and UCG2. However, formalizing the concept of a trustworthy model in these use case goals is more challenging due to the possible interpretations of trustworthiness. In UCG1, additional trustworthiness was addressed by accounting for the different ICU populations between AF and non-AF patients, aiming to reduce potential bias and increase the model's reliability. For UCG2, domain knowledge was incorporated to enhance prediction accuracy. This approach not only improved predictions but also facilitated more straightforward dosing advice by predicting the pharmacokinetic (PK) constant, which can be directly translated into the required dose, rather than merely predicting plasma concentration.

UCG1 was tackled in Chapter 5 through the proposed AF risk models. In Chapter 5, the proposed AF risk models were built by matching AF patients with non-AF patients to eliminate potential time bias, as AF patients often have longer ICU stays. Simply using the moment of AF diagnosis (which tends to be later) and a random

moment for non-AF patients (which tends to be earlier) introduced a bias, making the model less robust over time. Matching AF with non-AF patients increased robustness and enabled AF risk prediction throughout the ICU stay. The original model was a result from the European Society of Intensive Care Medicine (ESICM) 2021 International ICU Datathon and even resulted in a third place. The AF models were validated across multiple ICU datasets, addressing the accuracy aspect of trustworthiness. Chapter 5 also discussed the general approach to building a classification model for the ICU, focusing on predicting a diagnosis over time. This included feature selection (using Powershap), feature engineering (such as feature windows), model design, and the validation approach. For the UQ part of UCG1, the proposed models used inherent probabilities. Although these probabilities are often considered uncalibrated [4], I thoroughly evaluated their calibration for the matched moments and across time on the test set and showed that these were calibrated.

To properly evaluate these probabilities, two metrics were proposed: the (adjusted) Expected Calibration Error (ECE) and the Expected Signed Calibration Error (ESCE). The standard ECE evaluates calibration by binning the risk probabilities, however, the bin size is a hyperparameter that must be chosen. This choice can affect the robustness of the ECE, especially with large bins or limited samples. Therefore, in Chapter 5, I adjusted the ECE to calculate the metric as the mean across multiple bin sizes. Additionally, I removed the absolute value from the ECE to quantify any consistent calibration bias. For example, if the ESCE is consistently positive, it indicates underconfidence, suggesting that the true probabilities are higher, and vice versa. Understanding consistent under- or over-confidence is essential, as underconfidence is less severe than overconfidence, where actual probabilities are much lower, making the model less trustworthy. This addresses the UQ part of UCG1 and also contributes to research goal RG2:

RG2: "Adjust current metrics to quantify coverage error and biases, such as overconfidence or underconfidence, in uncertainty quantification methods across all confidence levels for both regression and classification tasks."

The ESCE addresses the overconfidence and underconfidence aspects of RG2 for classification tasks by providing a single metric that encompasses all confidence levels. This makes it possible to optimize model parameters to minimize potential calibration bias. Returning to UCG1, the interpretability aspect of the trustworthiness of the models was addressed by presenting and discussing the Shapley values of the features. This analysis gave rise to a medical question that will be explored in Chapter 6 using Causalteshap: "Is Noradrenalin associated with an increased risk of AF, and what variables explain or influence this risk?".

With a potential solution to UCG1 and the classification part of RG2, what about

UCG2? In Chapter 3, I proposed a Catboost regressor to predict piperacillin plasma concentrations and compared it to a GP, a neural network, and a PopPK model, where the proposed model outperformed all these models on both an internal (Ghent University Hospital) and external (University Medical Centre of Groningen) dataset. The predictive accuracy part of the trustworthiness was evaluated using the external test set. In Chapter 4, I extended the methodology to predict the clearance rate of two antimicrobials instead of the absolute concentration of a single antimicrobial by leveraging pharmacological domain knowledge. This creates a model that can predict how fast this class of antimicrobials are cleared from the body, which is essential for dose optimization. This was again validated on an internal and an external dataset. Additionally, in both chapters 3 and 4, the models are explained and interpreted using Shapley values to address the interpretability aspect of trustworthiness. These models were a result of a close collaboration with prof. dr. Jan De Waele and dr. Thomas De Corte from the Ghent University Hospital ICU demonstrating the relevance and importance of including physicians in the model building process to achieve trustworthy solutions. Additionally, this opened the opportunity to validate the models in a real ICU and for actual physicians and quantify their trustworthiness and robustness. The clinical study is still ongoing research, however, the models were presented to physicians to understand the requirements in research under review under the name: "Towards trustful machine learning for antimicrobial therapy using an explainable artificial intelligence dashboard".

