SCUOLA
NORMALE
SUPERIORE

Class of Science

Ph.D. in Data Science

35°cycle

Explainable AI methods and their interplay with privacy protection

Scientific Disciplinary Area **INF/01**

Candidate	Supervisors
Dr.ssa Francesca Naretto	Prof. ssa Fosca Giannotti Prof. ssa Anna Monreale

Academic year 2022-2023

Abstract

Recent years have seen the emergence of Machine Learning models, which are accurate but lack transparency in their decision-making processes. The field of Explainable Artificial Intelligence has emerged to address this issue, but many questions remain unanswered. This Ph.D. Thesis presents two key contributions: (i) a novel variant of a local rule-based explanation method that provides stable and actionable explanations, and (ii) an investigation into the relationship between Data Privacy and Explainable Artificial Intelligence, examining their synergies and tensions.

For (i), an improvement of a local explanation method is designed, using factual logic rules to explain black-box decisions and providing actionable counterfactual logic rules for suggesting changes in instances to achieve different outcomes. Explanations are generated from a decision tree that mimics the local behavior of the black-box model. The decision tree is obtained through a stability and fidelity-driven ensemble learning approach, where neighbor instances are synthetically generated using a genetic algorithm guided by the black-box behavior.

Regarding (ii), two perspectives on privacy are addressed: (a) how Explainable Artificial Intelligence can enhance individuals' privacy awareness and (b) how Explainable Artificial Intelligence can compromise privacy. A framework called Expert is developed to predict users' privacy risk and provide explanations, focusing on human mobility data. Additionally, a visualization module is incorporated to display mobility data explanations on a map. To assess privacy exposure, instead, a new membership attack for Machine Learning models is proposed, and a methodology called reveal is introduced to evaluate the privacy risks associated with local explainers based on surrogate models. The experimental analysis demonstrates that global explainers pose a more significant threat to individual privacy compared to local explainers.

These findings highlight the delicate balance between explainability and privacy in developing Artificial Intelligence systems.

Contents

Chapter 1

Introduction

In the modern world, people are generating an enormous amount of data through their use of social media, apps, websites, and other digital tools. During the last few years, the availability of smartphones and wearable devices, technologically highly sophisticated, allow us to track numberless statistics about users, starting from the movements during the day, the appointments and commitments, as well as the social interactions. Companies and organizations exploit this data to gain a better understanding of their customers and provide more personalized experiences. In particular, the abundance of data available allows for advanced analyses and research that were once impossible. Given this data-rich setting, in the past few years, there has been a significant increase in research related to Data Mining, Machine Learning, and Artificial Intelligence, with applications in a wide range of fields, including economics, marketing, and healthcare. We actually employ Artificial Intelligence systems every day of our lives for a multitude of tasks: from language translators to recommendation systems that can suggest movies, music, and contacts to marketing chatbots and automated financial investing or social media monitoring. However, the use of these new technologies gives rise to several concerns.

In fact, the increased reliance on data threatens the Privacy of people whose data are used for the purposes mentioned above. In particular, the public release of human data issues concerns about personal information being leaked or individuals being re-identified, even if the data has been de-identified. One reason for this is that meta-identifiers such as behavior or address can still be utilized to recognize specific individuals within a dataset.

When employing Machine Learning and Artificial Intelligence techniques, privacy issues are harder to perceive and identify, but privacy risks always exist. Indeed, these kinds of models are often trained on sensitive data, such as medical records, weblogs, or human mobility data, which can lead to the disclosure of personal information through the discovery of hidden patterns. Their learning and use lead to the attribute disclosure problem, where personal information is revealed, or even the identity disclosure problem, where the individual can be re-identified. To address these concerns, Europe introduced the General Data Protection Regulation (GDPR) [1], aimed at protecting individuals' privacy rights.

The scientific research in this area is focused on studying two main challenges: how to evaluate privacy risks and how to design effective privacy protection methods. Privacy risk evaluation techniques involve using Machine Learning and statistical models to assess the level of privacy risk to users in a given dataset. Privacy protection methods aim to minimize the risk of user re-identification by ensuring a minimum level of privacy protection.

In recent years, several methods combining privacy risk evaluation and privacy protection techniques have been proposed [2–4]. These methods first evaluate the user's risk and then apply privacy protection methods to minimize the risk. To assess the privacy risk, different techniques can be used to simulate various re-identification attacks, depending on the attacker's level of knowledge. Based on the results of the evaluation, the appropriate anonymization method can be applied to reduce the user's risk.

However, these techniques have some limitations. First, the algorithms employed may not be efficient or scalable, making them unsuitable for processing large datasets. Indeed, these algorithms require computing the privacy risk for all possible background knowledge that an attacker may possess, which is time-consuming [5]. Additionally, these techniques assume a static dataset as input and are not incremental, meaning that if a record changes, the computation of the privacy risk must be restarted from the beginning. In an era with millions of new records generated per hour, this approach is not ideal. Another limitation is that re-identification techniques are data-dependent, and the type of input data determines the type of theorized attack. This limitation requires developing new re-identification attacks for each new type of data. Finally, these techniques are developed from the companies' perspective and not the user's, as the attacks are tailored to an entire dataset and not to the evaluation of a single individual's privacy risk.

Recently, with the advent of Machine Learning, the Data Privacy scenario changed. When working in this setting, in fact, the sensitive data employed in the training of the models can be kept private, avoiding the release of them to the public. However, several attacks on the privacy of Machine Learning models have been published recently. In fact, these models, during the training phase, learn information from sensitive data, which can then be exploited by a malicious attacker to undermine the privacy of people within the training dataset. Examples of such privacy attacks are the Membership Inference Attack, with the goal of determining the membership of a person to the original training dataset, or the reconstruction attacks.

To address these challenges, several privacy-preserving techniques have been proposed in the literature. Differential privacy [6] is one of the most prominent approaches to privacy protection in the context of data analysis. It provides a rigorous mathematical definition of privacy, which ensures that the inclusion or exclusion of an individual's data in a dataset does not affect the outcome of the analysis significantly. Differential privacy can be applied either to the original dataset or to the Machine Learning model. In both cases, it achieves privacy protection by adding random noise to the output of the analysis, which ensures that an individual's data cannot be inferred from the output. This technique has been successfully applied to a variety of data analysis tasks, including Machine Learning [7, 8]. However, the addition of noise may also affect the accuracy of the analysis, which can be a trade-off between privacy and utility. Another approach to protecting privacy is homomorphic encryption, which enables the analysis of encrypted data without the need to decrypt it [9]. This technique ensures that the data remains secure while enabling the analysis to be performed. However, homomorphic encryption is still in its infancy, and the techniques are computationally intensive, which limits their practical use.

The usage of Machine Learning models endangers not only user privacy but also the transparency and ethical integrity of the decision-making process. These models, often referred to as "black-box models," are complex and opaque, making it difficult to comprehend the process behind their decisions. Black-box models, such as Deep Neural Networks, Ensemble classifiers, and Support Vector Machines, are frequently used by Artificial Intelligence systems due to their high accuracy, but they can also be trained on biased and unfair datasets, leading to unfair or incorrect decisions. The reliance on these opaque models poses a risk of violating ethical principles and creating decision systems lacking in transparency. Companies across various industries have begun integrating black-box models into their products and applications, which can result in safety and liability concerns. In safety-critical applications like medicine, finance, and automation, this is especially relevant. The GDPR introduced a set of clauses also for automated decision-making in response to these concerns, including the "right of explanation" for individuals to obtain "meaningful explanations of the logic involved" when automated decision-making is employed. However, the implementation of such a principle is challenging, and it requires technology capable of explaining the logic for the decisions of black-box models. Without such technology, the right to an explanation will remain a distant goal. Therefore, the need for an explanation technology is urgent and represents a significant scientific challenge. For these reasons, the field of Explainable Artificial Intelligence is nowadays one of the most studied. However, there are several challenges.

As a first concern, explanations can be provided at different levels of granularity and can target different stakeholders [10]. For example, explanations can be provided at the individual level to explain the prediction for a specific instance or at the group level to explain the behavior of the model for a specific subgroup of instances [11]. Explanations can also target different audiences, such as domain experts, non-expert users, or regulators [12].

One of the most interesting challenges is to develop generalizable explanation techniques that work across different types of models and domains, obtaining an agnostic method. Another challenge is to ensure that the explanations provided are meaningful and actionable for the intended audience [13]. Finally, there is a need for evaluation metrics and benchmarks to assess the quality of the explanations provided by different techniques [14–18].

The use of Machine Learning algorithms presents two main limitations: they might jeopardize privacy, and their complexity makes it challenging for humans to comprehend them, leading to difficulties in evaluating their effectiveness. Privacy issues and lack of transparency in the use of Machine Learning models are widely acknowledged in various regulations and documents, both in Europe and worldwide, indicating their significance in ensuring ethical Artificial Intelligence practices. These concerns are specifically highlighted in the Ethics Guidelines For Trustworthy AI[1], a crucial publication by the High-Level Expert Group on Artificial Intelligence. Although these two issues may initially seem distinct, they are actually interconnected [19–21]. Providing explanations to users can address the need for transparency but may compromise the privacy of other users whose information is revealed in the explanation. On the other side, techniques like differential privacy, used to protect user privacy, may affect the accuracy and transparency of the model, obtaining explanations that are not faithful to the original data [22].

This Ph.D. Thesis is funded by the XAI European Project ERC (Grant Id 834756).

1.1 Contribution and Organization of the Thesis

This Ph.D. thesis addresses two key research questions. The first question is whether it is possible to provide stable and actionable explanations that help users understanding how Artificial Intelligence systems work and why they make certain decisions. The second research question concerns the relationship between Privacy and Explainability. We explore how Explainable Artificial Intelligence methods can help understanding users' privacy risks by providing more transparency into the decision-making processes of Artificial Intelligence systems. However, we also examine the potential privacy exposure associated with the explanation methods, particularly with regard to surrogate-based explainers.

The thesis begins with Part I, which first provides an overview on the types of data and Machine Learning models used throughout the rest of the thesis (Chapter 14); then, it presents an overview of the literature related to Explainable Artificial Intelligence and Privacy in Chapter 2 and Chapter 3, respectively. Chapter 4 concludes this part by discussing the laws and rules that try to govern the Artificial Intelligence field. The content of Chapter 2 is mainly based on material that appeared in the following publication:

> Francesco Bodria, Fosca Giannotti, Riccardo Guidotti, Francesca Naretto, Dino Pedreschi and Salvatore Rinzivillo. *Benchmarking and Survey of Explanation Methods for Black Box Models*, Accepted to Data Mining & Knowlege Discovery, 2023.

Part II addresses the first research question, starting with a benchmark of existing explanation methods. Based on this evaluation, we propose a new, local, post-hoc explanation

[1] Link to the Ethics Guidelines for a Trustworthy AI

method that is more stable and actionable than existing approaches. This method generates rules and counterfactuals that can be used to improve the interpretability of Artificial Intelligence systems. This Part is composed of two chapters, the first one, Chapter 5, in which is presented the benchmark of explanation methods available in the literature, and the second one, Chapter 6, that defines a new explanation method. This Part is mainly based on the works that appeared in the following publications respectively:

> Francesco Bodria, Francesca Naretto, Fosca Giannotti, Dino Pedreschi. *Benchmark analysis of black-box local explanation methods.* XAI.it - Italian Workshop on Explainable Artificial Intelligence, pages 73-87, 2022.
>
> Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Francesca Naretto, Franco Turini, Dino Pedreschi and Fosca Giannotti. *Stable and actionable explanations of black-box models through factual and counterfactual rules.* Data Mining and Knowledge Discovery, 2022

Finally, in Part III, we explore the relationship between Privacy and Explainability in greater depth. We analyze how explanations can help users become more aware of their privacy risks, as well as the potential privacy risks associated with surrogate-based explainers. In particular, in the first part we tackle the possible synergies between Explainable Artificial Intelligence and Privacy, focusing on the task of privacy risk assessment. The literature in this setting evaluates the data privacy risk of people by simulating privacy attacks, exploiting some well established framework, such as PRUDEnce [2]. However, these methods have high computational complexity, provoking practical limitations in online user-centric applications that require an up-to-date privacy exposure indicator. For this reason, we propose and develop Expert, a framework which exploits Machine Learning models to predict the privacy risk. In addition, this framework also provides users with explanations. In this way, the explanations are exploited to increase the user understanding about their privacy risks. This contribution presented in Chapters 8 & 9 is based on the following publications:

> Francesca Naretto, Roberto Pellungrini, Anna Monreale, Franco Maria Nardini, Mirco Musolesi. *Predicting and Explaining Privacy Risk Exposure in Mobility Data*, Discovery Science - 23rd International Conference, DS 2020.
>
> Francesca Naretto and Roberto Pellungrini and Nardini, Franco Maria and Fosca Giannotti. *Prediction and Explanation of Privacy Risk on Mobility Data with Neural Networks*, ECML/PKDD 2020 Workshops, XKDD2020.
>
> Francesca Naretto, Roberto Pellungrini, Daniele Fadda, Salvo Rinzivillo. *EXPHLOT: EXplainable Privacy assessment for Human LOcation Trajectories*, Submitted to ECML/PKDD 2023.

After the presentation of possible synergies, in Part IV we analyze the possible tensions about Privacy and Explainable Artificial Intelligence. We begin this part with Chapter 11 which present the definition of a novel membership attack, namely Aloa, in which the main objective is to infer the membership of some records with minimum amount of attacker's information. Following, in Chapter 12 we propose a privacy risk assessment framework for local and global explainers and we demonstrate that they may lead to higher privacy exposure compared to black-box models. The works on which this Part is based are the following:

> Simone Rizzo, Francesca Naretto, Anna Monreale. *Agnostic Label-Only Membership Inference Attack.* Submitted to 17th International Conference on Network and System Security, 2023.
>
> Francesca Naretto, Anna Monreale, Fosca Giannotti. *Evaluating the Privacy Exposure of Interpretable Global Explainers*, The Fourth IEEE International Conference on Cognitive Machine Intelligence (CogMi), 2022.
>
> Francesca Naretto, Anna Monreale, Fosca Giannotti. *Evaluating the Privacy Exposure of Interpretable Global and Local Explainers.* Submitted to

Our findings sheds light on the delicate balance that must be struck between explainability and privacy in the development of Artificial Intelligence systems. Through this research, we hope to contribute to developing more secure, transparent, and ethical AI systems that can benefit society while respecting individual Privacy and rights.

Part I

Setting the stage

The extensive use of Machine Learning algorithms across a range of domains, including healthcare, finance, and social media, has given rise to concerns about the potential privacy violations faced by individuals, as well as the possibility of bias or discrimination in the outputs generated by Artificial Intelligence models. Additionally, ensuring the explainability of Artificial Intelligence has become crucial in building trust and accountability in these systems. In fact, users and stakeholders are often skeptical about the usage of these systems due to their opaqueness. The significance of this issue is also due to the current legislation, such as the General Data Protection Regulation, which governs the use of personal data, and the Artificial Intelligence Act, which was recently published and regulates the use of Artificial Intelligence systems. The goal of this Ph.D. thesis is to delve into the theoretical foundations, algorithms, and practical applications of both Data Privacy and Explainable AI, and to propose innovative solutions to tackle the challenges and trade-offs between these two concepts.

The first Part of this Ph.D. Thesis provides an overview of the fundamental concepts and algorithms necessary for understanding the research presented in this work, focusing on the two core research areas of Privacy and Explainable Artificial Intelligence. The Part begins by outlining the background and motivation for the research conducted, which includes a comprehensive review of the relevant literature.

The related literature about Explainable Artificial Intelligence is described in details in Chapter 2, while the state of the art in the context of Privacy is presented in Chapter 3. Regarding the Explainable Artificial Intelligence topic, the Chapter starts with a presentation of the taxonomy of the field, presenting the core concepts, such as the difference between local and global explanations, and post-hoc and intrinsic methods. In Section 2.2, the state-of-the-art in the field of explanation methods for tabular data is presented, organized depending on the kind of output of the explanation methods: feature importance, rule, counterfactual and example. In particular, we present LIME, LORE and SHAP, three local post-hoc tabular explanation methods exploited in this Thesis, as well as TREPAN, a global post-hoc explanation method based on trees. Following, in Section 2.3, it is reported the state-of-the-art explanation methods for sequential data, such as Attention methods and Shaplets-based methods. The Chapter concludes with a description of the evaluation measures for validating the goodness of explanations, in Section 2.4. In this last part, we present two different kind of evaluation metrics: *qualitative*, based on the user experience, and *quantitative*, based on the mathematical validation of the output of the explanation methods.

Following the topic of Explainable Artificial Intelligence, we present the other core topic of this Thesis, which is Privacy, in Chapter 3. This Chapter begins by describing the various algorithms and methods used in the field of Data Privacy, including a detailed explanation of their underlying principles, which are the basis for the majority of the works presented in the rest of the Thesis. The topic of Privacy is composed of two main research fields: the *Privacy Risk Assessment*, in which the objective is to evaluate the privacy risk exposure of the data under analysis, and the *Privacy Protection*, in which protection mechanisms are proposed to shield the Privacy of the users. The Privacy Risk Assessment is a core topic in this Thesis and is presented in Section 3.2: firstly we describe the privacy risk assessment methodologies available in the state-of-the-art for assessing the Privacy on a given dataset. We then deal with the assessment of privacy exposure in the context of Machine Learning models, which is a recent threat to the sensitive data of the users which are part of the training data of the model. Lastly, we present the Privacy Protection mechanisms, such as K-anonymity and Differential Privacy. We conclude this Chapter with a description of the concept of *Privacy by design* and *User Centric Privacy Protection Methods*, which are inspiring concepts for the problems tackled in this Thesis.

We conclude this Part by presenting the existing regulations for Explainable Artificial Intelligence and Privacy in Chapter 4, focusing on the General Data Protection Regulation and the recently published Artificial Intelligence Act.

By the end of this Chapter, the reader will have a comprehensive understanding of the theoretical and practical aspects of the research in the field of Data Privacy and Explainable AI, which will serve as a foundation for the subsequent chapters that delve into the details of the studies. In reference to the Data and Machine Learning models utilized in the subsequent Sections of this Thesis, their detailed presentation is omitted from this Section due to its already extensive length. It is important to acknowledge that these data and machine learning models are widely accepted and established within the current state of the art. This is not the case for the literature related to Explainable AI and Privacy, where we focus on recent research and the emerging challenges. For this reason, we prefer to present in this Part these concepts in details, while a more comprehensive understanding of the formalization of the employed data and the specific types of machine learning models employed, we refer interested readers to consult the Appendix 14.

Chapter 2

Explainable Artificial Intelligence

Today Artificial Intelligence is one of the most important scientific and technological areas, with a tremendous socio-economic impact and a pervasive adoption in many fields of modern society. The impressive performance of Artificial Intelligence (AI) systems in prediction, recommendation, and decision making support is generally reached by adopting complex Machine Learning (ML) models that "hide" the logic of their internal processes. As a consequence, such models are often referred to as "black-box models" [23–25]. Examples of black-box models used within current AI systems include deep learning models (neural networks with several layers) and ensemble models, such as Random Forest (RF) and GCForest (GcForest), presented in Section 14. The high performance of such models in terms of accuracy has fostered their adoption, even if the opaqueness of black-box models may hide potential issues inherited by training on biased or unfair data [26]. Thus there is a substantial risk that relying on opaque models may lead to adopting decisions that we do not fully understand or, even worse, violate ethical principles. Companies are increasingly embedding ML models in their AI applications, incurring a potential loss of safety and trust [27]. These risks are particularly relevant in high-stakes decision making scenarios, such as medicine, finance and autonomous systems. In 2018, the European Parliament introduced in the GDPR, a set of clauses for automated decision-making in terms of *a right of explanation* for all individuals to obtain "meaningful explanations of the logic involved" when automated decision making takes place. Also, in 2019, the High-Level Expert Group on AI presented the ethics guidelines for trustworthy AI. Despite divergent opinions among legals regarding these clauses [28–30], everybody agrees that the need for the implementation of such a principle is urgent and that it is a huge open scientific challenge. An in-depth analysis on the regulations for AI systems and Data Privacy is proposed below, in Section 4.

As a reaction to these practical and theoretical ethical issues, in the last years, we have witnessed the rise of a plethora of explanation methods for black-box models [14, 16, 23] both from academia and from industries. Thus, eXplainable Artificial Intelligence (XAI) [31] emerged as investigating methods to complement AI, to make accessible and interpretable the internal logic and the outcome of the model, making such process human understandable. Due to the increasing interest into this subject, the landscape of XAI is nowadays enormous. For this reason, in Section 2.1 we first present the taxonomy we refer to for this Thesis. Following, we describe the state-of-the-art methods for explaining tabular data, in Section 2.2, as well as the methods for sequence data, in Section 2.3. All the material presented in this Section is part of a survey on XAI published in [11]. Given the speed at which new algorithms are being published in this area, the same taxonomy presented below and categorization adopted in our work are exploited in a *XAI Live Survey*[1] where the existing methods are further analyzed and continuously updated with emergent approaches.

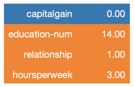
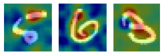
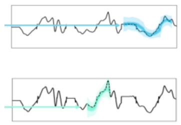
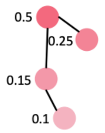
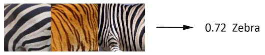
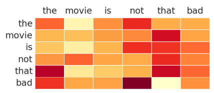
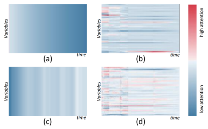
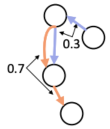
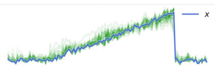
In the following, we first present the taxonomy of XAI, in Section 2.1, describing the main structure of the methods available in the field. Then, we move to explanation methods for tabular data, presented in Section 2.2. The description of these methods is organized depending on the kind of output explanation considered, namely feature importance, rules, counterfactuals and examples. We then present the methods for sequence data, in Section 2.3, also in this case they are presented based on the kind of output explanations. Lastly, we conclude the Chapter by presenting the actual state of the art in the context of evaluation measures for XAI methods, in Section 2.4.

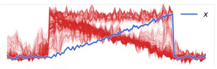
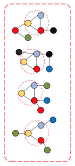
2.1 Explainable Artificial Intelligence taxonomy

In this Section we present a novel taxonomy of XAI methods based on the type of explanation returned. The categorization presented here refers to the Survey [11], written during my Ph.D. In this work, we highlighted the need for an updated systematic categorization of explanation methods, based on the type of explanation returned. Our approach aims at proposing a categorization from the point of view of the users. In fact, a user which requires an explanation first know which data he/she is dealing with, then the type of explanation they can have for that data type, and finally, he/she can select the best XAI method that can be used to obtain such explanation among the available one, comparing the properties offered by the method as well as a first general evaluation. Hence, our fist division regards the kind of input data used and the kind of explanation the user is looking for. A general overview of the landscape of XAI methods is presented in Table 2.1: in this table are present the most popular kinds of data and explanation kinds. A reader should use Table 2.1 as follows. First, he/she should identify through the column header the data type of her problem setting. After that, each row offers an alternative type of explanation with an example. For instance, if we are interested in images, we should look to the

[1] `https://kdd-lab.github.io/XAISurvey/`

Figure 2.1: Explanation-based taxonomy with examples divided for different data type.

<table>
<tr><th>TABULAR</th><th>IMAGE</th><th>TEXT</th><th>TIME SERIES</th><th>GRAPHS</th></tr>
<tr>
<td><b>Feature Importance (FI)</b>
A vector containing a value for each feature. Each value indicates the importance of the feature for the classification.

capitalgain 0.00
education-num 14.00
relationship 1.00
hoursperweek 3.00</td>
<td><b>Saliency Maps (SM)</b>
A map which highlight the contribution of each pixel at the prediction.
</td>
<td><b>Sentence Highlighting (SH)</b>
A map which highlight the contribution of each word at the prediction.
the movie is not that bad</td>
<td><b>Series Highlighting</b>
A score for every point in the series which highlight the contribution to the prediction.
</td>
<td><b>Node Highlighting</b>
A score for every node in the graph which highlight the contribution of that node to the prediction.

 </td>
</tr>
<tr>
<td><b>Rule-Based (RB)</b>
A set of premises that the record must satisfy in order to meet the rule's consequence.
$r = Education \leq College \rightarrow \leq 50k$</td>
<td><b>Concept Attribution (CA)</b>
Compute attribution to a target "concept" given by the user. For example, how sensitive is the output (a prediction of zebra) to a concept (the presence of stripes)?

 </td>
<td><b>Attention Based (AB)</b>
This type of explanation gives a matrix of scores which reveal how the word in the sentence are related to each other.

 </td>
<td><b>Attention Based</b>
This type of explanation gives a matrix of scores which reveal how the points in the series are related to each other.

 </td>
<td><b>Edge Highlighting</b>
A score for every edge in the graph which highlight the contribution of edge to the prediction.

 </td>
</tr>
<tr>
<td colspan="4"><b>Prototypes (PR)</b>
The user is provided with a series of examples that characterize a class of the black box
$p = Age \in [35, 60], Education \in [College, Master] \rightarrow$ "$\geq 50k$"
$p =$ $\rightarrow$ "cat"
$p =$ "... not bad ..." $\rightarrow$ "positive"


<b>Counterfactuals (CF)</b>
The user is provided with a series of examples similar to the input query but with different class prediction
$q = Education \leq College \rightarrow$ "$\leq 50k$"
$c = Education \geq Master \rightarrow$ "$\geq 50k$"
$q =$ $\rightarrow$ "3"
$c =$ $\rightarrow$ "8"
$q =$ The movie is not bad $\rightarrow$ "positive"
$c =$ The movie is that bad $\rightarrow$ "negative"
</td>
<td><b>Graph Prototypes</b>
Identifying which part of the graph has influenced the prediction
</td>
</tr>
</table>

second column. Here we can find saliency maps and concept attribution as image-specific explanation types. The last rows reports visualizations and examples of prototypes and counterfactuals, i.e., instance-based explanations, which are available independently from the data type analyzed. Finally, once the reader has selected the desired/most-suitable explanation type on Table 2.1, she can find in the corresponding section an overview of the most well-known and used explanation methods able to return that kind of explanation. For the sake of completeness, other types of data, increasingly present in literature, such as graphs [32] and time series [33], have been included in a separate section.

To summarize the taxonomy of the XAI methods, we propose a small diagram, in Figure 2.2. In the following, we summarize the fundamental distinctions adopted to annotate such approaches presented in Figure 2.2:

- *INtrinsically (IN)* explainable methods are explainable by design methods that returns a decision, and the reasons for the decision are directly accessible because the model is transparent.
- *Post-Hoc (PH)* explanation methods are black-box explanation are that provides explanations for a black-box model that takes decisions.
- *Global (G)* explanation methods aim at explaining the overall logic of a black-box model. Therefore the explanation returned is a global, complete explanation valid for any instance;
- *Local (L)* explainers aim at explaining the reasons for the decision of a black-box model for a specific instance.
- *Model-Agnostic (A)* explanation methods can be used to interpret *any type* of black-box model;
- *Model-Specific (S)* explanation methods can be used to interpret *only a specific type* of black-box model.

In addition to the taxonomy just presented, it is worth mentioning that in the context of XAI there are different terms and definitions used differently according to different areas of the field. In the following we clarify the most popular terms in this context with the definition we refer to in the remaining of this Thesis:

- *Explanation* [16, 23] is an *interface* between humans and an AI decision-maker that is both comprehensible to humans and an accurate proxy of the AI. Consequently, explainability is the ability to provide a *valid* explanation.
- *Interpretability* [23], or comprehensibility [34], is the ability of stakeholders to understand relevant aspects of the modeling process. Interpretability and comprehensibility are tied to the evaluation of the model complexity.

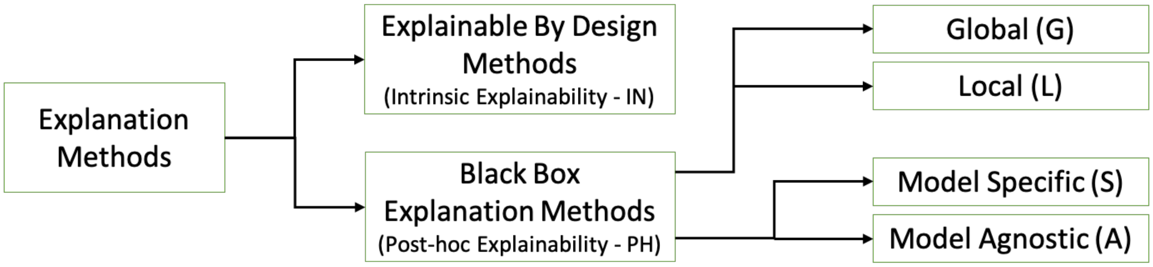

Explanation Methods
Explainable By Design Methods
(Intrinsic Explainability - IN)
Black Box Explanation Methods
(Post-hoc Explainability - PH)
Global (G)
Local (L)
Model Specific (S)
Model Agnostic (A)

Figure 2.2: Existing taxonomy for the classification of explanation methods.

- *Transparency* [16], or equivalently understandability or intelligibility, is the capacity of a model of being interpretable itself. Thus, the model allows humans to direct understand its internal mechanism and its decision process.

- *Complexity* [10] is the degree of effort required by a user to comprehend an explanation. The complexity can consider the user background or eventual time limitation necessary for the understanding.

2.2 XAI for Tabular data

In this section, we present a selection of approaches for explaining decision systems acting on tabular data. In particular, we present the following types of explanations based on: *Features Importance* (FI, Section 2.2.1), *Rule* (RB, Section 2.2.2), *Prototype* (PR) and *Counterfactual* (CF) (Section 2.2.5). Table 2.1 summarizes and categorizes the explainers. The methods are sorted by the explanation type they produce. For every explanation method is provided the author name, the year of publication, and the data type it can handle. In addition, Table 2.1 specifies if the method is intrinsic (IN) or Post-hoc (PH), if it provides Global explanations (G) or Local one(L), and if it is an Agnostic method(A) or a model Specific one (S). Methods with code available are highlighted in blue.

2.2.1 Feature Importance for Tabular data

Feature importance is one of the most popular type of explanation returned by local explanation methods. For feature importance-based explanation methods, the explainer assigns to each feature an importance value which represents how much that particular feature was important for the prediction under analysis. Formally, given a record x, an explainer $f(\cdot)$ models a feature importance explanation as a vector $e = \{e_1, e_2, \dots, e_m\}$, in which the value $e_i \in e$ is the importance of the i^{th} feature for the decision made by the black-box model $b(x)$. For understanding the contribution of each feature, the sign and the magnitude of each value

e_i are considered. W.r.t. the sign, if $e_i < 0$, it means that feature contributes negatively for the outcome y; otherwise, if $e_i > 0$, the feature contributes positively. The magnitude, instead, represents how great the contribution of the feature is to the final prediction y. In particular, the greater the value of $|e_i|$, the greater its contribution. Hence, when $e_i = 0$ it means that the i^{th} feature is showing no contribution for the output decision. An example of a feature based explanation is $e = \{age = 0.8, income = 0.0, education = -0.2\}, y = deny$. In this case, *age* is the most important feature for the decision *deny*, *income* is not affecting the outcome and *education* has a small negative contribution.

LIME Local Interpretable Model-agnostic Explanations [35], is a local model-agnostic explainer which returns explanations as features importance vectors. The main idea of Lime is that the explanation may be derived locally from records generated randomly in the neighborhood of the instance that has to be explained. The key factor is that it samples instances both in the vicinity of x (which have a high weight) and far away from x (low weight), exploiting π_x, a proximity measure able to capture the locality. We denote b the black-box and x the instance we want to explain. To learn the local behavior of b, Lime draws samples weighted by π_x. It samples these instances around x by drawing nonzero elements of x uniformly at random. This gives to Lime a perturbed sample of instances $\{z \in \mathbb{R}^d\}$ to fed to the model b and obtain $b(z)$. They are then used to train the explanation model $g(\cdot)$: a sparse linear model on the perturbed samples. The local feature importance explanation consists of the weights of the linear model. A number of papers focus on overcoming the limitations of Lime, providing several variants of it. Lime [36] is a deterministic version in which the neighbors are selected from the training data by an agglomerative hierarchical clustering. iLime [37] randomly generates the synthetic neighborhood using weighted instances. aLime [38] runs the random data generation only once at "training time". kl-lime [39] adopts a Kullback-Leibler divergence to explain Bayesian predictive models. qlime [40] also consider nonlinear relationships using a quadratic approximation. A part from tabular data, Lime can be used also on other data types. In Figure 2.3 (upper part) are reported examples of Lime[2] explanations relative to our experimentation on Adult (a/b) and German (c/d) [3]. We predicted the same record using LG and CAT, and then we explained it. Interestingly, for Adult (plots a/b), Lime considers a similar set of features as important (even if with different values of importance) for the two models: on 6 features, only one differs. A different scenario is obtained applying Lime on German (plots c/d): different features are considered important by the two classifiers. However, the confidence of the prediction between the two classifiers is quite different: both of them predict the output correctly, but CAT has a higher value, suggesting that this could be the cause of differences between the two explanations.

SHAP SHapley Additive exPlanations [41], is a local model-agnostic explanation method

[2] We refer to the original version of Lime

[3] For reproducibility reasons, we fixed the random seed.

computing features importance by means of Shapley values[4], a concept from cooperative game theory. The explanations returned by Shap are *additive feature attributions* and respect the following definition: $g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i$, where $z' \approx x$ as a real number, $z' \in [0, 1]$, $\phi_i \in \mathbb{R}$ are effects assigned to each feature, while M is the number of simplified input features. Shap retains three properties: *(i) local accuracy*, meaning that $g(x)$ matches $b(x)$; *(ii) missingness*, which allows for features $x_i = 0$ to have no attributed impact on the Shap values; *(iii) stability*, meaning that if a model changes so that the marginal contribution of a feature value increases (or stays the same), the Shap value also increases (or stays the same). The construction of the Shap values allows to employ them both *locally*, in which each observation gets its own set of Shap values; and *globally*, by exploiting collective Shap values. We highlight that Shap can be realized through different explanation models that differ in how they approximate the computation of the Shap values. In particular, there are five strategies: *KernelExplainer* is the model-agnostic one, while *LinearExplainer*, *TreeExplainer*, *GradientExplainer*, and *DeepExplainer* are model-specific. Besides, similalry to Lime, Shap can be used on other data types. We applied *LinearExplainer* to LG, *TreeExplainer* to XGB, and *KernelExplainer* to CAT. In Figure 2.3 (lower part) we report the application of Shap on Adult through the *force plot* showing each feature contributes to push the output value away from the base value, which is an average among the training dataset's output values. The red features are pushing the output value higher, while the ones in blue are pushing it lower. For each feature is reported the actual value for the record under analysis. Only the features with the highest Shap values are shown in this plot. In the first force plot, the features that are pushing the value higher are contributing more to the output value: from a base value of 0.18 it is reached an actual output value of 0.79. In the force plot on the right, the output value is 0.0, and *Age*, *Relationship* and *Hours Per Week* are contributing to pushing it lower. Figure 2.4 (left and center) depicts the Shap values through a *decision plots*: the contribution of all the features are reported in decreasing order of importance. The line represents the feature importance for the record under analysis and it starts at its actual output value. In the first plot, predicted as $> 50k$, *Occupation* is the most important feature, followed by *Age* and *Relationship*. For the second plot, instead, *Age*, *Relationship* and *Hours Per Week* are the most ones. Besides the local explanations, Shap also offers a global interpretation of the model-driven by the local interpretations. Figure 2.4 (right) reports a global decision plot that represents the feature importance of 30 records of Adult.

DALEX [42] is a local and global, post-hoc, model-agnostic explanation method. Dalex reveals the features importance through an implementation of a *variable attribution* approach [43] that consists of a decomposition of the model's predictions in which each decomposition can be seen as a local gradient and it is used to identify the contribution of

[4] We refer the interested reader to: `https://christophm.github.io/interpretable-ml-book/shapley.html`

each attribute. Dalex also allows the calculus of Shap values. In Figure 2.5 are reported local explanations returned by Dalex for XGB on Adult. On the left are reported two explanations for a record classified as class $> 50k$, while on the right for classified as $< 50k$.

MAPLE [44] is a local post-hoc model-agnostic explainer that can also be used as a transparent model due to its internal structure. It combines random forests with feature selection methods to return feature importance based explanations. In particular, Maple is based on *SILO*, employed for obtaining a local training distribution based on the random forest leaves, and on *DStump* used to rank the features by importance. Maple considers the best k features from *DStump* to solve a weighted linear regression problem. Similarly to Lime, it returns these coefficients as features explanation.

CIU Contextual Importance and Utility [45], is a local, post-hoc, model-agnostic explainer based on the idea that a feature that might be important in a context may be irrelevant in another one. ciu explains the model's outcome based on the *contextual importance (CI)* approximating the overall importance of a feature in the current context, and on the *contextual utility (CU)* estimating how good the current feature values are for a certain output class. CI and CU are calculated through Monte Carlo simulations. We highlight that ciu does not require creating a surrogate model to employ for deriving the explanations. **EBM**, Explainable Boosting Machine [46] is an intrinsic local and global model specific method. EBM is a variant of a Generalized Additive Model (gam) [47], i.e., a generalized linear model that incorporates nonlinear forms of the predictors. For each feature, EBM uses a boosting procedure to train the generalized linear model: it cycles over the features, in a round-robin fashion, to train one feature function at a time and mitigate the effects of co-linearity. In this way, the model learns the best set of feature functions, which can be exploited to understand how each feature contributes to the final prediction. **NAM**, Neural Additive Models [48] is a local model-specific explainer defined as a variant of gam but tailored for neural networks. nam aims at combining the performance of deep neural networks, with the inherent intelligibility of gam. As a result, nam is able to learn graphs that describe how the prediction is computed by training multiple deep neural networks in an additive fashion such that each neural network attends to a single input feature. **CoFrNets** Continued Fractions Nets [49] is similar to nam but instead of approximating activations using neural networks, it uses continued functions. The output of a neuron is calculated as a continuous fraction of the previous one until it gets to the input layer. The propriety of continued fractions to represent every possible real number allows CoFrNets to express any possible function as in Neural Networks. On the other hand, since it is a simple fraction calculation, it is possible to compute the contribution of each input to the final output and produce feature importance explanations.

Features Importance-based Explainers Comparison. Feature importance based explanation methods provide an importance value for each feature of the record in the input. The importance of the features is computed in different ways, depending on the kind of

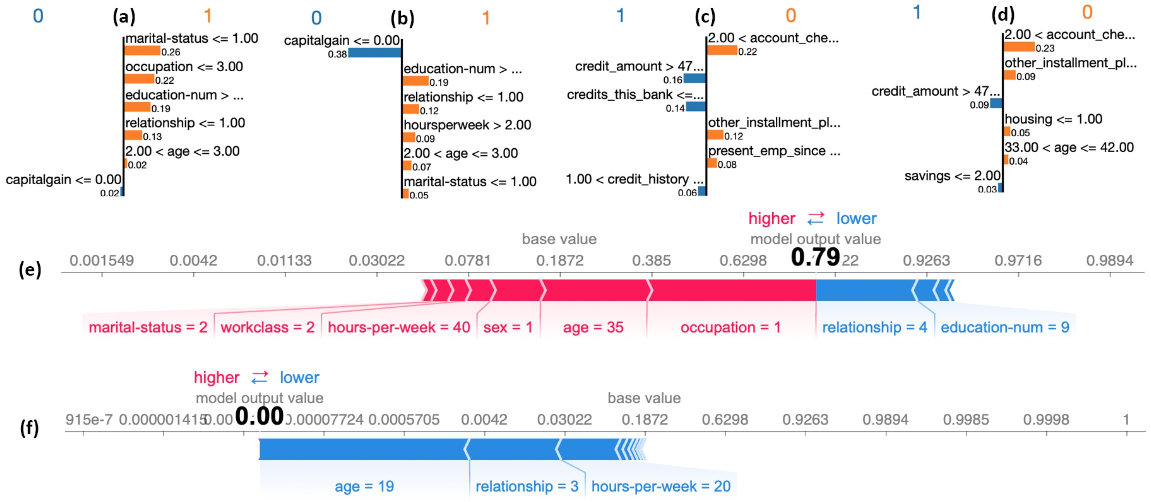


Figure 2.3: *TOP*: LIME application on the same record for ADULT (a/b), GERMAN (c/d): a/c are the LG model explanation and b/d the CAT model explanation. All the models correctly predicted the output class. *BOTTOM*: Force plot returned by SHAP explaining XGB on two records of ADULT: (e), labeled as class 1 ($> 50K$) and, (f), labeled as class 0 ($\leq 50K$). Only the features that contributed more (i.e. higher values) to the classification are reported.

explanation methods exploited. The majority of the explainers are post-hoc and local, even if there are examples of methods that also provide global explanations, allowing an in-depth analysis of the overall behavior of the machine learning model (SHAP, DALEX and CIU). Some explainers, such as LIME and all its variants, create a synthetic neighborhood set used to train a surrogate model, and extract the features' importance from it. These methods are suitable for context in which the explanation is online, since they are very efficient, as they use randomization techniques and surrogate models that are very simple and quick to train. As a weak point, the randomicity and the simplicity of the surrogate models may not best represent the data space under analysis. On the other hand, explainers that do not require the creation of a surrogate model but based on some mathematical procedure such as game theory for SHAP, the decomposition of predictions exploiting local gradients for DALEX, or Monte Carlo simulations for CIU might require a longer computational time w.r.t. LIME-like approaches. In addition, their internal workings may be difficult to understand, shifting the difficulty from understanding the machine learning model to understanding how the explanation method works. Among feature importance-based explainers, model-specific explainers, such as NAM and COFRNETS, are tailored for explaining neural networks and aim at approximating the activation functions. Overall, these explanation methods are fast, except for the model-agnostic variants of SHAP and DALEX, because they might require a

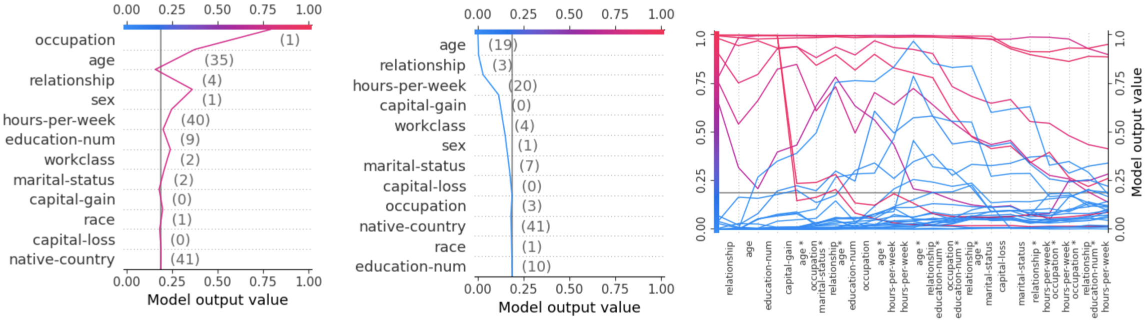


Figure 2.4: Shap on Adult: a record labelled $> 50K$ (top-left) and one as $\leq 50K$(top-right). They are obtained applying the TreeExplainer on a XGB model and then the *decision plot*, in which all the input features are shown. At the bottom, the application of Shap to explain the outcome of a set of records by XGB on Adult. The interaction values among the features are reported.

greater computing time due to their different approximations. Unfortunately, the output provided by these explainers is usually quite difficult to understand for non-experts since there are several variables and the plots provided are usually non-self-explanatory. As an example, we can think of Shap: in the plots in output, each feature importance is given by the output value, the base value, and, depending on the kind of explanator exploited, one or more arrays of feature importance.

2.2.2 Rule-based Explanation

Decision rules give the end-user an explanation about the reasons that lead to the final prediction. A decision rule r, also called *factual* or *logic* rule [50], has the form $p \rightarrow y$, in which p is a premise, composed of a Boolean condition on feature values, while y is the consequence of the rule. In particular, p is a conjunction of split conditions of the form $x_i \in [v_i^{(l)}, v_i^{(u)}]$, where x_i is a feature and $v_i^{(l)}, v_i^{(u)}$ are lower and upper bound values in the domain of x_i extended with $\pm\infty$. An instance x *satisfies* r, or r *covers* x, if every Boolean conditions of p evaluate to true for x. If the instance x to explain satisfies p, the rule $p \rightarrow y$ represents then a candidate explanation of the decision $g(x) = y$. Moreover, if the interpretable predictor mimics the behavior of the black-box in the neighborhood of x, we further conclude that the rule is a candidate local explanation of $b(x) = g(x) = y$. We highlight that, in the context of rules, we can also find the so-called *counterfactual rules* [50]. Counterfactual rules have the same structure of decision rules, with the only difference that the consequence of the rule $\overline{y}$ is different w.r.t. $b(x) = y$. They are important to explain to the end-user what should be changed to obtain a different output. An example

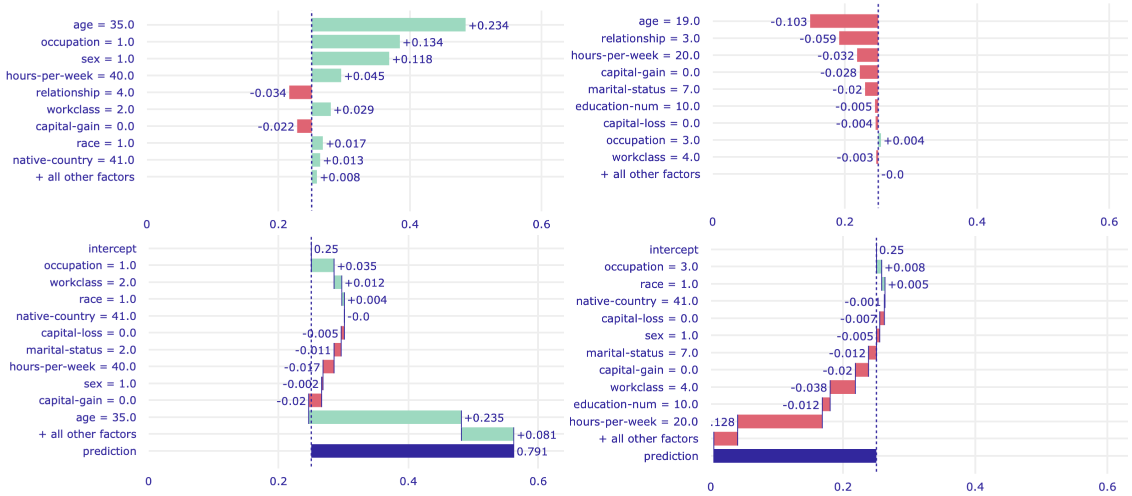


Figure 2.5: Explanations of Dalex for two records of Adult: $b(x) = 0$ (≤ 50) (left), $b(x) = 1$ ($> 50K$) (right) to explain an XGB in form of Shapley values (top), break down plots (bottom). The y-axis is the features important, the x-axis the positive/negative contribution.

of a rule explanation is $r = \{age < 40, income < 30k, education \leq Bachelor\}, y = deny$. In this case, the record $\{age = 18, income = 15k, education = Highschool\}$ satisfies the rule above. A possible counterfactual rule, instead can be: $r = \{income > 40k, education \geq Bachelor\}, y = allow$.

ANCHOR [51] is a global and local model-agnostic method that outputs rules, called *anchors*, as explanations. The idea is that, for decisions on which the anchor holds, changes in the rest of the instance's feature values do not change the outcome. Formally, given a record x, r is an anchor if $r(x) = b(x)$. To obtain the anchors, Anchor perturbs the instance x obtaining a set of synthetic records employed to extract anchors with precision above a user-defined threshold. Anchor exploits a multi-armed bandit algorithm [52] for the synthetic generation of the dataset, and rely on bottom-up approach and a beam search to find the anchors. Figure 2.6 reports some rules obtained by applying Anchor to explain XGB trained on Adult. The first rule has a high precision (0.96%) but a very low coverage (0.01%). It is interesting to note that the first rule contains *Relationship* and *Education Num*, which are the features highlighted by most of the explainers analyzed so far. In particular, in this case, for having a classification $> 50k$, the *Relationship* should be husband and the *Education Num* at least bachelor degree. *Education Num* can also be found in the second rule, in which case has to be less or equal to College, followed by the *Maritial Status*, which can be anything other than married with a civilian. This rule has an even better precision (0.97%) and suitable coverage (0.37%).

$x_1 =$ { *Education* = *Bachelors,*
Occupation = *Prof-specialty, Sex* = *Male,*
NativeCountry = *Vietnam, Age* = *35,*
Workclass = *3, HoursWeek* = *40,*
Race = *Asian-Pac-Islander,*
MaritialStatus =*Married-civ,*
Relationship = *Husband,*
CapitalGain = *0,*
CapitalLoss = *0*}, $> 50k$

r_{anchor} = { *EducationNum* > *Bachelors,*
Occupation $\leq$ *3.00,*
HoursWeek > *20,*
Relationship $\leq$ *1.00,*
34 < *Age* $\leq$ *41* } $\rightarrow\ > 50k$

r_{lore} = { *Education* > *5-6th, Race* > *0.86,*
WorkClass $\leq$ *3.41,*
CapitalGain $\leq$ *20000,*
CapitalLoss $\leq$ *1306* } $\rightarrow\ > 50k$

c_{lore} = {*CapitalLoss* $\geq$ *436* } $\rightarrow\ \leq 50k$

$x_2 =$ { *Education* = *College,*
Occupation = *Sales, Sex* = *Male,*
NativeCountry = *US, Age* = *19,*
Workclass = *2, HoursWeek* = *15,*
Race = *White,*
MaritialStatus = *Married-civ,*
Relationship = *Husband,*
CapitalGain = *2880,*
CapitalLoss = *0* }, $\leq 50k$

r_{anchor} = {*Education* $\leq$ *College,*
MaritialStatus > *1.00* }
$\rightarrow\ \leq 50k$

r_{lore} = {*Education* $\leq$ *Masters,*
Occupation > *-0.34,*
HoursWeek $\leq$ *40,*
WorkClass $\leq$ *3.50*
CapitalGain $\leq$ *10000,*
Age $\leq$ *34*} $\rightarrow\ \leq 50k$

c_{lore} = {*Education* > *Masters* } $\rightarrow\ > 50k$
{*CapitalGain* > *20000* } $\rightarrow\ > 50k$
{*Occupation* $\leq$ *-0.34* } $\rightarrow\ > 50k$

Figure 2.6: Two example of explanations of Anchor and Lore for Adult to explain an XGB model. x_i is the the original instance, r_{anchor} is the rule provided by Anchor, r_{lore} is the rule provided by Lore, and c_{lore} is the counterfactual rule provided by Lore for obtaining the other prediction.

LORE LOcal Rule-based Explainer [50], is a local model-agnostic explainer that provides explanations in the form of rules and counterfactual rules. Lore is tailored explicitly for tabular data. It exploits a genetic algorithm for creating the neighborhood of the record to explain. Such a neighborhood produces a more faithful and dense representation of the vicinity of x w.r.t. Lime. Given a black-box b and an instance x, with $b(x) = y$, Lore first generates a synthetic set Z of neighbors through a genetic algorithm. Then, it trains a decision tree classifier g on this set labeled with the black-box outcome $b(Z)$. From g, it retrieves an explanation that consists of *(i)* a *factual* decision rule, that corresponds to the path on the decision tree followed by the instance x to reach the decision y, and *(ii)* a set of counterfactual rules, which have a different classification w.r.t. y. This counterfactual rules show the conditions that can be varied on x in order to change the output decision. In Figure 2.6 we report the factual and counterfactual rules of Lore for the explanation of the same records showed for Anchor. It is interesting to note that, differently from Anchor and the others models proposed above, Lore explanations focuses more on the *Education Num*, *Occupation*, *Capital Gain* and *Capital Loss*, while the features about the relationship are not present.

RuleMatrix [53] is a model-agnostic explainer that provides both local and global explanations, specifically tailored for the visualization of the rules extracted. Given a training dataset and a black-box, rulematrix executes a rule induction step, in which a *rule list* is extracted by sampling the input data and their predicted label by the black-box. Then, the rules extracted are filtered based on thresholds of confidence and support. Finally, it outputs a visual representation of the rules.

Local Rule-based Explainers Comparison. The rule-based methods presented are all based on creating a surrogate model from which to extract the rules. In this category we find Anchor and rulematrix, which provide both local and global explanations by relying on simple rule extraction methods. The simplicity of these methods make them efficient even if, as in the case of Lime-like approaches, they may suffers in terms of goodness of explanations provided. Lore is the only explainer that provides only local explanations. Differently from the other methods, does not require to have access to the original training data, and, due its synthetic generation process, provides more faithful explanations. Thus, it may be good in settings where the black-box training dataset is unavailable, while rulematrix and Anchor, need to access the training data. Rule-based explanations are considered closer to human reasoning w.r.t. the feature importance-based explanations. In addition, they exploit easy to understand mechanisms, allowing users of different background to understand how the explanation method works, increasing the trust. However, these explainers usually require a greater running time w.r.t. the feature importance ones.

2.2.3 Global Tree-based Explainers

One of the most popular ways for generating explanation rules is by extracting them from a decision tree. In particular, due to the method's simplicity and interpretability, decision trees explain black-box models' overall behavior. Some explanation methods acting in this setting are model-specific explainers exploiting structural information of the black-box model under analysis.

TREPAN [54] is a model-specific global explainer tailored for neural networks. Given a neural network b, TREPAN generates a decision tree g that approximates the network by maximizing the gain ratio and the model fidelity. In particular, to leverage abstraction, TREPAN adopts n-of-m decision rules on which only n out of m conditions must be satisfied in order to fire the rule.

DecText is a global model-specific explainer tailored for neural networks [55]. DECTEXT resembles TREPAN with the difference that it considers four different splitting methods. It also considers a pruning strategy based on fidelity to reduce the final explanation tree's size. In this way, DECTEXT can maximize the fidelity while keeping the model simple. Both TREPAN and DECTEXT are presented as model-specific explainers but they can be practically employed to explain any black-box as they do not use any internal information of neural networks.

MSFT [56] is a global, post-hoc, model-specific explainer for random forests that returns a decision trees. MSFT is based on the observation that, even if random forests contain hundreds of different trees, they are quite similar, differing only for a few nodes. Hence, it adopts dissimilarity metrics to summarize the random forest trees using a clustering method. Then, for each cluster, an archetype is retrieved as an explanation.

CMM Combined Multiple Model procedure [57], is another global, post-hoc, model-specific explainer for tree ensembles. The key point of CMM is the data enrichment. Given an input dataset X, CMM first modifies it n times. On the n variants of the dataset, it learns a black-box. Random records are generated and labeled using a bagging strategy on the black-boxes. The authors were able to increase the size of the dataset to build the final decision tree.

STA Single Tree Approximation [58], is another global, post-hoc, model-specific explainer tailored for random forests. In STA the decision tree is constructed by exploiting test hypothesis on the trees in the forest to find the best splits. The explainers proposed are tailored for a specific machine learning model: TREPAN and DECTEXT explain neural networks, while CMM and STA are tailored for random forests and MSFT is for any ensemble method. Among them, some explainers exploit an enrichment of the data to improve the extraction of the tree (CMM, TREPAN, DECTEXT), while the others exploit the training dataset by applying some strategies based on dissimilarity metrics (MSFT) or test hypothesis (STA). Among the different methods, only DECTEXT and TREPAN apply some strategies with the goal of maximizing the model fidelity, even if they are tailored for small feedforward neural networks. The exploitation of trees to explain the global behavior of a more complex machine learning model have several benefits, such as a fast computation and a simple process to extract the explanations, based on transparent strategies. However, the trees extracted may be very deep, making the explanation model difficult to understand even in cases of simple datasets. Furthermore, the effectiveness of such explanations for very deep feed forward networks has not be judged yet.

2.2.4 Global Rule-based Explainers

In this section we present global explainers that do not extract decision trees as global interpretable model but lists or sets of rules. The majority of the methods described in the following extract rules by exploiting ensemble methods or rule-based classifiers. The explainers considered are all agnostic.

SkopeRules is a global, post-hoc, model-agnostic[5] explainer on the RULEFIT [59] idea to define an ensemble method and then extract the rules from it. SKOPE-RULES employs fast algorithms such as bagging or gradient boosting decision trees. After extracting all the possible rules, SKOPE-RULES removes rules redundant or too similar by a similarity threshold. Differently from RULEFIT, the scoring method does not solve the L1 regularization. Instead, the weights are given depending on the precision score of the rule.

Scalable-BRL [60] is an interpretable rule-based model that optimizes the posterior probability of a Bayesian hierarchical model over the rule lists. The theoretical part of this approach is based on [61].

GLocalX [62] is a global model-agnostic post-hoc explainer which adopts the *local to global* paradigm, i.e., to derive a global explanation by subsuming local logical rules. GLocalX start from an array of local explanation rules and following a hierarchical bottom up fashion merges those covering similar records and expressing the same conditions.

Global Rule-based Explainers Comparison. This small section comprises global explanation methods that extract rules in entirely different ways: either they exploits an ensemble method (SKOPE-RULES), a rule-based model (SCALABLE-BRL) or several local explanations (GLocalX). In terms of goodness of explanations, SKOPE-RULES and SCALABLE-BRL are tailored for an overall explanation of the machine learning model, focusing mostly on the data in input. GLocalX, instead, exploits local explanations and hence is able to tackle the problem from a different point of view, merging several local explanations. The output of these methods is a list of rules and even if there are techniques to filter out meaningless rules, the complexity of the explanation may be huge.

Rules-based Explainers Comparison. In this section, we presented a great variety of methods that provide logical rules as explanations exploiting different strategies. Independently from the strategy, due to the simplicity of the rules, they are often the preferred

[5] `https://skope-rules.readthedocs.io/en/latest/skope_rules.html`

explanation for non-expert people. The majority of the explainers presented in this section are based on the extraction of decision trees as surrogate models (LORE, TREPAN, CMM, STA, DECTEXT, MSFT), or ensemble methods based on decision trees, such as SKOPE-RULES. The remaining methods that do not rely on decision trees extract the rules in other ways, such as rule-based classifiers (again a surrogate model), as in the case of ANCHOR, SCALABLE-BRL and of RULEMATRIX. To further increase the comprehensibility of the explanation, some explainers correlate the explanations by graphical visualizations, such as RULEMATRIX, ANCHOR and SKOPE-RULES. Overall, the majority of the explainers require a long computing time due to the different enrichment of the data or the use of rule-based classifiers, which are among the longest interpretable models to train. Hence, they may be better fitted for offline explanations. Depending on the complexity of the ML model in input, the explanations may be complex, such as deep trees or long lists of rules.

2.2.5 Prototype-based Explanations

A prototype, also called archetype or artifact, is a record highlighting the characteristics which identify a group of objects belonging to the same class. Prototypes serve as examples, i.e., the user can understand the black-box reasoning by looking to records similar to the prototype. Thus, a prototype is a local explanation. A prototype can be *(i)* a record of the training set close to the input data x, *(ii)* a centroid of a cluster to which the input x belongs to, or *(iii)* even a synthetic record generated following an ad-hoc process. Depending on the explanation method considered, different definitions and requirements to find a prototype are can be considered.

MMD-CRITIC [63] is a *before the model* explanation method which produces prototypes and criticisms as explanations using *Maximum Mean Discrepancy (MMD)* measure. While prototypes explain the dataset's general behavior, criticisms represent records that are not well explained by the prototypes. MMDCRITIC selects prototypes by measuring the difference between the distribution of the instances and the instances in the whole dataset. The set of instances nearer to the data distribution are called prototypes, and the farthest are called criticisms.

ProtoDash [64] is a variant of MMDCRITIC. However, differently from MMDCRITIC, PROTODASH also returns non-negative weights which indicate the importance of each prototype.

PS Prototype Selection (PS) [65] seeks a set of prototypes that better represent the data under analysis by solving a set cover optimization problem with constraints on the properties the prototypes. After that, the prototypes are employed to learn a nearest neighbor rule classifier to be used as model.

TSP Tree Space Prototype [66], is a local post-hoc explainer tailored for explaining tree ensemble methods. The goal of TSP is to find prototypes for each class in the tree space of the tree ensemble b w.r.t. a given notion of proximity between trees.

Privacy-Preserving Explanations(PPE) [19] is a local post-hoc model-agnostic explainer that outputs prototypes and shallow trees as explanations while considering the concept of *privacy in explainability* while producing *privacy protected explanations.* The trade-off between privacy and comprehensibility is obtained through *micro aggregation* of the data, i.e., clustering. The clusters' centroids are used as prototypes for the finale explanation/prediction.

Protoype-based explanation comparison The strength of prototypes is the possibility of analyzing black-box behavior by comparison between the record under analysis and its analogues, which is a type of reasoning widely used by humans. Moreover, they allow data analysis before and after the black-box is applied. For this reason, in this section we may find explanation methods that are *before the model*, i.e. they explain the dataset without considering the black-box model, such as MMDCRITIC, PS and PROTODASH. On the other hand, local post-hoc explainers, such as TSP and PPE, provide prototypes based on the decisions of the black-boxes. Among the different methods proposed, one of the most promising ones id MMDCRITIC because outputs both prototypes and criticisms. In this setting we can also find a novel application, namely PPE, which produces privacy-protected prototypes, creating a link between two crucial ethical concepts: transparency and privacy. Indeed, using prototypes as explanations, although it may be useful for end-user understanding, may release sensitive information about the users in the training set, when the explanation method exploits the training dataset.

2.2.6 Counterfactual-based Explanations

Counterfactual explanations suggest what should be different in the input instance to change the outcome of the black-box model [67, 68], i.e., they describe a dependency on the attributes that led to a particular decision. Counterfactual explanations can be considered as prototypes' opposite. Thus, also counterfactuals are local explanations. In [69], counterfactual explanations are formalized as follows. Given a black-box model b that outputs the decision $y = b(x)$ for an instance x, a counterfactual explanation consists of an instance x' such that the decision for b on x' is different from y, i.e., $b(x') \neq y$, and such that the difference between x and x' is *minimal.* The different values between x and a counterfactual x' reveals what should have been different in x for having a different outcome. An ideal counterfactual is *minimal* because it should alter the values of the variables as little as possible to find the closest setting under which y is returned instead of $\neg y$. Concerning counterfactual explanations, there are many properties which are desired for this kind of explanations and for the explanation methods returning them. Examples are validity, minimality, similarity, plausibility, discriminative power, actionability, causality, diversity, efficiency, robustness, etc. [67, 70, 71]. To better understand the complex context and the many available possibilities, we refer the interested reader to [69, 72–74] while we briefly present only the most representative methods in this category.

WACH [67] is among the first paper to propose a counterfactual explainer, and probably is the most famous one. The loss function minimized by [67] is defined as $\lambda(b(x') - y')^2 + d(x, x')$ where the first term is the quadratic distance between the desired outcome y' and the classifier prediction on x', and the second term is the distance d between x and x'. λ balances the contribution of the first term against the second term. The distance function d adopted is a crucial characteristic in any counterfactual explainer. In [67] is adopted the Manhattan distance weighted with the inverse median absolute deviation of each feature.

CEM Contrastive Explanations Method [75], is a post-hoc and model-specific explainer tailored for neural networks. In particular, CEM can return *Pertinent Positives (PP)*, which can be seen as prototypes, and are the minimal and sufficient factors that have to be present to obtain the output y; and *Pertinent Negatives (PN)*, which are counterfactuals factors, that should be minimally and necessarily absent. Given x, CEM considers $x_1 = x + \delta$, where δ is a perturbation applied to x. CEM is formulated as an optimization problem over the perturbation variable δ.

C-CHVAE [76] is a local model-agnostic post-hoc explainer that accounts for *plausibility* when generating counterfactuals. Indeed, the loss function optimized controls that counterfactuals are not local outliers and that are close to correctly classified observations. Moreover, this method can generate counterfactuals without requiring a distance function for the input space at the cost of using a Variational AutoEncoder.

DICE Diverse Counterfactual Explanations [77] is a model-agnostic post-hoc explainer which solves an optimization problem with constraints to account for *plausibility* and *diversity* evaluated through distance functions. Plausibility avoiding the generation of unfeasible counterfactuals while diversity provides different ways of changing the outcome class.

FACE Feasible and Actionable Counterfactual Explanations [78] is a model-agnostic post-hoc explainer that focuses on returning actionable counterfactuals, i.e., records *coherent* with the input data distribution. In particular, FACE uncovers "feasible paths" for generating counterfactual, i.e, the shortest path distances defined via density-weighted metrics starting form the input instance. Finally, it uses a shortest path algorithm to find all the records that satisfy the requirements.

CFX [79] is a model-specific post-hoc explainer for Bayesian Network Classifiers. The explanations are built from relations of influence between variables, indicating the reasons for the classification. In particular, this method's main achievement is that it can find pivotal factors for the classification task that, if removed, would lead to a different classification.

Counterfactual-based Explanations comparison Counterfactual-based explanations are gaining attention during the past few years due to their ability to suggest what to do to achieve a different outcome w.r.t. the one predicted by the black-box. There are several characteristics to consider in a counterfactual, such as plausibility, which requires the explanations to be feasible, and actionability, so that the counterfactual can not suggest changing the values of unfeasible variables, such as age or sex. Satisfying these characteristics is of utmost importance because otherwise the counterfactuals obtained may not be applicable or understandable by the end user. For example, a counterfactual might require changing age or height, factors that cannot be changed, thus making the counterfactual unfeasible. Among the methods proposed, C-CHVAE deals with the plausibility of the counterfactuals proposed and FACE tackles both the plausibility and the actionability. The majority of the algorithms proposed solves an optimization problem based on a distance function and some perturbation of the original data (CEM, WACH, C-CHVAE) and only a few methods exploit different approaches, such as Variational Autoencoder, as C-CHVAE. Most of the proposed methods are local and post-hoc, with CFX and CEM specifically designed for certain models, while the others are agnostic. Among the different methods proposed, CEM is a promising solution since it provides both prototypes and counterfactuals, allowing for an in-depth analysis, such as MMDCritic and Lore.

2.3 XAI for Sequence data

Due to the tremendous amount of data generated by sensors over time, there is a widespread diffusion of ML models working on sequence data and in particular on time series [82]. A sequential data has been defined in Section 14.1.2. There are areas such as the medical or the financial field where temporal data is of particular importance and where black-box ML models are applied to provide support on decision-making for various tasks. For this reasons, recently we are assisting to emerging proposal for explainability related to time series [33]. The most important difference with the other types of data relies on the type of explanation produced.

2.3.1 Attention-based Explainers

Attention was first proposed for images in [83] to improve the model performance. The authors managed to show through an attention layer which parts of the image contributed most to realize the caption. Formally, the attention layer is indeed a layer to put on top of the model that, for each word, ij of the sentence x generates a positive weight α_{ij}, i.e., the *attention* weight. This value can be interpreted as the probability that a word ij is in the right place to focus on producing the next word in the caption. Attention mechanisms allow models to look over all the information the original sentence holds and learn the context [84, 85]. Therefore, it has caught the interest of XAI researchers who started using these weights as an explanation. The explanation e of the instance x is composed of the attention values (α), one for each feature x_i. Attention is nowadays a delicate argument, and while it is clear that it augments the performance of models, it is less clear if it helps gain interpretability and the relationship with model outputs [86].

Attention Based Heatmap [87] is a local, intrinsic, model-specific explainer, based on the attention mechanism. It produces a heatmap explanation similar to the one used

Type	Name	Ref.	Authors	Year	Data Type	IN/PH	G/L	A/S
FI	LRP	[80]	Bach et al.	2015	ANY	PH	L	A
	Lime	[35]	Ribeiro et al.	2016	ANY	PH	L	A
	Shap	[41]	Lundberg et al.	2017	ANY	PH	G/L	A
	Maple	[44]	Plumb et al.	2018	TAB	PH/IN	L	A
	EBM	[46]	Nori et al.	2019	TAB	IN	G/L	A
	NAM	[48]	Agarwal et al.	2020	TAB	IN	L	S
	CIU	[45]	Anjomshoae et al.	2020	TAB	PH	L	A
	DALEX	[42]	Biecek et al.	2020	ANY	PH	G/L	A
RB	TREPAN	[54]	Craven et al.	1996	TAB	PH	G	S
	MSFT	[56]	Chipman et al.	1998	TAB	PH	G	S
	CMM	[57]	Domingos et al.	1998	TAB	PH	G	S
	DECTEXT	[55]	Boz et al.	2002	TAB	PH	G	S
	STA	[58]	Zhou et al.	2016	TAB	PH	G	S
	SCALABLE-BRL	[60]	Yang et al.	2017	TAB	IN	G/L	A
	Lore	[50]	Guidotti et al.	2018	TAB	PH	L	A
	RULEMATRIX	[53]	Ming et al.	2018	TAB	PH	G/L	A
	Anchor	[51]	Ribeiro et al.	2018	ANY	PH	G/L	A
	GLOCALX	[81]	Setzu et al.	2019	TAB	PH	G/L	A
	SKOPERULE	[59]	Gardin et al.	2020	TAB	PH	G/L	A
PR	PS	[65]	Bien et al.	2011	TAB	IN	G/L	S
	MMDCritic	[63]	Kim et al.	2016	ANY	IN	G	S
	PROTODASH	[64]	Gurumoorthy et al.	2019	ANY	IN	G	A
	TSP	[66]	Tan et al.	2020	TAB	PH	L	S
CF	CEM	[75]	Dhurandhar et al.	2018	ANY	PH	L	S
	CFX	[79]	Albini et al.	2020	TAB	PH	L	S
	DICE	[77]	Mothilal et al.	2020	TAB	PH	L	A
	C-CHAVE	[76]	Pawelczyk et al.	2020	TAB	PH	L	A
	FACE	[78]	Poyiadzi et al.	2020	ANY	PH	L	A

Table 2.1: Summary of methods for explaining black-boxes for tabular data. The methods are sorted by explanation type: Features Importance (FI), Rule-Based (RB), Counterfactuals (CF), Prototypes (PR), and Decision Tree (DT). For every method, there is a data type on which it is possible to apply it: only on tabular (TAB) or any data (ANY). If it is an Intrinsic Model (IN) or a Post-Hoc one (PH), a local method (L) or a global one (G), and finally if it is model-agnostic (A) or model-specific (S).

