RANKINGS AND IMPORTANCE SCORES AS MULTI-FACETS OF EXPLAINABLE MACHINE LEARNING

CHIARA BALESTRA

Dissertation
zur
Erlangung des Doktorgrades (Dr. rer. nat.)
der
Fakultät für Informatik
der
Technischen Universität Dortmund

APRIL 2024, DORTMUND

Angefertigt mit Genehmigung der Fakultät für Informatik
der Technischen Universität Dortmund

Chiara Balestra: *Rankings and importance scores as multi-facets of explainable Machine Learning*

GUTACHTER:
Prof. Dr. Emmanuel Müller
Prof. Dr. Andreas Mayr
Prof. Dr. Tijl De Bie

DORTMUND, APRIL 2024

ABSTRACT

Rankings represent the natural way to access the importance of a finite set of items. Ubiquitous in real-world applications and machine-learning methods, they mostly derive from automated or human-based importance score assignments. Many fields involving rankings, such as Recommender Systems, feature selection, and anomaly detection, overlap with human-derived scoring systems, such as candidate selection and operational risk assessments. Rankings are explicitly hard to evaluate; several challenges derive from concerned biases, fairness issues, and also from their derivation and evaluation.

This thesis spins around deriving importance scores and rankings as solutions in various contexts and applications. Starting from unsupervised feature importance scores based on an unconventional use of Shapley values for unlabeled data, it will touch a more applied field with an ad-hoc unsupervised methodology for reducing the dimensionality of collections of gene sets. We then focus on feature importance scores in a time-dependent context, focusing on detecting correlational concept drifts in the univariate dimensions of unlabeled streaming data. The whole work is commonly characterized by seeking to improve abstract concepts of trustworthiness and reliability, with an open eye on the consistency of evaluations and methods. In this direction, we add insights into using saliency importance score assignments for interpreting time series classification methods and define desirable mathematical properties for ranking evaluation metrics. Furthermore, we use Shapley values to interpret unsupervised anomaly detection deep methods based on features bagging. Lastly, we introduce some future and current challenges related to fairness issues in rank aggregations and some possible extensions of the current work.

PUBLICATIONS

[BFM24] Chiara Balestra, Antonio Ferrara, and Emmanuel Müller. "FairMC – fair Markov Chain rank aggregation methods." In: *DaWaK*. 2024.

[Bal+22] Chiara Balestra, Florian Huber, Andreas Mayr, and Emmanuel Müller. "Unsupervised Features Ranking via Coalitional Game Theory for Categorical Data." In: *DaWaK*. 2022.

[BLM23a] Chiara Balestra, Bin Li, and Emmanuel Müller. "On the Consistency and Robustness of Saliency Explanations for Time Series Classification." In: *arXiv preprint arXiv:2309.01457* (2023).

[BLM23b] Chiara Balestra, Bin Li, and Emmanuel Müller. "slidSHAPs – sliding Shapley Values for correlation-based change detection in time series." In: *DSAA*. 2023.

[Bal+23] Chiara Balestra, Carlo Maj, Emmanuel Müller, and Andreas Mayr. "Redundancy-aware unsupervised ranking based on game theory: Ranking pathways in collections of gene sets." In: *Plos one* 18.3 (2023), e0282699.

[BMM24] Chiara Balestra, Andreas Mayr, and Emmanuel Müller. "Ranking evaluation metrics from a group-theoretic perspective." In: *under review*. 2024.

[KBM24] Simon Klüttermann, Chiara Balestra, and Emmanuel Müller. "On the efficient Explanation of Outlier Detection Ensembles through Shapley Values." In: *PAKDD*. 2024.

[LBM22] Bin Li, Chiara Balestra, and Emmanuel Müller. "Enabling the visualization of distributional shift using Shapley values." In: *distSHIFT@NIPS*. 2022.

ACKNOWLEDGMENTS

I would like to thank everybody who advised me about research during my Ph.D. and helped me get to the final version of my thesis, particularly Prof. Emmanuel Müller, Prof. Andreas Mayr, Anton Tsitsulin, and my colleagues in Bonn and Dortmund.

CONTENTS

II About trustworthiness of evaluations and methods

1 INTRODUCTION AND THESIS OVERVIEW

Ordering items is an intrinsically human task; ordered lists appear in everyday life, easily understandable and maneuverable. An ordered list or *ranking* is easily interpretable, summarizing the relevance of the items in the defined context straightforwardly. Rankings often appear as a by-product of given importance score assignments. Given importance or relevance scores, we can easily rank items; however, the contrary does not usually hold.

Being a powerful tool, we find rankings not only in concrete everyday applications but also as solutions to complex tasks in the Machine Learning community; typical examples are Recommender Systems and Information Retrieval techniques [SMR08; Lin10]. From raising the understanding of black-box models to ranking candidates in job applications, assigning importance scores to items has become common and often necessary. The goal is to create summarizing scores assessing the value of the single items in finite sets based on their contributions to a task. Scores can be both the result of human scoring attributions or of complex machine learning techniques. Once available, as mentioned, scores can be used to rank elements and obtain an ordered list. We believe rankings can offer solutions to the most disparate problems. The present thesis aims to prove this claim in several domains, from data mining to genetics applications, from time series applications and explainability needs, where rankings and importance scores, primarily based on Shapley values assignments, will offer straightforward solutions. The conducting rope of our work is to be found in the proposed solutions more than in the addressed problems, where rankings and importance scores will offer solutions to challenges deriving from disparate contexts.

Our starting point is a data mining context, where the focus is often on the data structure. Unlabeled data form a special case where the input features are available and observed, while no labels are connected to the data points, thus precluding any prediction model. On the one hand, the spread of sensors, making collecting time-dependent data points easier and cheaper, offers a consistent source of unlabeled data; on the other hand, labels are often hard to obtain, delayed in online data streams, or undefined in multi-objective contexts. Thus, the study of correlations among variables is particularly interesting in the ever-increasing applications' setups where data are primarily unlabeled. Our work addresses several problems raised when dealing with unlabeled data, going from feature ranking and selection to change point detection and interpretability of anomaly detection methods. We particularly focus on Shapley values [Sha+53], an essential tool for assigning importance scores to items in cooperative contexts that additionally satisfy desirable mathematical properties. We introduce Shapley values importance scores and their rankings to keep track of the concept structure in unlabeled data streams [BLM23b], reduce the dimension of tabular unlabeled data [Bal+22], and rank pathways in collections of gene sets [Bal+23]. We make sense of the information conveyed by unlabeled data by quantifying the "correlation" among groups of univariate dimensions, deriving Shapley values-based features' importance scores, and ranking and selecting features through them.

Shapley values are "interpretable" scores introduced in machine learning to assess the role of single features in model predictions [LL17]. More generally, Shapley values allow for ranking the players concerning their influence in a so-called *game*; although their acknowledgment is often limited to the interpretable machine learning community, Shapley values are much more than interpretable scores. We use Shapley values both as importance scores for explaining anomaly detection scores [KBM24] and also to extend their application far beyond claiming they are interpretable scores. However, using Shapley values-based importance scores to rank items does not arrive without disadvantages. Their intrinsic exponential complexity demands approximation methods whose performance highly differs in the various setups such that the rankings offered by the approximated scores present slight differences; these differences proved hard to examine, and the innumerable inconsistencies among evaluations represent a second facet we incur when ground truth labels are unavailable. Namely, the inconsistencies among metrics are an issue plaguing all the state-of-the-art literature [TDV21]; using one metric instead of another produces somewhat different evaluations, thus introducing additional challenges due to a lack of reproducibility and coherence. We further investigate the inconsistency issue in the contexts of ranking evaluation metrics and saliency importance scores for time series classification methods.

In conclusion, this thesis discusses the role of importance scores and rankings in various contexts; through the eyes of Shapley values scores, we will explore unlabeled time series, unlabeled tabular datasets, and genetic data. We will then review the traps of the inconsistencies among evaluation methods, discussing ranking evaluation metrics [BMM24] and saliency importance scores for time series classification methods [BLM23a]. The critic's eye on the methods and metrics' trustworthiness, consistency, and interpretability is a common strand in our work, and, with our last contribution, we introduce Shapley values to interpret bagging models for unsupervised deep anomaly detection methods [KBM24].

1.1 OVERVIEW AND CONTRIBUTIONS

We organized this thesis into two main parts: Part I (Chapters 4, 5, and 6) focuses on unsupervised importance scores; Part II (Chapters 8, 9, and 10) focuses on the consistency of metrics and saliency interpretation methods and on Shapley values to interpret bagging models-based anomaly detectors. Chapter 2 contains the description of common notations and fundamental concepts, while Chapters 3 and 7 provide, respectively, the introduction and the related work necessary to Part I and Part II.

Here, we present an overview of our major contributions and the general structure of the thesis. Table 1.1 presents a bird's-eye view of our contributions. In the *challenges* we address, we focus on various aspects: (i) the construction of unsupervised (feature) importance scores, (ii) the consistency of evaluations and methods, and (iii) the trustworthiness and interpretability of methods.

	core aspects		
	unsupervised scores	consistency	trustworthiness and interpretability
method	Chapters 4, 5, 6		Chapter 6,10
application	Chapter 5		
analysis		Chapters 8, 9	Chapters 9

Table 1.1: OVERVIEW OF THE THESIS CONTRIBUTIONS.

1.1.1 *Part I: Shapley values-based unsupervised feature importance scores*

In the first part of the thesis, we aim to study the structure of unlabeled data using its characterizing correlations, with the scope of reducing their original dimension or detecting change points in unlabeled data streams. In particular, we work with three different types of unlabeled data, each introducing its challenges: unlabeled tabular data sets, collections of gene sets, and unlabeled data streams. The structure of this first part follows the respective data types and correlated challenges. The absence of a prediction goal is common in our starting setups, and rankings and Shapley values-based importance scores commonly characterize our solutions.

Unsupervised feature importance scores represent a world relatively unexplored for one main reason: it is hard to get such scores without a prediction task, as the question "Important for what?" is not easily answerable. Consequently, in machine learning, feature importance scores are prevalent in supervised learning to measure the roles of features for the task at hand. As mentioned, the literature does not fully cover unsupervised and semi-supervised cases. Traditional measures, such as pairwise correlation metrics and pseudo-labels, represent, in most cases, the only alternative to obtain unsupervised importance scores; however, the first ones are not sensitive to higher-order interactions, while the second ones are not intrinsically interpretable. Nevertheless, feature importance scores represent an extremely useful tool, e.g., in feature selection, explainable machine learning, recommendations, and information retrieval tasks. We answer the question "How to quantify the information brought by a feature in an unlabeled context?" by relying on Shapley values, assuming that they offer fair and interpretable scores' assignments [Sha+53]. We briefly introduce the structure of the chapters of the present part.

Chapter 4 focuses on unlabeled tabular data retaining most of the information conveyed by the original data set; our challenge is obtaining an unsupervised feature selection approach. We believe in a ranking-based solution and look for a way to rank features based on available information. We interpret the unlabeled tabular data as a "game", where features collaborate in giving all the information they contain and use Shapley values as feature importance scores. Considering the information theory-based goal, we use the total correlation as a value function; however, this decision restricts the approach's applicability to categorical data. Furthermore, as using the total correlation as the value function implicitly assigns prominent positions to features highly correlated with many others, by introducing a pruning criterion, we avoid selecting highly correlated features and favor diversity. Hence, we use the Shapley values scores within a greedy selection algorithm to obtain a correlation-free ranking of the features and a subset of features representative of the entire dataset.

We switch then to collection of gene sets, a specific data type typical of Bioinformatics; the challenge here is the data's high dimensionality and low maneuverability. Driven by the necessity of aggregating genes dependent on their biological and chemical function, gene sets reflect these relations, grouping genes depending on their roles. Above gene sets, we find macroscopic structures, so-called *collections of gene sets* that further group the gene sets based on their biological role [Lib+15]. Collections of gene sets are generally characterized by a vast size and low interpretability of the many gene sets, ranging between hundreds to several thousand partly overlapping gene sets; the chances for the human eye to recollect the information conveyed by the collections and the contained sets of genes are low. Chapter 5 proposes to solve the challenge by ranking gene sets based on Shapley values and reducing the collections to the top k gene sets in the ordered lists. The setup is general enough to produce importance scores in any family of sets. We create a value function to measure the distribution of genes throughout the whole collection. The obtained Shapley values are positively correlated with the size of the gene sets, and they are unaware of intersections among them; hence, overlapping sets are located in similar ranking positions. A pruning criterion addresses the issue based on a reformulation of the *Jaccard score*, and the sets' final ranking shows no correlation with the sets' sizes and low overlap among similarly ranked sets. Furthermore, the pruning method also affects the correlation between the size of gene sets and their position in the rankings, although not directly meant to solve this issue. The obtained rankings show excellent behavior in solving a min-max-max problem, i.e., minimizing the overlap and maximizing both the importance scores assigned by Shapley values and the coverage of genes. Note that in typical Gene Set Enrichment Analysis applications, the gene sets are statistically tested for over- or underrepresented genes for specific phenotypic traits, and the results are then aggregated through multiple hypothesis correction procedures. Reducing the collections' sizes is generally useful to maintain a high significance level, and we do this independently from a particular phenotype.

Finally, Chapter 6 considers unlabeled data streams and time series data. Distributional changes or *concept drifts* are a severe problem when dealing with time series data [Lu+18]. *Supervised drift detection* methods concern changes in the conditional distribution of the label y with respect to the input features X, while *unsupervised drift detection* approaches concern only the distribution of X when the label y is unavailable. In Chapter 6, we focus on distributional shifts inducing changes in the correlation structure of multivariate time series; generally, those distributional changes are hard to visualize within the chaotic behavior of the time series. We develop a method based on the importance scores from Chapter 4, here used to obtain a proxy representation of the time series in terms of its correlation structure, i. e., the SLIDSHAP series. The SLIDSHAP series makes concepts visually detectable, and, using univariate statistical tests on its univariate dimensions, we can formally confirm their hypothetical localization. The slidSHAP series uses the Shapley values as time-dependent functions, thus considering the changes in the correlation structure along the passing time. By using a correlation-based approach to create the slidSHAP series we were able to overcome the challenges of classical change point detectors; state-of-the-art change detectors fail to analyze complex correlation structures among the time series input dimensions, limiting to covariance evaluation studies [Qah+15; AK18] and often missing the complex multivariate interactions [KMB12; She+17].

1.1.2 *Part II: trustworthiness of methods and evaluations*

In Part I, we focus on solutions based on rankings and importance scores. In Part II, we ask whether the evaluation provided by these scores and the obtained ordering are trustworthy. A key concept is the one of *consistency*, which is related to the trustworthiness of both methods and evaluations; a consistent method or metric acts or performs in the same way over time or under the same conditions. In the relative chapters, we will mathematically define what "consistency" means for ranking evaluation metrics and saliency importance scores for time series classification approaches. Furthermore, we will work with Shapley values by providing interpretable scores for black-box models, particularly for bagging models-based unsupervised anomaly detectors.

We briefly introduce our next challenges. Scores and rankings often go one to one. Scores are usually more informative than rankings but more challenging to compare, particularly in cases where the arrays of scores are not characterized by similar scale or variance. Rankings are broadly spread; they derive not only from importance scores but also from Recommender Systems (RS) and Information Retrieval (IR) techniques [AT05; SMR08], feature ranking and selection approaches [KD22], surveys, and questionnaires where scores for single items are not provided as well as from (fair) rank aggregation methods [Lin10]. The evaluation of rankings is particularly challenging, with contradictory evaluations being commonplace. Examples are metrics such as recall@k and NDCG, both used in feature selection approaches but also in RS- and IR evaluation. Chapter 8 introduces desirable theoretical properties for ranking evaluation metrics and provides the mathematical framework underpinning each. Since the state-of-the-art literature predominantly confines itself to narrow, highly specific contexts, we raise the metrics from the context-specific requirements to a general projection on mathematical group structures, namely on *symmetric groups*. This setup allows us to detach from specific machine learning contexts and prove some general mathematical results on the nature of the single metrics. Symmetric groups are the most general mathematical structure on which we could represent rankings, thus explaining our choice.

However, the lack of consistent evaluations, approaches, and methods is not limited to ranking evaluation metrics. In the first place, in post-hoc explanations, we often find inconsistent behaviors [Kri+22] that mine the trustworthiness of both the methods we are trying to explain and the explanations themselves. Chapter 9 focuses on time series, a data type where most implemented methods still lack explanations. Here, attention-based models were revealed to help obtain time-dependent explanations [Kaj+19; Son+18; Cho+16]; initially implemented for images, attention weights are used as feature-time importance scores and visualized in saliency maps. However, treating time windows of data streams as images and applying computer vision explanation methods remains questionable. Highly relevant for claiming the trustworthiness of the methods, two main issues arise when using saliency maps to explain time series data predictions: the *consistency* and the *robustness* of the obtained explanations. The reported experiments show that saliency explanation methods for time series classification do not satisfy these properties. In images, the semantic meaning of columns and rows is equivalent, and perturbation methods through data augmentation processes guarantee consistent behaviors; however, in time series, the time structure makes time series data semantically different and introduces dependence among the observations in the various timestamps. We examine saliency explanations from popularly used approaches on multiple deep classification models [HS97; Lea+17; Vas+17] and study these issues on various real-world datasets.

Nevertheless, post-hoc explanations and saliency maps revealed their utility in interpreting black box models, spotting bugs and fairness-related issues in machine learning models, communicating their results, and raising user trust. The proof that these explanations are, in most cases, not perfect and that we need to be cautious when using them is not sufficient to stop working towards more accessible methods and explainability in AI. By introducing the BAGGED SHAPLEY VALUES in Chapter 10, we propose a use case of Shapley values as interpretable scores for bagging-based unsupervised anomaly detectors; the importance scores represent the role of the single input features to the deduction of the corresponding anomaly score. The particular setup avoids the well-known exponential complexity proper of Shapley values; the BAGGED SHAPLEY VALUES are computed in polynomial time.

2 BACKGROUND AND NOTATION

This chapter establishes the notions and definitions used in the thesis and reviews the necessary background. We begin with the basic definitions of cooperative games and rankings.

2.1 NOTATION AND COMMON SYMBOLS

Throughout the work, we use bold capital letters or matrices, calligraphic letters $\mathcal{M}, \mathcal{N}$ for sets, lowercase Greek letters $\sigma, \mu, \nu, \dots,$ for rankings and lowercase letters $a, b, c, \dots$ for scalars and functions. Some notable symbols and exceptions to these rules are presented in Table 2.1 below.

Symbol	Description
$\mathcal{N}$	a set of players, $\lvert\mathcal{N}\rvert = N$
$(\mathcal{N}, f)$	a game, with set of players $\mathcal{N}$ and f as value function
i	a player of a game
f	the value function of a game
$\phi_f(i)$	Shapley value of player i in a game
$\Delta_f(\mathcal{S}, i)$	the marginal contribution of i to $\mathcal{S}$ in the game $(\mathcal{N}, f)$
S_N	symmetric group over N elements
σ	a permutation or ranking $\sigma \in S_N$
t_s	a time stamp
w_s	a time window

Table 2.1: Common symbols and notation.

2.2 COOPERATIVE GAMES

At least half of the thesis revolves around Shapley values, a concept derived from Cooperative Game Theory (CGT). We first define what a coalition game is.

Definition 2.2.1. *A* cooperative *(or* coalitional*)* game *is a pair $(\mathcal{N}, f)$ where $\mathcal{N}$ is a finite set of players $\mathcal{N} = \{1, \dots, N\}$ and f is a function over the power set of players $\mathcal{P}(\mathcal{N})$, i.e., $f : \mathcal{P}(\mathcal{N}) \mapsto \mathbb{R}$.*

We refer to f as the value function.

The value function assigns to *coalitions* (or sets) of players a real number, and it is usually assumed to satisfy the following properties:

- $f(\emptyset) = 0$,
- (**non-negativity**) $f(\mathcal{A}) \geq 0$ for any $\mathcal{A} \subseteq \mathcal{N}$, and
- (**monotonicity**) for any $\mathcal{A}, \mathcal{B} \subseteq \mathcal{N}$ and $\mathcal{A} \subseteq \mathcal{B}$, $f(\mathcal{A}) \leq f(\mathcal{B})$.

Under the monotonicity assumption, the *grand coalition* $\mathcal{N}$ is the set assuming the maximum of the value function f. We are interested in assigning each player his worth; these scores should sum up to $f(\mathcal{N})$ and be "fair" concerning the actual value brought by each player to the coalitions. Among various possibilities, a solution to this allocation problem is represented by the Shapley values; given a cooperative game $(\mathcal{N}, f)$ and a player $i \in \mathcal{N}$, we indicate the Shapley values of the player i as $\phi_f(i)$. Furthermore, the Shapley values are the unique allocations that additionally satisfy

1. the *Pareto optimality* or *efficiency property*, i.e., $\sum_{i \in \mathcal{N}} \phi_f(i) = f(\mathcal{N})$,
2. the *null-player* or *dummy property* , i.e., given $i \in \mathcal{N}$ such that $f(\mathcal{A} \cup \{i\}) = f(\mathcal{A})$ for each $\mathcal{A} \subseteq \mathcal{N}$ it holds $\phi_f(i) = 0$,
3. the *linearity property*, i.e., given two games $(\mathcal{N}, f)$, $(\mathcal{N}, g)$ on $\mathcal{N}$ it holds $\phi_{f+g}(i) = \phi_f(i) + \phi_g(i)$,
4. and the *symmetry property*, i.e., given $i, j \in \mathcal{N}$ such that $f(\mathcal{A} \cup \{i\}) = f(\mathcal{A} \cup \{j\})$ for each $\mathcal{A} \subseteq \mathcal{N}$ implies $\phi_f(i) = \phi_f(j)$.