Now, for the UQ, Chapter 3 introduced an ensemble method of quantile regressors to provide a predictive distribution, where I took the assumption that the predictive distribution is a Gaussian distribution. Although this assumption can be considered stringent, it made it possible to convert the ensemble of quantile gradient boosting regressors to a Normal or Gaussian distribution. Because a Normal distribution only has two parameters, only two quantile regressors would have been enough; however, I added a third one for robustness and to alleviate the risk of crossing quantiles. Now, to optimize this predictive distribution, we are also required to evaluate this distribution. Therefore, I extended the Prediction Interval Coverage Percentage (PICP) to a distribution variant called the (Absolute) Distribution Coverage Error ((A)DCE), which was used to perform hyperparameter optimization on the quantile ensemble to provide a calibrated predictive distribution. The (A)DCE tackles the regression part of RG2, while the quantile ensemble provides the UQ for tackling the last part of UCG2.

Until now, we have focused on building trustworthy models with the ultimate goal of providing dosing advice, as shown by the antimicrobial plasma concentration prediction models. However, the AF models do not directly offer dosing advice, instead, they provide an estimate of a possible outcome: the risk of AF. This opens up the possibility of analyzing treatments using such an outcome model. The analysis of treatment effects was also formulated in RG3:

RG3: "Identify predictive variables in conditional average treatment effect models while minimizing false positives for higher trustworthiness."

Drawing inspiration from Powershap, which uses Shapley values to evaluate hypotheses, I designed a similar approach to identify predictive variables, as presented in Chapter 6 with Causalteshap. Causalteshap posits two null hypotheses:

1. The feature impact of a prognostic feature is the same under control as under treatment in a CATE model.

2. For a prognostic feature, the difference in feature impact between treatment and control should not differ from that of a random feature, aiming to eliminate false positives.

By combining statistical tests for these hypotheses and quantifying feature impacts with Shapley values, Causalteshap provides a method to address RG3. Maintaining low false positive rates is crucial in this solution, as incorrect predictive variables can negatively influence or even harm decision-making by focusing on irrelevant factors. This aspect is therefore vital to the trustworthiness of the solution. To validate this, Causalteshap was subjected to extensive synthetic benchmarks and compared with other methods. The results demonstrated its ability to maintain low false positive rates, thereby fully addressing the requirements of RG3.

We now turn to the final research goal, RG4:

RG4: "Enhance dosing models to identify regions with limited or no overlap in treatment outcomes utilizing uncertainty quantification for reliable dose-response predictions."

This research goal focuses on providing UQ for dosing models when exploring "what-if" scenarios, such as "What if the dose was X for this patient?". The dose in question might not be present in the existing data for similar patients, and naive UQ methods, like standard conformal prediction, fail to provide accurate UQ for these inferences, as demonstrated in Chapter 7. The proposed solution involves quantifying the generalized propensity, i.e. the probability of a given dose in a continuous treatment, and adjusting the estimated prediction interval using weighted conformal prediction. This approach is essential for addressing "what-if" questions in dosing models and is required to fully address RG4. The newly proposed benchmarks thoroughly test the proposed method, and the results confirm its effectiveness, allowing us to confidently state that this tackles the final research goal, RG4.

Now, with the four specific research goals and the two use case goals discussed, we can revisit the overarching research goal. Every RG or UCG contributes to the overarching goal, each providing its specific solution to a part of the overarching goal.

The goal was set very broadly, and with the use cases, I showed some possible solutions. We can say that parts of the overarching goal are tackled, although there is still work to do to reach a trustworthy treatment decision support model for the ICU.

8.2 The path to a treatment decision support model in the ICU

Tackling all the presented research goals helps to advance the technological solutions to create the proposed dosing advice assistant and contribute to the overarching goal of this dissertation. However, they are not sufficient. These solutions are but ropes for a bridge to cross a canyon, yet they do have an impact on the field.