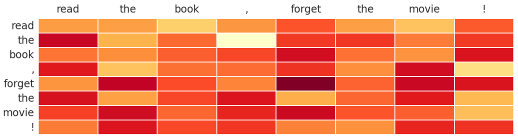


Figure 2.7: Attention based heatmap matrix generated from the method presented in [88]. The row and the columns of the matrix correspond to the words in the sentence “Read the book, forget the movie!”. Each matrix value shows the attention weight α_{ij} of the reannotation of the i-th word w.r.t. the j-th.

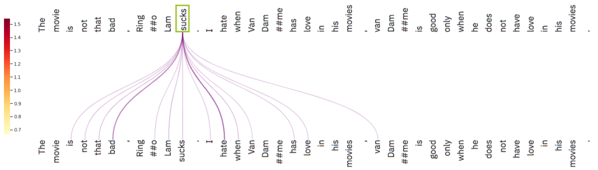


Figure 2.8: Attention based representation of BERT for a sentence taken from `imdb` using the visualization of [90]. The greater the attention between two words, the bigger the line. Here is selected only the attention related to the word “sucks”.

for SMs by using the weights of the black-box. It can only be applied to attention-based methods, such as BERT, in which the weights α_{ij} of the attention layers are used as score for every word in the sentence. The higher the score, the more important the word, therefore the redder the heatmap.

Attention Matrix [88] looks at the dependencies between words for producing explanations. It is a self-attention method, also called *intra-attention*, which relates different positions of a single sequence to compute its internal representation. The attention of a sentence x composed of N words can be understood as an $N \times N$ matrix, where each row and columns represent a word in the input sentence. The matrix values are the attention values of every possible combination of the tokens. This matrix is a representation of values pointing from each word to every other word [89] (Figure 2.7). We can also visualize this matrix with a focus on the connection between words [90] as in Figure 2.8, where the thickness of the lines is the self-attention value between two tokens.

Attention Based Explainers Comparison. Attention is a mechanism good for improving the model performance but not usable for explanation. As described in [86], it is unclear what relationship exists between attention weights and model outputs. Learned attention weights are frequently uncorrelated with gradient-based measures of feature importance, and it is possible to identify very different attention distributions that nonetheless yield equivalent predictions.

2.3.2 Shaplet-based explainers

Shapelet is the most characteristic explanation for time series data. They are time series subsequences that are maximally representative of a class. Shapelets are more interpretable, faster, and more accurate than k-Nearest Neighbors (kNN) [91] which is a traditional approach to perform time series classification [92]. As usual, SMs can be used to highlight which part of the series has contributed the most to the classification. In the following, we briefly illustrate some peculiar explainer for time series classification.

In [93], is introduced a transparent by design method named **Weighted-kNN** that extends the classic majority-voting kNN by proposing weighting schemes. By emphasizing the nearer neighbors using a weighting scheme, it is possible to improve the kNN classifier's quality and stability. The nearest neighbors are considered part of the prototypical explanation.

LASTS, i.e., Local Agnostic Shapelet-based Time Series explainer [94], is a variation of ABELE for time series. As explanation LASTS returns exemplars and counterexamples composed of subseries with a shapelet-based rule. An example of a rule is: "if these shapelets are present and these others not, then x is classified as y".

DOCTORXAI [95] is a local post-hoc model-specific explainer acting on sequential data in the medical setting. In particular, it exploits a medical ontology to perturb the data and to generate neighbors. DOCTORXAI is designed on healthcare data, but it can theoretically be applied to every type of sequential data with an ontology.

Time Series Explainers Comparison. For time series data, kNN weighting schemes are the most common approach. Shaplet-based explanations are promising, and new approaches using autoencoders or ontologies are being developed to improve time series explanations.

2.4 Evaluation Measures for Explanations

There is a wide debate on how to evaluate the quality of the explanation methods and often it is formulated as properties of the returned explanations aimed at capturing concepts as goodness and usefulness of explanations [14–18, 23]. In the following, we describe a selection of established methodologies for the evaluation of explanation methods both from the quantitative and qualitative point of view which are typically used to judge the output of XAI methods. *Quantitative evaluation* focuses on the performance of the explainer and on on the goodness of the explanations returned. In the following, we present the general idea of each metric used later on for benchmarking. Since every metric may vary in its application depending by data type, further details are provided into the various sections.

- *Fidelity* aims to evaluate how good is f at mimicking b. There are different implementations of fidelity, depending on the type of explanator under analysis [50]. For example, in methods where there is a creation of a surrogate model g to mimic b, fidelity compares the prediction of b and g on the instances used to train g.

- *Stability* aims at validating if similar instances obtains similar explanations. Stability can be evaluated through the *Lipschitz constant* [96] as $L_x = \max\frac{\|e_x - e_{x'}\|}{\|x - x'\|}, \forall x' \in \mathcal{N}_x$ where x is the explained instance, e_x the explanation and $\mathcal{N}_x$ is a neighborhood of instances x' similar to x.

- *Deletion* and *Insertion* [97] are metrics that remove the features that the explanation method f found important and see how the performance of b degrades. The intuition behind deletion is that removing the "cause" will force the black-box to change its decision. Among the deletion methods, there is the *Faithfulness* [96]. It aims to validate if the relevance scores indicate true importance: we expect higher importance values for attributes that greatly influence the final prediction[6], a public library written in Python. Given a black-box b and the feature importance e extracted from an importance-based explanator f, the faithfulness method incrementally removes each of the attributes deemed important by f. At each removal, the effect on the performance of b is evaluated. In general, a sharp drop and a low area under the probability curve mean a good explanation. On the other hand, the insertion metric takes a complementary approach. Typically, insertion and deletion evaluations are tailored for specific type of explainers called Feature Importance explainers for tabular data, Saliency Maps for image data, and Sentence Highlighting for sequential data.

- *Monotonicity* [98] can be seen as an implementation of an insertion method: it evaluates the effect of b by incrementally adding each attribute in order of increasing importance. In this case, we expect that the black-box performance increases by adding more and more features, thereby resulting in monotonically increasing model performance.

- *Running Time*: the time needed to produce the explanation is also an important evaluation.

It is worth noting, to the best of our knowledge, there are currently no purely objective evaluation measures that can select the best explainer. A different approach to evaluate explainers consists in generating a synthetic ground truth explanations and compare them with those returned by the explainers [99]. However, this evaluation method through synthetic explanations cannot be transferred to an objective evaluation on real data because if we knew the ground truth explanation, we would not need an explainer. *Qualitative evaluation* is important to understand the actual usability of explanations from the point of view of the end-user: they satisfy human curiosity, find meanings, safety, social acceptance and trust. In [10], the evaluation criteria for the qualitative evaluation are systematized into three categories:

- *Functionally-grounded* metrics aim to evaluate the interpretability by exploiting some formal definitions that are used as proxies. They do not require humans for validation. The challenge is to define the proxy to employ, depending on the context. As

[6] An implementation of the faithfulness is available in AIX360

an example, we can validate the interpretability of a model by showing the improvements w.r.t. to another model already proven to be interpretable by human-based experiments.

- *Application-grounded* evaluation methods require human experts to validate the specific task under analysis [100, 101]. They are usually employed in specific settings. For example, if the model is an assistant in the decision making process of doctors, the validation is done by the doctors.

- *Human-grounded* metrics evaluate the explanations through humans who are not experts. The goal is to measure the overall understandability of the explanation in simplified tasks [102, 103]. This validation is most appropriate for general testing notions of the quality of an explanation.

2.5 Discussion

In this Chapter, we presented an overview of the state of the art in the field of Explainable Artificial Intelligence (XAI), with a specific focus on methods applicable to tabular data and time series. The choice of these two types of data si due to the fact that they play a significant role throughout the remainder of this thesis.

For tabular data, a variety of explanation techniques have been developed, including rules, counterfactuals, feature importance, and examples. Among the popular methods in this domain, we have discussed Shap (SHapley Additive exPlanations), Lime (Local Interpretable Model-agnostic Explanations), and Lore (Local Rule-based Explanations). They are all local methods, hence they explain only one single instance at a time, and agnostic, meaning that they can explain every kind of ML classifier. It is worth noting that Lime is widely used but also considered the most fragile method in terms of its stability. Surrogate-based models, despite their usefulness, suffer from instability due to the inherent randomness in their approach. On the other hand, Shap, with its solid theoretical foundation, has the potential to generate reliable and faithful explanations by providing a comprehensive understanding of the underlying model. However, the explanations provided by Shap are difficult to understand for a non expert. In addition, the only available agnostic Shap method requires an approximation for the computation of the feature importance, which takes a long computational time. Regarding rule-based and counterfactual explanations, they have gained attention in recent years due to their logic formalization, which enables a deeper understanding of the internal decision-making process of AI models. However, the literature lacks standardized metrics for comparing such models, which presents an area for future research.

Regarding time series data, xai methods are relatively less abundant due to the inherent complexity of this data type. Nonetheless, we can still find techniques such as feature importance, often based on shaplets, as well as rule-based and counterfactual explanations.

The analysis presented in this Chapter goes beyond the theoretical perspective by also considering a practical evaluation. In fact, we empirically tested several XAI methods, specifically focusing on the most popular ones that have readily available Python libraries. By comparing the outputs of these methods and utilizing state-of-the-art metrics for quantitative analysis, we have identified various limitations and challenges in this domain.

One major finding is the instability exhibited by the majority of the existing methods, as they tend to provide different explanations for the same input record. Additionally, many explanations are not faithful either to the ML model itself (indicating that the surrogate models employed in XAI methods do not accurately mimic the behavior of the black-box classifier) or to the black-box model's behavior (meaning that the features deemed important by the XAI method do not align with the actual importance as perceived by the black-box model).

These limitations must be carefully considered, as the availability of faithful and stable explanations is crucial for enhancing user trust in ML models and promoting the widespread applicability of such models in various real-world contexts, such as autonomous driving.

Overall, this Chapter provided a comprehensive review of the current state of the art in XAI, highlighting both the theoretical foundations and practical evaluations. The limitations and challenges found highlights the importance of further research and development to address these issues and advance the field of XAI.

Chapter 3

Data Protection in Artificial Intelligence

Data privacy is a critical issue in today's digital age, as the vast amounts of personal information collected and stored by organizations, governments, and individuals can be vulnerable to unauthorized access and misuse [104]. The term *Data Privacy* refers to the protection of personal information from unauthorized access, use, disclosure, alteration, or destruction. Personal information can include a wide range of data, such as names, addresses, social security numbers, financial information, medical records, and browsing history. The increasing use of technology has made it easier to collect, store, and share this information, but it has also made it more difficult to protect the release of sensible information. In fact, during the past ten years, data privacy has been intensively studied due to the increased usage of AI systems, in which sensible data of the users are employed to train various models, also in critical contexts, such as hospitals, hence these data may be at risk. In practice, the availability of human data allow researchers and companies to study and improve their services through the use of these data. However, in order to do so, human datasets are released to allow people to work with them. Unfortunately, there have been several circumstances in which personal information was unintentionally disclosed, such as the Cambridge Analytica scandal and many others that followed during the last few years, such as the leak of data from Uber [105]. Due to this situation, a growing number of people are having concerns about their privacy. Meanwhile, the majority of the countries around the world are working on new privacy laws. This is due to the fact that the majority of the countries in the world recognized the privacy of the individual as a fundamental right that has to be protected. Privacy law was born due to the presence of new technologies and it is quickly evolving in order to keep up with technological innovation. In 2018 the European Union applied the General Data Protection Regulation (GDPR) [1]: a law for the data protection and the privacy of the individuals in the European Union. It addresses the provisions and requirements to apply during the processing of personal data of individuals. Moreover, some basic privacy rights are stated, such as the right to erasure and the right to rectification. Not only the European Union updated its law, but also US and China, highlighting the fact that people are concerned about their privacy. These new regulations forced companies and organizations to re-define their activities in order to be compliant with the law. An in-depth analysis on the regulations for Data Privacy and Artificial Intelligence Systems is presented in Section 4.

During the past years, Data Privacy has evolved: while initially the focus was on the public release of the data, with the purpose of assessing whether the sensitive information contained within data could be discovered by malicious users, nowadays, with the increasing use of ML, data privacy research also includes the analysis of these models. In the following, the first approach is referred to as *Privacy of the data*, in which the goal is to analyze a dataset for the public release, while the second approach, related to ML, is referred to *Privacy of the model*.

In the remaining of this Chapter we overview the field of Data Privacy, first presenting the Privacy-by-Design methodology and then, focusing on *privacy risk assessment* techniques, presented in Section 3.2, which have the goal to evaluate the privacy risk of users, and *privacy protection* techniques, described in Section 3.3, in which the goal is to protect users privacy, avoiding privacy breaches.

3.1 Privacy by design

The main challenge in the setting of data privacy for data mining and machine learning applications is the data and models quality. In particular, the *trade-off* between the quality of the released data or machine learning models and the privacy protection level guaranteed ([2,106]). In fact, increasing the privacy protection of the dataset leads to a loss in quality, that is critically important for data mining and machine learning based services. Therefore, there is the need of finding a solution that ensures the individuals in the dataset to be safe while allowing companies and researchers to use the dataset for research purposes as well as different machine learning models and services.

A seminal work in this setting is the one proposed by Ann Cavoukian in [107]. Here, the author theorized the concept of *privacy-by-design*: a methodology to embed the privacy protection into the design of the service and information system. There are seven principles to fulfill the privacy-by-design:

1 A proactive system and not a reactive one. The main idea is to prevent privacy breaches. In order to do so, the main task is to develop services in which the privacy aspects are handled before invasive events happen.

2 Privacy as a default setting. Privacy is part of the system and therefore is a concern

of the companies and not of the single individual.

3 Privacy embedded into the design of the service. The main idea is to embed the privacy protection into the service, differently from the traditional methods that protect the dataset retrospectively;

4 Full functionality – positive-sum, not zero-sum. The final goal is to have a privacy system in which there aren't trade-offs. All the privacy requirements have to be satisfied.

5 End-to-end security – full life-cycle protection. The privacy has to be assured during all the steps of the life-cycle of the data in the system. This is linked with the constraint of having privacy measures embedded into the system by design.

6 Visibility and transparency – keep it open. The privacy measures embedded into the system must be visible and transparent to the users and the providers. Moreover, these measures have to be subjected to independent verification.

7 Respect for user privacy – keep it user-centric. The main subject of privacy is the individual that generates data. For this reason, the systems have to be user-friendly and in general user-oriented.

From these guidelines, different works have been proposed. One of the most interesting for the purposes of this thesis is resented in [106] by Monreale et al. In this paper, the authors applied the concept of privacy-by-design for big data data analytics and social mining. Their goal is to protect the dataset for the specific purpose of the data mining application that has to be applied. In order to achieve this goal, the main idea is to upgrade from a reactive system, where the dataset is protected retrospectively, to a proactive system, in which the privacy protection is embedded into knowledge discovery technologies by design (as depicted in the guideline point 1). In their work, they proposed a methodology for purpose-driven protection, i.e. a methodology that depends on the kind service and analysis to be developed. In this way, the privacy requirements are incorporated into the service from the beginning. Therefore, it becomes feasible to overcome the trade-off between privacy protection and quality of the service, due to the fact that only the specific service is considered and not a general analytical setting. The methodology proposes to design a privacy-protection strategy tailored to the service to be developed and the type of attacks on data that may jeopardize individual privacy. Moreover, this technique is applicable to every kind of datasets. This general methodology can be extended to take into consideration also machine learning models and their protection that can be obtained also without perturbing data but directly acting on the models. As a consequence, instead of considering only privacy attacks and protection strategies for data, the methodology needs to consider also attacks and mitigation techniques tailored to machine learning models.

3.2 Privacy Risk Assessment

The objective of *Privacy Risk Assessment* techniques, also called *privacy evaluation*, is to determine the privacy risk of the users in a dataset. The main idea behind these techniques is that the privacy risk of an individual depends on how difficult it is to discover information about hims/her, given a dataset under analysis.
Due to the advent of ML models and in general of AI Systems, which employ real human data to learn complex tasks, the Privacy Risk Assessment now focuses on two settings:

- *Privacy of the data.* The privacy of the dataset when publishing a dataset. This is the most studied approach, in which the objective is to evaluate the risk of privacy for the users in a dataset, before knowing for which task the dataset is going to be used. This task is crucially important not only for ML applications, but also for various studies of different kinds;

- *Privacy of the model.* The privacy of the users when publishing a ML model trained exploiting the data under analysis. In fact, the ML model learns patterns and correlations among the data in the training set and this information may be found and exploited by a malicious adversary.

In addition at the distinction between privacy risk assessment for the data or for the models, we can distinguish also between different kinds of information disclosure: *attribute disclosure* and *identification disclosure* [104]. In the first case, the intruder is able to discover personal information about an individual, such as the value of an attribute in a masked dataset, or information about the user, such as age, gender, location, or occupation. In this case, usually the malicious adversary possesses some external information that allows him/her to create a link with the masked data. As an example of attribute disclosure, we can consider a ML model trained on medical records that contain information about patients' diagnoses, treatment history, and other sensitive information. The model may be able to predict certain attributes of individuals, such as whether they have a particular medical condition or not. In the case of *identification disclosure*, instead, the intruder is able to correctly identify an individual in the masked dataset. Also in this setting, the attacker might posses some external information that simplifies his/her task. For example, considering again the setting of privacy risk for ML models, if a model is trained on facial recognition data, the model may be able to identify individuals based on their facial features. The field of *identification disclosure* is crucial in this Thesis and we approach the problem both from the point of view of the privacy risk of the data, but also looking at the privacy of the model. Technically, to evaluate the risk of identification disclosure of the user in a dataset or from a model, we employ re-identification algorithms. They try to link the individuals with the attacker's external knowledge to evaluate the probability of being re-identified. In practice, these algorithms simulate a *privacy attack* to the dataset. The

privacy attack can be of different types, depending on the kind of data in input and on the knowledge the attacker has.
An important aspect when designing this kind of attacks is the background knowledge of the attacker. This is due to the fact that this kind of algorithms try to simulate possible scenarios of the real life, in which a potential malicious attacker can exploit different knowledge to obtain the information he/she wants. The attacker combines the background knowledge with the released dataset in order to enable new inferences and disclose the identity of users. There are several re-identification algorithms available: the traditional methods are based on statistical properties, but there are also ML models employed to evaluate the user's probability of being identified ([104], [2], [106], [108]). For the privacy evaluation of the data, in the following we present PRUDEnce, a framework for the privacy risk assessment and protection [2]. In the ML scenario, the most popular attack on identification disclosure is called Membership Inference attack (Mia) [109]. Training data membership inference attacks aim to determine whether a given data point was present in the training data used to build a model. Although this may not at first seem to pose a serious privacy risk, the threat is clear in settings such as health analytics where the distinction between case and control groups could reveal an individual's sensitive conditions. This type of attack has been extensively studied in the adjacent area of genomics [110], [111], and more recently in the context of ML [112], [109]. In the following, we present the state of the art in the field of privacy risk assessment, with a focus on the attacks and techniques proposed for the two kind of data exploited in this Thesis, e.g. tabular and sequential data. We will start with privacy risk assessment techniques available for analyzing the risk in the data, in Section 3.2.1, as well as the ones for studying the privacy of ML models, in Section 3.2.2.

3.2.1 Privacy risk assessment for data

The field of privacy risk assessment for data has been widely studied for several years now, on account of the need to assess the privacy risk of individual in the data before conducting any kind of analysis with them. One of the core point of the state of the art in this field is PRUDEnce, a framework which allows for the evaluation of the privacy risk for the individuals in a real dataset, proposed by Pratesi et al. in [2]. PRUDEnce is a general procedure for the privacy risk computation and protection which has been employed in different contexts, such as human mobility data, one of the kind of data exploited also in this Thesis, but also purchase data [113, 114]. In this Thesis, PRUDEnce has been exploited as a background procedure to assess the privacy risk of the users in a dataset for the works presented in the Part III. Technically, PRUDEnce's main goal is to allow the publication of the dataset while maintaining also the data utility for the final service. In fact, the setting under analysis considers a Data Analyst that requests a dataset to a Data Provider with the task of developing an analytical service. Clearly, the Data Analyst has some requirements about the data, critically important to develop the analytical service. However, the Data

Provider has to preserve the privacy of the individuals in the dataset. Given this setting, the Data Provider aggregates, selects and filters the original dataset D. In this way, it produces a set of datasets $\langle D_1, D_2, ..., D_z \rangle$ with different data structure and/or aggregation of the data. This procedure can be done several times, until the Data Provider considers the data delivery safe.
In particular, the procedure the Data Provider has to follow for the data delivery is composed by 4 steps:

1. *Identification of attacks*: identify a set of possible attacks that a malicious adversary might conduct in order to re-identify the individuals in the mobility datasets $\langle D_1, D_2, ..., D_z \rangle$;

2. *Privacy risk computation*: simulate the attacks identified in the previous point and compute the set of privacy risk values for every individual in the datasets $\langle D_1, D_2, ..., D_z \rangle$;

3. *Dataset selection*: among the datasets $\langle D_1, D_2, ..., D_z \rangle$, we select D as the dataset that has the best trade-off between the privacy risks and the quality of the data. This trade-off is tailored by the privacy risk that is tolerated and the requirements of the data that the Data Analyst asked for;

4. *Risk mitigation and Data delivery*: apply a privacy-preserving transformation on the chosen dataset D to eliminate the residual privacy risk. The result of this operation is a filtered mobility dataset D_{filt}. When D_{filt} is adequately safe, it is delivered to the Data Analyst.

The step two, called *Privacy risk computation*, is the one that evaluates the privacy risk of the individuals in the dataset. The privacy risk computation procedure theorized in [2] is general and requires the definition of a privacy attack. In fact, only the privacy attack is data-dependent. The *privacy risk computation* defined in PRUDEnce is the following:

1. Define an attack, based on a specific background knowledge category;

2. Consider a set of background knowledge configurations $B_1, B_2, ..., B_m$;

3. For all the configurations $B_1, B_2, ..., B_m$, compute all the possible instances $b \in B_k$ and its probability of re-identification;

4. For each individual, select the maximum privacy risk, i.e., the maximum probability of re-identification across all the instances $b \in B_k$.

In this way, the authors provided an *exhaustive* privacy risk evaluation technique, by considering all the possible background knowledge the attacker could have over a given dataset (or dataview of the original dataset). They generated different scenarios, starting from the one with minimum knowledge (only one information for each user) to the worst case scenario, that corresponds to the background knowledge equal to the original dataset. Clearly, the adversary background knowledge is context-dependent and determines the possible kinds of attacks. As an example, in the case of human mobility dataset, such as the ones employed for this Thesis, the background knowledge could be a set of locations visited by the user. In this setting the probability of re-identification is defined as the probability of correctly associate a record to a unique user, given the background knowledge under analysis. Mathematically, the probability of re-identification for each user corresponds to the division between the number of records that match the background knowledge over the total number of records in the dataset that matches the background knowledge. In formulae,

$$Pr_D(d = u|t) = \frac{supp_u(t)}{supp_D(t)} \tag{3.1}$$

in which D is the dataset under analysis, t the background knowledge considered and d the record under analysis. As mentioned earlier, in the attack proposed in [2] all the possible background knowledge an attacker could have has been considered. Therefore, the formulae in Eq. (3.1) has to be computed for each background knowledge under analysis. The risk of re-identification, also called *privacy risk*, corresponds to the maximum value among the different re-identification probability results, mathematically

$$Risk(u, D) = max(Pr_D(d = u|t)) \tag{3.2}$$

Overall the background knowledge $t \in BK$. Employing this re-identification technique we analyze all the possible configurations of the privacy attack defined in the procedure in step 1. Therefore, this methodology provides an exhaustive analysis.

In [2] the authors also proposed some privacy attacks specifically tailored for human mobility data. However, considering all these scenarios requires a long time to compute. Therefore, from a computational point of view, it is inefficient, especially when dealing with big data. Moreover, it is developed from the point of view of the companies and not of the user. In fact, in this setting, the attacks are tailored for an entire dataset, such as the ones stored in the servers of the companies, and are not suited for the evaluation of the privacy risk of only one person at the time. In particular, if a new record arrives, we need to recompute the privacy risk for every user in the dataset.

In order to solve these problems, Pellungrini et al. proposed a machine learning based approach for human mobility data [5]. Their idea is to employ machine learning algorithms to evaluate the privacy risk of the users. Technically, they employed the privacy risk computation procedure theorized in [2], but they proposed other kinds of privacy attacks for human mobility dataset and improved the efficiency of the privacy evaluation algorithm. In fact, they trained a machine learning model in order to let it learn the relation between the patterns present in the mobility data and the associated privacy risk. In order to do so, they employed a Random Forest regressor and a Random Forest classifier. For the training of these machine learning models the authors labelled different human mobility dataset. In order to do so, they employ the re-identification attack proposed in [2]. In this way, for each input record there is an associated label that defines its privacy risk. They did this procedure just once, at the beginning of the computation, in order to train the model. They tested their predictive method with different kinds of aggregated data as input. In particular, they employed raw trajectories, frequency vectors and probability vectors. With regards to frequency vectors, they computed a vector for each user composed by tuples containing $(location, frequency)$ for each location visited at least one time by the user under analysis. The probability vectors, instead, are vectors composed by (*location*,*probability*) tuples, in which the *probability* denotes the probability of finding that location among the ones visited by the user under analysis. In this way, they provide an efficient and flexible approach for privacy evaluation. Moreover, the results obtained are very promising: they are able to identify with high accuracy the class of people for which the privacy is at risk. They also evaluated the performances of the model when tested over a dataset that wasn't used for the training, showing that this approach does not depend on the specific dataset in input.

3.2.2 Privacy risk assessment for ML models

The increasing prevalence of smart technology in everyday life, such as self-learning and auto decision-making systems, is largely due to advancements in ML. Applications such as Gmail's spam filtering (Dada et al., 2019 [115]), YouTube's video recommendations [116], text correction software [117], and speech recognition [118] all utilize ML algorithms to improve their functionality. Additionally, ML is used in cybersecurity for spam detection [119]), malware detection [120], fraud detection [121], and bot detection [122].

While ML models can greatly enhance the capabilities of a system, it also presents potential vulnerabilities regarding the privacy of the users in the dataset exploited during the training phase. Attackers may exploit flaws in ML systems to infiltrate and manipulate them for malicious purposes, potentially compromising the system's reliability, confidentiality, and availability. As an example, we refer to [123], in which the authors showed that the use of ML models in healthcare can compromise the privacy of the patients by exposing some personal information. In the work of Fredrikson, the sensible information exposed was the patient's genetic markers. However, the privacy threats are not limited only to the medical context: in fact, in [124], the authors presented real privacy dangers of using deep learning in finance. In this field, the records used for the training and the model parameters are considered confidential. Hence, if these information are exposed, it is considered a privacy exposure.

Up to now, we can divide the existing threats in this context into two groups: (i) the first one, called *direct information exposure*, deals with direct, intentional or unintentional data breaches; while the second group of privacy attacks in this setting, called (ii) *inferred* *information exposure*, in which the attacker actively tries to infer information from the ML models. Regarding the group (i), it is composed by direct data breaches, such as data sharing by transmitting confidential information without the proper encryption. An example of this kind of exposure can happen with ML as a service (MLaaS). In this case, in fact, there still is ambiguity about how the data are processed, if the data are sent to the cloud or processed locally, or even what happens to the data when the process is finished. Due to these ambiguities, private data can be exposed directly from the owner and then incur in privacy exposures. However, in this dissertation, we mostly focus on the second group of attacks, since we focus on the setting in which there are not direct data exposures, but rather malicious adversaries that tries to infer information clearly kept private. This group of attacks is composed by many different kinds of attacks, growing day by day. They are often referred to as *privacy attacks on the model* and can be divided into two categories: (i) white-box attacks and (ii) black-box attacks. White-box attacks refer to attacks where the attacker has access to the model's architecture and parameters, allowing them to make targeted attacks exploiting this knowledge. Black-box attacks, on the other hand, refer to attacks where the attacker does not have access to the model's architecture and parameters, but can use the black-box as a Machine Learning as a Service, knowing only the input and output shape. Clearly, the last approach requires less knowledge with respect to the first one, hence the black-box attacks are commonly considered most threatening. For this reason, in this Thesis we refer to the last kind of attacks and in the following we describe in details the literature related to it. The remaining of this Section is organized as follows: we first present the *Membership Inference* attacks, in Section 3.2.2, in which the objective of the attacker is to determine the membership of a given record to the training dataset. In particular, we describe the Membership Inference Attack [109], the first attack of this kind published, and the Label Only Membership Inference Attack [125], exploited in the remaining of this Thesis. Then, we present the other kinds of privacy attacks available for attacking ML models: firstly, *model inversion and attribute inference* attacks, in Section 3.2.2, following, we describe the *model stealing* attacks, in Section 3.2.2, and we conclude with Section 3.2.2 the presentation of this kind of attacks with the *property inference* attacks.

Membership Inference

In this Section we present the membership inference attacks against ML models. This kind of attacks speculates whether or not the given data instance has contributed to the training step of the target model. The underline assumption is that if a record was used in the training phase of a model, it would gives a higher confidence score with respect to a record that was never seen by the trained ML model. In this setting, the first and most popular attack is the Membership Inference Attack (MIA), proposed by Shokri et al. [109]. The procedure of MIA is quite complex, since it needs to train *shadow models* able to mimic the behaviour of the black-box model. Then, exploiting the information of the shadow models, Mia fits one attack model for each output class of the original model. An in depth description of this methodology is reported below. Starting from this work, several other attacks have been proposed. Long et al. [126] proposed another approach in which the objective is to evaluate the membership of a given instance with more accuracy with respect to the Mia. To achieve this goal, they modify the shadow training, by creating shadow models with and without the instance under analysis. Then, the procedure follows similar to Mia. They show that they are able to achieve more precise results, but the procedure is adding an overhead affecting the computational time. To overcome this limitations, Salem et al. [127] proposed a methodology able to attack the ML model, but with fewer assumptions and less steps in the procedure. In fact, they showed that is not necessary to fit several shadow models, the knowledge of the target model structure and not even dataset resembling the training dataset distribution, Clearly, relaxing these assumptions fasten the procedure, degrading the effectiveness of the attack only by a small amount. Another interesting approach of this kind shows that it is not even necessary to obtain the confidence vector when dealing with text data [128]. Lastly, Hayes et al. [129] presented an application of the Membership Inference Attack to the Generative Adversarial Networks (GANs). In this setting there are both white-box and black-box attacks. Since we are interested in black-box models we focus on this kind. In the following, we describe in details two attacks which are going to be exploited in the remaining of this Thesis. In particular, we present the most popular attack, called Membership Inference Attack (Mia) [109], and his variation, the Label Only Membership Inference Attack [125].

Membership Inference Attack This method aims to identify if a specific data record was included in the training dataset for a model. When an adversary is aware of a record, discovering that it was utilized to train a model suggests that there has been a leak of information through the model. This can result in a violation of privacy, such as determining that a patient has a certain disease based on the knowledge that their clinical record was used to train a model related to that disease. This attack belongs to the privacy risk assessment techniques in particular to the identification disclosure methods. The objective of the attacker is to build a binary classification model Figure 3.1 that can recognize such differences in the target model's behavior and use them to distinguish members from non-members of the target model's training dataset based solely on the target model's output.

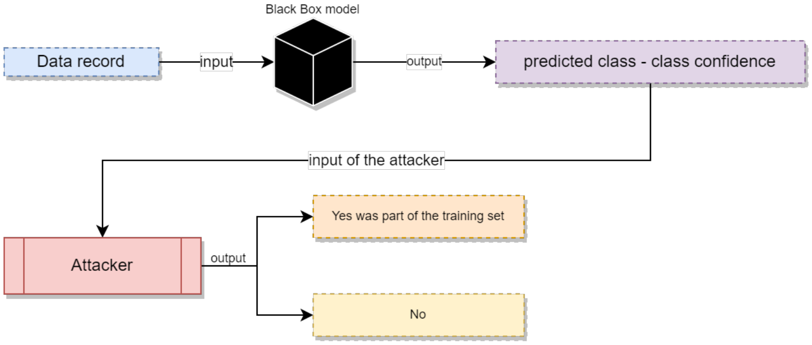


Figure 3.1: Membership inference attack in the black-box setting. The attacker queries the target model with a data record and obtains the model's output. The output is made by the vector of probabilities, one per class, that the record belongs to a certain class. This prediction vector, along with the label of the target record, is passed to the attack model, which infers whether the record was in or out of the target model's training dataset.

To carry out this attack we need some assumptions:

- Access to the black box's query function, which allows us to see the predict_probability (also called confidence values) for any input data we try.
- Knowledge of the input and output format of the model, as well as the type of machine learning architecture and how it was trained.
- Background knowledge about the population of the dataset used to train the model.

The attack succeeds if the attacker can correctly determine whether this data record was part of the model's training dataset or not. The typical measurements for determining the success of an attack are precision (the percentage of records identified as members that are members of the training dataset) and recall (the percentage of actual members of the training dataset that are correctly identified as members by the attacker). The intuition of this attack is that a good machine learning model is one that not only classifies its training data but generalizes its capabilities to examples it hasn't seen before [109]. This goal can be achieved with the right architecture and enough training data. But in general, machine learning models tend to perform better on their training data. Due to this behavior, we will always have higher confidence in the output of already-seen data than new ones. The attack model then learns to distinguish between the training inputs classified with high confidence and other, non-training inputs that are also classified with high confidence by finding the right decision boundary on the relation between input and confidence.

The algorithm starts by training shadow_models[1] to mimic the behavior of the black box. The dataset used to train these models is a disjointed set from the actual training set and can be generated in different ways:

- Model-based synthesis: in this method we randomly generate data inputs and submit them to the black box and by looking at the class confidence if it has a very high one we can assume it was part of the tr_set otherwise not. We continue to generate samples till we have a sufficient number of samples for each target class.

- Noisy real data: The attacker may have access to very similar data to the target model's training data. What we do is simply add noise to the data like adding standard deviation in each feature and for binary features flipping the bits.

- Statistics-based synthesis: In this approach instead the attacker can have statistical information on the data, like the marginal distribution of different features. In this way, it can sample new examples similar to the ones used by the target model.

Let's call this dataset D_{shadow}, we split it in two datasets D^{train}_{shadow} and D^{test}_{shadow} by following the 80-20% common split (Figure 3.2). Given k shadow models we divide the D^{train}_{shadow} and D^{test}_{shadow} in k sub-dataset each of them $(x, y_{true}) \in D_{shadow_i}$ where $y = f^{i}_{black\_box}(X)$ and we train each shadow model to imitate the output y generated by querying the black box. After training the shadow models in different portions of the starting D_{shadow} we merge their results in $(x, y_{true}, y_{shadow}, in/out) \in D_{attack}$ where $y_{shadow_i} = f_{shadow_i}(x)$ and in/out is assigned knowing that input data was coming from D^{train}_{shadow} or D^{test}_{shadow}. Once we have the D_{attack} ready, we can train different attack models one for each class to increase the accuracy. So, we split D_{attack} into c_{target} partitions one for each class and train a model specialized on prediction of the membership status for x.

[1] shadow_model is a blank model with the same architecture as the black box. The idea is that similar models trained on relatively similar data records using the same training algorithm behave similarly.

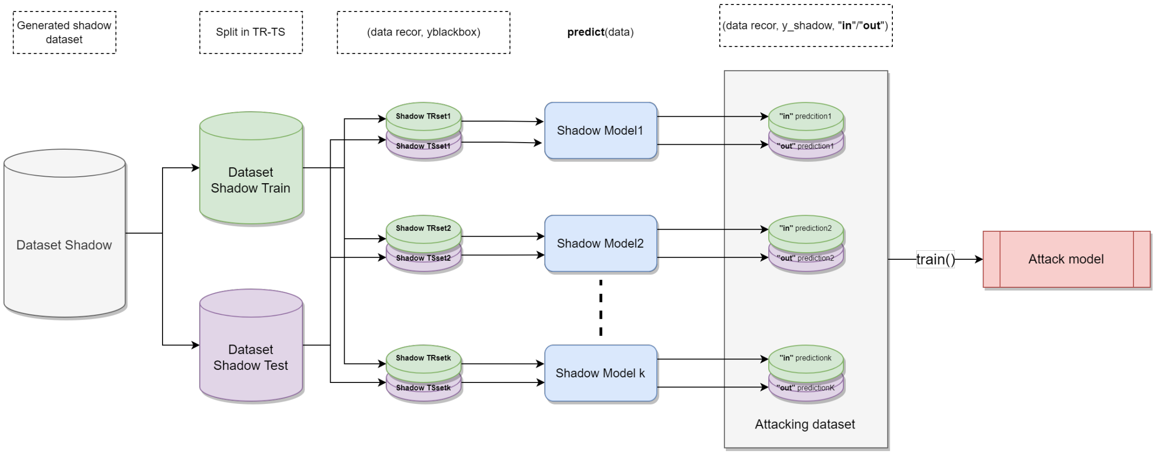


Figure 3.2: To train the attack model, the inputs and outputs from the shadow models are used. The Training dataset of a shadow model is queried in the model to get the output, and label it as "in." These output vectors are then added to the training dataset of the attack model. Additionally, the shadow model is queried with a separate test dataset not used in training, and label the outputs as "out." These are also added to the attack model's training dataset. This creates a dataset that represents the behavior of the shadow models on both their training and test datasets. Finally, multiple attack models are trained, each targeting a specific output class of the target model.

The success of the attack is correlated to the overfitting[2] of the model. From different studies and analyses the more the model is overfitted the more vulnerable to membership inference. Usually, the overfitting is estimated by the (traintest) gap [130] which is the difference between the accuracy of the target model on its training and test data. It was demonstrated in the plots of [109] that, as expected, bigger (train-test) accuracy gaps are associated with higher precision of membership inference.

Label only membership inference attack Label-only MI attack is a type of attack that targets a machine learning model by only using the hard labels (outputs) it produces [125]. This is different from traditional Membership Inference attacks, which require access to a model's "confidence vector," assigned to the output. In real-world scenarios, this information may not be available (confidence masking)[3]. The label-only MI attack can still be successful granting similar performance as the traditional attack (using confidence).

[2] Overfitting in machine learning occurs when a model is trained too well on the training data, and as a result, performs poorly on new, unseen data. This happens when the model is too complex and can memorize the noise in the training data, rather than learning the underlying pattern

[3] One time of protection against MIA attacks where the black-box model does not show the confidence probabilities.

The concept behind this is that by analyzing how model's predictions change when a small change is made to the input data, we can gauge the model's confidence in its predictions. This is known as "model robustness." A model is typically more reliable on data it has already seen during training, but may be less reliable on new data. Theoretically, the robustness of a model can be measured by determining the distance of a given point to the closest decision boundary[4] (see Figure 3.3).

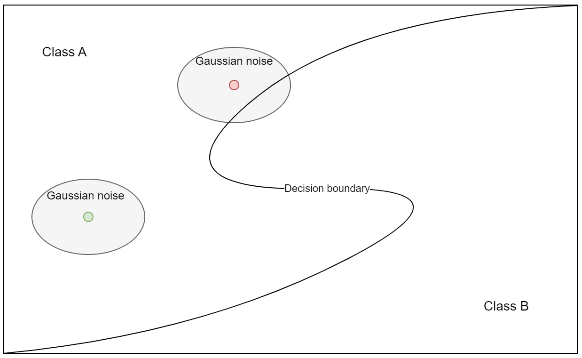


Figure 3.3: This diagram shows an example of decision boundary spitting the data in *class A* and *class B*. We can see two examples: the red dot and the green one. For the red one, after having created a batch of examples with Gaussian noise, we can see that some of them are miss-classified, instead for the green one, all of them are correctly classified. This gives us information about how close is the point to the decision, approximating the confidence vector for that example. The model is more robust to perturbation on the green point than the red one providing the information that is more probable to be part of the training set.

The greater the distance, the higher the model's confidence; the closer to the boundary, the lower the confidence. The algorithm of this attack differs from the traditional MIA attack, where instead of using multiple shadow models here they trained only one shadow_model in all the D_{shadow} and instead of using multiple attack models one per class they used only one attacker trained on the shadow model's predictions. To perturb the input they used a Gaussian noise whereas, in the attacker model, they use an iterative algorithm to find the best cut on the robustness score to classify the points as members or non-members of the training set. In the paper, they demonstrated that the "confidence masking" protection against MIA attacks is insufficient to prevent the leakage of private information. From the experimental results, they have shown that training with differential privacy or strong L2 regularizations are the only current defenses that meaningfully decrease leakage of private information, even for points that are outliers of the training distribution, highlighting again how overfitting is related to the success of the attack.

[4] A decision boundary is the region of a problem space in which the output label of a classifier is ambiguous.

Model inversion and Attribute Inference

In this Section we present model inversion and attribute inference attacks. This kind of attacks targets the privacy of the attributes. In this setting, the adversary tries to infer sensitive attributes of a given record from a released model, knowing only some non sensitive attributes. In this context we can find the work of Fredrikson et al [123], in which the authors exploits publicly available linear regression models to recover sensitive attributes, such as genomics information about the patient, from simple information about the patient, like age, height and age. Technically, the attack is formalized as a maximization of the posterior probability estimate of the sensitive attributes. Another interesting work in this setting is the one proposed by Salem at al. [131], in which the authors proposed a model inversion attack for online learning. They exploit a generative adversarial network to infer the sensible information exploiting the difference in the model before and after the gradient updates. Lastly, He et al. [132] proposed a novel set of attacks to compromise the privacy of queries at test time in the context of collaborative deep learning systems. This attack seems threatening since with no access to other participants' data, the attacker can recover an arbitrary input fed into the collaborative system.

Model Stealing

In this Section we describe the model stealing attacks, which can be of two kinds: hyper-parameter and parameter inference. This kind of attacks are tailored for contexts in which there is only a black-box access to the ML model. This kind of access is one of the most common since trained ML models are considered intellectual properties, hence extracting the model can be considered a privacy breach. As a consequence, the attacker in this setting tries to learn parameters or hyper-parameters information of the model to proceed later with even more complex attacks. In this context we can find a work of Tramer et al. [133], in which the authors presented an attack that finds parameters of a model simply observing its predictions in the form of confidence values.

Property Inference

Another kind of attack against ML models is the property inference. This kind of attacks tries to infer specific patterns of information by attacking the target model. An example of these attacks is the memorization attack [134], in which the objective is to find sensitive pattern of the training dataset from the model.

3.3 Privacy Risk Protection

The main purpose of the techniques and tools proposed in this Section is to protect the privacy of the users in the dataset. As in the case of privacy risk assessment techniques, also in this setting the problem of protecting the privacy of the users can be tackled from the point of view of the data or from the one of the ML models, specifically in the training phase. In the following, we first describe the techniques that work on the data, in Section 3.3.1, then we present the ones modifying the ML training phase, in Section 3.3.2.

3.3.1 Data protection mechanisms

This Section presents the relevant literature in the context of privacy protection mechanisms that works on data. The main purpose of the techniques and tools proposed in this Section is to lower the privacy risk of the user in the dataset by masking the original data. Therefore, at the end of the privacy protection process, the output dataset is a modified version of the original one with a lower privacy risk. In practice, with the techniques proposed in Section 3.2, we are able to assess the privacy risk of the users and if we find that there is a high risk of privacy breaches, we can apply one of the protection techniques presented in the following.

The main concern when applying this kind of methods is on the *data utility*: the masking method increases the protection of users' privacy, but often the final dataset is no longer useful for other applications, such as data mining and ML algorithms, that try to mine the data to find patterns and correlations. This is due to the fact that protection techniques try to obfuscate original information by adding noises or by reproducing the same information multiple times, losing possible trends and patterns in the data. Therefore, the main challenge in this setting is to find a trade-off between humans' privacy risk and the data quality of the masked dataset.
To achieve this goal, in [2], the first step is to evaluate the level of the privacy risk of the users in the dataset by employing some re-identification attacks, like the one proposed in Section 3.2. Then, a privacy protection method is applied to lower the overall privacy risk, given the dataset under analysis. Depending on the initial privacy risk level of the dataset, the variations applied from the masking method can be more or less onerous.

In this setting the mechanisms proposed are applied directly on the data, hence they are not compulsorily related to the context of ML. In fact, we can find two kinds of protection mechanisms in this context: either *context-free* procedures, in which the data is protected independently of the purpose that the data will be used for, or *context-aware* methods, in which it is assumed to know the context where the data is going to be used. This last group of procedures usually tackles the problem of achieving a good privacy-utility trade-off due to the greater knowledge of the setting they have.

When dealing with context free procedures, a first simple and naive solution is the *naive data anonimization.* With this name we refer to the simple removal of identifiers from the data, such as names and addresses. Clearly, this is a starting point, but it is not enough to provide privacy protection as shown in [135].

However, there are other solutions in this setting. First of all, **K-anonimity** [136], which is one of the most popular and employed method due to its simplicity. The objective of this technique is to provide a dataset in which each participant's information cannot be distinguished from at least $k-1$ other participants whose information is in the dataset. The k-anonymity method and works as follows: first, it analyzes the quasi-identifiers present in the data, so that it is then able to generalize them up to the point in which an individual's data entry is indistinguishable from others $k-1$ entries in the dataset [136]. k is a user-defined variable and therefore this technique allows the user to define the privacy protection level for the dataset. Different variants of this methodology have been proposed, depending on the kind of data considered. As an example, k-anonimity has been applied for human mobility datasets, but also in the context of querying tabular database ([137], [138], [139]).

However, Aggarwal et al. [140] showed that the k-anonimity technique perform poorly when dealing with high dimensional data. In addition, depending on the different background knowledge considered during the evaluation of the privacy protection method, k-anonymity may not guarantee privacy [141]. For these reasons, different approaches have been adopted to protect user privacy comprehensively. In [141] the notion of l-diversity is theorized. With this technique, each equivalence class has at least l "well represented" values for each sensitive attribute. This technique improved the privacy achievement of k-anonymity, but it has some limitations. In particular, there are re-identification attacks able to achieve attribute disclosure, as it has been proven in [142]. In [142] the author presented another approach, specifically designed to overcome this limitation: t-closeness. In this approach, the distribution of values of a sensitive attribute in any equivalence class needs to be close to the empirical distribution (i.e., the distance between the two distributions should be no more than a threshold t).

In this context we also find another popular solution: **Differential Privacy**. It is another approach for solving the problem of protecting the individuals of a dataset from privacy breaches. The first theoretical definition of differential privacy was published by Dwork in [143], in which there is also a rigorous mathematical definition of the concept of privacy and privacy breaches. The main concept of differential privacy is that the protection of the dataset should be independent on the kinds of information it contains. Technically, an algorithm is said to be deferentially private if its behavior changes *in a small way* if a single user is added or is removed from the dataset. Therefore, regardless the details of anyone in the dataset, the privacy of the individuals are still guaranteed [104]. Technically, the first formulation of the differential privacy refers to the definition of ϵ-differential privacy ($\epsilon - DP$ for short in the following), which states that, for $\epsilon \geq 0$, an algorithm A satisfies $\epsilon - DP$ if and only if for any pair of datasets D, D^1 that differs in only one element, we have:

$$\mathcal{P} = [A(D) = t] \leq e^{\epsilon}\mathcal{P}[A(D^1 = t)] \forall t \tag{3.3}$$

in which $\mathcal{P} = [A(D) = t]$ denotes the probability that the algorithm A outputs t. Following this definition, the privacy loss is:

$$PL = \ln \frac{\mathcal{P} = [A(D) = t]}{\mathcal{P} = [A(D^1) = t]} \tag{3.4}$$

Considering the privacy loss, the differential privacy mechanisms try to approximate the effect of considering or not an individual as part of the dataset. In practice, this procedure aims at ensuring that each individual included in the data has a small effect.

There are many different methodologies that allows for achieving a differentially private dataset. One of the most common method is the Laplace Mechanism [143]. In this approach, given a target function f and a fixed privacy budget $\epsilon \geq 0$, the randomized algorithm is $A_f(D) = f(D) + x$, where x is a perturbation random variable, drawn from a Laplace distribution. In addition, the authors also defined Δ_f, which is the global sensitivity of the function f. The formal definition is: $\Delta_f = \sup|f(D) - f(D^1)|$, in which all the different pairs (D, D^1), that differs in only one element, are considered. Clearly, the main objective is to find the best sensitivity, dataset wise. However, it is not always easy to find it, especially in the case of deep learning models [144]. The differential privacy has become more and more important also due to an important characteristic: the composition property. Given two mechanisms, with privacy budget ϵ_1 and ϵ_2 applied to the same dataset, together they use a privacy budget of $\epsilon_1 + \epsilon_2$. Hence, the composition of multiple differential private mechanisms consumes a linearly increasing privacy budget. This is a crucial characteristic since it allows to employ this mechanisms also in decentralized settings. Hence, before sharing the data, each participant can apply a differential privacy randomization method and achieve a local differential privacy.

Recently, Google proposed RAPPOR [145] to allow web browser developers to privately collect usage statistics. In addition, there is also the Pufferfish framework [146]. It can be used to create new privacy definitions tailored for specific applications.

Another procedure that belongs to this category is the **Semantic Security**. The main idea of this methodology is that the advantage of an adversary with background information should be cryptographically small. In this setting, the advantage of an adversary refers to a measure of how successfully an adversary can attack a cryptographic algorithm. Theoretically, this approach seems interesting, however, up to now it is not feasible in practice [147].

Lastly, there are **Information Theoretic** privacy methodologies. They are context-aware procedures, in which the solutions proposed model the datasets exploiting information about the tasks the data are used for. In this context, there is an interesting work from Huang et al. [135]. In their work, they propose a context-aware privacy framework, called generative adversarial privacy (GAP), that exploits Generative Adversarial Networks (GAN) to generate protected datasets. The framework is composed by two components: one sanitize the data by removing private attributes, while the other exploits the GAN to try to infer private information. Hence, their approach focuses on the data and also the privacy protection level is defined through them.

3.3.2 Training Phase protection mechanisms

In this Section are outlined the most common mechanisms for protecting the privacy of the users when using ML models during the training phase. We remark that this kind of methods are tailored for neural network models. In this setting, there are mainly two techniques: differential privacy and encryption.

We presented the formal definition of differential privacy in Section 3.3.1, when it is applied to the data in a context-free manner. However, there is also another variant of this technique, which aims at modifying ML algorithms to satisfy differential privacy. In this context, the differential privacy mechanisms can be applied at different levels: either on the input data, on the loss or objective function, on the gradient updates, on the output and on the labels [8]. Clearly, when the differential privacy is applied to the input data, this corresponds to the first case, proposed above in Section 3.3.1. For the case of objective function perturbation and output perturbation, there are methods available only for convex objective functions [148]. The most popular way to apply differential privacy in ML models is the gradient perturbation. This kinds of techniques requires the gradient norms to be bounded, which is not always the case, especially for deep learning models. To overcome this problem, usually the procedure of clipping the gradient is applied [149]. Recently, Bu et al. [150] proposed to apply the Gaussian Differential Privacy to deep learning models to better analyze the privacy budget exhaustion during the training. Overall, there is a great amount of works in this context, which tries to apply different variants of the differential privacy. However, the application of differential privacy methods have one main limitation: the clipping mechanisms and the addition of noise yields loss of utility.

For this reason, also other mechanisms have been explored. In particular, Homomorphic Encryption [151–153]. These algorithms are tailored for neural network methods and they still suffers from many limitations. As an example, only the linear functions can be computed in this setting. Due to this limit, the application of this kind of methods are still in its infancy.

3.4 Discussion

In this Chapter, we provided an overview of the state of the art in Privacy, with a focus on two main directions: privacy of the data and privacy of the ML models. The first direction examines the privacy of users' data, particularly in cases where datasets are released, as an example release for research purposes or for marketing purposes. The second direction addresses the privacy of users when the actual data is kept private, but the ML model trained on that data is made public.

Beside the two directions already mentioned, in the context of privacy, there are two primary tasks are considered: privacy assessment and privacy protection. Privacy assessment mainly revolves around simulating privacy attacks on datasets or ML models to evaluate their privacy exposure. On the other hand, privacy protection aims to strike a balance between preserving the quality of data and safeguarding privacy. One popular strategy for privacy protection is differential privacy, which involves adding noise to the original data. While this approach effectively protects privacy, it can also obscure patterns and valuable insights within the data when utilized for data mining and ML purposes. Also in this setting, the privacy protection techniques can be applied to the data or to the ML models.

The analysis conducted in this Chapter focuses mostly on privacy risk assessment methods for data publishing which is one of the major topics of this thesis. Our analysis reveals that these methods tend to be slow, not available as online services, and often operate at the group level rather than the individual level, due to the formalization of the attacks exploited but also due to the structure of the processes proposed. In terms of privacy risk assessment for ML models, the findings raise concerns, even though executing such attacks requires a complex setup involving shadow models and attack models. This complexity makes it challenging for attackers to successfully compromise the ML model. However, the direction proposed by LabelOnly, an attack method with less requirements with respect to the standard Mia, is leading the way for more powerful and faster attacks in this contexts.

Overall, this Chapter highlights the importance of privacy in both data and ML models. The identified limitations in privacy risk assessment methods underscore the need for more efficient and individual-level assessments, especially for data publishing. Similarly, while privacy protection strategies such as differential privacy offer strong privacy guarantees, further research is necessary to mitigate the potential loss of valuable information due to added noise. Future work should also focus on addressing the challenging task of assessing privacy risks in ML models, considering the intricate nature of the available attacks.

By addressing these challenges and advancing the field of privacy, we can enhance user trust, promote responsible data practices, and facilitate the development of privacy-preserving ML models that can effectively balance privacy concerns with data utility.

Chapter 4

Artificial Intelligence laws and rules

This Ph.D. dissertation discusses some of the ethical issues in the context of Artificial Intelligence extremely important nowadays: *Privacy* and *Explainable AI*. In fact, thanks to the technological advancements made during the last two decades, Artificial Intelligence (in the following referred to as AI) has been applied in various contexts of our daily life, thus also making it a requirement for ethical discussion regarding the use of these systems, as well as legislation to regulate their proper usage. To appreciate how much AI is an embedded component of our lives, we need only to mention that products that use AI include interactive maps and navigators, text editors with auto-corrector or auto-completion services, or digital assistants, such as Alexa, now present in most of our homes, as well as social networks. In addition, there are also biometric re-identification systems, such as face recognition systems for unlocking our phones, but also for surveillance and security, exploited by government facilities and airports. Last but not least, AI is also crucial in recommendation systems, which are exploited during working tasks, such as the one in the Google products, and during free time, such as Spotify or Netflix suggestions. From this brief description of how much AI is now an indispensable part of our lives, we can now clearly sense the need, highlighted in recent years, to evaluate the ethical aspects of AI use and possible regulation of its use. In fact, in 2017 Chouldechova pointed out that nowadays, companies are increasingly embedding ML models in their AI applications, incurring a potential loss of safety and trust [27]. To overcome these problems, there is the need of tackling the trustworthyness of AI products. In particular, the field of *data privacy* has been tackled firstly. The reason behind it is that the growing proliferation of technology, the capacity to amass, stockpile, and exchange information has been greatly facilitated, albeit at the cost of concomitant challenges in safeguarding sensitive information. Notably, data privacy has undergone heightened scrutiny over the preceding decade, owing to the expanded use of AI systems, which necessitate the use of sensitive data from users for model training, particularly in critical domains like hospitals, thereby placing such data at greater risk. In practical terms, human data availability allows researchers and companies to scrutinize and improve their services by leveraging these data. Yet, to accomplish this, human datasets must be made accessible to allow for their utilization. Unfortunately, inadvertent disclosure of personal information has occurred several times, exemplified by the Cambridge Analytica debacle and subsequent episodes, such as the data breach at Uber. [105]. The significance of Data Privacy in the context of AI is not the only ethical consideration that has been identified. An example of great importance is the article published in 2016 by ProPublica[1] which exposed the racial bias present in the algorithm COMPAS, utilized by US judges to determine the recidivism risk of prisoners. This incident was caused by the algorithm being trained on biased real-world data, where most released Black and Latino individuals was found to have committed crimes again. Consequently, the algorithm learned and replicated this discriminatory pattern, without taking into account the circumstances of the crimes committed. What makes this case particularly noteworthy is that the discrimination of the algorithm was identified through an explanation of its reasoning. As a result of this and other similar cases, explainability in AI algorithms has become increasingly critical in recent years.

As a result, there has been a growing recognition in recent years that legal frameworks need to be adapted to account for the ubiquitous presence of AI in various aspects of our lives. Given the anticipated growth of AI applications in the future, legal experts have highlighted the need to update existing legal instruments or develop new ones to address these technological developments and their associated legal implications.

In the remaining of this Chapter we first describe the principal components of the General Data Protection Regulation, in Section 4.1, a core regulation for Europe which impacts also the majority of the other countries of the worlds. Then, we conclude the Chapter by presenting other relevant laws and regulations for Artificial Intelligence in industrialized nations, in Section 4.2.

4.1 General Data Protection Regulation

Personal data holds high value: corporations like Facebook and Google earn revenue by trading personal data with advertisers. Given the enormous financial interests involved, estimated to be around trillion dollars, the European Union at the beginning of the second decade of year 2000, started questioning whether such companies prioritize the best interests of their users. Starting from this need, the European Union on the 25 of May 2018 published the GDPR (General Data Protection Regulation) [1]. It is a legislative instrument that lays down a set of rules governing companies' use and management of personal data, regardless of

[1] ProPublica

their number of employees or incomes. The term *personal data* is crucial for the GDPR since it refers to any information that can be used to identify a specific individual. Essentially, it encompasses any confidential details that one would prefer to keep private and out of unauthorized possession. Various examples of personal data include but are not limited to: name, phone number, address, date of birth, bank account number, passport number, social media posts, geotagging, health records, race, religious affiliation, and political opinions. To illustrate, think of personal data as a jigsaw puzzle, in which each individual piece alone may not carry significant meaning, but when interconnected, they portray a comprehensive and vivid picture of one's life. The main concern when using personal data is that of data breach. A data breach refers to any incident that results in the unauthorized loss, theft, destruction, or alteration of personal data. Unfortunately, such breaches occur frequently in modern times. As an example, one of the most famous incidents involved Equifax, a credit reporting agency in the United States, in which almost half of the population of the country had their personal information, including name, date of birth, and social security number, stolen as a result of a data breach. Another notable case involved Cambridge Analytica, a political consulting firm that surreptitiously acquired information from 50 million Facebook profiles, and subsequently provided it to the 2016 Trump campaign in the USA. These examples of incidents highlight the significant real-world repercussions of data breaches. The GDPR and similar legal frameworks from other countries seek to regulate such incidents to minimize the harm caused to individuals and organizations.

The GDPR grants internet users several new rights to protect their data privacy. The most important ones, also cited in other regulations around the world, are the right to know how exactly your data is being collected and used, the right to request information about the personal data collected about you, free of charge and the right to have your data deleted from records (in case you need to disappear). In addition, the GDPR also allows for the right to have any mistakes in your data corrected and for the right to refuse data processing, including marketing efforts. To ensure compliance with these rights and avert personal data breaches, GDPR requires each company to define a document called a *Data Protection Impact Assessment* (DPIAs). DPIAs are required for companies engaging in high-risk data processing activities that could potentially impact people's freedoms. By conducting a DPIA, a company can assess the potential risks associated with such activities and implement necessary measures to ensure the protection of personal data. When a data breach occurs, the affected company must notify their supervisory authority within 72 hours, as well as inform users in a timely manner. This ensures that users can take necessary measures to protect their personal information. Examples of such high-risk activities include the usage of new technologies for data processing, tracking the location of individuals and processing genetic or biometric data. In addition to the DPIAs, all businesses must have a *Privacy Policy* that clearly outlines their data processing activities. The privacy policy must include contact details of the company and its representatives, explanations about the reason why the company is collecting user data along with the period of time the information will be retained, a list of the rights that users have with respect to their data and also the contact details for an EU representative and Data Protection Officer.

There are two key categories of people cited in the GDPR: the *Data Controller* and the *Data Processor*. These two categories of people are very important to the understanding of this work, as they are the foundational part of the PRUDENCE method [2], our basis for calculating the privacy risk. The data controller is responsible for determining the purposes and methods of processing personal data within an organization. If an organization is responsible for deciding why and how personal data should be processed, it is considered the data controller. Employees who process personal data within an organization do so to fulfill the tasks of the data controller. An organization becomes a joint controller when it collaborates with one or more organizations to determine the why and how of personal data processing jointly. Joint controllers must enter into an agreement outlining their respective responsibilities for complying with GDPR rules. Key elements of the arrangement must be communicated to individuals whose data is being processed. The data processor is a third-party entity that processes personal data only on behalf of the controller. The processor's obligations towards the controller must be outlined in a contract or legal document. This includes the handling of personal data once the contract has been terminated. Common activities of data processors include providing IT solutions such as cloud storage. The data processor may subcontract a portion of its task to another processor or appoint a joint processor only with prior written authorization from the data controller. There may be instances where an organization serves as both a data controller and data processor.

The GDPR imposes strict penalties on individuals or organizations that violate its provisions in handling personal data. To ensure that companies comply with the legal and ethical standards of personal data processing, non-compliance can lead to severe financial penalties, including a maximum fine of up to 20 million euros or 4% of the annual global turnover of the guilty company. The GDPR became law in 2018 and since then, several well-known companies have already been penalized for non-compliance. To name a few, British Airways received a fine of 230 million dollars for exposing the booking details of 500,000 passengers in a cyber attack. Google, instead, faced a smaller penalty of 57 million dollars for withholding crucial information from users while setting up new Android phones, preventing them from knowing the nature of the data collection practices to which they were agreeing. These examples involve a great amount of money due to the severity of the damage committed but also to the size of the company, which in these cases is enormous. Although the severity of penalties may vary depending on the size of the business, all organizations are held to the same standards when it comes to complying with the GDPR.

4.2 Soft law for Artificial Intelligence in industrialized nations

It is very interesting that this need to update legal instruments has occurred almost simultaneously in most developed states. In fact, albeit with different declinations and with different social and ethical concerns, even non-democratic countries, such as the People's Republic of China, have proposed soft-law programs intending to create hard-law instruments.

In 2019, President Donald Trump signed the American AI Initiative [154], which has since been amended and renamed the National Artificial Intelligence Initiative Act, effective January 2021. The goal of the act is to promote technological advancement within the country, while maintaining high standards of quality and safety during both the production and use of AI. The act seeks to protect privacy, freedom, and civil rights as core values in the use of AI, to secure the United States' economic advantage. President Biden has called for the establishment of federal offices to enforce these laws.

Similarly, in 2018, the European Union (EU) initiated a Communication Letter to the European Parliament, titled "Artificial Intelligence for Europe"[2], to promote AI development in three key areas. The first pillar aims to encourage the use of AI in both public and private sectors to consolidate Europe as a leading technological power. The second pillar seeks to govern AI-related social and economic transformations, and the third pillar involves building ethical and legal guarantees for the development of reliable and human-centric AI. These three pillars were included in an official act called the Coordinated Plan on Artificial Intelligence[3], which has been signed by all member states, including Norway and Sweden.

It is noteworthy that the European Union has identified specific areas for technological advancement: health, mobility, security, energy and industrial production, and financial services, as stated in the Coordinated Plan on Artificial Intelligence. The EU's commitment to creating a favorable environment for the development of ethical, reliable, and human-centric AI products was expressed as early as 2018. This goal has been reiterated in subsequent official documents, including the Ethics Guidelines for Trustworthy AI produced by the High Level Expert Group. The group highlights key criteria for the development of ethical AI applications, including respect for privacy, transparency, accountability, fair behavior, and security. It emphasizes the importance of making the EU a key center for developing such technologies.

Currently, in the European context, member states have not created individual AI legislation for each state, except for the United Kingdom. Having already initiated the procedure for leaving the European Union at the time of the first official act on AI, they

[2] Link to the official communication for the Artificial Intelligence for Europe on the 25 April 2018

[3] Link to the Coordinated Plan on Artificial Intelligence

decided to proceed independently. In 2018, the UK published the AI Sector Deal[4], a policy paper outlining guidelines for AI in the UK, with particular emphasis on the search for talent and resources in this area, including from abroad. Investments and aid to companies in the sector then supported these ideas. Similarly to the European prototype, an AI Council was also established in the UK - a group of experts to help implement the above legislation. In 2021, a new document, called the National AI Strategy, was also issued, which, like the European one, clearly references a desire to be a leader in the AI field. In contrast to European ideals, however, the UK's focus is on investment, while seeking to limit inequality and respect public interest.

Looking beyond the European context, the Asian continent has also made significant strides in the field of AI. China, in particular, began considering the AI industry from a legislative standpoint as early as 2017 with the publication of the New Generation Artificial Intelligence Development Plan[5]. Like their European and American counterparts, there is a clear intention to establish China as the world leader in AI, with the creation of an expert group known as the AI Strategy Advisory Committee. The timeline for development is ambitious, with a projected value of €128 billion for the AI industry by 2030. Notably, some national companies have been identified for specific support to consolidate their position, such as Alibaba for smart city apps and Tencent for image diagnostics. Ethical values are also referenced, albeit indirectly.

Japan and South Korea have also initiated national AI development plans in the Asian context. The Japanese approach is particularly interesting[6], as their development plan focuses heavily on addressing current societal issues such as slow economic growth and an aging population. Healthcare is their primary focal point due to the aging population, followed by mobility and the industrial production of robots, in which Japan currently holds a global advantage. Like the UK, there is also a declared intention to attract talent from abroad.

Alongside these regulations, there are also various soft law documents, which are declarations of intent, developed by international organizations or groups of experts or research centers. The list of these documents is very long, but we will briefly discuss some of the most essential documents for the European context, as it is closer to our context.

In 2021, UNESCO published a document called the Recommendation on the Ethics of Artificial Intelligence which focuses on respecting the fundamental rights of human beings when interacting with AI. The document emphasizes the need for transparency, a fundamental ethical value of Explainable AI. The document mentions the need to understand the results given by AI and the possibility of human control over the behavior of AI, thus citing the concept of human-centered AI. Lastly, the document highlights the need to direct the

[4] Link to the AI Sector Deal, last update May 2019

[5]

[6] Link to the Japan plan to the development of Artificial Intelligence

development of AI towards collective well-being and environmental sustainability, moving towards a greener direction, which has not been mentioned by world powers individually.

In 2019, the G20 also published an annex on AI[7]. In this case, the world powers mainly focused on the development of a human-centered artificial intelligence, respecting equity, safety, transparency, and accountability.

Finally, one of the most important documents at the European level is the previously mentioned Ethics Guidelines for Trustworthy AI, published by the High-Level Expert Group in 2019. This document is significant because it is currently one of the most comprehensive available. It presents guidelines for achieving an AI that respects ethical principles, and seven fundamental principles are listed: human intervention and oversight, technical robustness, safety, data privacy and governance, transparency, non-discrimination and fairness, social and environmental well-being, and accountability.

4.2.1 The Artificial Intelligence Act for the EU

The last section of this chapter concerns the recent AI Act[8]. In fact, on April 21, 2021, the European Commission made public a proposal for a Regulation addressed to the European Parliament and the Council establishing rules on AI and amending certain Union legislative acts. In particular, at the beginning of this regulation, several laws are mentioned that legitimize its existence, in particular Article 114 TFEU, regarding the functioning of the internal market, as well as Article 16 of the same treaty, regarding the personal data of European citizens.

In addition to these legal statements, it is interesting to note that this regulation is actually the result of a very extensive work that started more than three years before its publication, which involved many AI experts, including researchers, industrialists, and not just political figures.

The project under analysis aims to address the dangers connected to some AI applications, without forgetting to promote the development and diffusion of this technology, without limiting the market. Given these premises, the Commission has chosen to propose a risk-based approach. This approach had already been selected in other contexts by the European Union, including the GDPR. Therefore, having chosen a risk-based approach, the Proposal divides AI applications into four risk classes, subjecting each risk class to a different regulatory regime. In the various annexes published together with the regulation, we find lists of technologies prohibited within the European Union, due to the potential harm to human dignity, as well as to a wide range of individual rights. In this list, we find, for example, social credit rating systems that have prejudicial consequences for the subjects involved in contexts not connected to that in which the starting data was collected, as well as systems that exploit the person's vulnerabilities.

[7] G20 annex

[8] Link to the AI act

Among the various systems considered high risk and therefore prohibited, we also find real-time biometric identification systems. However, unlike others, these systems have been deemed important for security purposes. Therefore, they are allowed only in missing persons cases, to prevent terrorist attacks, or to pursue suspects of serious crimes.

Apart from this aspect of prohibited systems, there are also listed high-risk systems. In this context, it is not easy to understand exactly what the high-risk systems are. Certainly, those listed in Annex III are considered high-risk, but not only: there are several annexes, not easy to understand, that delimit the risk perimeter through different annexes.

Examples of high-risk systems are those used in credit activity and for managing migratory flows. For these high-risk systems, there are stringent rules to be followed. For example, there are requirements in terms of the quality of the datasets used, transparency of the decision-making process applied, comprehensibility of the results, as well as the possibility of human control and intervention in case of doubts about the behavior of the AI under analysis. Given all these requirements, the Proposal of the AI Act cites the need to create a procedural and documentary risk management system. This process is delegated to the technology supplier company under analysis. Once this documentary process is drafted, the market entry of a high-risk AI product will not be easy, as the product must undergo a procedure that evaluates its compliance with the requirements. In case of a positive outcome, the CE mark will be placed.

The other category of risk for AI systems is the so-called non-risk category. In this case, the Proposal still recommends evaluating transparency, data origin, and all the analyses required in case of high risk, but in this context, they are not binding, and no conformity evaluation is necessary for placing the product on the market.