The formal definition of Shapley values is as follows:

Definition 2.2.2. *Given a coalitional game $(\mathcal{N}, f)$ and a player $i \in \mathcal{N}$, the* Shapley value *of i is defined by*

$$\phi_f(i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus i} \frac{1}{N\binom{N-1}{|\mathcal{A}|}} \left[f(\mathcal{A} \cup i) - f(\mathcal{A}) \right]. \tag{2.1}$$

We refer to the quantity

$$\Delta_f(\mathcal{A}, i) = f(\mathcal{A} \cup i) - f(\mathcal{A})$$

as *marginal contribution* of player i to the subset $\mathcal{A}$, c.f.. Figure 2.1.

Note that each player is part of 2^{N-1} sub-coalitions of players. The immediate consequence of the exponential number of subsets of players available is the computational hardness of Shapley values; thus, it is often required to substitute the full computation of

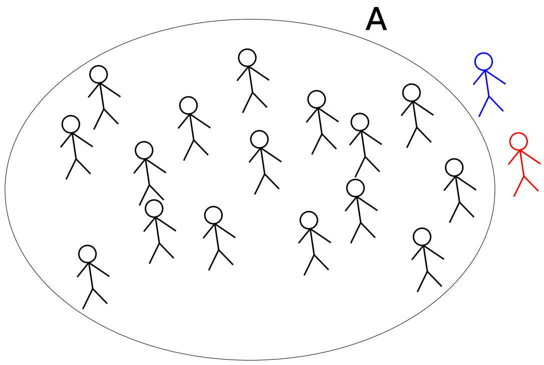


Figure 2.1: MARGINAL CONTRIBUTION. Given the value function f and a set of players $\mathcal{A}$, the marginal contributions of player • to $\mathcal{A}$ is given by $f(\mathcal{A} \cup \bullet) - f(\mathcal{A})$.

Shapley values with one of their available approximations [Cam+18; CGT09; BC21]. Furthermore, common characteristics might characterize players, and the Shapley values guarantee that "similar" players obtain similar scores. We first explain the phenomenon using a toy example, but we will return to this argument later in the thesis.

2.2.1 *Glove game*

A classic example of a cooperative game is the so-called "glove game". Consider the set of players $\{a, b, c\}$; a and b are right-hand gloves, while c is a left-hand glove.

A coalition, i.e., a subset of $\{a, b, c\}$, gets 1 as a worth if it contains a pair of gloves (left + right) and 0 if it does not. A person is wearing one pair of gloves at a time; therefore, adding more gloves to a coalition already containing a pair of gloves is useless. We represent this mathematically – any set of gloves already containing a pair does not increase its worth when including more gloves. The *grand coalition* $\{a, b, c\}$ contains one pair of gloves, i.e., the pair $\{a, c\}$ or the pair $\{b, c\}$. Therefore, it has a value equal to 1. Note that the value function assigns 1 to the grand coalition and 0 to the empty set. After computing the Shapley values, we find

$$\phi_f(a) = \phi_f(b) = \frac{1}{6} \quad \text{and} \quad \phi_f(c) = \frac{2}{3}.$$

Players a and b get the same Shapley values since they are essentially indistinguishable. Shapley values scores do not detect "redundancy" among a and b. After including one element among a and b, including the other does not yield any advantages. We refer to this similarity among players as *redundancy*, and we say that the Shapley values are unaware of redundancy among players.

2.3 RANKINGS

Shapley values are assignments of importance scores to the single players. Having an array of scores makes ordering the players in the game an immediate task. In Part II, we deal with ranking evaluation metrics and the consistency of importance scores for interpreting time series classification methods.

Rankings are mathematically formalized as elements of symmetric groups.

Definition 2.3.1. *The symmetric group S_N of a set of elements $\mathcal{N} = \{1, \dots, N\}$ is the set of the bijective functions from $\mathcal{N}$ to $\mathcal{N}$ and whose group operation is the function composition.*

In the definition, a fundamental role is played by group theory, a branch of mathematical algebra that is not part of this thesis. We refer to the literature for additional insights [Dur08]. Notably, S_N has size $N!$.

Elements of S_N are indicated with Greek letters, e.g., $\sigma, \nu, \dots$. Permutations, or rankings, are surjective with respect to the elements in $\mathcal{N}$; for each $i \in \mathcal{N}$, $\sigma(i) \in \mathcal{N}$ indicates the element in which the item i is sent by the function σ. Coherently, we write $\sigma \circ \mu$ to indicate the function composition, i.e.,

$$\begin{aligned} \sigma \circ \mu : \mathcal{N} &\mapsto \mathcal{N} \\ i &\mapsto \sigma \circ \mu(i) = \sigma(\mu(i)) \in \mathcal{N}. \end{aligned}$$

The composition of orderings, as well as the one among functions, is *not commutative*, i.e., $\sigma \circ \mu \neq \mu \circ \sigma$.

In some contexts, we are interested in considering only the top k elements of the rankings. We use the notation

$$\sigma_{|k} = (\sigma(1), \dots, \sigma(k))$$

to indicate the ranking of the first k elements and $\text{set}(\sigma_{|k})$ is the set of the first k elements ranked regardless of the ordering.

The cycle decomposition theorem states that each permutation can be rewritten uniquely as the composition of relatively disjoint *cycles*[1].

[1] A cycle $\sigma = (i_1, \dots, i_N)$ is a permutation satisfying $\sigma(i_j) = i_{j+1}$ for $1 \leq j \leq N-1$ and $\sigma(i_N) = i_1$. Two cycles are *disjoint* if they have no common elements.

Theorem 2.3.2 (Cycle decomposition theorem). *Any permutation σ on a finite set $\mathcal{N}$ admits a cycle decomposition, i.e., it can be expressed as a product of a finite number of pairwise disjoint cycles.*

We use the notation of cycles to indicate a *(single) swap*, i.e., a permutation $\sigma = (j\ k) \in S_N$ such that

$$\sigma(i) = \begin{cases} j & \text{if } i = k \\ k & \text{if } i = j \\ i & \text{if } i \neq j, k \end{cases}$$

swapping only the two elements j, k in $\mathcal{N}$, i.e., $\sigma(i) = j$ if $i = k$, $\sigma(i) = k$ if $i = j$ and $\sigma(i) = i$ if $i \neq j, k$.

Part I

SHAPLEY VALUES-BASED UNSUPERVISED FEATURE IMPORTANCE SCORES

3 INTRODUCTION AND RELATED WORK

Born in 1947, Shapley values are a popular concept deriving from Game Theory. A game is the mathematical formalization of the interplay among intelligent items, i.e., the *players*. The aim for the players in a game is to obtain a high *payoff*, i.e., to maximize their gains. Players can be egoistic and compete with each other; their behavior is formalized in the *Competitive Game Theory*. The other possibility is presented when players *cooperate* by forming *coalitions* or teams and aim to achieve high payoffs for the entire team. The two theories are described in entirely different setups. A key concept in Competitive Game Theory is the Nash equilibrium that determines the strategy of the single players; in contrast, in Cooperative Game Theory, a key concept is the attribution to the single players of the teams' payoffs, where Shapley values play a fundamental role. Recently, Game Theory has been applied in computer science. The substantial differences between the two theories are reflected in the types of applications they have. We find Competitive Game Theory mostly in reinforcement learning applications and to represent evolutionary behaviors; Cooperative Game Theory was introduced in the early 2000s through the fundamental concept of Shapley values [Sha+53] for feature selection [CDR05] and interpretable machine learning [LL17].

Shapley values define importance scores for players in a game, hence admitting that they can ordered with respect to their importance. Their definition is flexible and straightforward, particularly adaptable to various scenarios. In computer science, they are mostly known in interpretable machine learning, where they have been broadly used; Lundberg et al. [LL17] was one of the pioneering works that launched them in the community. Shapley values satisfy the need to interpret black-box machine learning models by explaining single predictions[1] by fairly assigning a score to the input's features representing their relevance to the model output. The method was then extended to global explanations in future works. However, interpretable machine learning is not the only field in which Shapley values are applied. We find them for supervised and semi-supervised selection of features [CDR05; Pfa+16] and for the study of single nodes importance in deep and graph neural networks [DM21]. All the fields in which Shapley values have been

[1] SHAP is a local explanation approach.

introduced are characterized by a concrete prediction model and labeled data; nevertheless, it is non-trivial to define the value function[2] on unlabelled data without relying on model predictions. In the state-of-the-art literature, an application of Cooperative Game Theory to the unsupervised world, where data are unlabeled, was still missing. We face here the challenges of reducing the dimension of the unlabeled datasets and detecting possible changes in the correlation structure of unlabeled time series.

In contexts where high redundancy among features might degrade the performance of machine learning or statistical techniques, unsupervised feature selection and concept drift detectors are based on the inherent properties of unlabeled data. In Chapter 4, we create Shapley values-based unsupervised feature importance scores. Mainly focusing on unlabelled tabular data, where columns and rows represent the features and the samples collected, we interpret each component as a realization of random variables. This setup allows us to interpret the features as random variables and define a meaningful value function that measures the correlation in subsets of features. Then, we transfer these importance scores to unlabeled time series. Time series are a special data type where additional information is added, i.e., the time dimension. The same observables are tracked through the passing of seconds, minutes, and days. Predictor models can provide a forecast for future time stamps within an interval of confidence [MJK15], anomaly detectors aim to classify unlabeled observations into anomalous or normal after the training on historical data [Mal+15], concept drift detectors point out the timestamps (or time windows) where significant changes in the dependency among labels and input variables are observed [Lu+18]. Critical and not yet fairly developed are unsupervised concept drift detectors aiming to detect changes only happening in the correlation structure of the univariate input dimensions of the data stream. Chapter 6 introduces the slidSHAP series, a proxy representation of the data stream's time-changing correlation structure. The slidSHAP series represents how the importance scores introduced in Chapter 4 change over time. We use the slidSHAP series as a visualization tool [LBM22] and actively detect concept drifts in the data stream [BLM23b].

Highly specific contexts often require ad-hoc solutions. In Bioinformatics, Shapley values offered a tool to point out the most relevant genes to specific phenotypic traits. Thanks to the microarray games [MPB07], Shapley values' exact computation is efficient in this particular setup[3]. We transfer the problem to the unsuper-

[2] Each game is defined with the set of players $\mathcal{N}$ and the value function $f : \mathcal{P}(\mathcal{N}) \mapsto \mathbb{R}$.

[3] Shapley values' exact computation is a proven NP-hard problem.

vised case, willing to reduce the dimension of collections of gene sets by first assigning “importance scores” to the gene sets independently from the phenotype of interest and finally ranking them given the computed scores. Chapter 5 presents the methodology as a preprocessing to classical Gene Set Enrichment Analysis (GSEA) methods and as a support to obtain reduced dimensions for the collections of gene sets.

The following three chapters in this part are based on published work as cited below:

- Chapter 4, “Unsupervised Features Ranking via Coalitional Game Theory for Categorical Data”, is based on [Bal+22];
- Chapter 5, “Redundancy-aware unsupervised ranking based on game theory: ranking pathways in collections of gene sets”, is based on [Bal+23];
- and Chapter 6, “SLIDSHAPs– sliding Shapley Values for correlation based change detection in time series”, is based on the works [BLM23b] and [LBM22].

We now proceed with an overview of the related work relevant to all chapters.

3.1 RELATED WORK

The conducting strand of Part I are “unsupervised Shapley values” in various contexts and applications. Each topic touched on is characterized by specific and distinguished literature. We distinguish several categories of the related work: (i) Shapley values in machine learning and beyond, (ii) unsupervised feature ranking and selection, (iii) methodologies to handle redundancy collections of gene sets, and (iv) concept drift and change point detection in time series.

We give an overview of each field below.

3.1.1 *Shapley values in machine learning and beyond*

Shapley values have been introduced in machine learning in 2005 by Cohen et al. [CDR05]. The work proposed using the Shapley values as importance scores to select the relevant features of a machine learning prediction model. Successively, the method was transposed to semi-supervised learning [Pfa+16]. In both cases, the value function was introduced as the trained model’s accuracy or generalization error. However, the big breakthrough in computer science was the debut of Shapley values in interpretable machine learning. In 2017, SHAP [LL17] was introduced as a straightforward and interpretable tool to identify the influence of the input's features on a black-box model prediction; aware of the computational complexity of Shapley values, the authors proposed an inherent approximation method based on the computations of the gradient of the model predictions. New insights on Shapley values and their applications continue appearing in the literature [Cat+21]. We additionally find Shapley values explaining anomaly detection methods [TC20; Tak19b] and, recently, Shapley values started finding applications in time series data [Gui+20; Ben+21b; Ant+21; Tak19a]; In 2019, some works [Zhe+19; ZK20] introduced the Shapley values for concept drift detection.

Shapley values sound promising for many applications; the flexible definition explains their extensive use and applications derived in machine learning [Roz+22]. As a downside, there is the computational complexity of the Shapley values, a proven NP-hard problem necessitating the use of appropriate approximations[4]. Several approximations exist [CGT09; BC21] and additionally appear in the state-of-the-art literature [CKL22], but the problem is only partly solved. The issue decelerated the development and application of Shapley values in high-dimensional contexts. However, in some applications [MPB07] and game constructions [5], they can be computed in polynomial time.

3.1.2 *Feature importance scores and unsupervised feature selection approaches*

Feature selection methods look for selecting features relevant to the task at hand, thus reducing the dimensionality of the data that needs to be handled. Many approaches first compute feature importance scores and then rank the features based on them. Feature importance scores are the results of analyzing relationships among features, the class label, and the correlation among variables [VE14a]; Shapley values have been used as feature importance scores [Pfa+16; CDR05] for supervised selection. However, these scores are unaware of correlations among variables [She+17], thus leading to a necessary integration of a redundancy awareness concept.

In recent years, unsupervised feature selection methods have raised strong interest in the community [SCM20; WTL15; Li+17]. As a representative sample within the vast number of unsupervised feature selection methods, we find UDFS [Yan+11] that creates

[4] The value function f needs to be an exponential number of times as a function of the number of players.

[5] Examples are the airport games, the SOUG games, and the microarray games.

pseudo-labels to perform a selection of features in unlabelled data; MCFS and NDFS [CZH10; Li+12] that focus on keeping the clustering structure; LS [HCN05] that selects features by their local preserving power; and PFA [Lu+07], that tries to eliminate the downside of PCA while keeping the information within the data. Most of these algorithms tend to select features as a by-product of retaining a clustering structure in the data. Finally, PFA [Zhu+19] is meant to select only non-redundant variables using a new definition of distance in the k-nearest neighbors approach.

3.1.3 *Collections of gene sets- challenges and methods*

Collections of gene sets are arbitrarily derived from the biological function of the gene sets, leading to high-dimensional and overlapping families of sets [Lib+15]; these collections are usually available on public databases. Two main challenges are evident from the setup: first of all, often, there is no apparent agreement among the various online databases [Sto+18], and secondly, the large sizes of the collections hinder the interpretability and maneuverability in their use, particularly for GSEA. In the state-of-the-art literature, several proposals tried to solve the mentioned challenges. Some visual tools mitigate the lack of interpretability by visualizing the redundancy among the gene sets and among the collections, while other tools propose modifying the gene sets and merging them to obtain a higher agreement among databases and non-redundant, single and unified collections of gene sets [Bel+15; Ier+08; Dod+12]. Although practically useful, visual tools do not solve the fundamental issues, and modifying the gene sets potentially contrasts with the biological meaning of the collections themselves.

The collections of gene sets are fundamental for GSEA. We recall Enrich [Che+13; Kul+16; Xie+21] among the enrichment analysis tools. The GSEA methods commonly aim to assess potential over- or under-representation in biological contexts of specific genes. Methods such as [Sub+05; Mat+18] rely on statistical tests, e.g., the Fisher exact test, requiring correction for multiple hypothesis testing; given the high dimensions of the collections of gene sets, the number of statistically significant gene sets is shrunk to the bottom, thus cutting out potential important gene sets because of the low threshold. Notably, using Bonferroni correction [BH95], the level of significance α is inverse-proportional to the size of the collections, e.g., $\frac{\alpha}{n}$, where n is the number of gene sets in the collection.

3.1.4 *Concept drift detection*

Time series data might suffer from distributional changes that can hinder prediction models, anomaly detectors, and other machine learning or statistical methods trained on historical data. We find two types of drifts in time series data, depending on whether they involve the distribution of the labels and the input variables or only the univariate dimensions. The second type can also be observed in unlabeled data.

Concept drift and change point detectors are specific to one of these scenarios. Common examples of supervised concept drift detectors are offered by Halstead et al. [Hal+21] and Gama et al. [Gam+04], while an overview of unsupervised concept drift detector methods is given in [Gem+20]. Concept drift detectors often rely on a two-step procedure, i.e., (1) they create a new time series representation, then (2) they detect changes over the representations; however, the first step is optional, as the detection can often be performed on the original data. Among representation methods, we find [BG07; CMO16; Cos+17]; other approaches are based on meta-information vectors [Hal+21]. Typically, the detection step is performed by comparing the current data distribution with a reference historical data buffer, where some approaches measure the distributional discrepancy between data in different time periods [DP11; Das+06]. Qahtan et al. [Qah+15] propose to track covariance changes in a transformed artificial low-dimensional space obtained by applying PCA on the time series. To overcome the simplificist approach in [Qah+15], [AK18] uses mean and covariance to represent the concepts in multivariate data streams. However, the state-of-the-art literature still did not grab the complexity of all correlations among univariate dimensions; thus, the correlations among sets of univariate dimensions are still potentially meaningful to detect changes in unlabeled data streams.

4 UNSUPERVISED FEATURES RANKING VIA COALITIONAL GAME THEORY FOR CATEGORICAL DATA

Many algorithms suffer from the curse of dimensionality. As a result, reducing the features in the data to a smaller set is often of great utility in disparate contexts, e.g., to increase the interpretability or reduce the runtime of the algorithms. Feature selection aims to reduce the number of features, often using feature importance scores to quantify the relevance of single features to the task. Shapley values helped in supervised contexts, where the importance of the variables could be directly quantified for specific predictions, and the scores were directly applied to select "important" features [Pfa+16; CDR05]; the value function was defined as the accuracy or the generalization error of the trained model. Unfortunately, the approaches based on Shapley values are limited to labeled data where the shared information among predictors and labels can be directly quantified [VE14b]. Unfortunately, not all real-world data are labeled; when labels are unavailable, obtaining them is either impossible or extremely costly. For the unsupervised case, an appropriate value function to plug in the Shapley values can only rely on the probability distribution of variables and the quantification of their interactions.

We propose the first synergy between Shapley values and unlabeled data and use Shapley values as feature importance scores to rank the feature with respect to the information they contain. The scores directly represent the contribution of single features in explaining the datasets' structures. We find the basis in Information Theory, from which we take the notion of *information* conveyed by a feature. The use of the total correlation as a value function allows for quantifying higher-order interactions among features, while the Shapley values allow for aggregating these measures as unique importance scores. Additionally, our feature importance scores include a notion of redundancy awareness, making them a tool to achieve redundancy-free feature selection. The previous literature, mainly investigating anomaly detection and clusters, either failed to address the redundancy-elimination issue [She+17] or could not return importance scores for single features [Zhu+19].

Most unsupervised selection methods focus on keeping the original clustering structure of the data, e.g., MCFS and NDFS [CZH10; Li+12], or focus on selecting features dependently on their local preserving power [HCN05]. Some methods use basic, well-known approaches to study the unlabelled data and obtain an unsupervised selection of features; PFA [Lu+07] is constructed based on the principal component analysis PCA, while PFA [Zhu+19] on the definition of the k-nearest neighbors, with an additional constraint based on the correlation among data points. Finally, a different current of works focuses on first creating pseudo-labels, i.e., assigning a label to each data point; adding pseudo-labels transforms an unlabelled dataset into a labeled one, thus allowing the use of the most commonly used supervised feature selection approaches. However, none of the previously proposed unsupervised feature selection methods offer direct scores for evaluating the value of the single features. Furthermore, they are either too simple to grab the whole data structure (e.g., pairwise correlations) or too complicated to be directly interpretable (e.g., pseudo-labels).

Conversely, our total correlation-based Shapley values offer feature importance scores that are easily accessible for further analysis. As already remarked, however, Shapley values are well known to be computationally expensive. Their exact computation requires 2^N evaluations of a value function where N is the number of players. The computational complexity of these scores makes their application unfeasible as soon as the number of players increases. Several approximations appeared in the literature, and an easy solution to reduce the computational runtime of Shapley values is represented by Castro et al. [CGT09] approximation, i.e., the most common Shapley values' approximation that does not rely on additional assumptions on the players. In the next chapter, we will use it to accelerate the computation of Shapley values.

In conclusion, our main contributions can be summarized in the following points:

1. we state the possibility of unconventionally using Shapley values as unsupervised feature scores;
2. we derive an unsupervised feature ranking and selection method, lowering the redundancy among features retained while maximizing the coverage of information originally contained in the data.