One of these ropes is Powershap, which simplifies model building and prototyping by identifying relevant features from a large pool of candidates. Powershap is ideal for things such as prototyping, first model design, and model optimization. There are still some improvements possible, like defining your random feature to make it more similar to the features in your dataset. For example, if your dataset only contains categorical features, it might be more beneficial to use a random categorical feature instead of the default random uniform feature. Additionally, Powershap can be used in a more complex optimization pipeline to reach higher performances depending on the use case. For feature selection, there is no single solution for all use cases, and often combining multiple methods or other SOTA approaches can be beneficial depending on the use case [5].

In the same vein, Causalteshap offers a new approach to identifying predictive values in advanced causal AI models. Traditionally, many of these methods quantify treatment-interaction effects, which are either linear relationships or continuous relationships [6]. Sometimes these interaction effects are step-functions, such as an on-or-off switch, or simply non-linear, requiring different causal AI models. A dedicated method to flag a feature as predictive or not, similar to a feature selection algorithm, also makes interpretation easier. Additionally, if the false positive rate is very low, we can trust the method and use it in higher-risk situations, such as healthcare. Although it is made for binary treatment, it is not necessarily limited to it. Causalteshap can be used in continuous treatments as well. For example, we can set one dose as the control ($T = 0$) and another dose as $T = 1$, Causalteshap will then find the features that explain the difference between these two doses. Another approach would be to have controls with no dose ($T = 0$), and simply compare the difference to individuals that got any dose ($T = 1$). Other variations of these are possible where we reduce the analysis to a binary setting. The potential of Causalteshap lies in guiding treatment guidelines to enhance treatment effectiveness and reduce potential harm. Additionally, it can be used in treatment research, such as clinical studies, to select more specific patients or analyze them differently.

With the help of Powershap, the antimicrobial models proposed in Chapters 3 and 4 are novel in their field, where traditionally only PopPK models are used to estimate the antimicrobial concentration in a patient. These models can help combat antimicrobial resistance and treat patients more effectively and individually [7, 8]. The combined antimicrobial model of Chapter 4 opens the path for creating a unified antimicrobial model that can predict the concentration and provide dosing advice for a whole class of antimicrobials or even more, given the data available to train these models. However, to truly quantify the potential impact of these models, they must also be evaluated in clinical studies. There can be multiple stages to this process. In a first study, the model can be evaluated in the back-end, and in real-time, it predicts the concentration and compares it with observed concentrations. If the first stage was successful, in the second stage, the physician is informed by providing the predicted concentration and the suggested dose, allowing the physician to make a decision. Afterwards, we can evaluate defined outcomes such as length of stay, mortality, or concentration attainment to quantify the clinical benefit. On top of this, the UQ in the presented models can be improved. Currently, the quantile ensemble assumes a normal distribution for the predictive distribution, which can be seen as stringent. To enhance this, we could use a portion of the data to create a calibration dataset and apply conformal prediction, such as conformal predictive systems (CPS), to provide predictive distributions [9]. We can expand this approach using weighted CPS to provide UQ even in cases of distribution shift, such as in different hospitals or patient populations [10]. This would address calibration issues in these datasets and offer a solution to the challenges encountered during the external validation of the piperacillin model in Chapter 3. The downside here is that we trade predictive performance for better calibration guarantees in the UQ as there is not much data.

Likewise, the AF prediction models are novel in their field because they rely only on electronic health record data, compared to other approaches that use ECG data. These models were also evaluated on three ICU datasets, demonstrating that recalibrating a model for a new setting is beneficial (and potentially necessary) when there is a significant dataset shift. From a causal perspective, this is straightforward because the patient populations might not fully overlap, and a prediction model optimizes correlations rather than causal relationships. Additionally, there may be many local biases, as treatment methods and guidelines can vary across hospitals. These biases might not be present in the data, or they might be present but ignored by the model as they might be non-informative for the prediction in that hospital. Nevertheless, the AF risk models can also be evaluated in a two-stage clinical study, similar to the antimicrobial models mentioned previously. This approach allows us to assess the clinical benefit of these models for ICU care. Similarly, we can improve the UQ method using conformal prediction, as it currently relies only on the models' probabilities. Practically, we can create a calibration set and use Venn-Abers predictors to calibrate the risk probabilities, ensuring guaranteed coverage [11].