Finally, the Proposal also analyzes AI that interacts with humans. Examples of these products are so-called Deep fakes, where AI is capable of generating and manipulating extremely realistic content, as well as products capable of detecting user emotions or profiling users based on their biometric data. In these cases, regardless of the risk the product poses, specific transparency obligations are required: these products must inform the user who is using them that they are interacting with an intelligent agent and not a human being, as well as an explicit reference to what the AI's goals are in that context, such as emotional or biometric profiling.

All these constraints and precautions that need to be taken not only for banned and high-risk AI but also for non-risk AI, if not met, will result in sanctions. At present, as this is only a proposal, figures have been proposed for sanctioning non-compliance with the law, with higher sanctions in the case of prohibited AI. However, these figures could still change. Moreover, no sanctions have been chosen for each non-compliance, leaving the choice of appropriate sanctions for these cases to the Member States.

From a regulatory point of view, the AI Act is a Proposal and is therefore not yet in force. The Proposal was presented on April 21, 2021, and since then, a public consultation procedure lasting 3 months has begun, followed by a period of re-discussion of the proposed text. Following this, there is a complex legislative process before the Proposal becomes law. Therefore, it is still not possible to know whether the text will actually become law and with what modifications, or even when this will happen, if it happens. Nevertheless, most parliamentarians are confident in the need for a legislative act regarding the risk assessment of products containing AI. Even in the case of a positive outcome, once the law is approved, there will surely be a transition period to allow sector operators to comply with the requirements.

In conclusion, the current state of AI law consists of proposals, intentions, and perspectives, but not binding regulations. This perspective is not only European but global, as most developed countries have expressed strategic development plans at both national and supranational levels, but none of these plans are binding or enforceable in court. In practice, there are few disciplines of hard law currently in effect. However, this trend is understandable when we consider the fact that AI is a rapidly developing technology that lacks any form of regulation. Thus, it is necessary to start with broad ethical principles before delving into specific applications.

Despite the absence of hard law, several states are working on creating specific regulations for AI, the most significant of which is the European AI Act. The importance of the AI Act is not due to being in Europe but to what is commonly known as the "Brussels effect." This effect has been seen in the past, specifically with the GDPR, which came into effect in May 2018 and quickly became a world standard. The AI Act appears to be going in the same direction, with a clear extraterritorial scope that applies to any AI system or service that impacts European citizens, regardless of where its provider or user is located. The AI Act adopts a risk-based approach that bans certain technologies, proposes strict regulations for "high-risk" ones, and imposes stringent transparency criteria for others. If adopted, the AI Act will undoubtedly significantly impact the EU and beyond. A crucial question is whether we have the technology to comply with the proposed regulation and to what extent the requirements of this regulation can be enforced.

Additionally, there is a desire to steer technological development towards user-centered AI without stopping the market and technological process.

4.3 Discussion

This Chapter analyzes the legal aspects of Privacy and Explainable AI. The increasing integration of AI in various aspects of our daily lives necessitates ethical discussions and regulatory measures to ensure responsible usage. The Chapter emphasizes the importance of evaluating ethical aspects and potential regulations due to AI's pervasive presence in technologies like interactive maps, digital assistants, biometric systems, and recommendation algorithms.

Data privacy emerges as a crucial ethical concern, as AI systems often rely on sensitive user data for training. However, the widespread collection, storage, and exchange of data present challenges in safeguarding personal information. High-profile incidents, such as the Cambridge Analytica scandal and Uber's data breach, highlight the inadvertent disclosure of personal data and the risks associated with it. The significance of data privacy in AI applications is further exemplified by the European Union's introduction of the General Data Protection Regulation (GDPR) in 2018, which establishes rules for companies' use and management of personal data.

The Chapter also addresses the implications of data breaches, which involve the unauthorized loss, theft, or alteration of personal data. Several high-profile data breaches, including Equifax and Cambridge Analytica, underscore the real-world consequences of such incidents. The GDPR and similar legal frameworks aim to regulate data breaches and minimize the harm caused to individuals and organizations.

Overall, the Chapter presents a small overview of the legal regulations for privacy and explainability in AI. It emphasizes the need for responsible usage, legislation, and regulatory measures to safeguard personal data, mitigate algorithmic bias, and prevent data breaches in an increasingly AI-driven world.

Part II

Local Post-hoc Explanations for Tabular Data

This Part of the Thesis deals with the topic of Post-hoc Explanations for tabular data, explained locally. The field of Explainable Artificial Intelligence can be divided into two main categories: *explainable by design* methods, in which the explanation is intrinsic, part of the Machine Learning model, or *post-hoc* methods, in which the objective is to provide an explanation for a Machine Learning model not interpretable and already trained, hence from the outside of the model. We focus our work on the latter kind of Explainable methods. In this setting, the methods can be of two different kinds: *local*, hence focusing on explaining the reasons for the particular prediction for a single record, of *global*, explaining the overall behaviour of the Machine Learning model. In this Part, we focus on the local explanations.

This Part starts with a benchmark of the local post-hoc explanation methods available in the literature, presented in Chapter 5. The analysis presented in this Chapter is published in [11] and in [155]. In particular, we focus our analysis on Shap, Lore, Lime and Dalex, among the most popular xai methods with available Python libraries. To compare these methods' explanations output, we exploit the state-of-the-art metrics for this particular setting, such as *fidelity* and *insertion and deletion* methods. From the analysis conducted, it is evident that the explanation techniques developed so far suffer from numerous limitations. First, most of these methods exploit approximation or randomization techniques, making the generated explanations unstable: given an input record multiple times to the same methods, we never receive the same explanation in the output. This is a considerable limitation for us as it limits the system's reliability and thus may lead users to have less confidence in the AI system. In addition to this limitation, there is the problem that most of these methods provide explanations that are not faithful to the black-box under analysis. This result is evident from the values of metrics such as insertion and deletion, in which variables considered most important are removed from the record under analysis to see how the black-box prediction changes. Lastly, the explanations currently provided are complex for a non-expert user to understand, as they are composed of multiple different indices and codifications, which makes the process of understanding the explanation difficult.

To overcome these limitations found during the benchmark, we designed and developed a new version of Lore. Lore is a local explanation algorithm specific for tabular data, which returns factual rules and counterfactual rules. It is based on a genetic algorithm, which is exploited to generate a synthetic neighborhood around the point under analysis. After creating this neighborhood, Lore extracts a surrogate model, from which it later extracts the explanation. To overcome the above limitations, we generate multiple neighborhoods, extract a surrogate model for each of them, and then merge them through a procedure consisting of filters and pruning, allowing the creation of a more stable surrogate. In addition, we allow the creation of constraints on the variables to have actionable counterfactual rule explanations. The new version of Lore is presented in Chapter 6 and this work has also been published in [13].

Chapter 5

Benchmark of Local Post-hoc explanations

In this chapter we present a technical analysis of state-of-the-art methods in the context of post-hoc and local explanations for tabular data. As presented in the chapter 2.1, the field of Explainable AI and in particular of post-hoc local explanations, is a remarkably flourishing field at this time and thus several papers have been published, although most of them are only presented in a theoretical way and do not have a usable library. Therefore, in this chapter the methods under analysis are the state-of-the-art ones that are also provided with a usable and maintained library. We focus on local explanations, in which given an input record the explainer provides a specific explanation, which offers reasons for the classification in analysis. In addition, these explanations are post-hoc, in that we assume that the classifier has already been trained and is working, so the explainer performs its work only by query functions, such as *predict* and *predict proba* (the former returns the predicted class, while the latter returns the probability vector, in which the membership probabilities of each class are present). Ultimately, the focus in this case is on tabular data. This choice is due to the fact that tabular data is the most widely used data in the field of Data Mining and Data Privacy, and for this reason, research on this type of data was the first to be tackled in XAI as well.

The field of validating the goodness of an explanation is still an open challenge due to the plurality of explanations available and the different ways in which these explanations are extracted. This field is composed of two main kinds of explanation validation: quantitative, in which the focus is on the technicality of the explanations, in particular on the performance of the explainer, and qualitative, in which users are tested for empirically validating the comprehension of the explanations. For the time being, in the state of the art, the two most widely used metrics for validating explanations for tabular data are *fidelity*, *stability*, *faithfulness* and *monotonicity*. In the following we remark the general behavior and objectives of these metrics, however, an in-depth description of these metrics can be found in Section 2.4.

5.1 Dataset

For the tabular data we consider three benchmark datasets: all of them have different characteristics that may affect the performance of the explanation methods. For all of them, we apply a standard pre-process: we replaced the categorical variables using a TargetEncoder, we replaced the missing values using the mean (of median) of the column under analysis, and we removed the outliers by visualizing the statistical distribution of the variables. We analyzed ADULT[1]: a binary classification with the task of predicting if a person earns more or less than 50K per year. It has 14 attributes (numerical and categorical) and 48842 records. Then, we considered GERMAN[2]: a binary classification for predicting the credit risk of a person. It has 20 attributes, mostly categorical, with 1000 records. Lastly, COMPASM[3]: a multi-class dataset, in which the goal is to predict the recidivism of a convicted person, with 3 classes of risk recidivism. It has 21800 records and 10 variables, all of them categorical except *age*.

5.2 Black-box models

For comparing the explanations, we define and train three ML models for each dataset:

- *Logistic Regression* (LG), a simple model based on probabilities, but prone to noise and overfitting,
- *XGBoost*[4] (XGB), a ensemble model, with overall better prediction performance with respect to LG,
- *CatBoost*[5] (CATBOOST), a variant of gradient boosting algorithm tailored for handling missing values and categorical data.

The performance of the black-box models are reported in Table 5.1. In the table is reported the weighted $F1$ score. This score is calculated by taking the mean of all per-class $F1$ scores while considering each class's support. In this way we take into account the imbalance among the classes. From this table we can clearly see that all the ML models perform well, with a slightly lower $F - 1$ score for the LG model for all the datasets. This behaviour was

[1] ADULT: `https://archive.ics.uci.edu/ml/datasets/adult`

[2] GERMAN: `https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data)`

[3] COMPASM: `https://www.kaggle.com/datasets/danofer/compass`

[4] https://xgboost.readthedocs.io/en/stable/

[5] https://catboost.ai/

	Adult			German			CompasM		
black-box	LG	XGB	CAT	LG	XGB	CAT	LG	XGB	CAT
F1-score	0.65	0.82	0.80	0.66	0.75	0.79	0.63	0.69	0.68

Table 5.1: Weighted F1 score for the various black-boxes and datasets.

Dataset	**Black-Box**	**Fidelity**					**Faithfulness**		
		Lime	Shap	Dalex	Anchor	Lore	Lime	Shap	Dalex
Adult	LG	0.98 (0.21)	0.61 (0.43)	0.35 (0.03)	**0.99** (0.05)	0.98 (0.03)	0.10 (0.30)	**0.38** (0.37)	0.08 (0.03)
	XGB	**0.98** (0.03)	0.88 (0.02)	0.64 (0.07)	**0.98** (0.03)	**0.98** (0.04)	0.03 (0.32)	**0.36** (0.49)	0.27 (0.31)
	CAT	0.96 (0.32)	0.78 (0.51)	0.70 (0.15)	**0.99** (0.21)	0.98 (0.43)	0.10 (0.32)	**0.44** (0.37)	0.11 (0.30)
German	LG	**0.98** (0.06)	0.91 (0.23)	0.57 (0.21)	0.73 (0.09)	**0.98** (0.12)	**0.23** (0.60)	0.19 (0.63)	0.20 (0.03)
	XGB	**0.99** (0.10)	0.82 (0.02)	0.65 (0.03)	0.80 (0.03)	0.98 (0.21)	0.16 (0.26)	**0.44** (0.21)	0.31 (0.09)
	CAT	**0.98** (0.05)	0.67 (0.12)	0.63 (0.09)	0.62 (0.31)	**0.98** (0.35)	0.34 (0.33)	**0.43** (0.32)	0.33 (0.12)
CompasM	LG	**0.95** (0.31)	0.83 (0.41)	0.23 (0.03)	0.53 (0.46)	0.82 (0.03)	0.12 (0.56)	**0.41** (0.54)	0.11 (0.08)
	XGB	**0.97** (0.21)	0.43 (0.33)	0.45 (0.23)	0.67 (0.42)	0.87 (0.03)	0.19 (0.44)	**0.56** (0.38)	0.13 (0.13)
	CAT	**0.98** (0.27)	0.54 (0.10)	0.55 (0.30)	0.22 (0.92)	0.81 (0.02)	0.22 (0.42)	**0.57** (0.32)	0.18 (0.07)

Table 5.2: Comparison on fidelity and faithfulness of the explanation methods. We report the mean and the standard deviation over a subset of 50 test set records. In bold are reported the best results obtained, while we underline the second best result for each metric.

expected since the LG model is the most simple model among the one selected. For the German dataset we can see that the CatBoost is the model performing better. This is due to the great number of categorical variable that are present in this dataset: CatBoost is tailored for handling them, while the other models may suffer from the presence of this great amount of categorical data. Lastly, the overall performance of the ML models for the CompasM dataset are lower w.r.t. the other two datasets. This is due to the fact that this dataset is a multi-class classification, making it more difficult to achieve higher performance. For these experiments the datasets were split into train and test with ratio $80\% - 20\%$.

5.3 Explanation methods

For validating the local and post-hoc explanations on tabular data, we refer to five explanation methods, already presented in Section 2.2, all with a working library available in Python. We consider Lime, one of the first local post-hoc explanation methods that outputs feature importance. The process of Lime starts with a synthetic generation of records, used as a training set for training a surrogate model tailored for the locality of the record under analysis, easier to interpret w.r.t. the black-box model we aim at explaining. Practically, for creating and running the Lime explainer we selected the Gaussian random sampling

Dataset	Black-Box	Stability				
		Lime	Shap	Dalex	Anchor	Lore
Adult	LG	24.37 (2.74)	1.52 (4.49)	5.40 (0.10)	**22.36** (8.37)	21.76 (11.80)
	XGB	10.16 (6.48)	2.17 (2.18)	6.00 (0.06)	26.53 (13.08)	**30.01** (20.52)
	CAT	0.35 (0.43)	0.03 (0.01)	4.3 (0.04)	6.51 (4.40)	**27.80** (70.05)
German	LG	18.8 (0.73)	19.01 (23.4)	12.54 (0.05)	101.0 (62.7)	**622.1** (256.7)
	XGB	26.08 (14.5)	38.43 (30.6)	5.12 (0.10)	121.4 (98.4)	**725.8** (337.2)
	CAT	2.49 (9.91)	15.92 (10.71)	3.54 (0.9)	123.7 (76.86)	**756.7** (348.2)
CompasM	LG	0.51 (0.21)	0.54 (0.10)	11.42 (19.24)	112 (23.52)	**321.3** (261.4)
	XGB	0.676 (0.30)	13.67 (21.64)	6.00 (0.06)	97.20 (18.04)	**229.1** (39.61)
	CAT	2.49 (9.91)	14.22 (10.01)	4.33 (0.04)	100.7 (60.60)	**526.9** (341.5)

Table 5.3: Comparison on the stability metric. We report the mean and the standard deviation over a subset of 50 test records.

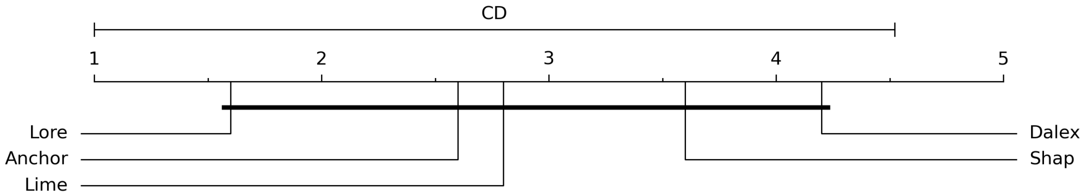


Figure 5.1: Critical difference plot for Nemenyi test ($\alpha = 0.05$). We compare the tabular explanations in terms of fidelity and stability computable for all the explanation kinds.

generation, with 5000 synthetic samples to generate for each record to explain. In the context of feature importance, we also consider Shap, one of the most popular explainers. The process of Shap is based on game theory, in which each variable is associated with an importance value, called Shap *values*. To calculate the Shap values, e.g. the importance of the variables, the approach follow the structure of a game, in which every variable is a player. The game is played with or without the variable to understand the changes in the prediction outcome. In practice, there are several implementations to extract Shap values, depending on the kind of black-box to explain as well as the kind of approximation we are aiming at. For the experiments of this Chapter we considered the LinearExplainer, an explainer able to exactly calculate the Shap values for linear models. We applied it for the LogisticRegression. Then, we considered the TreeExplainer, another exact computation of Shap values for tree-based models, to explain XGBoost and KernelExplainer for CatBoost. Then, we consider Dalex, another feature importance method that allows for the calculation of the feature importance in several ways. For these experiments we select the *break down*. We also consider Anchor and Lore, which outputs rules (also counterfactual rules for the latter). ANCHOR perturbs the instance x obtaining a set of synthetic records employed to extract anchors with precision above a user-defined threshold. The anchors are rules with a particular property: for decisions on which the anchor holds, changes in the rest of the instance's feature values do not change the outcome. In this setting, we require a coverage of 0.95 for ANCHOR. Regarding LORE, instead, the process is similar to the one of LIME, with the exception that the synthetic generation is based on a genetic algorithm, hence with a focus on the locality of the record under analysis. For the application of LORE we select the *genetic and random* synthetic generation, with 4000 records to create.

5.4 Metrics for validating an explanation

To evaluate the goodness of the explanations extracted with the different methods available we refer to some quantitative metrics. As discussed in Section 2.4, the objective of evaluating the goodness of post-hoc explanations is still an open challenge, but there are two main distinctions: quantitative and qualitative measures. For the first one the focus is on the technicality of the explanations, in particular on the performance of the explainer, while the qualitative metrics, refer to users for empirically validating the comprehension of the explanations. In this work, we refer to quantitative measures. In the following we briefly describe all the metrics used in this benchmark.

For tabular data, one of the metric most used is the *fidelity*: the objective of this metric is to measure how good the explanation method is at mimicking the black-box decisions. In other words, it analyzes the completeness of the explanation method $\mathcal{E}$ w.r.t. the black-box model b. To validate the fidelity, there is no a single formula in the literature. In fact, depending on the kind of explanation method considered, the evaluation of fidelity may have a different specialization [50]. In methods where there is a creation of a surrogate model g to mimic b, such as LIME, the fidelity is computed with the accuracy of the predictions of g w.r.t. b on the instances used to train g [50]. For methods without a surrogate model, a very simple model can be created using the explanation and then the fidelity is computed as the accuracy of such model on the prediction of the black-box. The closer to one, the better.

Another measure we considered is the *stability*: it aims at validating how stable the explanations are for similar records. The main idea is that, if we have two similar records, also the explanations should be close. To calculate this metric the *Lipschitz constant* [96] is exploited: given a record to explain x and a neighborhood $\mathcal{N}_x$ and x' composed of instances similar to x, the explanation method E provides explanations e_x and $e_{x'}$ and the stability is computed: $L_x = \max\frac{\|e_x - e_{x'}\|}{\|x - x'\|}, \forall x' \in \mathcal{N}_x$. Intuitively, the higher the value, the better is the model to present similar explanations for similar inputs.

Other metrics have been proposed [99] with the aim of validating the goodness of explanations by changing the input record, depending on the explanations. The idea is that it is possible to validate the correctness of explanations by removing (in order of importance) the features that the explanation method considers important. By doing so, if the explanation is faithful, then the performance of the black-box b should degrades. The intuition behind this deletion methods is that removing the "cause" will force the black-box to change its decision. In this work, we consider the *faithfulness* [96], which aims at validating whether the importance scores obtained from the explanation method indicate true importance. An implementation of the faithfulness metric exploited in these experiments is available in AIX360. Mathematically, given a black-box b and the feature importance e extracted from an explanation method, the faithfulness removes attributes in order of importance given by e. At each removal, the effect on the performance of b is evaluated and these values are then employed to compute the overall correlation between feature importance and model performance. It results in a value range $[-1, 1]$: the higher the value, the better the faithfulness.

We also consider *monotonicity* that takes the complementary approach w.r.t. *faithfulness*. It evaluates the effect of b by incrementally adding each attribute in order of increasing importance. In an opposite way than before, we expect that the black-box performance increases by adding more and more features, thereby resulting in monotonically increasing model performance[6]. Beside these metrics, during the comparison of different explanation methods, standard metrics like *accuracy*, *precision* and *recall* are also evaluated, as well as the *running time*.

5.5 Discussion of the results

The results obtained from the applications of the metrics proposed in Section 5.4 are reported in Table 5.2 for the fidelity and faithfulness, while in Table 5.3 we report the stability. The monotonicity is not reported since for every method it was *False*, showing that no method is compliant with this requirement. To obtain this results we explained 50 records from the test set of each dataset. With respect to the fidelity, the best results are obtained by ANCHOR, followed by LORE and LIME. SHAP and DALEX, instead, achieved the worst results, with particularly low values for DALEX. This behaviour might be due to the different process for constructing an explanations: while LORE and LIME have a similar approach of constructing a synthetic neighbourhood around the record under analysis, while SHAP and DALEX do not follow this approach. Overall we can see that our experiments show that rule-based methods have very high fidelity, correctly replicating the black-box behavior. Regarding the faithfulness, instead, all the models do not reach optimality. Among the methods under analysis, SHAP achieves the best performance, being the metrics with values between -1 and 1: the intuition is that the most important features for SHAP are actually important also for the black-box model under analysis. However, since the results

[6] An implementation of monotonicity and faithfulness is available in AIX360

are lower that 50% in all cases except for the CompasM with the xgb, this behaviour is not happening for all the records under analysis. Nevertheless, Shap turns out to be the best in this context, followed by Dalex and Lime. In terms of stability, Lore is the best method, followed by Anchor. Again, the rule-based models seems to be the one performing better with respect to the metrics considered. However, we can clearly see that the standard deviation for all these experiments is quite high, highlighting that the models are suffering from instability.

In Figure 5.1, we report an overall ranking evaluation of the explanation methods in terms of *fidelity* and *stability*. From this plot, we can clearly see that Lore and Anchor, which are the rule-based methods, perform better than the feature importance ones considering both the metrics. This result is particularly interesting because feature importance methods are more studied than logical explanations even though the latter are more similar to human thinking. [155].

This fact is also highlighted by the results on stability, that are good for Lore, even if not perfect, followed by Anchor. Regarding the feature importance methods, Lime also has excellent fidelity, but unfortunately this method suffers in terms of stability due to its random generation of the neighborhood. Shap and Dalex, instead, do not exhibit a good fidelity but are better in terms of stability w.r.t. Lime.

From the empirical analysis conducted in this Chapter, it is clear that the various methods proposed in the literature still suffer from several limitations. First, in terms of faithfulness no method performed well, as well as in terms of stability, although in this case the rule-based methods are better than those based on feature importance. In addition, the feature importance based methods propose explanations more difficult to understand for a non-expert user, while the rule-based ones are easier to understand thanks to the logic nature of the rules.

5.6 Discussion

In this Chapter, we introduced a benchmark for explanation methods for tabular data, exploiting metrics from existing literature to facilitate quantitative comparisons among different explanation methods. In particular, we considered feature and rule based explanation methods, which are among the most popular methods for tabular data.

The quantitative analysis results indicate that rule-based explanation methods exhibit superior performance for tabular data, demonstrating higher fidelity and stability with respect to the feature importance methods. The rule-based methods provide explanations that accurately reflect the decision-making processes of black-box models.

Overall, no single method emerged as the dominant choice, underscoring the challenge of simultaneously achieving effectiveness and robustness in generating explanations.

As future work, we aim to address the limitations of the presented methods by proposing a novel methodology focused on producing stable explanations. We strive to ensure that given the same input multiple times, consistent explanations are obtained.

Chapter 6

Stable and Actionable Local Explanations

One of the primary challenges for Artificial Intelligence (AI) applications is to provide meaningful and stable explanations for the decisions made by black-box classifiers, according to previous studies [31, 156] and in agreement with the benchmark presented in Section 5, in which the critical issues of the methods in the literature are highlighted, including limitations in terms of stability. Despite the requirement by regulators that automated decisions should be explained, new algorithms for decision-making continue to be developed, resulting in a lack of transparency in ML models that can perpetuate or reinforce forms of injustice by learning biased habits from the data. For this reason, a variety of proposals for explaining classification models have emerged, ranging from global approaches to local and model-agnostic to model-specific approaches.

The aim of this research is to explain the decisions made by black-box classifiers on specific input instances by providing meaningful and stable explanations of the involved logic. The objective is to achieve a model-agnostic method that works by analyzing the input-output behavior of the black-box locally, in the instance's neighborhood to explain. The research is based on several assumptions: (i) the language used to offer explanations should be as close as possible to a formal reasoning language such as propositional logic, and the user should be able to understand the semantics of elementary logic rules taught in secondary schools or undergraduate courses; (ii) explanations are interesting if they answer both the factual and counterfactual questions about why a specific decision was made and what conditions would change the black-box decision; and (iii) the black-box system can be queried multiple times to reconstruct its logic completely.

Using logic rules is a step toward making explanations comprehensible, but it is not enough to achieve meaningful explanations. The reconstruction logic of the black-box in the neighborhood of the instance to explain should be consistent with the black-box decisions, a property known as fidelity. Additionally, the counterfactual answer should consist of a minimal number of changes to the feature values of the instance to explain (minimality), and such changes should allow for actionable recourse, a property known as actionability. The approach used for generating explanations should also guarantee stability of its output against local perturbations of the input, and it should be general enough to encompass not only tabular data but also images, texts, and multi-label data.

This research aims to advance state-of-the-art approaches, including previous work such as Factual Local Rule-based Explanations (FLore) [50], by proposing a comprehensive method that extends the coverage of comprehensibility, fidelity, minimality, and generality to also include stability and actionability. The proposed method is Stable and Actionable Local Rule-based Explanations (LORE$_{sa}$), which builds a simple, interpretable local decision tree predictor from an ensemble of balanced sets of neighbor instances generated through a genetic algorithm. Each set is used to extract a decision tree classifier, which is then merged into a single decision tree classifier. LORE$_{sa}$ provides a (counter)factual explanation from the decision tree that approximates the behavior of the black-box around the instance being explained. The (counter)factual explanation is a pair composed of the factual rules that characterize the conditions for a specific black-box decision and the counterfactual rules that indicate the minimal number of changes required in the feature values to change the black-box decision.

6.1 Problem Formulation and Explanation Definition

We first set the basic notation for classification models. Afterwards, we define the *black-box outcome explanation problem*, and the notion of *explanation* that our method will be able to provide.

A classifier is a function $b : \mathcal{X}^{(m)} \to \mathcal{Y}$ which maps data instances (tuples) x from a feature space $\mathcal{X}^{(m)}$ with m input features to a decision y in a target space $\mathcal{Y}$ of size $L = |\mathcal{Y}|$, as presented in Chapter 14.2. We write $b(x) = y$ to denote the decision y taken by b, and $b(X) = Y$ as a shorthand for $\{b(x) \mid x \in X\} = Y$. If b is a probabilistic classifier, we denote with $b_p(x)$ the vector of probabilities for the different labels. The domain of a feature can be continuous or categorical. In this case, we assume that a predictor is available as a function that can be queried at will. In the following, b will be a *black-box* predictor, whose internals are either unknown to the observer or they are known but uninterpretable by humans. Examples include neural networks, SVMs, ensemble classifiers [23, 24]. Instead, we denote with c an *interpretable* (comprehensible) predictor, whose internal processing leading to a decision $c(x) = y$ can be given a symbolic interpretation which is understandable by a human. Examples of such predictors include rule-based classifiers, decision trees, decision sets, and rational functions [23, 24], as presented in Section 14.2.1.

Given a black-box b and an instance x, the *black-box outcome explanation problem* consists of providing an explanation e for the decision $b(x) = y$. We approach the problem by learning an interpretable predictor c that reproduces and accurately mimes the *local* behavior of the black-box. An explanation of the decision is then derived from c. By *local*, we mean focusing on the behavior of the black-box in the neighborhood of the specific instance x, without aiming at providing a description of the logic of the black-box for all possible instances. The neighborhood of x is not given, but rather it has to be generated as part of the explanation process. However, we assume that some knowledge is available about the characteristics of the feature space $\mathcal{X}^{(m)}$, in particular the ranges of admissible values for the domains of features and, possibly, the (empirical) distribution of features. Nothing is instead assumed about the training data/process of the black-box.

Definition 1 (Black-Box Outcome Explanation) *Let b be a black-box, and x an instance whose decision $y = b(x)$ has to be explained. The* black-box outcome explanation problem *consists of finding an explanation $e \in E$ belonging to a human-interpretable domain E.*

Interpretable predictors are specific of the black-box and of the instance to explain and they must agree with the black-box decision.

Definition 2 (Explanation through Interpretable Model) *Let $c = \zeta(b, x)$ be an interpretable predictor derived from the black-box b and the instance x using some procedure ζ, and s.t. $c(x) = b(x)$. An explanation $e \in E$ is obtained through c, if $e = \varepsilon(c, x)$ for some explanation logic ε over c and x.*

These definitions are parametric in the domain E of explanations, which has to be instantiated. We define it by adopting a combination of factual and counterfactual rules. Formally, we define an explanation e as:

$$e = \langle r = p \rightarrow y, \Phi \rangle$$

The first component $r = p \rightarrow y$ is a *factual* decision rule describing the reason for the decision value $y = b(x) = c(x)$. The second component Φ is a set of *counterfactual* rules, namely rules describing a (minimal) number of changes in the feature values of x that would change the decision of the predictor to $y' \neq y$. As an example, the following is an explanation for the decision to reject the loan application of instance $x_0 = \{\mathit{age} = 22, \mathit{sex} = \mathit{male}, \mathit{income} = \mathit{800}, \mathit{car} = \mathit{no}\}$:

$$\begin{aligned} e{=}\langle r{=}\{\mathit{age}{\leq}25, \mathit{sex}{=}\mathit{male}, \mathit{income}{\leq}\mathit{900}\}{\rightarrow}\mathit{deny}, \\ \Phi{=}\{\{\mathit{income}{>}900\}{\rightarrow}\mathit{grant}, \{\mathit{sex}{=}\mathit{female}\}{\rightarrow}\mathit{grant}\}\rangle \end{aligned}$$

In this example, the decision *deny* is due to the age lower or equal than 25, the sex that is male, and an income lower or equal than 900 (see component r). In order to obtain a different decision, the applicant should have a greater income, or be a female (see component Φ).

In a *factual rule* r of the form $p \rightarrow y$, the decision y is the *consequence* of the rule, while the *premise* p is a boolean condition on feature values. We assume that p is a conjunction of *split conditions* of the form $a_i {\in} [v_i^{(l)}, v_i^{(u)}]$, where a_i is a feature and $v_i^{(l)}, v_i^{(u)}$ are lower and upper bound values in the domain of a_i extended with[1] $\pm\infty$. An instance x *satisfies* r, or r *covers* x, if the boolean condition p evaluates to true for x, i.e., if $sc(x)$ is true for every $sc \in p$. The rule r in the example above is satisfied by x_0, and not satisfied by $x_1 {=} \{age{=}22, sex{=}male, income{=}1000, car{=}no\}$. We say that r *is consistent* with the interpretable predictor c, if $c(x) = y$ for every instance x that satisfies r. Consistency means that the rule provides a sufficient condition for which the predictor outputs y. If the instance x to explain satisfies p, the rule $p \rightarrow y$ represents then a candidate explanation of the decision $c(x) = y$. Moreover, if the interpretable predictor mimics the behavior of the black-box in the neighborhood of x, we further conclude that the rule is a candidate local explanation of $b(x) = c(x) = y$.

Consider now a set δ of split conditions. We denote the update of p by δ as $p[\delta] = \delta \cup \{(a \in [v_i^{(l)}, v_i^{(u)}]) \in p \mid \nexists w_i^{(l)}, w_i^{(u)}.(a \in [w_i^{(l)}, w_i^{(u)}]) \in \delta\}$. Intuitively, $p[\delta]$ is the logical condition p with ranges for attributes overwritten as stated in δ, e.g., $\{age{\leq}25, sex{=}male\}[age{>}25]$ is $\{age{>}25, sex{=}male\}$. A *counterfactual rule* for p is a rule of the form $p[\delta] \rightarrow y'$, for $y' \neq y$. We call δ a *counterfactual.* Consistency w.r.t. c is meaningful also for counterfactual rules, denoting now a sufficient condition for a reverse decision y' of the predictor c. A counterfactual δ describes *which* features to change and *how* to change them to get an outcome different from y. Continuing the loan example, changing the income to any value > 900 will change the predicted outcome of b from *deny* to *grant*. A desirable property of a consistent counterfactual rule $p[\delta] \rightarrow y'$ is that it should be *minimal* [67, 158] with respect to x. Minimality can be measured (see [50]) with respect to the number of split conditions in $p[\delta]$ not satisfied by x. Formally, we define $nf(p[\delta], x) = |\{sc \in p[\delta] \mid \neg sc(x)\}|$ (where $nf(\cdot, \cdot)$ stands for the number of falsified split conditions[2]). In the loan example, $\{age{>}25, income{>}1500\} \rightarrow grant$ is a counterfactual with two conditions falsified. It is not minimal as the counterfactual $r = \{age{\leq}25, sex{=}male, income{>}900\} \rightarrow grant$ has only one falsified condition. In summary, a counterfactual rule $p[\delta] \rightarrow y'$ is a (minimal) *motivation for reversing* the decision outcome of the predictor b.

In this work, we add to the properties of consistency and minimality of counterfactual rules, the one of *actionability* (also called *feasibility*), which is intended to prevent generating invalid or unrealistic rules. E.g., a counterfactual split condition $age \leq 25$ is not actionable

[1] Using $\pm\infty$ we can model with a single notation typical univariate split conditions, i.e., equality ($a = v$ as $a \in [v, v]$), upper bounds ($a \leq v$ as $a \in [-\infty, v]$), strict lower bounds ($a > v$ as $a \in [v + \epsilon, \infty]$ for a sufficiently small ϵ). However, since our method is parametric to a decision tree induction algorithm, split conditions can also be multivariate, e.g, $a \leq b + v$ for a, b features (as in oblique decision trees [157]).

[2] When clear we write nf as shorthand of $nf(p[\delta], x)$.

Algorithm 1: LORE$_{sa}$ (x, b, K, U)

```
Input  : $x$ - instance to explain, $b$ - black-box, $K$ knowledge, $U$ constr.
Output: $e$ - (counter)factual explanation of $x$
$\mathcal{D} \leftarrow \emptyset$;                                          // init.  empty set of decision trees
for $i \in \{1, \ldots, N\}$ do
    $Z_{=}^{(i)} \leftarrow genetic(x, \mathit{fitness}_{=}^{x}, b, K)$;            // neighborhood generation
    $Z_{\neq}^{(i)} \leftarrow genetic(x, \mathit{fitness}_{\neq}^{x}, b, K)$;           // neighborhood generation
    $Z^{(i)} \leftarrow Z_{=} \cup Z_{\neq}$;                                   // merge neighborhoods
    $Y^{(i)} \leftarrow b(Z^{(i)})$;                                        // apply black-box
    $d^{(i)} \leftarrow buildDecisionTree(Z^{(i)}, Y^{(i)})$;                    //build decision tree
    $\mathcal{D} \leftarrow D \cup \{d^{(i)}\}$;                               // add decision tree to list
$c \leftarrow mergeDecisionTrees(\mathcal{D})$;                               // merge decision trees
$r = (p \rightarrow y) \leftarrow extractDecisionRule(c, x)$;                          // factual rule
$\Phi \leftarrow extractCounterfactuals(c, r, x, U)$;                        // extract counterfactual
return $e \leftarrow \langle r, \Phi \rangle$;
```

for a loan applicant of age 30 because she cannot change her age. Formally, we assume a set U of *constraints on features* of the form: $a = x[a]$, meaning that the attribute a cannot be changed (e.g., $age = 30$ or $sex = male$); or, $a \leq x[a]$ (resp., $a \geq x[a]$), meaning that the attribute a cannot be increased (resp., decreased). Actionability requires that the premise $p[\delta]$ of a counterfactual rule must satisfy the conditions specified in U, i.e., $p[\delta] \rightarrow U|_{p[\delta]}$ is a true formula, where $U|_{p[\delta]}$ are the constraints in U involving attributes occurring in $p[\delta]$. Going back to our example if $U = \{age = 22\}$, then the counterfactual $\{age{>}25, income{>}1500\} \rightarrow grant$ is not actionable.

We can now formally introduce our notion of explanation.

Definition 3 (Explanation) *Let $c = \zeta(b, x)$ be an interpretable predictor such that $c(x) = b(x)$, and U a set of constraints. A local (counter)factual explanation $e = \langle r, \Phi \rangle$ is a pair of: a rule $r = (p \rightarrow y)$ consistent with c and satisfied by x; and, a set $\Phi = \{p[\delta_1] \rightarrow y', \ldots, p[\delta_v] \rightarrow y'\}$ of counterfactual rules for p consistent with c such that $p[\delta_i]$ satisfies U, for $i = 1, \ldots, v$.*

Unless otherwise stated, we will simply write "an explanation" instead of "a local (counter)factual explanation". According to Definition 2, we will design a solution to the outcome explanation problem by defining: *(i)* the function ζ that computes an interpretable predictor c for a given black-box b and an instance x, and *(ii)* the explanation logic ε that derives a (counter)factual explanation e from c and x as in Definition 3.

6.2 Local Rule-based Explanation

We propose LORE$_{sa}$, a Stable and Actionable LOcal Rule-based Explanation method, described in Algorithm 1 as extension of LORE [50]. LORE$_{sa}$ takes in input a black-box b, an instance x to explain, a set of constraints U, and a knowledge base K which contains information about feature distributions (domain of admissible values, mean, variance, probability distribution, etc.). LORE$_{sa}$ first generates N sets of neighbor instances $Z = \{Z^{(1)}, \dots, Z^{(N)}\}$ of x through a *genetic algorithm*. The knowledge base K is exploited in genetic mutation to be consistent with the distributions of the features. Next, LORE$_{sa}$ labels the generated instances with the black-box decision. For each labelled neighborhood $Z^{(i)}$ a decision tree $d^{(i)}$ is built, and all such trees are merged into a single interpretable predictor c still in the form of a *decision tree*. Rules and counterfactual rules are extracted from c, satisfying the constraints in U.

LORE$_{sa}$ fits the definitions of the previous section as follows: lines 1-9 in Algorithm 1 implement the ζ function for extracting the interpretable decision tree c, which approximates locally the behavior of the black-box b; and lines 10-11 implement the function ε to extract the (counter)factual explanation e from the logic of the decision tree.

Stability of the explanation process follows from the "bagging-like" approach of building and aggregating several decision trees. In fact, it is well-known that decision trees are unstable to small data perturbations [159]. Bagging is a widespread method to stabilize decision trees [160]. Experiments will confirm this by contrasting stability metrics of LORE$_{sa}$ with its "single-tree" version LORE. Resorting to bagging, however, produces a collection of interpretable explainers. We need then to aggregate them at symbolic level – which is different from standard bagging, where aggregation is at prediction time. For this, we have a merging procedure in line 9 of Algorithm 1.

The actionability of the counterfactuals follows from taking into account the constraint set U on admissible feature changes (Alg. 1, line 11). The search for counterfactuals will also consider the minimality requirement.

In the following, we discuss the details of LORE$_{sa}$ by motivating the design choices by the expected properties of the explanation process: locality, fidelity and stability, comprehensibility, actionability, and generality.

6.2.1 Locality: Neighborhood Generation

The goal of this phase is to identify sets of instances $Z^{(i)}$, whose feature are close to the ones of x, in order to be able to reproduce the behavior of the black-box b locally to x. Since the aim is to learn a predictor from $Z^{(i)}$, such a neighborhood should be flexible enough to include instances with decision values equal and different from $b(x)$. In Algorithm 1, first we extract balanced subsets $Z^{(i)}_{=}$ and $Z^{(i)}_{\neq}$ (lines 2–3), where instances $z \in Z^{(i)}_{=}$ are such that $b(z) = b(x)$, and instances $z \in Z^{(i)}_{\neq}$ are such that $b(z) \neq b(x)$, and then we define $Z^{(i)}$

$= Z_{=}^{(i)} \cup Z_{\neq}^{(i)}$ (line 4). We depart from instance *selection* approaches [161], and in particular the ones based on genetic algorithms [162], in that their objective is to select a subset of instances from an given training set. In our case, instead we cannot assume that the training set used to learn b is available, or not even that b is a supervised machine learning predictor for which a training set exists. Instead, our task is similar to instance *generation* in active learning [163], which also includes evolutionary approaches [164].

We adopt an approach based on a *genetic algorithm* which generates $Z_{=}^{(i)}$ and $Z_{\neq}^{(i)}$ by minimizing the following fitness functions:

$$fitness_{=}^{x}(z) = I_{x \neq z} + d(x,z) + l(b_p(x), b_p(z))$$
$$fitness_{\neq}^{x}(z) = I_{x \neq z} + d(x,z) + (1 - l(b_p(x), b_p(z)))$$

where $d : \mathcal{X}^{(m)} \to [0,1]$ is a distance function in the feature space (hence $d(x,z)$ is close to zero when two instances are similar with respect to their features), $l : \mathcal{R} \to [0,1]$ is a distance function in the label space with respect to the prediction probability b_p (hence $l(b_p(x), b_p(z))$ is close to zero when two instances are similar with respect to their label probabilities), and the function $I_{x \neq z}$ returns zero if z is not equal to x, and ∞ otherwise. The genetic neighborhood process tries to minimize these fitness functions. Therefore, $fitness_{=}^{x}(z)$ looks for instances z similar to x (term $d(x,z)$), but not equal to x (term $I_{x \neq z}$), for which the black-box b has a similar behavior (term $l(b_p(x), b_p(z))$). On the other hand, $fitness_{\neq}^{x}(z)$ leads to the generation of instances z similar to x, but not equal to it, for which b returns a different decision. We underline that $fitness_{=}^{x}(x)$=$fitness_{\neq}^{x}(x)$=∞. Hence, the minimization occurs for $z \neq x$.

A key element for the fitness functions are the distances $d(x,z)$ and $l(b_p(x),\ b_p(z))$. Concerning $d(x,z)$, we account for mixed types of features by a weighted sum of Simple Matching distance (SM) for categorical features, and of the normalized Euclidean distance (NE)[3] for continuous features. Assuming h categorical features and $m-h$ continuous ones, we use:

$$d(x,z) = \frac{h}{m} \cdot SM(x,z) + \frac{m-h}{m} \cdot NE(x,z).$$

Our approach is parametric to d, and it can readily be applied to improved heterogeneous distance functions [165]. In the following, a small parenthesis dedicated to a comparison of a few distance functions to see the behaviour of the parametric structure of our model.

Comparison among distance functions A key element of the neighborhood generation is the distance function used by the genetic algorithm. In this section we show how the explanations of LORE$_{sa}$ are affected by different distance functions. For example, [67] shows that considerable differences of the counterfactual instances occur at the variation

[3] See *NormalizedSquaredEuclideanDistance* at Wolfram.

X	distance	*silhouette*	*fidelity*	*coverage*	*precision*	*complexity*	*instability*
`compas`	*neuclidean*	**.54 ± .22**	**.99 ± .00**	**.44 ± .16**	1.00 ± .03	**4.97 ± 2.15**	**.21 ± .32**
	cosine	.50 ± .24	**.99 ± .00**	.43 ± .16	**1.00 ± .02**	5.00 ± 2.11	.24 ± .39
	nmeandev	.27 ± .26	**.99 ± .00**	.29 ± .18	.99 ± .11	5.10 ± 1.86	.24 ± .44
`fico`	*neuclidean*	.52 ± .17	**.98 ± .01**	**.40 ± .21**	.98 ± .10	**9.49 ± 3.77**	**.07 ± .04**
	cosine	**.54 ± .12**	**.98 ± .01**	.39 ± .19	**.99 ± .07**	9.88 ± 3.66	.27 ± .31
	nmeandev	.14 ± .17	**.98 ± .01**	.19 ± .19	.94 ± .21	9.78 ± 3.52	.18 ± .16
`german`	*neuclidean*	**.70 ± .57**	**1.00 ± .00**	**.87 ± .11**	**1.00 ± .00**	.98 ± .84	**.80 ± 1.97**
	cosine	.66 ± .57	**1.00 ± .00**	.78 ± .18	**1.00 ± .00**	1.09 ± .90	.97 ± 1.33
	nmeandev	.61 ± .60	**1.00 ± .00**	.85 ± .15	**1.00 ± .00**	**.73 ± .66**	.90 ± 1.27

b	distance	*silhouette*	*fidelity*	*coverage*	*precision*	*complexity*	*instability*
DNN	*neuclidean*	.61 ± .17	**.99 ± .01**	.55 ± .20	1.00 ± .02	6.96 ± 3.97	**.12 ± .38**
	cosine	**.62 ± .14**	**.99 ± .01**	**.56 ± .19**	**1.00 ± .01**	6.54 ± 3.82	.13 ± .38
	nmeandev	.12 ± .23	**.99 ± .01**	.21 ± .24	.96 ± .19	**6.22 ± 3.04**	.13 ± .45
NN	*neuclidean*	.50 ± .27	**.99 ± .01**	.32 ± .18	.99 ± .10	6.93 ± 3.79	**.84 ± .99**
	cosine	**.50 ± .24**	**.99 ± .00**	.32 ± .15	**.99 ± .07**	**6.88 ± 3.76**	1.08 ± 1.26
	nmeandev	.31 ± .31	**.99 ± .01**	**.37 ± .23**	.99 ± .09	6.93 ± 3.76	1.00 ± 1.17
SVM	*neuclidean*	**.48 ± .25**	**.99 ± .01**	.39 ± .17	.99 ± .10	**7.28 ± 4.18**	**.18 ± .09**
	cosine	.46 ± .27	**.99 ± .01**	**.39 ± .15**	**.99 ± .06**	7.32 ± 4.22	.56 ± .15
	nmeandev	.28 ± .27	**.99 ± .01**	.28 ± .21	.96 ± .18	7.56 ± 4.40	.22 ± .32

Table 6.1: Aggregated evaluation metrics over datasets (top) and black-boxes (bottom) w.r.t. distance functions in the neighborhood generation of LORE$_{sa}$.

of the distance function adopted by their stochastic optimization approach. As alternative distances to the normalized euclidean distance (*neucliden*) adopted by LORE$_{sa}$, we report results using the *cosine* distance and the normalized mean deviation (*nmeandev*) distance. Experiments over the `compas`, `fico` and `german` datasets, and over DNN, NN, and SVM black-boxes are reported in Table 6.1. There is no major difference in terms of fidelity and precision, whilst *neucliden* has the best performance or is a close runner up for all other metrics.

With regard to $l(b_p(x),\, b_p(z))$, we account for sparse numeric vectors by adopting the cosine distance. If b is not a probabilistic classifier, then $l(b_p(x), b_p(z))$ is replaced by identity checking, namely $l(b(x), b(z)) = 0$ if $b(x) = b(z)$, and 1 otherwise.

Genetic algorithms [166] are inspired by the biological metaphor of evolution and are based on three distinct aspects. *(i)* The potential solutions of the problem are encoded into representations that support the *variation* and *selection* operations. In our case, these representations, generally called chromosomes, correspond to instances in the feature space $\mathcal{X}^m$. *(ii)* A fitness function evaluates which chromosomes are the "best life forms", that is, most appropriate for the result. These are then favored in *survival* and *reproduction*, thus shaping the next generation according to the fitness function. In our case, these instances correspond to those similar to x, according to $d(\cdot, \cdot)$, and those similar/different

```
Algorithm 2: $genetic(x, fitness, b, K)$
    Input  : $x$ - instance to explain, $fitness$ - fitness function,
             $b$ - black-box, $K$ knowledge base
    Params: $n$ - population size, $g$ - nbr of generations,
             $p_c$ - prob crossover, $p_m$ - prob mutation
    Output: $Z$ - neighbors of $x$
    $P_0 \leftarrow (x \mid \forall 1, \dots, n)$; $i \leftarrow 0$;          // population init.
    while $i < g$ do
        $P' \leftarrow crossover(P_i, p_c)$;                          // mix records
        $P'' \leftarrow mutate(P', p_m, K)$;                          // perform mutations
        $S \leftarrow evaluate(P'', fitness, b)$;                     // evaluate population
        $P_{i+1} \leftarrow select(P'', S)$;                          // select sub-population
        $i \leftarrow i + 1$                                          // update population
    $Z \leftarrow P_i$
    return $Z$;
```

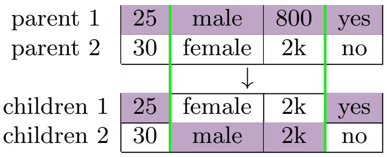


Figure 6.1: Crossover.

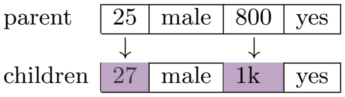


Figure 6.2: Mutation.

to the outcome returned by the black-box $b_p(x)$, according to $l(\cdot, \cdot)$, for the fitness function $fitness^x_=$ and $fitness^x_{\neq}$ respectively. *(iii)* Mating (called crossover) and mutation produce a new generation of chromosomes by recombining features of their parents. The final generation of chromosomes, according to a stopping criterion, is the one that best fits the solution.

Algorithm 2 generates the neighborhoods $Z^{(i)}_=$ and $Z^{(i)}_{\neq}$ of x by instantiating the evolutionary approach described in [167]. Using the terminology of the survey [164], it is an instance of generational genetic algorithms for evolutionary prototype generation. However, prototypes are a condensed subset of a training set that enable some optimization in training predictors. We aim instead at generating new instances that separate well the decision boundary of the black-box b. The usage of classifiers within fitness functions of genetic algorithms can be found in [168]. However, the classifier they use is always the one for which the population must be selected or generated from and not another one (the black-box) like in our case. Algorithm 2 first initializes the population P_0 with n copies of the instance x to explain. Then it enters the evolution loop that begins with the crossover operator applied to a proportion p_c of P_i: the resulting and the untouched instances are inserted in P'. We use a *two-point crossover* which selects two parents and two crossover

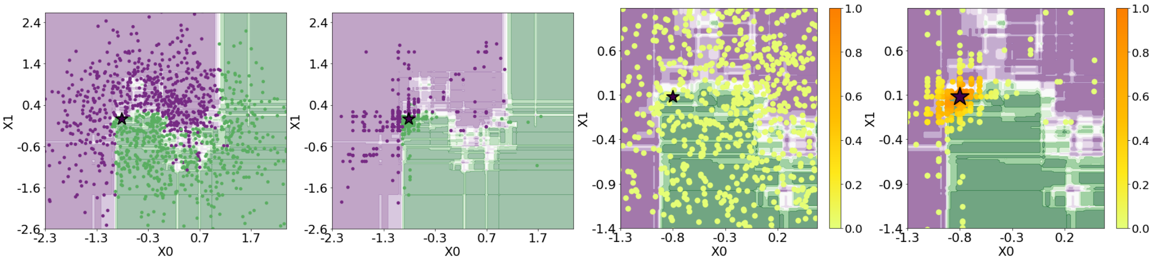


Figure 6.3: Black-box boundary: purple vs green. Starred instance x. Uniformly random (1^{st}) and genetic (2^{nd}) neighborhoods. In the (3^{rd}) and (4^{th}) plot is reported the density with levels in the bar (best view in color).

features and swap the crossover feature values of the parents (see Figure 6.1). Next, a proportion of P', determined by the p_m probability, is mutated (see Figure 6.2) by exploiting the feature distributions given by the knowledge[4] base K. Mutated and unmutated instances are added in P''. Instances in P'' are evaluated according to the fitness function, and the top n of them w.r.t. the fitness score are selected to become P_{i+1} – the next generation. The evolution loop continues until g generations are completed[5] The best individuals are returned. LORE$_{sa}$ runs Alg. 2 twice, once using the fitness function $\mathit{fitness}^x_=$ to derive neighbor instances $Z^{(i)}_=$, and once using the function $\mathit{fitness}^x_{\neq}$ to derive $Z^{(i)}_{\neq}$. Finally, setting $Z^{(i)}{=}Z^{(i)}_= \cup Z^{(i)}_{\neq}$ guarantees that $Z^{(i)}$ is balanced.

Figure 6.3 shows an example of neighborhood generation for a black-box consisting of a random forest model on a bi-dimensional feature space. The figure contrasts uniform random generation (1^{st}, 3^{rd} plots) around a specific instance x (starred) to our genetic approach (2^{nd}, 4^{th} plots). The latter yields a neighborhood that is denser in the boundary region of the predictor. The density of the generated instances is a key factor in extracting correct and faithful local interpretable predictors and explanations. For instance, a purely random procedure like the one adopted in Lime [35] does not account for sources of variability, like the randomness of the sampling procedure in the neighborhood of the instance to explain [169]. On the contrary, the genetic approach of LORE$_{sa}$ is driven by minimization of the fitness functions, hence less variable neighborhoods are generated. As a further

[4] K is assumed to include the probability mass functions of discrete features and the density function of continuous features. In experiments, K is empirically estimated from the set of instances to explain (not used for training the black-box) by taking the frequencies of values for discrete features, and by selecting the best fit of the empirical density of continuous features with one of the following families of distributions: uniform, normal, exponential, gamma, beta, alpha, chi-square, Laplace, log-normal, power law. We also assume that features are independent, hence, we do not infer the joint distribution.

[5] In the implementation of LORE$_{sa}$, we set the number of instances $n = 500$, the number of generations $g = 20$, the probabilities of crossover $p_c = 0.7$ and of mutation $p_m = 0.5$. In the following are also reported experiments showing the effect of varying these parameters.

issue, simply centering the neighborhood generation on the instance to explain may not be the best strategy to approximate the black-box decision boundary. [170] and [171] propose neighborhood generation approaches that enhance locally important features with respect to globally important ones by moving the center of the generation towards the decision boundary. The two fitness functions in the genetic generation procedure of LORE$_{sa}$ enforce the same effect.

6.2.2 Comprehensibility

We achieve high-level comprehensibility of explanations by extracting them in the form of factual rules and sets of counterfactual rules. Given the decision tree c, we derive an explanation $e = \langle r, \Phi \rangle$ as follows. The factual rule $r = p \rightarrow y$ is formed by including in p the split conditions on the path[6] from the root to the leaf satisfied by x, and setting $y = c(x) = b(x)$. By construction, r is consistent with c and satisfied by x. Consider now the counterfactual rules in Φ. Algorithm 3 looks for all paths in the decision tree c leading to a decision $y' \neq y$ (line 1). For one of such paths, let q be the conjunction of split conditions in it. By construction, $q \rightarrow y'$ is a counterfactual rule consistent with c. Notice that the counterfactual δ for which $q = p[\delta]$ has not to be explicitly computed[7]. All such q's can be ranked by the number of split conditions not satisfied by x, a.k.a. the number of features to be changed in x. The $q \rightarrow y'$'s with minimal number of changes are returned as counterfactuals (lines 6-8).

6.2.3 Actionability

The counterfactuals provided by LORE$_{sa}$ support actionable recourse. This is implemented in Algorithm 3 by filtering from the candidate counterfactuals $q \rightarrow y'$ those not satisfying the constraints U on features (lines 4-5). Since both the premise q and the constraints U are logic formulae, the test amounts at checking validity of the implication $q \rightarrow U|_q$. For the basic form of constraints that we have considered (conjunction of equality/comparison conditions) the test is straightforward. In principle, however, more complex premises (e.g., multivariate) can be dealt with by resorting to automatic theorem proving.

Let assume that the decision tree in Figure 6.4 is the merged decision tree c. Let x=\{*age*=22, *sex*=*male*, *income*=800, *car*=*no*\} be the instance for which the decision *deny* (e.g., of a loan) has to be explained. The path followed by x is the leftmost in the tree. The

[6] The set of split conditions in the path is also called a direct reason, and it is not necessarily minimal. Minimal sets (called sufficient conditions, or prime implicant explanations) are considered in [172, 173]. We do not further purse minimizing the factual explanation as experiments shows LORE$_{sa}$ returns very small rules.

[7] However, it can be done as follows. Consider the path from the leaf of p to the leaf of q. When moving from a child to a father node, we retract the split condition. E.g., $a_i \leq v_i^{(u)}$ is retracted from $\{a_j \in [v_j^{(l)}, v_j^{(u)}]\}$ by adding $a_i \in [v_i^{(l)}, +\infty]$ to δ. When moving from a father node to a child, we add the split condition to δ.

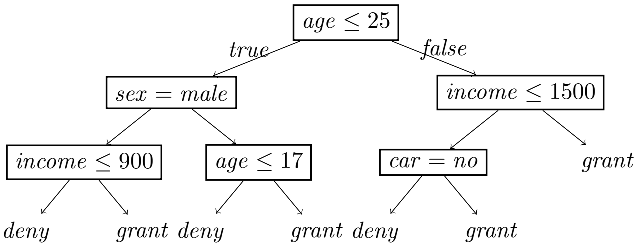


Figure 6.4: Example of decision tree locally mimicking the black-box behavior.

decision rule extracted from the path is $\{age{\leq}25, sex{=}male, income{\leq}900\}{\to}deny$. There are four paths leading to *grant*: $q_1{=}\{age{\leq}25, sex{=}male, income{>}900\}$, $q_2{=}\{17{<}age{\leq}25, sex{=}female\}$, $q_3{=}\{age{>}25, income{\leq}1500, car{=}yes\}$, and $q_4 = \{age{>}25, income{>}1500\}$. The number of changes for the q_i's are as follow: $nf(q_1, x){=}1$, $nf(q_2, x){=}1$, $nf(q_3, x) = 2$, $nf(q_4, x){=}2$. Therefore, the set of minimal counterfactuals is $\Phi{=}\{q_1{\to}grant,\ q_2{\to}grant\}$. Assuming that $U{=}\{sex{=}male\}$, then $q_2{\to}grant$ is not actionable, hence the set of actionable counterfactuals is $\Phi{=}\{q_1{\to}grant\}$.

Finally, we point out that an actionable counterfactual rule $q \to y'$ can be used to generate an *actionable counterfactual instance*. Among all possible instances that satisfy $q \to y'$, we choose the one that differ minimally from x. This is done by looking at the split conditions falsified by x: $\{sc \in q \mid \neg sc(x)\}$, and selecting for features appearing in an sc the lower/upper bound that is closer to the value of the feature in x. For instance, the $q_1{\to}grant$ counterfactual instance of x is $x' = \{age{=}22, sex{=}male, income{=}900{+}\epsilon)\}$. We also check that x' constructed in this way is a valid counterfactual, i.e., $b(x'){=}grant$. If this does not occur, x' is not returned as a counterfactual instance.

6.2.4 Generality

Following the approach of Lime [35], LORE$_{sa}$ can be adapted to work on images and texts. Moreover, inspired by [95], we show how it deals with multi-label data.

Image and Text Data. In the pre-processing strategy of Lime, an instance in the form of an image or a text is mapped to a vector of binary values. For images, each element in the vector indicates the presence/absence of a contiguous patch of similar pixels (called super-pixels). For words, it indicates the presence/absence of a specific word in the text. This reduces the problem to the analysis of tabular data, and we can reuse LORE$_{sa}$ as introduced so far. Due to the binary nature of data involved, the genetic neighborhood approach boils down to generate instances by suppressing super-pixels or words from the instance to explain. This is close to the way that Lime works, but with a fitness optimizing approach instead of a purely random suppression. As for Lime, the generated instances may not be realistic images or texts.

Multi-labelled Data. The formulation of LORE$_{sa}$ admits so far binary and multi-class black-boxes. Multi-labelled classifiers return, for an input instance x, one or more class labels. This case is common, for instance, in health data, where more than one disease may be associated with a same list of symptoms. In particular, probabilistic multi-labelled classifiers return a vector of probabilities $b_p(x)$ whose sum is not necessarily 1, as in the multi-class case. Rather, the i^{th} element in $b_p(x)$ is the probability that the i^{th} label is included in the output (with a typical cut-off at 0.5). LORE$_{sa}$ can be extended to (probabilistic) multi-labelled black-boxes by adopting multi-class decision trees in the function *buildDecisionTree*() of Algorithm 1. Factual rules will be of the form $p \rightarrow y_1, \ldots, y_k$, with $k \geq 1$. Counterfactual rules will be of the form $p[\delta] \rightarrow y'_1, \ldots, y'_{k'}$, with $k \geq 1$ and such that $\{y_1, \ldots, y_k\} \neq \{y'_1, \ldots, y'_{k'}\}$ (but possibly with proper inclusion).

6.3 Experiments

After presenting the experimental setting and the evaluation metrics, we compare LORE$_{sa}$ against the competitors through: *(i)* a qualitative comparison of explanations provided, *(ii)* a quantitative validation of the explanations based on synthetically generated ground truth, and *(iii)* a quantitative assessment of the proposed method and comparison with state-of-the-art approaches in terms of several metrics[8]. Moreover, the Appendices report further experiments: *(iv)* comparing different neighborhood generation methods, *(v)* showing the impact of different distance functions in genetic neighbor generation, *(vii)* illustrating the effect of the parameters on the genetic neighbor generation, *(vii)* providing statistical evidence of the differences among LORE$_{sa}$ and its competitors, and *(viii)* reporting on running times.

6.3.1 Experimental Setup

We experimented with ten tabular datasets, one image dataset, one text dataset, and one multi-labelled dataset. Table 6.2 reports the dataset details. Almost all tabular datasets have both categorical[9] and continuous features. For most of the datasets, instances regard attributes of an individual person, and the decisions taken by a black-box target socially sensitive tasks.

[8] LORE$_{sa}$ has been developed in Python, using *deap* [174, `https://github.com/DEAP/deap`] for genetic neighborhood generation, and the optimized version of CART [175] offered by *scikit-learn* (`https://scikit-learn.org/stable/modules/tree.html`) for decision tree induction. The source code of LORE$_{sa}$, the datasets, and the scripts for reproducing the experiments are publicly available at `https://github.com/francescanaretto/LORE_sa`. Experiments were performed on Ubuntu 20.04 LTS, 252 GB RAM, 3.30GHz × 36 Intel Core i9.

[9] The number of features is calculated prior to one hot encoding.

	tabular data										images	text	multi-l.
#	adult	bank	churn	compas	compas-m	fico	german	iris	wine-r	wine-w	mnist	20news	medical
n	32,561	150k	3,333	7,214	7,214	10,459	1,000	150	1,599	54,898	70,000	18,846	978
m	10	13	19	11	11	23	20	4	11	11	28×28	37,096	1449
L	2	2	2	2	3	2	2	3	6	7	10	2	45
X_{bb}	.83	.89	.85	.81	.64	.72	.76	.91	.71	.55	.99	.91	.52
X	.80	.89	.83	.75	.61	.69	.70	.87	.55	.41	.98	.75	.98

Table 6.2: Top: datasets summaries. n: instances, m: fetures, L: labels. Bottom: average accuracy of all black-boxes (DNN, NN, RF and SVM) on training X_{bb} and test X.

	DNN	NN	RF	SVM
X	.69 ± .24	.75 ± .17	.78 ± .12	.68 ± .16
X_{bb}	.72 ± .26	.76 ± .17	.88 ± .11	.77 ± .13

Table 6.3: Average accuracy and stddev of the black-box classifiers.

A random subset of each dataset, denoted by X_{bb}, was used to train the black-box classifiers while the remaining part, denoted by X, was used as instances to explain – in brief, the *explanation set*. For tabular data, the split was 70%-30% and stratified w.r.t. the class attribute. For mnist, 20news, and medical we followed the split custom in the relevant literature[10]. We denote with $\hat{Y} = b(X)$ the decisions of b on X, and with $Y = c(X)$ the decisions of c on X. We assume that the dataset used to train the black-box is unknown at the time of explanation. Hence, we can only rely on the set X of instances to explain. Indeed, the knowledge base K is derived from the explanation set as stated in Footnote 4. Similarly, information about features' domains required by the competitor methods is computed from X.

We trained and explained the following black-box models: Random Forest (RF), Support Vector Machine (SVM) and Neural Network (NN) as implemented by *scikit-learn*, and Deep Neural Networks (DNN) implemented by *keras*[11]. For each black-box, for each dataset, we performed a random search for the best parameter setting[12]. Average classification accuracies are shown in Table 6.2 (bottom) and in Table 6.3. We compare LORE$_{sa}$ against Lime [35], Maple [44], Shap [41], Anchor [51] and brl [53]. We also compare the counterfactuals of LORE$_{sa}$ with the stochastic optimized counterfactuals Soc [176] as implemented by the alibi library[13], and against the brute force coutnerfactual explainer (bf) as implemented by the fat-forensics library[14]. Unless stated otherwise, default parameters are used for LORE$_{sa}$ and all the other methods[15].

[10] http://qwone.com/~jason/20Newsgroups/, http://yann.lecun.com/exdb/mnist/

[11] Black-boxes: https://scikit-learn.org/, https://keras.io/.

[12] Details of the parameters can be found in LORE$_{sa}$ repository.

[13] https://github.com/SeldonIO/alibi

[14] https://fat-forensics.org/.

[15] We highlight that for SHAP we used the KernelSHAP explainer that can be adopted for any black-box

6.3.2 Evaluation Metrics

We evaluate the performances of explanation methods under various perspectives. The measures reported in the following are stated for a single instance to be explained. The metrics obtained as the mean value of the measures over all the instances in the explanation set X, can then be used to evaluate the performances of the explanation methods. Let $x \in X$ be an instance to explain.

Correctness. We will evaluate the correctness of explanations under controlled situations where ground truth is available. Let e and $\widetilde{e}$ be the binary vectors indicating the presence/absence (1/0) of a feature in the explanation for x of a given method, and in the ground truth respectively. For rule-based explanations, presence means that the feature appears in the premise of the rule. For feature importance vectors, presence means that the feature has non-zero magnitude. We measure the *correctness* of an explanation w.r.t. the ground-truth using the *f1-score*:

$$\mathit{f1\text{-}score}(e, \widetilde{e}) = 2 \cdot \frac{\mathit{recall}(e, \widetilde{e}) \cdot \mathit{precision}(e, \widetilde{e})}{\mathit{recall}(e, \widetilde{e}) + \mathit{precision}(e, \widetilde{e})}$$

where the *precision* is the percentage of features present in e that are also in $\widetilde{e}$, and the *recall* is the percentage of features in $\widetilde{e}$ that are also in e.

When ground truth is not available, we will consider the following measures to evaluate specific properties of an explanation process.

Silhouette. We measure the quality the neighborhood[16] in a local approach by measuring how similar is x to instances in $Z_=$ compared to instances in $Z_{\neq}$. Let $d(x, S)$ denote the mean Euclidean distance between x and instances in S. Inspired by clustering validation [175], we define:

$$\mathit{silhouette}(x) = \frac{d(x, Z_{\neq}) - d(x, Z_=)}{\mathit{max}\{d(x, Z_{\neq}), d(x, Z_=)\}}$$

High silhouette results from accurate neighborhood generation (Section 6.2.1).

Fidelity. It answers the question: how good is the interpretable predictor c at mimicking the black-box b? Fidelity can be measured in terms of accuracy [10] of the predictions $Y = c(Z)$ of the interpretable predictor c w.r.t. the predictions $\hat{Y} = b(Z)$ of the black-box

model. Also, as background knowledge for SHAP we used the medoid of the training set. We highlight that, different choices of the background knowledge can significantly impact on the outcome as illustrated in [177, 178]. However, we relied on the medoid because as illustrated in the tutorial for KernelSHAP on tabular data provides the best trade-off between reliability and efficiency. We did not compare against other counterfactual explainers as this is out from the purpose of the analysis conducted. We refer to [69] for a comprehensive survey and benchmarking.

[16] In order to evaluate the neighborhood generated by an explainer, it must be available. BRL, MAPLE and SHAP do not use a notion of neighborhood to return the explanation. However, the SHAP library allows access to the permutation of x tested to determine the Shapely value approximations. We used this set of instances as the neighborhood for SHAP.

b, where Z is the neighborhood of x generated by the local method. High fidelity of c results from both accurate neighborhood generation (Section 6.2.1) and predictive performance of the learning algorithm.

Complexity. It is a proxy of the comprehensibility of an explanation, with larger values of complexity denoting harder to understand explanations [24]. For rule-based explanations, as complexity we adopt the size of the rule premise (for LORE$_{sa}$ we consider only the factual rule). Low complexity results from general (non-overfitting, stable) local interpretable surrogate predictors and a direct method to extract the rule. For feature importance vectors, as complexity we adopt the number of non-zero features. For instance in LIME are those of the local surrogate linear regressor.

Stability. It measures the ability to provide similar explanations to similar instances. Also named *robustness* or *coherence*, it is a crucial requirement for gaining trust by the users [179]. We measure it through the local Lipschitz condition [96]:

$$instability(x) = max_{x_i \in \mathcal{N}_k(x)} \frac{\|e_i - e\|_2}{\|x_i - x\|_2} \tag{6.1}$$

where $\mathcal{N}_k(x)$ is the set of the $k = 5$ instances in $X \setminus \{x\}$ closest to x w.r.t. Euclidean distance, e is the binary vector of the explanation of x, and e_i is the binary vector of the explanation of $x_i \in \mathcal{N}_k(x)$. Intuitively, the larger is the ratio the more different are the explanations for instances close to x. Low instability (or, high stability) results from general (non-overfitting, stable) local interpretable surrogate predictors. While low instability could be the result of under-fitting, this is not the case of *local* explanation methods which, being local and being based on random components, are not prone to exhibit the same explanation for different instances. In addition, we consider also sensitivity of a local explanation method to randomness introduced in the neighborhood generation. This is measured by the distance of explanations generated for a same instance over multiple calls to the explanation method:

$$instability_{si}(x) = max_{e_i, e_j \in \mathcal{E}_k(x)} \|e_i - e_j\|_2 \tag{6.2}$$

where $\mathcal{E}_k(x)$ is the set of the explanations obtained by calling the method $k = 5$ times on the same input instance x. A low same-instance instability is obtained when similar explanations are returned over multiple runs. Instances and explanations are normalized before calculating the instability measure.