4.1 FEATURE IMPORTANCE MEASURES

Consider a N-dimensional unlabeled dataset containing D instances. We interpret each dimension as the realization set of a random variable, refer to the set of variables as $\mathcal{N} = \{X_1, \ldots, X_N\}$ and to each dimension X_i as ith feature or variable. Supervised feature selection methods often assign internally to subsets of features an importance score and output that subset that maximizes the mentioned score. We propose to rank features considering their average contribution to all the possible subsets of features in our unsupervised setting. The higher the average contribution of a feature is, the more convenient it is to keep it within the selected features. Additionally, we will introduce redundancy awareness in these scores.

Given a function f that assigns a value to each subset of features, it is not trivial to assess the *importance* of single features as each feature belongs to 2^{N-1} subsets of features. Shapley values offer summary scores representing the values of the features in the task at hand; we recall that, in unsupervised contexts, the usefulness of features is often related to the correlation structure or clustering properties of the unlabeled data points. Therefore, throughout the chapter, we stick to a value function f that captures the maximal *information* contained in the data. We then compute the Shapley values feature importance scores and obtain a ranking prioritizing features highly correlated with the rest of the dataset.

4.1.1 *Feature importance scores*

We obtain feature importance scores using the fundamentals of Coalitional Game Theory. As explained in Section 2.2, each game is fully represented by the set of players $\mathcal{N}$ and a set function f. Each subset $\mathcal{A} \subseteq \mathcal{N}$ is mapped to $f(\mathcal{A}) \in \mathbb{R}$ where f is the value function; we assume f satisfies the usual monotonicity and non-negativity properties.

Working with unlabelled data, we can not rely on ground truth labels. Hence, we define a value function relying on intrinsic properties of the dataset; we opt for a value function measuring the independence of the features in $\mathcal{A} \subseteq \mathcal{N}$. One possible initialization for f is the "total correlation", a concept deriving from Information Theory.

Definition 4.1.1. *The* total correlation C *of a set of variables* $\mathcal{A} \subseteq \mathcal{N}$ *is defined as*

$$C(\mathcal{A}) = \sum_{X \in \mathcal{A}} H(X) - H(\mathcal{A}). \tag{4.1}$$

$H(\mathcal{A})$ is the Shannon entropy of the subset of discrete random variables $\mathcal{A}$, i.e.,

$$H(\mathcal{A}) = -\sum_{\vec{x} \in \mathcal{A}} \mathbb{P}_{\mathcal{A}}(\vec{x}) \log \mathbb{P}_{\mathcal{A}}(\vec{x}) \tag{4.2}$$

where $\mathbb{P}_{\mathcal{A}}()$ is the joint probability mass function of thr random vriables in $\mathcal{A}$. Finally, $H(X)$ is the Shannon entropy of a unique random variable X, i.e.,

$$H(X) = -\sum_{x \in X} \mathbb{P}_X(x) \log \mathbb{P}_X(x). \tag{4.3}$$

We choose the total correlation as it satisfies the monotonicity and non-negativity properties, and it can be easily extended such that it satisfies $f(\emptyset) = 0$; furthermore, it has an intuitive meaning. Shannon entropy [Cov99] measures the uncertainty contained in a random variable X considering how uniform data are distributed: its value is close to zero when its probability mass function $\mathbb{P}_X$ is highly skewed while, as the distribution approaches a uniform distribution, its value increases. Moreover, the Shannon entropy is a monotone non-negative function and can be extended such that $H(\emptyset) = 0$. We assume all features in $\mathcal{N}$ are discrete as the extension of Shannon entropy to continuous variables is not monotone [LR78]. As a consequence of Shannon entropy's properties, the total correlation $C(\mathcal{A})$ is close to zero if the variables in $\mathcal{A}$ are independent, and it increases when they are correlated. To study the impact of adding a feature Y to $\mathcal{A} \subseteq \mathcal{N}$, we compute the value function of the incremented subset $f(\mathcal{A} \cup Y)$ and compare it with $f(\mathcal{A})$. The difference

$$f(\mathcal{A} \cup Y) - f(\mathcal{A}) = H(\mathcal{A}) + H(Y) - H(\mathcal{A} \cup Y)$$

is non-negative and measures how much $\mathcal{A}$ and Y are correlated. The quantity $H(\mathcal{A}) + H(Y) - H(\mathcal{A} \cup Y)$ is the marginal contribution of Y to $\mathcal{A}$ as defined in Chapter 2; if $\mathcal{A}$ and Y are independent, then the marginal contribution of Y to $\mathcal{A}$ equals zero. Vice versa, the marginal contribution grows the stronger the correlation between Y and $\mathcal{A}$ is. As importance score, we assign to X_i the average of its marginal contributions, and we indicate it as $\phi(X_i)$, i.e.,

$$\phi(X_i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus X_i} \frac{1}{N\binom{N-1}{|\mathcal{A}|}} [H(\mathcal{A}) + H(X_i) - H(\mathcal{A} \cup X_i)] \tag{4.4}$$

corresponding to the *Shapley value* of the player X_i in the game $(\mathcal{N}, f)$ when f is the total correlation (cf. Equation (2.1)); we drop the notation of the Shapley values with f for readability. We underline that Shapley values represent a fair (in the sense of *individual fairness*) assignment of resources to players based on their contributions to the game. We use the scores $\phi(X_i)$ to rank the features in the dataset $\mathcal{N}$; however, Shapley values do not consider redundancies and, as a consequence of the individual fairness, linearly dependent features obtain equal Shapley values.

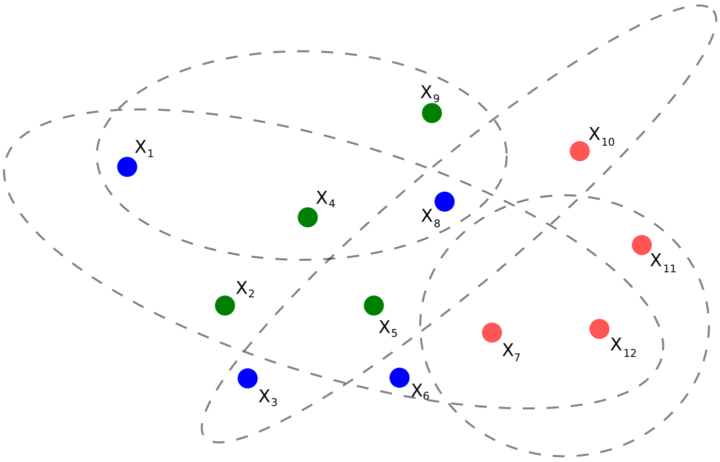


Figure 4.1: SHAPLEY VALUES consider interactions within all possible players X_is. Here, each bullet represents a player, and the same coloring highlights players similarly characterized.

4.1.2 *Importance scores of low correlated features*

As we have already underlined, correlated features are characterized by similar Shapley values. We use a dataset with three sets of correlated features, color-coded in Figure 4.1. The aim is to select features from subsets with different colors; however, the highest Shapley values may be obtained by correlated features, i.e., features with the same colors. Before addressing the problem of redundancy-awareness inclusion in Shapley values, we show that the Shapley values rank features that are not correlated with the rest of the dataset in low positions.

Theorem 4.1.2. *Given a subset of features $\mathcal{B} \subset \mathcal{N}$ that satisfies the following properties*

1. *for all $X_j \notin \mathcal{B}$ and for all $\mathcal{A} \subseteq \mathcal{N} \setminus \{X_j\}$,*

$$H(\mathcal{A}) + H(X_j) = H(\mathcal{A} \cup X_j)$$

2. *for all $X_i \in \mathcal{B}$ and for all $\mathcal{A} \subseteq \mathcal{N} \setminus \{X_i\}$,*

$$H(\mathcal{A}) + H(X_i) \geq H(X_i \cup \mathcal{A})$$

then $\phi(X_i) \geq \phi(X_j)$ for all $X_i \in \mathcal{B}$ and $X_j \notin \mathcal{B}$.

Proof. From 1 we know that, since the marginal contribution of $X_j \notin \mathcal{B}$ to any $\mathcal{A} \subseteq \mathcal{N} \setminus \{X_j\}$ is equal to zero,

$$\phi(X_j) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus \{X_j\}} \frac{1}{N\binom{N-1}{|\mathcal{A}|}} \cdot 0 = 0.$$

Algorithm 1 Pseudo-code. The svfs algorithm.

```
procedure svfs(set of features $\mathcal{N}$, parameter $\epsilon$)
    $\mathcal{S} = \emptyset$
    while $\mathcal{N} \neq \emptyset$ do                ▷ till we did not empty $\mathcal{N}$
        while $X \in \mathcal{N}$ do
            if $H(X) + H(\mathcal{S}) - H(\mathcal{S}, X) > \epsilon$ then
                $\mathcal{N} = \mathcal{N} \setminus X$
            end if
            $\mathcal{S} = \mathcal{S} \cup \arg\max_{X \in \mathcal{N}} \{\phi(X)\}$
            $\mathcal{N} = \mathcal{N} \setminus \mathcal{S}$
        end while
    end while
end procedure
$\mathcal{S}$          ▷ the algorithm returns the set of selected features $\mathcal{S}$
```

Furthermore, for any $X_i \in \mathcal{N}$ and $\mathcal{A} \subseteq \mathcal{N}$, we know that $H(\mathcal{A} \cup X_i) \leq H(\mathcal{A}) + H(X_i)$ from Shannon entropy's properties [Cov99]. Hence, all marginal contributions are non-negative. Hence, $\phi(X_i) \geq 0 = \phi(X_j)$ for all $X_i \in \mathcal{B}$ and $X_j \notin \mathcal{B}$.

This concludes the proof. □

Thus, Shapley values are non-negative and equal to zero if and only if the feature is non-correlated with any subset of features when using total correlation as a value function. Moreover, features highly correlated with other subsets of features get high Shapley values.

4.2 REDUNDANCY REMOVAL

As a second step, we address the challenge of adding redundancy awareness to Shapley values. For this purpose, we develop a pruning criterion based on the total correlation and greedily rank features to get a redundancy-free ranking while still looking for features with high Shapley values. Based on this ranking, our feature selection method selects the variables ranked first by Shapley values, which show little dependencies.

We propose two algorithms. The Shapley value Feature Selection (svfs) needs a parameter ϵ representing the correlation among features that we are willing to accept; hence, svfs requires some expert knowledge of the dataset to specify the parameter ϵ in a suitable interval. The Shapley value Feature Ranking (svfr) works automatically with an included notion of redundancy. The two algorithms lead to consistent results as shown in Section 4.4.5. Both algorithms select the highest-ranked feature at each step among the ones left.

Algorithm 2 Pseudo-code. The svfr algorithm.

```
procedure svfr($\mathcal{N}$)
    $\mathcal{S} \leftarrow \arg\max_{X \in \mathcal{N}}\{\phi(X)\}$
    ordered $\leftarrow$ [ ]
    ordered[0] $\leftarrow \arg\max_{X \in \mathcal{N}}\{\phi(X)\}$
    $j = 1$
    $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{S}$
    while $\mathcal{N} \neq \emptyset$ && $j < N$ do        ▷ till we did not empty $\mathcal{N}$
        for $X \in \mathcal{N}$ do
            $\mathrm{rk}(X) = \phi(X) - H(X) - H(\mathcal{S}) + H(\mathcal{S}, X)$
        end for
        ordered[$j$] $\leftarrow \arg\max_{X \in \mathcal{N}}\{\mathrm{rk}(X)\}$
        $\mathcal{S} \leftarrow \mathcal{S} \cup \arg\max_{X \in \mathcal{N}}\{\mathrm{rk}(X)\}$
        $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{S}$
        $j++$
    end while
end procedure
ordered  ▷ the algorithm returns the array of ordered features
```

In both algorithms, svfr and svfs, we use a total correlation-based pruning criterion; $H(\mathcal{A}) + H(X) - H(\mathcal{A} \cup X)$ represents the strength of the correlation among a random variable X and a set of random variables $\mathcal{A}$. It is equal to zero if and only if X and $\mathcal{A}$ are independent.

svfs's inputs are the set of unordered features $\mathcal{N}$ and the parameter $\epsilon > 0$; the parameter ϵ, representing the maximum correlation that we are willing to accept within the set of selected features, plays the role of a stopping criterion. Whenever ϵ is high, we end up with the ordering given by Shapley values alone; instead, for $\epsilon \approx 0$ the criterion can lead to the selection of the only features that are uncorrelated with the first one. The optimal range of ϵ highly depends on the dataset. We show that svfs is robust with respect to the choice of ϵ. At each iteration, svfs excludes from the ranking the features Xs that are correlated with the already ranked features $\mathcal{S} \subseteq \mathcal{N}$ more than ϵ, i.e.,

$$H(X) + H(\mathcal{S}) - H(\mathcal{S}, X) > \epsilon,$$

computes the Shapley values of all remaining features X and adds to $\mathcal{S}$ the feature whose Shapley value is the highest. When no features are left, it stops and returns $\mathcal{S}$.

svfr takes as an input $\mathcal{N}$ and outputs a feature ranking without additional parameters. The ranking is aware of correlations as each of the Shapley values $\phi(X)$ is penalized using the correlation measure $H(X) + H(\mathcal{S}) - H(X \cup \mathcal{S})$ where $\mathcal{S}$ is the set of already ranked features, and X_i is a new feature to be ranked. This algorithm provides a complete ranking of features and can be prematurely stopped, including an upper bound of features we are willing to rank. The absence of the additional parameter ϵ is the main advantage of SVFR over SVFS. The pseudo-codes of the algorithms are represented in Algorithm 1 and Algorithm 2 respectively.

4.3 SCALABLE ALGORITHMS

The size of $\mathcal{P}(\mathcal{N})$ being exponential in N, computing Shapley values involves 2^N evaluations of the value function. We use approximated Shapley values to obtain scalable versions of SVFR and SVFS. We implement three versions of the algorithms that differ only in the computations of Shapley values used:

- the "full algorithm" uses the full computation of the Shapley values;
- the "bounded algorithm" considers only subsets up to size k fixed to compute the Shapley values;
- and the "sampled algorithm" uses the approximation proposed by Castro et al. [CGT09] based on n random sampled subsets of features.

The time complexity for the sampled algorithm is $\mathcal{O}(D \cdot n)$, for the bounded algorithm is $\mathcal{O}(D \cdot N^k)$ while for the full algorithm is $\mathcal{O}(D \cdot 2^N)$ where N is the number of features and D the number of samples in the dataset.

4.4 EXPERIMENTS

We show that our feature ranking method outperforms competing representative feature selection methods in terms of redundancy reduction. Metrics such as NMI, ACC, and redundancy rate are often used in the previous literature to evaluate unsupervised feature selection methods. NMI and ACC focus on the cluster structure in the data; as SVFS and SVFR are not optimizing for retaining the clustering of the data, we compare our methods with the competing methods using the redundancy rate. The redundancy rate of $\mathcal{S} \subseteq \mathcal{N}$ is defined in terms of pairwise Pearson correlations, i.e.,

$$\mathrm{Red}(\mathcal{S}) = \frac{1}{2m(m-1)} \sum_{X,Y \in \mathcal{S}, X \neq Y} \rho_{X,Y} \tag{4.5}$$

	versatile quality notion	feature ordering	iterative selection	redundancy awareness	higher-order interactions
UDFS	✗	✓	✗	✓	✓
MCFS	✗	✓	✗	✓	✗
NDFS	✗	✓	✗	✓	✗
SPEC	✓	✓	✗	✗	✗
LS	✓	✓	✗	✗	✗
PFA	✓	✗	✗	✓	✗
FSFC	✗	✗	✓	✓	✗
our	✓	✓	✓	✓	✓

Table 4.1: Summary table. Competing methods and our proposed method.

where $\rho_{X,Y} \in [0,1]$ is the Pearson correlation of features X and Y. It represents the averaged correlation among the pairs of features in $\mathcal{S}$. It varies in the interval $[0,1]$: a redundancy rate close to 1 shows that many selected features in $\mathcal{S}$ are strongly correlated. At the same time, a value close to zero indicates that $\mathcal{S}$ contains little redundancy. In the experiments, we use the *redundancy rate* as evaluation criteria, re-scaling it to the interval $[0,100]$ via the maximum pair-wise correlation to facilitate the comparison among different datasets.

4.4.1 *Datasets and competing methods*

We show comparison against significant unsupervised feature selection approaches, i.e., SPEC [ZL07], MCFS [CZH10], UDFS [Yan+11], NDFS [Li+12], PFA [Lu+07], LS [HCN05] and FSFC [Zhu+19]. Table 4.1 illustrates a summary of the properties of the various methods analyzed in comparison with our approach.

We use various synthetic and publicly available datasets:

- the Breast Cancer dataset [ZS88],
- the Big Five Personalities Test dataset[1],
- and the FIFA dataset[2].

Generally, we consider subsets of the full dataset to apply the full versions of the algorithms and investigate the performance of the approximations of svfr and svfs at the end of the section.

[1] Available at `kaggle.com/BigFivePersonalitiesTestDatset`

[2] Available at `kaggle.com/FIFA21dataset`

	Breast Cancer	Big Five balanced	Big Five unbalanced	FIFA	Synthetic
NDFS	36.30	22.11	20.75	18.97	1.49
MCFS	20.26	23.59	18.79	20.63	3.74
UDFS	33.59	28.13	35.18	57.73	4.06
SPEC	13.89	39.09	21.46	42.14	29.4
LS	7.05	28.83	58.25	48.28	100.00
PFA	5.10	23.22	34.28	57.42	35.84
PFA	8.74	22.64	20.99	36.45	2.12
svfr	6.68	15.65	18.02	14.79	1.51

Table 4.2: Redundancy rate evaluation. Redundancy rates of the three selected features using the competing algorithms and svfr on different datasets. The lowest rates are highlighted .

The datasets are all categorical or discrete. The first 50 features in the Big Five dataset are the categorical answers to the personality test's questions and are divided into 5 personality traits (10 questions for each personality trait). We select questions from different personalities to apply the full algorithm and restrict to 10000 instances. We consider the 5000 highest-rated players for the FIFA dataset by the overall attribute.

4.4.2 *Redundancy awareness*

We compare the feature selection results of our algorithm against the competitors by evaluating the redundancy rate in Table 4.2. For the FIFA dataset, we select 15 features from the entire data that characterize the "agility", "attacking" and "defending" skills of the football players; we keep the whole datasets for Breast Cancer and synthetic data. In the case of the Big Five Personalities Test dataset, we select respectively 5 questions from three different personality traits for the balanced dataset and 9 features from one trait, and 3 from other two personality traits in the case of the unbalanced dataset. To avoid bias towards the random selection of personality traits and features in the Big Five data, we average the redundancy rate over 30 trials on randomly selected personalities and variables both in the case of the balanced and unbalanced setup.

In each column, bold characters highlight the lowest redundancy rate. We use svfr to rank the features and select the three highest-ranked features. Consequently, we specified the parameters of the

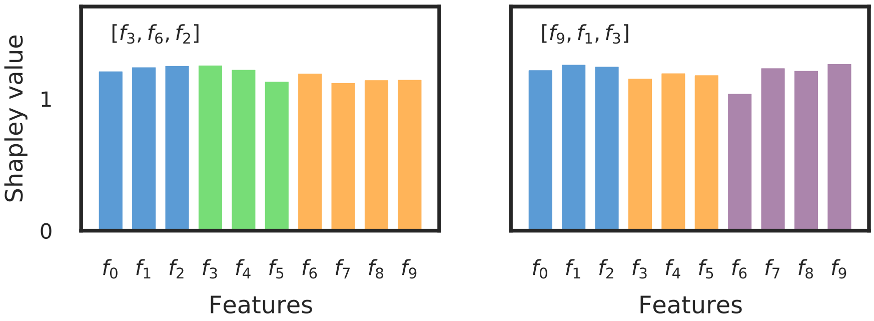


Figure 4.2: BARPLOT OF SHAPLEY VALUES AND FEATURE SELECTIONS. Results given by SVFS ($\epsilon = 0.3$) for the Big Five dataset restricted to 10 features. Different personality traits are color-coded.

competing methods in order to get a selection of features as close to three features as possible. We use for FCFS, we set $k = 4$ for the BC dataset; $k = 8$ for the FIFA dataset; $k = 8$ for the synthetic data; for Big Five dataset, we use different k at each re-run such that the number of selected variables varies between 2 and 5 and then we average the redundancy rates. Finally, for NDFS, MCFS, UDFS, and LS, we used $k = 5$ (k being the number of clusters in the data); for the other competitors, we specified the number of features to be selected.

Table 4.2 illustrates that SVFR outperforms the competing methods in nearly all the cases. In particular, while SVFR achieves low redundancy rates in all datasets, the competing algorithms show big differences in performance in the various datasets. On the Breast Cancer data and the synthetic dataset, respectively, PFA and NDFS slightly outperform SVFR. However, they do not keep an average low redundancy rate on the other datasets. For reproducibility, we make the code on GitHub publicly available[3].

4.4.3 *Relevance of unsupervised feature selection and effectiveness*

In Figure 4.2, each plot corresponds to a subset of features of the Big Five dataset, i.e., 10 features selected from three different personality traits. Running SVFS with $\epsilon = 0.3$ we detect correlated features and avoid selecting them together as shown in the plots. Using the scaled versions of our algorithms from Section 4.3 we can extend the approach towards the complete Big Five dataset.