To further enhance the possibilities, the four proposed metrics (DCE, ADCE, adjusted ECE, and ESCE) add to the evaluation possibilities for predictive distribution in regression (DCE and ADCE) or probability evaluation in classification (ECE and ESCE). While these metrics can be visually interpreted using calibration plots, having quantified values simplifies optimization. In the case of predictive distributions, there are some alternatives such as the Probability Integral Transform (PIT) [12] or the Continuous Ranked Probability Score (CRPS) [13], which both work on the cumulative predictive distribution. These methods provide different insights and are best used alongside calibration plots. In this context, DCE and ADCE serve as quantified representations of calibration plots in a single value. For classification, the ECE with a single bin size is already frequently used for probability calibration evaluation. However, optimizing ECE for a specific bin size will only guarantee it for that specific bin size. The adjusted ECE addresses this by averaging across all bins, improving overall calibration. Additionally, the ESCE provides further information for UQ evaluation.

Moreover, the propensity-weighted conformal prediction enhances the possible UQ solutions for a dosing advice model. However, if the model generalizes well in regions with limited to no overlap, the UQ may be underconfident due to limited data support in those regions. This is not ideal, but remains informative as it still conveys the missing data or limited overlap. Also, if the underlying propensity estimation is erroneous or misses confounders, this can harm the coverage guarantee which must be taken into account when using the method. Nevertheless, this approach still offers the possibility of calibrated prediction intervals or predictive distributions (using weighted CPS [10]) for dose-response models.

Looking ahead, a significant challenge also still lies in effectively using UQ in medical models. For instance, how should a physician respond if the model indicates a 79% risk of AF at a given moment? What dosing strategy should be employed if the predictive distribution is wide, such as a 44% chance of overdosing, 41% chance of therapeutic dosing, and 15% chance of underdosing? What if the prediction interval is infinite? Future research can focus on evaluation or creating strategies by combining concepts such as clinical utility, i.e. the likelihood that the intervention or the prediction improves the health outcome [14], with UQ, such as in the work of Cortes-Gomez et al. [15]. Simply adding UQ to a predictive model for a medical use case is, unfortunately, not sufficient for its adoption. We must also address other critical aspects, such as the needs and requirements of physicians and how we present the models to them. Also, if the clinical benefit is lacking, there is little incentive for adoption. Depending on the use case, it is essential to discuss the requirements of physicians at least before and during the model design process, as misalignment can hinder adoption [8, 16, 17]. Additionally, presenting these predictions, the model and the associated uncertainty is also vital. Not every interpretability or explainability method is suitable for presenting the model, and some may require user education [17]. This is a distinct field of

research that must be considered when building a model.

Finally, the ethical aspects must also be taken into account. Who should be responsible? What data can we use? How do we ensure correct interpretations of the model? How do we handle privacy concerns? Addressing these diverse components, in addition to the methods presented in this work, is essential for developing a trustworthy model that healthcare professionals can confidently adopt. Therefore, it is essential to also include physicians throughout the model development. Their early and continuous involvement provides the essential clinical grounding, ensuring the resulting AI not only meets technical specifications but also aligns with real-world needs, helping every aspect of trustworthy AI and creating meaningful clinical adoption.

In conclusion, building trustworthy models for healthcare goes beyond just proposing technical solutions that incorporate UQ, interpretability, or speed up processes. It is about ensuring these models are practical in real-world settings, ethically sound, and genuinely beneficial to healthcare professionals. As we strive to create the "personal chef" for medicine, we must keep these principles in mind. Every step forward, whether technological, like the contributions from this dissertation, or otherwise, brings us eventually closer to truly improving the quality of patient care.

"All we have to decide is what to do with the time that is given us."

–J.R.R. Tolkien, The Fellowship of the Ring

References

[1] Arezoo Bozorgmehr and Birgitta Weltermann. *Prediction of Chronic Stress and Protective Factors in Adults: Development of an Interpretable Prediction Model Based on XGBoost and SHAP Using National Cross-sectional DEGS1 Data.* JMIR AI, 2(1):e41868, May 2023.