Coverage and Precision. These measures apply to rule-based explanations $p \to y$ only (for LORE$_{sa}$ we consider only the factual rule). Let Z be the neighborhood of x generated by the local method. The coverage of the explanation is the proportion of instances in Z that satisfy p. The precision is the proportion of instances $z \in Z$ satisfying p such that $b(z) = y$. Coverage and precision are competing metrics which respectively estimate the generality of the rule and the probability it correctly models the black-box behavior locally to the instance to explain. They depend both on the characteristics of the neighborhood generation (Section 6.2.1) and on the predictive performance of the learning algorithm.

Changes. An indicator of the quality of a counterfactual is the number of changes w.r.t. the instance x. For a set of counterfactual instances, such as those provided by Soc, we count the mean number of features whose value is different from x. For a set of counterfactual rules $p[\delta] \rightarrow y$, provided by LORE$_{sa}$, we count the mean number of falsified split conditions $nf(p[\delta], x)$. For LORE$_{sa}$, we expect a small number of changes thanks to the selection of counterfactual paths in the surrogate predictor with minimum number of changes (Section 6.2.2). However, actionability of counterfactuals maybe achieved at the cost of a larger number of changes (Section 6.2.3).

Dissimilarity. We measures the proximity between x and the counterfactual x' generated as the distance between x and the counterfactual instance x' that we obtain by applying to x the changes described by $p[\delta]$. We calculate the distance using the same function described in Section 6.2.1. The lower the better.

Plausibility. We evaluate the plausibility of the explanations in terms of the goodness of the counterfactuals returned by using the following metrics based on distance and outlierness [180].

Minimum Distance Metric. As a straightforward but effective evaluation measure, we adopt proximity. Given the counterfactual x' returned for instance x, x' is plausible if it is not too much different from the most similar instance in a given reference dataset X. Hence, for a given explained instance x, we calculate the plausibility in terms of Minimum Distance $MDM = \min_{\bar{x} \in X/\{x\}} d(x', \bar{x})$ where the lower the MDM, the more plausible is x' the more reliable is the explanation, because x' resembles a real instance in X.

Outlier Detection Metrics. We also evaluate the plausibility of the counterfactuals by judging how much they appears as outliers. The lower the scores the more plausible they are. In particular, we estimate the degree of outlierness of a counterfactual x' returned for an instance x by employing the outlier detection technique Isolation Forest (IsoFor) [181].

6.3.3 Qualitative Evaluation

We qualitatively compare LORE$_{sa}$ explanations with those returned by competitors on an instance x of the `compas-m` dataset, assuming a NN as the black-box. The instance and the explanations are shown in Figure 6.5.

The factual rule r of LORE$_{sa}$ clarifies that x is considered at high risk of recidivism because of his young *age* and of the number of *previous detections.* The counterfactuals Φ show that the risk would have been lowered to *Low* for an older individual, or *Medium* for various reasons some of which are not actionable, e.g., different age, sex or race. The counterfactuals Φ^* are obtained by considering the set of constraints U=\{ *age=20, age_cat=Less than 25, race=Afr.-Am., sex=Male*\}. In this case, the decision $b(x)$ would have been different only with a lower number of prior arrests or with a larger number of days between the screening and the arrest.

The competitor rule-based explainers suffer from a few weaknesses. Anchor returns

x = {age = 20, priors_cnt = 3, days_b_screen_arrest = 0,
is_recid = 1, is_violent_recid = 0, two_year_recid = 1,
length_of_stay = 1, age_cat = Less than 25, sex = Male,
race = Afr.-Am., charge_degree = F}
b(x) = High

LORE$_{sa}$

r = {age ≤ 20.5, days_b_screen_arrest ≤ 0.50,
priors_cnt > 2.5, 0.5 < length_of_stay ≤ 3.5,
sex = Male, race ≠ Asian } → High
Φ ={ { age > 28.5 } → Low, {age > 23.5} → Medium,
{sex = Female} → Medium, {race = Asian} → Medium
{days_b_screen_arrest > 7.5} → Medium,
{priors_cnt ≤ 2.50} → Medium } }
Φ* ={ { days_b_screen_arrest > 7.5 } → Medium,
{ priors_cnt ≤ 2.50 } → Medium }

ANCHOR

e = {age_cat=Less than 25 > 0.0, age_cat=25 - 45 ≤ 0.0,
age ≤ 25.0, days_b_screen_arrest ≤ 1.0,
0.0 < race=Afr.-Am. ≤ 1.0, race=Caucasian ≤ 0.0,
race=Asian ≤ 0.0, race=Other ≤ 0.0,
priors_cnt > 2.0, 0.0 < length_of_stay ≤ 1.0,
0.0 < two_year_recid ≤ 1.0, 0.0 < is_recid ≤ 1.0,
0.0 < charge_degree=F ≤ 1.0, charge_degree=M ≤ 0.0,
sex=Female ≤ 0.0, sex=Male ≤ 1.0 } → High

BRL

e = {12.04 < age ≤ 34.41} → High

LIME

e = [(age, 0.02), (priors_cnt, -0.01),
(is_violent_recid, 0.002),
(days_b_screen_arrest, 0.002), (sex=Male, -0.002),
(charge_degree=M, 0.002), (is_recid, 0.001),
(age_cat=25 - 45, 0.001),
(charge_degree=F, -0.001)]

SHAP

e = [(age, -0.7)]

MAPLE

e = [(age, -1.84), (priors_cnt, 0.19),
(two_year_recid, 0.04),
(days_b_screen_arrest, -0.28), (sex=Female, 0.06),
...
(charge_degree=F, -0.02),
(charge_degree=M, 0.02)]

SOC

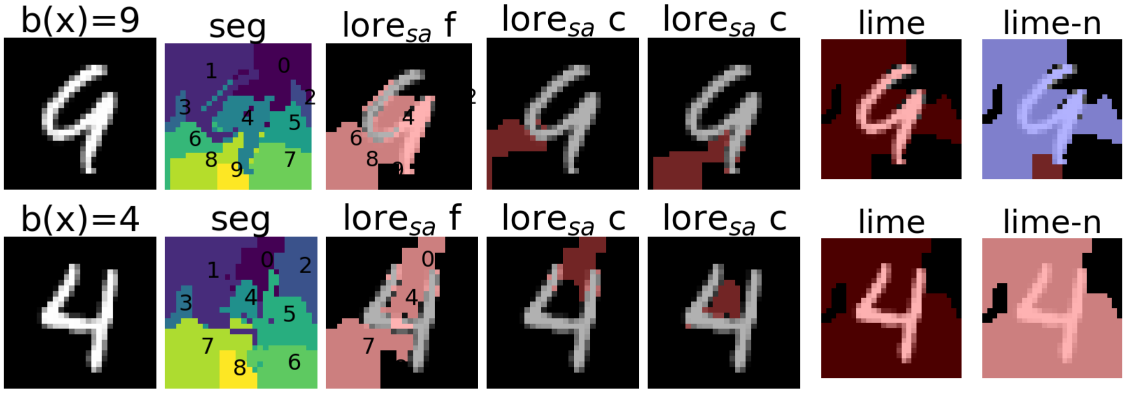
e = {[(age, 17.31)], [(priors_cnt, 4.34)],
[(age_cat=25 - 45, 1.0), (race=Other, 1.0)],
[(is_violent_recid, 10), (charge_degree=M, 10.)] }
b(x[e]) = Medium

Figure 6.5: Explanations for an instance x of the `compas-m` dataset classified as *High* risk of recidivism by a NN black-box.

various conditions, involving many features, in order to guarantee high precision. Thus, its explanation result hard to read and unnecessarily complex. BRL bases its explanation on a rule with a single feature, which on the example instance is *age*. Even though it is (partly) correct, the user can hardly trust such a simple and minimal justification. We will show next that BRL is indeed not particularly good in mimicking black-boxes' behaviors. The feature importance-based explainers LIME, SHAP and MAPLE provide a list of features with a score of their relevance in the decision. The most important features for LIME, i.e., *age* and *priors_cnt*, are in line with the factual rule of LORE$_{sa}$. SHAP attributes the decision of the black-box only to *age*. MAPLE provides a (unnecessarily long) list of features (shortened for space reasons) with scores in agreement with the other explainers. Regarding counterfactuals, SOC suggests a set of changes to x's feature values turning the risk prediction to *Medium*. Compared to Φ^*, the changes are either non-actionable (e.g., *age*=17.31) or less informative or impossible (e.g., *priors_cnt*=4.34).

Explanations on Images, Texts & Multi-label Data. We compare LORE$_{sa}$ explanations for images and texts with LIME explanations.

Figure 6.6 shows such comparison on two images of `mnist`. Both methods adopt the same segmentation shown in the second column of the figure. The factual explanations of LORE$_{sa}$, shown visually in the 3^{rd} column of Figure 6.6, clearly attribute the classifications for *9* and *4* to the presence of super-pixels s_8, s_6, s_4 and s_7, s_0, s_4, respectively. The absence



LORE$_{sa}$

$r_1 = \{s_8,\ s_6,\ s_4\ \} \rightarrow \{\mathit{9}\}$

$\Phi_1 = \{\ \{\neg s_6\} \rightarrow \{\mathit{4}\},\ \{\neg s_8\} \rightarrow \{\mathit{6}\}\ \}$

$r_2 = \{s_7,\ s_0,\ s_4\ \} \rightarrow \{\mathit{4}\}$

$\Phi_2 = \{\ \{\neg s_0\} \rightarrow \{\mathit{9}\},\ \{\neg s_4\} \rightarrow \{\mathit{9}\}\ \}$

Figure 6.6: Explanations of LORE$_{sa}$ and LIME for two instances x (one per row) of the `mnist` dataset classified as *9* and *4* by a RF black-box. Meaning of columns is 1st: instance x, 2nd: superpixel segmentation, 3rd: LORE$_{sa}$ factual rule, $4-5^{th}$: LORE$_{sa}$ counterfactuals, 6th: LIME explanation, 7th: LIME counterfactuals (towars unspecified class).

of some of such super-pixels (4th column), would have changed the black-box decision as shown in Φ_1 and Φ_2. For instance, the image of *9* would have been classified as *4* if the area of the super-pixel s_6 would have been white. The explanation returned by LIME are less intuitive both when considering only the super-pixels pushing the classification towards a class (5th column), or pushing the classification towards another (unspecified) class (6th column).

Figure 6.7 reports the explanations of LORE$_{sa}$, LIME and ANCHOR for a text from the `20news` dataset. All methods adopt the same document vectorization. LORE$_{sa}$ shows that the text is classified as *atheism* because of the simultaneous presence of some words in the factual rule. The absence of specific words in the counterfactual rules would change the classification to *christian*. LIME explanation is in agreement with the one of LORE$_{sa}$ as the words *edu*, *com* and *religion* have negative weight on the classification towards *atheism*. The explanation of ANCHOR highlights the presence of *religion* and *religious*, but it also includes less meaningful words.

Figure 6.8 reports an example of explanation derived for multi-labelled classification using the `medical` dataset. The instance x is labelled with the diseases corresponding to *Class 12* and *Class 38*. The explanation is the conjunction of symptoms in the factual rule r. A single label would have been returned by the black-box if *cough* were absent and,

x = {*Could an atheist accept a usage in which religious literature or tradition is viewed in a metaphorical way? [...] It's also entirely unclear, and to me quite unlikely, that one could take a contemporary religion like that and divorce the metaphoric potential from the literalism and absolutism it carries now in many cases.*}
$b(x)$ = *atheism*

LORE$_{sa}$

r = {*Christianity, com, religion, edu, religious, atheist, believes, cons*} → *atheism*
Φ ={ {¬ *religion*} → *christian*, {¬ *com*} → *christian*, {¬ *religious*} → *christian*, {¬ *edu*} → *christian* }

Lime

e = *[(Christianity, 0.05), (want, 0.04), (edu, -0.02), (com, -0.02), (good, 0.02), (religion, -0.02)]*

Anchor

e = {*religion, religious, set, an* } → *atheism*

Figure 6.7: Explanations of LORE$_{sa}$ and Lime for an instance x of the `20news` dataset classified as *atheism* by a NN black-box.

x = {*15-month, chest, cough, fever, focal, male, normal, pneumonia, x-ray*}
$b(x)$ = {*Class 12, Class 38*}
r = {¬ *hydronephrosis, cough, fever, minimal* } → { *Class 12, Class 38* }
Φ ={ {¬ *cough*, ¬ *pneumonia*} → {*Class 12*}, {¬ *cough, hypertrophy*} → {*Class 38*} }

Figure 6.8: LORE$_{sa}$ explanations for an instance x of the `medical` dataset classified as *Class 12* and *Class 38* by a RF black-box.

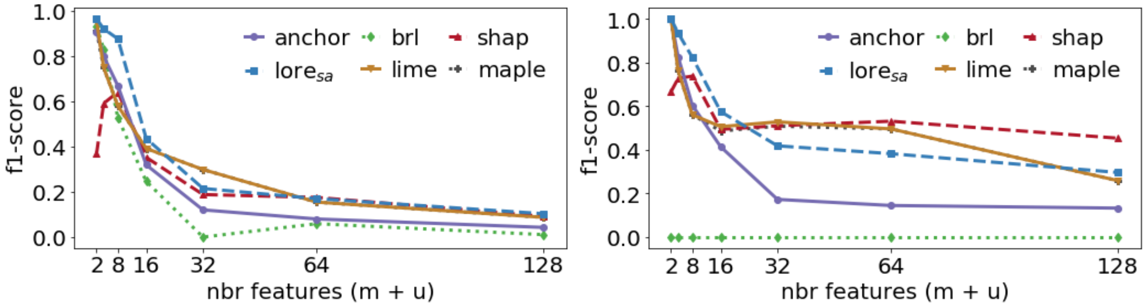


Figure 6.9: Correctness metric by varying the total number of features $m+u$. *Left:* synthetic rule-based classifiers. *Right:* synthetic linear regressors.

either *pneumonia* were absent or *hypertrophy* were present. We cannot compare with Soc, because it is not able to deal with multi-labelled classification.

In conclusion, we believe that the reported examples of factual, counterfactual, and actionable explanations of LORE$_{sa}$ offer to the user a clearer and more trustable understanding than what is offered by the other explainers.

6.3.4 Ground Truth Validation

By synthetically generating transparent classifiers and using them as black-boxes, we can compare the explanations provided by an explainer with the ground-truth decision logic of the black-box [99]. In particular, the *f1-score*() metric accounts for the correctness of the explanations.

In order to have a comparison as fair as possible among methods returning different types of explanations, we build two types of black-boxes: rule-based classifiers and linear regressor-based. The former are closer to rule-based explainers, the latter to feature importance explainers. In both cases, we start from datasets of m binary informative features and u Gaussian-noise uninformative features. The total number of features $m+u$ varies over $\{2, 4, 8, 16, 32, 64, 128\}$ and, for a fixed $m+u$, we generate 100+100 such datasets where $m < \min\{32, m+u\}$[17]. The informative features are generated following the approach of [182] implemented in `scikit-learn`[18]. Thus, we have 700 synthetic datasets for training rule-based classifiers and 700 for training linear regressors. Each dataset contains 10,000 instances, 1000 of which are used as explanation set.

Rule-based black-boxes are obtained by training a decision tree from a synthetic dataset, and then extracting rules from such a decision tree. The ground-truth explanation for an instance x is the rule satisfied by x in the black-box. Linear regressors black-boxes are obtained by an adaption of the approach of [183]. The ground-truth explanation for an instance x is the gradient of the instance in the decision boundary closest to x. Additional details[19] can be found in [99].

Figure 6.9 reports the *f1-score* metric at the variation of the total number of features $m+u$ in synthetic datasets. Each point shows the mean *f1-score* over the explanation sets of such datasets. LORE$_{sa}$ outperforms the other explainers when $m+u \leq 16$. For larger values of $m+u$, LORE$_{sa}$ performance is comparable to those of LIME and SHAP for rule-based classifiers, and slightly lower than their performance for linear regressors.

6.3.5 Quantitative Evaluation

We quantitatively assess the quality of LORE$_{sa}$ and of the competitor explainers through the other evaluation metrics of Section 6.3.2.

In order to evaluate the importance of the trees merging strategy employed by LORE$_{sa}$ for deriving the single local decision tree, we implemented a variant that avoids the merging operation. We call it LORE$^{d}_{sa}$ and works as follows. After learning the decision trees $d^{(1)}, d^{(2)}, \dots, d^{(N)}$ on their corresponding local neighborhood $Z^{(1)}, Z^{(2)}, \dots, Z^{(N)}$ labeled by the back-box b, we use each tree $d^{(i)}$ for labeling its training neighborhoods, $Z^{(i)}$,

[17] We specified 32 as maximum number of features m because typically tabular datasets with columns having clear and interpretable semantics have less than 30 features (like those used in the experiments). Thus, since our purpose is not to perform a scalability test but a correctness test, we selected this upper limits.

[18] `https://scikit-learn.make_classification`.

[19] We highlight that the transformation of features importance and of rules into binary vectors indicating the presence of a feature is a simplification adopted to make possible the comparison of explainers returning different types of explanations using the same metric.

	silhouette	*fidelity*	*complexity*	*instability*	*instability$_{si}$*
Anchor	.116 ± .51	.912 ± .21	4.950 ± 8.20	.174 ± 0.29	.651 ± .949
BRL	.019 ± .30	.869 ± .09	**1.998 ± 1.23**	.889 ± 0.45	n.a.
Lime	.444 ± .49	.904 ± .23	9.733 ± 1.47	.787 ± 1.58	.159 ± .142
Lore	.408 ± .49	.996 ± .01	4.917 ± 3.69	.123 ± 0.22	.259 ± .847
Maple	.127 ± .56	.949 ± .09	29.014 ± 3.25	.651 ± 1.66	n.a.
Shap	.463 ± .56	n.a.	6.070 ± 3.84	.608 ± 0.58	**.017 ± .052**
LORE$_{sa}$	**.569 ± .46**	.992 ± .20	3.986 ± 3.93	**.073 ± 0.07**	.107 ± .081
LORE$^{d}_{sa}$	**.569 ± .46**	**.999 ± .01**	5.105 ± 4.29	.083 ± 0.08	.107 ± .066

	Anchor	Lore	BRL	LORE$_{sa}$	LORE$^{d}_{sa}$
coverage	.284 ± .32	.492 ± .27	.344 ± .30	**.742 ± .27**	.485 ± .26
precision	.912 ± .21	.993 ± .07	.732 ± .22	.772 ± .26	**.998 ± .02**
h-mean	.433 ± .25	.657 ± .11	.468 ± .25	**.694 ± .25**	.615 ± .22

Table 6.4: Aggregated evaluation metrics over experimental datasets and black-boxes.

i.e., $Y_d^{(i)} = d^{(i)}(Z^{(i)})$. Then, we compute the union of the new labeled neighbors, i.e., $D^Z = \bigcup_{\forall i \in [1,N]} (Z^{(i)}, Y_d^{(i)})$ and we use D^Z to learn the final decision tree c.

For sake of compactness, to quantitatively compare all the explanation methods we report only aggregate results, i.e., mean and standard deviation of the metrics over all datasets and black-boxes. Table 6.4 (top) reports the *silhouette*, *fidelity*, *complexity*, *instability*, and *instability$_{si}$* metrics. LORE$_{sa}$ overcomes all the other explainers on 3 metrics, and it is runner-up on the other 2 metrics.

As expected, LORE$_{sa}$ considerably improves the *complexity* and the two *instability* metrics with respect to its predecessor Lore while maintaining the same level of *fidelity*. In terms of *complexity*, LORE$_{sa}$ is the second best performer after the BRL approach which, on the other hand, has lower performance on the other metrics and is one of the most stable. The only competitors with lower *instability* are Shap and Maple which provide more complex explanations. Moreover, our experimental results show that LORE$_{sa}$ has also lower complexity and instability with respect to LORE$^{d}_{sa}$ highlighting the importance of the merging procedure for the stability. The better performance of LORE$_{sa}$ is paid with a slightly higher runtime required to get an explanation due to the merging procedure that is on average 315.59 ± 185.74 seconds among all datasets and black-box models 315.59, while it is on average 285.23 ± 179.83 seconds for LORE$^{d}_{sa}$. We underline that having an efficient implementation of this algorithm is not the focus of this work. Nonetheless, in Table 6.6 we report the running time (in secs) of producing an explanation for three experimental datasets and for the SVM black-box. LORE$_{sa}$ performances are in line with Anchor and Soc, and better than Maple and BRL. They are instead worse than Lime and Shap. The vast majority of running time (> 90%) of LORE$_{sa}$ is used by the genetic neighborhood generation. The implementation, however, can be readily sped up by parallelising the generation of $Z_{=}^{(1)}, Z_{\neq}^{(1)}, \ldots, Z_{=}^{(N)}, Z_{\neq}^{(N)}$ ($2 \cdot N$ independent calls to Algorithm 2).

method	RF	SVM	NN	DNN
Anchor	3.02	3.04	3.33	3.41
BRL	3.72	3.77	3.96	1.72
Lime	3.93	3.03	4.25	3.33
Maple	3.67	4.00	3.61	**1.41**
Shap	3.15	3.45	3.71	3.19
Lore	**2.09**	**2.36**	**2.61**	**1.41**

method	compas	adult	german
Anchor	4.48 ± 6.43	101.60 ± 203.72	2.81 ± 0.84
Lime	1.49 ± 0.24	3.10 ± 0.83	**0.20 ± .03**
Maple	743.62 ± 0.02	34643.04 ± 0.01	273.08 ± 0.02
BRL	53.20 ± 0.01	621.20 ± 0.68	33.10 ± 0.02
Shap	**0.29 ± 0.31**	**0.46 ± 0.60**	0.86 ± 0.15
Soc	3.52 ± 0.02	39.18 ± 2.80	4.72 ± 0.07
Lore	8.02 ± 0.36	62.48 ± 4.55	7.76 ± 0.19

Table 6.6: Running time (mean ± stdev) in secs for SVM.

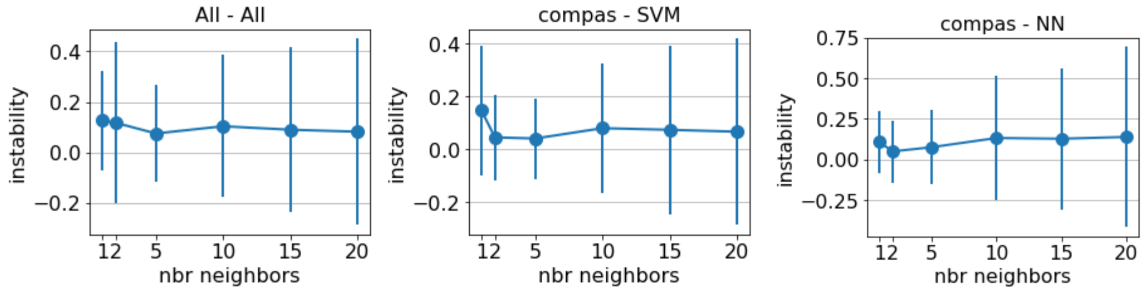


Figure 6.10: Instability metric by varying the number N of decision trees in LORE$_{sa}$.

Figure 6.10 shows how instability behaves varying the number N of local neighborhoods/decision trees generated by LORE$_{sa}$. Similar results are obtained for LORE$^{d}_{sa}$. There is a (local) minimum at $N = 5$, which is the value set by default in LORE$_{sa}$. Finally, with respect to the *instability*$_{si}$ metric[20], we point out that BRL and Maple are deterministic methods, hence the metric does not apply to them. Shap, which has the best performances, bases its explanation process on permutations of x with respect to a set of *base values*. Using a single background value as the medoid of the training set, as suggested in SHAP tutorials' can markedly limit the variability of the permutations of x. This explains the low *instability*$_{si}$ value. On the other hand, different background values could lead to different explanations [177, 178].

In Table 6.4 (bottom) we report the *coverage* and *precision* metrics for the rule-based explainers under analysis. Furthermore, to capture both measures with a single value, we also report the *harmonic mean (h-mean)* of coverage and precision. We notice that, LORE$_{sa}$, LORE$^{d}_{sa}$ and Lore overcome Anchor and BRL for both indicators. LORE$_{sa}$ considerably improves the rule coverage paying something in precision; however, looking at the *h-mean* LORE$_{sa}$ is the best performer. This is another beneficial effect of the bagging-like approach, which improves on generality (less overfitting) of the interpretable predictor. A Friedman

[20] Differently from *instability*, the *instability*$_{si}$ metric is not normalized – see (6.1), (6.2). Hence, the columns for the two metrics in Table 6.4 (top) cannot be compared to each other.

method	ovr	adult	bank	churn	compas	compas-m	fico	german	iris	wine-r	wine-w
Anchor	3.12	3.08	3.24	3.24	3.05	3.11	3.11	3.80	4.14	3.02	2.82
brl	3.72	3.65	3.87	3.53	4.36	3.79	3.38	3.56	4.01	3.91	3.50
Lime	3.74	3.10	4.08	3.88	4.53	4.15	3.54	2.81	4.15	3.62	3.90
Maple	3.46	3.54	3.21	2.70	3.36	4.26	3.85	2.11	3.48	4.41	3.92
Shap	3.49	3.27	2.85	3.57	3.45	3.89	4.62	3.31	3.52	3.67	3.51
lore$_{sa}$	**2.19**	**2.12**	**2.27**	**2.32**	**2.12**	**2.02**	**2.21**	**2.22**	**2.82**	**2.35**	**2.17**

Table 6.7: Mean rank of explainers by dataset over all combinations of black-boxes and metrics (*silhouette*, *fidelity*, *complexity*, *instability*).

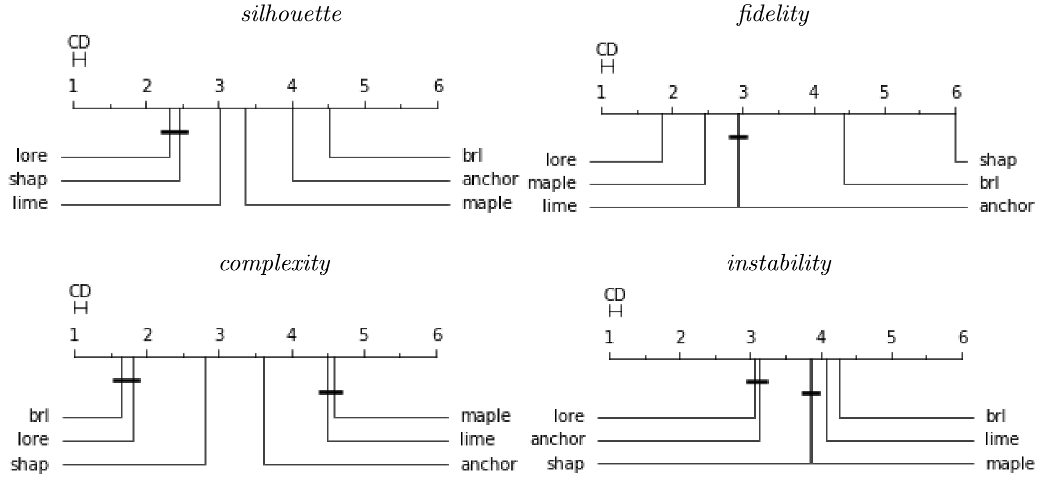


Figure 6.11: Critical difference diagrams using the Nemenyi test at $\alpha = 0.05$. The name Lore in the plots indicate the lore$_{sa}$ method.

test [184] on each of the metrics rejects the null hypothesis of zero difference among the methods (*p-value* $< 10^{-5}$). In the following, a small parenthesis on the Friedman test is presented.

Statistical tests Tables 6.7 and 6.5 report the mean rank values (ranging from 1 to 6) among the different explainers for a given dataset (resp., black-box) over all combinations of black-boxes (resp., datasets), and of the evaluation metrics of *silhouette*, *fidelity*, *complexity*, and *instability*. The first column of Tables 6.7 reports the overall mean rank. It is readily checked that lore$_{sa}$ ranks the best in general (p-value < 0.001 using a Wilcoxon signed rank test), for each dataset, and for each black-box. For the `compas-m`, `bank` and `fico` datasets and for the RF and SVM black-boxes, lore$_{sa}$ ranks markedly higher than the competitors. Figure 6.11 shows four Critical Difference (CD) diagrams [184]. They dis-

	Soc	BF	LORE$_{sa}$	LORE$^{d}_{sa}$
dissimilarity	0.170 ± 0.27	**0.056 ± 0.07**	*0.093 ± 0.03*	0.111 ± 0.00
MDM	0.166 ± 0.28	0.067 ± 0.08	*0.026 ± 0.00*	**0.019 ± 0.01**
IsoFor	1.074 ± 0.09	1.221 ± 0.36	**1.007 ± 0.00**	*1.060 ± 0.07*

Table 6.8: Aggregated evaluation metrics estimating the proximity of the counterfactual explanations in terms of dissimilarity and the plausibility as MDM and IF scores. The lower the better for all the measures: in **bold** the best performer, in *italic* the runner up.

play the statistical significance of the observed paired differences in performances between pairs of the explanation methods. Two methods are tied if the null hypothesis that their performances are the same cannot be rejected using the Nemenyi test at α=0.05. LORE$_{sa}$ performs better than the compared methods with regards to fidelity, and the differences are statistically significant. For each of the other metrics, the method tied to LORE$_{sa}$ is always a different one. Hence, LORE$_{sa}$ wins over any other method in at least 3 out of the 4 metrics.

Table 6.8 compares LORE$_{sa}$ with the merging variant and with two competitors with respect to the counterfactual part of the explanation. We highlight that LORE$_{sa}$ is not an explainer directly returning counterfactual instances on its own. However, counterfactual instances can be created by modifying the instance under analysis x according to the counterfactual rules in Φ. We notice that the brute force approach BF has the lowest dissimilarity but LORE$_{sa}$ and LORE$^{d}_{sa}$ achieve closer results. Soc is the worst performer with respect to this metric, meaning that the counterfactual instances returned by Soc are not highlighting minimal changes with respect to x to change decision. Furthermore, LORE$_{sa}$ alternatives return the most plausible counterfactuals with respect to the the MDM and IsoFor metrics. There is not a clear winner but overall the plausibility scores of LORE$_{sa}$ are better being always the best performer or the runner up, i.e., lower, than those of BF and Soc, enabling it to be used also as a possible counterfactual explainer.

In Table 6.9, we compare LORE$_{sa}$ with the counterfactual explainer Soc that is typically used as a baseline [69]. Mean and standard deviations are reported for the number of counterfactual instances (Soc) or counterfactual rules (LORE$_{sa}$) produced, and the changes metrics (number of changes to instance x to revert the black-box outcome). For all the reported datasets and black-boxes, LORE$_{sa}$ produce less changes than Soc. On the other hand, Soc returns more counterfactuals. The number of counterfactuals returned by LORE$_{sa}$ could be increased trading off with changes, simply by relaxing the requirement of minimality in Algorithm 3. Let us now denote with LORE$_{s\underline{a}}$ with underlined $\underline{\text{a}}$ the execution of LORE$_{sa}$ with in input dataset-specific constraints U stating features that cannot be changed: *age*, *race*, *sex*, *native-country*, *marital-status* for `adult`; *age* for `bank`; *state*, *state-area*, *state* for `churn`; *age*, *age-cat*, *race*, *sex* for `compas-m` (shown as `cps-m` in the table). As expected, it turns out that LORE$_{s\underline{a}}$ produces less counterfactuals its counterpart

X	Explainer	*no. cf.*	*changes*	b	Explainer	*no. cf.*	*changes*
adult	Soc	9.6 ± 1.0	3.8 ± 2.7	DNN	Soc	9.9 ± 0.7	2.6 ± 1.7
	LORE$_{sa}$	2.9 ± 2.7	1.3 ± 0.5		LORE$_{sa}$	2.4 ± 1.6	1.2 ± 0.5
	LORE$_{s\underline{a}}$	1.8 ± 1.7	2.2 ± 0.4		LORE$_{s\underline{a}}$	1.2 ± 0.4	2.5 ± 0.5
bank	Soc	3.5 ± 2.4	1.6 ± 0.6	NN	Soc	7.2 ± 2.4	5.0 ± 3.3
	LORE$_{sa}$	1.4 ± 0.9	1.3 ± 0.5		LORE$_{sa}$	2.5 ± 2.4	1.3 ± 0.5
	LORE$_{s\underline{a}}$	1.6 ± 0.8	1.5 ± 0.2		LORE$_{s\underline{a}}$	1.4 ± 0.9	2.2 ± 0.4
churn	Soc	8.4 ± 1.8	5.8 ± 3.7	RF	Soc	7.5 ± 2.3	3.6 ± 2.7
	LORE$_{sa}$	2.0 ± 1.9	1.5 ± 0.7		LORE$_{sa}$	2.4 ± 2.1	1.3 ± 0.6
	LORE$_{s\underline{a}}$	1.5 ± 0.9	2.3 ± 0.5		LORE$_{s\underline{a}}$	1.9 ± 1.2	2.2 ± 0.5
cps-m	Soc	5.2 ± 1.8	2.9 ± 1.4	SVM	Soc	6.9 ± 3.5	3.2 ± 2.7
	LORE$_{sa}$	3.5 ± 2.2	1.1 ± 0.3		LORE$_{sa}$	3.0 ± 2.3	1.2 ± 0.5
	LORE$_{s\underline{a}}$	1.8 ± 1.1	1.3 ± 0.2		LORE$_{s\underline{a}}$	1.6 ± 1.1	2.2 ± 0.4

Table 6.9: Aggregated evaluation metrics for counterfactuals over experimental datasets and black-boxes. LORE$_{s\underline{a}}$ is LORE$_{sa}$ with constraints U in input.

Algorithm 3: *extractCounterfactuals*(c, r, x, U)

```
Input  : $c$ - decision tree, $r$ - rule, $x$ - instance to explain, $U$ - constraints
Output: $\Phi$ - set of counterfactual rules for $p$
$Q \leftarrow getPathsWithDifferentLabel(c, y)$;                // get paths with $y' \neq y$
$\Phi \leftarrow \emptyset$; $min \leftarrow +\infty$;          // initialize counterfactual set
for $q \in Q$ do
    if not $q \rightarrow U|_q$ then
        continue;                                               // skip rule if constraints not satisfied
    $qlen \leftarrow nf(q, x) = |\{sc \in q \mid \neg sc(x)\}|$
    if $qlen < min$ then $\Phi \leftarrow \{q \rightarrow y'\}$; $min \leftarrow qlen$;
    else if $qlen = min$ then $\Phi \leftarrow \Phi \cup \{q \rightarrow y'\}$;
return $\Phi$;
```

ignoring the actionability. This is due to the filtering of the counterfactual rules that do not satisfy the feature constraints. On average, such counterfactual require more changes to the instance x to explain, but still less than Soc.

6.5 Discussion

In this Chapter, we introduce LORE$_{sa}$, a method for tabular local explanations that is independent of the underlying black-box model. LORE$_{sa}$ generates informative and factual decision rules as well as actionable counterfactual rules.

Through an exhaustive experimental evaluation comparing LORE$_{sa}$ with state-of-the-art methods, we demonstrate significant improvements in the stability of explanations.

Additionally, LORE$_{sa}$ consistently ranks among the top performers or runners-up in various quantitative metrics. The stability of the explanations is achieved by employing a novel bagging-like approach that generates and aggregates multiple local decision trees.

For future work, we envision several directions to expand the applicability of LORE$_{sa}$. Firstly, we acknowledge that synthetically generated instances may not fully capture correlations among attributes, such as age and education level. To address this, we propose integrating domain knowledge, such as dependencies or causal relationships, into the generation of neighborhood instances and the inference process of the interpretable predictor.

Secondly, in the context of multi-class problems, it is worth exploring alternative definitions of $\mathit{fitness}_{\neq}$ to guide the selection of counterfactual rules towards specific class values. For instance, in a credit risk rating scenario, there might be a need to provide counterfactuals that lead to a lower risk label.

Chapter 7

Summary of Part II

In this Part, we presented a benchmark of the most popular Explainable AI methods for local, post-hoc and tabular explanations. This analysis highlighted several limitations of the state-of-the-art algorithms. In particular, the explainable methods available are unstable, slow and difficult for a non-expert user to understand. For this reason, we proposed LORE$_{sa}$, a black-box agnostic method for local explanations providing informative, factual decision rules and actionable counterfactual rules. An exhaustive experimental evaluation with state-of-the-art methods has shown that LORE$_{sa}$ largely improves as per stability of explanations while ranking top or runner-up in several other quantitative metrics. The stability of the provided explanations is achieved by adopting a novel bagging-like approach in generating and aggregating several local decision trees. In addition, the LORE$_{sa}$ approach holds significant importance not only in terms of enhancing stability and actionability but also in its suitability for a general audience without expertise in computer science and mathematics. One noteworthy aspect is that the explanations provided by the approach are expressed in the form of logical rules, resembling the familiar human logic of ”if-then-else.” This characteristic renders them easily comprehensible even for individuals without technical knowledge. This is a crucial consideration as many existing explanation methods in the literature shows good performance in terms of quantitative metrics but are challenging for non-experts to grasp due to their reliance on complex mathematical formulations. Consequently, a notable advantage of LORE$_{sa}$ lies in its ability to generate explanations that are accessible and understandable to a wider range of individuals.

Several potential directions can be identified for future work to broaden the applicability of LORE$_{sa}$. Firstly, it is worth considering the integration of domain knowledge, such as dependencies or causal relationships among attributes, into the generation of neighborhoods and/or the inference process of the interpretable predictor. This would address the issue of synthetic instances not accurately reflecting correlations among attributes, for instance, between age and education level.

Secondly, in the context of multi-class problems, alternative definitions of $\mathit{fitness}_{\neq}$ could be explored to guide the selection of counterfactual rules towards specific class values. For example, in the domain of credit risk rating, it may be necessary to generate counterfactuals aimed at achieving a lower risk label.

Another interesting area for future research involves adapting LORE$_{sa}$ to handle images and texts. The binary encoding approach, which models the presence or absence of super-pixels/words, encounters similar challenges as encountered by Lime, resulting in the generation of unrealistic synthetic instances. To overcome this limitation, more sophisticated encodings, such as those utilizing autoencoders, can be employed to generate realistic neighborhoods of images and texts (Guidotti et al., 2019).

Lastly, it is important to note that LORE$_{sa}$ assumes unrestricted querying of the black-box model. However, when limitations on the number of admissible queries are in place, such as in real-world scenarios, the neighborhood generation phase must account for these constraints. In such cases, an active learning variant of the genetic approach can be adopted to optimize the limited number of queries.

Part III

Predicting and Explaining the Privacy Risk

This Part presents our research on the interplay between Privacy and Explainable AI, with a focus on utilizing explanations to enhance users' awareness on privacy risks.

Data Privacy is a significant concern in the digital age due to the vast amounts of personal information that organizations, governments, and individuals collect and store. An area of growing research interest is big data analytics, including mobility data analytics, which is crucial for new knowledge-based services and applications. However, the use of this type of personal data raises concerns about potential leakage of sensitive information [104]. For example, analyzing mobility data can reveal details of individuals' private lives [185]. Similar privacy issues arise in other contexts, such as supermarket purchases or weblogs. To mitigate privacy risks while retaining the valuable data characteristics useful for Data Mining and Machine Learning application, researchers have developed tailored privacy protection techniques [186] [187] [188] [3]. For enabling a practical application of these techniques, Pratesi et al. proposed PRUDEnce [2], a framework for systematic individual privacy risk assessment based on personal datasets, presented in Section 3.2.1. PRUDEnce assists data controllers in complying with the EU GDPR (Section 4). It is based on the idea that the privacy risk assessment can be performed by simulating an attacker who wants to exploit a privacy attack on a dataset. At this point, the privacy risk value associated with a user is given by the probability of success of that privacy attack. PRUDEnce assumes a worst-case scenario approach for the privacy risk computation, and therefore, it evaluates all the possible background knowledge configurations for a potential adversary generating them with a combinatorial approach directly from the data of a user. After an exhaustive evaluation of all possible attack configurations, it computes the maximum risk of re-identification (or privacy risk). Therefore, while the framework provides a comprehensive methodology for worst-case privacy risk assessment, its computational complexity is high. Moreover, PRUDEnce is designed to support data providers (companies) in identifying portions of data with high privacy risk by simulations of the attacks. The computation requires the availability of the entire dataset, like that stored in the servers of the companies. The high computational complexity becomes a non-negligible practical limitation in some online user-centric applications where it is useful to have a continuously up-to-date indicator of privacy exposure. In other words, PRUDEnce is not suited for providing personalized recommendations in terms of risks associated with sharing personal trajectories. Indeed, for any new user requiring risk evaluation, the system should re-compute the privacy risk against the whole dataset. Moreover, it does not provide any explanation of the privacy risk derived by the system. In user-centric applications, providing users with an explanation of the reasons for the identified privacy risk might contribute to raising their self-awareness.

In this thesis, to overcome the computational complexity drawback and to enhance users' awareness, we propose Expert, an EXplainable Privacy ExposuRe predicTion framework that exploits *(i)* machine learning (ML) models for predicting a user's individual privacy risk and *(ii)* local explainers for producing explanations of the predicted risk.

This framework is modular and can be tailored to specific data input and explanation

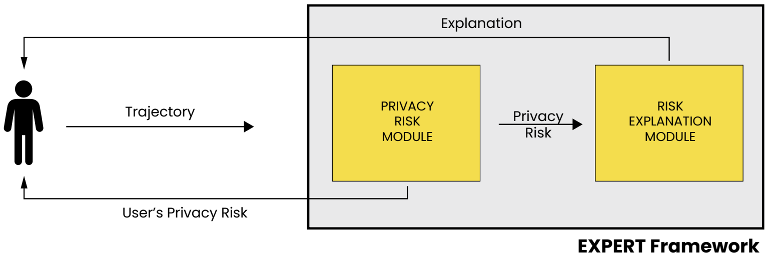


Figure 7.1: The general structure of the proposed framework EXPERT.

requirements to achieve desired outcomes. By utilizing the EXPERT framework, users can gain a better understanding of privacy risks associated with their data, allowing them to take appropriate measures to protect their privacy.

Figure 7.1 depicts the architecture of EXPERT, which is composed of two main modules: the *privacy risk prediction* module, which takes as input the user's data and, exploiting a trained ML model, predicts the privacy risk level of that user, and the *explanation* module which produces the explanation of the predicted risk.

For the prediction task, EXPERT has to exploit ML models able to handle the class-imbalance problem effectively; this is because in the context of privacy risk prediction, two scenarios are frequent: *(i)* most of the users in the data have a low privacy risk due to the simulation of an attack based on a limited background knowledge; or *(ii)* most of the users have a high privacy risk typically due to a strong background knowledge possessed by the adversary. However, in any case, the objective of EXPERT is to have a predictor that preserves the privacy of risky users while providing the freedom of using data-driven services to users with low privacy risk.

In this thesis, we instantiate this framework for assessing and explaining privacy issues in mobility data. We present different variants of the framework, depending on the kind of Machine Learning models exploited, as well as the Explainable AI techniques considered. In Chapter 8, we present EXPERT for human mobility profiles, which are expressed by a tabular setting. In this case, we first derive some indicators describing human mobility from raw trajectories. Then, we apply some Machine Learning models tailored for this kind of data to predict the privacy risk, as well as Explainable AI methods, such as LORE and LIME. Then, in Chapter 9, we present a variant that works directly on human trajectories in their raw format. In other words, the system directly works on sequential data. This variant is designed to optimize prediction performance by exploiting a customized Machine Learning structure for providing fast and accurate predictions, as well as a visualization tool for better exploring the explanation obtained. The work presented in this Part is published in [189–191].

Chapter 8

EXPERT for Mobility Profiles

In this chapter we present how to design and develop Expert for predicting and explaining the privacy risk of mobility profiles, i.e., sets of mobility indicators describing and summarizing the mobility behavior of an individual. In the last 10 years, in the field of mobility data analytics important effort has been dedicated to the mathematical modelling of the main aspects of human mobility dynamics [192,193] and the use of mobility features for studying and understanding various phenomena such as pollution [194], well-being of a region, public health [195], and so on. However, most of the mobility indicators describe characteristics of the mobility at individual level and thus, this can lead to risk of re-identification. As a consequence, the privacy risk assessment has to be conducted also in case the analytical framework leverage mobility indicators and thus, does not access directly raw trajectory data. In this particular context, first, Expert extracts from human mobility data an individual mobility profile describing the mobility behavior of any user. Second, for each user it exploits PRUDEnce to compute the associated privacy risk. Third, it uses the mobility profiles of the users with their associated privacy risks to train a ML model. Finally, for a new user, along with the prediction of risk, Expert also provides an explanation of the predicted risk. Expert exploits two state-of-the-art explanation techniques, i.e., Shap [41] and Lore presented in Chapter 6. Thus, the ML model is the result of several steps: *(i)* the empirical computation of the individual privacy risk, *(ii)* the extraction of individual profiles from human mobility data, summarizing users' mobility behavior, and *(iii)* the training of a ML model.

8.1 Learning a Prediction Model for Privacy Risk of Mobility Profiles

The objective of our approach is to train a ML model to predict the privacy risk level of users based solely on their individual mobility profile. Thus, given a human mobility dataset of n user trajectories, we propose to derive the training dataset $\langle M, \Gamma \rangle$, where M is a set of n individual mobility profiles, and Γ is the vector of their associated privacy risk levels. Since, the privacy risk is related to a specific attack (see Section 3.2.1), the procedure for *building a training dataset* depends on the adversary attack modelling. As a consequence, given a specific attack, characterized by a background knowledge configuration B_h, the procedure performs the following two steps:

- *Mobility Profile Extraction*: Given a mobility dataset $\mathcal{D}$, for every user trajectory T_u we propose to extract a mobility profile to characterize her mobility behavior. To this end, we propose to derive a set of well-known mobility features (presented in the next section). We denote by $M_u \in M$ the mobility feature vector of a user.

- *Privacy Risk Computation*: For each user u a privacy risk value is computed by simulating an attack with background knowledge configuration B_h on the mobility dataset $\mathcal{D}$. Since the goal is to predict the privacy risk level, the privacy risk vector is discretized to get a set of risk classes[1], and the vector of n user's privacy risk levels Γ.

After the execution of the above two steps, we get a training set $\langle M, \Gamma \rangle$. The derived training dataset $\langle M, \Gamma \rangle$ is used to train a predictive model which will be used within Expert to immediately estimate the privacy risk level of previously unseen users, whose data were not used in the learning process. Clearly, in prediction time, in order to predict the privacy risk of a new trajectory instance the process requires, first the computation of the mobility profile for that user and then, the application of the predictive model. Among the different ML methods, we propose to employ models able to handle classification tasks with imbalanced data. Indeed, as we show in our experiments, one of the characteristics of our training data is that most of the users have high privacy risk. Our goal is to get a predictor able to guarantee the privacy protection of risky users while providing the freedom of using data-driven services to users with low privacy risk. Thus, the optimal predictor should be characterized by a low probability of misclassifying a high risk user as a low risk one, while maintaining also good performance with respect to the classification of low risk users. In this section, we propose to apply the GcForest model [196], a decision tree ensemble approach with performance highly competitive to deep neural networks in a broad range of tasks. It is especially suitable to handle highly extra-imbalanced data [197]. GcForest relies on multiple layers of parallel forests of trees whose output is then concatenated to re-represent data to subsequent layers. In our experiments we compare GcForest against models such as decision tree, logistic regression, and random forest.

[1] In our experiments we discretize the risk in two main classes: low risk (privacy risk ≤ 0.5) and high risk (privacy risk > 0.5).

8.1.1 Mobility Profile Extraction

The goal of this step is to construct the matrix M representing the set of individual mobility profiles, expressed by a set of mobility features that describe and summarize the mobility behavior of an individual. In our setting, we employ measures widely used in the literature [5, 198]. Some of them describe only the mobility behaviour of an individual, while others describe an individual mobility behaviour in relation to collective mobility characteristics. Table 8.1 reports all the mobility measures used in the study. First of all, we define V as the number of visits of a user, it corresponds to the total number of locations in the user's trajectory. To quantify the erratic behaviour of a user during the day we compute the average number of daily visits $\overline{V}$, dividing V by the total number of days in the period of observation. *Locs*, instead, is the number of distinct locations visited by a user during the period of observation, while $Locs_{ratio}$ represents the fraction of locations covered by a user. We compute it by dividing *Locs* by the total number of locations available in the territory. We also evaluated some measures about the distances travelled by the users. We define D_{max} as the maximum distance travelled by each user, i.e. the longest trip for each user. This measure is then employed for the computation of D_{max}^{trip}: it is the ratio between the maximum distance travelled D_{max} and the maximum distance that is possible to travel in the area of observation. We also consider D_{sum}, i.e., the sum of all the distances travelled by a user. This value is then used in the definition of $\overline{D_{sum}}$, which is the average of D_{sum} over the period of observation (expressed in days). We also consider the *radius of gyration* [199] representing the characteristic distance travelled by a user during the period of observation and is defined as $r_g = \sqrt{\frac{1}{N}\sum_{i \in L} w_i(r_i - r_{cm})^2}$, in which $i \in L$ is the visited location by a user, w_i represents a user's frequency of visits at a location i, r_i denotes the geographical description of the location i and it is a bi-dimensional vector, while r_{cm} is the center of mass of the user under consideration. Mathematically, the latter is defined as $r_{cm} = \frac{1}{V}\sum_{1 \in L} r_i$. We also measure the *mobility entropy* E as the predictability of a user's trajectory. We employ the Shannon entropy measure [200]: $E = -\sum_{i \in L} p_i \log_2 p_i$, in which p_i is the probability of the location i for the user under analysis. For each user, we also consider three locations that characterize a user's mobility: the most visited location, the second most visited location and the least visited location. Typically, the most visited location corresponds to user's home, while the second most visited location is users' work place. For each one of these locations, we evaluate the frequency of visits during the period of observation w_i, where i represents the specific location under analysis. We also define $\overline{w_i}$ as the daily average of the frequency of visits at the location i for the user under analysis. Then, we denote by w_i^{pop} the frequency of visits divided by the popularity of the location, i.e. the total frequency of the location in the dataset. In this way, we normalize the frequency of the user for a particular location considering the behaviour of all the users in the dataset. For these three locations, we also consider U_i, i.e., the number of distinct users that visited the location i in the period of observation. Out of U_i, we also compute U_i^{ratio}, in which

Notation	**Description**	**Notation**	**Description**
V	visits	$\overline{V}$	daily visits
D_{max}	max distance	D_{sum}	sum distances
D_{max}^{tot}	max distance over total max distance for a user	$\overline{D}_{sum}$	D_{sum} per day
D_{max}^{trip}	D_{max} over area	$Locs$	distinct locations
$Locs_{ratio}$	$Locs$ over area	R_g	radius of gyration
E	mobility entropy	E_i	location entropy
U_i	individuals per location	U_i^{ratio}	U_i over individuals
w_i	location frequency	w_i^{pop}	w_i over the total frequency of location i
$\overline{w}_i$	daily location frequency	PT_j	Path time per user

Table 8.1: Mobility features of the individual mobility profile.

the number of distinct users that visited the location i is divided by the total number of users in the dataset. The last measure we consider for each of the three locations is the entropy. In this case, we compute a *location entropy* E_i, that represents the predictability of a visit at the location i defined as: $E = -\sum_{u \in U_i} p_u \log_2 p_u$, where U_i is the set of users that visited the location i and p_u is the probability that a user u visited the location i. When working with trajectories, we have also a temporal information: each trajectory is composed by $\langle l_i, t_i \rangle$, in which t_i is the timestamp corresponding to time of arrival of a user at a location l_i. We exploit this information to compute the *path time* [198], i.e., the time occurring between the first and last visit of a trajectory.

8.1.2 Privacy risk computation

The goal of this module is to compute for each user trajectory in $\mathcal{D}$ a privacy risk value by using a re-identification attack. We propose to apply the PRUDEnce framework (Chapter 3.2.1) that enables the definition and simulation of any desired privacy attacks over the entire dataset. Several attacks might be defined on the basis of the type of background knowledge possessed by an adversary [2, 5]. We instantiate our risk computation using the location sequence attack, introduced in [187, 201], where the adversary knows a subset of the locations visited by the individual and the temporal ordering of the visits. Given an individual u, we denote by $L(T_u)$ the sequence of locations $l_i \in T_u$ visited by u. The background knowledge category of a location sequence attack is defined as follows:

Definition 4 *Let h be the number of locations l_i of an individual u known by the adversary. The Location Sequence background knowledge is a set of configurations based on h locations, defined as $B_h = L(T_u)^{[h]}$, where $L(T_u)^{[h]}$ denotes the set of all the possible h-subsequences of the elements in the set $L(T_u)$.*

We indicate with $a \preceq b$ that a is a subsequence of b. Each instance $b \in B_h$ is a location subsequence $X_u \preceq L(T_u)$ of length h. Given a record $T \in \mathcal{D}$ we define the matching function as: *matching*(T, b) = *true* if $b \preceq L(T)$, *false* otherwise. PRUDEnce uses this function to compute the probability of re-identification for any instance of background knowledge enabling the privacy risk computation for each trajectory, as presented in Section 3.2.1.

8.2 Risk Explanation Module

The last module of EXPERT is the *explainer* aiming at providing the end-user with an explanation for the predicted risk label. The objective is to increase users' awareness about the privacy risks. EXPERT is modular with respect to the explainer allowing the use of any explanation method suitable to tabular data. Since the goal is to explain a specific decision, *local* methods [13, 41, 202] are more suitable for this task. The main difference between them is the type of explanation returned. LIME [202] and SHAP [41] are mainly based on the notion of feature importance and LORE [13] instead provides a logical rule-based explanation for the prediction. In our experiments we considered LORE and SHAP as explainers. Given our ML model and an individual trajectory belonging to a user u, transformed into the mobility profile M_u and labeled with a specific privacy risk level r_u by our model, LORE builds a simple, interpretable predictor by first generating a balanced set of neighbor instances of the given M_u through an ad-hoc genetic algorithm, and then extracting from such a set a decision tree classifier. A *local explanation* is then extracted from the obtained decision tree. The local explanation is a pair composed by *(i)* a *logic rule*, corresponding to the path in the tree that explains why M_u has been labeled as r_u by the predictor, and *(ii)* a set of *counterfactual rules*, explaining which changes in M_u would invert the risk class assigned. SHAP (SHapley Additive exPlanations) is a local approach for interpreting model predictions that assigns to each feature an importance value for a particular prediction. Moreover, for each model's prediction SHAP defines an *explanation* model. The main idea is that the explanation model is an interpretable approximation of the original model and works with simplified input data. SHAP exploits the collaborative game theory to determine the importance value of a feature for the instance prediction.

8.3 Experiments

We experimentally validate the different components of our framework by analyzing the performance of: *i)* the prediction module implemented with different machine learning models by varying their complexity; and *ii)* the explanation module by comparing two state-of-the-art approaches.

8.3.1 Data

We use data containing GPS tracks of private vehicles in Tuscany (Italy) provided by Octo Telematics. We selected trajectories from an area comprising two major urban centers, Prato and Pistoia, considering the period from 1st May to 31st May 2011, for a total of 8651 distinct vehicles. We performed two different transformations of the original data in order to obtain two different datasets. In the first dataset, called ISTAT, trajectory points are generalized according to the geographical tessellation provided by the Italian National Statistics Bureau (ISTAT): each point is substituted with the centroid of the geographical cell to which it belongs. We then remove redundant points, i.e., points mapped to the same cell at the same time, obtaining 2274 different locations with an average length of 31.9 points per trajectory. With respect to the second dataset, called VORONOI, we first apply a data-driven Voronoi tessellation of the territory [203], taking into consideration the traffic density of an area, and then we used the cells of this tessellation to increase the granularity of the original trajectories. The algorithm also performs interpolation between non adjacent points[2]. We obtained 1473 different locations with an average length of 240.2 points per trajectory.

For both datasets we computed the mobility features M to extract the users' mobility profiles and the privacy risk according to the simulation of the location sequence attack (Section 8.1) with four background knowledge configurations B_h using $h = 2, 3, 4, 5$, getting four different risk datasets, $\Gamma_{h=2,3,4,5}$. We discretized the risk values in intervals: $[0, 0.5]$ and $(0.5, 1]$ named LOW and HIGH risk class, respectively. Then, we built our classification datasets merging each risk dataset with the feature-based mobility profiles: $\langle M, \Gamma_h \rangle$, as explained in Section 8.1. To better handle the imbalance in the data, we learned our predictive models using stratified sampling, undersampling and 5-fold cross-validation. Tables 8.3 & 8.2 report the class balance after under-sampling the majority class. We also performed hyper-parameter tuning by grid search in the parameter space[3].

8.3.2 Predicting the Privacy Risk

We validate the effectiveness of the prediction module of EXPERT by comparing four different ML models: Decision Tree (DT), Logistic Regression (LG), Random Forest (RF)[4], and GCForest (GCFOREST)[5]. Decision Tree and Logistic Regression are two well-known, white-box models. Random Forest and GCForest [196] are ensemble models proven to be effective when dealing with imbalanced data. This task is characterized by strong imbalance of the two risk classes, therefore being a challenging machine learning problem, where the classifier performance in terms of accuracy is less significant due to the dominance of the majority class on the metric.

Indeed, as discussed in Section 8.1, our desiderata is a classifier with a conservative approach with respect to high risk users, to avoid their misclassification as low risk users.

[2] Voronoi tessellation obtained by using: `http://geoanalytics.net/V-Analytics/`

[3] Hyper-parameter settings: `https://github.com/francescanaretto/prp`

[4] `https://scikit-learn.org/stable/`

[5] `https://github.com/kingfengji/GCForest`

B_h	Class Balance	Under-sampling	Metric	dt	lg	rf	GcForest
h=2	High=77 Low=23	High=40 Low=60	$F1_{high}$	0.92 (0.00)	0.92 (0.00)	**0.94** (0.00)	**0.94** (0.02)
			P_{high}	0.90 (0.01)	0.91 (0.01)	0.91 (0.00)	**0.92** (0.01)
			R_{high}	0.93 (0.01)	**0.96** (0.00)	0.95 (0.00)	**0.96** (0.00)
			$F1_{low}$	0.69 (0.02)	0.71 (0.01)	**0.75** (0.01)	**0.75** (0.01)
			P_{low}	0.73 (0.02)	0.77 (0.01)	0.81 (0.01)	**0.82** (0.01)
			R_{low}	0.66 (0.02)	0.42 (0.03)	**0.70** (0.09)	**0.70** (0.02)
h=3	High=93 Low=7	No under-sampling	$F1_{high}$	0.96 (0.00)	0.92 (0.00)	**0.97** (0.00)	**0.97** (0.03)
			P_{high}	0.95 (0.01)	0.94 (0.01)	**0.96** (0.00)	**0.96** (0.00)
			R_{high}	0.96 (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)
			$F1_{low}$	0.70 (0.02)	0.71 (0.01)	0.75 (0.01)	**0.79** (0.03)
			P_{low}	0.72 (0.02)	0.77 (0.03)	0.83 (0.03)	**0.84** (0.03)
			R_{low}	0.70 (0.06)	0.41 (0.03)	0.70 (0.04)	**0.74** (0.05)
h=4	High=95 Low=5	No under-sampling	$F1_{high}$	0.96 (0.00)	0.96 (0.00)	**0.97** (0.00)	**0.97** (0.00)
			P_{high}	0.96 (0.05)	0.95 (0.00)	0.96 (0.00)	**0.97** (0.00)
			R_{high}	0.97 (0.00)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)
			$F1_{low}$	0.73 (0.02)	0.70 (0.02)	0.77 (0.02)	**0.80** (0.02)
			P_{low}	0.75 (0.02)	0.80 (0.01)	**0.85** (0.02)	**0.85** (0.09)
			R_{low}	0.70 (0.01)	0.45 (0.03)	0.74 (0.05)	**0.76** (0.03)
h=5	High=96 Low=4	No under-sampling	$F1_{high}$	0.96 (0.04)	0.96 (0.00)	**0.97** (0.00)	**0.97** (0.00)
			P_{high}	0.96 (0.04)	0.95 (0.00)	**0.97** (0.00)	**0.97** (0.00)
			R_{high}	0.96 (0.01)	**0.98** (0.00)	**0.98** (0.00)	**0.98** (0.00)
			$F1_{low}$	0.73 (0.03)	0.70 (0.03)	0.78 (0.02)	**0.80** (0.02)
			P_{low}	0.72 (0.03)	0.80 (0.05)	0.83 (0.02)	**0.85** (0.02)
			R_{low}	0.70 (0.03)	0.46 (0.03)	0.75 (0.04)	**0.76** (0.03)

Table 8.2: Predictive models evaluation on mobility profiles derived from Istat.

On the other hand, we aim at achieving high precision and recall for both high and low risk users. As a consequence, for the performance evaluation of the machine learning models, we select the following indicators: *i)* precision (P_{high}) and recall (R_{high}) on high risk; *ii)* precision (P_{low}) and recall (R_{low}) on low risk; and *iii)* the two corresponding *F1-Score* for low ($F1_{low}$) and high ($F1_{high}$) risk. In a setting where the size of high risk class is larger than that of the low risk one, achieving good performance for the low risk users is difficult. The results for the two datasets are shown in Tables 8.2 and 8.3. We note that Istat represents a typical situation in the privacy context, where a high number of risky users exists. We also built Voronoi to present a balanced situation and to verify how our models behave in such a case. In general, we found that the ensemble methods have good performance in terms of both *F1-Score* on high risk and *F1-Score* on low risk. This means that these models are suitable for our target. More precisely, we observe that, although GcForest and RF have comparable performance, for Istat, that is extra imbalanced, GcForest performs slightly better than RF on the low risk class. Moreover, ensemble methods also outperform the white-box classifiers and again, their advantage is more evident in Istat;

B_h	Class Balance	Under-sampling	Metric	dt	lg	rf	GcForest
h=2	High=28 Low=72	High=30 Low=70	$F1_{high}$	0.71 (0.02)	0.65 (0.07)	0.75 (0.02)	**0.80** (0.01)
			P_{high}	0.73 (0.01)	0.73 (0.02)	0.78 (0.01)	**0.79** (0.01)
			R_{high}	0.74 (0.04)	0.77 (0.03)	0.72 (0.02)	**0.80** (0.03)
			$F1_{low}$	0.87 (0.00)	0.86 (0.01)	**0.89** (0.01)	**0.89** (0.00)
			P_{low}	0.70 (0.01)	0.89 (0.01)	0.87 (0.01)	**0.90** (0.02)
			R_{low}	0.85 (0.01)	0.82 (0.02)	**0.91** (0.01)	0.86 (0.01)
h=3	High=55 Low=45	No under-sampling	$F1_{high}$	0.88 (0.01)	0.88 (0.01)	**0.92** (0.01)	**0.92** (0.01)
			P_{high}	0.89 (0.01)	0.88 (0.01)	**0.91** (0.00)	**0.91** (0.00)
			R_{high}	0.86 (0.02)	0.89 (0.03)	**0.92** (0.01)	**0.92** (0.01)
			$F1_{low}$	0.84 (0.02)	0.82 (0.01)	**0.87** (0.01)	**0.87** (0.01)
			P_{low}	0.80 (0.02)	0.83 (0.03)	**0.88** (0.09)	**0.88** (0.01)
			R_{low}	**0.89** (0.02)	0.81 (0.02)	0.87 (0.01)	0.86 (0.01)
h=4	High=57 Low=43	High=40 Low=60	$F1_{high}$	0.91 (0.00)	0.90 (0.00)	**0.93** (0.00)	**0.93** (0.00)
			P_{high}	0.91 (0.01)	0.88 (0.00)	0.92 (0.00)	**0.94** (0.01)
			R_{high}	0.91 (0.02)	**0.92** (0.01)	**0.92** (0.01)	0.91 (0.01)
			$F1_{low}$	0.84 (0.01)	0.80 (0.01)	**0.87** (0.01)	**0.87** (0.01)
			P_{low}	0.84 (0.03)	0.84 (0.01)	**0.85** (0.01)	**0.85** (0.01)
			R_{low}	0.84 (0.02)	0.77 (0.03)	**0.88** (0.01)	**0.88** (0.02)
h=5	High=62 Low=38	High=50 Low=50	$F1_{high}$	0.93 (0.01)	0.93 (0.01)	**0.94** (0.00)	**0.94** (0.01)
			P_{high}	0.92 (0.03)	0.90 (0.01)	0.94 (0.01)	**0.95** (0.02)
			R_{high}	0.93 (0.02)	0.93 (0.02)	**0.94** (0.01)	**0.94** (0.01)
			$F1_{low}$	0.83 (0.01)	0.80 (0.03)	**0.86** (0.01)	**0.86** (0.02)
			P_{low}	0.83 (0.03)	0.83 (0.03)	**0.86** (0.03)	**0.86** (0.02)
			R_{low}	0.84 (0.03)	0.84 (0.03)	**0.87** (0.02)	0.86 (0.03)

Table 8.3: Predictive models evaluation on mobility profiles derived from Voronoi. Each metric is averaged over the 5-fold cross validation.

especially, they considerably improve the classification scores for the more difficult category of low-risk users. Indeed, we found that GcForest increases of 0.04–0.06 (0.09–0.13) points the R_{low} (P_{low}) of DT and of 0.28–0.33 (0.05–0.07) points the R_{low} (P_{low}) of LR. Clearly, these results contribute to have GcForest with the best $F1_{low}$ for every value of h, while still maintaining a conservative behaviour highlighted by the high values of recall on high risk class (R_{high}). Regarding Voronoi, we further notice that, although the data are more balanced, the ensemble methods always maintain the conservative approach for high risk users (high R_{high}) while improving the overall classification for low risk users ($F1_{low}$). Overall, these results suggest that GcForest is the most suitable option for our specific predictive task with RF as a close second one.

8.3.3 Explaining the Privacy Risk

Regarding the explanation task in our experiments, we employed Lore [13] and Shap [41]. We adopted the following experimental methodology: we selected the best models from the

Setting		Jaccard	Cohe
Top-k Features	RF GcForest	0.133 ± 0.063 0.096 ± 0.101	0.472 ± 0.393 ±
No-zero Features	RF GcForest	0.133 ± 0.063 0.165 ± 0.072	0.816 ± 0.767 ±

Table 8.4: Shap vs Lore in the Istat dataset with $h = 2$.

LORE $\overline{w}_{home}^{pop} \leq 0.36, U_{home} \leq 1722, E \leq 1.09, \overline{w}_{work} \leq 0.82 \implies HighRisk$

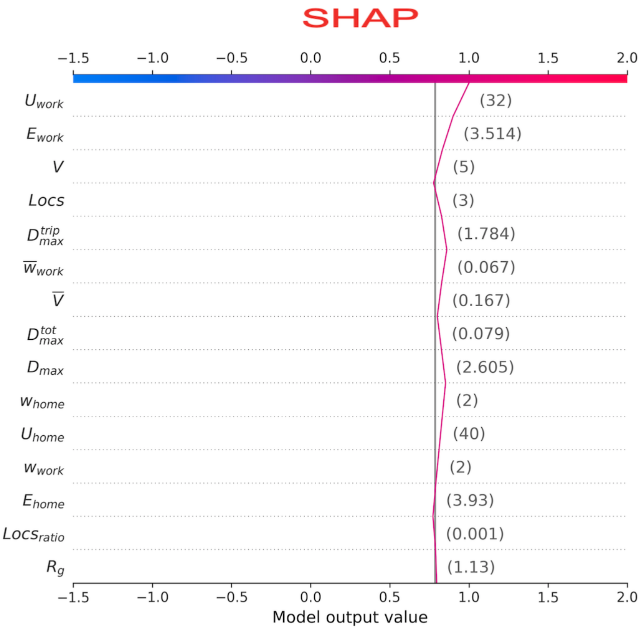


Figure 8.1: Shap vs Lore: Table 8.4 quantifies the similarity between the two explanations. Shap visualization (right) and the Lore rule (left) represent the explanations for a specific record classified as high risk by GcForest.

k-fold validation presented above and its associated train and test datasets. In particular, we used a RF and a GcForest model for $h = 2$ on the Istat dataset. For Shap we trained the *Kernel Explainer* on the training dataset. For Lore, we chose a genetic generation of the neighborhood and the Euclidean distance as distance among the neighbors. We performed a comparative analysis to evaluate the compactness and comprehensibility of returned explanations. To this end, we considered the diversity of the explanation structure provided by the two methods: Lore outputs rules with premises of variable lengths, while Shap, outputs the importance of each feature in the data. Thus, we considered two different settings: i) *no-zero features*, where in the Shap result we only keep features with importance values different from zero; and, ii) *top-k features*, that tries to automatically identify the k features with highest importance values. The value k depends on the record explanation under analysis. To detect the best k for each explanation, we used an elbow-like approach which, given the Shap result, first sorts in descending order the importance values and then, calculates the segment s bounded by the biggest and the smallest importance values. At this point, it selects the importance value m with the maximum distance from the segment s. Thus, only features with importance values greater than or equal to m are kept. For analyzing the compactness of the explanations we considered their average lengths:

Lore explanations have an average length of 2.9 ± 1.3 (RF) and 3.8 ± 1.4 (GcForest), against the average lengths of paths of the decision tree of 7.8 ± 1.5. Shap explanations have an average length of 17.1 ± 3.1(RF) and 16.2 ± 3.2 (GcForest) for the *no-zero features* setting, which decrease to 9.8 ± 6.3 (RF) and 8.3 ± 7.1 (GcForest) for the *top-k features* setting. Hence, Lore provides more compact explanations with respect to the paths of the decision tree and the Shap importance values. We also compare the two explanation types in terms of semantic coherence. To this end, we propose to use the *Jaccard similarity* to highlight the degree of common features used for the explanations and *coherence* measure aiming at capturing the percentage of features used in Lore explanations which are important also in Shap explanations. The *Jaccard similarity* measure, is defined as $\frac{1}{n}\sum_{i=1}^{n}\frac{F_i^{lore}\cap F_i^{shap}}{F_i^{lore}\cup F_i^{shap}}$ while the *coherence* is defined as $\frac{1}{n}\sum_{i=1}^{n}\frac{F_i^{lore}\cap F_i^{shap}}{|F_i^{lore}|}$. Here, F_i refers to the set of features included in the explanation for the record i. Table 8.4 reports the results of the coherence analysis. Regarding the *no-zero features* setting, we found out that the Jaccard similarity is close to zero, highlighting that the intersection of the two feature sets is quite small compared to their union. Concerning the coherence, a value equal to 1 means that all the features of Lore are also in Shap explanations. Results highlight that Shap explanations contain the majority of the features used by Lore. In the *top-k features* setting, we observe a general decrease in the values of both measures. This means that the majority of the features that Lore uses in its rules are actually among the least important features of Shap. Thus, when considering only the *top-k* features the discrepancy between Shap important values and Lore increases. Our analysis highlights that the two methods consider different important features for providing explanations. Lore explanations tend to be more compact and easy to understand due to the logic structure of the rules. Shap outputs a visualization and a large amount of information, which might potentially be difficult for a user to navigate. Indeed, a large number of the values of the importance features are close to zero. Moreover, given a feature used in an explanation, Lore provides a richer information that could help in understanding more about certain mobility habits that contribute to a specific risk value. For example, let us analyze Figure 8.1, where we provide Shap (right) and Lore (left) explanations for a high risky user according to GcForest. With Shap a user can only understand which feature (with its specific value indicated between parentheses) is important or not for classification, while the Lore rule provides a user with a more detailed motivation, which includes the set of conditions on features that a user satisfies. For example, for the Lore explanation a user can understand that their risk depends on the fact that she travelled more than 0.09 km (D_{max}), their home location is visited by less than 1772 distinct users, and their work location is not enough popular in the data. This reasoning is not supported by the Shap result. After the local explanation evaluation, we also performed a comparative analysis of global feature importance among all the ML models (Table 8.5). An interesting result is that the number of locations (*Locs*) is the most important feature for lg, dt and GcForest, while for RF

dt	**lg**	**rf**	**GcForest**
$Locs$ (0.45)	$Locs$ (0.35)	D_{sum} (0.15)	$Locs$ (0.07)
D_{max} (0.10)	E_{home} (0.14)	$Locs$ (0.13)	U_{work} (0.04)
U_{work} (0.06)	E_{work} (0.12)	$Locs_{ratio}$ (0.08)	$Locs_{ratio}$ (0.03)
$\overline{D}_{sum}$ (0.06)	W_{work} (0.10)	$\overline{D}_{sum}$ (0.07)	U_{home} (0.03)
U_{home} (0.06)	$\overline{D}_{sum}$ (0.08)	U_{work} (0.07)	D_{max}^{trip} (0.02)

Table 8.5: Global top-5 most important features of machine learning models.

it is in the second position. Moreover, LR is the only one which considers the entropy of locations (home and work) as important features.