Figure 4.3 represents the Shapley values of features in a 12 dimensional synthetic dataset where subsets of correlated features are color-coded. We measure the ability of the algorithm to select

[3] Code available at `chiarabales/unsupervised_sv`

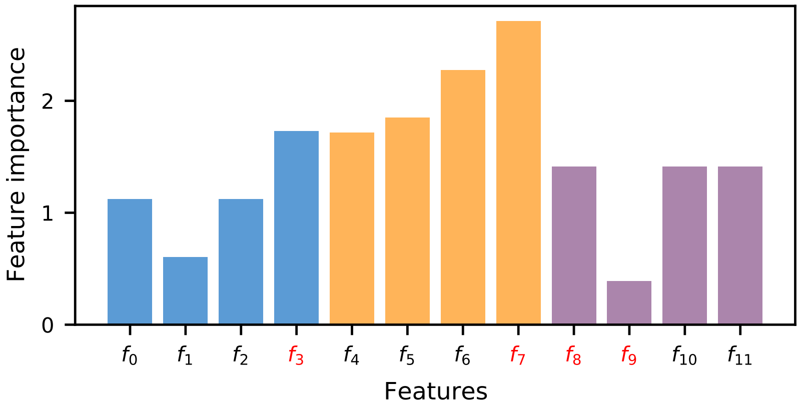


Figure 4.3: Unsupervised Shapley values-based feature importance scores. Correlated subsets of features are color-coded; features selected by SVFS with $\epsilon = 1$ are in red color.

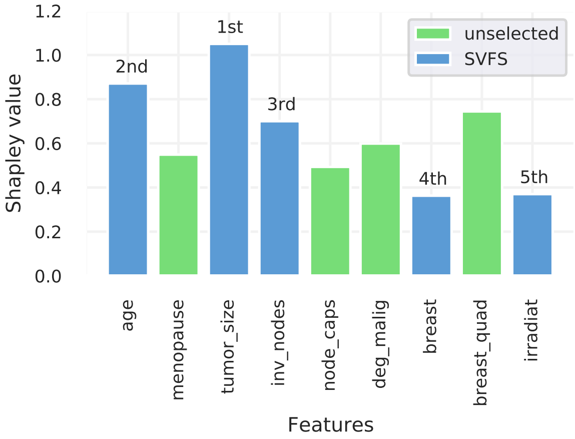


Figure 4.4: Shapley values for Breast cancer data. in green, the ordering of features' selection by SVFS when $\epsilon = 0.5$.

features from different subsets of correlated features; SVFS selects one feature from each subset of correlated features. In particular, when $\epsilon = 1$, SVFS achieves this goal by selecting $\{f_8, f_7, f_3\}$ while the ranking given by the Shapley values alone is $\{f_8, f_{10}, f_{11}\}$ which belong to the same subset of correlated features. This nicely underlines the inability of Shapley values to detect correlations and the necessity of integrating correlation awareness to perform a feature selection.

4.4.4 *Interpretation of feature ranking*

We apply SVFS when $\epsilon = 0.5$ to the Breast Cancer dataset. In Figure 4.4, the resulting Shapley values and the ordering of selected

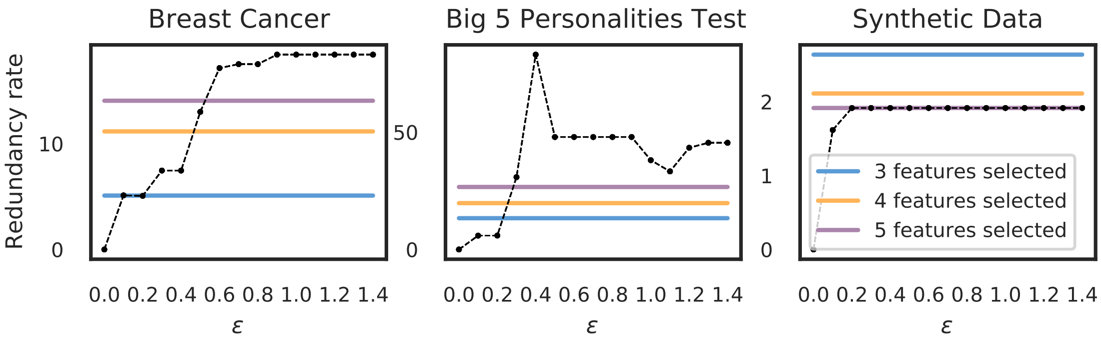


Figure 4.5: Redundancy rates of the selected features' sets as a function of ϵ for SVFS (bullets points connected by the dashed line) and for $3, 4$, and 5 selected features when using SVFR.

features are displayed. The selection resulting from SVFS shows a low redundancy rate while the selected features (e.g., the size of the tumor, age, and the number of involved lymph nodes) are clearly in line with domain knowledge on risk factors for disease progression (label). Furthermore, the comparison with the ranking without redundancy awareness nicely highlights the importance of our approach to avoid redundancies when possible.

4.4.5 *Comparison among the proposed algorithms*

In Figure 4.5, we plot a comparison among SVFS and SVFR with respect to the redundancy rate on three datasets with different values of ϵ. As benchmarks, we use for SVFR the selection of 3, 4, and 5 features, respectively, while for SVFS, ϵ varies in the interval $[0, 1.4]$ with steps of size 0.1.

Using the number of features as a stopping criterion in SVFR would produce consistent results for SVFS: as an example, using the breast cancer data the ranking given by SVFR, i.e., $[2, 0, 4, 6, 8, 5, 1, 3]$, is consistent with the selection given by SVFS respectively using $\epsilon = 0.2$ and $\epsilon = 0.6$, i.e., $[2, 0, 8]$ and $[2, 0, 3, 8, 6]$.

Table 4.3 shows a full comparison among the SVFR and SVFS on three representative datasets. We recommend applying SVFS when no previous knowledge of the data is available, and it is hard to establish an optimal range for ϵ. Vice versa, one could apply SVFR when the expertise in the dataset domain allows determining a reasonable number of features as a stopping criterion. Observing the ranking given can provide insights to the non-expert on which features to keep and which can be discarded for further analysis.

	ϵ	Big Five	Synthetic Data	Breast Cancer
SVFS	0.2	[11, 0, 5]	[8, 7, 0]	[2, 0, 8]
	0.3	[11, 0, 10]	[8, 7, 2]	[2, 0, 4, 6]
	0.4	[11, 0, 14]	[8, 7, 3]	[2, 0, 4, 8, 6]
	0.5	[11, 0, 14, 9]	[8, 7, 3]	[2, 0, 3, 8, 6]
	0.6	[11, 0, 14, 5]	[8, 7, 3]	[2, 0, 3, 8, 6]
	0.7	[11, 0, 14, 13]	[8, 7, 3]	[2, 0, 3, 4, 8, 6]
	0.8	[11, 0, 14, 13]	[8, 7, 3, 0]	[2, 0, 3, 4, 5, 8, 6]
SVFR	-	[11, 0, 5, 10, 12, 8, 6, 2]	[8, 7, 3, 0, 6, 5, 2, 10]	[2, 0, 4, 6, 8, 5, 1, 3]

Table 4.3: RANKINGS BY SVFS. For various ϵ and first 8 ranked features by SVFR. Features are color-coded in order to simplify the visualization.

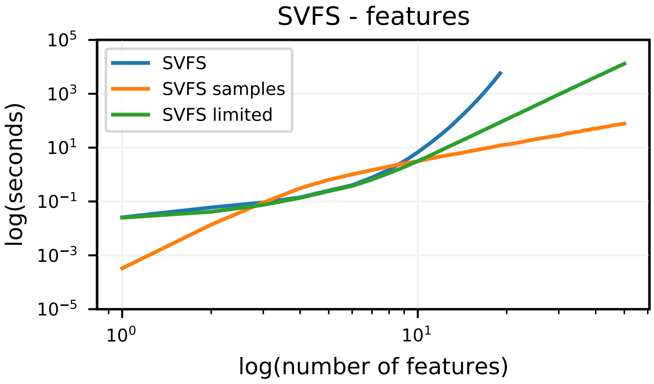


Figure 4.6: RUNTIME. Log-log plots of the run-time as a function of the number of features for the approximated and full SVFS ($\epsilon = 0.5$, $D = 1000$). The full SVFS is stopped with 20 features.

4.4.6 *Run-time analysis*

Due to the full computation of Shapley values, the run-time of SVFR and SVFS increases exponentially with the number of features as shown by Figure 4.6. Using the approximated algorithms, this growth turns out to be slower. In particular, when using the sampled algorithm, the run-time increases only linearly with the number of features while the growth of the bounded algorithm's run-time is polynomial in the number of features. In the additional material, we show the log-log plot of the run-time for an increased number of samples in the dataset. For each algorithm, we use random subsets of the Big Five dataset and average over 10 trails.

We further compare the rankings of the approximated and full algorithms using the *recall@k* metric, interpreting rankings of the full version of SVFR as ground truth. We use the Big Five dataset,

	algorithm	$k = 1$	$k = 3$	$k = 5$
BIG5	random	0.04	0.19	0.33
	sampled	0.04	0.37	0.49
	bounded	0.08	0.56	0.55
FIFA	random	0.06	0.24	0.35
	sampled	0.00	0.33	0.40
	bounded	1.00	0.67	0.80

Table 4.4: recall@k. For $k \in \{1, 3, 5\}$, recall@k of a random ranking and the rankings given by svfr using the sampled and bounded algorithms to the full svfr ranking for 15 features randomly chosen from the FIFA and Big Five datasets. The best approximation is highlighted in each column.

randomly selecting 5 questions from 3 different personalities and average the scores over 100 trails (see Table 4.4). Overall, the results for the approximated algorithms outperform random ordering – but still deviate often from the full versions. It is worth noting that the bounded algorithm using subsets up to size 5 performs better than the sampled version.

5 REDUNDANCY-AWARE UNSUPERVISED RANKING BASED ON GAME THEORY: RANKING PATHWAYS IN COLLECTIONS OF GENE SETS

In the recent years, we have witnessed an increasing awareness of the importance of understanding data and obtaining interpretable models in the Machine Learning and Bioinformatics communities. It is often argued that techniques to reduce the dimensionality of data could increase the maneuverability and, consequently, the understanding of large data. Notably, the Coalition Game Theory framework led to prosperous developments, where Shapley values have been extended to feature selection (cf. Chapter 4) and explainable machine learning [LL17]. In Bioinformatics, importance scores based on Shapley values have been adapted to study the interaction among genetic and phenotypic characteristics for gene sets prioritization analysis [Luc+10; Mor+08]. Surprisingly, these applications are privileged in terms of the usual issue of computational complexity. In particular, the introduction of microarray games [MPB07] reduces the computational challenges of exact Shapley values' computation to polynomial time; the same holds whenever the assumption that the game can be written using only binary relationships is satisfied, e.g., "anomalous" vs. "normal", "in" vs. "out" among others. Using microarray games, Sun et al. [Sun+20] implemented Shapley values to rank genes by their relevance concerning the individual genes' synergistic influence in a gene-to-gene interaction network.

We focus on collections of gene sets, aiming to reduce their size in an unsupervised fashion; given the multiple label goals characterizing their application, they can be assimilated into an unlabeled setup. In Genetics, gene sets, or *pathways*, are grouped in collections concerning their biological function, leading to the birth of several collections of gene sets' databases. These high-dimensional, overlapping, and redundant families of sets [Lib+15] preclude immediate maneuverability and a straightforward interpretation of the biological meaning and the bioinformatic technologies applied to them. The overlaps among pathways in collections of gene sets are a well-known problem: biologically, genes participate in numerous pathways representing various biological processes. On the one hand, techniques exist to aggregate overlapping gene sets to create larger pathways. Stoney et al. [Sto+18] point out the lack of agreement among the various collections of gene sets' databases and proposed methods to aggregate them, maximizing the gene coverage and not altering the gene sets themselves. Sadly, this method does not apply to inter-database applications, and including information on the interactions among gene sets within the single collections has yet to be tackled in the literature. On the other hand, some recent solutions proposed tools for visualizing redundancy among pathways, merging gene sets based on their similarity, and integrating them into a non-redundant single and unified pathway [Bel+15; Ier+08; Dod+12]. While aggregation methods could partly solve the problem of the large size of the collections, modifying biological pathways is hardly justifiable in this biological context; hence, the results that have been proposed so far have proved to be insufficient.

We aim to order and select gene sets from the collections of gene sets independently from any prediction goals, i.e., in an unsupervised fashion. We use Shapley values, thus relying on theoretical properties of fair allocation of resources, and propose a method to rank sets within a family of sets based on the distribution of the singletons and their sizes. We obtain sets' importance scores by implementing microarray games without incurring the typical exponential computational complexity. Moreover, we address the challenge of constructing redundancy-aware rankings where, in our case, redundancy is a quantity proportional to the size of intersections among the sets in the collections. We use the obtained rankings to reduce the dimension of the families, therefore showing lower redundancy among sets while still preserving a high coverage of their elements.

We evaluate our approach for collections of gene sets and apply GSEA techniques to the now smaller collections. One of the main applications of collections of gene sets is enrichment analyses, e.g., the assessment of the potential over-representation or under-representation of the analyzed genes in specifically biologically annotated gene sets via Fisher tests and the GSEA algorithm [Sub+05; Mat+18]; among the enrichment analysis tools, we recall Enrich [Che+13; Kul+16; Xie+21], a web-based tool that provides various types of visualization summaries of collective functions of gene lists. GSEA methods perform a statistical test for each gene set in predefined collections in relation to single phenotypic traits. Due to the multiple hypothesis tests setup and the overlap among pathways, genes belonging to several pathways are tested several times; hence, the multiple testing naturally quickly becomes a major challenge [DVL08; Nob09] in high-dimensional collections. Among the various multiple test corrections, we recall the Bonferroni correction and the less conservative false discovery rate FDR [BH95; BY01]. As expected, reducing the high dimensionality of collections of gene sets could increase their maneuverability and interpretability, as well as the statistical power retained after correcting for multiple tests. In the classical supervised setup, selecting gene sets from these collections is possible only once the phenotypic trait for which we are willing to test has been fixed. On the contrary, our unsupervised approach allows for unremarkable differences in the number of significant gene sets for specific phenotypic traits, and the number of statistical tests performed can be drastically reduced.

In summary, our contributions can be summarized in the following points:

1. we propose unsupervised gene sets' importance scores based on the distribution of genes;
2. the defined setup solves the problem of the exponential computational complexity of Shapley values representing the family through a binary schema;
3. we show that the dimensionality-reduced collections are characterized by similar statistical significances for GSEA applications.

5.1 METHODS

In this section, we first introduce some basic notions of Coalitional Game Theory (CGT) and highlight some definitions we will use. An introductory toy example, e.g., the "glove game", for these rather abstract concepts can be found in Section 2.2.1.

5.1.1 *Cooperative Game Theory*

We introduced the basics from CGT and Shapley values in Section 2. We refer to Coalitional Games using the usual notation $(\mathcal{N}, f)$.

As already stated, the exact computation of Shapley values becomes infeasible as the number of players N increases. Recalling the definition of Shapley values from Equation (2.1), i.e.,

$$\phi_f(i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus i} \frac{1}{N\binom{N-1}{|\mathcal{A}|}} \left[f(\mathcal{A} \cup i) - f(\mathcal{A}) \right], \tag{5.1}$$

it is evident that the value function needs to be computed 2^N times. Due to the exponential complexity, computational problems arise when the number of players increases. However, one particular class of games, the *Sum-Of-Unanimity Games* (SOUG) [Sha+53], admits a polynomial closed-form solution, and microarray games are a special case of SOUG games. First, we will give a brief introduction to SOUG.

5.1.2 *Sum-Of-Unanimity Games (SOUG)*

Consider a set of players $\mathcal{N}$ and a coalition $\mathcal{T} \subseteq \mathcal{N}$; we can define the associated *unanimity game* $u_{\mathcal{T}}$ by

$$u_{\mathcal{T}}(\mathcal{S}) = \begin{cases} 1 & \text{if } \mathcal{S} \subseteq \mathcal{T} \\ 0 & \text{otherwise} \end{cases} \quad \text{for any } \mathcal{S} \subseteq \mathcal{N}.$$

It can be proved that given any cooperative game $(\mathcal{N}, f)$, the value function f can be written as the linear combination of unanimity games in a unique way, i.e.,

$$f(\cdot) = \sum_{\mathcal{T} \in \mathcal{P}(\mathcal{N})} \lambda_{\mathcal{T}}(f) u_{\mathcal{T}}(\cdot),$$

where $\lambda_{\mathcal{T}}(f) \in \mathbb{R}$ are called *unanimity coefficients* and are determined by the formula

$$\lambda_{\mathcal{T}}(f) = \sum_{\mathcal{S} \in \mathcal{P}(\mathcal{N})} (-1)^{t-s} f(\mathcal{S}).$$

As we see, the computation of $\lambda_{\mathcal{T}}(f)$, as well as the one of $\phi_f(i)$ becomes intractable if N increases.

On the other side, the SOUG allows for polynomial time computation of Shapley values. In particular, the computation in terms of the unanimity coefficients $\lambda_{\mathcal{T}}(f)$ is reduced to

$$\phi_f(i) = \sum_{\mathcal{T} \subseteq \mathcal{N} \setminus \{i\}} \frac{\lambda_{\mathcal{T}}(f)}{|\mathcal{T}|}$$

for each player i in $\mathcal{N}$.

It can be proven that any cooperative game $(\mathcal{N}, f)$ has a unique formulation as a sum of unanimity games. However, finding the equivalent SOUG of a game $(\mathcal{N}, f)$ is computationally equivalently hard as computing the Shapley values. Using SOUG brings the essential advantage of polynomial run-time when dealing with big families of sets, e.g., gene sets and pathways.

5.1.3 *Microarray games*

Let us consider $\mathcal{N} = \{P_1, \ldots, P_N\}$ the set of players; each $P_i \in \mathcal{N}$ is a set of elements and $\mathcal{N}$, i.e., the set of players, is a family of sets. We denote with

$$G = \{g \in P_i \mid P_i \in \mathcal{N}\} = \bigcup_{i \in \{1,\ldots,N\}} P_i$$

the elements belonging to at least one set P_i and $M = |G|$. Starting from $\mathcal{N}$ and G, we build a binary matrix $B \in \{0,1\}^{N \times M}$ where $B_{ij} = 1$ if $g_j \in P_i$ and $B_{ij} = 0$ otherwise. Transposing the definition given by Moretti et al. [MPB07], for each element $g_j \in G$, we look at the set of sets in which g_j is present; we call this set the support of g_j and denote it with $\mathrm{sp}(g_j)$. Mathematically, we obtain the support of g_j from the matrix B. The information about g_j is conveyed by the column B_j of B, and, by abuse of notation, we write $\mathrm{sp}(g_j)$ or $\mathrm{sp}(B_j)$ interchangeably. We define $\mathrm{sp}(B_j)$ as the set

$$\begin{aligned} \mathrm{sp}(B_j) &= \{P_i \in \mathcal{N} \mid B_{ij} = 1\} \\ &= \{P_i \in \mathcal{N} \mid g_j \in P_i\}, \end{aligned}$$

i.e., the set of the sets containing g_j. The *microarray game* is then defined as the cooperative game $(\mathcal{N}, f^*)$ where, for each $\mathcal{T} \subseteq \mathcal{N}$,

$$f^*(\mathcal{T}) = \frac{|\Theta(\mathcal{T})|}{|G|} = \frac{|\{g_j \in G \mid \mathrm{sp}(g_j) \subseteq \mathcal{T} \text{ and } \mathrm{sp}(g_j) \neq \varnothing\}|}{|G|}. \tag{5.2}$$

Here the adapted value function f^* computes the ratio between the number of genes' supports that $\mathcal{T}$ contains and the number of elements in G. As $|G|$ is fixed, we can simply say that $f^*(\mathcal{T})$ is proportional to the number of supports contained in $\mathcal{T}$; higher scores are achieved by sets covering the full distribution among sets of a high number of elements.

Following Sun et al. [Sun+20], we can easily express the value function as a linear combination of unanimity games where each column is interpreted as a unanimity game. Using this formulation of the value function, the computation of Shapley values is reduced to polynomial time.

5.1.4 *Computation of Shapley values and definition of the game*

As discussed in Chapter 4, Shapley values are a common solution to assign fair scores to players within a cooperative game. However, they show an inherent problem: redundant players get similar scores, thus implying that they are ranked in close positions. The "glove game" example in Section 2.2.1 clearly illustrates the problem. To solve this, we integrate a redundancy-awareness concept into Shapley values to rank players, taking possible overlapping among them into account. In particular, the player ranked at the $(i+1)$-th place should be the least overlapping possible with the first i-ranked players $\{P_1, \dots, P_i\}$. To achieve such a redundancy-aware ranking, we introduce different pruning criteria for players similar to the ones previously ranked.

Each set P_i contains a variable number M_i of elements, i.e., $P_i = \{g_1, \dots, g_{M_i}\}$, and the sets in $\mathcal{N}$ are arbitrarily large and can overlap. We construct a microarray game based on the binary matrix $B \in \{0,1\}^{N \times M}$ where $N = |\mathcal{N}|$ and $M = |\cup_{i=1}^{N} P_i|$. Each row of B represents a set P_i and $B_{ij} = 1$ if $g_j \in P_i$ while $B_{ij} = 0$ if $g_j \notin P_i$. Each column B_i represents the partial ordering relationship of the element g_j belonging to the set P_i.