[2] Bujar Raufi and Luca Longo. *Comparing ANOVA and PowerShap Feature Selection Methods via Shapley Additive Explanations of Models of Mental Workload Built with the Theta and Alpha EEG Band Ratios.* BioMedInformatics, 4(1):853–876, March 2024.

[3] Michael Weyns, Thibault Blyau, Bram Steenwinckel, Filip De Turck, Sofie Van Hoecke, and Femke Ongenae. *Explainable Knowledge Graph Embeddings for Industrial Process Monitoring & Control.* Information Fusion, page 103242, May 2025.

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. *On calibration of modern neural networks.* In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, page 1321–1330. JMLR.org, 2017.

[5] Mahshid Ebrahimi Warkiani and Mohammad Hossein Moattar. *A Comprehensive Survey on Recent Feature Selection Methods for Mixed Data: Challenges, Solutions and Future Directions.* Neurocomputing, 623:129372, March 2025.

[6] Hyung Park, Eva Petkova, Thaddeus Tarpey, and R Todd Ogden. *A Sparse Additive Model for Treatment Effect-Modifier Selection.* Biostatistics, 23(2):412–429, April 2022.

[7] T. De Corte, S. Van Hoecke, and J. De Waele. *Artificial Intelligence in Infection Management in the ICU.* In Jean-Louis Vincent, editor, Annual Update in Intensive Care and Emergency Medicine 2022, pages 369–381. Springer International Publishing, Cham, 2022.

[8] Thomas De Corte, Jarne Verhaeghe, Sofie Dhaese, Sarah Van Vooren, Jerina Boelens, Alain G. Verstraete, Veronique Stove, Femke Ongenae, Liesbet De Bus, Pieter Depuydt, Sofie Van Hoecke, and Jan J. De Waele. *Pathogen-Based Target Attainment of Optimized Continuous Infusion Dosing Regimens of Piperacillin-Tazobactam and Meropenem in Surgical ICU Patients: A Prospective Single Center Observational Study.* Annals of Intensive Care, 13(1):35, April 2023.

[9] Vladimir Vovk, Jieli Shen, Valery Manokhin, and Min-ge Xie. *Nonparametric Predictive Distributions Based on Conformal Prediction.* Machine Learning, 108(3):445–474, March 2019.

[10] Jef Jonkers, Glenn Van Wallendael, Luc Duchateau, and Sofie Van Hoecke. *Conformal Predictive Systems Under Covariate Shift*, April 2024.

[11] Vladimir Vovk, Ivan Petej, and Valentina Fedorova. *Large-scale probabilistic predictors with and without guarantees of validity*. In Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1, NIPS'15, page 892–900, Cambridge, MA, USA, 2015. MIT Press.

[12] F. N. David and N. L. Johnson. *The Probability Integral Transformation When Parameters Are Estimated from the Sample*. Biometrika, 35(1/2):182–190, 1948.

[13] James E. Matheson and Robert L. Winkler. *Scoring Rules for Continuous Probability Distributions*. Management Science, 22(10):1087–1096, June 1976.

[14] *Definition of Clinical Utility - NCI Dictionary of Genetics Terms - NCI*. https://www.cancer.gov/publications/dictionaries/genetics-dictionary/def/clinical-utility, July 2012.

[15] Santiago Cortes-Gomez, Carlos Miguel Patiño, Yewon Byun, Steven Wu, Eric Horvitz, and Bryan Wilder. *Utility-Directed Conformal Prediction: A Decision-Aware Framework for Actionable Uncertainty Quantification*. In The Thirteenth International Conference on Learning Representations, October 2024.

[16] Cyra-Yoonsun Kang and Joo Heung Yoon. *Current Challenges in Adopting Machine Learning to Critical Care and Emergency Medicine*. Clinical and Experimental Emergency Medicine, 10(2):132–137, May 2023.

[17] De Corte, Thomas. *Antimicrobial stewardship in the intensive care unit : evolving challenges and data science opportunities*. PhD thesis, Ghent University, 2024.