8.4 Discussion

In this Chapter we introduce EXPERT, a framework designed to predict and explain users' privacy risks associated with the analysis of mobility data.

EXPERT exploits ML techniques that are specifically suited for handling extra-imbalanced data. In addition, it leverages local explainers to provide users with meaningful explanations regarding the predicted privacy risk.

Through an empirical evaluation using real-world data, we demonstrate the effectiveness of EXPERT in accurately predicting privacy risks and enhancing users' self-awareness regarding potentially risky mobility behaviors. However, one main limitation of the framework is that it relies on domain expertise for extracting users' profiles, which is necessary for the prediction process. In addition, the explanations provided by EXPERT are tied to the extracted features, making them challenging for non-expert users to understand.

Our future research agenda aims to address these limitations. Specifically, we plan to substantiate the prediction module by incorporating a machine learning model that does not require the extraction of mobility features. This approach will streamline the prediction process and make it more accessible to a broader range of users.

Chapter 9

EXPERT for Human Trajectories

In the previous Chapter we described EXPERT for individual mobility profiles, showing good prediction performance for the detection of the privacy risk, as well as several possibilities for the explanations to provide to the end user. However, this first variant of the framework has several limits: firstly, this approach requires the pre-processing of the trajectories, to obtain the individual mobility profiles. In addition, explaining this mobility profiles may be difficult for non experts since it exploits in depth concepts in the field of human mobility analysis. To overcome these issues, we propose an alternative version of EXPERT tailored for raw Human Trajectories. Our aim is to provide analysts with an actionable framework to predict and visualize privacy risk with an integrated explanation. The general architecture of this new variant of EXPERT is shown in Figure 9.1.

9.1 EXPERT for Trajectories: the predictive model

The EXPERT's objective is to predict the privacy risk of a human trajectory while providing the analyst with also an explanation to increase user awareness. Privacy risk is a continuous value in the interval $[0, 1]$. However, we decide to model the problem as a binary classification. Indeed, we are interested in distinguishing between HIGH risk and LOW risk users, in such a way that higher-risk users can be protected. Technically, we discretize the privacy risk obtained from the location-based attack: LOW risk or 0 (privacy risk ≤ 0.5) and HIGH risk or 1 (privacy risk > 0.5). The Γ vector generated in this way is then joined to the mobility dataset $\mathcal{D}$ and we use $\langle D, \Gamma \rangle$ to train a classification model. To avoid the problem of having to craft and compute features to be used as input data, Naretto *et al.* [190] propose to use methods applicable to raw trajectory data. In particular, we propose to address the privacy risk classification problem using state-of-the-art models, such as Long-Short Term Memory networks (LSTM), ROCKET and INCEPTIONTIME, introduced in the first Part of this thesis, in Section 14.2.2. We compare their performance

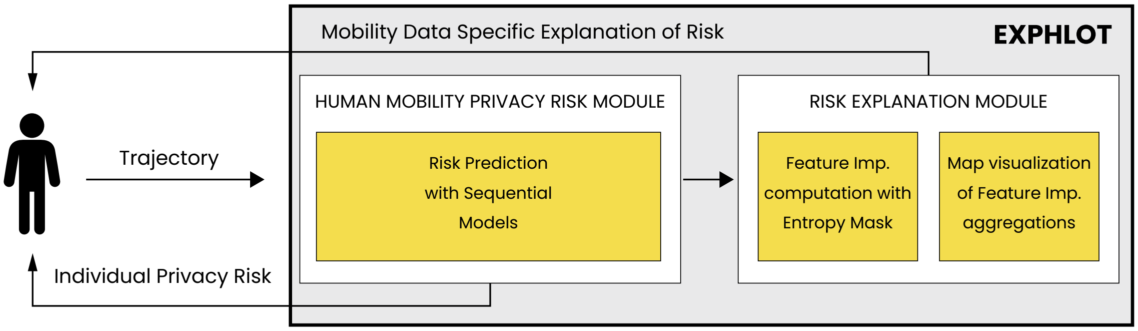

Mobility Data Specific Explanation of Risk
EXPHLOT
HUMAN MOBILITY PRIVACY RISK MODULE
Risk Prediction with Sequential Models
RISK EXPLANATION MODULE
Feature Imp. computation with Entropy Mask
Map visualization of Feature Imp. aggregations
Trajectory
Individual Privacy Risk

Figure 9.1: The general structure of the proposed framework.

and tie-efficiency to find the best method for our task. Regarding LSTM, they are a special kind of Recurrent Neural Network. They resemble the mnemonic approach of a person, in the sense that they have a memory on which they can read and write information, as well as delete the ones that are no longer needed. In this way, LSTMs are able to remember information about their inputs over a long period of time, avoiding the problem of vanishing gradients. The core idea that makes LSTM possible is that each cell is composed by a memory and three gates, each with a different purpose. Their combination makes it possible to have a memory that resembles the human one. The main intuition is that the gates protect and control the memory cell by allowing the information to pass through or not. This kind of neural networks have been considered the state-of-the-art for time series classification for many years. However, during the last years, other methods for time series classifications have become quite popular due to their extremely good performance, such as ROCKET and INCEPTIONTIME. However, LSTM are still extensively used and they are fast in time prediction, but quite slow to train, especially if compared with ROCKET. ROCKET is a fast and accurate time series classification algorithm that uses random convolutional kernels. At the operational level, we can divide the algorithm into two parts: a first part in which k randomly generated convolutional kernels are used to calculate a feature map from which, for each kernel, two aggregated features are extracted (*ppv* and *maximum value*); a second part in which the aggregated features are passed to a linear classification algorithm to obtain the actual result. The number k of kernels is the only hyper-parameter of the model. In theory, ROCKET can be used for both variable-length and fixed-length time series. To implement it for variable-length time series, however, kernels must be applied to all the time series in the dataset, i.e. kernels must be shorter than the length of the shortest time series. In the case where the length of the series varies greatly, as in our case, this approach is very inconvenient, because finding a kernel set that performs well overall and is applicable to any time series is difficult. We, therefore, chose a fixed-length approach, using

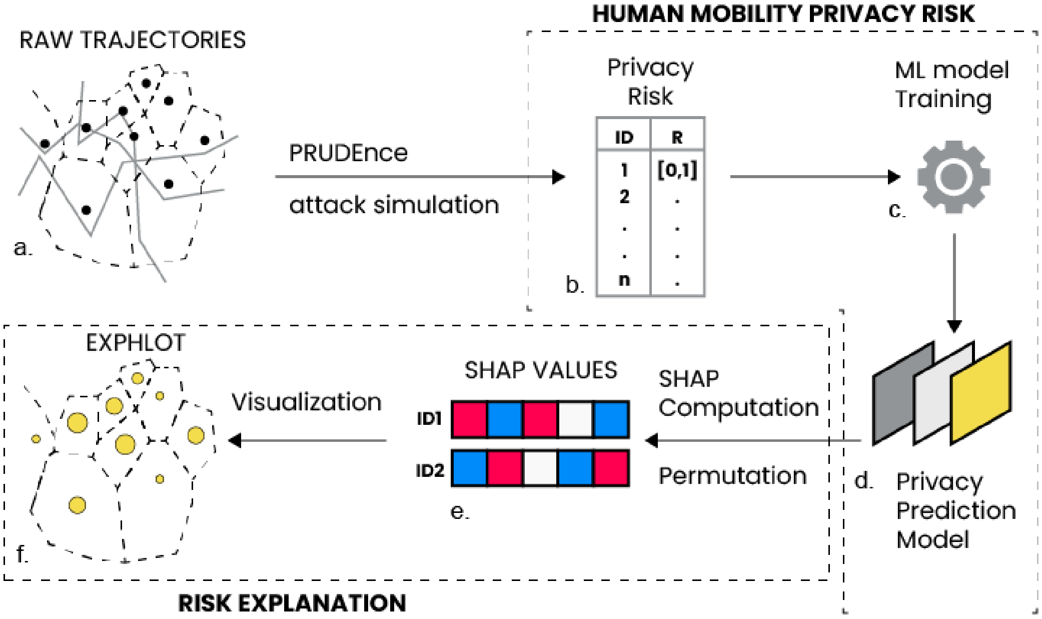


Figure 9.2: Exphlot analytical pipeline. Starting from the generalized trajectories (a) a privacy prediction model (d) is trained from a set of observations generated by a privacy risk model (b). The prediction is explained by means of Shap values (e) that are visualized within an analytical dashboard (f)

a padding that, being low amplitude or zero, keeps the result of the convolution operation on those segments close to zero and constant, and is thus cut off in the calculation of the features (*ppv* and *maximum value*). We chose Rocket over Minirocket [204] as the latter eliminates the random component in the choice of kernels' characteristics. Therefore, even though Minirocket is generally faster, we believe that a set of varied kernels fits better for our case, to capture the most diverse pattern possible. InceptionTime is an ensemble time series classification algorithm based on an ensemble of inception architectures. The Inception model is composed by convolutional layers and simultaneously applies several filters of different lengths to the input time series. This structure alleviates the vanishing gradient problem by enabling a direct flow of the gradient. It cannot be used on time series of variable length. To evaluate the performance of the models in choosing the best models and to compare them, we used a "conservative" approach. First and foremost, we wanted to protect high-risk users by preventing them from being classified as low-risk, so that their sensitive data would not be treated carelessly. Secondly, we wanted to maximize the possibility of sharing the data of low-risk users, thus preventing them from being classified as high-risk. For this reason, when choosing the models, we first took into account the recall of both classes, giving priority to the class representing *high privacy risk* (high), and the precision of both classes. The metrics considered for the models evaluation are: the overall accuracy, precision, and recall of the individual classes and F1 score.

9.2 EXPERT for Trajectories: Risk Explanation Module

For the Explanation Module of Expert our goal is to provide an explanation that is informative for both experts and users in the dominion of Human Mobility data. We chose to employ Shap to generate an attribution-based explanation for our models. Our aim is to indicate, for each individual, what parts of his movement lead to higher privacy risks. Given the nature of our specific ML models, we must employ the *Kernel Explainer*, which is the agnostic explainer of the Shap library. Clearly, depending on the size of the given data, the computation is more accurate but also longer in time. One possible solution, suggested also by the authors of Shap, is to exploit K-means clustering by selecting a large k and then feeding all the centroids obtained to the *Kernel Explainer*. In this way, we are able to represent the whole space under analysis by considering a small number of records. However, this solution for mobility data is not enough: Shap considers each location of the trajectory as a variable and for computing the Shap values all the permutations of variables are calculated as well as their relative interactions. This procedure is exponential in time if the number of variables is high, as in our case. The computation of Shap values becomes therefore unfeasible in a reasonable time. Mitchell et al. [205] propose several sampling strategies that can in theory speed up Shap values computation. However, many of the proposed strategies work under assumptions of bounds to the possible values or shape of the data. For human mobility, these bounds may not hold. For these reasons, we decide to apply the *PermutationExplainer* with a dynamic mask. This method can take as input a user-defined mask that allows certain features to be hidden, thus decreasing the individual evaluations made on these and the complexity of the calculation. In our setting, each feature corresponds to a location of the geographical map of our human mobility data. We used a binary mask to hide the features with the highest entropy, fully evaluating the locations with the lowest entropy. We formally define location entropy for each location i in the dataset with the Shannon Entropy equation: $E_i = -\sum_{u \in U_i} p_u \log_2 p_u$, where p_u is the probability that individual u visits location i and U_i is the set of all individuals visiting location i. This choice is motivated by several works in the field of privacy for mobility data. The importance of location entropy for privacy is thoroughly discussed by Rodriguez-Carrion et. al. [206], while in the work of Pellungrini et al. [5] entropy is proven to be one of the most important predictive features/locations also in ML models. The intuitive concept behind it is that location entropy is a measure of anonymity, in the sense that if a user passes through high-entropy locations, where therefore many different other people pass through, the uniqueness of his mobility profile is lost as it is blurred by the general movement. We, therefore, hide the top 70% of the highest entropy locations, evaluating only the 30% with the lowest entropy. In this way, we are focusing on those locations that have fewer individuals visiting in a more sporadic way and thus we are focusing on explaining HIGH-risk predictions. Thus, we are able to speed up the computation of the SHAP values. This is a key milestone of this work, since speeding up the process of providing an explanation makes possible to achieve an on-line interaction with the framework, one of our main objectives.

9.3 EXPERT for Trajectories: Risk and Explanation visualization module

The effective visualization of mobility properties can provide a boost to gaining deeper insights into spatial and temporal patterns. To manage the complexity of spatial resolutions, a widely adopted solution leverages spatial aggregation based on spatial partitioning [203, 207]. The process organizes close entities into groups and, for each group, a single centroid point is determined. Then the centroid points are used as seeds to partition the territory. In the scope of the current work, the data related to geography is linked to multiple dimensions and attributes, like mobility indicators, privacy risk prediction, and feature relevance. Moreover, many of these indicators may have multiple spatial scales, for example ranging from an urban building block resolution to a city district. Thus, we designed a visual interface where the set of locations of each trajectory is presented within two linked displays: a *dynamic map* with embedded graphics and a *bubble chart* (see Figure 9.10). The *dynamic map* shows for each location a visual mark, a circle, whose visual properties are linked to internal indicators of the location it represents. Each circle is driven by two visual variables, the area of the circle and the fill color, which both encode the same quantitative value. Without loss of generality, we can assume that these quantitative values are mapped to the $[0, 1]$ interval, in order to implement a pair of scale functions to determine the area and the color of each circle. The *Bubble Chart* contains the same set of circles of the map (to create conceptual links between the two displays) located accordingly to the respective values on the two axes. The user can decide which attributes are associated with which value. Any selection/filter activated on the Bubble Chart is propagated to the map (and *viceversa*).

The SHAP values are computed for every single individual trajectory. However, the domain expert is interested in the analysis of collective behavior. Thus, we aggregate the individual explanations into a global one using the aggregation procedure available within the SHAP library. This is especially important for all those instances where the data is not public or is under strict confidentiality constraints. From a geographical point of view, we considered for each location l the set of all the trajectories crossing l. For this subset of trajectories, a set of indicators is computed, such as *number of trajectories*, and *risk of re-identification*. For the latter, we compute several statistical indicators to have a compact representation of the distribution: min, max, first quartile, third quartile, median, and average.

This design achieves multiple objectives. First, it provides a user-driven exploration of the SHAP values, since the analyst can evaluate and compare the contribution of each location to the risk prediction and let the user visually identify zones containing locations with similar characteristics. Second, the possibility of navigating the map allows for a deeper investigation of local areas and provides a solution to limit cluttering when the number of locations is high. Third, geographic mapping allows a topological exploration of close locations, enabling the identification of general patterns, i.e. urban areas versus rural areas. Fourth, the expert can exploit the linked display to investigate relevant cases that are not directly evident from the map. The possibility of cross-selecting visual elements enables better identification of patterns and rules of the data.

9.4 Experiments

9.4.1 Dataset

For validating EXPERT we used GPS tracks of private vehicles, provided by Octo Telematics [208], an insurance company. We selected trajectories from the city area of Prato and Pistoia (Italy), with 8651 users observed in a period of one month, from 1st May to 31st May 2011. The data are collected by a GPS device that detects the position every 30 seconds, when the vehicle is not in motion the device automatically stops recording positions. The dataset considered is composed of one trajectory for each user. Hence, each trajectory contains all the points visited by the user in temporal order. On these trajectories, we applied a transformation, in the following called VORONOI, in which the territory is split in tiles based on a data-driven Voronoi tessellation [203]. This approach considers the traffic density of an area to create the tiles. Then, we used the cells of this tessellation to generalize the original trajectories. The algorithm applies interpolation between non-adjacent points[1]. The outliers were removed for each dataset h using a DBScan algorithm that takes into account also the label. After this pre-processing step, we obtained 1473 different locations, with an average length of 240.2 per trajectory. Given the processed dataset $\mathcal{D}$, for an in-depth validation of EXPERT, we considered *four background knowledge* configurations B_h using $h = 2, 3, 4, 5$ obtaining four different risk datasets, $\Gamma_{h=2,3,4,5}$ where, we recall, h represents the length of the background knowledge of the simulated attacker. We discretized the risk values in two classes: LOW, when the privacy risk is in the interval $[0, 0.5]$ and HIGH for the interval $]0.5, 1]$. At this point, we merged the privacy risk data with the trajectories to obtain the classification datasets for our supervised learning task, following the methodology explained in Chapter 3.2.1. Hence, we obtained 4 different datasets for our experiments. We remark that the datasets with the highest and lowest background knowledge are highly imbalanced, having the $\mathcal{D}_{h=2}$ with the 71% of users belonging to the LOW class, while for $\mathcal{D}_{h=5}$ has the 63% of records in the HIGH class. This is to be expected, as when the knowledge of the attacker is small, such as $h = 2$, the attack is less effective, having fewer people re-identified.

9.4.2 Expert Privacy risk prediction module

For all the models we split our datasets, using 80% for training and validation (10%) and 20% for testing. We selected the best hyperparameters for both models through an extensive grid search, presented in the following.

We validate the effectiveness of the privacy risk prediction for trajectory data by using the Long Short Term Memory (LSTM) [209], ROCKET and INCEPTIONTIME. The code for the experiments was written in Python 3.6 with Tensorflow 2.0 and Keras 2.2.5 [2].

This prediction task has two challenges described in the following. Firstly, the two risk classes are strongly imbalanced. Table 9.1 reports the distribution of the two classes for the VORONOI dataset. We observe different imbalance between the classes. Interestingly, $h = 2$ shows the higher imbalance with the least represented class being the *high risk* one. Another challenge is due to the length of the trajectories. Indeed, as highlighted in Figure 9.4, the trajectories considered are quite long.

In this setting our main goal is to obtain a *conservative* classifier from the point of view of users with high privacy risk. In practice, we aim at predicting correctly the users that have a high privacy risk to avoid to release highly sensitive data. At the same time, we want to give the possibility of sharing the data of people that have a low privacy risk.

[1] Voronoi tessellation obtained using `http://geoanalytics.net/V-Analytics`.

[2] Code available on `https://github.com/francescanaretto/Privacy-Risk-onMobility-Data-with-LSTMs`

Dataset	***h* = 2**		***h* = 3**		***h* = 4**		***h* = 5**	
	high	low	high	low	high	low	high	low
VORONOI	28%	72%	55%	45%	57%	43%	62%	38%

Table 9.1: Class distribution for the VORONOI dataset.

Hence, we also want to reduce the probability to classify low risk users as risky ones. For these reasons, we focused on obtaining classifiers that have a good precision and recall on the two classes.

For all the methods proposed in this section we padded the trajectories so to have a dataset with sequences of the same length.

Regarding the LSTM, we selected the network structure proposed in Figure 9.5: there are 2 layers of LSTM neurons, with 35 neurons in the first LSTM layer and 20 in the second layer. Each layer has a recurrent dropout of 0.3 and the layer of dropout is 0.2. With this architecture, we executed a hyper-parameter tuning by performing grid search in the parameter space. We considered as parameters the number of epochs, the size of the batch and the optimizer. The results obtained from the grid searches are reported in Table 9.3. iii) We applied 5-fold cross validation with stratified sampling to validate our methodology. For our experimentation we used a small dataset (8651 trajectories) and we show in Table 9.1 that it is also suffers from class imbalance. For these reasons, during the training of our models, we found several cases of overfitting, which are typical in this setting. We solved the overfitting problem for all models by introducing an early stopping criterion driven by a validation set, with patience of 4, i.e., the training of the neural network stops if no gain on validation is observed for 4 consecutive epochs. Moreover, we added dropout, both internally in the LSTM layers (0.3 for each layer) and externally, between layers (0.2). Finally, we decreased the neurons for each layer (obtaining 35 and 20 neurons, respectively) and limited the neural network complexity to two layers of LSTM (as presented in Figure 9.5). These measures allows us to avoid the overfitting of the LSTM. To further analyze the resulting models, we plot the neural network: in Figure 9.6 we presented the histograms of the weight distributions per kernel, and within each kernel, per gate. The shapes of the weights distributions in the kernel layer show a homogeneous distribution, while for the recurrent layer the gates have a Gaussian-like shape. Lastly, the bias can be found close to zero and close to one, that is what we expected, due to the setting of our problem. Overall these plots indicate that the trained LSTM does not overfit[3].

For the application of ROCKET, we choose to generate 10000 *random convolutional kernels*, which is the maximum number of kernels allowed to obtain an improvement in prediction performance while avoiding an increase in computational time. We selected the fixed length approach, with kernels of length $\{2, 7, 9, 11\}$. In principle, every classifier can

[3] The analysis of the LSTM has been performed with the `see-rnn` package: `https://github.com/OverLordGoldDragon/see-rnn`

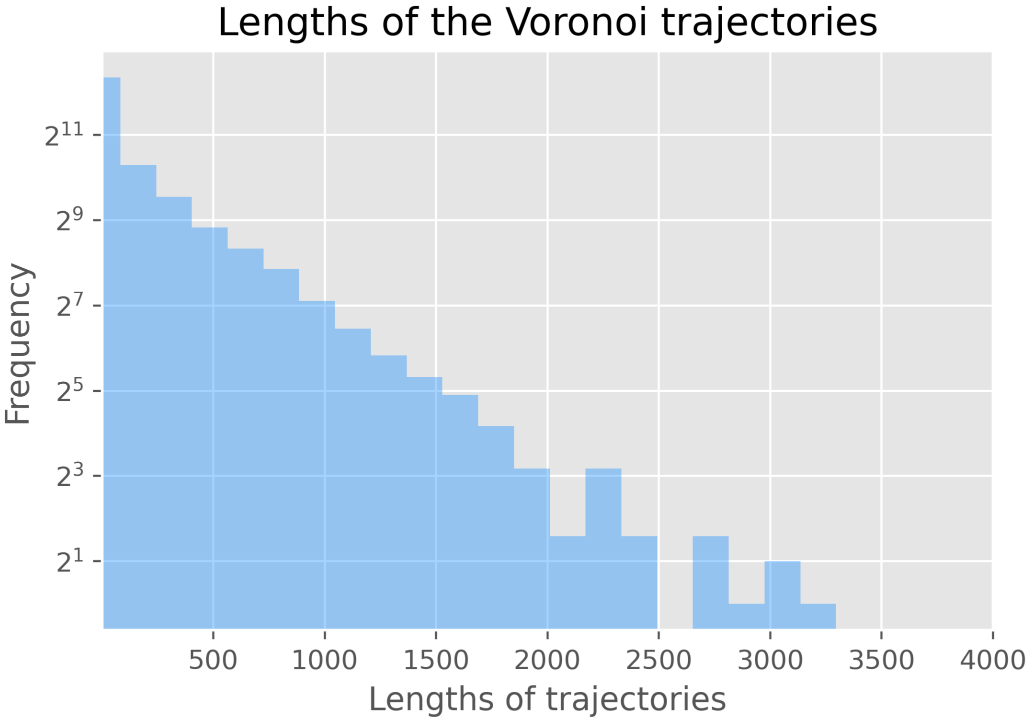


Figure 9.3: Distribution of the lengths of the trajectories for the VORONOI dataset.

Figure 9.4: This plot shows the log-scale frequency distribution of the lengths of the trajectories for the VORONOI dataset.

be applied, but we limit our analysis to linear models, as in the original paper of ROCKET. To find the best hyper-parameters for each of the four configurations of our dataset, we conducted an extensive grid search, considering: scaling the dataset, with Standard Scaler or Normalization, balancing the dataset, with over or under sampling techniques and several linear classifiers, such as the Ridge or the SDG classifier. The result of this analysis is reported in Table 9.2.

Regarding INCEPTIONTIME, we applied normalization to all the datasets which improves performances in all settings. In addition, we estimated parameters by grid search and found that the default ones worked better for all the datasets except $\mathcal{D}_{h=2}$, the one with the highest imbalance between the classes. Therefore, for $\mathcal{D}_{h=2}$ we have $batch\_size = 16$, $nb\_filters = 32$, $depth = 9$, $kernel\_size = \{8, 4, 2\}$, $use\_bottleneck = True$, $use\_residual = False$. For the other datasets we maintain defaul parameteres $batch\_size = 64$, $depth = 6$,

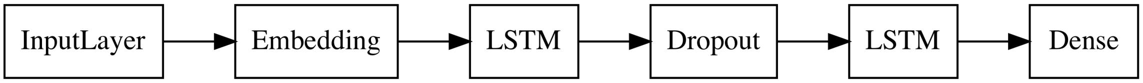


Figure 9.5: The structure of our neural network. We choose two lstm layers with a Dropout layer in between, to avoid overfitting.

Dataset	Scaling	Rebalancing	Model	Hyperparam
$h = 2$	Std	RandomUnderSampler 60% (0) - 40% (1)	RidgeClassifier	$alpha = 10$, $class\_weight$ =None
$h = 3$	Original	Original	RidgeClassifier	$alpha = 10$, $class\_weight$ =None
$h = 4$	Norm	Original	SGDClassifier	$alpha = 0.46$, $class\_weight$ =balanced
$h = 5$	Std	RandomOverSampler 40% (0) - 60% (1)	SDGClassifier	$alpha = 0.1$, $class\_weight$=None

Table 9.2: Best choice configurations for Rocket for each of the four datasets.

$nb\_filters = 32$, $kernel\_size = \{10, 20, 40\}$, $use\_bottleneck = True$, $use\_residual = False$.

The predictive performance of Rocket, InceptionTime, and lstm are reported in Table 9.4. Overall, all the models perform well, achieving good precision and recall for both classes, even in unbalanced settings. For the most unbalanced case, which is the $h = 2$, Rocket and InceptionTime perform better than lstm, showing better generalization capabilities. However, Rocket achieves the highest recall on class high, which is the most important class for our setting, being the class of the users with high risk of privacy. InceptionTime, instead, while having generally good metrics, does not perform well on the recall for high class. The real benefit of Rocket over other models is in training time. Table 9.5 presents a comparison of the training time between InceptionTime and Rocket. While training the lstm model can take many hours, the other two models are faster. Rocket is the quickest, with a training time of just a few minutes, allowing us to achieve the *online* interaction with the end user we are aiming at.

9.4.3 Mobility Privacy Risk Explanation

To provide an explanation in this context we refer to Shap. When using Shap, we can obtain a local explanation of a prediction based on the importance of each feature. While Shap is primarily used for extracting local explanations, summing up the local explanations can provide a global explanation as well.

In this study, we first analyze the use of Shap tailored for lstm models. By applying Shap, we aim to demonstrate the limitations of using this approach for these types of

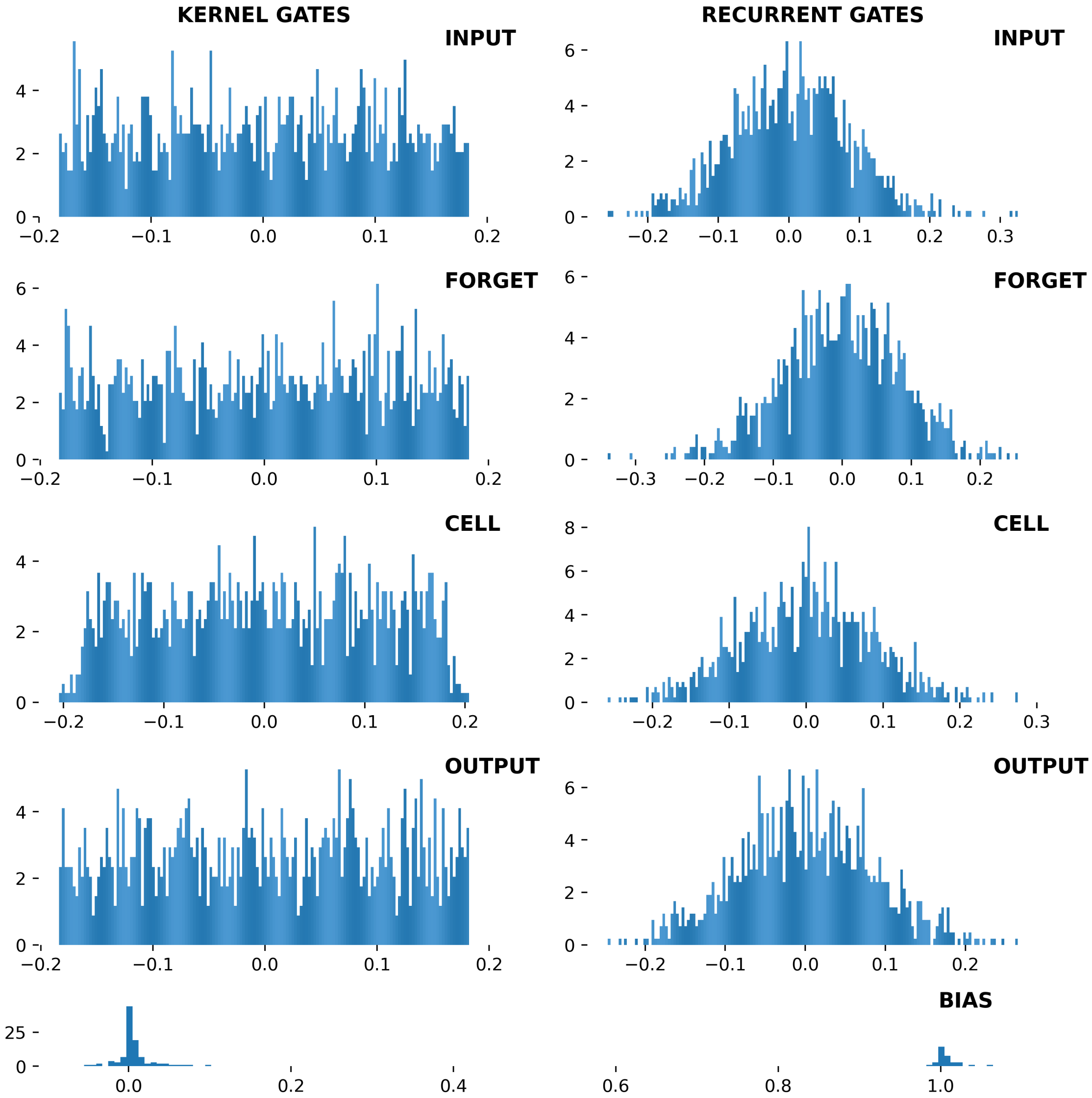


Figure 9.6: The histograms represent the weight distributions on the LSTM per kernel, and within each kernel, per gate.

models. Specifically, we will focus on the limitations in terms of computational time and the actual comprehensibility of the explanations obtained. To address these limitations, we propose a visualization tool for the explanations obtained applying Shap.

B_h	Parameter	Istat	Voronoi
h=2	Batch	64	64
	Epoch	20	100
	Optimizer	Adadelta	Adamax
h=3	Batch	64	32
	Epoch	20	100
	Optimizer	Adamax	Adadelta
h=4	Batch	64	64
	Epoch	40	80
	Optimizer	Adadelta	SGD
h=5	Batch	64	32
	Epoch	40	20
	Optimizer	Adadelta	Adamax

Table 9.3: The results obtained from the grid search of each model.

	$h = 2$			$h = 3$		
	Rocket	Inception	Lstm	Rocket	Inception	Lstm
Acc	0.81	**0.84**	0.80	**0.88**	0.87	**0.88**
P_{low}	**0.91**	0.88	0.90	0.89	0.86	**0.90**
P_{high}	0.63	**0.72**	0.62	**0.88**	**0.88**	**0.88**
R_{low}	0.81	**0.89**	0.81	0.84	**0.85**	0.84
R_{high}	**0.80**	0.70	0.76	0.91	0.89	**0.92**
F1	0.78	**0.80**	0.76	**0.88**	0.87	**0.88**

	$h = 3$			$h = 4$		
	Rocket	Inception	Lstm	Rocket	Inception	Lstm
Acc	**0.90**	0.89	0.89	0.91	0.90	**0.92**
P_{low}	**0.90**	0.87	**0.90**	0.87	0.86	**0.89**
P_{high}	0.90	**0.91**	0.89	0.93	0.93	**0.94**
R_{low}	0.86	**0.89**	0.84	0.88	0.88	**0.89**
R_{high}	**0.93**	0.90	0.92	0.92	0.92	**0.93**
F1	**0.90**	0.89	0.89	0.90	0.90	**0.91**

Table 9.4: Metrics of Rocket, InceptionTime and lstm compared for each dataset h. For precision P and recall R we present the values for both classes (*high* and *low* risk. From a privacy perspective R_{high} is the most important value as it represents the fraction of correctly predicted high risk individuals.

SHAP to explain LSTMs In the following, we report the results obtained by applying Shap to the lstm model trained with background knowledge $h = 2$. In particular, we trained the explainer on the same training set employed for the training of the lstm model. Then, we tested it on the test set on which we also tested our lstm model. We remark that the same locations can be found multiple times in a trajectory and hence in Shap since it considers each point of the trajectory as a feature. We present our results by anonymizing the names for the different locations. This is due to privacy issues: we trained our models

	InceptionTime		Rocket		lstm	
Dataset	Training	Test Time	Training	Test	Training	Test
$h = 2$	16h49min	6sec	2min32sec	44sec	8h50min	60sec
$h = 3$	20h7min	6sec	3min	40sec	5h30min	60sec
$h = 4$	4h	4sec	7min	16sec	5h50min	60sec
$h = 5$	9h24min	5sec	8min	17sec	6h15min	60sec

Table 9.5: Training times on the train set and prediction times on the test set for ROCKET and InceptionTime. Overall Rocket is the fastest model in training.

on real human mobility data, and presenting the real explanation provided by Shap could reveal some sensitive information such as home and work address.

A first result we obtained from the application of Shap confirmed our expectations from the lstm: Shap does not consider as important the padding added to normalize the trajectories. In Figure 9.8, we report the result obtained by the application of Shap on a record that the lstm classified as high risk. In Figure 9.9, we also report the explanation of a record that was classified as low risk. For each record, we plot the *expected value* and the *shap values* of the actual class predicted by the black-box model.

To analyze the results obtained, we look at the locations suggested by Shap. For each local explanation, we look at the top 3 most important locations for the prediction, as indicated by Shap. Then, we considered the top 3 most frequent locations of the same trajectory. Thus, we investigate how many users have one of their 3 most frequently visited locations as their top 3 most important locations for prediction. The results obtained are shown in Table 9.6. It is interesting to note that for the low risk class the top 3 locations that Shap considers most important are the most frequent ones in the trajectory of the user under analysis (the first and the second most frequent locations cover more of the 90% of the records). For the high risk class, the distribution is smoother, but the majority of the locations under analysis are among the first and the second most frequent locations. Theoretically, if the attacker knows information such as the most frequent locations (home address and the workplace), she has an advantage in the process of identity discovery. However, we discover that both for the high risk and the low risk class, the majority of the locations that Shap considered important, are among the top-2 most frequent locations of the user under analysis. Analyzing this result further, we found another interesting evidence: the *relative frequency*, presented in Table 9.6. At the beginning of the section we mentioned that Shap considers each location in the trajectory as a variable and hence it assigns to each of it an importance value. For each user, we first sorted, in decreasing order, the locations by their Shap values and selected the top-3 locations with the highest Shap values. These locations are the most important for the classification of that individual as indicated by Shap. For each of these 3 locations, we computed the frequency of visits of the user under analysis. Regarding the user, we are also able to compute the total number of visits she made during the period of observation. We then calculated the ratio between the frequency of visits of the top 3 locations and the total number of visits for that individual. Finally, we averaged it over the total number of users, obtaining an averaged normalized relative frequency. For the LOW risk class, the frequencies are quite high, while for the HIGH risk class they are lower. This result suggests that for the people that are in the LOW risk class, the most important location for the prediction of their privacy risk is a location that they tend to visit often. In contrast, for the HIGH risk class, we can see that the most important location is, on average, one visited less often. An example of this phenomenon is reported in Figure 9.9, that is an explanation of a record labeled as LOW risk: here Feature 336, indicated by SHAP as the most important location for the prediction of the subject, is the second most frequent location (work address) in the trajectory of the subject. Moreover, Feature 338, indicated by SHAP as the second most important location for the prediction of the subject, is the most frequent location (home address) in the trajectory of the subject. A similar situation can bee seen for the record in Figure 9.8. In this case the top seven Features for the prediction are all the most frequent locations for that user. These preliminary results suggest a connection between privacy risk (and consequently the explanation given by SHAP), and the individual movement behavior of users. Moreover, the explanation of SHAP indicates that the frequency of visit is a much more determining factor for the lower class of risk, suggesting, as expected, that visiting more frequently the same location may hide one user's movement in the crowd.

In the case of SHAP, local explanations can be summed up to obtain a global explanation as shown in Figure 9.7. This plot represents the explanation for all records predicted as HIGH risk. In this case it is very difficult for the analyst to understand which are the most relevant locations that contribute to the high (or low) risk.

Both in the case of local explanations and of global ones, the plots generated by SHAP are often complex and require additional analysis to be useful. In our case, we found that computing certain average values, such as the *relative frequency*, was necessary to gain meaningful insights from the plot. In addition, the features used in our study, such as latitude and longitude points, require semantic context to be properly interpreted. Therefore, relying solely on the SHAP plot without considering the associated feature semantics may lead to incorrect or incomplete conclusions. Summing up, the linear layouts proposed by SHAP has two main limitations: first, the high number of features does not allow a clear reading of those locations with smaller contributions; second, the topological and spatial relations among locations are not evident.

In addition, the computation of the SHAP values for long trajectories require a huge amount of time and space, making the process unfeasible for an on-line setting, such as the one of EXPERT.

The visual interface introduced in Section 9.3 addresses these two limitations. Figure 9.10 shows a screenshot of the interface showing the SHAP values associated with the

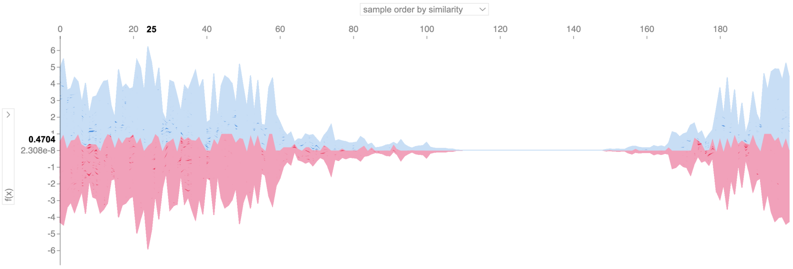


Figure 9.7: Shap Force Plot visualization of the contributions of various locations towards HIGH risk. The standard visualization does not provide any significant information to domain experts.

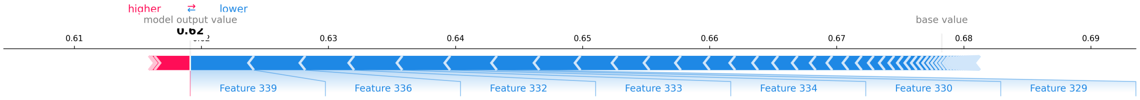


Figure 9.8: The local explanation obtained employing SHAP. This record was classified as HIGH risk with high probability.

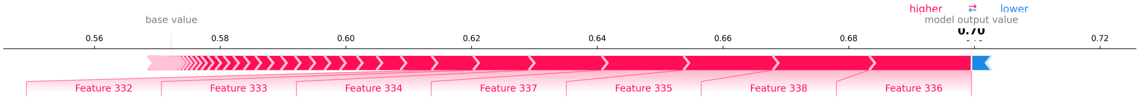


Figure 9.9: The local explanation obtained employing SHAP. This record was classified as LOW risk with high probability.

prediction of HIGH risk for each location.

This visualization allows an analyst to immediately understand which areas of the map present the highest contribution for the model towards risk classification. Our map allows for a much more intuitive understanding of the contributions of each location with respect to the classical SHAP visualization.

Moreover, our visualization can help the analyst understand the dependence of privacy risk on the mobility behaviors of the collectivity. For example, the cluster of locations in Figure 9.10 along a country road shows a high contribution to the HIGH risk, confirming the

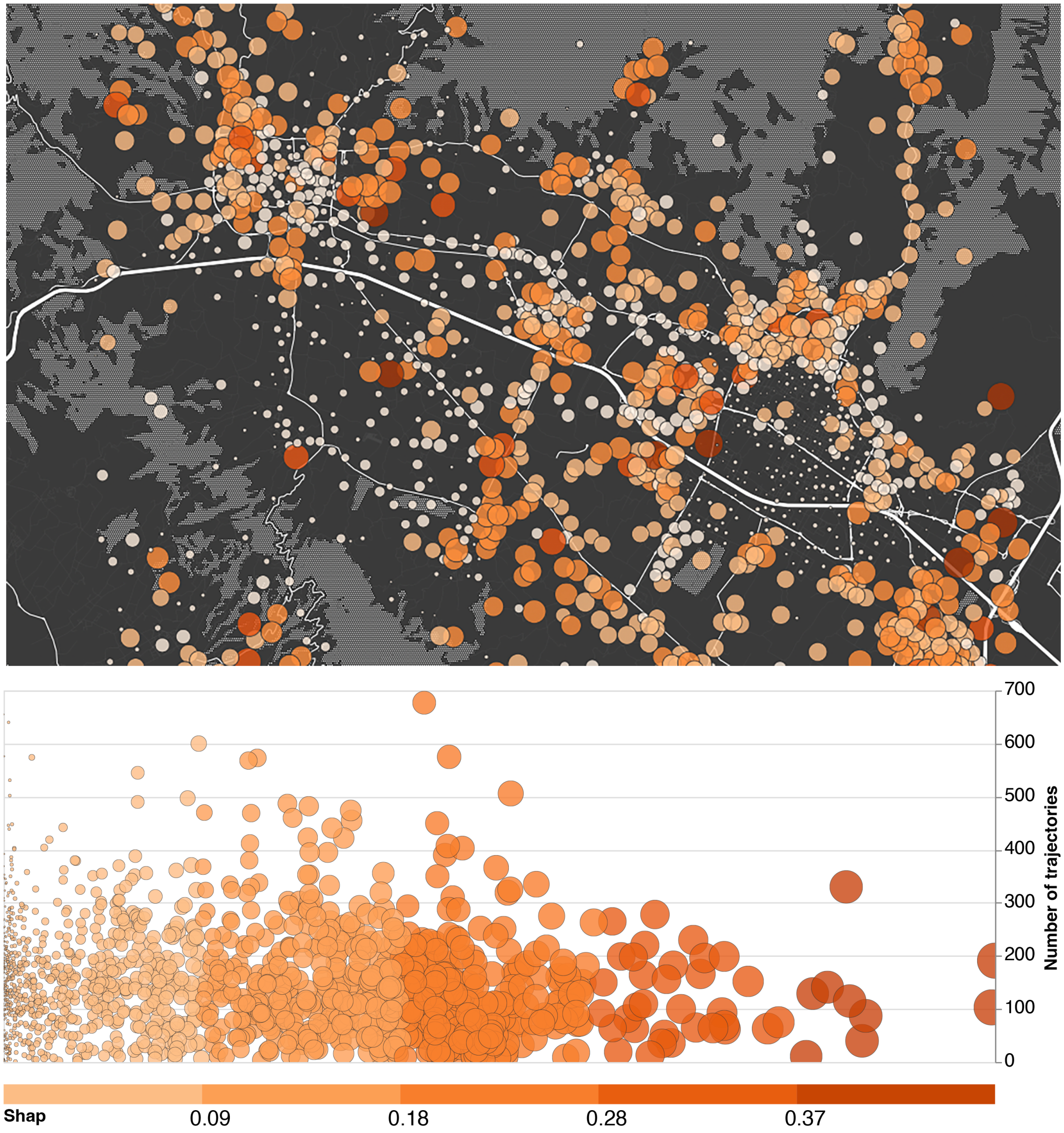


Figure 9.10: Visual interface for the exploration of explanation and prediction of privacy risk. Each circle represents the contribution to the prediction of HIGH risk, with area and color proportional to the value.

Shap Rank	Class High			Class Low		
	Relative Freq.	**Loc.**	**Users**	**Relative Freq.**	**Loc.**	**Users**
top 1	0.350 (0.261)	f_1	0.493	0.605 (0.273)	f_1	0.649
		f_2	0.200		f_2	0.273
		f_3	0.082		f_3	0.013
top 2	0.344 (0.26)	f_1	0.477	0.609 (0.27)	f_1	0.640
		f_2	0.196		f_2	0.264
		f_3	0.089		f_3	0.0176
top 3	0.349 (0.26)	f_1	0.501	0.554 (0.28)	f_1	0.635
		f_2	0.183		f_2	0.252
		f_3	0.088		f_3	0.018

Table 9.6: Exploration of the locations highlighted by SHAP. We considered separately users classified as HIGH risk from the ones that are LOW risk. The column *Shap Rank* refers to the locations ranked by importance values: for each user, we first sorted, in decreasing order, the locations by their SHAP values and selected the top-3 locations with the highest SHAP values. The *relative frequency* column reports the average frequency of all the top-3 locations for each user, as explained in details above. The columns *Loc* and *User* have to be considered together: for each user, given the location that SHAP has identified as top $i-th$, we report if it corresponds to one of the top-3 most frequent places of the user. In the table, we show the percentage of users who have SHAP's top $i-th$ location as the $j-th$ frequency of visits. As an example, for the LOW risk case, almost 65% of the users have the top-1 location of SHAP that is also her most frequently visited location.

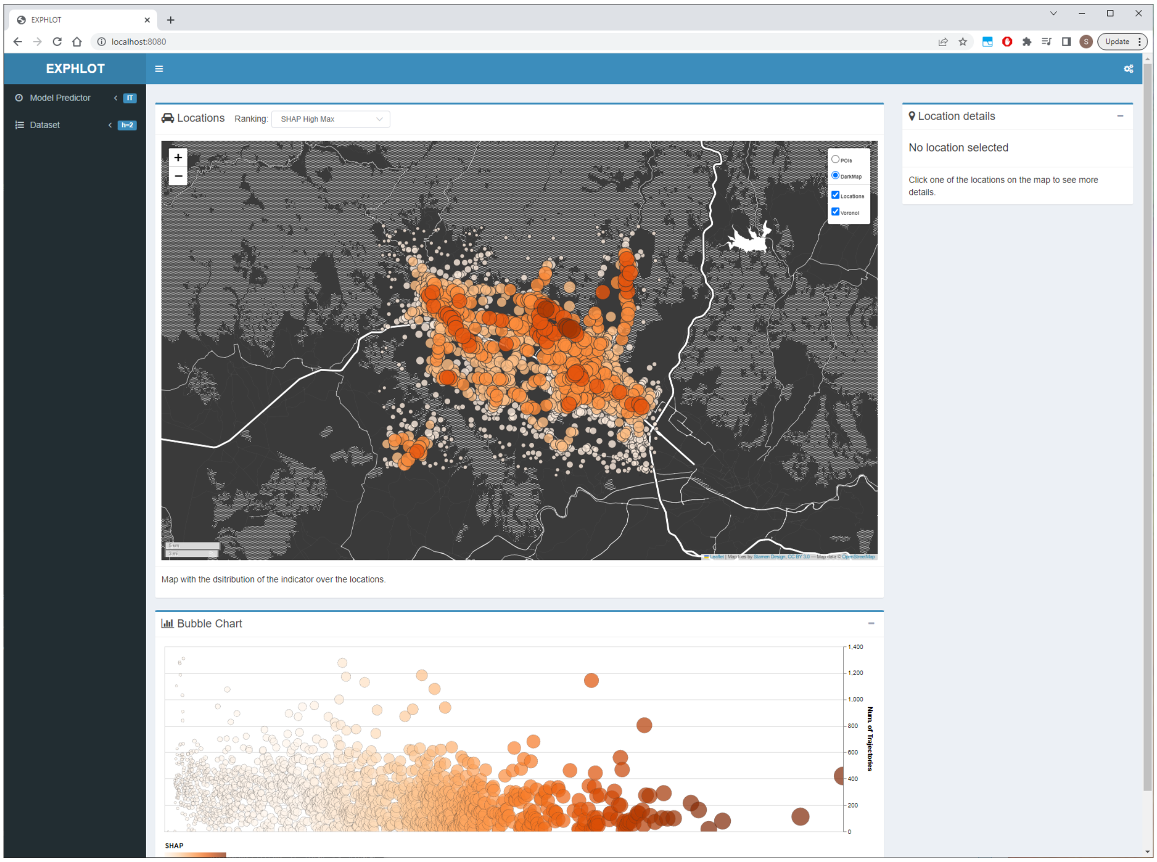
intuition that low-traffic roads are more prone to privacy exposures. Moreover, the urban surroundings present a lower level of risk, even if it is possible to visually detect different privacy levels in two close municipalities: the south-east town has very low-risk levels; the north-west town has a higher risk level.

9.5 Exploring results with the Visual Dashboard

In this Section we show how the visual interface can help the analyst and/or the end-user to investigate the properties of the data after the risk prediction and explanation have been computed. We begin the presentation of this visual tool in Section 9.5.1, in which the general framework is presented with the different widgets and components. Then, we present two ways in which the visual dashboard can be exploited:

- *Aggregate visualization.* This visualization is tailored to analyst, which aims at understanding the general mobility behaviour of the overall dataset, with the overall most important locations for the privacy risk prediction highlighted. This visualization is presented in Section 9.5.2;

- *Single user visualization.* This visualization is tailored to the end-user, having the objective to visualize her mobility, with the most important locations for the privacy risk highlighted. This visualization is reported in Section 9.5.3.



Ranking: SHAP High Max

No location selected

Click one of the locations on the map to see more details.

Map with the dsitribution of the indicator over the locations.

Figure 9.11: Overview of the web application

9.5.1 Visual Widgets and HOW-TO user guide

The visual dashboard is composed of multiple widgets:

- on the left the sidebar contains details about the dataset that is loaded in the interface. The user can choose to explore different datasets, obtained accordingly with the exploration described in Section 9. The dimensions to be explored are (1) the model predictor to use for risk prediction (this visualization is limited to Rocket and InceptionTime which are the ones performing best in this setting, as described in the experiments (Section 9.4) and (2) one of the four datasets, depending on the level of knowledge of the attacker, specifically $h = 2$, $h = 3$, $h = 4$, $h = 5$. The general appearance of the visualization tool is depicted in Figure 9.12.

- The central map is implemented by means of a tile-base background map and a

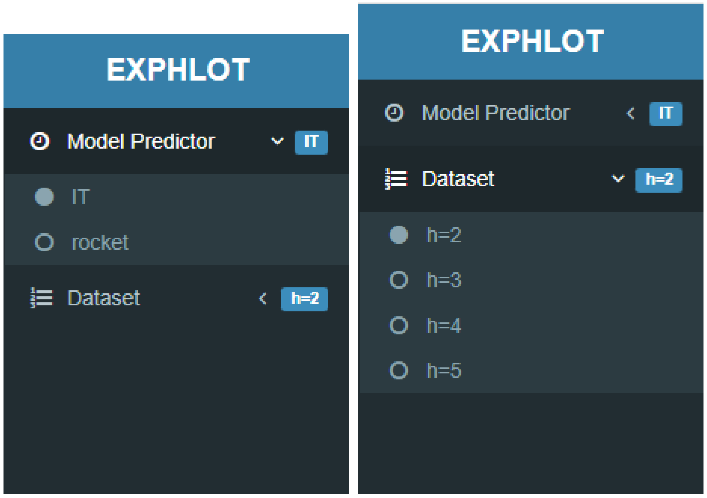
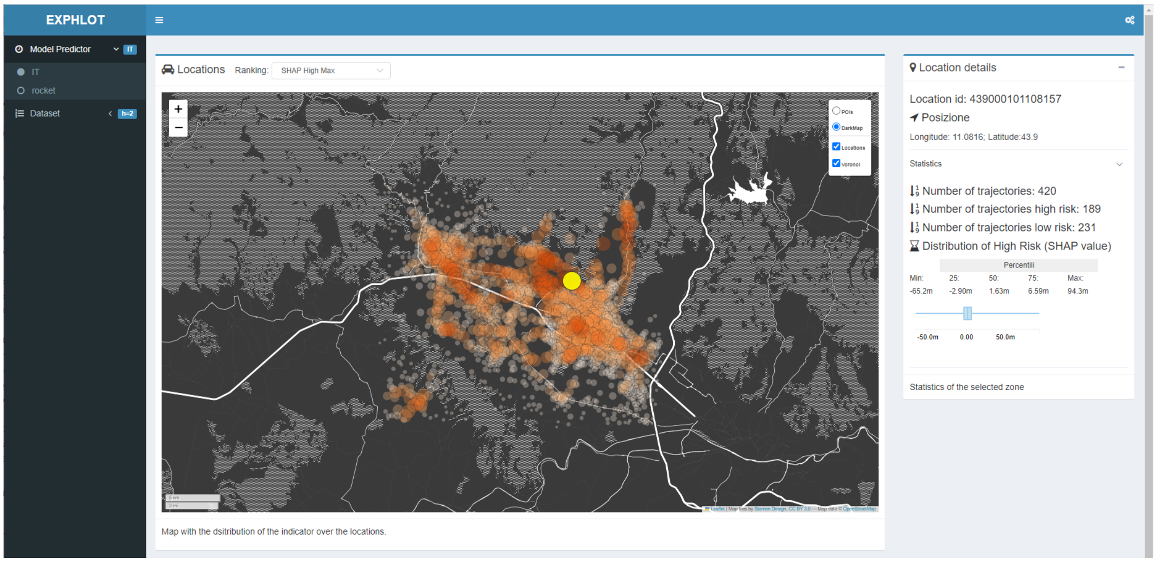


Figure 9.12: Selection of the set of locations to be displayed. (Left) Selector for the prediction model to be used. (Right) Selection of one of the four datasets

customized visualization overlayed. The visualization of the location is following the mapping described in Section 9.4.3. In the top part of this map, the user can select a series of indicators derived from the privacy risk estimation and explanation: number of total trajectories, number of trajectory with LOW risk, number of trajectories with HIGH risk, minimum SHAP value, max SHAP value. In the example of Figure 9.11, the SHAP max value is displayed. The area and fill color of each circle is proportional to this value. The location, of course, is determined as the centroid of the corresponding VORONOI cell. The map can be browsed by usual gestures (pan, zoom, change of background map). The user can also select a specific location by clicking on the corresponding circle (see Figure 9.13).

- The bubble chart on the bottom is representing the set of locations and circles of the map on a set of orthogonal axes. In this parallel view, the user can further explore the relation of the privacy risk contribution and other properties of the data. This display is also interactive and it is possible to click on circle to highlight the location on the map and show details in the right box.

- The box on the right is populated when a location is selected and it shows additional detail of the location with statistics of the privacy risk predicted in that location.



Locations Ranking: SHAP High Max

Map with the dsitribution of the indicator over the locations.

Location details

Location id: 439000101108157

Posizione

Longitude: 11.0816; Latitude:43.9

Statistics

Number of trajectories: 420

Number of trajectories high risk: 189

Number of trajectories low risk: 231

Distribution of High Risk (SHAP value)

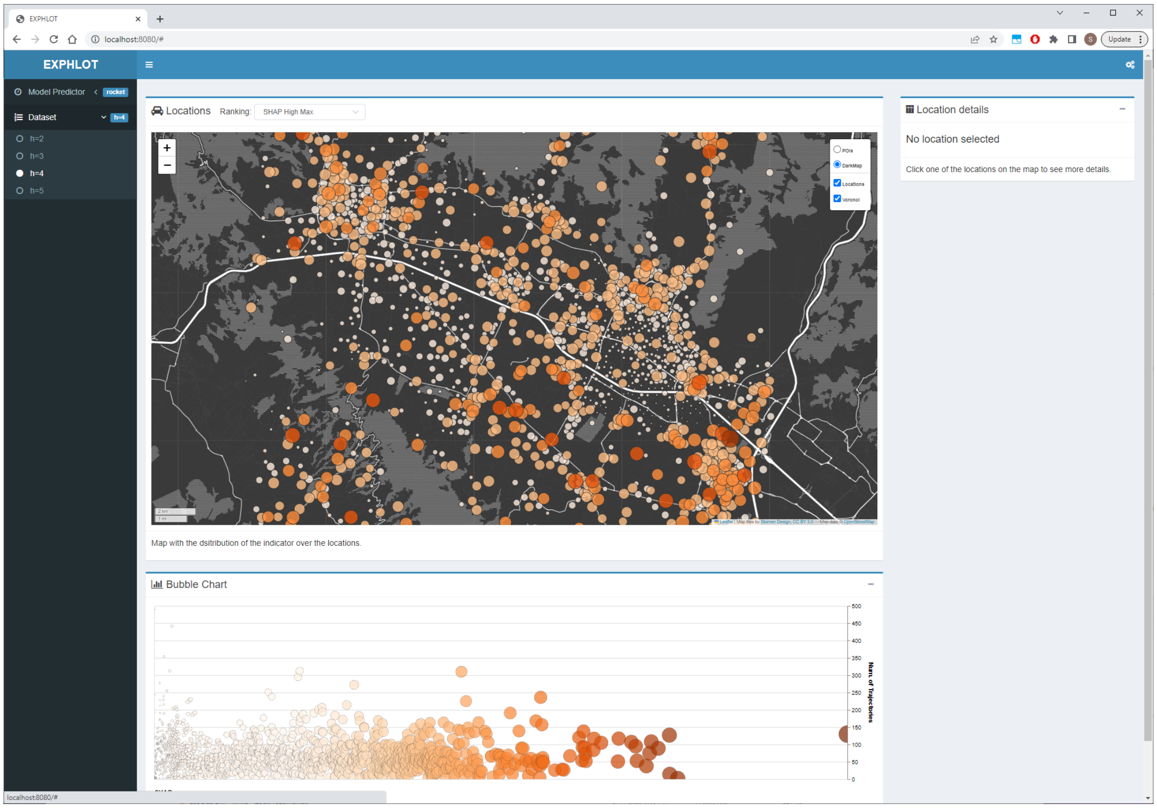
Percentili				
Min:	25:	50:	75:	Max:
-65.2m	-2.90m	1.63m	6.59m	94.3m

Statistics of the selected zone

Figure 9.13: Selection of a location by clicking on the corresponding circle. On the right, a box shows additional details of the location

9.5.2 Aggregate Visualization of mobility privacy risk

In this Section we report the *aggregate visualization*, one of the two possible usages of this visualization tool for EXPERT. This visualization is tailored for analysts, having the objective is to explore and analyze the overall mobility behaviour of the dataset under analysis. The goal then is to provide the analyst a visualization in which he/she can observe the overall importance of the locations. For an in-depth analysis, we present to the analyst not only the most important locations, based on the SHAP values, but also other statistics, such as the number of trajectories passing from the location under analysis, being them HIGH or LOW risk, and the distribution of the risk. In the following, we present the possible visualization for the analyst through pictures. We report a visual comparison of the four datasets (depending on the level of knowledge the adversary possesses) for the ROCKET predictor. In Figure 9.14, we can observe the overall behaviour of the users in the dataset with adversary knowledge $h = 2$. This is the most difficult setting for the attacker, which can exploit only a knowledge of 2 locations per trajectory. For this reason, it is also the setting in which the privacy exposure is lower, as described in Section 9.4. However, from this first visualization we can already see the areas most important for predicting the HIGH risk. Moving on, in Figure 9.15, we have the same trajectories as before, but this time with an attacker knowledge of 3 locations. Visually, we can observe that this dataset already has more important locations for the HIGH risk prediction. This pattern is even



Ranking: SHAP High Max

No location selected

Click one of the locations on the map to see more details.

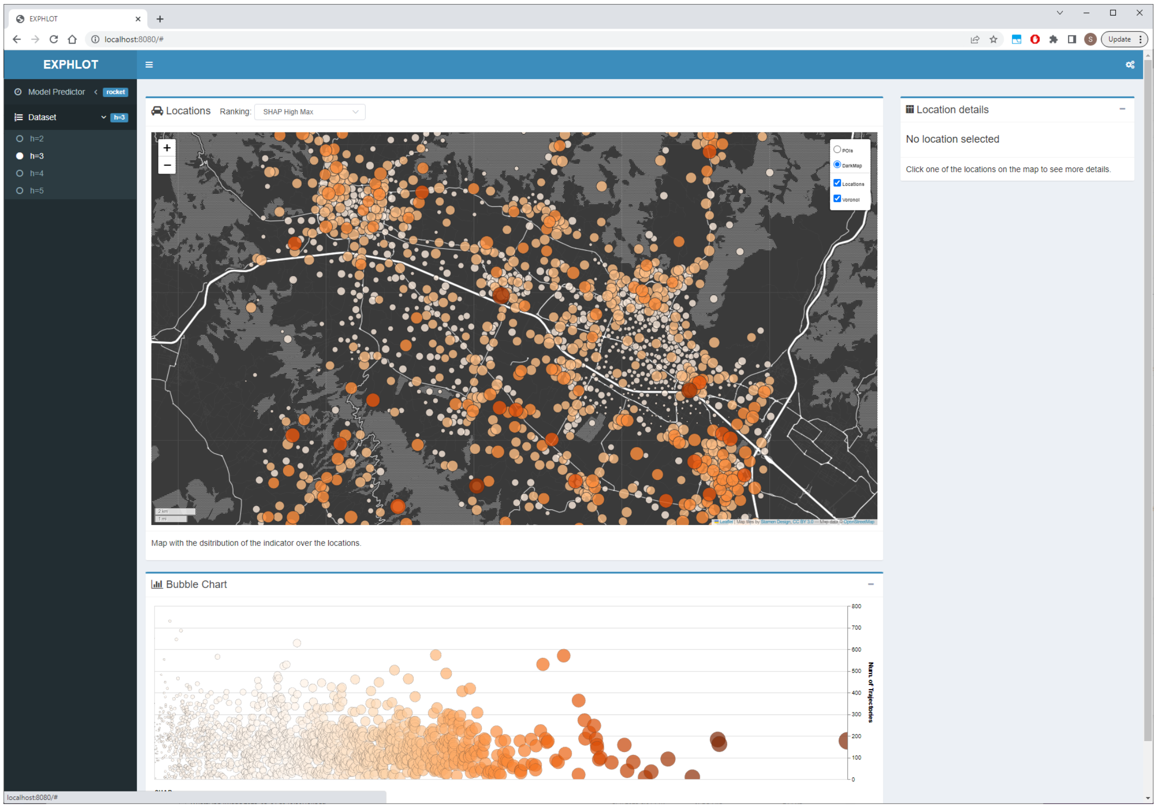
Map with the dsitribution of the indicator over the locations.

Figure 9.14: Visualization for ROCKET and $h = 2$

more observable when looking at Figure 9.16 and 9.17, in which the attacker knowledge increases. By clicking on the specific location, the analyst obtains statistics about it.

9.5.3 End-user Visualization of mobility privacy risk

In this Section we present how the visual tool can be exploited from the point of view of a single end-user. In this case, in fact, the user is primarily interested in observing her trajectories, with the most important locations for the risk prediction highlighted. An example of this kind of visualization is provided in Figure 9.18, in which we can observe the mobility behaviour of a single user, classified as HIGH risk by the ROCKET predictor, for the attacker's background knowledge $h = 3$. In this case, the locations visited by the user are reported in a scale of purple, with a darker color for the most important locations for the prediction at hand. From this first visualization, the end-user will have a general overview of her mobility. However, for an in-depth analysis we also provide Figure 9.19, in which the user can observe her trajectories (in purple scale), as well as the aggregate



Ranking: SHAP High Max

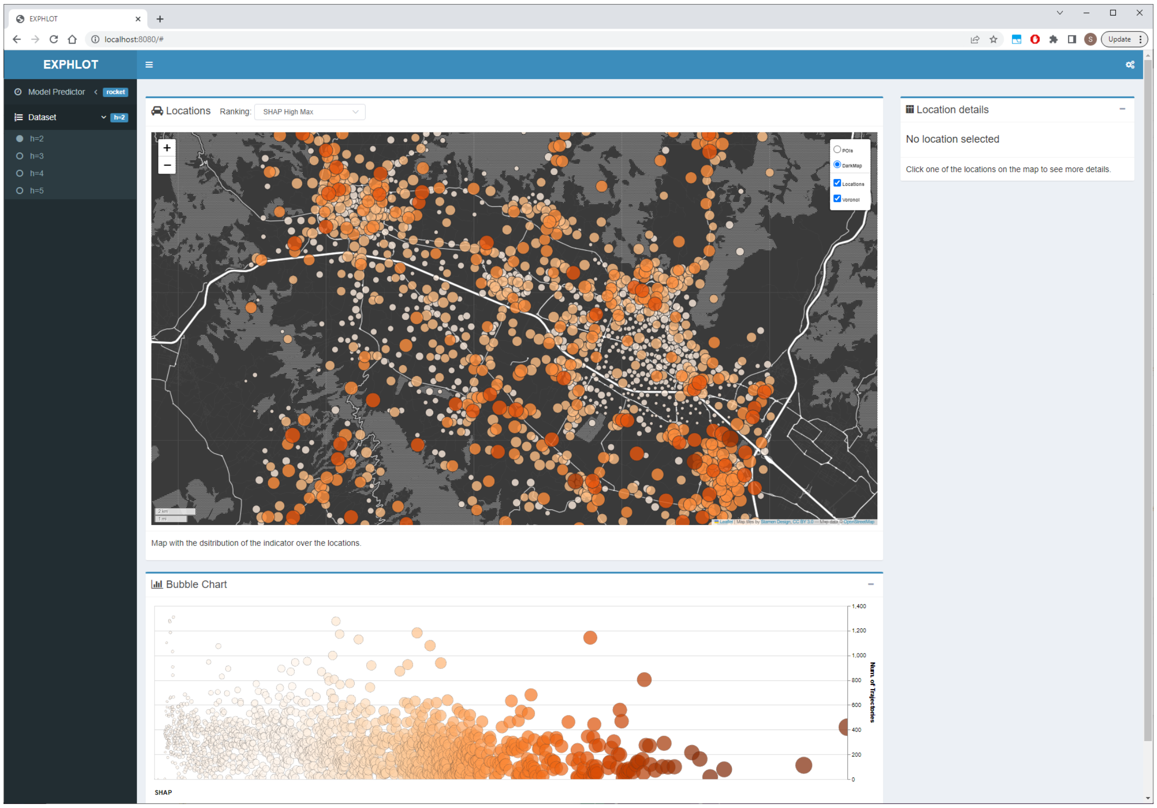
Map with the dsitribution of the indicator over the locations.

Location details

No location selected

Click one of the locations on the map to see more details.