Given a set $P_i \in \mathcal{N}$, the Shapley value of P_i is computed following these two steps as proposed in [Sun+20]:

1. from the matrix B, we get the dictionary $\mathcal{A}$ as

$$\mathcal{A} = \{\mathrm{sp}(B_j)\}_{j \in M} \subseteq \mathcal{P}(\mathcal{N}). \tag{5.3}$$

   Each set in $\mathcal{A}$ represents the support of the corresponding element $g_j \in G$.

2. each Shapley value is computed through the formula:

$$\phi_{f^*}(P_i) = \frac{1}{M} \cdot \sum_{j=1}^{M} \left(\mathbb{1}(P_i \in \mathrm{sp}(B_j)) \cdot \frac{1}{|\mathrm{sp}(B_j)|} \right), \tag{5.4}$$

   where $\mathbb{1}$ is the standard indicator function returning 1 if the argument is satisfied, i.e., if $P_i \in \mathrm{sp}(B_j)$, and 0 otherwise. We drop here the notation $\phi_{f^*}(P_i)$ as Equation (5.4) is a reformulation of the Shapley values in terms of microarray games; from here on, we simply write $\phi(P_i)$.

Once computed, we can then use the Shapley values to order the sets P_is: the higher the Shapley value of a set P_i, the more important the set is in the microarray game defined. The importance scores measure the number of elements g contained in P_i re-scaled with the size of their supports. If P_i contains elements g rarely included in other sets, it will get a higher score. Each $\phi(P_i)$ is a real number in $[0,1]$ and from Shapley values' properties we know that $\sum_{i=1}^{N} \phi(P_i) = 1$. However, as already mentioned, the ranking of sets given by the Shapley values alone is unaware of a possible "overlap" among players.

5.1.5 *Definition of goals: redundancy and coverage*

Different “redundancies” can appear among players in a cooperative game depending on the game’s structure. When using random variables, “redundancy” often refers to the correlation among sets of variables (cf. Chapter 4); here, we aim for a redundancy-aware ranking in families of sets. We state that two sets are *redundant* if they share a large number of elements, i.e., if the size of their intersection is large compared to the size of their union. To measure the redundancy among sets, we use the *Jaccard index* J [Jac01]; given two sets A and B two sets, their Jaccard index is

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}. \tag{5.5}$$

The Jaccard index is a real number in $[0,1]$ where $J(A,B) = 0$ if and only if $A \cap B = \varnothing$ and $J(A,B) = 1$ if and only if $A = B$. Thus, the Jaccard index is direct proportional to the size of the intersection among the sets A and B.

Having set the reduction of redundancy within importance scores-based rankings as a goal, we still do not want to compromise with the *coverage* of G. We hereby define various types of pruning criteria and will compare them with respect to coverage *and* redundancy.

1. **Redundancy** – as redundancy measure, we assign to a family of sets $\mathcal{S}$ the *Jaccard rate* or *Jaccard score* $J_{\text{score}}(\mathcal{S})$, i.e.,

$$J_{\text{score}}(\mathcal{S}) = \frac{1}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{P_i,P_j \in \mathcal{S}, i \neq j} J(P_i,P_j). \tag{5.6}$$

The Jaccard score $J_{\text{score}}(\mathcal{S})$ represents the average Jaccard index among pairs of sets in $\mathcal{S}$; it is a non-negative real number in $[0,1]$ and $J_{\text{score}}(\mathcal{S}) = 0$ if and only if all pairs of sets in $\mathcal{S}$ do not overlap.

2. **Coverage** – per definition, the family of players $\mathcal{N}$ represents one possible coverage of G since each element $g \in G$ is contained at least in one set $P_i \in \mathcal{N}$. Given a family of sets $\mathcal{S} \subseteq \mathcal{P}(\mathcal{N})$, the quantity

$$c_G(\mathcal{S}) = | \cup_{P_i \in \mathcal{S}} P_i | \cdot \frac{100}{|G|} \tag{5.7}$$

measures the *coverage of G given by* $\mathcal{S}$, i.e., the percentage of elements $g \in G$ that are included at least in one set in $\mathcal{S}$. There is an obvious trade-off between the coverage given by

$\mathcal{S}$ and its dimension. In the application to collections of gene sets, we will investigate our methods' success in preserving the entire set's coverage while reducing the dimension of $\mathcal{N}$.

Rankings based on Shapley values show a clear tendency to rank bigger sets first. This is reasonable, as the Shapley value counts the number of times the argument of the indicator function is satisfied; sets with larger sizes are, hence, the ones for which the indicator function arguments' are most often satisfied. However, these bigger sets are more likely to overlap as they probably contain elements over-spread through the sets in the family. Hence, when looking for rankings with low redundancy, we will also affect this tendency to rank smaller sets later and bigger ones first.

5.1.6 *Different pruning criteria and different rankings*

Given the definition of Shapley values as in Equation 5.4, we obtain importance scores for each of the sets in the family $\mathcal{N}$ in polynomial time. We can use these scores to rank the sets in $\mathcal{N}$ in a naive manner and refer to this ranking as SV. As mentioned in the previous section, this ranking favors larger sets: a set P_i is contained in a larger number of supports if its size is larger, i.e., the expression $\mathbb{1}(P_i \in \text{sp}(g_j))$ assumes more often value 1 for larger sets. Moreover, it is worth noticing that given two sets P_i and P_j with equal size, the importance score of P_i is larger than that of P_j if P_i contains rarer elements than P_j. On the other hand, as the Shapley values tend to rank larger sets first, we expect a high coverage of G when selecting even a small number of sets. Moreover, the value function does not include any awareness of overlap among sets, and the ranking SV allocates sets P_i, P_j in similar ranking positions when $J(P_i, P_j) \approx 1$.

We introduce various pruning criteria to penalize overlapping sets and not rank them similarly. The introduced pruning criteria are functions of the Jaccard index among sets, such that low Jaccard rates characterize pairs of subsequently ranked sets. We provide a detailed comparison of the obtained rankings in Section 5.2. Using the application to collections of gene sets, we illustrate the properties of each of the obtained rankings and how to select the most useful for the purpose at hand. An overview of the proposed methods highlights that

- the proposed pruning criteria are rather flexible and can be adapted to optimize specific properties and
- there is not a perfect and unique choice fulfilling all goals.

The obtained rankings are constructed one on top of the other. Moreover, the rankings are constructed using a greedy approach that selects one set at a time; as the computation of the Shapley values using the constructed microarray game is not computationally expensive, this still leads to feasible run times for the entire ranking. The Shapley values need to be re-computed after each iteration as, after selecting some pathways, they no longer sum to 1. Moreover, removing one set can change the ordering of the other sets as the Shapley values depend on the distribution of the elements among the sets.

1. Penalized Ordering (PO) – the Shapley values are used to obtain the first ranked set $\tilde{P}_1$, i.e., the one whose Shapley value is the highest.
   In the second step, all Shapley values are re-computed for the not-yet ranked sets, and we subtract to them $J(\tilde{P}_1, P)$, i.e., the Jaccard index among $\tilde{P}_1$ and the to-be-ranked set P. For each set $P \neq \tilde{P}_1$, the importance score $S_2(P)$ at the second step reads
   $$S_2(P) = \phi^{(2)}(P) - J(\tilde{P}_1, P). \tag{5.8}$$
   The penalty score aims to penalize highly overlapping sets with $\tilde{P}_1$. The set $\arg\max_P S_2(P)$ obtains the second position and the process restarts. The penalty grows at each step as we add it to the Jaccard index with the last ranked pathway: in particular, after selecting the first n sets, the score $S_{n+1}(P)$ obtained by the set P (where P has not been ranked yet at step $n+1$) is given by the following recursive formula
   $$\begin{aligned} S_1(P) &= \phi^{(1)}(P) \\ S_{n+1}(P) &= \phi^{(n+1)}(P) - \textstyle\sum_{i=1}^{n} J(\arg\max_{\bar{P}} S_i(\bar{P}), P), \\ &\qquad\qquad\qquad\qquad \text{if } n \geq 1 \end{aligned}$$
   and, at the step $n+1$, the algorithm ranks the set $\tilde{P}_{n+1} = \arg\max_P S_{n+1}(P)$.

   We underline that the Shapley values are re-computed after each iteration, and $\phi^{(n)}(\cdot)$ represents the Shapley value function at iteration n. Recomputing the Shapley values is necessary for two main reasons: first, to satisfy the efficiency property, i.e., $\sum_{P_i} \phi^{(m)}(P_i) = 1$ for each m where the sum is computed over the sets which have not been ranked yet; second, the set of sets "not yet ranked" changes at each iteration, implying a (possible) different order of the sets when the Shapley values are re-computed.

2. Penalized Ordering with Rescaling (POR) – POR ranking adds to PO a rescaling of the penalty; the ordering obtained using PO automatically increases the penalty after each iteration. After a sufficiently large number of iterations, this process can lead to penalties larger than the Shapley values themselves and sets end up being assigned negative importance scores. Two major issues are connected with this:

   a) negative importance scores are hardly interpretable, and

   b) the pruning criteria can become too harsh with respect to the Shapley values.

   Thus, we propose to re-scale the penalty: to compute the POR ranking, the term $\sum_{i=1}^{n} J(\arg\max_{\bar{P}} S_i(\bar{P}), P)$ is re-scaled to the interval $[0, \max_P\{\phi_i(P)\}]$ at each iteration. The importance scores are defined as follows:

$$\begin{aligned} S_1(P) &= \phi^{(1)}(P) \\ R_n &= \frac{\max_P \phi^{(n)}(P)}{\max_P \sum_{i=1}^{n} J(\arg\max_{\bar{P}} S_i(\bar{P}), P)} \\ S_{n+1}(P) &= \phi^{(n+1)}(P) - \sum_{i=1}^{n} J(\arg\max_{\bar{P}} S_i(\bar{P}), P) \cdot R_n, \\ &\qquad\qquad\qquad\qquad\qquad\qquad \text{if } n \geq 1. \end{aligned}$$

   The complexity of the algorithm proposed does not change.

3. Artificial Ordering (AO) – the introduction of an artificial set represents our attempt to avoid penalizing each set multiple times for containing the same elements. The artificial set AP_n is updated in each iteration n. It is initialized at step 1 to $AP_1 = \arg\max_P \phi^{(1)}(P)$. At the nth iteration, AP_n is updated with the elements of the last ranked set

$$AP_n = \cup_{i=0}^{n} \arg\max_P S_i(P);$$

   hence, at each n, AP_n includes all genes belonging to one of the previously selected pathways. The importance score is defined as in PO, but penalizing with a unique Jaccard index with the set AP_n instead of using the sum of Jaccard indices with previously ranked sets, i.e., the scores are computed as

$$\begin{aligned} S_1(P) &= \phi^{(1)}(P) \\ S_{n+1}(P) &= \phi^{(n+1)}(P) - J(AP_n, P), \qquad \text{if } n \geq 1. \end{aligned}$$

   In AO the penalty depends on the elements in G that the first n ranked sets have covered. This avoids multiple penalties for the same overlapping elements; thus, the penalties will be *softer* with respect to these.

4. Artificial Ordering with Rescaling (AOR) – as in the re-scaled version of PO, the re-scaling is done for AO on the term $J(AP_n, P)$ to the interval $[0, \max_P\{\phi_i(P)\}]$. Again, the aim is to avoid too harsh penalties, eventually causing negative importance scores. The scores look like the following:

$$\begin{aligned} S_1(P) &= \phi^{(1)}(P) \\ R_n &= \frac{\max_P \phi^{(n)}(P)}{\max_P J(AP_n,P)} \\ S_{n+1}(P) &= \phi^{(n+1)}(P) - J(AP_n, P_j) \cdot R_n, \qquad \text{if } n \geq 1. \end{aligned}$$

   As in POR, the complexity of the algorithm proposed does not change.

The SV ranking is not functional for both maximizing the importance scores given by the Shapley values and lowering the redundancy among the subsequent pairs of ranked sets. Thus, we introduced four different pruning criteria for constructing final rankings. The two pruned rankings, PO and POR, consider only overlappings among pairs of sets. The penalty is increased at each step by adding the Jaccard rate among the last pair of sets; thus, if elements are contained in multiple family sets, these elements will affect the penalty terms multiple times. As this might be problematic for small sets containing some of these often-appearing elements in the long run, we introduced the artificial set AP to create the AO and AOR rankings. Using the artificial pathways, we solve the problem of multiple punishments as the overlaps with single elements are penalized exactly once. In both AOR and POR, the re-scalings attenuate the effect of the pruning criteria, such that the scores are kept positive for the sets whose penalties are higher than the Shapley values. Note that each penalty still orders the set with the highest Shapley value first. The orderings start to differ from each other in the second-ranked set.

The pseudo-code is available for reproducibility while the implemented code is publicly available at github[1].

5.2 RANKING PATHWAYS IN COLLECTIONS OF GENE SETS

Implementing our game-theoretic concept provides a new framework to reduce the dimension of families of sets. The obtained rankings are used to select the first n ranked sets, allowing for a lower overlap among sets and high coverage of the elements with a lower number of sets, thus increasing the interpretability of the

[1] Code available at chiarabales/geneset_SV

	correlation with pathways' sizes	**redundancy**	**coverage**
sv	positive correlation	reference level	reference level
po	negative correlation	much less	same
por	no correlation	less	less
ao	negative correlation	much less	same
aor	no correlation	less	less

Table 5.1: Comparison. Original sv ranking and pruned rankings.

families of sets. Moreover, unlike previously introduced methods, our approach does not alter the sets.

Collections of gene sets are a promising application of our rankings and the source of inspiration for the proposed method. In particular, the collections of gene sets $\mathcal{F}$ are sets of pathways, and pathways P_i are sets of genes g_j. We use these collections as a show case for our method.

We use this application to illustrate that the rankings proposed

- provide a ranking of the original pathways in the collections of gene sets without modifying them;
- *reduce the redundancy* among subsequently ranked pathways;
- maintain a *high coverage* of the genes in the collection of gene sets when selecting the first n ranked pathways;
- do not favor larger gene sets;
- reduce the size of the collections of gene sets, thus increasing interpretability.

The experiments emphasize that the choice of which pruning criterion to use highly depends on the goal, and there is not a *unique correct* way of choosing which ranking to use.

In Table 5.1, we summarize the properties of the different rankings based on the analysis of different data sets. We present the results for four collections of gene sets, i.e., the KEGG, CGN, CM, and TFT LEGACY. To complete our analysis, we investigate the effects of the different pruning criteria and descending pathways selections on the gene set enrichment analysis, looking for the significance of pathways for different association traits.

We included additional analyses comparing our game theoretic approach with more classical enrichment analysis methods in Table 5.3. There, we also focused on the ability to detect significant pathways after reducing the gene set with respect to 38 phenotypic

	SV	PO	POR	AO	AOR
KEGG	0.49	-0.21	0.15	0.23	0.19
CGN	0.43	-0.51	0.019	-0.702	-0.041
CM	0.759	-0.531	0.019	-0.702	-0.041
TFT LEGACY	0.679	-0.460	0.354	-0.828	0.458

Table 5.2: CORRELATION AMONG PATHWAY' SIZE AND POSITION. Kendall's τ coefficients among the position in the ranking and the size of the gene sets.

traits. Given that we present arbitrary choices both for collections of gene sets and phenotypic traits, our experiments and analyses should be seen as exploratory in the spirit of illustrative case studies. We explicitly do not claim the generalizability of these results.

5.2.1 *Correlation with the size of pathways*

The Shapley value function assigns to a set $P \in \mathcal{F}$ a positive real number incorporating information on the distribution of its elements in the other sets of $\mathcal{F}$. This leads to a positive correlation with the size of pathways, i.e., larger sets are more likely to get a higher Shapley value. In Table 5.2, Kendall's τ scores measure the ordinal association between the size of pathways and their position in the rankings. The table clearly displays that when ranking the pathways using SV, we tend to rank larger pathways first. Using AO and PO, this effect is reversed in most collections of gene sets; in particular, AO and PO rankings select small pathways first while larger ones are ranked last in the orderings. In AOR and POR, there is no clear tendency of a correlation between the dimension of pathways and the position in the ranking.

Our goal was to reduce the redundancy among subsequently ranked pathways. Hence, we indirectly affect the strength of the correlation between ranking position and size as larger pathways are more likely to show overlapping among them. The rankings SV, PO, and AO, show similar behaviors across the different studied collections of gene sets, while the re-scaled pruning criteria show no clear tendency.

5.2.2 *Redundancy awareness*

The introduced pruning criteria ensure rankings that consider the overlap among subsequent ranked gene sets. We evaluate the redundancy using the Jaccard score defined in Equation (5.6). To get comparable numbers throughout different collections of gene sets, we re-scale the Jaccard scores to the *maximum Jaccard score*, i.e., the maximum Jaccard index among any pairs of pathways within the collection of gene sets. In Figure 5.1, we plot the re-scaled Jaccard scores as a function of the number of pathways. We first compute the rankings, then select the respective first n ranked gene sets and compute the Jaccard rate of the obtained set. The lower the Jaccard rate, the better the ranking performs in selecting non-overlapping pathways.

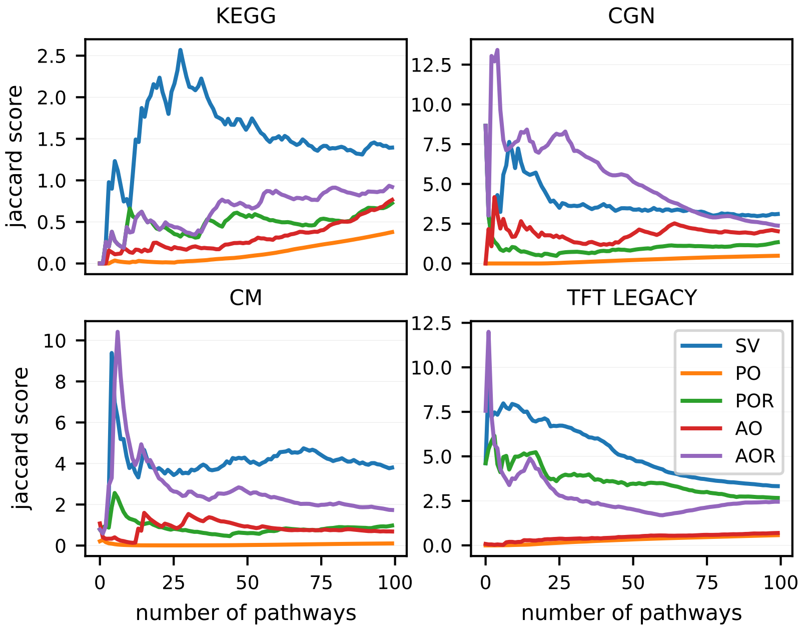
The table in Figure 5.1 shows the re-scaled Jaccard rates for some reference values (10, 20 and 40% of the pathways). The ranking achieved the lowest Jaccard rates is PO; AO performed well in (almost) all collections of gene sets. PO and AO use strong penalties such that highly overlapping pathways were ranked far from each other. Moreover, we note that the classical SV ordering performed the worst in all but one case as it is unaware of redundancies.

5.2.3 *Coverage of gene sets*

We investigate the ability of our methods to cover the genes using a limited amount of pathways. In Figure 5.2, we plot the coverage of the genes in percentage when only considering the first n ranked pathways. The SV ranking gets a generally high coverage of genes in the collections of gene sets. We note that the orderings SV, POR, and AOR clearly outperform the rankings given by PO and AO.

Moreover, we compare the different rankings using some reference levels: in particular, the table in Figure 5.2 gives insights into the proportions of the genes that can be covered using only a limited percentage of pathways (10, 20 and 40% of the pathways). The high coverage achieved by SV is due to the correlation between the size of the pathways and their positions in the ranking; however, not the same can be argued about POR and AOR as we demonstrated that the correlation with the gene sets' sizes had been reduced. Maximizing Shapley values while minimizing redundancy achieves outstanding performances in both cases. The lower performances of PO and AO are explained by the pruning criteria, which are generally harsh for overlapping gene sets; hence, they select first small pathways ranked in the lowest positions by Shapley values alone, as already argued (cf. Table 5.2).

On the other hand, we observe that the rankings do not outperform the original SV ordering in covering the entire gene set; the advantages of the penalized orderings are evident when considering that the performances of the newly proposed rankings are



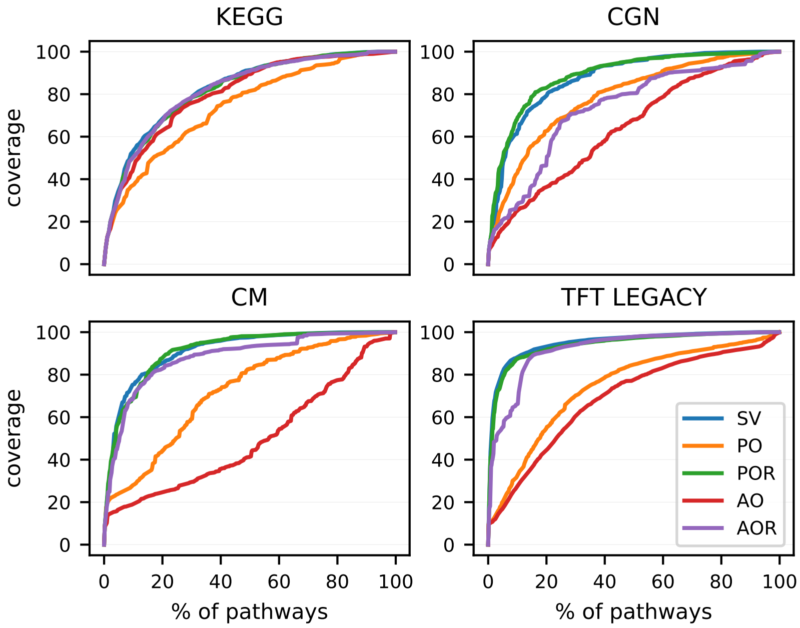
	KEGG			CGN			CM			TFT LEGACY		
%	10	20	40	10	20	40	10	20	40	10	20	40
sv	1.95	2.11	1.37	3.73	3.11	2.75	3.87	4.13	2.16	4.31	3.10	2.80
po	0.02	0.04	0.21	0.14	0.42	0.72	0.02	0.08	0.21	0.39	0.65	1.09
por	0.49	0.47	0.46	0.79	1.07	1.45	0.51	0.90	1.37	3.5	2.51	2.22
ao	0.15	0.19	0.38	1.21	1.97	1.36	1.19	0.80	0.63	0.56	0.81	1.36
aor	0.49	0.45	0.77	5.82	2.97	2.20	2.56	2.00	1.25	1.70	2.39	2.43

Figure 5.1: Redundancy awareness. The plots show the average re-scaled Jaccard scores of sets of pathways ranked up to *j*-th position (*x* axis); we select up to 100 pathways in each collection. The table shows the re-scaled Jaccard scores of the first 10, 20, and 40% of the collections of gene sets. The minimum Jaccard score in each column is highlighted.

close to the original sv ranking while retaining a much smaller amount of redundancy and not preferring large pathways.

5.2.4 *Number of significant pathways*

Finally, we investigate how the proposed rankings relate to gene set enrichment analysis using only the first *n* ranked pathways. We use Fisher's exact test [Fis35; Agr18] to determine whether a pathway is significant or not to a specific phenotypic trait and apply multiple hypothesis testing corrections for the p-values (Bonferroni or FDR correction [BH95; DVL08]). Using the proposed method



	KEGG			CGN			CM			TFT LEGACY		
%	10	20	40	10	20	40	10	20	40	10	20	40
SV	52.7	68.4	86.2	61.3	79.3	93.4	75.4	85.6	95.9	88.2	93.3	97.0
PO	37.0	52.4	74.7	42.6	62.7	81.6	28.0	44.2	73.2	32.3	55.1	78.6
POR	50.0	67.7	84.6	68.0	82.9	93.5	68.9	87.9	96.2	87.5	91.8	96.1
AO	43.7	63.0	81.2	25.0	36.3	57.7	19.1	24.5	35.6	27.4	44.7	70.7
AOR	50.0	67.6	85.9	28.0	46.4	77.9	68.9	82.7	91.7	66.1	91.0	96.6

Figure 5.2: CUMULATIVE COVERAGE OF GENE SETS. The plots show the genes' coverage as functions of the number of pathways using the different rankings. The table shows the genes' coverage when selecting the 10, 20, and 40 % of the pathways respectively. In each column, the highest coverage is highlighted.

of ranking and selecting, we obtain smaller collections of gene sets; afterward, we test for associations with specific phenotypic traits and compare the number of significant pathways found in the original collections of gene sets and in the reduced ones.

Figure 5.3 illustrates for each collection of gene sets the number of statistically significant pathways founds for some association traits, i.e., *blood platelet count*, *blood white count* and *sitting height*. The plots refer to the FDR correction for multiple hypotheses testing. The number of significant pathways in each collection of gene sets is represented in each plot as a blue dashed line. We observe that the number of significant pathways found when limiting the number of tested pathways using the introduced rankings highly

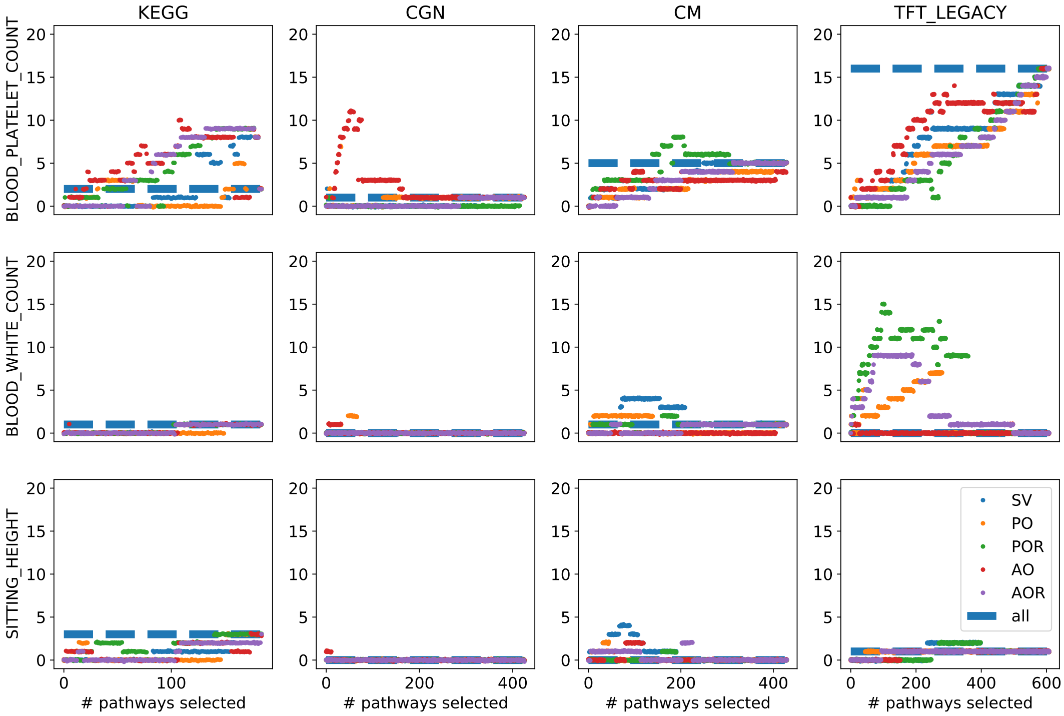


Figure 5.3: SIGNIFICANT PATHWAYS DETECTED. In each plot, the x-axis represents the number of pathways included in the multiple statistical testing; the y-axis represents the number of statistically significant pathways found. The plots refer to FDR correction for multiple testing with $\alpha = 0.05$ for three selected association traits an for four collections of gene sets.

depends on the collection of gene sets and the particular trait. In some settings using only a limited number of pathways may lead to a higher number of pathways reaching significance. In contrast, we detected fewer significant pathways in other settings. This happens when significantly associated pathways are not ranked among the first n, which obviously can occur when applying our unsupervised feature selection techniques. Similarly, Table 5.3 shows the number of significant gene sets when applying Enrich on two additional reduced collections of gene sets.

In conclusion, the number of significant pathways discovered when using the proposed rankings remains, on average, the same as using the whole collection of gene sets. Reducing the collections of gene sets to a limited amount of pathways using unsupervised approaches like the one we propose might lead to better interpretability and handling of gene sets *but not necessarily* to a higher statistical power in enrichment analyses. Whether the number of significant gene sets found is increasing or decreasing highly depends on the phenotypic trait and the collection of gene sets used.

	KEGG13							BIOCARTA						
	SV	PO	POR	AO	AOR	ALL	ENR	SV	PO	POR	AO	AOR	ALL	ENR
height	-	7	-	7	7	7	-	-	2	2	2	-	2	2
bl. platelet count	-	-	-	1	-	1	-	-	-	-	-	-	-	-
standing height	-	-	-	-	3	3	-	-	2	2	2	-	2	-
bl. red count	-	-	1	1	-	1	-	-	-	2	2	-	2	11
heel tscore	-	-	3	-	-	3	4	-	-	-	2	2	1	1
bl. white count	-	-	-	-	-	1	-	-	-	-	-	-	-	-
bl. eosinophil count	-	-	-	-	-	-		-	-	-	-	-	-	3
sitting height	-	-	-	-	-	-	-	-	-	-	-	-	-	-
trunk fat free mass	-	-	-	-	-	1	-	-	-	-	-	-	-	-
trunk pred. mass	-	-	-	8	8	8	-	-	1	-	1	-	-	-
body fat free mass	-	-	-	-	-	-	-	-	-	-	-	-	-	-
body water mass	-	-	-	-	-	-	-	-	-	-	-	-	-	-
systolic bl. pressure	-	-	-	-	-	-	-	-	-	-	-	-	-	-
basal metabolic rate	-	-	-	-	-	-	-	-	-	-	-	-	-	-
impedance body	-	-	-	-	-	-	-	-	-	-	-	-	-	-
bmi	-	-	-	-	-	-	-	-	-	-	-	-	-	-
height-size@10	-	-	-	-	-	-	-	-	-	-	-	-	-	-
arm pred. mass r	-	-	-	-	-	-	-	-	1	-	1	-	-	-
arm fat free mass r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
leg fat free mass r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
leg pred. mass r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
lung fev1fvc ratio	-	-	-	-	-	-	-	-	-	-	-	-	-	-
impedance leg r	-	-	-	-	-	-	1	-	-	-	-	-	-	1
impedance arm r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
lung fvc	-	-	-	-	-	-	1	-	-	-	-	-	-	1
weight	-	-	-	-	-	-	-	-	-	-	-	-	-	-
whratio	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hair pigment	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hip circumference	-	-	-	-	-	-	-	-	-	-	-	-	-	-
trunk fat mass	-	-	-	-	-	-	-	-	-	-	-	-	-	-
body fat mass	-	-	-	-	-	-	-	-	-	-	-	-	-	-
arm fat mass r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
leg fat mass r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
trunk fat %	-	-	-	-	-	-	-	-	-	-	-	-	-	-
body fat %	-	-	-	-	-	-	-	-	-	-	-	-	-	-
arm fat % r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
leg fat % r	-	-	-	-	-	-	-	-	-	-	-	-	-	-
cardiovasc. disease	-	-	-	-	-	-	2	-	-	-	-	-	-	2

Table 5.3: NUMBER OF SIGNIFICANT PATHWAYS DETECTED in the first 40% of the various rankings and the complete collection ALL (with Fisher Exact Test and FDR correction) against the Enrich method ENR.

5.3 DISCUSSION

Shapley values are often used to assign a fair value to players based on their contribution to the game. We often mentioned how they ignore eventual redundancies among players, hindering their performance and possibly inducing a bias in the resulting scores towards redundant players.

In this chapter, we proposed a game-theoretical approach to incorporate redundancy-awareness into Shapley values to rank gene sets in an unsupervised fashion. In particular, we proposed different ways to penalize the overlapping sets so they are not progressively selected.

We proposed four different pruning criteria, and we were able to show that the orderings obtained are

- *not favoring larger players* – applying the redundancy-aware pruning criteria avoids that larger gene sets are ranked first;
- *redundancy free* – the combination of Shapley values with the redundancy reduction criteria shows high effectiveness in maintaining the importance of sets given by Shapley values while reducing the redundancy among the first-ranked pathways;
- *achieving high coverage* – the obtained rankings still lead to high coverage of genes. We showed that a positive correlation with the size of sets is not the unique solution to achieve high coverage of the genes, i.e., the original Shapley values ranking is not performing much better than the orderings which rank first small sets, keeping low redundancy rates;
- *on average do not have a high influence on the number of detected significant pathways* - having fixed the collection of gene sets, the number of significant pathways detected when applying GSEA techniques with multiple hypotheses testing corrections increases and decreases compared to the full gene set, depending on the phenotype.

5.3.1 *Comparison among pruning criteria*

In light of the results in Figure 5.1 and Figure 5.2, we conclude that AO leads to the least favorable ranking in covering the genes. In contrast, the two re-scaled orderings together with SV are the best-performing ones regarding coverage. Regarding reducing redundancy among pathways, all penalized orderings can achieve this goal by outperforming the SV ranking. Lastly, when comparing the different methods with respect to the correlation with the size of pathways, we see that only AOR and POR do not lead to any specific correlation.

Hence, we conclude that POR is the best ordering one could choose when the aim is to optimize the ranking for redundancy elimination and cover the genes without incurring specific correlations with the gene sets' sizes.

5.3.2 *Limitations*

Our results suggest that using our tool as a pre-processing step for collections of gene sets, we get similar numbers of significant pathways when checking for association with phenotypic traits (although relying on much fewer pathways). In other words, we evaluated the significance of the top-ranked pathways with respect to phenotypic traits, yielding results that were very similar to those of the unfiltered collections of gene sets. In some cases, however, we observe fewer significant pathways than when considering the whole collection. If the interest is increasing the statistical power for specific collections of gene sets and phenotypic traits, and if possible, we still suggest using a supervised method to reduce the number of pathways. It is fundamental to remember that our rankings based on Shapley values are unsupervised. They are based on the structure of the collections of gene sets and pathways and their overlaps; thus, they are not meant to necessarily "increase" the number of significant pathways detected in supervised contexts.

6 SLIDSHAPS – SLIDING SHAPLEY VALUES FOR CORRELATION-BASED CHANGE DETECTION IN TIME SERIES

For volatile multivariate time series, variations in the distributions of the input dimension and the correlation structure present an open challenge. The different distributions before and after a change-point hinder the performance of most of the predictive methods, mostly requiring re-training of the models. State-of-the-art change point detectors are based on the prediction error rate of online classifiers, assuming that distributional drifts appear in $\mathbb{P}(y|X)$ [Hal+21; BG07; Gam+04]. The detection of these last change points represents a severe problem, as volatile data labeling is often either expensive or delayed in streaming data, and the increasing number of scenarios where time series data are collected without labels and the necessity of dealing with distributional changes on input variables forced the development of unsupervised concept drift detection methods [Gem+20]. Those methods find change points by comparing the current data distribution with a reference historical data buffer; they rely on two main steps, i.e., (1) a new time series representation and (2) the detection of changes over the representations. Several approaches exist in the literature: some using the mean values of adaptive windows to represent the univariate dimensions [BG07], linear and non-linear features representing the whole time series [CMO16] and multidimensional Fourier transformation to get information from the frequency domain [Cos+17]. Recently, combinations of multiple statistical features have been proposed as meta-information vectors [Hal+21]. Other approaches measure the distributional discrepancy between data in different time periods, e.g., the Hellinger distance between two distributions [DP11] and data partition via Kdq-Tree and generalizations of the Kulldorff's spatial scan statistic [Das+06]. Unfortunately, they all do not monitor the correlation changes among input variables, and, as a consequence, feature correlation remains mainly studied using simple covariance. Tracking covariance changes in a transformed artificial low-dimensional space obtained by applying PCA on the time series [Qah+15] has been used to detect concept drifts; however, the approach limits to tracking covariance changes in the extracted space. To overcome this limitation, [AK18] uses mean and covariance to represent the concepts in multivariate data streams. Generally, however, classical concept drift detectors struggle with detecting changes in correlations of multivariate time series' input variables.

We focus on unsupervised change detection, tracking correlation changes in the input variables without class labels using Shapley values. As already seen, Shapley values are not limited to trustworthy machine learning. Their popularity derives from the flexible definition of the value function based on which they are computed. However, as time dependency represents an additional challenge, applying Shapley values in time series data is still not fairly explored. Only recently, they started finding applications in time series data; they appeared to explain black-box models on time series data [Gui+20] or as an extension of the classical SHAP [Ben+21b], for anomaly detection through the reconstruction error of autoencoders [Ant+21], single instances studies [Tak19a], or analyzing the gradients to identify the main features affecting the anomalies [Ngu+19]; they further appeared in consulting and business applications [Sal+21]. We further recall some works applying Shapley values for drift detection. Among them, we find Zheng et al. [Zhe+19] using Shapley values for drift detection for labeled series and Zhao et al. [ZK20], that employ Wasserstein and Energy distances to detect feature drifts without labels; SHAP [LL17] and LIME [RSG16] are here used as post-hoc interpretation for the detected drifts. All the mentioned works rely on labeled data streams or represent a pure transposition of time-independent methods to the time series context; again, state-of-the-art literature misses an approach able to handle unlabeled streaming data.

Our work represents the first attempt to study concept drifts in unlabeled data using Shapley values as a fully unsupervised change points detector for multivariate time series. The SLIDSHAPs method detects correlation-based changes through a representation of the correlation structure of the input data. Our approach underlines distributional changes even in a few univariate input variables, thus being more sensitive to changes than any prior change point detection method; in contrast to covariance-based approaches, Shapley values aggregate in N scores the correlations scores in any subsets of the input variables (see Table 6.1). The SLIDSHAP series clearly outlines that changes in a single or a few input variables potentially affect the correlation structure of the whole time series dimensions.

We use this foundational game-theoretic concept to extrapolate information on the correlation structure of data streams and achieve higher sensitivity towards multiple changes in the empirical evaluation of both synthetic and real-world data. We summarize the advantages of SLIDSHAPs in two main points:

	unsupervised & classifier-free	pairwise correlations	multivariate correlations	visualization
supervised methods [Gam+04; Bae+06; Fri+14]	✗	✗	✗	✓
covariance-based [AK18; Qah+15]	(✓)	✓	✗	✗
Shapley value-based [Ben+21b; ZK20; Zhe+19]	(✓)	✗	✗	✓
slidSHAPs [BLM23b]	✓	✓	✓	✓

Table 6.1: Comparison among change point detection methods.

1. slidSHAPs is a change point detector for unlabelled data that only relies on the correlations among univariate dimensions of the time series;
2. the slidSHAP series allows for visualizing whether a change in the underlying distributional concepts happens.

6.1 methods

We introduce slidSHAPs to visualize and detect correlation changes in unlabelled multivariate discrete time series. As we hypothesized, even distributional changes happening only in a few univariate dimensions can drastically change the correlation structure among all univariate dimensions. We map the time series into the slidSHAP series, where we implicitly encode correlations among the time series' input variables as a function of time; the resulting time series has a different dependency on time from the original timestamps. We use the slidSHAP series to detect change points through statistical tests, and we finally relocate the found points to the original time notion.
Figure 6.1 provides an overview of our method; we go through each step in the following sections.

6.1.1 *Time series and sliding windows*

We indicate with $X = (X_1, \dots, X_N)$ a multivariate N-dimensional discrete time series where X_i is the i-th univariate dimension; we currently restrict the approach to time series whose dimensions assume only a finite number of values, i.e., either categorical or discrete and finite.

With t_0, we indicate the first timestamp on which the time series is defined. For each timestamp $t_k > t_0$, $X(t_k)$ is a N-dimensional

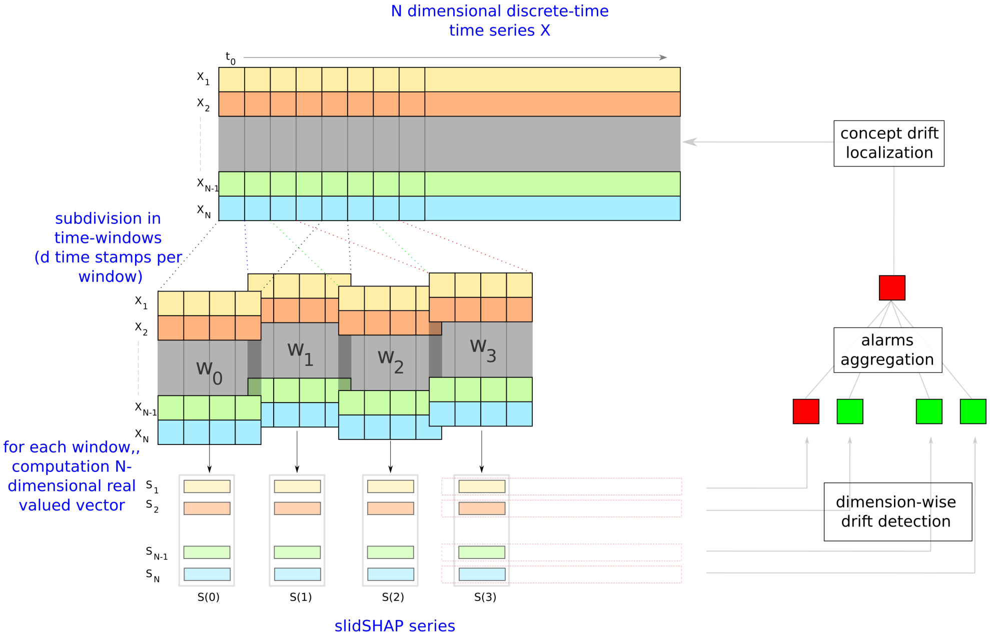


Figure 6.1: SCHEMATIC VISUALIZATION. The SLIDSHAPs approach for unlabelled time series data.

vector of discrete values, i.e., $X_i(t_k) \in D_i$ and $|D_i|$ is finite. Using this notation, we define the "overlapping sliding windows" series $\{w_s\}_{s\in\mathbb{N}}$ as a series of time windows dependent on two parameters, i.e., the window length $d \in \mathbb{N}$ and the overlap $a \in \mathbb{N}$ among adjacent windows. Each window w_s contains d timestamps, it is written as

$$w_s = \{t_{s(d-a)}, \dots, t_{s(d-a)+d-1}\}, \tag{6.1}$$

and a is the number of timestamps lying in the overlap among adjacent windows, i.e., $|w_s \cap w_{s+1}| = a$ for all $s \in \mathbb{N}$. At the current timestamp t_T we have created

$$M(T) = \left\lfloor \frac{T-d+1}{d-a} \right\rfloor + 1$$

time windows.