Figure 9.15: Visualization for ROCKET and $h = 3$

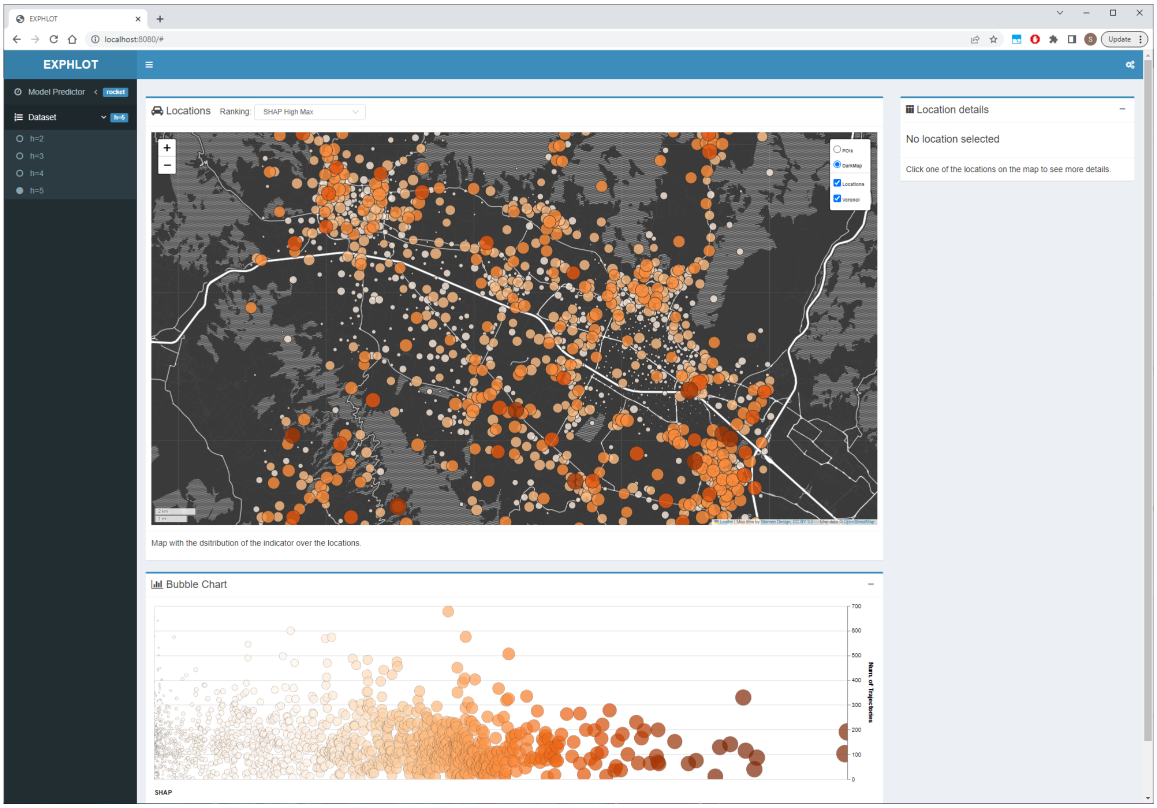


No location selected

Click one of the locations on the map to see more details.

Map with the dsitribution of the indicator over the locations.

Figure 9.16: Visualization for ROCKET and $h = 4$



Ranking: SHAP High Max

Map with the dsitribution of the indicator over the locations.

No location selected

Click one of the locations on the map to see more details.

Figure 9.17: Visualization for ROCKET and $h = 5$

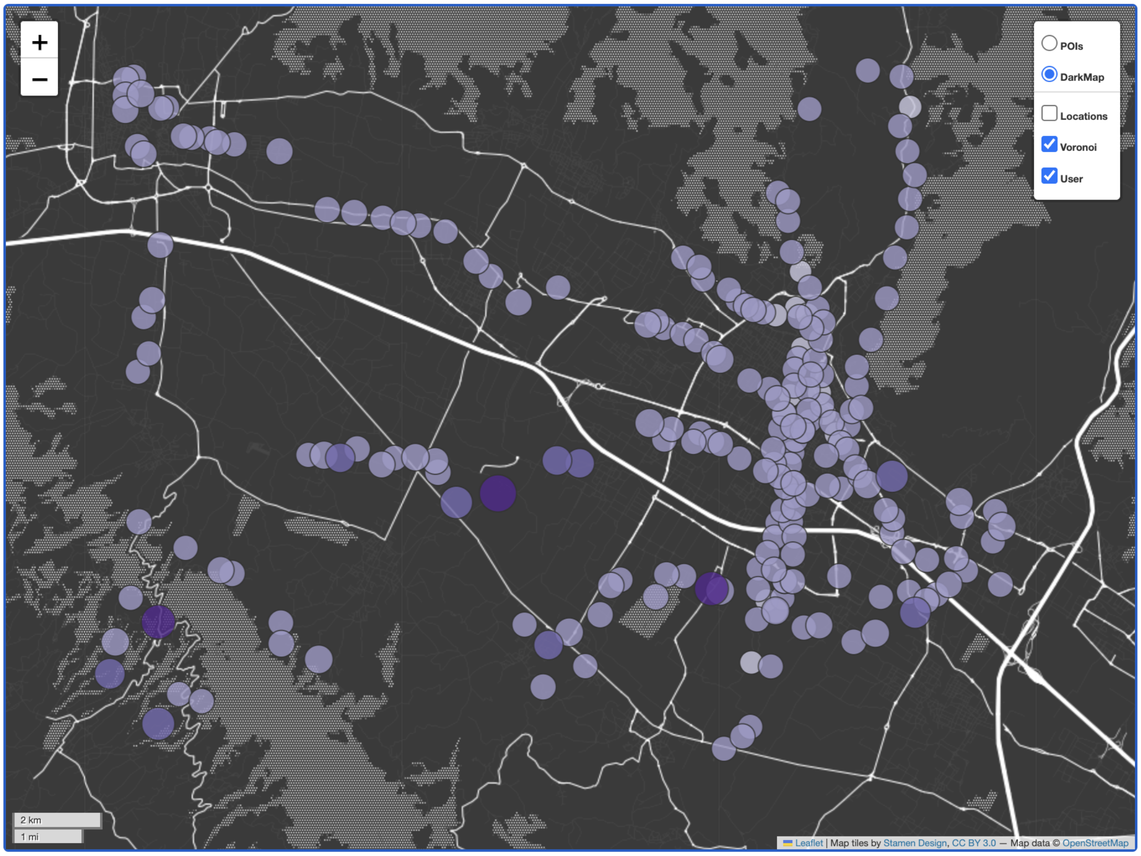

+
−
○ POIs
● DarkMap
□ Locations
☑ Voronoi
☑ User
2 km
1 mi
Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

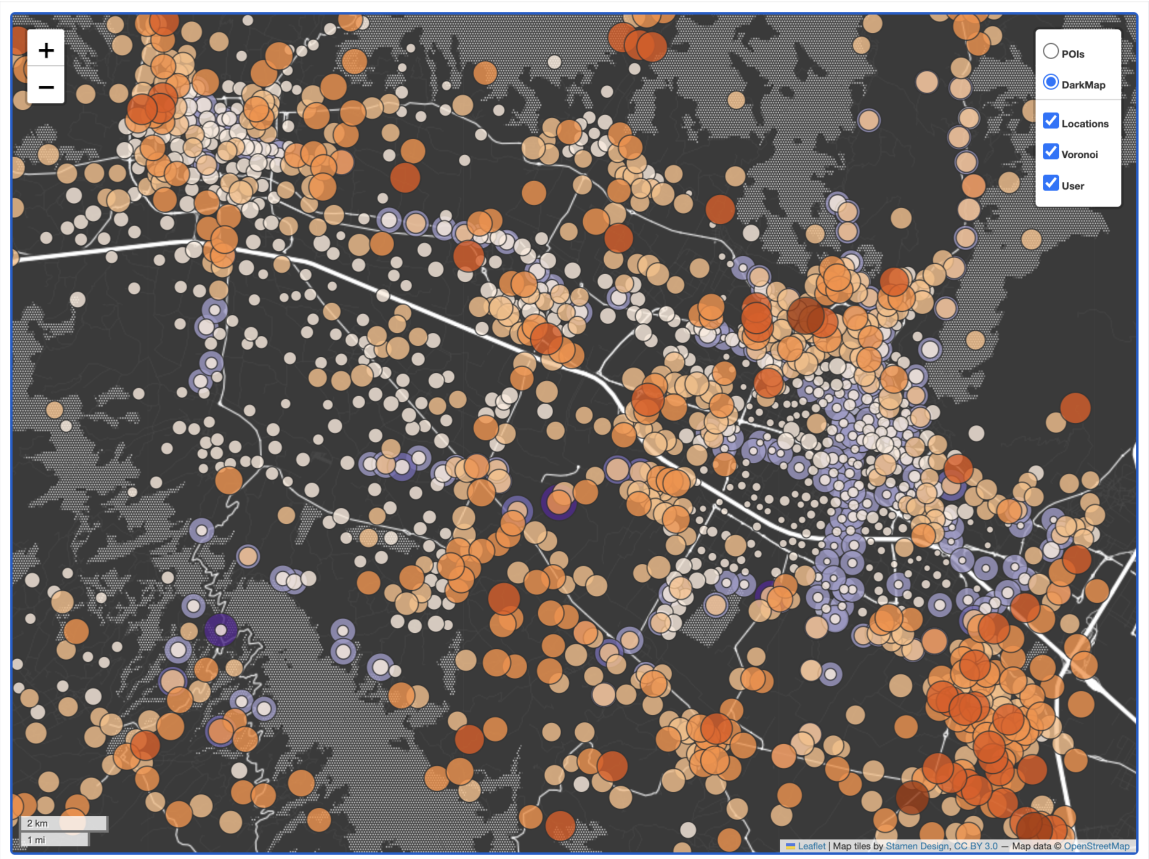
Figure 9.18: Visualization for a single record, predicted as HIGH risk, with ROCKET and $h = 3$

visualization of the overall dataset. In this way, the user can observe her behaviour with respect to the people's overall behavior. In particular, for the user under analysis, we can observe that her trajectory does not contain locations that are actually important for the prediction of HIGH risk for most people in the dataset.

9.6 Discussion

In this Chapter we introduce a variant of EXPERT, which is a privacy assessment, prediction, and explanation framework specifically designed for human mobility data represented as trajectories.

We enhance existing privacy risk assessment frameworks by employing machine learn-


○ POIs
● DarkMap
☑ Locations
☑ Voronoi
☑ User

+
−
2 km
1 mi
Leaflet | Map tiles by Stamen Design, CC BY 3.0 — Map data © OpenStreetMap

Figure 9.19: Visualization for a single record, predicted as high risk, with Rocket and $h = 3$

ing models tailored for sequential data, including InceptionTime, lstm, and Rocket. Additionally, we develop heuristic techniques to compute Shap values efficiently and create a customized visualization tool for human mobility data analysis.

Our framework demonstrates accurate prediction of privacy risk in human mobility data and effectively explains the predictive models using fast Shap value calculation. The visualization tool provides an intuitive and interactive map-based representation, showcasing the essential contributions and information about the privacy risk.

To validate our framework, we conducted experiments on real, confidential human mobility data, revealing new insights into the nature of privacy risk. Our work equips privacy analysts and experts in the field with an interactive and actionable tool for understanding the privacy risk of human mobility data in a fast and intuitive manner.

In terms of future directions, we aim to leverage the efficiency of privacy risk prediction by developing a visual analytics environment. This environment will couple prediction and visualization, enabling experts to analyze the results of different privacy mitigation techniques. It will serve as a "what-if" simulation module, allowing analysts to modify the data and assess privacy risk in an interactive process. This functionality will greatly assist in developing appropriate privacy protection measures based on techniques such as generalization or deletion.

Another interesting direction involves integrating additional data quality measures into the framework. This will enable further experimentation with different protection measures on the data prior to its release, providing more comprehensive insights into the effects of these measures on privacy.

Chapter 10

Summary of Part III

In this Part we analyzed a possible synergy between Data Privacy and Explainable Artificial Intelligence, by tackling the problem of increasing the user self-awareness in the task of privacy risk assessment. In particular, we present EXPERT, a framework that addresses the computational complexity issue of the state-of-the-art methodology for privacy risk assessment, namely PRUDEnce, and enhances users' awareness by leveraging Machine Learning models to predict individual privacy risks. Lastly, EXPERT exploits local explainers to produce explanations of predicted risks. The proposed framework is modular so that it can be tailored to specific data input and explanation requirements to achieve desired outcomes. In this Thesis, we present two main variants of EXPERT, both tailored for human mobility data, which are among the most dangerous sensitive data due to their structure. The first variant deals with human mobility data in the form of tabular data, with features extracted from the trajectories. The second one, instead, works directly with the trajectory data. For both of the cases, we present an in-depth analysis both on the *privacy risk prediction* module, analyzing the best classifiers depending on the task under analysis, as well as on the *privacy risk explanation* module. In particular, for the tabular setting, EXPERT were able to provide good prediction performance and high-fidelity explanations. However, the usage of features makes the comprehensibility of the explanation difficult, if not impossible. In fact, these explanations are tailored just for experts, which can understand the meaning of every single feature and can hence have an overall understanding of the reasons that lead the classifier to a particular prediction. Given these results, we then move to the second setting, in which EXPERT analyzes the privacy risk directly on the raw trajectories. In this case, we conducted an in-depth analysis to provide the end-users with a high-performing classifier for sequential data, also considering the time consumption of the training procedure to achieve an online interaction. For EXPERT with trajectories, in particular, we propose a visualization tool for the trajectory setting, in which the user can visualize her mobility behaviour on a map, with different visualizations for the locations that are more important for the prediction under analysis. By utilizing the Expert framework, users can gain a better understanding of privacy risks associated with their data, allowing them to take appropriate measures to protect their privacy. In particular, the last variant of this framework has two main targets: the end user, that is the subject of the data under analysis, who can see its movements through a *single* map visualization, but also the analyst, who can study the *aggregate* data visualization of Expert. Therefore, the framework can have two different uses: in the first case it can be used by the user to improve the understanding of his or her privacy risk and possibly make appropriate changes, while in the second case it can be used as an analysis tool, useful in both research and industry contexts, to analyze the data prior to the publication of actual datasets.

Part IV

Privacy exposure of explanation methods

In this Part we analyze the relationship between Explainable Artificial Intelligence and Privacy from the perspective of the privacy exposure of explanation models.

In fact, one of the possible drawback of Artificial Intelligence systems based on ML models is their potential vulnerability against different attacks, such as Model Inversion attack [210] and Membership Inference Attack (Mia) [109], presented in the first part of this Thesis, in Chapter 3.2.2. These privacy attack methods can potentially infer the data used for training the model by simply querying the model. Thus, privacy mechanisms such as differential privacy [143] are typically applied to counter the potential privacy exposure. However, the problem of privacy attacks against the Machine Learning models may also affect explanation methods. In fact, explainers are learned functions derived by exploiting the predictive knowledge of a black-box model learned on a private dataset. Thus, they could leak information about this private dataset. Despite this potential risk, only a few works address privacy issues in Explainable Artificial Intelligence [19, 20]. For this reason, in this Part we analyze this research problem in depth, focusing on membership attacks. We start this Part by presenting, in Chapter 11, the formal definition of a novel membership attack, namely Aloa, a variant of the LabelOnly attack, presented in Chapter 3.2.2. Aloa is a membership attack whose objective is to determine the membership of people in the training data of the black-box model. Therefore, by applying this attack against a black-box we obtain the privacy exposure of the model under analysis. Following, in Chapter 12 we present reveal, a privacy risk assessment framework for global and local explainers bases on surrogate models. The Chapter related to Aloa describes the work published in [211], while part of the contribution related to reveal is published in [212] and the remaining contributions are submitted to a journal paper [213].

Chapter 11

Agnostic Label-Only Membership Inference Attack

The increasing prevalence of smart technology in everyday life, such as self-learning and auto decision-making systems, is largely due to advancements in Machine Learning: applications such as Gmail's spam filtering (Dada et al., 2019 [115]), YouTube's video recommendations (Mwinyi et al., 2018 [116]), text correction software (Ghosh & Kristensson, 2017 [117]), and speech recognition (Nassif et al., 2019 [118]) all exploits Machine Learning algorithms to improve their functionality. While Machine Learning (ML) can greatly enhance the capabilities of a system, it also presents potential vulnerabilities. Attackers may exploit flaws in machine learning systems to infiltrate and manipulate them for malicious purposes, potentially compromising the system's reliability, confidentiality, and availability. Adversarial attacks, such as crafting special input data and poisoning the training dataset, are commonly employed by attackers to evade intrusion detection systems and mislead ML classifiers. In addition to these attacks on ML models, in which the objective is to make the model behave badly, the study of attacks on the privacy of machine learning models has recently attracted much attention from the scientific community. One of the most popular privacy attacks against ML models is the Membership Inference Attack (Mia) [109], which aims to discern between records that were used during the training phase of the machine learning model and not. An in-depth description of Mia is proposed in Section 3.2.2. This kind of attack is very risky for privacy and secrecy. In fact, knowing the membership of a record to a sensitive dataset used for training a model might enable the re-identification of users and inference of their sensitive data [104]. Moreover, reconstructing part of the training data of a model could conflict with trade secrets. Indeed, sometimes training data might be the result of successful corporate experience and investments which can lead to important competitive advantages. Thus, organizations owning such data do not want to disclose them to competitors. This type of attack was first published in 2017 by Shokri et al. [109], and then some variants have been proposed. Another interesting attack is Label-Only Membership Inference Attack [125], a variant of Mia in which the adversary determines the membership of a record to the training data of a machine learning model only using its hard predictions and knowledge about statistics of the training data. It exploits this knowledge for computing a robustness score of the model representing a proxy for the model prediction confidence. In this Chapter, we present Aloa (Agnostic Label-Only Membership Inference Attack) an enhanced variant of the LabelOnly attack, with improved performance and good stability in the prediction metrics. As LabelOnly, Aloa only exploits the hard labels of the predictions without the need to access the confidence vectors, a requirement that is mandatory in the original version of Mia. However, differently from LabelOnly we design a perturbation mechanism, enabling the computation of a robustness score of the machine learning model, which is data agnostic to the training data distributions. In other words, our attack model, used during the learning of the attack model, does not exploit the knowledge of the distribution of the features in the training data. The robustness score is the key factor in determining the record membership. We evaluate Aloa using three datasets having different characteristics. The experimental results highlight that our attack allows for better stability with respect to the standard Mia and an enhanced performance up to 3 percentage points in terms of accuracy in predicting the records membership. Even if this enhancement may seem small, for the privacy setting, this is extremely risky since it means that the adversary may have a higher probability of re-identifying people in the dataset. In addition, we relax the assumption that the attacker needs a dataset following the same distribution as the original training dataset, making the attack easier to perform with respect to the competitors. In the following, we present the methodology of Aloa in Section 11.1 and consequently, we present the experiments conducted in Section 11.2.

11.1 Aloa methodology

In this Section we present Aloa (*A*gnostic *L*abel-*O*nly membership inference *A*ttack), which is a variant of LabelOnly attack, presented in Section 3.2.2. The LabelOnly attack is based on the assumption of knowing the statistical distributions and the domain of the features in training data of the black-box. This knowledge is exploited for applying a perturbation to each feature tailored to its type and its statistical distribution. Contrary to the LabelOnly, we propose a variant of this attack completely *agnostic* with respect to the training data and the type of classifier to be attacked.

Threat Model. A membership inference attack aims to determine whether or not a given data record belongs to the training dataset of a specific classification model. To conduct an attack, the adversary can exploit specific prior knowledge that can be accessed. For this attack, we assume an adversary has black-box access to the classifier b. In other

```
Algorithm 4: ALOA ($b$, $D_s$, $p_{min}$, $p_{max}$, $k$, $n$)
Input  : $b$ - classifier,
         $D_s$ - dataset for training the shadow models,
         $p_{min}$, $p_{max}$ - perturbation percentage range,
         $k$- number of neighbours to be generated
Output: $threshold_{split}$ - split threshold found for the dataset $D_s$
$\{D_{s^1}, \ldots, D_{s^k}\} \leftarrow RandomSample(D_s, k)$
$S \leftarrow \emptyset$; $D \leftarrow \emptyset$; $Scores \leftarrow \emptyset$
for $i \in \{1, \ldots, k\}$ do
    $D^{\text{train}}_{s^i}, D^{\text{test}}_{s^i} = split\_train\_test(D_{s^i})$
    $S \leftarrow S \cup train\_shadow(D^{\text{train}}_{s^i})$
    $D^{\text{IN}}_{s^i} \leftarrow$ Assign the IN label to each record in $D^{\text{train}}_{s^i}$
    $D^{\text{OUT}}_{s^i} \leftarrow$ Assign the OUT label to each record in $D^{\text{test}}_{s^i}$
    $D \leftarrow D \cup D^{\text{IN}}_{s^i} \cup D^{\text{OUT}}_{s^i}$
for $x^i \in D$ do
    $N^i_x \leftarrow Noisy\_Neighborhood\_Generation(x^i, p_{min}, p_{max}, n)$
    $rScore_{x^i} \leftarrow Robustness\_Score(N^i_x, S, b(x^i))$
    $Scores \leftarrow Scores \cup rScore_{x^i}$
$threshold_{split} \leftarrow Iterative\_Thresholding(D, Scores)$
return $threshold_{split}$
```

words, the adversary can only query the model to obtain a prediction and, as in [125], the model only returns hard labels to queries. The adversary does not know the model architecture, e.g., the type of classifier, its hyper-parameters used for the training, and the algorithm used for the training. Lastly, the adversary has knowledge about the total number of classes, the class labels, and the input format. To perform ALOA we do not need to know distributions of the original training dataset, nor during the training of the shadow model, nor in the perturbation mechanism, in contrast to LABELONLY.

Learning ALOA. Given a black-box b, trained on a dataset D^{train}_b, ALOA attack targets it by exploiting only the hard labels, i.e. $b(x) = \widehat{y}$, and deriving a robustness score by an agnostic data perturbation. This score enables ALOA to determine if a record x belongs to the training data D^{train}_b of the black-box model under attack. The pseudo-code of the algorithm is reported in Algorithm 4. The process to create ALOA model requires as input a dataset D_s: $(x^i, y^i)_s$ in which x^i_s has the same format of training data of b and y^i_s is the predicted class obtained querying the black-box model b. Given the agnostic nature of ALOA, it does not rely on any assumptions about D_s, which may include a completely random dataset.

After querying the black-box model for labeling each x^i_s, ALOA splits the dataset D_s into training and testing datasets, obtaining D_s^{train} and D_s^{test} respectively, and then it trains one or more shadow models, $s^i(\cdot)$ on sub-samples of D_s^{train} (lines 6-7). The goal is to mimic the behaviour of b, by also having the knowledge of which records are part of the training set and which are not. In particular, as reported in Algorithm 4, ALOA constructs a dataset D, where assigns the label IN to each record in the training data of the shadow models and the label OUT to those belonging to their test data (lines 8-10).

At this point, ALOA performs its core process: the *agnostic perturbation* of the data used for training and testing a given shadow model (line 12, Alg. 4). We call this procedure *Noisy Neighborhood Generation* and we report its pseudo-code in Algorithm 5. For each data record x_s^i of the shadow dataset D_s, it generates a neighborhood of n records obtained perturbing the values of its attributes. Since the goal is to perturb each data record in their local vicinity without using any knowledge of the dataset's domain or attributes distribution, making the algorithm completely domain agnostic, our perturbation mechanism adds noise values to each attribute of the record under analysis. Given an instance x_s^i composed by m attribute-value pairs (a_j, v_j), to generate the noise value for perturbing v_j ALOA adds or subtracts to v_j a noise values $\nu = p \times v_j$ (lines 10-14, Alg. 5). The value p is a percentage randomly generated from a uniform distribution in the range $[p_{min}, p_{max}]$ (line 5, Alg. 5). The noise value ν is added or subtracted with a probability equal to 50% (i.e., following a Bernoulli process).

After this perturbation, ALOA computes for each record in the shadow dataset the *robustness score* to estimate the confidence of the shadow model s in predicting the record label (line 13, Alg. 4). This score is formally defined as follows:

$$rScore_{x_s^i}(N_{x_s^i}) = \begin{cases} 0 & \text{if} \;\; s(x_s^i) \neq b(x_s^i) \\ \frac{\sum_{x' \in N_{x_s^i}} F(s(x'), s(x_s^i))}{|N_{x_s^i}|} & \text{otherwise} \end{cases} \tag{11.1}$$

Where $F(s(x'), s(x^i))$ is a function returning 0 in case the shadow model predicts a label for the neighbor x' which is not coherent with the label predicted for x^i. In other words, in case the shadow model is faithful to black-box model on x^i, the robustness score on this record is computed as the fraction of perturbed records having coherent labels with x^i. This score has values in the range $[0, 1]$: values close to 1 mean that the classifier is robust to perturbations, thus the model is confident in predicting the record; while values close to zero register low confidence of the classifier in the prediction, indeed, in this case, several neighbors have the opposite class label to the record under analysis, meaning that the model is unsure of the prediction since it is very close to the boundary.

Once each record of the shadow dataset has its robustness score, we get a dataset where for each record x_s^i we have its score $rScore_{x_s^i}$ and the label IN, in case x_s^i belongs to the training dataset of the shadow model, or OUT if it belongs to the test dataset. Now, using the iterative thresholding procedure, ALOA finds the threshold value on the score that

Algorithm 5: Noisy_Neighborhood_Generation(x, p_{min}, p_{max}, n)

```
Input  : $x$ - a record composed by $m$ attribute-value pairs $(a_j, v_j)$,
         $p_{min}$, $p_{max}$ - perturbation percentage range,
         $n$ - number of neighbours to be generated
Output: $N_x$ - Set of new generated records

$N_x \leftarrow \emptyset$
for $t \in \{1, \ldots, n\}$ do
    $x' \leftarrow x$
    for $j \in \{1, \ldots, |x'|\}$ do
        $p \leftarrow randomNumber(p_{min}, p_{max})$
        if $v_j$ == $0$ then
            $v_j == randomNumber()$

        else
            $\nu \leftarrow v_j \times p$
            if $randomBoolean() == True$ then
                $v_j \leftarrow v_j + \nu$
            else
                $v_j \leftarrow v_j - \nu$
    $N_x \leftarrow N_x \cup \{x'\}$
return $N_x$
```

optimizes the accuracy in separating records with class label In and Out (line 15, Alg. 4).

ALOA application. Once Aloa has been trained, an adversary can use it to determine whether a given record belongs to the training dataset of the black-box model b or not. Given a record x, having the same shape as the records D_b^{train} on which the black-box was trained, our attack performs the following steps:

1. Aloa applies the Noisy Neighborhood Generation procedure, presented in Algorithm 5, to the record x. The result is a set of synthetic neighbors N_x which are perturbed through our agnostic procedure;

2. Exploiting the neighborhood N_x, Aloa computes the Robustness Score *rScore* of the record x applying Eq. (1);

3. The best threshold value $threshold_{split}$, found during the training of Aloa, is used to discern whether the record x is part of the training set or not: if $rScore \geq threshold_{best}$ then it will be predicted as part of the training set, otherwise not.

11.2 Aloa experiments

In this section we report the results obtained testing Aloa attack, presented in Section 11.1[1]. We organize this section as follows: first, we present the datasets used and their pre-processing (Section 11.2.1); then, we describe the different trained black-box models on which we tested the validity of our attack (Section 11.2.2). Lastly, in Section 11.2.3 we present the results of Aloa attacks to all the ML models, comparing the performance with respect to the original Mia and LabelOnly attack, and discussing the privacy risk of each of them.

11.2.1 Datasets

We use three classification datasets, each with different characteristics. We consider Adult, a bench-marking dataset composed of $48,842$ records and 15 variables, both numerical and categorical. This dataset describes employees with information like age, job, capital loss, capital gain, marital status, etc. The labels have values $<= 50K$ or $> 50K$ (in the following referred to respectively as Class 0 or Class 1), indicating whether the person will earn more or less than $50k$ in a fiscal year. This dataset was also used as a validation set of the attack for the Membership Inference Attack [109] and LabelOnly [125]. We perform Aloa also against Bank, which is a public dataset containing information on the customers of a bank, intending to classify the people as good or bad creditors. It is formed by $150,000$ records and 10 numerical variables, with information like age, monthly income and the number of loans already opened. The selection of this dataset is due to the huge amount of records available as well as the peculiarity of having only numerical variables. Lastly, we also consider Synth dataset, which is a synthetic dataset generated by exploiting a Gaussian mixture model. It has $30,000$ records and 30 numerical variables, with 15 classes. The selection of this dataset is due to the multi-class problem and to test the behaviour of the attack in a controlled environment due to the synthetic creation of the dataset.

For Adult, we removed the null values and analyzed the Pearson correlation among the variables, dropping some of them to obtain a correlation degree less than 80%. For the remaining categorical variables, we applied a one-hot encoding. For Bank, we removed the null values and the correlation analysis did not highlight any correlation value higher than 75%; thus we did not drop any variable. No further pre-processing was needed since the variables were all numerical. For Synth, instead, we did not perform any kind of pre-processing since the dataset was synthetically generated.

After the pre-processing step, we split each dataset into two subsets: (i) 70% of the original dataset (called D_b) is used to train and test the black-box models; (ii) the remaining 30% of the pre-processed data dataset (called D_s) is used for the learning process of the different attacks.

[1] The code developed for the experiments is in Python 3.8 and will be available upon acceptance

Data	Metric	DT	DT-O	RF	RF-O	NN	NN-O
Adult	TR Acc	0.84	1	0.84	1	0.83	1
	TS Acc	0.81	0.78	0.82	0.85	0.82	0.79
Synth	TR Acc	0.78	1	0.81	1	0.78	0.97
	TS Acc	0.77	0.69	0.79	0.78	0.78	0.70
Bank	TR Acc	0.84	1	0.98	1	0.93	1
	TS Acc	0.61	0.59	0.87	0.89	0.92	0.90

Table 11.1: Prediction performance of the black-box models for all the dataset selected. We report the Accuracy score both for the training set and the testing one to better appreciate the difference in performance in generalization capability for the *generalized* and *overfitted* models. Overall, we achieve good performace for all the models presented.

11.2.2 Black-boxes

Given each pre-processed dataset D_b, we split it into D_b^{train} (70% of it) and D_b^{test} (30% of it). We use D_b^{train} for training the black-box models. The ML models selected are described in the following:

1. Decision Tree (DT), selected for its simplicity, but prone to overfitting and to noise data;
2. Random Forest (RF), an ensemble model composed of multiple decision trees, with better performance with respect to the DT;
3. Neural Network (NN), a feed-forward network with some hidden layers, varying from 1 to 3, depending on the data in input;

For all selected models, we trained two variants: one, called *regularized*, with very good performance and with a good level of generalization, and another version called *overfitted* on purpose, thus specific to the input training dataset and with poor generalization capabilities. The choice of learning two variants of ML models resides in the fact that it has been experimentally proved that Mia leads to higher privacy risk when attacking overfitted models ([214, 215]). For this reason, in our experiments, we also want to evaluate how privacy exposure changes in relation to the level of overfitting of black-box models. We report the classification performance of these models in Table 11.1[2]. The results reported in this table show that all the black-box models have an overall good performance, with comparable performance for the RF and NN models, and a slightly worse prediction performance for the DT models, as expected. The model performance reported in the table also shows a different behaviour of the *regularized* models w.r.t. the *overfitted* ones.

[2] The results reported refer to the best set of hyperparameters determined by a grid search. The results were validated with a 3 fold cross validation

11.2.3 Evaluation of ALOA and comparison against competitors

In this section we present the privacy threats obtained by applying Aloa, LabelOnly and the original Mia to the trained black-boxes. In order to train all the attacks we need to have the shadow dataset D_s having the same format as the data used for training the black-box model. We employed two variants of this dataset, denoted as D_s^{stat} and D_s^{rand}, in our experiments. The former was designed to have the same statistical distribution as the original training dataset, whereas the latter was generated completely at random. We used D_s^{stat} for learning the LabelOnly attack because the procedure described in [125] requires training the shadow models on a dataset with similar distributions to those of the training data of the black-box, and it also exploits the distribution knowledge in the computation of the robustness score. Although, Aloa does not require the use of D_s^{stat}, as it is agnostic to the training data distributions, we conducted experiments with both D_s^{stat} and D_s^{rand} to evaluate the effectiveness of Aloa and having a complete comparison with LabelOnly. To ensure a clear understanding of the performance of the attack, we have balanced the D_s used for creating the attack models: having 50% of the rows of class In and 50% of class Out. This setting is the same used in [125] to make clear the comparison between our proposal and the attacks in the literature. Indeed, the balanced setting enables the possibility to compare the attack performance based on accuracy that, in this case, cannot be influenced by the under or over representation of one class with respect to the others. In this way, if the attack has more than 50% of accuracy, it poses a threat to privacy.

The results of the attacks are reported in Table 11.2 for Adult, Bank and Synth. Aloa was run three times for each black-box, with $n = 1000$ perturbations for each record of D_s (the same n is used for the training of LabelOnly attack), p_{min} and p_{max} set to 0.10 and 0.50 respectively, and a Bernoulli probability $p = 0.50$ for adding or subtracting the noise value. Mia was created with 8 shadow models and nn as final attack models. For LabelOnly we applied the same hyper-parameters as in the work [125]: $n = 1000$ perturbations, with a Bernoulli flip probability of 0.60% and a Gaussian noise with $\sigma = 0.04$. We remark that Mia and LabelOnly were tested on the D_s^{stat} due to the assumptions needed, while Aloa was tested both on D_s^{stat} and on D_s^{rand}.

Regarding Adult dataset, Mia and LabelOnly attacks performance is coherent with the one presented in their original papers. For the Mia, overall the attack against *regularized* models is not effective, apart from the decision tree with 51% of accuracy. On the other hand, the *overfitted* models are easily attacked, in particular rf-o and nn-o. However, the attack on the dt-o is not posing a privacy threat. This result may be due to the poor prediction performance of the dt-o for Adult. In fact, the overall accuracy of the model is 48%, suggesting that the model is not able to learn patterns in the data. Hence, the attack cannot have sufficient information from the confidence. By looking at the LabelOnly attack, it is not effective for all the *regularized* models, while it poses privacy threats for all the *overfitted* ones. Analyzing Aloa in both experimental settings, we have the same performance as LabelOnly on the *overfitted* models with 54% dt, 55% rf, 60% nn. Instead, by looking at the *regularized* models, we have in general better performance: the attack has gained 1-3% points in the attack compared to LabelOnly. With Aloa based on D_s^{stat} we are always better than LabelOnly except for the regularized nn, for which we have the same performance. Hence, for Adult dataset Aloa poses the worst privacy threats both for the *overfitted* and *regularized* models. Among the ML models, the attack generates more privacy leakage for the rf and nn models. This finding is reasonable because, as highlighted in prior works, more complex models learn more information.

For Bank dataset, the results are in line with the ones described for Adult, even if overall they are slightly lower. Interestingly, the improvement in terms of privacy threats posed by Aloa is more significant for the rf-o model (+3%) and lower for the nn-o one (+1%). This result may be due to the different structure of this dataset: it is composed of only a few numerical variables.

In Synth dataset we can better appreciate the effectiveness of Aloa: the trend is again that the attacks undermine the privacy more in the case of *overfitted* models, while *regularized* ones remain in danger, but with a lower privacy risk. Both Aloa and LabelOnly have better privacy threats with respect to Mia. However, Aloa in both settings shows better or comparable performance with respect to LabelOnly with an improvement for rf-o and nn-o. Comparing the two experimental setting of Aloa, our results indicate that the performance of our attack is generically consistent for both D_s^{stat} and D_s^{rand}, showing at most a discrepancy of 1% in accuracy. More importantly, they also demonstrate that even if our attack assumes an adversary with weaker knowledge with respect to LabelOnly, we achieve higher or comparable privacy risks. These findings have significant implications for privacy protection in ML models.

Overall, the experiments show that Aloa poses a worrying privacy risk, especially if the model is overfitted. The more complex a model is, the easier it is to overfit and experience higher privacy leakage. Comparing Aloa against the LabelOnly attack, we note that we have comparable or better performance for the *overfitted* models. This behaviour may be the result of the agnostic perturbation we perform, which is independent of the distributions of the input variables and hence Aloa is not affected by the slight changes in the data. We remark that this property is valid for both case where we use D_s^{stat} and D_s^{rand} since the perturbation mechanism remains always agnostic. Regarding Aloa against the original Mia, the performance of our method is overall better with the exception of rf-o. For this model, in fact, the accuracy of the Mia attack is always higher w.r.t. both LabelOnly and Aloa, highlighting that in the case of overfitted rf the added knowledge of the prediction probability has a greater impact in this setting. However, for the regularized rf and nn, instead, Mia shows higher accuracy and precision for the In class but an extremely low recall and hence F_1 score, showing that this attack is not stable.

To conclude, Aloa performed overall better than the LabelOnly, with an improvement up to 3%. Although this may seem like a small increase, it is a significant improvement

		Adult				Bank				Synth			
Attack	Model	P_{IN}	R_{IN}	$F1_{IN}$	Acc	P_{IN}	R_{IN}	$F1_{IN}$	Acc	P_{IN}	R_{IN}	$F1_{IN}$	Acc
MIA$_{\text{stat}}$	DT	0.51	0.53	0.52	0.51	0.50	0.58	0.54	0.51	0.49	0.51	0.50	0.49
	DT-O	0.48	0.62	0.55	0.48	0.49	0.49	0.49	0.49	0.51	0.47	0.49	0.51
	RF	0.45	0.27	0.34	0.47	0.53	0.16	0.24	0.51	0.73	0.04	0.08	0.51
	RF-O	0.59	0.68	0.63	**0.61**	0.67	0.60	0.63	**0.65**	0.90	0.86	0.88	**0.88**
	NN	0.53	0.04	0.08	0.50	0.45	0.03	0.06	0.50	0.52	0.30	0.38	0.51
	NN-O	0.55	0.94	0.69	**0.59**	0.53	0.85	0.65	0.54	0.58	0.59	0.58	0.58
LabelOnly$_{\text{stat}}$	DT	0.50	0.62	0.55	0.50	0.51	0.79	0.62	0.51	0.58	0.84	0.69	**0.62**
	DT-O	0.52	0.85	0.65	**0.54**	0.59	0.98	0.74	0.65	0.63	1.00	0.77	**0.70**
	RF	0.51	0.78	0.62	0.51	0.50	0.76	0.61	0.51	0.54	0.94	0.68	**0.57**
	RF-O	0.53	0.83	0.65	0.55	0.55	0.84	0.66	0.57	0.56	1.00	0.72	0.61
	NN	0.50	0.55	0.53	0.50	0.50	0.70	0.58	0.50	0.51	0.91	0.65	0.51
	NN-O	0.56	1.00	0.71	0.60	0.59	0.80	0.68	0.63	0.54	1.00	0.70	0.57
ALOA$_{\text{stat}}$	DT	0.51	0.81	0.63	0.52	0.51	0.80	0.62	**0.51**	0.58	0.84	0.69	**0.62**
	DT-O	0.53	0.86	0.65	**0.54**	0.59	1.00	0.74	**0.66**	0.63	1.00	0.77	**0.70**
	RF	0.52	0.51	0.52	**0.52**	0.51	1.00	0.67	**0.52**	0.54	0.83	0.66	**0.57**
	RF-O	0.54	0.65	0.59	0.55	0.56	0.98	0.71	**0.60**	0.58	0.96	0.72	0.63
	NN	0.53	0.49	0.51	**0.53**	0.50	0.76	0.60	0.49	0.51	0.89	0.65	**0.52**
	NN-O	0.56	1.00	0.72	**0.60**	0.58	0.98	0.73	**0.64**	0.55	1.00	0.71	**0.59**
ALOA$_{\text{rand}}$	DT	0.52	0.83	0.64	**0.53**	0.49	0.66	0.56	0.49	0.59	0.81	0.68	**0.62**
	DT-O	0.53	0.86	0.65	**0.54**	0.59	0.95	0.73	0.64	0.63	0.95	0.76	**0.70**
	RF	0.51	0.44	0.47	**0.52**	0.49	0.71	0.58	0.48	0.54	0.97	0.69	**0.57**
	RF-O	0.55	0.66	0.59	0.55	0.56	1	0.72	**0.60**	0.57	0.98	0.72	0.62
	NN	0.50	0.64	0.56	0.50	0.50	0.68	0.58	0.51	0.51	0.91	0.66	**0.52**
	NN-O	0.56	1	0.72	**0.60**	0.60	0.84	0.70	**0.64**	0.54	1	0.70	0.58

Table 11.2: Results of the attacks on the three datasets for all the black-box models selected. In bold are highlighted the highest privacy risks. We remark that for the MIA and LABELONLY we exploit the statistical dataset D_s^{stat}, while ALOA was tested both on the D_s^{stat} and on the D_s^{rand}, showing that it is completely agnostic w.r.t. the data and good stability. ALOA is the one with the highest privacy threats overall, showing good stability since we achieve similar performance for all the datasets.

in the context of privacy assessment, where every gain in performance can shed light on the privacy leakage of a model. Additionally, ALOA is more stable and the perturbation employed is data agnostic, without knowledge of the distribution of the features. Importantly, our attack showed better results in attacking regularized models compared to others.

Comparison between regularized and overfitted models. Recently, several works have empirically shown that if the model being attacked is overfitted, the attack will be much more damaging to the users of the training set [214, 215]. For this reason, we study the behavior of both models that generalize well and those that are overfitting. From the results in Table 11.2, all the *overfitted* models exhibit a higher degree of privacy leakage than regularized models, as evidenced in all three datasets, and particularly in the third one.

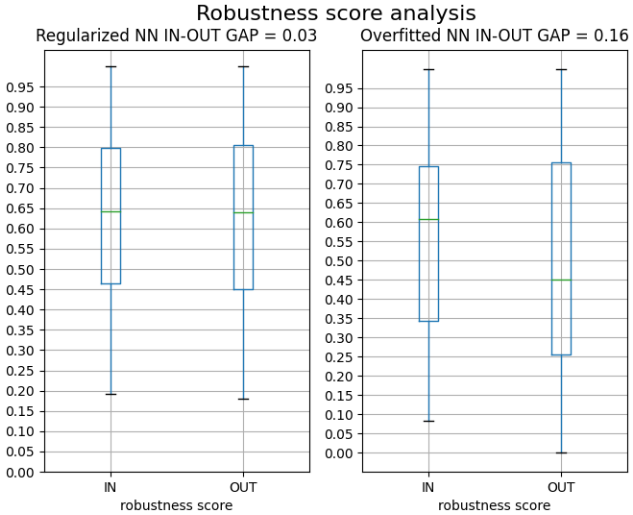


Figure 11.1: These two box plots show the robustness score behaviour for *overfitted* and *regularized* NN on Synth dataset. It is possible to see that the *overfitted* model exhibits a larger difference between the average In and Out robustness scores, which could potentially enable an attacker to distinguish between the two classes more easily. This confirms the existing link between model overfitting and privacy risk and the train-test gap [130]. On the other hand, the *regularized* model displayed a smaller gap between the two classes, indicating that separating the two classes is more difficult.

This dataset highlights the vulnerability of models that are not properly regularized and exhibit a gap between training and test accuracy. As outlined in [130], the gap between training and test accuracy is directly proportional to the efficacy of the accuracy of an attack - the larger the gap, the more effective the attack. To better analyze this aspect, we took advantage of the Synth dataset, which allows for a controlled study in which ML models achieve excellent performance and it is easy to overfit ML models. In Figure 11.1, it is possible to examine the difference in the performance of Aloa for NN and NN-O trained on the Synth dataset. In particular, we present a box plot on the robustness score, which shows that the *overfitted* model exhibits a larger difference between the average In and Out robustness scores, which could potentially enable an attacker to distinguish between the two classes more easily. In this way, we empirically prove the existing link between model overfitting and privacy risk as well as the train-test gap [130].

Analysis on the number of shadow models. There are many conflicting opinions in the literature about the use of shadow models, i.e., models that mimic the behavior of the original black-box. In fact, in the first publication of Mia [112] the authors used a

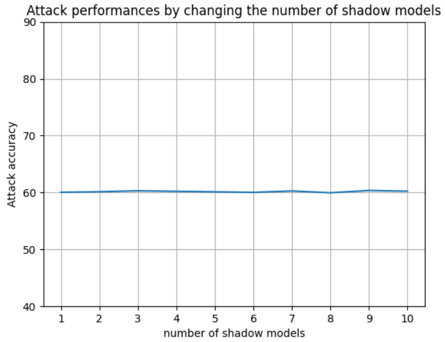


Figure 11.2: The performance of Aloa by changing the number of shadow models from 1 to 10 for the nn-o trained on the Adult dataset. It is clear that the performance of the attack are not affected by the number of shadow models.

large number of shadow models, while in LabelOnly [125], the authors present an attack model exploiting only one shadow model. We present the results with only one shadow model like [125] because we analyzed the effectiveness of using different shadow models, and our results highlight that for our purposes, using one or k models does not lead to any improvement. This behavior can be clearly seen in Figure 11.2, in which the performance of the attack on Adult is the same whether using only one or ten shadow models. Given this finding, our experiments were conducted with just one shadow model due to the better performance in terms of time.

11.3 Discussion

In this Chapter, we introduce Aloa, a variant of the LabelOnly membership inference attack. Our proposed attack is designed to be completely data agnostic, both in terms of shadow model training and perturbation mechanism. The perturbation employed in Aloa does not rely on any knowledge about the statistical distributions or domains of the features in the training data. This agnostic approach is a significant advantage from a privacy protection perspective, as it highlights the vulnerability of models to attacks that can be executed without any specific knowledge or assumptions.

Our experimental results demonstrate that Aloa outperforms the traditional LabelOnly attack, achieving an improvement of up to 3% in terms of attack accuracy, despite assuming an adversary with weaker prior knowledge. In the context of privacy assessment, even small gains in attack performance can provide valuable sensitive insights into the individuals represented in the data.

Additionally, Aloa exhibits good stability in terms of prediction performance compared to standard MIA attacks. It also demonstrates superior results in attacking regularized models when compared to other existing attacks. These findings highlight the robustness and effectiveness of ALOA as a method for evaluating the privacy of machine learning models.

Overall, ALOA offers a more comprehensive and reliable approach for assessing the privacy risks associated with machine learning models. Its agnostic nature and improved attack accuracy make it a significant concern for privacy protection, emphasizing the need for enhanced privacy measures and defenses.

Chapter 12

Evaluating the privacy exposure of Explainers

Employing sensitive data to train Machine Learning algorithms poses privacy issues, even if the data are kept private. With the widespread availability of big data, a new era is started in which decisions are being made based on the knowledge distilled from digital traces generated by the use of digital tools that are now present in everyday life. These traces are being collected and analyzed at individual, group, and societal levels, allowing for the development of powerful Artificial Intelligence (AI) systems that can be used in critical domains such as medicine, finance, and autonomous vehicles. However, these AI systems are often based on complex ensemble models and neural networks that are referred to as "black box" models due to their opaque internal structure and decision-making process. This lack of transparency and interpretability can limit the trust in these systems, especially in high-stakes decision-making. To address this, the eXplainable Artificial Intelligence (XAI) literature has developed two families of explainers: local explainers, which explain the reason for a specific instance classification, and global explainers, which explain the logic of the machine learning model as a whole. Additionally, AI systems based on machine learning models are vulnerable to various attacks, such as Model Inversion attacks and Membership Inference attacks, which can infer the data used for training the model by simply querying the model. In recent years, the number of privacy attacks of this kind increased considerably, having several variants of these attacks, with different assumptions at the beginning [125, 214, 216]. Thus, privacy mechanisms such as differential privacy are applied to counter potential privacy exposure. During the last year, other works have addressed the problem of the privacy threats posed by explainers. In particular, in [20], the authors attack the privacy of back propagation based explanations, which exploits the gradient, and perturbation based methods, such as SmoothGrad and LIME. Due to the structure of the explanations, they limit their analysis to neural networks, showing that

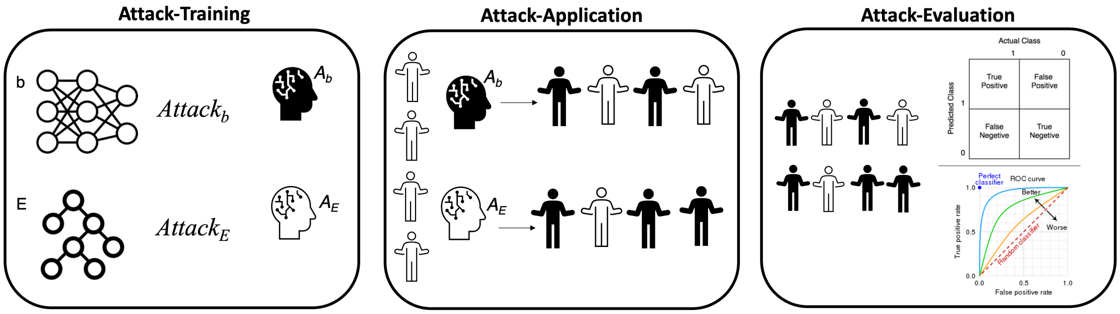


Figure 12.1: Schema of REVEAL, for Privacy Exposure of black-box models and their explainers. The framework is composed by three modules: the first one, *Attack Training* is devoted to train the chosen attack against the black-box and its explainer. Following, the *Attack Application* applies the trained privacy attacks to a dataset predicting the membership of each record. Lastly, the *Attack Evaluation* evaluates the changes of the privacy exposure when attacking the black-box and its explainer.

the back propagation-based explanations give rise to privacy risk, especially for minorities, while LIME and SmoothGrad do not. Lastly, also Quan et al. [21] deal with this topic from the point of view of images. They evaluate the effectiveness of the Membership Inference Attack and of the Evasion Attack against several explainers, such as SmoothGrad, LIME, IntGrad and GradCam.

In contrast to existing literature, our work introduces a novel framework, called REVEAL (pRivacy Exposure eValuatE surrogAte expLainer), for systematically evaluating the privacy risk associated with black-box models and their explainers, whether they are local or global, based on surrogate models. This framework is agnostic to the black-box structure and is generic in the privacy attack and the surrogate-based explainers used. The primary goal of REVEAL is to detect any changes in privacy exposure that may occur when publishing the black-box and/or its explainers. In the following, we first define the methodology of REVEAL, in Section 12.1 and then present the experiments, both for the local and global explainers, based on surrogate models, in Section 12.2.1.

12.1 Methodology

In this Section, we introduce REVEAL (pRivacy Exposure eValuatE surrogAte expLainer), a framework designed for assessing the privacy risk of black-box models and their explainers based on surrogate models. This framework, depicted in Figure 12.1, consists of three main modules: *Attack-Training*, which trains the attack models to be simulated, *Attack-Application*, which executes the trained attacks and *Attack-Evaluation*, which quantifies the privacy exposure introduced by an explainer. Algorithm 6 reports the pseudo-code of the whole assessment framework.

The instantiation of the three modules depends on the assumed threat model, the type of privacy attack to be performed, as well as the type of surrogate explainer used for explaining the black-box model. In the following, we describe the objective and role of each module within the framework.

Algorithm 6: PrivacyRiskExposure(b, E, D^{test}, Attack_b, Attack_E, BK)

```
$(A_b, A_E) \leftarrow$ Attack-Training$(b, E, \mathit{Attack}_b, \mathit{Attack}_E, BK)$
$(D^{\text{test}}_{\text{b-member}}, D^{\text{test}}_{\text{E-member}}) \leftarrow$ Attack-Application$(A_b, A_E, D^{\text{test}})$
$[\Delta_{Acc}, \Delta_P, \Delta_R] \leftarrow$ Attack-Evaluation$(D^{\text{test}}_{\text{b-member}},\ D^{\text{test}}_{\text{E-member}})$
return $[\Delta_{Acc}, \Delta_P, \Delta_R, \Delta_{F_1}]$
```

Algorithm 7: Attack-Training $(b, E, \mathit{Attack}_b, \mathit{Attack}_E, BK)$

```
$X_a^{\text{train}} \leftarrow \mathit{GenerateAttackDataset}(BK)$
$Y_b \leftarrow b(X_a^{\text{train}})$
$A_b \leftarrow$ Learning: $\mathit{Attack}_b(X_a^{\text{train}}, Y_b)$,
$Y_E \leftarrow E(X_a^{\text{train}})$
$A_E \leftarrow$ Learning: $\mathit{Attack}_E(X_a^{\text{train}}, Y_E)$,
return $A_b,\ A_E$
```

Attack-Training Module Given the back-box model b and its explainer E, the first module aims at learning two privacy attack models: the first one, namely A_b, is tailored to attack the black-box model b, while the second one, referred to as A_E, is tailored to attack the explainer E. As explained in Section 3.2.2, different attacks can be conducted for auditing a machine learning model. However, one of the most used attacks is the membership inference attack [109], aiming at inferring the membership of records to the training data of the machine learning model. This type of attack is also the foundation of other attacks aiming at extracting records from training data [217]. In this thesis, we propose to instantiate the model under analysis with learning algorithms for training these kinds of attacks. We highlight that the function $\mathit{Attack}_b(\cdot)$, aiming at learning the attack model A_b, can be different from $\mathit{Attack}_E(\cdot)$ that is used for learning A_E. This difference could be due to the fact that the black-box and the explainers might be ML models completely different that do not allow the attack under similar assumptions. For example, we could have a black-box that does not return the confidence vector for each prediction while its explainer could return it. Consequently, $\mathit{Attack}_E(\cdot)$ could exploit this additional information. The two functions executed in this module may be implemented using one of the algorithms available in the literature, such as the Membership Inference Attack [109], the Label-Only Attack [125], the Aloa Attack introduced in Chapter 11 or any other attack for ML models. Moreover, global and local explainers might require to design and develop a slightly different learning procedure for the attack. As an example, in the following, we propose learning an ensemble of attack models for attacking local explainers and assessing the privacy risks introduced by these types of explainers. The pseudo-code of this module is reported in Algorithm 7. We highlight that before training the two attacks, this module also generates the dataset X_a^{train} useful for learning the attacks. Such a dataset is labeled by using both the black-box (line 2, Alg. 7) and the explainer (line 4, Alg. 7). The type of attack dataset generated strongly depends on the background knowledge of the adversary BK. For example, if an adversary knows the distribution of the black-box training data, the attack can exploit this knowledge for the generation of the attack dataset. The performance of the attacks can be heavily affected by the properties of this dataset.

Algorithm 8: Attack-Application($A_b, A_E,\ D^{\text{test}}$)

```
1 $D^{\text{test}}_{\text{b-member}} \leftarrow A_b(D^{\text{test}})$
2 $D^{\text{test}}_{\text{E-member}} \leftarrow A_E(D^{\text{test}})$
3 return ($D^{test}_{b\text{-}member}, D^{test}_{E\text{-}member}$)
```

Attack-Application Module The second module of our framework is called *Attack-Application* and applies the attack models learned in the previous module A_b and A_E for inferring the membership of individual records to the training of b. The pseudo-code of this module is reported in Algorithm 8. In particular, given a set of records D^{test}, this module conducts the two attacks against the black-box and the explainer, respectively, and for each record outputs their membership prediction inferred by the two attack models, i.e., the labeled datasets $D^{\text{test}}_{\text{b-member}}$ and $D^{\text{test}}_{\text{E-member}}$ (line 1-2, Alg. 8). The two sets of labeled records are the base for computing and assessing the *Privacy Risk Exposure* for both the black-box model and its explainer. The instantiation of this module strongly depends on the attacks learned in the previous module and on the type of explainers (global vs. local). Indeed, later in this chapter, we will show that this module is the main difference between the assessment of global and local explainers.

Attack-Evaluation Module The output of the second module is then fed into the third and final module, the *Attack-Evaluation* module. This module aims to analyze and quantify the change of privacy risk exposure between the black-box model b and its explainer E. The analysis can be performed using different metrics that evaluate the performance of the attack models in predicting the membership of the individual records to the training data

```
Algorithm 9: Attack-Evaluation($D^{\text{test}}_{\text{b-member}}$, $D^{\text{test}}_{\text{E-member}}$)
1 $C_{\text{b-member}} \leftarrow ConfusionMatrix(D^{\text{test}}_{\text{b-member}})$
2 $C_{\text{E-member}} \leftarrow ConfusionMatrix(D^{\text{test}}_{\text{E-member}})$
3 $[\Delta_{Acc}, \Delta_P, \Delta_R, \Delta_{F_1}] \leftarrow Compute\Delta(C_{\text{b-member}},\ C_{\text{E-member}})$
4 return $[\Delta_{Acc}, \Delta_P, \Delta_R, \Delta_{F_1}]$
```

of b. This module first evaluates the confusion matrix for the attack against the black-box, A_b (line 1, Algorithm 9), then for the attack against the explainer, A_E (line 12 Algorithm 9). From these partial results, the module performs the evaluation in terms of standard ML metrics. In particular, this module computes the difference in privacy exposure in terms of accuracy (Δ_{Acc}) , precision (Δ_P), recall (Δ_R) and f-measure (Δ_{F_1}) of the two attack models. In other words, each Δ_μ is computed as $\Delta_\mu = \Delta^E_\mu - \Delta^b_\mu$, where μ denotes one of the metrics among accuracy, precision and recall. Analyzing only accuracy for evaluating membership inference attacks could be inadequate because these metrics associates equal costs to false positive (false memberships) and false negative (false non-memberships). The first type of error reduces the utility of the attack, while the second one reduces the identification of real members. An attack should maximize the true positive rate (or recall) because it measures how many members are identified. We highlight that negative values of Δ for a given measure μ mean that the explainer tends to mitigate the privacy risks of the black-box, i.e., the explanation procedure is confusing the attack; positive values of Δ instead highlight higher privacy risks due to the level of transparency introduced by the explainer; lastly, $\Delta = 0$ means that pairing an explainer with a black-box classifier is not increasing the privacy risks.

12.1.1 REVEAL for global and local explainers

REVEAL is a framework for assessing the privacy exposure in black-boxes and their explainers. The methodology presented is generic and can work with any ML model, as well as any surrogate-based explainer, but needs to be instantiated differently depending on the attack considered due to the different background knowledge possessed by the adversary. In this thesis, we propose to instantiate such a framework with attacks belonging to the family of *membership* attacks and we investigate the impact of different levels of adversary background knowledge on the success of the attack. In particular, we investigate the privacy exposure of global and local explainers under the attacks Mia, LabelOnly and Aloa. Although we have already described Mia and LabelOnly in Chapter 3.2.2 and Aloa in Chapter 11, in the following, we will report a compact description of these three attack models for facilitating the reader in the understanding of the rest of the chapter.

Membership Inference Attack Mia [109] assume that a ML algorithm is used to train a classifier b that captures the relationship between data records and their labels. In order to attack b trained on D_b^{train}, Mia defines an attack model $A(\cdot)$: it is a machine learning model able to discern if a record was part of the training dataset D_b^{train} or not. Note that, D_b^{train} is composed by $(x^i, y_o^i)_b$, where y_o^i is the true labels associated to x_b^i. In practice, the attack $A(\cdot)$ is a binary classifier that predicts In if the record was part of the training set or Out otherwise. $A(\cdot)$ is trained on a dataset D_a^{train}: $(x^i, y^i)_a$, where each x_a^i is composed by the label predicted by the classifier b for a record under analysis and its probability vector $\overline{y^i}$ of length L obtained by querying a shadow model $s^i(\cdot)$ mimicking b; while y_a^i is the correct membership label and that can be In or Out. The attack model $A(\cdot)$ is a voting model composed of L machine learning models: one for each output class of the classifier model under attack. The key factor in this attack is the knowledge of the probability vector: given how the probabilities in $\overline{y}_b$ are distributed around the true value of the record, the attack model computes the membership probability $\Pr\{(x, y) \in D_b^{train}\}$, which is the probability that x belongs to the In class, i.e., it is part of the training set. To obtain the dataset $(x^i, y^i)_a$, on which the Mia model $A(\cdot)$ is trained, the authors used *shadow models*. In the original paper, the authors assume a black-box setting in which there is no knowledge about either the type of classifier to be attacked or the training dataset used to train it. In the following, we use the term black-box model to indicate the classifier to be attacked. To overcome the limitation of absence of knowledge on data and model, they employed a set of k shadow models $s^i(\cdot)$: machine learning models trained to mimic the decisions of the black-box model $b(\cdot)$ we would like to attack. These shadow models are trained on D_s^{train}: $(x^i, y^i)_s$, in which x_s^i has the same format and similar distribution w.r.t. to the dataset employed to train the black-box model X, while y_s^i is the predicted class obtained querying the black-box model $b(\cdot)$. After the training, we know which record was part of the training dataset (class In) for each shadow model and which was part of the test one (class Out). Hence, we can exploit this information to create a supervised training dataset for training the attack model $A(\cdot)$, which is D_a^{train}.

We highlight that the datasets employed for training the shadow models are disjoint from the unknown dataset used to train the black-box model. Shokri et al. [109] tested different kinds of training data for the shadow models: (i) a *random* dataset, where data are randomly generated and then labeled querying the black-box model; (ii) a *statistical* dataset, in which the attacker knows the statistical distribution of the original training dataset, hence he/she can exploit this information to create a synthetic dataset; (iii) a *noise* dataset, in which the attacker knows a portion of data from the same distribution of the original training dataset, but with some noise. These different types of training datasets for the shadow models allow for privacy attacks of different strengths: from the least severe attack, the random one, to the most powerful, i.e., the noise one. Clearly, the three types of datasets correspond to three different levels of background knowledge of the adversary.

Label Only Attack Membership Inference Attack Choquette-Choo et al. [125] design a variant of MIA which relaxes some requirements of the original attack. An in-depth description of this attack model is reported in Section 3.2.2. Given a black-box model b, LABELONLY $A_{LO}(\cdot)$ targets it by exploiting only the hard labels, i.e., the output predictions of the model under analysis. Hence, the probability vector $\overline{y^i}$, employed by MIA, is not exploited in LABELONLY. In particular, it develops a procedure that derives the robustness of a model to perturbations and uses it as a proxy for model confidence in its predictions. The basic intuition is that records which exhibit high robustness belong to the training dataset. $A_{LO}(\cdot)$ exploits a dataset D_s^{train} for training only one shadow model $s(\cdot)$, i.e., a ML model mimicking the decision of black-box model b. The dataset D_s^{train}: $(x^i, y^i)_s$ is composed of records with the same format and similar distribution w.r.t. to the dataset employed to train the black-box model b, and is labeled by the predicted class obtained querying b. After training the shadow model, we know which record was part of the training dataset (class IN) of the shadow model and which was part of the test one (class OUT). For each tuple x_s^i the algorithm generates a set of records resulting from its perturbation and labels the generated records using the trained shadow model. Analyzing the percentage of generated records having the same predicted class of x_s^i, the algorithm computes the robustness score of the black-box with respect to the x_s^i classification. Then, the attack uses an iterative thresholding procedure on the robustness scores, assigned to each record of the training and testing dataset of the shadow model, to find a threshold on the scores to well separate the records with class IN and OUT. The attack will use this threshold for classifying new records as part of the training of the black-box or not.

Agnostic Label Only Attack Membership Inference Attack ALOA, a variant of the LABELONLY attack, has been presented in Chapter 11. Similarly to the LABELONLY, ALOA does not require access to the probability vector. However, this privacy attack has weaker assumptions regarding LABELONLY, since it does not need to know any statistics about the data used for training the ML model to attack. $A_{\text{aloa}}(\cdot)$ exploits a dataset D_s^{train} for training only one shadow model $s(\cdot)$, i.e., a ML model mimicking the decision of black-box model b. The dataset D_s^{train}: $(x^i, y^i)_s$ is composed of randomly generated records with the same format of the training dataset of the black-box model b, and is labeled by the predicted class obtained querying b. At this point, similarly to the other attacks, we know which record was part of the shadow model's training dataset (class IN) and which was part of the test (class OUT). At this point, ALOA generates a set of synthetic records by perturbing the record under analysis. This perturbation procedure is completely agnostic and does not exploit any kind of statistics about the original dataset. As in LABELONLY, the percentage of generated records having the same predicted class of the record under analysis is used to compute the robustness score of the black-box. At this point, the robustness score is exploited to find the threshold that best separates the classes IN and

Out.

REVEAL for global explainers

Instancing reveal for the assessment of the privacy risk exposure of global explainers is straightforward. A global explainer based on a surrogate model E is a ML model which is imitating the global behavior of a black-box classifier b. As a consequence, it is enough to follow the procedure described in the previous section, implementing the training of one of the membership-based attack models presented above. In particular, in the first module, *Attack-Training*, trains both *(i)* a privacy attack, named A_b, against b is trained, being it Mia, LabelOnly or Aloa; and *(ii)* a privacy attack, named A_E, against the explainer E. Then, these two attacks are fed into the *Attack-Application* module, which applies these attacks to a test dataset, namely D^{test}. The result will be to obtain two labeled datasets: one which for each element of the dataset has the membership class In or Out determined by the attack A_b, and one with the class determined by the attack A_E. Lastly, in the *Attack-Evaluation* module it is quantified the probability of success of the two attacks computing the difference in the performance of both concerning *precision*, *recall*, and *accuracy.*

REVEAL for local explainers

When employing the reveal framework to assess the privacy vulnerability of a black-box model and its local models, it is essential to tailor the attack methodology to the specific scenario being analyzed, where E represents a collection of local surrogate models. In this context, each local explainer is customized to describe a small portion of the decision boundary of the black-box. Therefore, to ensure that the entire decision boundary of the black-box model b is properly described, it is imperative to consider a variety of local explainers that capture different types of local knowledge. This means that if an adversary wants to jeopardize the privacy of a black-box attacking its local explainer, it needs to generate a set of local explainers that all together approximate the black-box's global behavior. To this end, we propose a privacy attack procedure designed to target local surrogate-based explainers. Specifically, the procedure assumes E as a set of local surrogates, i.e., $E = e_1, e_2, \dots, e_n$. Following the pseudo-code outlined earlier in Algorithm 7, the $Attack_E$ is computed as an ensemble of multiple attacks, with one attack tailored for each local surrogate model in E. The resulting ensemble of attacks is denoted as $A_E = Ae_1, A_{e_2}, \dots, A_{e_n}$ and is passed, along with the attack tailored for the black-box model A_b, to the *Attack-Application* module. In this setting, the module needs to evaluate the effectiveness of the ensemble of attacks A_E. The application of A_E can be instantiated in different ways, depending on the specific information assumed by the attack. In the following, we present two ways for implementing *Attack-Application* in the local setting depending on the knowledge the attack produces. In particular, we consider two approaches: the *Confidence Vector* Approach, based on the prediction probabilities vectors, applicable to every membership attack based on ML attack models, such as the original Mia; and the *Threshold* Approach, tailored for the attacks which do not create a ML model, but a thresholding procedure, such as LabelOnly and Aloa.

For the **Confidence Vector Approach**, we apply an evaluation procedure that exploits the prediction probabilities vectors outputted by the attack models. This setting is tailored for methods such as Mia, which trains a ML attack model for each target output from the black-box model. Having created these attacks based on ML models, we assume to have access to the prediction confidence vectors, $c = [c_{\text{In}}, c_{\text{Out}}]$, where c_{In} is the probability that the record belongs to class In, while c_{Out} is the probability that the record belongs to class Out and the sum of all the two elements is equal to 1. Hence, we exploit this information to identify among the different attacks only the ones that are the most confident record-wise. Technically, for each record x, we apply all the attack models, obtaining a confidence vector for each one, i.e., $C_x = \{c^{A_1}, c^{A_2}, \dots, c^{A_n}\}$, where n is the number of attacks for the n local explainers. At this point, for each vector c^{A_i}, we compute the absolute difference between the two probabilities, i.e., $d_i = |c_{\text{In}}^{A_{e_i}} - c_{\text{Out}}^{A_{e_i}}|$.

Once we get the corresponding d value for each attack model, we select only the attack models expressing significant confidence in their decisions. To this end, we select the models A_{e_j} having a d_j value above the average. In particular, we use the following constraint for selecting the attack set: $\{A_{e_j} | d_j \geq (avg(d_1, d_2, \dots, d_n) + \sigma(d_1, d_2, \dots, d_n))\}$, where σ is the standard deviation. Among the top attack models selected, we apply a majority voting procedure to select the final membership prediction for each record.

In the case of **Threshold Approach**, the *Attack-Evaluation* strategy is tailored for membership attacks that do not train ML models as attacks but use a thresholding procedure. Examples of attacks of this family are LabelOnly and Aloa. In this setting, we exploit the different information available, which is the threshold found and used by each attack for the membership prediction. Given the record x under analysis, by applying the attack A_{e_i} and we obtain a robustness score $s^{A_{e_i}}$, which is compared to the score threshold $st^{A_{e_i}}$ for determining In or Out class. Hence, we exploit the absolute distance between the robustness score and the score threshold (i.e., $d_i = |s^{A_{e_i}} - st^{A_{e_i}}|$) to identify the most reliable attacks. In particular, we are interested in the attacks which have a greater distance between the robustness score of the record and the score threshold. We select only the top attack models, exploiting the *elbow* method, i.e., we select the most important models with a d value greater or equal to the one corresponding to the knee in the curve of the ordered d values Formally, we select the following set of attack models: $\{A_{e_j} | d_j > \text{elbow}(d_1, \dots, d_n)\}$. We apply a majority voting strategy to obtain the final membership prediction on the set of attack models selected. These two evaluation methods presented are only two possible initializations, dependent on the privacy attack considered.