Sliding windows are commonly used in concept drift detectors [BG07; Bae+06; Das+06]; however, most existing approaches focus on specific statistical features of the sliding windows, leading to an intrinsic inability to detect changes in feature correlations. We aim to create a feature extraction function over the sliding windows that outputs representative features with more easily detectable change points. The following sections introduce the SLIDSHAP series, a novel feature extractor for unlabeled streaming data based on sliding windows.

Algorithm 3 PSEUDO-CODE. SLIDSHAP series computation.

```
procedure SLIDSHAPs($X, d, a, T$)   ▷ $N$-dimensional time series $X$, sliding window length $d$, overlap among adjacent sliding windows $a$, current timestamp $T$
    $s \leftarrow 0$
    $S \leftarrow [\,]$
    while $s \leq \left\lfloor \frac{T-d+1}{d-a} \right\rfloor + 1$ do   ▷ sliding on windows
        for $i \in \{1, \ldots, N\}$ do   ▷ iterate over dimensions of $X$
            $w_s = \{t_{s(d-a)}, \ldots, t_{s(d-a)+d-1}\}$
            $S_i(s) = \phi(X_i^{w_s})$
            $S \leftarrow S_i(s)$
        end for
        $s \leftarrow s + 1$
    end while
end procedure
S   ▷ return the SLIDSHAP series
```

6.1.2 SLIDSHAP *series creation*

Given a multivariate time series X with N-dimensions, we can interpret the value $X_i(t_k)$ as the realization of a discrete random variable X_i; given the set of timestamps $\{t_0, \ldots, t_T\}$, the set $\{X_i(t_0), \ldots, X_i(t_T)\}$ contains $T+1$ independent realizations of the random variable X_i. Similarly, $\{X(t_0), \ldots, X(t_T)\}$ is the set of realizations of a N-dimensional discrete random variable. This interpretation allows us to study the correlations among the input variables of the time series and does not consider the temporal dependency among timestamps.

Given a game, Shapley values represents a way of fairly distributing resources among players [Sha+53] and, as already argued, can be used in contexts unrelated to interpretable machine learning. Given a set of players $\mathcal{N} = \{X_1, \ldots, X_N\}$ and a value function f, recall the Shapley values' definition (2.1) , i.e.,

$$\phi_f(i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus X_i} k_{\mathcal{A}} \cdot [f(\mathcal{A} \cup \{X_i\}) - f(\mathcal{A})] \tag{6.2}$$

where $k_{\mathcal{A}}$ depends on the number of players N and the size of the set $\mathcal{A}$ [Sha+53].

We proposed Shapley values within an unsupervised feature selection method in Chapter 4. Given a set of N discrete random variables $\mathcal{N} = \{X_1, \ldots, X_N\}$, we interpreted $\mathcal{N}$ as a set of players and encode the correlations within subsets of features of an unlabelled tabular data set with categorical entries using the Shapley values by using the total correlation as value function[1]; to simplify the notation we denote the Shapley values simply with $\phi(X_i)$. The proposed encoding enables extracting information from the data set based on the correlation structure. Features obtaining high Shapley values are highly correlated with subsets of other variables, while features with lower Shapley values are uncorrelated with the resting variables. We used the Shapley values to rank the features with respect to their ability to represent the correlation structure of the entire data set.

Now, we interpret the realizations of a time series on a time window $\{t_k, \dots, t_{k+d-1}\}$ as a discrete tabular data set with N columns and d rows; this allows us to compute a Shapley value for each column, i.e., for each univariate dimension of the time series using the total correlation. We aim to detect change points appearing in the input dimensions of the time series. We trace the distributional changes during the time using the Shapley values of the time series' input variables when restricted to the sliding windows $\{w_s\}_{s \in \mathbb{N}}$ from Equation (6.1); in each window w_s, we deal with d observations of the N-dimensional random variable X, thus working with a discrete (categorical) tabular data set with N columns and d rows. For each dimension X_i, we get the Shapley value $S_i(s) = \phi(X_i^{w_s})$, i.e., the Shapley values of the input variable X_i when we restrict the observations to the time window w_s. $\phi(X_i^{w_s})$ considers the correlations of X_i with the other dimensions X_j of the time series in w_s; for each w_s, we obtain a N-dimensional real-valued vector $S(s) = [S_1(s), \dots, S_N(s)]$ and refer to it as the slidSHAP *value*; we define the sequence $\{S(s)\}_{s \in \mathbb{N}}$ the slidSHAP *series*.

In Section 6.1.1, we have introduced the time-dependent sliding windows $\{w_s\}_{s \in \mathbb{N}}$; the slidSHAP series inherits the time-dependency from the windows and not the same time notion as the original time series. Figure 6.1 represents a visual schema for the slidSHAP series construction process while Algorithm 3 shows the pseudo-code; the implementation is available online[2]. We extrapolate information about the input dimensions correlation structure from the original discrete time series X with discrete finite values and transfer the change point detection problem to a new N-dimensional real-valued series. We interpret the slidSHAP series as a projection of the time-dependent correlation structure of the original time series. Note that the sliding windows are partly overlapping. Given two indices i, j, the intersection $w_i \cap w_j$ is non-empty if they are sufficiently close; hence, the information covered by $S(i)$ and $S(j)$ relates to X on partly overlapping time windows.

[1] The definition reads $f(\mathcal{A}) = H(\mathcal{A}) - \sum_{X \in \mathcal{A}} H(X)$ where $H(\cdot)$ is the discrete Shannon entropy.

[2] Code available at `chiarabales/slidSHAPseries`

The setup of the parameters a and d is essential to modulate the *granularity* of the SLIDSHAP series.

6.1.3 *Change point detection*

The SLIDSHAP values are based on the distributions and the correlations among the univariate dimensions of the time series and are label-independent. When dealing with real-world time series, often only a few input variables are subject to distributional changes; however, those distributional changes could affect the correlation structure of the whole input variables. Moreover, the change points are often hardly visually detectable, especially when they do not directly affect specific statistical features in which the variables vary.

We expect that distributional changes in the input variables modify the correlation structure of X on the sliding time windows, and the SLIDSHAP values encode the correlations among the univariate dimensions of X through the time; eventually, the SLIDSHAP series reflects these distributional changes, allowing us to use it to detect change points in X. We target change point detection using statistical tests under the assumption of i.i.d. of the SLIDSHAP values. As well as on the original time series X, the SLIDSHAP series is unlabeled; therefore, we have only access to its dimensions' distributions to detect change points. We employ two statistical tests to look for distributional changes on each dimension S_i

- the **Student's t-test**
- and the **Kolmogorov-Smirnov test** (or KS test) [Con99; Rei+16].

We analyze their performances in the empirical evaluation. Both tests check whether statistical significance exists for two samples drawn from the same distribution. We consider a reference sequence of SLIDSHAP values S_{ref} of length m ending at $s \in \{0, \dots, M(T)\}$ and a new sequence S_{new} with length n precedent to it, i.e.,

$$\mathcal{S}_{\text{new}} = \{S(s) \mid s \in \{s-n+1, \dots, s\}\} \tag{6.3}$$

$$\mathcal{S}_{\text{ref}} = \{S(s) \mid s \in \{s-m-n, \dots, s-n+1\}\}; \tag{6.4}$$

We call the two data sequences of SLIDSHAP values *buffers*. F_{ref} and F_{new} are respectively the empirical cumulative distribution functions of $\mathcal{S}_{\text{ref}}$ and $\mathcal{S}_{\text{new}}$ and we test whether there is statistical significance of $\mathcal{S}_{\text{new}}$ and $\mathcal{S}_{\text{ref}}$ to be drawn from the same probability distribution function. The user can define the sizes of the new and reference buffers.

We deal with multiple testing corrections by performing a number N of statistical tests, i.e., one for each univariate dimension of the SLIDSHAP series. Among the various multiple hypothesis corrections available, we choose to stick to the Bonferroni correction [BA95; BH95; RGL19], where the null hypothesis is rejected if the minimum p-value among all N tests is smaller than $\frac{\alpha}{N}$. For each $s \geq \min\{m, n\}$, we conduct a set of such dimension-wise statistical tests and compare the performances using the two statistical tests in Section 6.3.8. For drift detection, statistical tests are commonly applied on the original time series [Das+06; Rei+16; YWP18]; the inventive step we have introduced is detecting the change points in the SLIDSHAP series and then relocating them to the original timestamps of the time series data. The following section presents how to transfer the detected change points to the original timestamps.

6.1.4 *Change points aggregation and re-location*

Due to the construction of the windows, each change point in X is covered by multiple sliding windows. Hence, we need to aggregate the detected change points on the SLIDSHAP series to rebuild the single change event on X; the change point detection over the SLIDSHAP series results in alarms on the corresponding sliding windows, not single timestamps. Our ultimate goal is to relocate the change point positions to the timestamps where they initially took place. In this section, we introduce aggregation and re-location.

The sliding windows have length d and overlap a; given a change point t_{change} in X, the windows containing information about t_{change} are $\left\lfloor \frac{d}{d-a} \right\rfloor > 1$ and the corresponding change points in the SLIDSHAP series is going to be tested in

$$\left\lfloor \frac{d}{d-a} \right\rfloor + m + n$$

statistical tests by moving $\mathcal{S}_{\text{ref}}$ and $\mathcal{S}_{\text{new}}$ one step forward a time on each univariate dimension. Being aware that it is less likely to detect statistical significance for the presence of distributional change when testing the first and last SLIDSHAP values involved in the change event, we first detect alarms on each univariate dimension of the SLIDSHAP series then we aggregate the alarms using multiple testing corrections. At this point, for each window w_s, we have detected an aggregated p-value; due to the dilating effect of change events in SLIDSHAP series, we check for sequences of p-values being below the significance level α, i.e., we trigger the change point alarm if and only if we find a continuous sequence of $(m+n) \cdot \gamma$ corrected p-values below α. The parameter $\gamma \in \mathbb{R}_+$ scales the number of sequential corrected p-values to be below α before producing an alarm on X and typically ranges in $\left[\frac{1}{2}, 1\right]$; when $(m+n) \cdot \gamma$ rejections of the null-hypothesis are detected in sequence, we get an alarm at the current time window w_T. Finally, we relocate the last timestamp $\tilde{t}_{\text{change}}$ of w_T as the corresponding change point in the original time series data X.

We underline that we only consider abrupt drifts in the time series, i.e., the distributional changes happen in specific timestamps that need to be located. The dilating effect makes change points in X incremental changes in the SLIDSHAP series. The same holds for gradual and incremental drifts in the original time series, such that our model is easily extendable to non-abrupt concept drifts.

6.2 RUN-TIME ANALYSIS

The SLIDSHAP series computation inherits the exponential runtime from the Shapley values computation. From a complexity analysis point of view, the computation of the SLIDSHAP series has a complexity of $\mathcal{O}\left(d \cdot 2^N \cdot \frac{T-d}{d-a}\right)$ where T is the number of instances to be processed, N is the number of input variables, d and a are the length and overlap of the windows. The complexity $\mathcal{O}\left(d \cdot 2^N\right)$ derives from the Shapley values' computation [Bal+22]. However, several approximation techniques can be applied [CGT09; Mit+22; Cam+18; BC21; CKL22] thus accelerating the computation of the entire SLIDSHAP series to polynomial time $\mathcal{O}\left(d \cdot N \cdot \frac{T-d}{d-a}\right)$. On the other hand, multivariate time series data in real-world scenarios usually either do not contain a large number of dimensions or are easily reduced to a lower dimensionality [Hal+21; Qah+15].

6.3 EMPIRICAL EVALUATION

We evaluate SLIDSHAPs on both change point detection and visualization of correlation changes. We compare our model against a set of selected representative unsupervised concept drift detection methods. We consider several synthetic and real-world time series datasets with discrete values. We use ground-truth labeling for the allocation in time of the concept changes. We acquire a binary set of change point alarms after fixing the significance level α, and we compare them with the real change point timestamps. In summary, we evaluate (1) the change point detection performance and delay in Section 6.3.5, and (2) the visualization of the change events in the SLIDSHAP series in Section 6.3.6.

	instances	variables	discrete values	change points
LED	90000	7	2	9
ADD	30000	5	10*	5
MUL	30000	5	10*	5
COE	30000	5	10*	5
AND	30000	5	10*	5
OR	30000	5	10*	5
XOR	30000	5	10*	5
MIX	30000	10	10*	5
BC	286	9	2∼11	3
PH	33659	10	4∼13	5
KDD	16000	40	1∼7	7
MSL	9809	54	1∼2	4

Table 6.2: DATASETS SUMMARY. The number of discrete values indicated with $*$ refers to the range in which the independent input variables are sampled; the other variables' ranges depend on the type of correlation with the ones sampled.

6.3.1 *Datasets description*

We created synthetic datasets with correlation changes in the input variables. We construct various concepts, i.e., distributions and correlations among the input variables, and concatenate them at specific timestamps t_{change} such that the input variables follow a correlation structure till t_{change} and another from $t_{\text{change}} + 1$ for each change point. The various correlation changes in the input variables are meant to evaluate the sensitivity of SLIDSHAPs in detecting different kinds of correlation changes. We constructed two types of synthetic datasets: the first type includes datasets with only 5 features, where the correlations are of a specific type; the second type includes one dataset with random types of correlation changes at the change points and contains 10 features. Each synthetic dataset is constructed as follows: we generate 6 different distributions, each containing 5000 instances, and concatenate them to simulate 5 change points.

I TYPE DATASETS. For each distribution, we select 2 or 3 variables to be involved in the change event, while at least one variable always follows the same distribution. Each random sampled variable varies in the set of integers $\{1, \ldots, 10\}$. In ADD, initially $X_3 = X_1 + X_2$, while X_1, X_2, X_4 are independently randomly sampled; after each change point, the relation among X_1, X_2, X_3 changes, e.g., to $X_2 = X_1 + X_3$. MUL contains only multiplicative relation among X_1, X_2 and X_3; and changes in a multiplication relations, e.g., $X_3 = X_1 \cdot X_2$. In COE, we included linear combinations, e.g., $X_3 = c_1 X_1 + c_2 X_2$ where c_1 and c_2 assume various values. Furthermore, we create some datasets, including logical feature correlations. In AND, OR and XOR, one binary variable depends on the value of the other two variables; after fixing a value c, we include correlations of the type

- $X_3 = \mathbb{1}[(X_1 > c)\&(X_2 > c)]$ in AND,
- $X_3 = \mathbb{1}[(X_1 > c) \mid (X_2 > c)]$ in OR,
- $X_3 = \mathbb{1}[(X_1 > c) \neq (X_2 > c)]$ in XOR

where $\mathbb{1}$ is the usual Boolean function that returns 1 in the case the condition is satisfied and returns 0 otherwise.

II TYPE DATASETS. The dataset MIX contains 10 input variables where X_1, X_2, X_3 are randomly sampled from $\{1, \ldots, 10\}$. For each distribution, all the other variables are constructed using one randomly chosen correlation function of X_1, X_2, X_3 among "addition", "multiplication", "linear combination", "and", "or" and "xor". Thus, MIX contains a mixture of the correlation above changes. In addition to these synthetic datasets, we also consider one commonly used synthetic dataset in literature, containing changes in the data distribution instead of explicitly in feature correlation; the LED dataset [Fri+14; PVP18] is a commonly used synthetic dataset describing the digit displayed on a seven-segment LED display. A binary 7-dimensional binary vector represents a digit; it contains a total of 9 change points, one for every 10000 instance. Each subset contains the vectorial representations of the 10 single digits except one; the change events consist in changing absent digits.

REAL-WORLD DATASETS. Finally, we included some real-world datasets, interpreting them as time series data and including change points by concatenating different subsets [Ho05]. As real-world datasets, we used the following publicly available categorical datasets: the *Breast Cancer dataset* (BC) [DG17], the *Poker Hand dataset* (PH) [DG17], the *KDD Cup 99 dataset* (KDD) [DG17] and the *Mars Science Laboratory dataset* (MSL) [Hun+18]. BC contains purely categorical features describing breast tumors of patients. We concatenate subsets of patients in different age groups to simulate concept drifts. PH contains one million randomly drawn poker hands. Five features describe the 4 possible suits and another five features describe the 13 possible ranks. We create virtual drift as in [Bif+13] by sorting the ranks and suits and taking a subset with 33659 instances for our experiments. KDD contains both numerical and categorical features describing instances of network intrusion records. We use all features in our experiments and discretize the numerical features into five categories. A subset with 16000 instances from HTTP and SMTP services is selected, and concatenating instances create the concept drifts from the two services.

Finally, MSL contains telemetry data from the NASA Curiosity Rover on Mars. We discard the numerical telemetry value and only consider the 54 remaining binary features. Data are collected from different channels, and we consider the channels' changes as concept drifts.

6.3.2 *Evaluation metrics and competitors*

We evaluate the performance of slidSHAPs compared to competitors for unsupervised concept drift detection. The actual change points' timestamps t_{change} are known in each dataset, while t^*_{change} represent the detected change point's timestamps. Following Pesaranghader et al. [PV16; PVP18], we introduce an *acceptable delay length* Δ to determine the true positive TP, false positive FP, and false negative FN detected change points. Whether t^*_{change} is a TP, FN, or FP is determined by the relative position of the labeled change point t_{change} and the detected position t^*_{change}. A change point alarm is a TP if it belongs to the interval set $\{t_{\text{change}}, \ldots, t_{\text{change}} + \Delta\}$, i.e., the delay characterizing its detection is smaller or equal to the accepted delay Δ.

As evaluation metrics, we used some change detection performance metrics, such as precision, recall, and F1-score, and the *average delay*, defined as

$$\text{avgDELAY} = \sum_{\text{change point} \in \text{TP}} \frac{t^*_{\text{change}} - t_{\text{change}}}{\text{number of TP}}. \tag{6.5}$$

We compare slidSHAPs with the principal unsupervised and classifier-free drift detection approaches, e.g., HDDDM [DP11], ADWIN [BG07], PCA-CD [Qah+15] and Kdq-Tree [Das+06] and reported their performances. Various univariate unsupervised drift detectors exist based on statistical features or distribution discrepancies. Among them, PCA-CD [Qah+15] detects drifts computing the divergence metrics on the data's principal components. First, the principal components are computed on a reference window, and samples from a new window are projected onto them. The result of the divergence metric between the reference and test window scores is used as a discriminator factor: if a fixed threshold is reached, a concept drift is detected. Based on the Kullback-Leibler divergence, the Kdq-Tree concept drift detection method [Das+06] partitions data via constructing a Kdq-Tree; the output score of the comparison between the reference window and the test window is also here compared to a threshold. Finally, ADWIN [BG07] uses sliding windows to detect changes by keeping updated statistics.

The discriminator factor here is the difference among the averages of the collected statistics over the reference and new buffers; the obtained score is compared against a threshold. However, all these univariate drift detector methods often fail to detect correlation drifts without significant deviations in the specific statistical features they track (e.g., mean and variance, among others). We only report the results of ADWIN to represent their similar performance.

6.3.3 *Parameters setups*

HDDDM is a batch-based approach. We set the data batch size to be two time windows for each dataset. Looking at the performances, we noticed that HDDDM showed a generally large delay in detecting changes; therefore, for HDDDM, we implement a relaxation of the criterion, such that true positives are detected using $\Delta = 10 \cdot d$. ADWIN works uniquely by detecting drifts on univariate time series. Therefore, we train one model for each input data dimension and let them run in parallel for the N dimensions of the time series data. A change point alarm is triggered if one change point is detected on at least one dimension. For the other competitors KDQ-TREE and PCA-CD, we use the default parameter setting from the GitHub implementation[3].

6.3.4 *Experiment setup*

In the statistical test, we set the reference and the new buffer sizes $m = n = 10$ and $\gamma = 1.0$, such that the statistical tests are based on sufficient data instances while keeping the prediction delay low. The significance level is set to $\alpha = 0.01$ and the *acceptable delay length* Δ to $5 \cdot d$ where d is the sliding window length, namely all change point alarms within 5 window size after the real change point are considered as true positives. The parameter setups for the competing methods are summarized in Setion 6.3.3. In all experiments, we implemented the t-test and the KS-test; by default, we opted to report the results obtained using the t-test. Section 6.3.8 compares the two tests.

In SLIDSHAPs, the two parameters window length d and overlap a influence the construction of the sliding windows and are set through an empirical evaluation of the data stream; we conducted experiments using $\{10\%, ..., 90\%\}$ as nine different overlap rates $\frac{a}{d}$ and fixed windows length d. The empirical evidence suggests keeping the overlap rate in the range $50 - 80\%$. Further details can be found in Section 6.3.7. In the experiments, we constructed the

[3] mitre/menelaus

		slidSHAPs			HDDDM			ADWIN		
		P	R	F1	P	R	F1	P	R	F1
synthetic	LED	0.333	0.667	0.444	0.063	0.556	0.114	0.111	0.111	0.111
	ADD	0.800	0.800	0.800	0.200	0.200	0.200	-	0.000	-
	MUL	0.571	0.800	0.667	-	0.000	-	-	0.000	-
	COE	0.800	0.800	0.800	0.111	0.200	0.143	-	0.000	-
	AND	0.667	0.400	0.500	0.222	0.400	0.286	-	0.000	-
	OR	1.000	0.400	0.571	0.167	0.200	0.182	-	0.000	-
	XOR	1.000	0.400	0.571	0.250	0.200	0.222	-	0.000	-
	MIX	0.714	1.000	0.833	0.143	0.400	0.211	-	0.000	-
realworld	BC	0.750	1.000	0.857	1.000	0.667	0.800	1.000	0.667	0.800
	PH	1.000	0.400	0.571	0.133	0.400	0.200	-	0.000	-
	KDD	0.800	0.571	0.667	0.350	1.000	0.519	0.064	1.000	0.121
	MSL	0.500	0.250	0.333	0.333	0.250	0.286	0.667	0.500	0.571

Table 6.3: Performance summary - precision, recall and F1 score. For each dataset, the largest F1 score is highlighted. In the gray-shaded area, ADWIN can not detect any change point.

slidSHAP series using a fixed window length $d = 100$ and overlap $a = 70$; for the small dataset BC instead, we used $d = 10$ and overlap $a = 8$. For the real-world high-dimensional datasets KDD and MSL, we computed the slidSHAP series using the approximation described in Section 6.3.10.