Data	Class Balance	Metric	dt	rf	trepan-rf	nn	trepan-nn
Adult		F_{1_1}	0.63 (0.02)	0.70 (0.02)	**0.98** (0.00)	0.67 (0.02)	**0.77** (0.02)
		P_1	0.60 (0.01)	0.69 (0.02)	**0.99** (0.00)	0.69 (0.02)	**0.82** (0.00)
	$C_1 = 24\%$	R_1	0.58 (0.05)	0.87 (0.03)	**0.98** (0.01)	0.67 (0.03)	**0.73** (0.01)
	$C_0 = 76\%$	F_{1_0}	0.90 (0.00)	0.86 (0.00)	**0.99** (0.00)	0.89 (0.00)	**0.99** (0.00)
		P_0	0.87 (0.01)	0.95 (0.00)	**0.98** (0.00)	0.90 (0.00)	**0.99** (0.00)
		R_0	0.92 (0.01)	0.80 (0.01)	**0.99** (0.00)	0.89 (0.01)	**0.98** (0.01)
Synth	**Class Balance**	**Metric**	**dt**	**rf**	**trepan-rf**	**nn**	**trepan-nn**
	$C_2 = 33\%$	F_{1_2}	0.77 (0.01)	0.99 (0.01)	0.95 (0.01)	1.00 (0.00)	**0.72** (0.01)
		P_2	0.96 (0.01)	0.98 (0.01)	**0.94** (0.00)	1.00 (0.00)	**0.75** (0.00)
		R_2	0.96 (0.01)	1.00 (0.00)	**0.98** (0.01)	0.99 (0.01)	**0.70** (0.01)
	$C_1 = 33\%$	F_{1_1}	0.81 (0.02)	0.89 (0.01)	**0.82** (0.01)	**0.93** (0.01)	0.67 (0.01)
		P_1	0.83 (0.00)	0.88 (0.00)	**0.84** (0.00)	**0.94** (0.00)	0.64 (0.00)
		R_1	0.80 (0.00)	0.89 (0.01)	**0.80** (0.01)	**0.93** (0.01)	0.72 (0.01)
	$C_0 = 33\%$	F_{1_0}	0.80 (0.02)	0.88 (0.01)	**0.82** (0.01)	**0.93** (0.01)	0.89 (0.01)
		P_0	0.80 (0.00)	0.89 (0.00)	**0.80** (0.00)	**0.92** (0.00)	0.90 (0.00)
		R_0	0.82 (0.00)	0.88 (0.01)	**0.86** (0.01)	**0.94** (0.01)	0.88 (0.02)
Bank	**Class Balance**	**Metric**	**dt**	**rf**	**trepan-rf**	**nn**	**trepan-nn**
		F_{1_1}	0.35 (0.01)	0.77 (0.01)	**0.99** (0.01)	0.78 (0.01)	**0.84** (0.01)
		P_1	0.38 (0.01)	0.83 (0.01)	**0.98** (0.02)	0.77 (0.01)	**0.86** (0.00)
	$C_1 = 8\%$	R_1	0.34 (0.01)	0.75 (0.04)	**0.99** (0.01)	0.76 (0.04)	**0.82** (0.01)
	$C_0 = 92\%$	F_{1_0}	0.95 (0.02)	0.92 (0.01)	**0.99** (0.01)	0.77 (0.01)	**0.95** (0.01)
		P_0	0.95 (0.00)	0.91 (0.00)	**0.99** (0.00)	0.78 (0.00)	**0.96** (0.02)
		R_0	0.95 (0.00)	0.92 (0.01)	**0.98** (0.01)	0.79 (0.01)	**0.95** (0.01)

Table 12.1: Predictive performance of the models for ADULT, SYNTH and BANK dataset on the test set. The results are validated with 3-fold cross-validation (we provide the mean and the standard deviation between brackets). This table highlights the extremely good predictive performance of TREPAN w.r.t. DT and RF, which is almost always the best model, except for SYNTH. TREPAN was trained to exploit an enriched dataset, but in this case we tested the predictive performance on the same test set of the black-boxes for comparison purposes.

12.2 Experiments

In this section, we present the experiments conducted to validate our methodology. We focus on tabular data and will use the datasets ADULT, BANK and SYNTH introduced in the previous chapter in Section 11.2.1. Section 12.2.1 presents the ML models employed in REVEAL for our experiments.

12.2.1 Machine Learning models and Explainers

For validating the proposed assessment methodology we used ADULT, BANK and SYNTH datasets. We split each dataset into two subsets: (i) 70% of the original dataset (called D_b) is used to train and test the black-box models; (ii) the remaining 30% of the pre-processed data dataset (called D_s) is used for the learning process of the different attacks.

On each of the three datasets we train different ML models: a simple *Decision Tree* (DT), a simple explainable by design method which is exploited as a benchmarking; a *Random Forest* (RF), an ensemble method based on trees; and a feedforward *Neural Network* (NN). We chose these ML methods to examine the behavior of our methodology on models that have completely different structural characteristics. The training results of these models are reported in Table 12.1. Overall, the performance of the models is good, except for the DT, which suffers greatly from the imbalance between the classes, especially high in the case of the BANK dataset.

After having trained the ML models, we also train the explainers. For the *global* case we consider TREPAN, a tree-based explainer fitted on an enhanced version of the original training dataset, labeled by the black-box model b. Therefore, we train a TREPAN-RF model for explaining the RF and a TREPAN-NN model for the NN. The performance of the TREPAN model is reported in Table 12.1, from which it is possible to see that the performance of the TREPAN models is extremely good for all the datasets. For the *local* case, instead, we select LORE, a post-hoc agnostic explainer that exploits a local DT surrogate model to extract rules and counterfactual rules. In this case, we train one local surrogate model for each record to explain. The average fidelity of these models is 0.97 ± 0.08.

12.2.2 Reveal results

After the fitting of the black-box models and their explainers, we can now test the performance of REVEAL. Following the experimental setting presented in [109], we consider two settings for the fitting of the shadow models: the worst case scenario, called *noise* dataset from now on, and the best case scenario, called *random* dataset. In the case of the noise dataset, we assume the attacker has access to a noise version of a set of data from the same distribution of the data exploited in the training set of the black-box. Technically, we add 10% of noise to a piece of the original dataset not exploited during the training of the black-box. For the random case, instead, we assume the attacker does not know the dataset used for training the black-box, apart from the number of variables of the original data. Therefore, the adversary randomly generates a dataset and labels it by querying the black-box. The choice of these two types of dataset is due to the different settings they create: the noise dataset assumes a favorable setting for the attacker, who, through some public information or misappropriation of information, is able to obtain a piece of data from the same distribution as the original one, albeit with some noise. This setting is unrealistic, but it is also the one in which the MIA allows greater privacy exposure. In addition, LABELONLY requires knowledge of statistical information from the original data, so this setting is in line with the assumptions of this attack. For the second setting, on the other hand, zero knowledge on the part of the attacker is assumed, and it is, therefore, a more realistic but also more worrisome case, as having a privacy risk, in this case, is much worse than the previous setting because it requires less knowledge on the part of the attacker. In addition, it is interesting to see the behavior of the various attacks in this setting: based on the work in the literature, we expect to have a decrease in privacy exposure for MIA and LABELONLY, while performance should remain roughly similar for the ALOA case, which is specifically developed for this setting.

For each combination of black-box model, explainer and kind of dataset to generate the shadow models, we train MIA, LABELONLY and ALOA, both for the global and the local explanations. Due to the different methodologies applied for the local and global case, in the following, we present the results separately.

Results attacking the Global Explainers To evaluate our methodology, we attack both the black-box models and their surrogate-based explainers employing three different attacks: MIA, LABELONLY and ALOA. For training each MIA, we train 6 shadow models with the objective of mimicking the black-boxes. The shadow models are trained to employ the best set of hyper parameters found using a grid search. All of the shadow models have an accuracy above 80%. Then, from the shadow models, we extract the supervised training dataset D_a^{train} to train the attack model. We remark that the MIA creates an attack model composed of a ML model for each label L. Hence, in our case, we obtain two (or three for the SYNTH dataset) RF attack models for each attack. This procedure is applied only in the MIA attack since LABELONLY and ALOA only produce one final attack model, independently of the number of classes the black-box model considers. Hence, for MIA, for the different attack models, we first search for the best set of hyperparameters, obtaining an accuracy above 94% for all the models when tested on a portion of test data D_a^{test}. For the LABELONLY and ALOA, we have just one shadow model, a RF as for the MIA, with an accuracy above 80%. After the fitting of the shadow model, both of the models require the computation of the robustness score, done creating 1000 perturbations for each record, and the final attack model is not a ML model but a thresholding model, adaptively selected depending on the data in input.

Regarding the *global* explainers, the results are reported in Table 12.2 and in 12.3, respectively for the *noise* dataset and the *random* dataset. In the tables are reported F_1, P and R for the IN class, which is the most important class for this setting, since it represents the re-identified users. Most importantly, we report the Δ_R, our evaluation metric for testing the privacy exposure change. This metric reports the difference between the recall of the black-box models w.r.t. the DT, as well as the difference between the recall of the black-box models with respect to the corresponding TREPAN models. In this setting the recall of the IN class is particularly important since it describes how many training records we can reconstruct. A positive value for Δ_R means that the privacy exposure of the DT or of the TREPAN model is higher w.r.t. the black-box models. From both of the settings, i.e. *noise* or *random*, we can see that for all the models, we have a positive Δ_R, highlighting worse privacy issues when attacking the DT and TREPAN models w.r.t attacking the black-boxes.

For the MIA attack, we can notice that we have a higher privacy exposure for the DT and the TREPAN models w.r.t. the black-boxes in both of the settings, i.e. *noise* and *random*. The only exception to this trend is $\Delta_R(\text{DT} - \text{RF})$ for BANK, in which the RF has a higher privacy exposure w.r.t. the DT, even if for a small amount. The negative values for this metric may be due to the poorer performance of the DT in this setting, which also makes the attack less robust. Regarding the black-boxes, it is possible to see that overall the *random* poses a smaller privacy treats, which is insignificant in the case of NN (the highest recall in this setting for NN is 0.33 for SYNTH).

We have the same trends presented for MIA also for LABELONLY, especially for the *noise* setting, even if the $\Delta_R(\cdot)$ show lower values w.r.t. the MIA. This result is due to the fact that this attack is better in attacking the black-box models w.r.t. MIA; hence there is a privacy exposure both when attacking the explainers and the black-boxes. Regarding the *random* case, LABELONLY obtains a higher privacy exposure w.r.t. MIA, even if with a decrease in performance w.r.t. the *noise* case. In particular, LABELONLY can attack both the NN and the RF for the SYNTH dataset. The trend of a higher privacy exposure for the black-boxes in LABELONLY w.r.t. MIA can also be seen for the RF of BANK and for the NN of ADULT, even if in a smaller way. However, we remark that the SYNTH dataset is the only synthetically generated dataset.

The performance of ALOA are similar to the LABELONLY for the *noise* case. We can also notice similar results regarding $\Delta_R(\cdot)$: the values are lower w.r.t. MIA, but all the attacks show a privacy exposure, both for the explainers and the black-boxes, the first higher than the latter. For the *random* setting, instead, ALOA shows a higher privacy exposure w.r.t. LABELONLY and MIA. This result was expected due to the procedure of the attack. In fact, ALOA does not require any kind of background knowledge about the original training dataset, not even its statistics. Therefore, having a privacy exposure with this attack highlights an even more dangerous setting since the attacker can perform it with the only assumption of knowing the shape of the input data, which is public information for on-demand services.

Figure 12.2 reports the Critical Difference Plot, showing the overall ranking of the three different attacks against global surrogate-based explainers and their black-boxes, both in the *noise* and *random* settings. From this plot, we can observe that there is no statistical difference among the attacks, showing an overall threatening setting for the privacy of the people in the training datasets. Regarding the ranking, ALOA against the TREPAN models is the one which exposes the highest privacy risk, followed by MIA and LABELONLY against TREPAN. However, the three methods against the global explainers have close values in the ranking, with a clear separation between them and the attacks against the black-boxes. In fact, all the attacks against the black-boxes are less powerful w.r.t. the attacks against the explainers, even if the level of privacy exposure remains high. For the attacks against the black-boxes, the ranking is ALOA, LABELONLY and MIA, but the last one is the lowest in the rank, significantly separated from ALOA and LABELONLY. We remark that, although this analysis highlights that the difference in performance among the three attacks is not statistically significant, we need to consider that Aloa is working with very poor background knowledge by the adversary.

Results attacking the Global Explainers To evaluate our methodology we attack both the black-box models and their surrogate based explainers employing three different attacks: Mia, LabelOnly and Aloa. For training each Mia, we train 6 shadow models with the objective of mimicking the black-boxes. The shadow models are trained employing the best set of hyper parameters found using a grid search. All of the shadow models have an accuracy above 80%. Then, from the shadow models, we extract the supervised training dataset D_a^{train} to train the attack model. We remark that the Mia assumes the attack model as an ensemble model composed of a ML model for each label L. Hence, in our case, we obtain two (or three for the Synth dataset) rf attack models for each attack. Also in this case, for the different attack models, we first search for the best set of hyperparameters, obtaining an accuracy above 94% for all the models, when tested on a portion of test data D_a^{test}. For the LabelOnly and Aloa we have just one shadow model, a rf as for the Mia, with an accuracy above 80%. After the fitting of the shadow model, both of the models require the computation of the robustness score, done creating 1000 perturbations for each record, and the final attack model is not a ML model but a thresholding model, adaptively selected depending on the data in input.

Regarding the *global* explainers, the results are reported in Table 12.2 and in 12.5, respectively for the *noise* dataset and the *random* dataset. In the tables are reported F_1, P and R for the In class, which is the most important class for this setting, since it represents the users that are re-identified. Most importantly, we report the Δ_R, which is our evaluation metrics for testing the changing in privacy exposure. This metric reports the difference between the recall of the black-box models w.r.t. the dt, as well as the difference between the recall of the black-box models with respect to the corresponding trepan models. In this setting the recall of the In class is particularly important since it describes how many training records we can reconstruct. A positive value for Δ_R means that the privacy exposure of the dt or of the trepan model is higher w.r.t. the black-box models. From both of the settings, i.e. *noise* or *random*, we can see that for all the models we have a positive Δ_R, highlighting worse privacy issues when attacking the dt and trepan models w.r.t attacking the black-boxes.

For the Mia attack, we can notice that we have a higher privacy exposure for the dt and the trepan models w.r.t. the black-boxes in both of the settings, i.e. *noise* and *random*. The only exception to this trend is $\Delta_R(\text{DT} - \text{RF})$ for Bank, in which the rf has a higher privacy exposure w.r.t. the dt, even if for a small amount. The negative values for this metric may be due to the poorer performance of the dt in this setting which make also the attack less robust. Regarding the black-boxes, it is possible to see that overall the *random* poses a smaller privacy treats, which is insignificant in the case of NN (the highest recall in this setting for NN is 0.33 for SYNTH).

We have the same trends presented for MIA also for LABELONLY, especially for the *noise* setting, even if the $\Delta_R(\cdot)$ show lower values w.r.t. the MIA. This result is due to the fact that this attack is better in attacking the black-box models w.r.t. MIA, hence there is a privacy exposure both when attacking the explainers and the black-boxes. Regarding the *random* case, LABELONLY obtains a higher privacy exposure w.r.t. MIA, even if with a decrease in performance w.r.t. the *noise* case. In particular, LABELONLY can attack both the NN and the RF for the SYNTH dataset. The trend of an higher privacy exposure for the black-boxes in LABELONLY w.r.t. MIA can be seen also for the RF of BANK and for the NN of ADULT, even if in a smaller way. However, we remark that the SYNTH dataset is the only synthetically generated dataset.

The performance of ALOA are similar to the LABELONLY for the *noise* case. We can notice similar results also in the case of the $\Delta_R(\cdot)$: the values are lower w.r.t. MIA, but all the attacks show a privacy exposure, both for the explainers and the black-boxes, the firsts higher than the latter. For the *random* setting, instead, ALOA shows a higher privacy exposure w.r.t LABELONLY and MIA. This result was expected due to the procedure of the attack. In fact, ALOA does not require any kind of background knowledge about the original training dataset, not even the statistics of it. Therefore, having a privacy exposure with this attack highlights an even more dangerous setting, since the attacker can perform it with the only assumption of knowing the shape of the input data, which is a public information for on-demand services.

Figure 12.2 reports the Critical Difference Plot, showing the overall ranking of the three different attacks against global surrogate-based explainers and their black-boxes, both in the *noise* and *random* settings. From this plot we can observe that there is not a statistical difference among the attacks, showing an overall threatening setting for the privacy of the people in the training datasets. Regarding the ranking, ALOA against the TREPAN models is the one which exposes the highest privacy risk, followed by MIA and LABELONLY against TREPAN. However, the three methods against the global explainers have close values in the ranking, with a clear separation between them and the attacks against the black-boxes. In fact, all the attacks against the black-boxes are less powerful w.r.t. the attacks against the explainers, even if the level of privacy exposure remains high. For the attacks against the black-boxes, the ranking is: ALOA, LABELONLY and MIA, but the last one is the lower in the rank, significantly separated from ALOA and LABELONLY.

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
MIA	ADULT	$F_{1_{\mathrm{In}}}$	**0.79** (0.01)	0.70 (0.01)	**0.77** (0.01)
		P_{In}	**0.80** (0.02)	0.80 (0.03)	**0.80** (0.00)
		R_{In}	**0.77** (0.01)	0.67 (0.01)	**0.72** (0.02)
		Δ_R	**0.10**	-	**0.05**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.79** (0.01)	0.63 (0.02)	**0.70** (0.01)
		P_{In}	**0.80** (0.02)	0.79 (0.00)	**0.79** (0.03)
		R_{In}	**0.77** (0.01)	0.53 (0.03)	**0.64** (0.00)
		Δ_R	**0.24**	-	**0.11**
	SYNTH	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\mathrm{In}}}$	**0.77** (0.01)	0.76 (0.01)	**0.78** (0.00)
		P_{In}	0.70 (0.01)	**0.70** (0.00)	0.70 (0.00)
		R_{In}	**0.85** (0.02)	0.82 (0.01)	**0.87** (0.03)
		Δ_R	**0.03**	-	**0.05**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.77** (0.01)	0.78 (0.02)	**0.79** (0.01)
		P_{In}	0.70 (0.01)	**0.70** (0.00)	**0.79** (0.03)
		R_{In}	**0.85** (0.02)	0.88 (0.03)	**0.90** (0.00)
		Δ_R	**0.03**	-	**0.02**
	BANK	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\mathrm{In}}}$	0.67 (0.01)	0.71 (0.03)	**0.75** (0.03)
		P_{In}	0.65 (0.02)	0.67 (0.02)	**0.67** (0.00)
		R_{In}	0.67 (0.01)	0.80 (0.02)	**0.85** (0.00)
		Δ_R	**-0.10**	-	**0.05**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.67** (0.01)	0.30 (0.00)	**0.69** (0.00)
		P_{In}	0.65 (0.02)	**0.79** (0.01)	0.65 (0.00)
		R_{In}	**0.67** (0.01)	0.25 (0.02)	**0.72** (0.02)
		Δ_R	**0.42**	-	**0.47**

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
LABELONLY	ADULT	$F_{1_{\mathrm{In}}}$	**0.73** (0.01)	0.81 (0.01)	**0.79** (0.00)
		P_{In}	0.81 (0.01)	**0.82** (0.00)	0.80 (0.00)
		R_{In}	**0.70** (0.02)	0.81 (0.01)	**0.81** (0.03)
		Δ_R	**-0.09**	-	**0.00**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.73** (0.01)	0.73 (0.02)	**0.79** (0.01)
		P_{In}	0.81 (0.01)	**0.80** (0.00)	**0.79** (0.03)
		R_{In}	**0.70** (0.02)	0.67 (0.03)	**0.80** (0.00)
		Δ_R	**0.03**	-	**0.13**
		Metric	**dt**	**rf**	**trepan-rf**
	SYNTH	$F_{1_{\mathrm{In}}}$	**0.85** (0.01)	0.98 (0.01)	**0.97** (0.00)
		P_{In}	0.86 (0.01)	**0.83** (0.00)	0.82 (0.00)
		R_{In}	**0.84** (0.02)	0.82 (0.01)	**0.93** (0.03)
		Δ_R	**0.02**	-	**0.11**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.85** (0.01)	0.86 (0.02)	**0.81** (0.01)
		P_{In}	0.86 (0.01)	**0.80** (0.00)	**0.81** (0.03)
		R_{In}	**0.84** (0.02)	0.90 (0.03)	**0.90** (0.00)
		Δ_R	**-0.06**	-	**0.00**
		Metric	**dt**	**rf**	**trepan-rf**
	BANK	$F_{1_{\mathrm{In}}}$	**0.79** (0.01)	0.78 (0.01)	**0.79** (0.00)
		P_{In}	0.80 (0.01)	**0.80** (0.00)	0.79 (0.00)
		R_{In}	**0.78** (0.02)	0.76 (0.01)	**0.80** (0.03)
		Δ_R	**0.02**	-	**0.04**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.78** (0.01)	0.78 (0.02)	**0.79** (0.01)
		P_{In}	0.80 (0.01)	**0.80** (0.00)	**0.80** (0.03)
		R_{In}	**0.78** (0.02)	0.77 (0.03)	**0.78** (0.00)
		Δ_R	**0.01**	-	**0.01**

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
ALOA	ADULT	$F_{1_{\mathrm{In}}}$	**0.78** (0.01)	0.81 (0.01)	**0.79** (0.00)
		P_{In}	0.81 (0.01)	**0.79** (0.00)	0.79 (0.00)
		R_{In}	**0.76** (0.02)	0.80 (0.01)	**0.81** (0.03)
		Δ_R	**-0.04**	-	**0.01**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.78** (0.01)	0.64 (0.02)	**0.78** (0.01)
		P_{In}	0.81 (0.01)	**0.81** (0.00)	**0.78** (0.03)
		R_{In}	**0.76** (0.02)	0.53 (0.03)	**0.79** (0.00)
		Δ_R	**0.23**	-	**0.26**
	SYNTH	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\mathrm{In}}}$	**0.85** (0.01)	0.72 (0.01)	**0.83** (0.00)
		P_{In}	0.86 (0.01)	**0.84** (0.00)	0.72 (0.00)
		R_{In}	**0.84** (0.02)	0.62 (0.01)	**0.80** (0.03)
		Δ_R	**0.22**	-	**0.20**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.85** (0.01)	0.85 (0.02)	**0.72** (0.01)
		P_{In}	0.86 (0.01)	**0.81** (0.00)	**0.75** (0.03)
		R_{In}	**0.84** (0.02)	0.90 (0.03)	**0.83** (0.00)
		Δ_R	**-0.06**	-	**-0.07**
	BANK	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\mathrm{In}}}$	**0.79** (0.01)	0.77 (0.01)	**0.77** (0.00)
		P_{In}	0.80 (0.01)	**0.65** (0.00)	0.79 (0.00)
		R_{In}	**0.78** (0.02)	0.78 (0.01)	**0.79** (0.03)
		Δ_R	**0.00**	-	**0.01**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\mathrm{In}}}$	**0.79** (0.01)	0.79 (0.02)	**0.79** (0.01)
		P_{In}	0.80 (0.01)	**0.80** (0.00)	**0.80** (0.03)
		R_{In}	**0.78** (0.02)	0.78 (0.03)	**0.80** (0.00)
		Δ_R	**0.00**	-	**0.02**

Table 12.2: Results of the application of MIA, LABELONLY and ALOA with the setting *noise* for global explainers. The values reported are the mean over a 3 fold cross validation, with the standard deviation between brackets.

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
Mia	Adult	$F_{1_{\text{In}}}$	**0.72** (0.01)	0.49 (0.01)	**0.78** (0.01)
		P_{In}	**0.78** (0.02)	0.77 (0.01)	**0.79** (0.00)
		R_{In}	**0.66** (0.01)	0.36 (0.01)	**0.77** (0.00)
		Δ_R	**0.30**	-	**0.41**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.72** (0.01)	0.43 (0.02)	**0.77** (0.01)
		P_{In}	**0.78** (0.02)	0.70 (0.01)	**0.78** (0.01)
		R_{In}	**0.66** (0.01)	0.32 (0.01)	**0.76** (0.00)
		Δ_R	**0.33**	-	**0.44**
	Synth	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	**0.80** (0.01)	0.70 (0.01)	**0.77** (0.00)
		P_{In}	0.71 (0.01)	**0.85** (0.00)	0.70 (0.00)
		R_{In}	**0.98** (0.04)	0.78 (0.01)	**0.84** (0.03)
		Δ_R	**0.20**	-	**0.06**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.80** (0.01)	0.45 (0.02)	**0.80** (0.04)
		P_{In}	0.71 (0.01)	**0.69** (0.00)	**0.70** (0.04)
		R_{In}	**0.98** (0.04)	0.33 (0.02)	**0.90** (0.02)
		Δ_R	**0.65**	-	**0.57**
	Bank	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	0.70 (0.02)	0.68 (0.03)	**0.73** (0.03)
		P_{In}	0.65 (0.04)	0.71 (0.02)	**0.64** (0.06)
		R_{In}	0.70 (0.10)	0.65 (0.02)	**0.85** (0.10)
		Δ_R	**0.05**	-	**0.20**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	0.70 (0.02)	0.27 (0.04)	**0.65** (0.02)
		P_{In}	0.65 (0.04)	**0.70** (0.10)	0.65 (0.03)
		R_{In}	0.70 (0.10)	0.23 (0.02)	**0.69** (0.10)
		Δ_R	**0.47**	-	**0.46**

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
LabelOnly	Adult	$F_{1_{\text{In}}}$	**0.30** (0.01)	0.46 (0.01)	**0.78** (0.01)
		P_{In}	**0.78** (0.02)	0.77 (0.01)	**0.79** (0.00)
		R_{In}	**0.55** (0.01)	0.35 (0.01)	**0.80** (0.00)
		Δ_R	**0.20**	-	**0.50**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.30** (0.01)	0.33 (0.02)	**0.67** (0.01)
		P_{In}	**0.78** (0.02)	0.77 (0.01)	**0.68** (0.01)
		R_{In}	**0.55** (0.01)	0.52 (0.01)	**0.66** (0.00)
		Δ_R	**0.03**	-	**0.14**
	Synth	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	**0.79** (0.01)	0.80 (0.01)	**0.80** (0.00)
		P_{In}	0.80 (0.01)	**0.78** (0.00)	0.79 (0.00)
		R_{In}	**0.78** (0.04)	0.76 (0.01)	**0.80** (0.03)
		Δ_R	**0.02**	-	**0.04**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.79** (0.01)	0.82 (0.02)	**0.73** (0.02)
		P_{In}	0.80 (0.01)	**0.80** (0.00)	**0.68** (0.01)
		R_{In}	**0.78** (0.04)	0.77 (0.02)	**0.77** (0.02)
		Δ_R	**0.01**	-	**0**
	Bank	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	0.72 (0.02)	0.70 (0.03)	**0.72** (0.03)
		P_{In}	0.79 (0.03)	0.65 (0.02)	**0.76** (0.06)
		R_{In}	0.76 (0.12)	0.73 (0.02)	**0.75** (0.10)
		Δ_R	**0.03**	-	**0.02**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	0.72 (0.02)	0.47 (0.04)	**0.57** (0.01)
		P_{In}	0.79 (0.03)	**0.65** (0.06)	0.66 (0.00)
		R_{In}	0.76 (0.12)	0.46 (0.00)	**0.61** (0.01)
		Δ_R	**0.30**	-	**0.20**

Attack	**Data**	**Metric**	**dt**	**rf**	**trepan-rf**
Aloa	Adult	$F_{1_{\text{In}}}$	**0.77** (0.01)	0.82 (0.01)	**0.77** (0.01)
		P_{In}	**0.82** (0.02)	0.78 (0.01)	**0.77** (0.00)
		R_{In}	**0.76** (0.01)	0.78 (0.01)	**0.82** (0.00)
		Δ_R	**0.02**	-	**0.04**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.77** (0.01)	0.63 (0.02)	**0.77** (0.01)
		P_{In}	**0.82** (0.02)	0.81 (0.01)	**0.77** (0.01)
		R_{In}	**0.76** (0.01)	0.52 (0.01)	**0.77** (0.02)
		Δ_R	**0.24**	-	**0.25**
	Synth	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	**0.86** (0.03)	0.71 (0.00)	**0.83** (0.00)
		P_{In}	0.84 (0.01)	**0.82** (0.00)	0.73 (0.00)
		R_{In}	**0.83** (0.03)	0.70 (0.00)	**0.80** (0.03)
		Δ_R	**0.13**	-	**0.10**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	**0.86** (0.03)	0.84 (0.01)	**0.72** (0.00)
		P_{In}	0.84 (0.01)	**0.80** (0.10)	**0.72** (0.02)
		R_{In}	**0.83** (0.03)	0.89 (0.02)	**0.86** (0.01)
		Δ_R	**0.06**	-	**0.03**
	Bank	**Metric**	**dt**	**rf**	**trepan-rf**
		$F_{1_{\text{In}}}$	0.76 (0.20)	0.85 (0.03)	**0.76** (0.01)
		P_{In}	0.77 (0.01)	0.64 (0.02)	**0.79** (0.01)
		R_{In}	0.75 (0.02)	0.78 (0.02)	**0.80** (0.00)
		Δ_R	**0.03**	-	**0.02**
		Metric	**dt**	**nn**	**trepan-nn**
		$F_{1_{\text{In}}}$	0.76 (0.20)	0.77 (0.00)	**0.79** (0.01)
		P_{In}	0.77 (0.01)	**0.10** (0.10)	0.78 (0.00)
		R_{In}	0.75 (0.02)	0.74 (0.11)	**0.78** (0.09)
		Δ_R	**0.01**	-	**0.04**

Table 12.3: Results of the application of Mia, LabelOnly and Aloa with the setting *random* for global explainers.

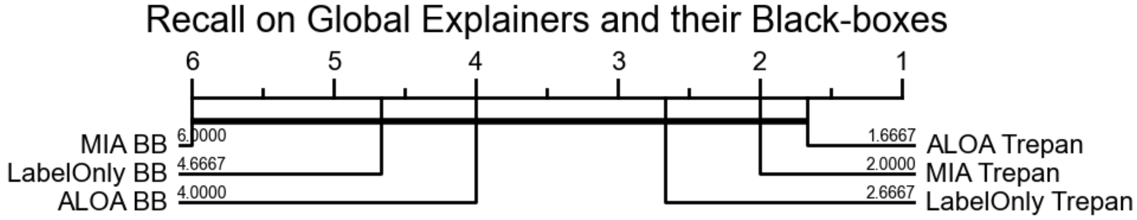


Figure 12.2: Critical difference plot for Nemenyi test with $\alpha = 0.05$ for the attacks performed on the *global* explainers and their black-boxes. The values reported results from the ranking procedure. They highlight that Aloa, Mia and LabelOnly against trepan have a small difference among them, while the attacks against the black-boxes are lower in the rank and clearly separated from the first three.

Results attacking the Local Explainers In this setting, we attack the local surrogate explainers. Differently from the global case, the local surrogate is a simple ML model which describes the behaviour of the black-box model close to the record under analysis and not the overall behaviour, as in the case of the global explainers. For this reason, we apply a different procedure presented in the methodology. In this setting, the procedure works as follows: firstly, in the *Attack training* procedure, we fit an attack for each surrogate model created ($E = \{e_1, e_2, ..., e_n\}$) together with the attack against the black-box model b, obtaining $A_E(\cdot)$, $A_b(\cdot)$. Then, in the *Attack application* procedure, we consider the resulting attack models as part of an ensemble classification method, having $A_E(\cdot)$ as an ensemble of different attacks. The last procedure, *Attack evaluation*, can be instantiated in different ways, depending on the attack considered. In 12.1.1, we presented the different instantiation of reveal in case the attack produces the prediction probabilities vector or not. In this experiment, we use the approach which exploits the probability vectors for Mia, while we exploit the other approach for LabelOnly and Aloa.

For the experiments conducted, for each dataset considered we select a set of records to explain from the test set exploiting a K-means clustering procedure, with k being the best value for the dataset under analysis. The choice of k is made by exploiting the elbow method. Due to computational limitations, we explain 3 records for each quantile of each cluster. Regarding the training of the local surrogate models obtained with Lore, the procedure requires synthetically generating a local neighbourhood around the record x under analysis and then fitting a local surrogate dt on the generated neighbours. Therefore, there are different parameters to set, in particular for this setting are important the kind of generation of the neighbourhood and the number of synthetic records to create. We conducted a search on these variables, obtaining similar results when considering the *genetic and random* generation, *genetic and probabilities* generation and the *counterfactual first search* generation. The other kinds of generations, such as the *random* one, show lower performance. Regarding the number of synthetic records to create, we use 10000. Similarly to the case of the global explainers, for each local surrogate model we train a Mia attack, with 6 shadow models, with accuracy above 80%. Also in this case, the models created for the attack are all rf. The same setting is applied to LabelOnly and Aloa.

The results of the attacks against the local explainers are reported in Table 12.2 and in 12.3, respectively for the *noise* dataset and the *random* dataset. In this setting, we observe a lower privacy exposure of the explainers w.r.t. the global setting. This result can be observed by analyzing the values of the ΔR: while in the global case, we mostly have positive values, highlighting an increase in privacy exposure when attacking the explainers instead of the black-boxes, in the local case, the values are closer to zero, with some negative values, implying that attacking the black-boxes produce a higher privacy exposure than attacking the local explainers. Regarding the *noise* case, Mia produces the highest privacy exposure, with positive Δ_R for all the configurations considered. However, the setting changes in the *random* case, having a lower privacy exposure for Mia and LabelOnly. Aloa, instead, gives similar results both for the *noise* and *random* case. This result can also be seen in Figure 12.3, in which is presented a Critical Difference plot for the Recall of the various attacks performed against the local explainers and their black-boxes. Also in this case, as in the global case, there is not a significant statistical difference among the attacks presented. However, in this plot, we can observe that Aloa and LabelOnly against the black-boxes are the highest in the rank, showing a higher privacy exposure w.r.t. Aloa against Lore and Mia against Lore, which is in the fourth position, equally matched and significantly separated from the first two. mia occupies the lowest positions of the ranking against the black-boxes and LabelOnly against the local models, with a clear distinction between them and the first fourth of the rank.

Attack	Data	Metric	rf	nn
Mia	Adult	$F1_{\text{In}}$	0.60 (0.00)	0.43 (0.02)
		P_{In}	0.54 (0.02)	0.30 (0.00)
		R_{In}	**0.68** (0.02)	**0.70** (0.02)
		ΔR_{In}	**0.01**	**0.17**
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.73 (0.03)	0.70 (0.02)
		P_{In}	0.71 (0.01)	0.66 (0.00)
		R_{In}	**0.84** (0.02)	**0.70** (0.00)
		ΔR_{In}	**0.02**	**0.18**
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.77 (0.01)	0.69 (0.01)
		P_{In}	0.64 (0.02)	0.68 (0.03)
		R_{In}	**0.83** (0.00)	**0.69** (0.00)
		ΔR_{In}	**0.03**	**0.44**

Attack	Data	Metric	rf	nn
LabelOnly	Adult	$F1_{\text{In}}$	0.77 (0.23)	0.28 (0.10)
		P_{In}	0.78 (0.78)	0.73 (0.12)
		R_{In}	**0.75** (0.21)	**0.30** (0.10)
		ΔR_{In}	**-0.06** (0.01)	**-0.37** (0.04)
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.72 (0.02)	0.75 (0.07)
		P_{In}	0.68 (0.03)	0.65 (0.08)
		R_{In}	**0.78** (0.01)	**0.82** (0.02)
		ΔR_{In}	**-0.04**	**0.08**
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.58 (0.02)	0.50 (0.00)
		P_{In}	0.66 (0.05)	0.64 (0.01)
		R_{In}	**0.52** (0.00)	**0.48** (0.05)
		ΔR_{In}	**-0.24**	**0.29**

Attack	Data	Metric	rf	nn
Aloa	Adult	$F1_{\text{In}}$	0.74 (0.00)	0.45 (0.02)
		P_{In}	0.79 (0.02)	0.51 (0.06)
		R_{In}	**0.72** (0.02)	**0.40** (0.03)
		ΔR_{In}	**-0.08** (0.01)	**-0.13** (0.04)
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.73 (0.00)	0.62 (0.00)
		P_{In}	0.70 (0.01)	0.60 (0.01)
		R_{In}	**0.76** (0.02)	**0.65** (0.00)
		ΔR_{In}	**-0.10** (0.01)	**-0.25** (0.04)
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.65 (0.01)	0.43 (0.01)
		P_{In}	0.58 (0.02)	0.47 (0.00)
		R_{In}	**0.71** (0.04)	**0.58** (0.09)
		ΔR_{In}	**-0.07** (0.01)	**-0.20** (0.04)

Table 12.4: Results of the application of Mia with the setting *noise* locally.

Attack	Data	Metric	rf	nn
Mia	Adult	$F1_{\text{In}}$	0.41 (0.00)	0.46 (0.03)
		P_{In}	0.30 (0.00)	0.70 (0.01)
		R_{In}	**0.64** (0.01)	**0.35** (0.04)
		ΔR_{In}	**0.28**	**0.03**
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.64 (0.03)	0.38 (0.00)
		P_{In}	0.60 (0.01)	0.35 (0.04)
		R_{In}	**0.71** (0.02)	**0.38** (0.01)
		ΔR_{In}	**-0.07**	**+0.05**
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.16 (0.05)	0.46 (0.03)
		P_{In}	0.27 (0.02)	0.34 (0.09)
		R_{In}	**0.12** (0.01)	**0.90** (0.01)
		ΔR_{In}	**-0.53**	**-0.13**

Attack	Data	Metric	rf	nn
LabelOnly	Adult	$F1_{\text{In}}$	0.67 (0.23)	0.34 (0.10)
		P_{In}	0.77 (0.25)	0.73 (0.12)
		R_{In}	**0.60** (0.21)	**0.20** (0.10)
		ΔR_{In}	**+0.25**	**-0.22**
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.71 (0.02)	0.63 (0.01)
		P_{In}	0.65 (0.04)	0.70 (0.01)
		R_{In}	**0.76** (0.02)	**0.60** (0.00)
		ΔR_{In}	**0.00**	**-0.17**
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.52 (0.03)	0.47 (0.00)
		P_{In}	0.65 (0.20)	0.60 (0.00)
		R_{In}	**0.50** (0.00)	**0.44** (0.01)
		ΔR_{In}	**-0.23**	**-0.02**

Attack	Data	Metric	rf	nn
Aloa	Adult	$F1_{\text{In}}$	0.69 (0.02)	0.42 (0.05)
		P_{In}	0.75 (0.02)	0.50 (0.01)
		R_{In}	**0.66** (0.01)	**0.38** (0.01)
		ΔR_{In}	**0.12**	**-0.02**
		Metric	**rf**	**nn**
	Synth	$F1_{\text{In}}$	0.70 (0.01)	0.59 (0.03)
		P_{In}	0.71 (0.04)	0.60 (0.03)
		R_{In}	**0.70** (0.00)	**0.60** (0.01)
		ΔR_{In}	**0.00**	**-0.10**
		Metric	**rf**	**nn**
	Bank	$F1_{\text{In}}$	0.68 (0.04)	0.45 (0.04)
		P_{In}	0.65 (0.00)	0.36 (0.00)
		R_{In}	**0.69** (0.02)	**0.60** (0.00)
		ΔR_{In}	**-0.09**	**-0.14**

Table 12.5: Results of the application of Mia with the setting *rand* locally.

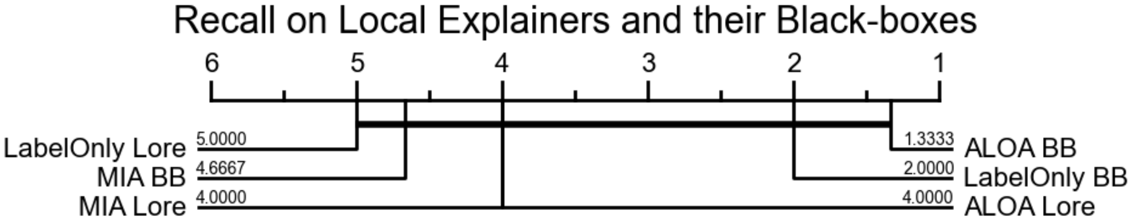


Figure 12.3: Critical difference plot for Nemenyi test with $\alpha = 0.05$ for the attacks performed on the *local* explainers and their black-boxes. The values reported results from the ranking procedure. Aloa and Mia against the black-boxes are the highest in the rank, posing a higher privacy threat. The ranking values show a clear separation between them and the other attacks.

Analysis on the number of records explained. For the *local* setting, the attack model A_E is an ensemble of multiple attacks, one against each of the local surrogate models created exploiting Lore. For this set of experiments, we create a local surrogate model for a set of records selected based on a K-means clustering procedure, explaining 3 records for each quantile of each cluster. Intuitively, it is the expectation that as the number of records explained increases, the privacy attack will yield better and better results. This is because explaining more records implies having more local surrogates, which thus better describes the data space under analysis. Consequently, attacking more local models that better describe the space under analysis should also improve the ensemble method of attacks. To validate this insight, we increase the number of records explained for each cluster. In particular, we consider Aloa with the *noise* dataset, which is the setting that shows a higher privacy exposure, and increase the number of elements for each of the datasets considered, ranging from 40 records up to 120 records. The results are reported in Figure 12.4. From this plot, it is possible to observe that, with a small number of records, the performance of the attack is low, highlighting that with few local explanations, the risk of privacy is low. This result aligns with our expectations, as limited local surrogate availability cannot represent all facets of the data space under analysis. However, the increase in the number of records explained also leads to an increase in privacy exposure, reaching a plateau starting from 80 records explained for all the datasets, i.e., starting from 80 records the increase in the number of records do not show an increase in privacy exposure. This behaviour in the analysis of the privacy risk is a finding already reported in literature [2] for the setting of Data Privacy.

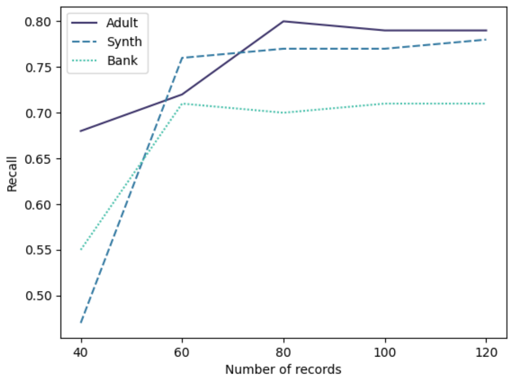


Figure 12.4: Results of Aloa increasing the number of records explained for the *local* setting. Starting from 80 records explained all the datasets reach a plateau in the privacy exposure.

12.3 Discussion

In this Chapter we introduce REVEAL, a framework designed to assess the privacy exposure of black-box models and their surrogate-based explainers, both local and global. This method is applicable to various types of black-box models and privacy attacks, making it a generic tool for privacy assessment.

Through our analysis, we uncover that attacking the privacy of explainers gives privacy exposures. The level of privacy risk varies depending on the specific privacy attack employed. Aloa stands out as a concerning privacy attack that requires minimal knowledge assumptions from the attacker, yet still achieves significant privacy breaches.

In our investigation, we observe that global explainers exhibit higher privacy exposure compared to their corresponding black-box models. This disparity is not observed in the case of local explainers, where the level of privacy exposure remains the same or even decreases compared to their black-box counterparts.

The results obtained from REVEAL highlight a worrying scenario in which user privacy is at risk, particularly when global explainers are used. These findings emphasize the delicate balance that must be maintained between explainability and privacy in the development of AI systems.

Understanding the trade-offs and vulnerabilities associated with explainability techniques is crucial for ensuring the privacy and security of sensitive user data. Further research and attention should be directed towards developing privacy-preserving methods for explainability, enabling the development of AI systems that provide both transparent decision-making processes and robust privacy protection.

Chapter 13

Conclusion

In this Ph.D. Thesis, we have addressed two main research questions. The first concerns the possibility of providing stable and actionable explanations that aid users in comprehending how Artificial Intelligence systems operate and the reasons that lead them to make particular decisions. The second research question revolves around the relationship between Privacy and Explainability. Both ethical values are essential in establishing Trustworthy Artificial Intelligence, but fulfilling one may potentially harm the achievement of the other and vice versa. As a result, we examined the relationship between these two ethical values, analyzing both their synergies and tensions.

The Thesis starts by presenting the relevant literature for this work, namely Machine Learning, Privacy, and Explainable AI. After laying the foundation for notation and methodologies, we presented the work conducted to answer the two research questions.

The first question originated from a benchmark analysis of the most popular post-hoc and local explanation methods for tabular data. In particular, we focused on: Lime, Lore, Shap, Dalex and Anchor. Some of them are feature importance methods, which output an importance value for each feature (Lime, Shap, Dalex), and the others are rule-based methods. Hence the output is in the form of logic rules, composed of a set of premises and a consequence. We analyzed these methods by exploiting the state-of-the-art metrics in this field, such as stability, fidelity and monotonicity. The results of this benchmark highlighted that the available methods in the literature still suffer from many limitations: most of the explainers are unstable. Hence if we feed the same record to the explainer more times, the output explanations will be different. This is a worrying problem since it makes the end users unsure about the reliability of the model itself. The situation does not improve when considering other metrics. In fact, none of the analyzed methods gave satisfactory results in terms of insertion and deletion metrics. This type of metric is essential in the context of feature importance explanations since it analyzes the goodness of the result obtained by directly testing the classifier. If the explanation states that a given feature is very important, this metric eliminates (or inserts) that feature to see an actual change toward the final prediction. Unfortunately, a negative result in this context further undermines the reliability of Machine Learning models and their explainers. Therefore, it is clear that the various methods proposed in the literature still suffer from several limitations. In addition, the feature importance-based methods propose explanations that are more difficult to understand for a non-expert user, while the rule-based ones are easier to understand thanks to the logical nature of the rules. In particular, the explainable methods available are unstable, slow and difficult to understand for a non-expert user. To address these issues, we have proposed LORE_{sa}, which is a black-box agnostic method for local explanations that provide informative, factual decision rules and actionable counterfactual rules. We first formalized the explanation algorithm and conducted an extensive experimental evaluation, comparing LORE_{sa} with the state-of-the-art methods. The results demonstrate that LORE_{sa} significantly improves the stability of explanations while ranking top or runner-up in several other quantitative metrics. We achieve stability of the explanations by utilizing a novel bagging-like approach in generating and aggregating several local decision trees.

There are a few potential directions for future work to expand the applicability of LORE_{sa}. Firstly, the synthetically generated instances may not respect the correlations among attributes, such as age and education level. Hence, it is worthwhile extending the approach by integrating domain knowledge, such as dependencies or causal relationships among attributes in the neighborhood generation and/or in the inference of the interpretable predictor. Finally, LORE_{sa} assumes that the black-box can be queried as many times as required. In cases where this is not possible, the neighborhood generation phase must consider constraints on the number of admissible queries, such as adopting an active learning variant of the genetic approach.

We then analyzed a possible synergy between Data Privacy and Explainable Artificial Intelligence: how to increase user self-awareness in the task of privacy risk assessment. In particular, we proposed EXPERT, a framework that addresses the computational complexity issue of the state-of-the-art methodology for privacy risk assessment, namely PRUDEnce, and enhances users' awareness by leveraging Machine Learning models to predict individual privacy risks. EXPERT also exploits local explainers to produce explanations of predicted risks. The proposed framework is modular, so that it can be tailored to specific data input and explanation requirements to achieve desired outcomes. This Thesis introduces two main variants of the EXPERT framework, specifically designed for analyzing human mobility data. This type of data is considered highly sensitive due to its particular structure. The first variant focuses on mobility profiles composed of features extracted from the trajectories and modeling the mobility behaviors of individuals; the second variant works directly with the raw trajectory data. For both variants, a comprehensive analysis is presented on the privacy risk prediction module, including an evaluation of the best classifiers, depending on the specific task at hand, as well as on the privacy risk explanation module. Regarding the setting based on mobility profiles, EXPERT provides good prediction performance as well as high fidelity explanations. However, the usage of features makes the comprehensibility of the explanation difficult, if not impossible. In fact, these explanations are tailored just for experts, which can understand the meaning of the features exploited during the training phase of the Machine Learning model and hence, (s)he can gain an overall understanding about the reasons that lead the classifier to a particular prediction. After obtaining promising results with the first variant of Expert that deals with mobility tabular data, we proceeded to analyze the second setting, which involves analyzing the privacy risk directly on the raw trajectories. To this end, we conducted an investigation to develop a high-performing classifier for sequential data that takes into account the time consumption of the training procedure to achieve real-time interaction. In addition to the privacy risk prediction module, we also analyzed the privacy risk explanation module, developing a visualization tool that enables users to visualize their mobility behavior on a map, with different visualizations for the locations that are more relevant for the prediction under analysis. We provided to the end-user both an aggregate visualization of the entire dataset and an individual visualization. The first one is tailored to analysts, having the objective to explore and analyze the overall mobility behavior of the dataset under analysis. The individual visualization, instead, is tailored to the end-user interested in observing her/his trajectories, with the most important locations for the risk prediction highlighted.The objective of Expert is to help users understand the privacy risks associated with their data and take necessary steps to protect their privacy. However, there are two areas for future improvement. Firstly, a major improvement is needed in the context of explanations. In fact, while in the context of tabular data there are many explainers available, such as Lime, Shap and Lore, in the context of raw trajectories, we only exploited Shap with a masking methodology due to the necessity of providing online explanations. However, other types of explainers are also available and could be explored further to achieve faster and more quantitative explanations. Secondly, regarding Expert for trajectories, the visualization aspect of the framework can be enhanced by developing a visual analytics environment that couples privacy risk prediction and visualization. This would allow analysts to modify data and reassess privacy risks in an interactive manner, and experiment with different protection measures before releasing the data. Additionally, integrating additional data quality measures could further aid in developing appropriate privacy protection measures.

Lastly, we analyzed the dangers of using Explainable Artificial Intelligence with respect to the Privacy, focusing on the privacy exposure of explanation methods. We introduced a novel privacy attack called Aloa, which is a variant of the LabelOnly approach. It can be used against black-box models and it is completely data agnostic, meaning it does not require any knowledge of the statistical distributions and domains of the features in the training data. Our experiments show that Aloa outperforms the traditional LabelOnly attack in terms of attack accuracy while assuming an adversary with weaker prior knowledge. This is a significant improvement in the context of privacy assessment, where any gain in performance can provide valuable, sensitive insights into the individuals represented in the data. In addition, the agnostic nature of our attack raises concerns from a privacy protection perspective, as it can be easily executed without any specific knowledge or assumptions. We also found that Aloa is stable in terms of prediction performance and performs better in attacking regularized models compared to other attacks. In conclusion, Aloa is a more robust and effective method for evaluating the privacy of ML models. Then, we presented reveal, a framework for assessing the privacy exposure of the black-box models and their surrogate-based explainers, being them local or global. The method proposed is generic and can be exploited for every kind of black-box model and of privacy attacks. The analysis conducted shows that attacking the privacy of the explainers, being local or global, gives rise to privacy exposures. Depending on the privacy attack considered, we have different levels of privacy risks, rising to particularly concerning situations with Aloa, a privacy attack that has a very little assumption of knowledge on the part of the attacker and yet still manages to achieve good results in terms of privacy breaches. However, global explainers show higher privacy exposure concerning their black-boxes, while this is not the case for local explainers, which show similar or lower levels of privacy exposure than their corresponding black-boxes.

The results obtained from the use of reveal raise concerns about the potential risks regarding the privacy of the users represented in the training data of machine learning models, mainly when global explainers are utilized. These findings emphasize the delicate balance that must be achieved between Explainable AI and Privacy in the development of Artificial Intelligence systems. As a future direction, we identify the need to develop mitigation strategies for explainers, which is a challenging task as it involves a trade-off among the accuracy of the explainer, the level of privacy protection, and the comprehensibility of the explanation. A possible approach could be to introduce generalization mechanisms for the explainers to find the optimal balance among these characteristics.

Part V

Appendix

Chapter 14

Data and Machine Learning Models

In this Ph.D. Thesis we focus our attention on Machine Learning models for tabular and sequence data. In this Chapter, we first present the terminology and the structure of the kinds of data in input. Following, we describe the state of the art of Machine Learning models for tabular and sequence data, with a focus on the methods employed in the experiments of this Thesis. This Chapter is organized as follows: in Section 14.1 we formalize the data considered, presenting the terminology related and the mathematical formulation of this kind of data. Then, in Section 14.2 we propose the state-of-the-art classifiers for tabular data (Section 14.2.1), and the ones for the sequence-based data (Section 14.2.2).

14.1 Data

In this Section we introduce the formal definition of the data exploited in this Thesis. In particular, we first present the formalization of the *tabular* data. This kind of data is exploited through the remaining of this Thesis. Following, we present the formalization of the *sequence* data. For this last kind of data, we focus on the formalization of sequential data for human mobility, namely trajectories, which are used in Chapter 9 in the context of privacy risk assessment, particularly relevant for this kind of data.

14.1.1 Tabular Data

A tabular dataset is a collection of data in the form of a table in which each column represents a variable and each row corresponds to a record of the dataset, e.g. a matrix. The values of the variables may be numeric, continuous or integers, depending on the variable under analysis, or categorical, as in the case of representation of information such as sex, colors, locations, or ordinal, in the case of categorical information in which there is a relative order, as an example, the size of clothes. Technically, we refer to a tabular dataset as $X = (N, M)$, in which N is the number of rows, or records, and M is the number of columns.

14.1.2 Sequence Data

Another kind of data exploited in this Thesis is the sequential one. In this case, instead of having a matrix representation of the dataset, each record is a list of events, with a relative order among them. A sequence $x = \{t_1, t_2, \dots, t_m\} \in \mathbb{R}^{m \times d}$ is an ordered set of m real-valued observations (or time steps), with dimensionality d. A sequence, also called time series, is *univariate* when $d = 1$, while, when $d > 1$ we name x a *multivariate time series.* This kind of data is extremely used in every day life: recording the values a sensor is registering is a sequential data, as well as the items brought at the supermarket and the web logs. Among them, for this Thesis we focused on a particular kind of sequential data: the trajectories from human mobility. In fact, human mobility data contain information about the movement of individuals during a given period of observation. They are typically collected by electronic devices, such as mobile phones and GPS devices installed in vehicles [218]. All the movements of a user in the period of observation are described using a sequence of spatio-temporal data points, i.e., a trajectory. In other words, each sequence item is a pair composed of a geographic location, often expressed in coordinates (generally latitude and longitude), and a timestamp indicating when the user stopped in or went through that location.

Trajectory A human mobility trajectory is a temporally ordered sequence of pairs, $T_u = (l_1, t_1), (l_2, t_2), \dots, (l_m, t_m)$, where $l_i = \langle x_i, y_i \rangle$ is the location identified by the latitude x_i and longitude y_i, while t_i $(i = 1, \dots, m)$ denotes the corresponding timestamp such that $\forall 1 \leq i \leq m \; t_i < t_{i+1}$.

We denote by $\mathcal{D} = T_1, \dots, T_n$ the *mobility dataset* that describes the complete history of movements of n individuals, in a specific period of observation.

14.2 Machine Learning Models

In this Ph.D. Thesis different classifiers for different tasks and kind of data are employed. Research in this branch of science has been receiving great interest for many years now, and because of this there are categorizations and different types of methods. In the following, we present the basic concepts of Machine Learning, and then focus more on the classifiers for specific data and tasks, starting from the one for tabular data, in Section 14.2.1, and then focusing on the classifiers for sequence data, in Section 14.1.2.

In the context of Machine Learning models (in the following referred to as ML for short), the first division is among *supervised* or *unsupervised* ML algorithms. Each of these approaches identifies a different objective in ML. In the rest of this Thesis, supervised models are employed.

In the case of *supervised* learning algorithms, we possess a ground truth knowledge of our data, that we can exploit during the training of the machine learning model. Technically, the dataset employed for the training of the model has a label associated to each record, denoting the target value of the record. In this way, the objective of the classifier is to learn to correctly label each record with the correct target value. To achieve this goal, the objective of a supervised learning model is to learn a function that approximates the relationships between the input records and their output values. In the setting of supervised learning algorithms, we can have two main kinds of tasks: *classification* and *regression* methods. In the case of *classification* algorithms, the goal is to predict a discrete or categorical class label. Therefore, the ML model has to predict the class the observation belongs. Being them supervised algorithms, they employ already known records with associated output class to learn the relations between data and classes. There are several learning algorithms that belong to this category, such as Decision Trees, Logistic Regression, Support Vector Machine, and the majority of the ensemble methods. In the case of *regression* models, instead, the goal is to predict a continuous quantity. The machine learning model estimates the most likely output value, learning the mapping function between the input variables and the continuous output value. Belonging to this category we can find several ML algorithms, such as Support Vector Regression, Linear Regression and several variants of Regression trees.

The other kind of algorithms in ml are the *unsupervised* learning algorithms. Differently from supervised learning algorithms, in this setting there is no basic knowledge about the expected outputs. Therefore, given the input, the output is unknown. In this case, the goal is to infer relationships among data and their structures. There are two main sub-categories of unsupervised learning algorithms: *clustering* algorithms and *associations* methods. In the case of clustering algorithms, the goal is to find the division of the dataset into classes. In order to do so, some of the algorithms might require additional knowledge, such as k-means, that requires as input parameter the number of clusters we are looking for. In this category, there are several algorithms, such as k-means, db scan, hierarchical clustering and so on. For the case of associations algorithms, they extract association rules from the dataset. These rules exploit patterns in the data, such as Apriori algorithm.

In formulas, a classifier is a function $b : \mathcal{X}^{(m)} \rightarrow \mathcal{Y}$ which maps data instances (tuples) x from a feature space $\mathcal{X}^{(m)}$ with m input features to a decision y in a target space $\mathcal{Y}$ of size $L = |\mathcal{Y}|$, i.e., y can assume one of the L different labels ($L = 2$ is binary classification, $L > 2$ is multi-class classification). We write $b(x) = y$ to denote the decision y taken by b, and $b(X) = Y$ as a shorthand for $\{b(x) \mid x \in X\} = Y$. If b is a probabilistic classifier, we denote with $b_p(x)$ the vector of probabilities for the different labels. Hence, we have that

$b(x) = y$ is the label with the largest probability among the L values in $b_p(x)$. An instance x consists of a set of m attribute-value pairs (a_i, v_i), where a_i is a feature (or attribute) and v_i is a value from the domain of a_i.

14.2.1 Classifiers for Tabular Data

In this Ph.D. Thesis we focus on *supervised classifiers*, i.e., methods in which the correct classification is exploited during the learning procedure, for tabular data. Technically, given a tabular data in the form $X = (N, M)$ we assume to have $L = (N, 1)$ which is a list of target classes, one for each record in the dataset X. In this way, the objective of the classifier is to learn to correctly classify the target value associated to the record under analysis. Classification of tabular data has been a fundamental problem in ML for decades, with numerous applications in diverse domains such as finance, healthcare, and marketing. Classifiers play a pivotal role in identifying patterns and making predictions from tabular data. However, there is no one-size-fits-all solution to this problem, and choosing the right classifier for a particular task can be challenging. In recent years, ML has emerged as a powerful tool for developing and evaluating classifiers for tabular data. In the following, we present some of the most popular classifiers for tabular data, employed also in the remaining of this Thesis.

Intrinsic Explainable ML methods

While many sophisticated ML algorithms have been developed, simple models, such as decision trees and logistic regression, remain widely used and relevant.

Despite their simplicity, this kind of models have been shown to achieve high accuracy on many benchmark datasets. They are often used as a baseline model for more complex algorithms, and can provide valuable insights into the underlying relationships between the features and the target variable. In fact these models are also called *intrinsic explainable* due to their simplicity it is easier to follow the decision making inside the model.

In this Chapter we will explore the strengths and weaknesses of logistic regression, in Section 14.2.1 as well as decision trees in Section 14.2.1.

Logistic Regression model The logistic regression (LG for short in the following) is a probability model, built on labelled samples. It is based on a mathematical function, employed the first time during the 19^{th} century in statistics for the description of growths in population. Nowadays it is widely used in statistic as well as ML domains. It is suitable for predictions in which the input is composed by numerical values, being them discrete or continuous. The output of this method is composed by discrete or categorical classes. In particular, depending on the kind of output considered, there are three types of LG models:

- *binary logistic regression*, in which the possible outcomes are only two: 0 or 1, that can represent categorical values, such as "risk" and "not risk", or "fraud" and "not fraud".
- *multinomial logistic regression*, in which the possible outcomes are a number of categories (at least three). These categories are without ordering, such as the classes of food (Vegan, Vegetarian, Not Vegetarian).
- *ordinal logistic regression*, in which the possible outcomes are a number of categories with ordering among them. An example of this setting is when the outcome is a rating over discrete numbers, such as from 0 to 5.

Technically, the LG method is composed by two main functions: a linear regression function and a sigmoid function (also called logistic function). The input data are fed into a linear regression function, Section 14.1, in which $h(x_i)$ represents the predicted value for the input x_i, while β is the vector of the regression coefficients. The output of this linear function is then fed into a sigmoid function, Eq. (14.2), in order to map this continuous value in a discrete value, that can be 0 or 1. In fact, as we can see from the plot of the logistic function, depicted in Figure 14.1, the function squashes the values between 0 and 1. However, the function is continuous between 0 and 1, allowing values that are between these two extremes. In order to classify the output as class 0 or class 1, the LG method requires a threshold value, also called boundary value. This value, that is usually set to 0.5, provides the model a decision boundary for considering the output belonging to the class 0 or 1.

$$h(x_i) = \beta^T x_i \tag{14.1}$$

$$g(z) = \frac{1}{1 + e^{-z}} \tag{14.2}$$

Therefore, the formula that is employed for the prediction with the LG model is reported in Eq. (14.3). It is composed by a linear regression function followed by a sigmoid function. It models the probability that the input record belongs to the default class. In this way, the output value is always bounded between 0 and 1. In particular, it tends to 0 if the value of z tends to negative infinity, while it tends to 1 if the value of z tends to positive infinity.

$$h(x_i) = g(\beta^T x_i) = \frac{1}{1 + e^{-\beta^T x_i}} \tag{14.3}$$

The LG model aims at predicting an output value such that the error difference between the predicted value and the true value is minimum. Therefore, for the optimization of the parameters, a gradient descent algorithm is employed to minimize the value of a cost function. In doing so, we minimize the error difference between the predicted and the actual

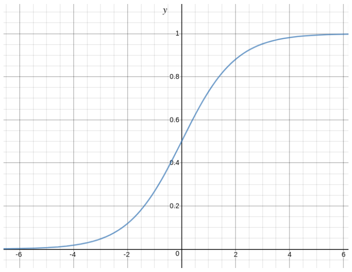


Figure 14.1: The plot of a sigmoid function (also called logistic function). The output value of the function $g(x)$ is always bounded between 0 and 1. When the value of z goes to positive infinity, $g(z)$ tends to 1, while when the value of z goes to negative infinity, then $g(z)$ tends to 0.

value. In this setting, the cost function employed is the log likelihood of parameters.
In order to derive the formula for the log likelihood of parameters, we consider the the formula in Eq. (14.3) for the computation of the predicted value. From the output of this previous formula, we can now compute the conditional probability associated to each record, defined in Eq. (14.4). Here, $(h(x_i))^{y_i}$ represents the value computed when the predicted output is equal to 1, while $(1 - h(x_i))^{1-y_i}$ is the one computed when the predicted value is equal to 0.

$$P(y_i|x_i;\beta) = (h(x_i))^{y_i}(1 - h(x_i))^{1-y_i} \tag{14.4}$$

From the conditional probability computed in Eq. (14.4), we can derive the likelihood of parameters, defined as the plausibility of a value for the input record under analysis, given the input record x_i and a specific parameter value for β, as represented in the formula in Eq. (14.5). Then, the log likelihood of parameters is employed as cost function for the LG method, using the formula expressed in Eq. (14.6). In this way, the cost function for the logistic regression method is proportional to the inverse of likelihood of parameters.

$$l(\beta) = \prod_{i=1}^{n}((h(x_i))^{y_i}(1 - h(x_i))^{1-y_i}) \tag{14.5}$$

$$J(\beta) = \sum_{i=1}^{n}(-y_i \log(h(x_i)) - (1 - y_i)\log(1 - h(x_i))) \tag{14.6}$$

Decision Tree A decision tree (DT for short in the following) is a decision-support tool, that employs a tree-like model of decisions and their possible consequences. Based on this concept, in machine learning, DT are a predictive model that learns a tree-based structure that goes from the observations available in the training set, to conclusions about the targets (in the leaves of the tree). The goal of the learning process is to create a model that predicts the value of a target variable based on the inputs variables. If the targets are discrete or categorical values, then the model is called *classification tree*, while in the case of continuous target values, the model is called *regression trees.*

Given a training set, in which each record is composed by several features with finite discrete domains, a DT has the internal nodes that represent decision about a specific feature. That is, each internal node is labelled with an input feature and a threshold value (or more than one, if the target is multi-labelled). Depending on the thresholds, the outgoing arcs available lead to different decision nodes. One important characteristic of the DT is that they list all the possible combinations of input features.

In general, given a a training dataset, there are many DT which fit the data. For this reason, one of the main concerns in this setting, is the selection of the best model of DT. However, the size of the search space for the optimal DT is exponential in the number of input features. Therefore, it is unfeasible to evaluate all the models in order to select the best one. In order to overcome this limitation, several efficient algorithms have been proposed. They construct sub-optimal DT with a good accuracy and in a reasonable amount of time. These algorithms are usually greedy search, that are based on locally optimum decisions for the splitting phase. In this setting there are several algorithms available, such as Hunt's algorithms, CART, ID3, SLIQ and so on. The majority of the algorithms available are a top-down approach that recursively applies a test and split procedure to the remaining part of the training dataset. For example, Hunt's algorithm, that is one of the earliest proposal, at each step takes as input the training samples that reach the decision node under analysis, D_t and tests if they belong to the same target class or to different ones. If the target class is the same, it means that the branch under analysis is concluded. Otherwise, the data in D_t are split using an input feature.

The decision of the best splitting feature among all the possible input features determines the diversity among the different algorithms and strategies available. When a *greedy approach* is employed, the choice is on the feature for which the split produces nodes with purer classes [219]. In practice, it selects the split that minimizes the impurity of the resulting smaller training set. In order to evaluate the impurity of the split, there are several measures available, such as Gini index, that is computed using the formula $Gini(t) = 1 - \sum_j [p(j|t)]^2$, in which t represents the decision node under analysis and $p(j|t)$ is the relative frequency of the class j for the node t. The goal is to find the split that has the closest value to zero. Another measure that is often employed is the entropy, defined as $Entropy(t) = -\sum_j p(j|t \log p(j|t)$. In this setting, an entropy value closes to zero means that the split produces two datasets that are pure, while a value closes to the logarithm implies that the records are equally distributed among the classes, that we want to avoid. Lastly, the misclassification error can be evaluated as well: $Error(t) = 1 - \max_i P(i|t)$, in which $P(i|t)$ represents the number of records that belongs to each class. Among these numbers, the maximum one is selected. Also in this case, an output value closes to zero represents the purer split. Once one of these measures is selected, the available splits for the node under analysis are computed and for each one the degree of impurity is evaluated. The one with lowest degree of impurity is then selected.

The DT have several advantages: first of all, they are fast and easy to implement. Moreover, once the model has been created, the prediction task corresponds to follow a path in the DT. Therefore, it is fast and it is easy to interpret, especially when the training set has a small number of features. Another great advantage is that it is robust to noises and it is able to handle irrelevant or redundant attributes.
However, there are also a number of disadvantages in using DT. First of all, the space of possible DT is exponentially large and depends on the number of input features. Therefore, the greedy strategies are not suitable for finding the best DT. Moreover, the relationships among features are not taken into consideration. In fact, each decision boundary involves only an attribute each time. Another important limitation of DT is the overfitting: if the DT goes too deep, it learns a model that fits really well only the data in the training set.

Ensemble methods based on trees

Ensemble methods have become increasingly popular in recent years as a powerful approach for improving the accuracy and robustness of ML models. Ensemble methods combine multiple models to create a single, more accurate model, by aggregating the predictions of individual models. Ensemble methods have been shown to achieve state-of-the-art performance on many benchmark datasets, and are widely used in industry and academia. They are particularly useful for handling complex datasets with high-dimensional features and nonlinear relationships between the features and the target variable.

In this Section, we will discuss some of the recent advancements and challenges in these models, and explore avenues for future research. We start with the presentation of Random Forest then, we move to advanced methods, such as XGBoost and CatBoost to conclude the presentation of DeepForest , a novel approach which merges the ensemble method theory with the structure of the neural networks

Random Forest Random Forest (RF for short in the following) were proposed for the first time in 1995, by Tin Kam Ho [220]. RF is a method part of the *ensemble methods*. The general idea of ensemble learning algorithms is that they construct a set of classifiers from the training data and the final prediction is carried on by combining the predictions made by multiple classifiers. The main idea behind this ensemble method is the so called *wisdom of crowds*: a large number of relatively uncorrelated small and simple classifiers, operating as a committee, will outperform any of the individual constituent models. Intuitively, the crowd of classifiers protect each other from the errors, providing that they are not all mak-

ing the same mistakes. In fact, even if some of the classifiers are wrong, other are right, so as a group the classifiers still make the correct prediction. RF are hence part of the ensemble learning methods, composed by a set of decision trees, presented above in Section 14.2.1. Each DT is built considering a random subset of the training data. Therefore, the resulting forest of trees is composed by trees that are built using different training samples. This increases the diversity in the forest. In fact, different sub-spaces generalize their classification in complementary ways, obtaining a combined classification improvement. In this way, RF is more robust and also less prone to overfitting, due to their structure and creation. During the prediction step, the record in input is fed to every DT in the forest. At this point, if the task is a classification one, the final output is computed with a majority vote. In the case of regression tasks, instead, the final result is the average of all the individual DT estimates.
The main advantage in using RF is that they achieve high accuracy and less variance, meaning that the predictions are correct for a large range of different records. Moreover, they overcome the main limitations of the decision trees, by avoiding overfitting and increasing the robustness. In terms of complexity, RF is a fast method, both from the point of view of the creation of the model as well as in the prediction step. In fact, even if the complexity of the creation of the model is higher w.r.t. a single DT, it can be easily parallelized. However, the simplicity of the decision trees allows for simple and straightforward interpretation of the result, but in the case of random forests the interpretability task is more complicated. Another important limitation is in the number of decision trees. In fact, in random forests the number of decision trees is a user-defined parameter and it is often difficult to find the best trade-off among the number of trees, the overfitting and the interpretability of the model.

In the original formulation, the author addresses the problem of avoiding overfitting by exploiting the random subspace method in the feature space. Later, in 2006, Leo Breiman and Adele Cutler proposed the introduction of the *bagging* method, e.g. the selection with replacement of a random sub-set of the training dataset for each DT. This change was proposed to control the variance of a collection of DT. In the rest of this Thesis work we refer to the RF as implemented in the Scikit-Learn library of Python[1] which contains the last version of the RF with the bagging method.

XGBClassifier XGBClassifier (eXtreme Gradient Boosting), also called XGB, is an ensemble method for classification and regression tasks, based on gradient boosting. Gradient boosting is a ML technique in which the model is an ensemble additive model made of *weak* learners. Technically, given a differentiable loss function to optimize, the process starts fitting one single weak learner, usually a DT very small, called *stump*, with depth and number

[1] Implementation of RF used in the experiments of this Thesis Scikit-Learn library

of nodes constrained. Then, the first DT is evaluated, given the loss function. Based on the loss of this first DT, another one is fitted to improve the predictions, focusing on the errors of the previous one, by adjusting the weights of the input records. Then, the process continues up until one of the ending condition is met, such as maximum number of weak learners or reaching of the wanted result in the loss function. This procedure, also called additive procedure, is employed by different kinds of ML models, such as AdaBoost and XGB. XGB is a variant of the Gradient Boosting, proposed by Tianqi Chen [221] in which decision trees are created sequentially. The key of the success of XGB resides the use of weights: each feature has a weight assigned when fed into the DT to predict the label. The weight of variables predicted wrong by the tree is increased and these variables are then fed to the second DT. In this way, several DT are trained and then use to make better predictions w.r.t. just a single tree. XGB is one of the best classifiers for tabular data up to now, both in terms of prediction performance and of computational time. In fact, the library available for Python is parallel, allowing a fast train even if the ensemble models are huge. In addition, XGB works well with big amount of data, even if there are missing values.

CatBoost CatBoost is a ML algorithm developed by Yandex, a Russian technology company [222]. It is based on the gradient boosting algorithm, which is a technique for building ensemble models merging several weak classifiers. It is especially designed to work with tabular data and it is used for a variety of tasks such as classification, regression, and ranking.

One of the key features of CatBoost is the ability to handle categorical variables with high cardinality, e.g. categorical variables with a large number of unique values. An example of this kind of categorical data might be a variable about the education: depending on the level (high school, bachelor degree, master degree etc.) and the field of education (economics, mathematics, philosophy) there may be a great number of different categorical values the variable can assume. To achieve great performance with categorical data, CatBoost exploits a unique algorithm, called *ordered boosting*, which sorts the categorical variables based on their impact on the target variable and then combines them with numerical features exploiting gradient boosting. CatBoost also has built-in features for handling missing values, which uses a combination of gradient boosting and random forests to build the final model. In this way, the model improves its accuracy and generalization. Another advantage of CatBoost is its ability to handle imbalance in the target variable, e.g. if one class has significantly more samples than the others. It does this by adjusting the weights of the samples during the training process. In addition, CatBoost includes several regularization techniques to prevent overfitting, such as L2 regularization, feature selection, and early stopping. CatBoost also supports GPU acceleration, which can significantly speed up the training process for large datasets.

Deep Forest Deep Forest, also called GcForest, is an ensemble method for classification tasks, tailored for tabular data, as well as images and text [196]. The GcForest has a cascade structure, composed of layers, as in the case of Neural Networks (nn). Each layer is made of ensemble methods, such as rf, XGBClassifier or ExtraTrees. The model works as follows: each record in input is fed to the first layer of the GcForest structure, which outputs a prediction probability vector for each of the sub-model of the layer. These vectors are concatenated to the record in input and fed to the next layer. Hence, going deep in the structure of the model, the layers have more information on which to make a prediction: not just the input record, but also the prediction probability vectors of the layer above. The structure of the GcForest is also depicted in Figure 14.2. This structure of the model was inspired by the structure of the nn. In fact, in their paper, the authors aim at proposing a novel ML model that takes the positive parts of the nn and overcome some of their disadvantages. In particular, from the experiments presented in their work, the author showed that the GcForest models are faster to train w.r.t. the nn, since they do not require hyper-parameter settings and their structure is apt to parallel implementation, making the computational time even less. In addition, nn with tabular data do not achieve the best performance, while the GcForest being composed of ensemble models, shows better performance.

ML methods for tabular classification

Artificial neural networks Neural networks are computing systems that are inspired by the biological neural networks that constitute the brains of the animals. They were first theorized in 1943 by McCulloch and Pitts in [223]. They are composed by a collection of connected units, called *neurons*. The connections among these neurons are capable of transmitting signals from a neuron to another one. The resulting structure aims at learning to approximate some function f^*, by analyzing the examples in input. One of the simplest scenarios that is often employed as an example of neural networks is a classification task, in which the neural network has to learn the function f^* given the input x and the expected output y: $y = f^*(x)$. There are a great number of tasks the neural networks are able to solve, both in a supervised and in an unsupervised learning manner. Depending on the type of task required, the kinds of artificial neural networks employed may vary. In the following, we are briefly presenting the basic structure of artificial neural networks for introducing the Long-Short Term Memory neural network, that is the model we employed for this project.

Neurons are the basic component of artificial neural networks. A neuron k is composed by a neuron state, also called activation, $a_k(t)$, that depends on the time t. The state of the neuron is updated iteration after iteration evaluating the *activation function.* Given the input value and the neuron state at the previous time t, it computes the new neuron's state at time $t+1$: $a_k(t+1) = f(a_k(t), v_k(t), \theta_k)$. Therefore, the activation function depends

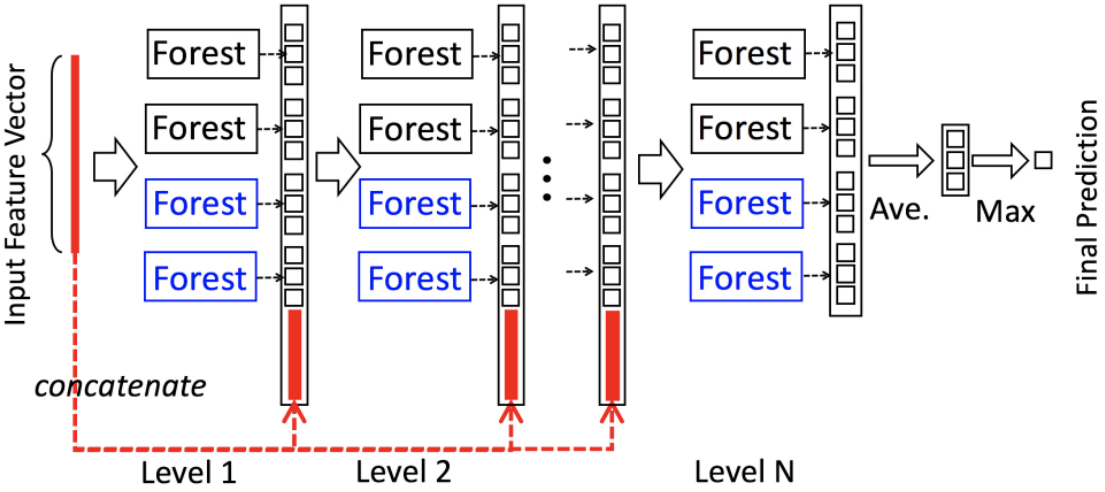


Figure 14.2: Structure of the GcForest model. Given a record in the form of a feature vector, it is fed into this cascade model, in which each layer is making predictions based on the sub-models (in the pictures there are forests). At the end of each layer the feature vector in input is correlated with the prediction probability vectors of the sub-models. The final prediction is given by the aggregation of the ones in the last layer and among them, the maximum one.

on the value in input from the previous neurons $v_k(t)$, the actual value of the neuron state $a_k(t)$ and a threshold θ_k that can be set to clearly discern among the possible outputs of the model. The result of the activation function is then fed into the output function, in order to obtain the actual output to deliver to other neurons in the successive timestamp. The output function is defined as $o_k(t) = f_{out} = (a_k(t))$. However, the output function could simply be the identity function. The other fundamental components of the neural networks are the *connections*, i.e. the edges between neurons. They represent the actual network structure of an artificial neural network. They transmit messages only in the direction from the starting neuron m to the ending neuron k. Each connection between two neurons has a weight assigned w_{km}. These weights are adjusted during the computation. Moreover, there could be a bias term, added at the total weighted sum of inputs. The bias factor works as a threshold for the activation function.
Therefore, the basic structure of a neural network is composed by layers of neurons, in which each neuron in a layer is connected with other neurons in the successive layer. Hence, each neuron has some input connections, from which it receives the input values on which it learns, and some output connections, on which it sends the output of its internal computation. Technically, there are several input connections for each neuron, therefore there is the need of a function, called *propagation function*, that computes the input value for the neuron k given the output values of the m predecessor neurons. In this way, we obtain the value of $v_k(t)$ previously defined as the value calculated from the previous networks and the actual input for the activation function. The mathematical formulation of the propagation function is $v_k(t) = \sum_m (x_m(t)w_{km} + b_k)$, where w_{km} is the weight of the connection under analysis, while b_k is the bias term. In Figure 14.3 a simple neuron is depicted. Only two layers differ from the model described above: the input layer, in which the neurons don't have any input connections, but only output connections; and the last layer, in which there are not output connections.

The main goal of neural networks is to solve a specific task learning a class of functions F from a set of observations. In particular, the main task is to find $f^* \in F$ that solves the task in an optimal sense. Therefore, the learning process employs a *cost function*, also called *loss function*, C, $C : F \rightarrow \mathbb{R}$, such that the optimal solution has the minimum cost: $C(f^*) \leq C(f)\ \forall f \in F$. In this way, the network has a way to evaluate how far away a solution is from the optimal solution. For the case in which the solution is data dependent, the cost function depends on the observation. There are several kinds of loss function that can be employed. Depending on the kind of task to solve, such as regression or classification, the loss function varies. For regression tasks, usually the squared error can be used, while for classification tasks, the categorical crossentropy, as well as the binary crossentropy, are employed. The learning process of neural networks involves the update of the weights. One of the most used methods is called *back-propagation* algorithm, proposed the first time in [224]. It is a method for the update of the weights for supervised learning algorithms,

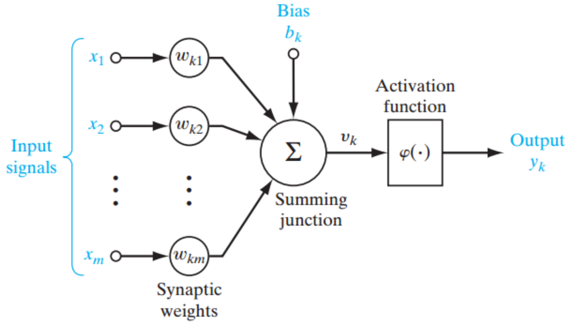


Figure 14.3: The simplest possible neural network is composed by a single neuron: the perceptron. The perceptron depicted here receives several inputs x_1, x_2, ..., x_m that are multiplied by their weights w_{k1}, w_{k2},...,w_{km}. The summing junction, also called propagation function, sums all these inputs and weights with a bias factor b_k. The result of the propagation function v_k is then fed into the activation function, which then outputs the actual output of the perceptron.

that follows a gradient descent approach, exploiting the chain rule. Technically, at the first iteration the weights are set randomly. Then, the training step computes an output value, based on the initial random weights and on the input values. Employing the loss function, the network defines the difference between the expected output and the actual one. At this point, the value of the loss function is employed to update the weights associated to the connections. With the backpropagation algorithm, the weights at the timestamp $t + 1$ are computed based on the weights at the previous timestamp t, the learning rate η and the cost function c: $w_{ij}(t + 1) = w_{ij}(t) - \eta \frac{\partial C}{\partial w_{ij}} + \epsilon(t)$.
One of the simplest kind of neural network is the *feedforward* neural network, in which the information flows from the input layer through the intermediate computations up to the output layer. The structure of a feedforward neural network is depicted in 14.4. Frequently, instead, there are also *feedback* connections, in which the outputs of the model are fed back into itself. The neural networks that have also feedback connections are called *recurrent neural networks* (RNN). A particular kind of RNN is the Long-Short Term Memory.

14.2.2 Classifiers for Sequence Data

A ML model able to classify time series is a model that takes in input a time series and outputs a label indicating the class to which the time series belongs. Time series classification (TSC) is a challenging task, as time series can have variable lengths and shapes, and can contain noise and uncertainties.

There are several types of machine learning models that can be used for TSC, including

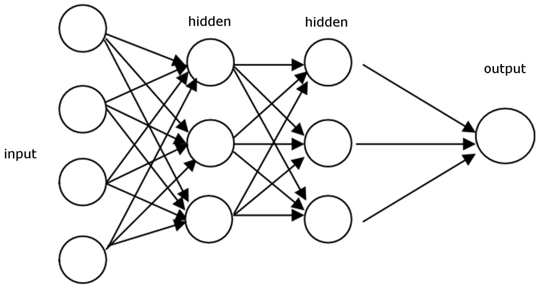


Figure 14.4: The basic structure of a feedforward neural network. The first layer is the input layer, in which there are no input connections. Then, there could be a number of hidden layer. The last layer is the output layer, that outputs the final value computed by the network.

traditional statistical methods, such as k-nearest neighbors (KNN), support vector machines (SVM), and decision trees, as well as deep learning methods, such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformers.

Traditional statistical methods for TSC typically extract hand-crafted features from the time series, such as statistical moments, spectral coefficients, and autocorrelation functions, and use these features as input to a classifier. Deep learning methods, on the other hand, learn features directly from the time series using deep neural networks, which are composed of several layers of processing units that learn progressively more complex representations of the input data.

Among the deep learning models used for TSC, CNNs have shown to be particularly effective, as they can extract relevant features from time series by applying convolutions over different temporal windows. InceptionTime is an example of a CNN-based model that has achieved state-of-the-art performance on several TSC benchmarks.

Overall, the choice of the most appropriate model for TSC depends on several factors, such as the complexity and variability of the time series, the size of the dataset, and the available computational resources.

Recurrent Neural Networks and lstm

Recurrent Neural Networks (RNNs) were first introduced by David Rumelhart's in 1986 in [225]. RNNs are the state of the art algorithms for the analysis of sequential data. They are a kind of neural networks able to handle time series data, due to its ability to exhibit a temporal dynamic behaviour. In practice, they are able to remember their inputs, due to their internal structure that works as a memory. Therefore, the final result depends on the

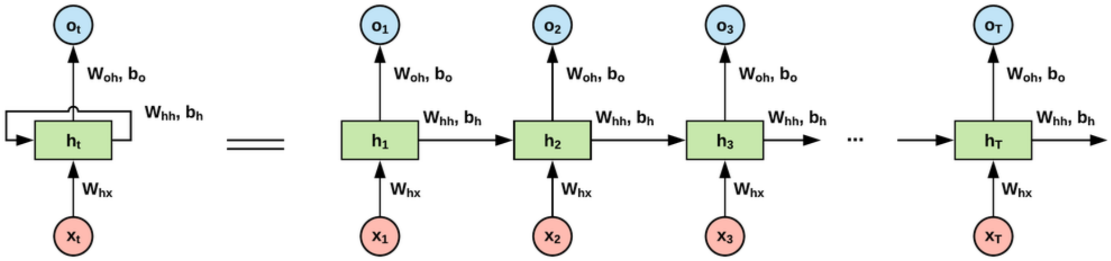


Figure 14.5: A simple RNN cell. On the left hand side the state of cell is unrolled in order to show the behaviour during consequent timestamps.

historical context of the input and not only on the last value seen by the model. Technically, in RNNs the information cycles through a loop: at each time step, in order to output a result, the network analyses the new input that is fed into the network as well as what it has learned previously. Therefore, at each time step, the RNN model has two inputs: the new example and "its memory value". This neural network basic neuron is depicted in Figure 14.5, in which there is a "vanilla" RNN. On the right hand side we can see the unrolled network, based on the different time steps. The novelty of this kind of neural network is the *Backpropagation Through Time* (BPTT) concept. It is a gradient-based technique for training recurrent neural networks. It is backpropagation algorithm especially tailored for sequence data. Hence it is a gradient-based technique employed in the learning process. Technically, at each time step it unfolds the network and then the backpropagation algorithm is applied. In this way, the gradient method is computed in order to compute the values for the weights of the network. The structure described so far produces an efficient neural network, able to memorize information during its training. In particular, when applied for small problems, these neural networks are able to learn long-term dependencies with good accuracy. However, when applied on real problems, RNN suffers from two main limitations: *exponential gradient* and *vanishing gradient*. Both these problems arise during the training of the neural network with BPTT, when the gradients are being propagated back up to the initial layer. The main problem is that the values of the gradients from the deepest layers undergo a number of matrix multiplication (due to the chain rule of the backpropagation). For this reason, if the initial value was small, it will continue shrinking until it vanishes (hence having a vanishing gradient problem) or otherwise if the value is large, it will keep on getting larger, up until it explodes (having an exploding gradient problem). Both for the case of vanish gradient and of exploding gradient, the main problem is that these values make the learning algorithm stops the learning process (or it slows it down considerably).

For the case of exploding gradient, there are some possible alternatives, such as the trunca-

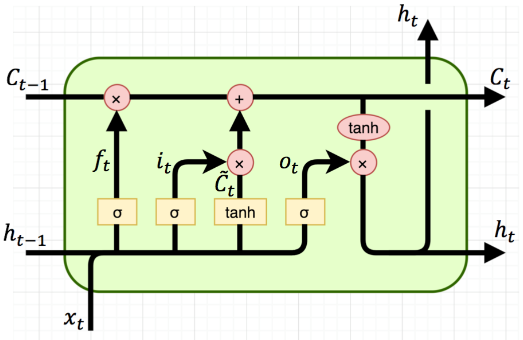


Figure 14.6: A cell of a LSTM neural network at time t. The new input is x_t, while $h_t - 1$ is the output value of the previous iteration. LSTM cells are composed by three different gates, respectively a forget gate (f_t), an input gate (i_t) that together determine the new cell state value, and an output gate (o_t).

tion of the gradient. However, for the vanishing gradient problem, there is not a straightforward solution. In order to solve this issue, Long Short Term Memory (LSTM) neural networks were proposed in 1997 by Hochreiter and Schmidhuber in [209].

LSTM networks are a special kind of recurrent neural network in which the memory is "extended". They resemble the mnemonic approach of a person in the sense that they have a memory on which they can read and write information, as well as delete the ones that are no longer needed. In this way, LSTMs are able to remember information about their inputs over a long period of time, avoiding the problem of vanishing gradients.
The core idea that makes LSTM possible is that each cell is composed by a memory (also called state cell) and three gates: input, forget and output gate. Each gate has a different purpose and their combination makes it possible to have a memory that resembles the human one. In Figure 14.6, there is depicted a LSTM cell at the timestamp t. Ideally, the gates protect and control the memory cell by allowing the information to pass through or not.

The first gate that operates is *forget gate*. It determines what information is not necessary and hence can be thrown away. As we can see from Figure 14.6, this gate takes as input the new value that is fed into the network at this time stamp (x_t) and the old value computed at the step before (h_{t-1}). Then, this gate evaluates a sigmoid function over these values. In Eq. (14.7) it is possible to find the mathematical formula applied for this step. With the notation W_f we refer to the matrix of weights for this gate and by b_f we refer to the bias applied for the current gate.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{14.7}$$

The output f_t from Eq. (14.7) is a number between zero and one, respectively denoting the willingness to maintain this information or to get rid of this information, considered not important. This result is then multiplied with the old cell state (C_{t-1}).

At this point, the input gate operates in order to determine what to store into the cell memory. It is actually composed by the computation of two functions. The first one takes as input the same data as before (h_{t-1} and x_t) and applies a sigmoid function. The second one, instead, feds the same inputs into a hyperbolic tangent layer to obtain a vector of new candidate values, that could be added to the state. In Eq. (14.8) there is the mathematical formulation of the computation of the input gate, while Eq. (14.9) represents the computation of the vector of candidate values. These two outputs will then be multiplied.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{14.8}$$

$$C_t^1 = \tanh\left(W_c * [h_{t-1}, x_t] + b_c\right) \tag{14.9}$$

At this point, it is possible to compute the new value for the cell state. The new value depends on the old one as well as on the results of the input and output gate (Eq. (14.10)).

$$C_t = f_t * C_{t-1} + i_t * C_t^1 \tag{14.10}$$

At this point, the new cell state has been updated and there is only the output gate that still has to operate. It determines which information to output at this current time stamp. It does so by applying a sigmoid function (Eq. (14.11)) and then by multiplying this result with a hyperbolic tangent function. This is done in order to obtain an output between $[-1; 1]$ (the formula in Eq. (14.12)).

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{14.11}$$

$$h_t = o_t * \tanh t \tag{14.12}$$

LSTM networks can be applied in different contexts. Usually, they are employed mostly for classification and prediction with data in the form of sequences, such as texts or time series data. They have been employed for language translation, text generation, handwriting recognition and labelling of images. There are several kinds of possible LSTM networks that are depicted in Figure 14.7. For the purposes of this Thesis, we focused our attention on the many-to-one kind of network. In fact, the setting of our problem corresponds to the binary classification of time series data. The two classes available represent the “at risk” class and the “not at risk” class for the task of privacy risk assessment, presented in the following, in Chapter 3.2.

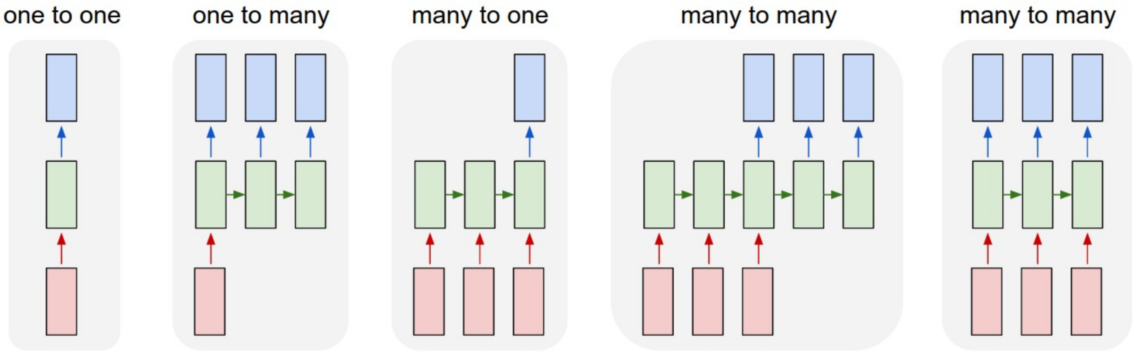


Figure 14.7: Different kinds of application of LSTM networks. In the first picture there is a one-to-one network, in which there is only one fixed-size input at each time-stamp. The second picture shows a one-to-many network, such as in the case of labelling of images. Then, a many-to-one network is depicted. This is the case of classification of time series data. A many-to-many network like the one depicted is employed for language translator. In the last picture, instead, there is a many-to-many network in which the output is synchronized with the input. This kind of networks are employed usually in contexts in which the output has to be delivered real-time, such as the labelling of videos.

Inception Time

Inception Time, a model proposed in [226], is an ensemble of deep Convolutional Neural Network (CNN) models aimed at providing a solution for Time Series Classification (TSC) that is similar to the role of “AlexNet” in computer vision. The authors achieved state-of-the-art accuracy on the UCR repository, which is the largest publicly available repository for TSC, while also reducing learning time compared to many other algorithms in the literature. It is important to note that the complexity of InceptionTime increases almost linearly with the length of the time series.

InceptionTime consists of an ensemble of five deep learning models, each of which consists of a cascade of Inception modules. While each model has the same structure, their weights are randomly initialized differently. The Inception modules simultaneously apply multiple filters of different lengths to the input time series, which allows the model to extract relevant features from both long and short time series. Notably, the time series have one dimension less than images, enabling the use of longer and more complex filters than those used in image recognition.

To provide more insight into the structure of InceptionTime, we will now present a detailed overview of the inception networks. As previously mentioned, InceptionTime is constructed using an ensemble of five Inception networks with randomly initialized weights. The classifier of each Inception network is comprised of two residual blocks, as illustrated in

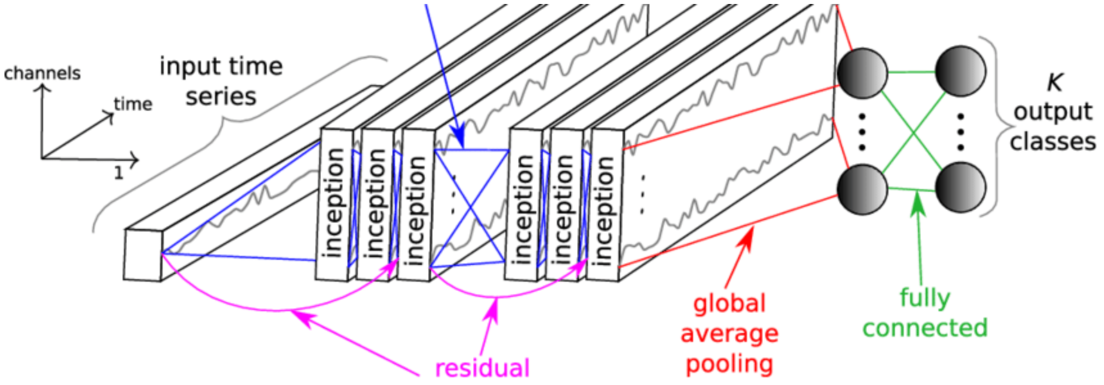


Figure 14.8: Inception Network for time series classification

Figure 14.8. Each residual block is composed of three Inception modules and is connected to the next block's input through a shortcut linear connection. This helps to alleviate the vanishing gradient problem by enabling a direct flow of the gradient. After the residual blocks, a Global Average Pooling layer is used to extract the average of the residual block's output over the time dimension. Finally, a fully connected softmax layer with a number of output neurons equal to the final number of classes is utilized. We will now provide a detailed description of the Inception module, as presented in Figure 14.9. It is important to note that the input to the module is a multivariate time series (MTS) with M dimensions. Firstly, a "bottleneck" layer convolves the input with m filters of length 1 and stride 1. This operation reduces the size of the time series from M to $m \ll M$, thus decreasing

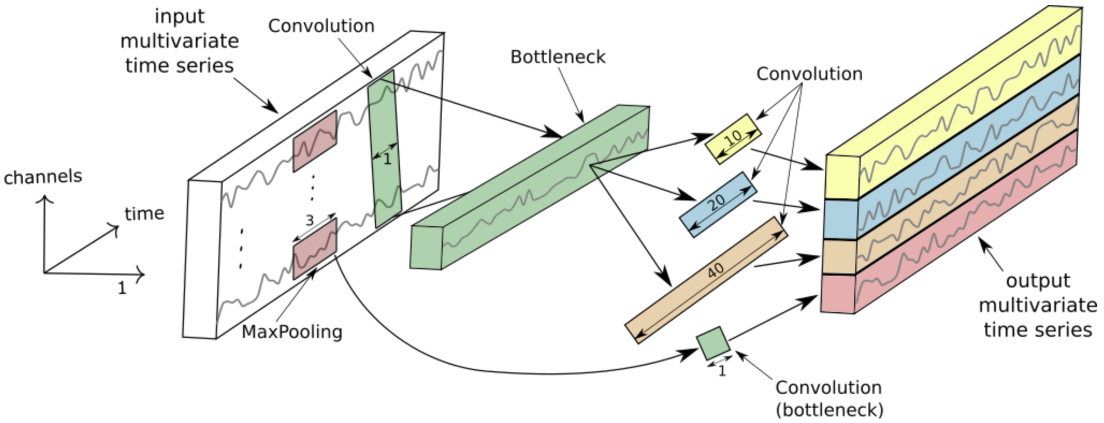


Figure 14.9: Inside of the inception module, for semplicity the bottleneck layer is represented with size $m = 1$

its complexity and preventing overfitting on small datasets. Next, filters of varying sizes are simultaneously applied to the output of the bottleneck layer. In order to maintain invariance to small perturbations, a MaxPooling operation is also performed in parallel. The MaxPooling output is produced by taking the maximum value of the time series within a sliding window of a given size. After MaxPooling, another bottleneck layer is applied to further reduce dimensionality. Finally, the outputs of each independent and parallel operation are concatenated to form the output MTS. These operations are repeated for each Inception module in the network, enabling the model to extract features of varying granularity from the time series through the use of multiple Inception modules with filters of different lengths. The filters' weights are initially set using Glorot's uniform technique [227], and the model is trained using the Adam optimization algorithm [228]. The default configuration of the proposed Inception module consists of three sets of 32 filters, with each set having a different filter length $l \in 10, 20, 40$, as well as a MaxPooling operation. The default bottleneck size is set to $m = 32$. The Receptive Field (RF) is a crucial concept in understanding how convolutional neural networks (CNNs) process time series data. Each neuron in a CNN depends on a region of the input signal, and the RF of a neuron can be defined as the region in the input space that the neuron can "see" and influence.

For time series data, the RF can be seen as the maximum field of view of the network, and a larger RF allows the network to detect longer patterns in the time series. The formula for computing the RF for a network of depth d with each layer having a filter length of k_i, $i \in [1, d]$, assuming convolutions with a stride of 1, is given by:

$$RF = \sum_{i=1}^{d} (k_i - 1) \tag{14.13}$$

From this equation, we can see that increasing the number of layers and increasing the length of the filters both increase the RF. However, adding one layer to the network only relatively increases the RF, while expanding the length of all filters greatly increases the RF as it increases the RF of all layers.

In the context of InceptionTime, the use of multiple Inception modules with filters of different lengths allows the network to have a large RF and capture features at different scales, from small details to longer patterns. Additionally, the use of residual connections and ensambling multiple networks further improves the performance and stability of the model.

Rocket

ROCKET [229] is a powerful and efficient algorithm for time series classification that achieves state-of-the-art accuracy on datasets in the UCR archive while reducing computational complexity. This is achieved through the use of a large number of random convolutional

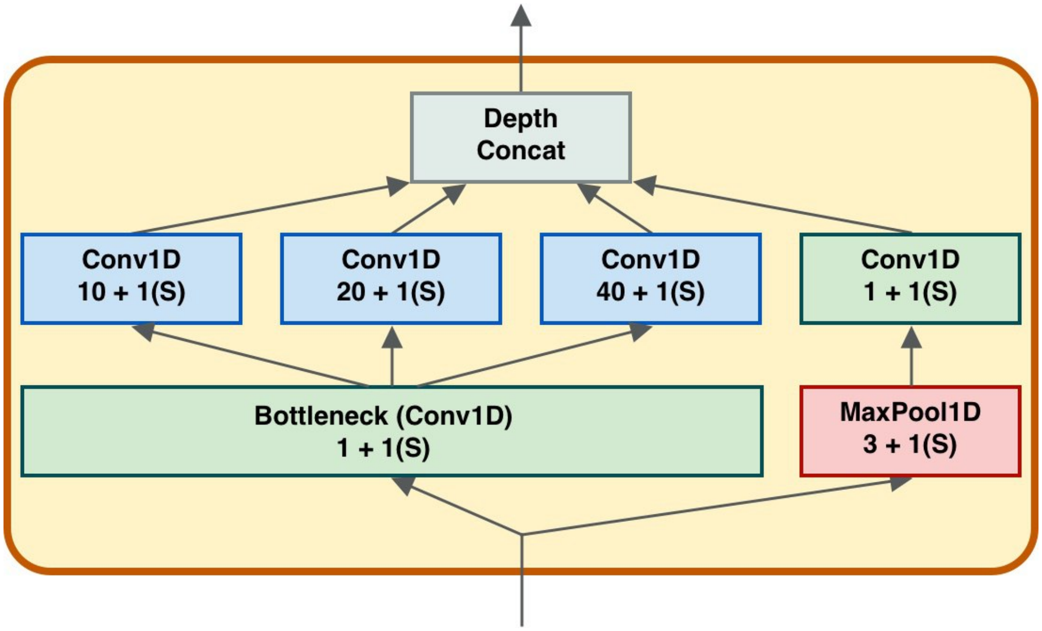


Figure 14.10: Inside of an Inception module.

kernels, which transform the time series into features that are then used to train a linear classifier. The model uses only one layer with a large number of kernels, keeping the time to compute convolutions low as the weights are not learned. The only hyper-parameter for this model is the number of kernels, which is set to 10000 in the original paper [229]. This hyper-parameter determines the trade-off between accuracy and computation time, and it is proven that increasing the number of kernels above this threshold does not improve accuracy significantly.

One of the key features of ROCKET is its resistance to configuration changes. The algorithm gave similar levels of accuracy on the development dataset despite many other possible configurations. This indicates that the model is able to generalize well to new problems, making it a useful tool for time series classification in a wide range of applications. Another advantage of ROCKET is its ability to run in parallel on multiple CPU cores, making it even more efficient than other state-of-the-art algorithms. The random convolutional kernels are applied to each time series independently, resulting in a set of features for each time series. The features extracted are the ppv, max, and min of the convolved time series. The ppv is the proportion of positive values in the convolved time series and it can be seen as a measure of how much the kernel matches the pattern in the time series. The max and min are the maximum and minimum values of the convolved time series, respectively, and they can be seen as measures of the strength of the match between the kernel and the time

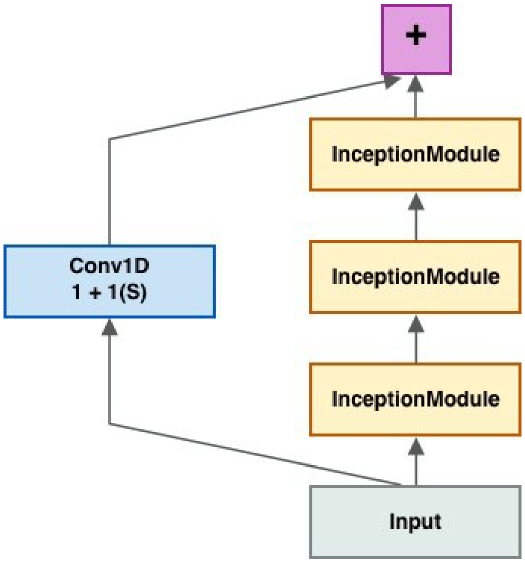


Figure 14.11: Residual connection in an Inception Network. The first number in the box Conv1D indicates the kernel size while the second indicates the size of the stride.

series.

After applying the random convolutional kernels, the resulting features are standardized to have zero mean and unit variance. Then, a linear classifier is trained using the resulting features. The linear classifier can be any linear model, such as logistic regression or support vector machines. In the original paper, a linear support vector machine (SVM) is used. The number of kernels k is a hyperparameter that can be tuned to balance between accuracy and speed. The authors found that $k = 10,000$ is a good trade-off between accuracy and speed.

One of the advantages of Rocket is that it is highly parallelizable, as each kernel can be applied to the time series independently. This makes it well-suited for large datasets with many time series. Additionally, the fact that the random convolutional kernels are generated randomly and independently of the time series means that Rocket can generalize well to new datasets without requiring fine-tuning of hyperparameters. The process involves the application of each kernel to the input time series, which produces a feature map by performing a sliding dot product between the kernel and time series. This operation calculates the dot product of the kernel ω with time series X at position i using dilation d as follows: $[X_i * \omega = \sum_{j=0}^{l_{kernel}-1} X_{i+(j\times d)} \times \omega_j$, where d is the dilation.] After obtaining feature maps, Rocket generates two aggregate features for each kernel, namely, the maximum value and the proportion of positive value (ppv) that shows the percentage of the input time series matching a given pattern in the kernel. This process has a linear computational complexity with respect to the number of examples and the length of the time series, and must be applied to both training and test sets. The formulation is $\mathcal{O}(k \cdot n \cdot l_{input})$, where k is the number of kernels, n is the number of examples, and l_{input} is the length of the time series. Finally, the transformed features are utilized to train a linear classifier. The Rocket algorithm can be used with any classifier, but it has been found to yield better results with certain types of classifiers, including logistic regression and ridge regression. Logistic regression and stochastic gradient descent (SGD) are particularly suitable for very large datasets because they allow for fast training, with the complexity of SGD being proportional to the number of parameters (determined by the number of features and the number of classes), but linear on the number of examples. On the other hand, ridge regression is a classifier trained with L_2 regularization, which works well when the number of features is greater than the number of training samples. While ridge regression is less scalable than SGD for large datasets, it can make use of generalized cross-validation to determine appropriate regularization.

Bibliography

[1] "Gdpr recital 71." `https://www.privacy-regulation.eu/en/r71.htm`. Accessed: 2019-02-08.

[2] F. Pratesi, A. Monreale, R. Trasarti, F. Giannotti, D. Pedreschi, and T. Yanagihara, "Prudence: a system for assessing privacy risk vs utility in data sharing ecosystems," *Transaction on Data Privacy*, vol. 1, pp. 139 – 167, 2018.

[3] A. Monreale and R. Pellungrini, "A survey on privacy in human mobility," *Trans. Data Priv.*, 2023.

[4] F. Pratesi, L. Gabrielli, P. Cintia, A. Monreale, and F. Giannotti, "PRIMULE: privacy risk mitigation for user profiles," *Data Knowl. Eng.*, vol. 125, p. 101786, 2020.

[5] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale, "A data mining approach to assess privacy risk in human mobility data," *ACM TIST*, vol. 9, no. 3, pp. 31:1–31:27, 2018.

[6] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, 2014.

[7] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, oct 2016.

[8] N. Papernot, M. Abadi, Úlfar Erlingsson, I. Goodfellow, and K. Talwar, "Semi-supervised knowledge transfer for deep learning from private training data," 2017.

[9] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, Association for Computing Machinery, 2009.

[10] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[11] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo, "Benchmarking and survey of explanation methods for black box models," *DAMI*, 2023.

[12] J. Schneider and J. Handali, "Personalized explanation in machine learning: A conceptualization," 2019.

[13] R. Guidotti, A. Monreale, S. Ruggieri, F. Naretto, F. Turini, D. Pedreschi, and F. Giannotti, "Stable and actionable explanations of black-box models through factual and counterfactual rules," *Data Mining and Knowledge Discovery*, 2022.

[14] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52138–52160, 2018.

[15] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89, IEEE, 2018.

[16] B. Arrieta *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges," *IF*, vol. 58, pp. 82–115, 2020.

[17] W. Samek *et al.*, *Explainable AI: interpreting, explaining and visualizing deep learning*, vol. 11700. Springer Nature, 2019.

[18] D. V. Carvalho *et al.*, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, p. 832, 2019.

[19] A. Blanco-Justicia, J. Domingo-Ferrer, S. Martínez, and D. Sánchez, "Machine learning explainability via microaggregation and shallow decision trees," *Knowledge-Based Systems*, 2020.

[20] R. Shokri, M. Strobel, and Y. Zick, "On the privacy risks of model explanations," in *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, 2021.

[21] P. Quan, S. Chakraborty, J. V. Jeyakumar, and M. Srivastava, "On the amplification of security and privacy risks by post-hoc explanations in machine learning models," 2022.

[22] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[23] R. Guidotti *et al.*, "A survey of methods for explaining black box models," *ACM computing surveys (CSUR)*, vol. 51, no. 5, pp. 1–42, 2018.

[24] A. A. Freitas, “Comprehensible classification models: a position paper,” *ACM SIGKDD explorations newsletter*, vol. 15, no. 1, pp. 1–10, 2014.

[25] F. Pasquale, *The black box society: The secret algorithms that control money and information.* Harvard University Press, 2015.

[26] A. Kurenkov, “Lessons from the pulse model and discussion. the gradient,” 2020.

[27] A. Chouldechova, “Fair prediction with disparate impact: A study of bias in recidivism,” *Big Data*, vol. 5, no. 2, pp. 153–163, 2017.

[28] B. Goodman *et al.*, “Eu regulations on algorithmic decision-making and a “right to explanation”,” in *ICML Workshop*, 2016.

[29] S. Wachter, B. Mittelstadt, and L. Floridi, “Why a right to explanation of automated decision-making does not exist in the general data protection regulation,” *International Data Privacy Law*, vol. 7, no. 2, pp. 76–99, 2017.

[30] G. Comandè, “Regulating algorithms’ regulation? first ethico-legal principles, problems, and opportunities of algorithms,” in *Transparent Data Mining for Big and Small Data*, pp. 169–206, Springer, 2017.

[31] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019.

[32] H. Yuan, H. Yu, S. Gui, and S. Ji, “Explainability in graph neural networks: A taxonomic survey,” *arXiv preprint arXiv:2012.15445*, 2020.

[33] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti, “Explainable AI for time series classification: A review, taxonomy and research directions,” *IEEE Access*, 2022.

[34] M. Gleicher, “A framework for considering comprehensibility in modeling,” *Big data*, vol. 4, no. 2, pp. 75–88, 2016.

[35] M. T. Ribeiro *et al.*, “” why should i trust you?” explaining the predictions of any classifier,” in *ACM SIGKDD*, pp. 1135–1144, 2016.

[36] M. R. Zafar and N. M. Khan, “Dlime: A deterministic local interpretable model-agnostic explanations approach for computer-aided diagnosis systems,” *arXiv preprint arXiv:1906.10263*, 2019.

[37] R. ElShawi *et al.*, “Ilime: Local and global interpretable model-agnostic explainer of black-box decision,” in *EADBIS*, pp. 53–68, Springer, 2019.

[38] S. M. Shankaranarayana and D. Runje, "Alime: Autoencoder based approach for local interpretability," in *International Conference on Intelligent Data Engineering and Automated Learning*, pp. 454–463, Springer, 2019.

[39] T. Peltola, "Local interpretable model-agnostic explanations of bayesian predictive models via kullback-leibler projections," *arXiv preprint arXiv:1810.02678*, 2018.

[40] S. Bramhall, H. Horn, M. Tieu, and N. Lohia, "Qlime-a quadratic local interpretable model-agnostic explanation approach," *SMU Data Science Review*, vol. 3, no. 1, p. 4, 2020.

[41] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *NeurIPS*, pp. 4765–4774, 2017.

[42] P. Biecek and T. Burzykowski, "Explanatory model analysis, 2020," vol. Data Science Series.

[43] M. Robnik-Šikonja and I. Kononenko, "Explaining classifications for individual instances," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, 2008.

[44] G. Plumb, D. Molitor, and A. S. Talwalkar, "Model agnostic supervised local explanations," in *Advances in Neural Information Processing Systems*, 2018.

[45] S. A *et al.*, "A python library for explaining machine learning predictions using contextual importance and utility," in *IJCAI Workshop*, 2020.

[46] H. Nori *et al.*, "Interpretml: A unified framework for machine learning interpretability," *arXiv:1909.09223*, 2019.

[47] T. J. Hastie and R. J. Tibshirani, *Generalized additive models*, vol. 43. CRC press, 1990.

[48] R. Agarwal, N. Frosst, X. Zhang, R. Caruana, and G. E. Hinton, "Neural additive models: Interpretable machine learning with neural nets," *arXiv preprint arXiv:2004.13912*, 2020.

[49] I. Puri, A. Dhurandhar, T. Pedapati, K. Shanmugam, D. Wei, and K. R. Varshney, "Cofrnets: Interpretable neural architecture inspired by continued fractions," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, 2021.

[50] R. Guidotti, A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini, and F. Giannotti, "Local rule-based explanations of black box decision systems," *CoRR*, 2018.

[51] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations.," in *AAAI*, vol. 18, pp. 1527–1535, 2018.

[52] M. N. Katehakis *et al.*, "The multi-armed bandit problem: decomposition and computation," *Mathematics of Operations Research*, vol. 12, 1987.

[53] Y. Ming *et al.*, "Rulematrix: Visualizing and understanding classifiers with rules," *IEEE TVCG*, vol. 25, no. 1, pp. 342–352, 2018.

[54] M. Craven and J. W. Shavlik, "Extracting tree-structured representations of trained networks," in *Advances in neural information processing systems*, pp. 24–30, 1996.

[55] O. Boz, "Extracting decision trees from trained neural networks," in *ACM SIGKDD*, 2002.

[56] H. Chipman, E. George, and R. McCulloh, "Making sense of a forest of trees," *Computing Science and Statistics*, 1998.

[57] P. Domingos, "Knowledge discovery via multiple models," *Intelligent Data Analysis*, 1998.

[58] Y. Zhou and G. Hooker, "Interpreting models via single tree approximation," *arXiv:1610.09036*, 2016.

[59] J. Friedman and B. E. Popescu, "Predictive learning via rule ensembles," *The Annals of Applied Statistics*, vol. 2, pp. 916–954, 2008.

[60] H. Yang *et al.*, "Scalable bayesian rule lists," in *ICML*, pp. 3921–3930, PMLR, 2017.

[61] B. Letham *et al.*, "Interpretable classifiers using rules and bayesian analysis," *AOAS*, vol. 9, no. 3, pp. 50–71, 2015.

[62] M. Setzu *et al.*, "Glocalx-from local to global explanations of black box ai models," *Artificial Intelligence*, 2021.

[63] B. Kim, R. Khanna, and O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," NIPS'16, 2016.

[64] K. S. Gurumoorthy, A. Dhurandhar, G. Cecchi, and C. Aggarwal, "Efficient data representation by selecting prototypes with importance weights," in *2019 IEEE International Conference on Data Mining (ICDM)*, pp. 260–269, IEEE, 2019.

[65] J. Bien and R. Tibshirani, "Prototype selection for interpretable classification," *The Annals of Applied Statistics*, pp. 2403–2424, 2011.

[66] S. Tan, M. Soloviev, G. Hooker, and M. T. Wells, “Tree space prototypes: Another look at making tree ensembles interpretable,” in *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, pp. 23–34, 2020.

[67] S. Wachter *et al.*, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” *HJLT*, 2017.

[68] A. Lucic, H. Haned, and M. de Rijke, “Why does my model fail?: contrastive local explanations for retail forecasting,” in *FAT* '20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, 2020.

[69] R. Guidotti, “Counterfactual explanations and how to find them: literature review and benchmarking,” *Data Mining and Knowledge Discovery*, pp. 1–55, 2022.

[70] A.-H. Karimi *et al.*, “Model-agnostic counterfactual explanations for consequential decisions,” in *International Conference on Artificial Intelligence and Statistics*, PMLR, 2020.

[71] K. Kanamori *et al.*, “Dace: Distribution-aware counterfactual explanation by mixed-integer linear optimization,” in *IJCAI-20, International Joint Conferences on Artificial Intelligence Organization*, 2020.

[72] A. Artelt *et al.*, “On the computation of counterfactual explanations–a survey,” *arXiv:1911.07749*, 2019.

[73] S. Verma, J. Dickerson, and K. Hines, “Counterfactual explanations for machine learning: A review,” *arXiv preprint arXiv:2010.10596*, 2020.

[74] R. M. Byrne and P. Johnson-Laird, “If and or: Real and counterfactual possibilities in their truth and probability.,” *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 46, no. 4, p. 760, 2020.

[75] A. Dhurandhar *et al.*, “Explanations based on the missing: Towards contrastive explanations with pertinent negatives,” in *NIPS*, 2018.

[76] M. Pawelczyk, K. Broelemann, and G. Kasneci, “Learning model-agnostic counterfactual explanations for tabular data,” in *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, 2020.

[77] R. Mothilal *et al.*, “Explaining machine learning classifiers through diverse counterfactual explanations,” in *FAT*, 2020.

[78] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, “Face: feasible and actionable counterfactual explanations,” in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 2020.

[79] E. Albini, A. Rago, P. Baroni, and F. Toni, "Relation-based counterfactual explanations for bayesian network classifiers," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI (2020, To Appear)*, 2020.

[80] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[81] M. Setzu *et al.*, "Global explanations with local scoring," in *ECML PKDD*, 2019.

[82] A. Theissler, "Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection," *Knowledge-Based Systems*, vol. 123, pp. 163 – 173, 2017.

[83] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, pp. 2048–2057, 2015.

[84] Z. Wu and D. C. Ong, "Context-guided bert for targeted aspect-based sentiment analysis," *arXiv:2010.07523*, 2020.

[85] D. Bahdanau *et al.*, "Neural machine translation by jointly learning to align and translate," in *ICLR*, 2014.

[86] S. Jain and B. C. Wallace, "Attention is not explanation," *arXiv preprint arXiv:1902.10186*, 2019.

[87] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXiv:1612.08220*, 2016.

[88] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *EMNLP*, 2016.

[89] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[90] B. Hoover, H. Strobelt, and S. Gehrmann, "exbert: A visual analysis tool to explore learned representations in transformers models," *arXiv preprint arXiv:1910.05276*, 2019.

[91] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[92] Y.-H. Lee, C.-P. Wei, T.-H. Cheng, and C.-T. Yang, "Nearest-neighbor-based approach to time-series classification," *Decision Support Systems*, vol. 53, no. 1, pp. 207–217, 2012.

[93] Z. Geler, V. Kurbalija, M. Ivanovic, and M. Radovanovic, "Weighted $k$nn and constrained elastic distances for time-series classification," *Expert Syst. Appl.*, 2020.

[94] R. Guidotti, A. Monreale, F. Spinnato, D. Pedreschi, and F. Giannotti, "Explaining any time series classifier," in *CogMI*, p. 1, IEEE, 2020.

[95] C. Panigutti *et al.*, "Doctor xai: an ontology-based approach to black-box sequential data classification explanations," in *FAT*, pp. 629–639, 2020.

[96] D. Alvarez Melis and T. Jaakkola, "Towards robust interpretability with self-explaining neural networks," *Advances in Neural Information Processing Systems*, vol. 31, pp. 7775–7784, 2018.

[97] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," in *BMVC*, 2018.

[98] R. Luss, P. Chen, A. Dhurandhar, P. Sattigeri, Y. Zhang, K. Shanmugam, and C. Tu, "Leveraging latent features for local explanations," in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, 2021.

[99] R. Guidotti, "Evaluating local explanation methods on ground truth," *Artificial Intelligence*, p. 103428, 2021.

[100] J. J. Williams, J. Kim, A. Rafferty, S. Maldonado, K. Z. Gajos, W. S. Lasecki, and N. Heffernan, "Axis: Generating explanations at scale with learnersourcing and machine learning," in *Proceedings of the Third (2016) ACM Conference on Learning @ Scale*, L@S '16, 2016.

[101] A. Suissa-Peleg, D. Haehn, S. Knowles-Barley, V. Kaynig, T. R. Jones, A. Wilson, R. Schalek, J. W. Lichtman, and H. Pfister, "Automatic neural reconstruction from petavoxel of electron microscopy data," *Microscopy and Microanalysis*, 2016.

[102] H. Lakkaraju, S. H. Bach, and J. Leskovec, "Interpretable decision sets: A joint framework for description and prediction," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684, 2016.

[103] B. Kim, C. M. Chacha, and J. A. Shah, "Inferring team task plans from human meetings: A generative modeling approach with logic-based prior," *Journal of Artificial Intelligence Research*, 2015.

[104] V. Torra, *Data Privacy: Foundations, New Developments and the Big Data Challenge*, vol. 28. Springer International Publishing, 2017.

[105] "Nytimes article." `https://www.nytimes.com/interactive/2018/12/10/business/location-data-privacy-apps.html`. Accessed: 2019-05-08.

[106] A. Monreale, S. Rinzivillo, F. Pratesi, F. Giannotti, and D. Pedreschi, "Privacy-by-design in big data analytics and social mining," *EPJ Data Science*, vol. 3, pp. 1 – 26, 2014.

[107] A. Cavoukian, "Privacy by design: The 7 foundational principles," *Information & Privacy Commissioner*, 2011.

[108] L. Rossi and M. Musolesi, "It's the way you check-in: identifying users in location-based social networks," in *Proceedings of the second ACM conference on Online social networks*, pp. 215 – 226, October, 2014.

[109] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2016.

[110] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J. V. Pearson, D. A. Stephan, S. F. Nelson, and D. W. Craig, "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays," *PLoS Genet.*, vol. 4, Aug. 2008.

[111] "Genomic privacy and limits of individual detection in a pool," *Nature Genetics*, vol. 41, pp. 965–967, Sept. 2009.

[112] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: A unifying framework for privacy definitions," in *Membership Privacy: A Unifying Framework for Privacy Definitions*, (New York, NY, USA), Association for Computing Machinery, 2013.

[113] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale, "Analyzing privacy risk in human mobility data," pp. 114 – 129, 2018.

[114] R. Pellungrini, F. Pratesi, and L. Pappalardo, "Assessing privacy risk in retail data," pp. 17–22, 11 2017.

[115] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems," *Heliyon*, vol. 5, no. 6, p. e01802, 2019.

[116] I. H. Mwinyi, H. S. Narman, K.-C. Fang, and W.-S. Yoo, "Predictive self-learning content recommendation system for multimedia contents," in *2018 Wireless Telecommunications Symposium (WTS)*, pp. 1–6, 2018.

[117] S. Ghosh and P. O. Kristensson, "Neural networks for text correction and completion in keyboard decoding," *CoRR*, vol. abs/1709.06429, 2017.

[118] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech recognition using deep neural networks: A systematic review," *IEEE Access*, vol. 7, pp. 19143–19165, 2019.

[119] A. M. Al-Zoubi, J. Alqatawna, H. Faris, and M. A. Hassonah, "Spam profiles detection on social networks using computational intelligence methods: The effect of the lingual context," *Journal of Information Science*, vol. 47, no. 1, pp. 58–81, 2021.

[120] G. D'Angelo, M. Ficco, and F. Palmieri, "Malware detection in mobile environments based on autoencoders and api-images," *Journal of Parallel and Distributed Computing*, vol. 137, pp. 26–33, 2020.

[121] A. Langevin, T. Cody, S. Adams, and P. Beling, "Synthetic data augmentation of imbalanced datasets with generative adversarial networks under varying distributional assumptions: A case study in credit card fraud detection," *Journal of the Operational Research Society*, pp. 1–28, 2021.

[122] S. Wang, Z. Chen, Q. Yan, B. Yang, L. Peng, and Z. Jia, "A mobile malware detection method using behavior features in network traffic," *Journal of Network and Computer Applications*, vol. 133, pp. 15–25, 2019.

[123] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proceedings of the 23rd USENIX Conference on Security Symposium*, SEC'14, p. 17–32, 2014.

[124] R. J. Bolton and D. J. Hand, "Statistical fraud detection: A review," in *Statistical Science*, 2022.

[125] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 1964–1974, PMLR, 18–24 Jul 2021.

[126] Y. Long, V. Bindschaedler, and C. A. Gunter, "Towards measuring membership privacy," 2017.

[127] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, “Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models,” 2018.

[128] C. Song and V. Shmatikov, “The natural auditor: How to tell if someone used your words to train their model,” *CoRR*, vol. abs/1811.00513, 2018.

[129] J. Hayes, L. Melis, G. Danezis, and E. D. Cristofaro, “Logan: Membership inference attacks against generative models,” 2018.

[130] M. Hardt, B. Recht, and Y. Singer, “Train faster, generalize better: Stability of stochastic gradient descent,” *CoRR*, vol. abs/1509.01240, 2015.

[131] A. Salem, A. Bhattacharyya, M. Backes, M. Fritz, and Y. Zhang, “Updates-leak: Data set inference and reconstruction attacks in online learning,” *ArXiv*, vol. abs/1904.01067, 2019.

[132] Z. He, T. Zhang, and R. B. Lee, “Model inversion attacks against collaborative inference,” in *Proceedings of the 35th Annual Computer Security Applications Conference*, ACSAC ’19, Association for Computing Machinery, 2019.

[133] N. Carlini, C. Liu, Úlfar Erlingsson, J. Kos, and D. Song, “The secret sharer: Evaluating and testing unintended memorization in neural networks,” 2019.

[134] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, “Stealing machine learning models via prediction apis,” 2016.

[135] C. Huang, P. Kairouz, X. Chen, L. Sankar, and R. Rajagopal, “Context-aware generative adversarial privacy,” *Entropy*, 2017.

[136] P. Samarati and L. Sweeney, “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression,” tech. rep., 1998.

[137] R.-H. Hwang, Y.-L. Hsueh, , and H.-W. Chung, “A novel time-obfuscated algorithm for trajectory privacy protection,” *IEEE Transactions on Services Computing*, vol. 7, pp. 126 – 139, April, 2004.

[138] T. Xu and Y. Cai, “Exploring historical location data for anonymity preservation in location-based services,” in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, pp. 1220 – 1228, March, 2008.

[139] C.-Y. Chow and M. F. Mokbell, “Enabling private continuous queries for revealed user locations,” *Advances in Spatial and Temporal Databases*, vol. 4605, pp. 258 – 275.

[140] C. C. Aggarwal, “On k-anonymity and the curse of dimensionality,” in *Proceedings of the 31st International Conference on Very Large Data Bases*, VLDB ’05, 2005.

[141] A. Machanavajjhala, J. Gehrke, and D. Kifer, “l-diversity: Privacy beyond k-anonymity,” in *Proceedings of the 22nd International Conference on Data Engineering*, 2006.

[142] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: Privacy beyond k-anonymity and l-diversity,” *2007 IEEE 23rd International Conference on Data Engineering*, pp. 106 – 115, June, 2007.

[143] C. Dwork, “Differential privacy,” in *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP’06, 2006.

[144] M. Lecuyer, V. Atlidakis, R. Geambasu, D. Hsu, and S. Jana, “Certified robustness to adversarial examples with differential privacy,” 2019.

[145] U. Erlingsson, V. Pihur, and A. Korolova, “Rappor: Randomized aggregatable privacy-preserving ordinal response,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS ’14, 2014.

[146] D. Kifer and A. Machanavajjhala, “Pufferfish: A framework for mathematical privacy definitions,” *ACM Trans. Database Syst.*, 2014.

[147] “Probabilistic encryption.,” 1984.

[148] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, “Differentially private empirical risk minimization,” 2011.

[149] R. Shokri and V. Shmatikov, “Privacy-preserving deep learning,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS ’15, 2015.

[150] J. Dong, A. Roth, and W. J. Su, “Gaussian differential privacy,” 2019.

[151] E. Hesamifard, H. Takabi, and M. Ghasemi, “Cryptodl: Deep neural networks over encrypted data,” 2017.

[152] T. Graepel, K. Lauter, and M. Naehrig, “Ml confidential: Machine learning on encrypted data.” Cryptology ePrint Archive, Paper 2012/323, 2012.

[153] L. J. M. Aslett, P. M. Esperança, and C. C. Holmes, “Encrypted statistical machine learning: new privacy preserving methods,” 2015.

[154] S. C. on Artificial Intelligence, "The national artificial intelligence research and development strategic plan: 2019 update," in *Executive Office of the President of the United States*, Curran Associates, Inc., 2019.

[155] F. Bodria, F. Giannotti, R. Guidotti, F. Naretto, D. Pedreschi, and S. Rinzivillo, "Benchmarking and survey of explanation methods for black box models," *CoRR*, vol. abs/2102.13076, 2021.

[156] X. Li, C. C. Cao, Y. Shi, W. Bai, H. Gao, L. Qiu, C. Wang, Y. Gao, S. Zhang, X. Xue, and L. Chen, "A survey of data-driven and knowledge-aware explainable AI," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 1, pp. 29–49, 2022.

[157] S. K. Murthy, S. Kasif, and S. Salzberg, "A system for induction of oblique decision trees," *J. Artif. Intell. Res.*, vol. 2, pp. 1–32, 1994.

[158] A. Lucic, H. Oosterhuis, H. Haned, and M. de Rijke, "Actionable interpretability through optimizable counterfactual explanations for tree ensembles," *CoRR*, vol. abs/1911.12199, 2019.

[159] L. Breiman, "Statistical modeling: The two cultures (with comments and a rejoinder by the author)," *Statistical Science*, vol. 16, pp. 199–231, 2001.

[160] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[161] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. M. Trinidad, and J. Kittler, "A review of instance selection methods," *Artif. Intell. Rev.*, vol. 34, no. 2, pp. 133–143, 2010.

[162] C. Tsai, W. Eberle, and C. Chu, "Genetic algorithms in feature and instance selection," *Knowl. Based Syst.*, vol. 39, pp. 240–247, 2013.

[163] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *Knowl. Inf. Syst.*, vol. 35, no. 2, pp. 249–283, 2013.

[164] J. Derrac, S. García, and F. Herrera, "A survey on evolutionary instance selection and generation," *Int. J. Appl. Metaheuristic Comput.*, vol. 1, no. 1, pp. 60–92, 2010.

[165] B. McCane and M. Albert, "Distance functions for categorical and mixed variables," *Pattern Recognit. Lett.*, vol. 29, no. 7, pp. 986–993, 2008.

[166] T. G. Karimpanal, "A self-replication basis for designing complex agents," GECCO '18, 2018.

[167] T. Bäck, D. B. Fogel, and Z. Michalewicz, *Evolutionary computation 1: Basic algorithms and operators*, vol. 1. CRC press, 2000.

[168] S. Wu and S. Olafsson, "Optimal instance selection for improved decision tree induction," in *IIE*, p. 1, IISE, 2006.

[169] Y. Zhang, K. Song, Y. Sun, S. Tan, and M. Udell, ""Why should you trust my explanation?" Understanding uncertainty in LIME," *arXiv*, vol. 1904.12991, 2019.

[170] Y. Jia, J. Bailey, K. Ramamohanarao, C. Leckie, and M. E. Houle, "Improving the quality of explanations with local embedding perturbations," in *KDD*, pp. 875–884, ACM, 2019.

[171] T. Laugel, X. Renard, M. Lesot, C. Marsala, and M. Detyniecki, "Defining locality for surrogates in post-hoc interpretablity," *CoRR*, vol. abs/1806.07498, 2018.

[172] A. Shih, A. Choi, and A. Darwiche, "A symbolic approach to explaining bayesian network classifiers," in *IJCAI*, pp. 5103–5111, ijcai.org, 2018.

[173] A. Darwiche and A. Hirth, "On the reasons behind decisions," in *ECAI*, vol. 325 of *Frontiers in Artificial Intelligence and Applications*, pp. 712–720, IOS Press, 2020.

[174] F. Fortin, F. D. Rainville, M. Gardner, M. Parizeau, and C. Gagné, "DEAP: evolutionary algorithms made easy," *J. Mach. Learn. Res.*, vol. 13, pp. 2171–2175, 2012.

[175] P. Tan, M. S. Steinbach, and V. Kumar, *Introduction to Data Mining.* Addison-Wesley, 2005.

[176] C. Russell, "Efficient search for diverse coherent explanations," in *FAT*, pp. 20–28, ACM, 2019.

[177] A. Gosiewska and P. Biecek, "Do not trust additive explanations," *CoRR*, vol. abs/1903.11420, 2020.

[178] M. Sundararajan and A. Najmi, "The many Shapley values for model explanation," in *ICML*, vol. 119 of *Proceedings of Machine Learning Research*, pp. 9269–9278, PMLR, 2020.

[179] R. Guidotti and S. Ruggieri, "On the stability of interpretable models," in *IJCNN*, pp. 1–8, IEEE, 2019.

[180] R. Guidotti and A. Monreale, "Data-agnostic local neighborhood generation," in *ICDM*, pp. 1040–1045, IEEE, 2020.

[181] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*, pp. 413–422, IEEE, 2008.

[182] I. Guyon, “Design of experiments of the NIPS 2003 variable selection benchmark,” in *NIPS Workshops*, 2003.

[183] A. Klimke, “RANDEXPR: A random symbolic expression generator,” Tech. Rep. 4, Universitat Stuttgart, 2003.

[184] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[185] Y.-A. de Montjoye *et al.*, “Unique in the crowd: The privacy bounds of human mobility,” *Sci. Rep.*, vol. 3, p. 1376, 2013.

[186] M. Terrovitis and N. Mamoulis, “Privacy preservation in the publication of trajectories,” in *MDM*, pp. 65–72, 2008.

[187] A. Monreale *et al.*, “Movement data anonymity through generalization,” *TDP*, vol. 3, no. 2, pp. 91–121, 2010.

[188] G. Cormode, C. M. Procopiuc, D. Srivastava, and T. T. L. Tran, “Differentially private summaries for sparse data,” in *ICDT ’12*, pp. 299–311, 2012.

[189] F. Naretto, R. Pellungrini, A. Monreale, F. M. Nardini, and M. Musolesi, “Predicting and explaining privacy risk exposure in mobility data,” in *Discovery Science - 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19-21, 2020, Proceedings*, Lecture Notes in Computer Science, Springer, 2020.

[190] F. Naretto, R. Pellungrini, F. M. Nardini, and F. Giannotti, “Prediction and explanation of privacy risk on mobility data with neural networks,” in *ECML PKDD 2020 Workshops*, Springer International Publishing, 2020.

[191] F. Naretto, R. Pellungrini, D. Fadda, and S. Rinzivillo, “Exphlot: Explainable privacy assessment for human location trajectories,” in *Under submission*, 2023.

[192] C. Song, T. Koren, P. Wang, and A.-L. Barabasi, “Modelling the scaling properties of human mobility,” *Nature Physics*, vol. 6, 10 2010.

[193] S. Jiang, Y. Yang, S. Gupta, D. Veneziano, S. Athavale, and M. C. Gonzalez, “The timegeo modeling framework for urban mobility without travel surveys,” *Proceedings of the National Academy of Sciences*, vol. 113, p. 201524261, 08 2016.

[194] G. Cornacchia, M. Böhm, G. Mauro, M. Nanni, D. Pedreschi, and L. Pappalardo, “How routing strategies impact urban emissions,” in *SIGSPATIAL/GIS*, pp. 42:1–42:4, ACM, 2022.

[195] M. Tizzoni, P. Bajardi, A. Decuyper, G. Kon Kam King, C. Schneider, V. Blondel, Z. Smoreda, M. C. Gonzalez, and V. Colizza, "On the use of human mobility proxies for modeling epidemics," *PLoS computational biology*, vol. 10, p. e1003716, 07 2014.

[196] Z.-H. Zhou and J. Feng, "Deep forest: Towards an alternative to deep neural networks," in *IJCAI*, pp. 3553–3559, 2017.

[197] Y.-L. Zhang *et al.*, "Distributed deep forest and its application to automatic detection of cash-out fraud," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 5, 2019.

[198] C. I. Muntean, F. M. Nardini, F. Silvestri, and R. Baraglia, "On learning prediction models for tourists paths," *ACM Trans. Intell. Syst. Technol.*, vol. 7, no. 1, pp. 8:1–8:34, October, 2015.

[199] L. Pappalardo, F. Simini, S. Rinzivillo, D. Pedreschi, F. Giannotti, and A.-L. Barabási, "Returners and explorers dichotomy in human mobility," in *Nature communications*, 2015.

[200] N. Eagle and A. S. Pentland, "Eigenbehaviors: identifying structure in routine," *Behavioral Ecology and Sociobiology*, vol. 63, pp. 1057–1066, 2009.

[201] N. Mohammed, B. C. Fung, and M. Debbabi, "Walking in the crowd: Anonymizing trajectory data for pattern analysis," in *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pp. 1441–1444, 2009.

[202] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?: Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135 – 1144, August, 2016.

[203] N. V. Andrienko and G. L. Andrienko, "Spatial generalization and aggregation of massive movement data," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 2, pp. 205–219, 2011.

[204] A. Dempster, D. F. Schmidt, and G. I. Webb, "MiniRocket," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ACM, aug 2021.

[205] R. Mitchell, J. Cooper, E. Frank, and G. Holmes, "Sampling permutations for shapley value estimation," *Journal of Machine Learning Research*, vol. 23, no. 43, pp. 1–46, 2022.

[206] A. Rodriguez-Carrion, D. Rebollo-Monedero, J. Forné, C. Campo, C. Garcia-Rubio, J. Parra-Arnau, and S. K. Das, "Entropy-based privacy against profiling of user mobility," *Entropy*, vol. 17, no. 6, pp. 3913–3946, 2015.

[207] J. Buchmüller, H. Janetzko, G. L. Andrienko, N. V. Andrienko, G. Fuchs, and D. A. Keim, "Visual analytics for exploring local impact of air traffic," *Comput. Graph. Forum*, vol. 34, 2015.

[208] "Octo telematics." `https://www.octotelematics.com/it/`. Accessed: 2019-05-08.

[209] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[210] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, 2015.

[211] A. Monreale, F. Naretto, and S. Rizzo, "Agnostic label-only membership inference attack," in *17th International Conference on Network and System Security*, Springer, 2023.

[212] F. Naretto, A. Monreale, and F. Giannotti, "Evaluating the privacy exposure of interpretable global explainers," in *4th IEEE International Conference on Cognitive Machine Intelligence, CogMI 2022, Atlanta, GA, USA, December 14-17, 2022*, pp. 13–19, IEEE, 2022.

[213] F. Naretto, A. Monreale, and F. Giannotti, "Evaluating the privacy exposure of interpretable global and local explainers," in *Submitted at Transactions on Data Privacy*, 2023.

[214] L. Song, R. Shokri, and P. Mittal, "Membership inference attacks against adversarially robust deep learning models," in *2019 IEEE Security and Privacy Workshops (SPW)*, 2019.

[215] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, 2018.

[216] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," *2019 IEEE Symposium on Security and Privacy (SP)*, 2019.

[217] N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, A. Oprea, and C. Raffel, "Extracting training data from large language models," 2021.

[218] Y. Zheng, "Trajectory data mining: An overview," *ACM TIST*, vol. 6, no. 3, pp. 29:1–41, 2015.

[219] L. Rokach and O. Maimon, “Top-down induction of decision trees classifiers - a survey,” *Trans. Sys. Man Cyber Part C*, vol. 35, no. 4, pp. 476–487, 2005.

[220] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, pp. 278–282 vol.1, 1995.

[221] T. Chen and C. Guestrin, “XGBoost,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM.

[222] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “Catboost: unbiased boosting with categorical features,” in *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), vol. 31, Curran Associates, Inc., 2018.

[223] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *Bulletin of Mathematical Biology*, vol. 52, pp. 99–115, 1988.

[224] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Neurocomputing: Foundations of research,” ch. Learning Representations by Back-propagating Errors, pp. 696–699, 1988.

[225] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Neurocomputing: Foundations of research,” ch. Learning Representations by Back-propagating Errors, pp. 696–699, 1988.

[226] G. Ismail Fawaz, Lucas Forestier, “Inceptiontime: Finding alexnet for time series classification.,” *Data Mining and Knowledge Discovery 34*, 2020.

[227] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, Proceedings of Machine Learning Research, 2010.

[228] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[229] A. Dempster, F. Petitjean, and G. I. Webb, “Rocket: exceptionally fast and accurate time series classification using random convolutional kernels,” *Data Mining and Knowledge Discovery*, 2020.