6.3.5 *Overall performance*

Tables 6.3 and 6.4 show the overall performance comparison. The slidSHAPs outperform the competitors with respect to the F1 score in all datasets except MSL; slidSHAPs also show dominating performance on average delay in most synthetic datasets. Moreover, slidSHAPs appears more sensitive in detecting different types of correlational changes than the other distribution-based detectors. Since there is no change in the mean value, ADWIN fails to find any change point in each synthetic dataset except LED, i.e., where the recall equals 0. On the other hand, ADWIN outperforms the slidSHAPs on the MSL dataset, which inherently contains correlation and distributional changes with respect to other statistical features. HDDDM only shows comparable results on BC and performs significantly worse on other datasets.

Regarding the average delay of the various methods, ADWIN predicts every incoming instance, generally showing a low average delay. Instead, the slidSHAPs detect the change on every incoming slidSHAP value, which intrinsically has a delay given by the sliding windows of length d; this mechanism causes our approach

		slidSHAPs	HDDDM	ADWIN
synthetic	LED	424 ± 47	199 ± 0	167 ± 0
	ADD	384 ± 55	599 ± 0	-
	MUL	369 ± 70	-	-
	COE	399 ± 40	559 ± 0	-
	AND	359 ± 0	599 ± 0	-
	OR	299 ± 0	559 ± 0	-
	XOR	329 ± 0	399 ± 0	-
	MIX	399 ± 91	199 ± 0	-
realworld	BC	37 ± 2	34 ± 18	16 ± 16
	PH	382 ± 10	427 ± 235	-
	KDD	455 ± 50	143 ± 90	36 ± 11
	MSL	487 ± 0	117 ± 0	237± 96

Table 6.4: Performance summary - average delay. The minimum detection delay in each dataset is highlighted. In the gray-shaded area, ADWIN can not detect any change point.

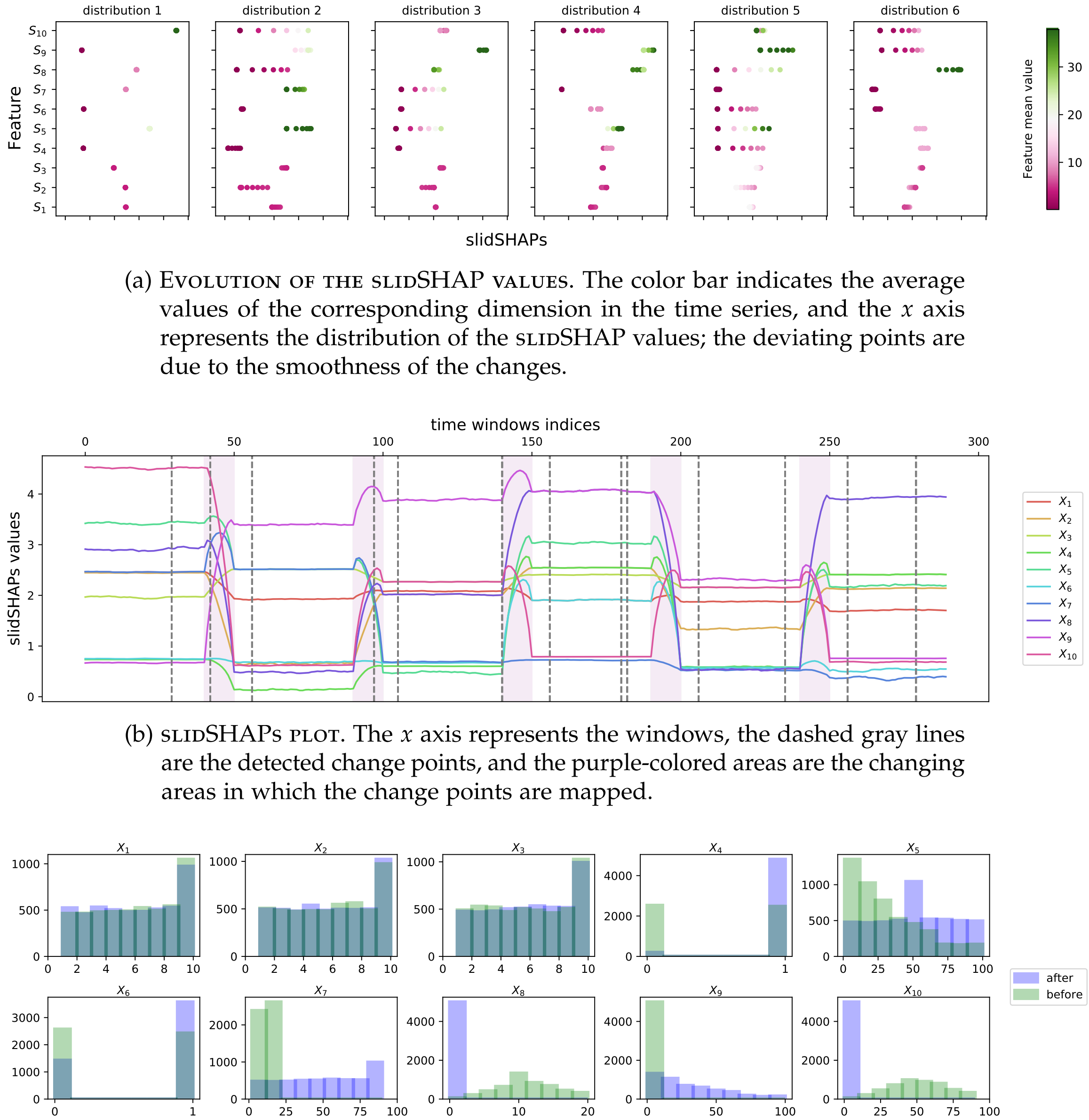
to detect change points with a larger delay. HDDDM waits for each batch of data; consequently, it shows the largest average delay.

6.3.6 slidSHAP *series analysis and visualization*

After fixing the length and the overlap among the sliding windows, the slidSHAP series represents the correlations among univariate dimensions of the time series; the univariate dimensions of the slidSHAP series follow more distinguishable trends than the original time series. Although we do not claim that slidSHAPs is an interpretable feature extraction approach, the slidSHAP series gives a visual hint to users on where the change points could potentially be located before statistically checking for their existence.

We used the MIX dataset for exploring the slidSHAP series ($d = 1000$ and $a = 900$) as a visualization tool; Figure 6.2a, 6.2b and 6.2c show the behavior changes in the slidSHAP series before and after distributional changes. In Figure 6.2b, solid lines are the univariate dimensions of the slidSHAP series and the purple areas are the windows in which the concept changes are mapped using the slidSHAP. Note that the MIX dataset only contains abrupt distributional changes. However, the slidSHAP series shows smooth changes between one distribution and the next: abrupt changes are expanded in the slidSHAP series to multiple subsequent time windows.

Furthermore, as the slidSHAP values are an aggregation of the correlation structure in subsets of the time series dimensions, changes in the slidSHAP series dimensions are observed for all



(a) Evolution of the slidSHAP values. The color bar indicates the average values of the corresponding dimension in the time series, and the x axis represents the distribution of the slidSHAP values; the deviating points are due to the smoothness of the changes.

(b) slidSHAPs plot. The x axis represents the windows, the dashed gray lines are the detected change points, and the purple-colored areas are the changing areas in which the change points are mapped.

(c) Distributions before and after the first change point in the original time series data distributions.

Figure 6.2: MIX dataset visualization with $d = 1000$, $a = 900$.

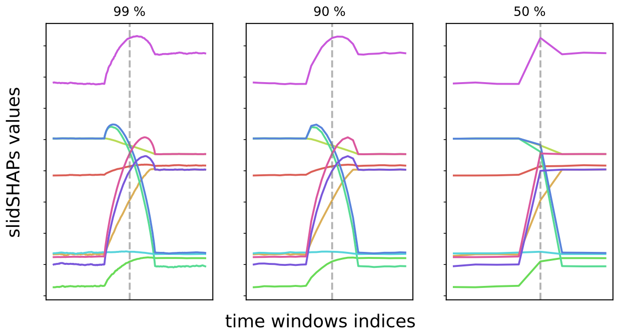


Figure 6.3: MIX DATASET - FIRST CHANGE POINT. Solid lines are the univariate dimensions of the slidSHAP series ($d = 1000$, $99\%, 90\%, 50\%$ overlap rate); dashed grey lines indicate the change points.

the dimensions and also the ones not affected by significant modification drifts (e.g., X_1, X_2 and X_3). Figure 6.3 shows the evolution of the SLIDSHAP values around the first change point in the MIX dataset. We keep using $d = 1000$ and vary the overlap rate $\frac{a}{d}$ among the sliding windows in $\{99\%, 90\%, 50\%\}$. Although the overall behavior of the SLIDSHAPs is similar using the various parameter settings, a difference in the smoothness in the SLIDSHAP series in the changing area is immediately noticed. The setting of the parameters d and a also influences the computation time of the whole SLIDSHAP series (cf. Section 6.2). It is worth noting that, to compute the Shapley values using the total correlation, the number of instances per time window should not be too low, e.g., under 100 instances. On the other hand, as the KS-test checks for samples drawn from equal distributions, if the changes in the SLIDSHAPs are too smooth, i.e., highly overlapping sliding windows, the KS-test loses statistical power, provoking a higher number of false positives. Selecting a balanced ratio among a and d (cf. Section 6.3.7), we can maximize the KS-test's statistical power to get the highest accuracy.

Following the style of [LL17], Figure 6.2a represents how the SLIDSHAP values change in the different distributions. We plotted the SLIDSHAP values against the average value assumed by the original time series univariate dimensions. Intuitively, a distributional change in the input space causes a change in the SLIDSHAPs value, which can be detected as a distributional change. The SLIDSHAP series also show some unobservable input space change points, where the amplitude of features stays in the same range while the feature correlation changes. In such cases, significant changes can still be observed in the SLIDSHAPs values. The changes in the time series are highlighted in the SLIDSHAP series and the evolution of

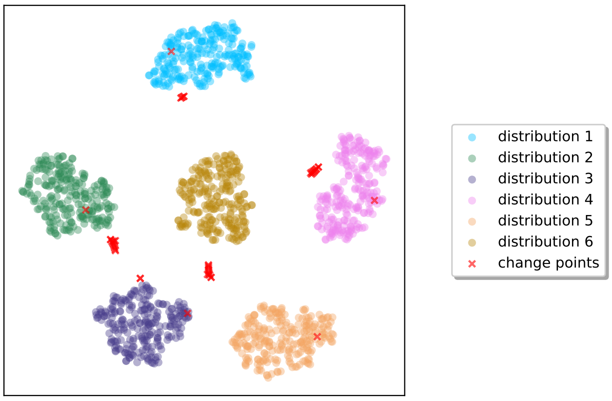


Figure 6.4: MIX DATASET - SLIDSHAP TSNE PLOT. A color per concept, while the red crosses are the SLIDSHAPs of the change events.

its univariate dimensions can be used to simplify the detection of such change points in the time series data.

In Figure 6.2c, we plot the different distributions of the time series data before and after the first change for each univariate dimension of the time series X. Finally, Figure 6.4 visualizes the 10-dimensional SLIDSHAP values of the MIX dataset in a two-dimensional space using tSNE. As depicted by the color-coded dots, data from different distributions are well-separable using their SLIDSHAP series. The SLIDSHAP values, whose corresponding sliding window overlaps the change position, i.e., red crosses, lie mostly apart from any cluster. Some are not well-distinguishable from the clusters, as they correspond to the change event's beginning and ending sliding windows and, therefore, do not show a significant difference to the previous or upcoming distribution.

6.3.7 *Parameter sensitivity*

The window length d and the overlap a are the two influential parameters for constructing the sliding windows in SLIDSHAPs. In practice, the sliding window length can be determined empirically by the dataset's size or prior knowledge of the data, e.g., an hour, a day. Moreover, the overlap a directly impacts the generated SLIDSHAP series. We conducted experiments using $\{10\%, ..., 90\%\}$ as nine different overlap rates $\frac{a}{d}$ and same window length d as in Section 6.3.5. Figure 6.5, shows the counts of TP, FP, FN, and the average delay for the various setups. Generally, the model detects more TP with increasing overlap rate, i.e., fine-granular SLIDSHAP series. However, the FP also explosively increases due to the enormous increase of SLIDSHAP values under a high overlap rate. Exploring using the different datasets introduced through the

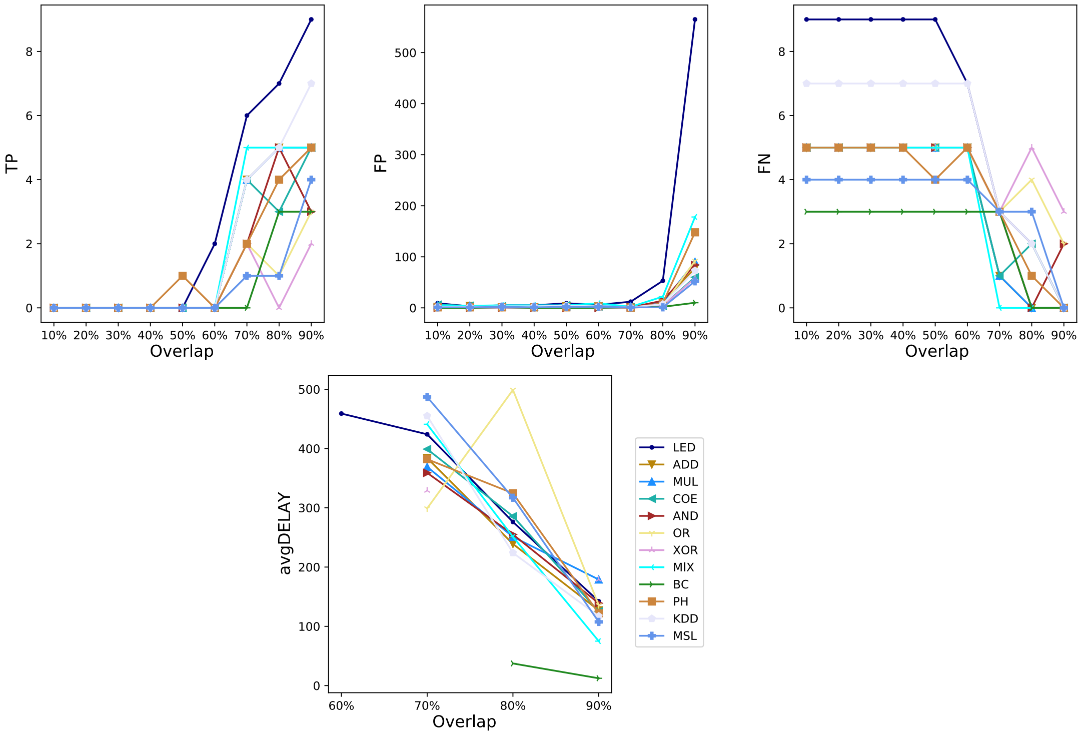


Figure 6.5: Parameter sensitivity. We study the impact of the overlap rate by fixed window length $d = 100$ ($d = 10$ for BC).

paper, we conclude that it is often reasonable to keep the overlap in the range $50 - 70\%$; furthermore, the window length can be set up differently to detect change points located with various interspaces among each other. Finally, the average delay fairly reflects that with a larger overlap rate, we need to conduct the statistical tests more often. Therefore, it ends up with less detection delay; we do not observe average delay rates below 60% due to the absence of true positives.

6.3.8 *KS-test versus t-test*

We tested our method using the KS-test and the t-test.

Student's t-test. We restrict to the case when dealing with two separate sets of independent and identically distributed samples, thus looking for statistical significance of equality for one variable from each population. The two-sample t-test takes the null hypothesis that the means of two populations are equal. The t-test assumes that the variances of the two populations are equal (although this assumption can be dropped).

Kolmogorov-Smirnov test. The KS-test is a non-parametric and distribution-free statistical test to compare continuous one-dimensional probability distributions. In the case of the *two-sample*

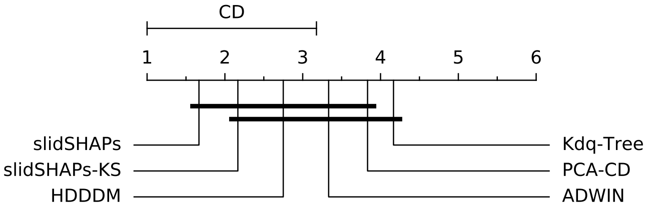


Figure 6.6: CRITICAL DIFFERENCE DIAGRAM. F1 score comparison using the Nemenyi test with a 95% confidence interval; lower ranks are better.

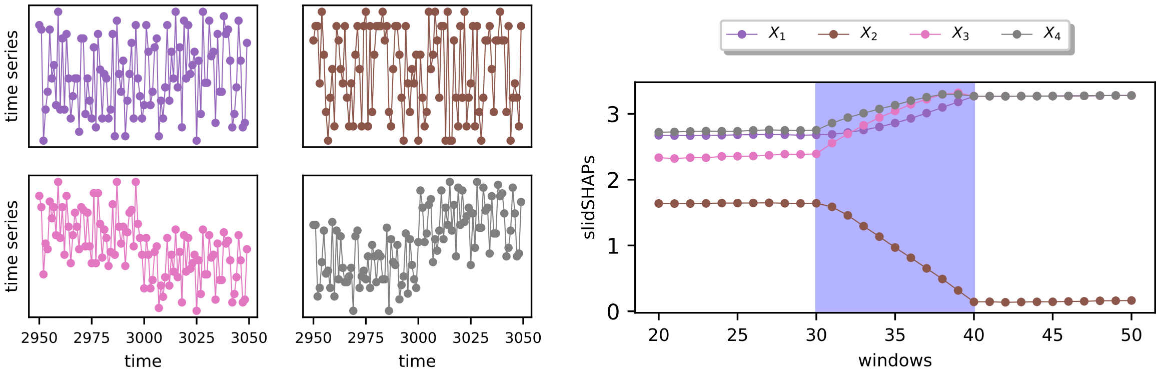


Figure 6.7: VISUALIZATION OF CONCEPT DRIFTS. On the left, 4-dimensional time series example with a change point at timestamp 3000. On the right, our SLIDSHAPs with blue-colored concept drift.

KS-test, it can be used to compare two sample sets. Given two samples and their empirical cumulative distribution functions F_1 and F_2, the KS-test assumes as null hypothesis that the two samples are drawn from the same distribution; thus, given a significance level α, the null hypothesis is rejected if

$$\underset{x}{sup}|F_1(x) - F_2(x)| > c(\alpha)\sqrt{\frac{m+n}{m \cdot n}} \tag{6.6}$$

where $c(\alpha) = \sqrt{-\ln(\frac{\alpha}{2}) \cdot \frac{1}{2}}$, m and n are the sizes of the two sample sets.

We replaced the default t-test with the KS-test (significance level $\alpha = 0.05$) in SLIDSHAPs. We refer to the new variant as SLIDSHAPs-KS and report the average ranking of F1 scores among competitors on all datasets. As shown in Figure 6.6, the SLIDSHAPs (with t-test) and SLIDSHAPs-KS (with KS-test) rank in first places.

6.3.9 *A remark on the change point visualization*

Visualization tools help the human eye understand what is happening in the data; having a visual grasp helps users to validate their assumptions and the outcomes of black-box models. However, the visualization of the time series' evolution through time is often not helpful: the human eye can not easily grasp the structure and the correlation changes among several observations at each timestamp. After fixing the length and the overlap among the sliding windows, the slidSHAP series represents the correlations among univariate dimensions of the time series. Although we do not claim slidSHAPs to be an interpretable feature extraction approach, the univariate dimensions of the slidSHAP series follow more distinguishable trends than the input variables of the original time series. To this end, Figure 6.7 gives an example of the contrast between the pretty chaotic behavior of the time series data and the smooth sequence of the slidSHAP values. Figure 6.7 shows a 4-dimensional time series with a distributional change at timestamp 3000. The change is hard to spot in the original time series. However, it is clearly visible in the lower plot where we represent how the correlation structure evolves through our slidSHAPs. It is clear that the correlation structure among the dimensions is strongly affected by the change point in a few dimensions. Such complex behavior of multivariate time series makes visualization of streaming data hard to interpret. Similarly, change detectors fail to analyze complex correlation structures among the time series input dimensions. Furthermore, all the slidSHAP series dimensions change when a change point appears in the time series input variables. We can visually distinguish distributional changes of the original time series in the slidSHAP series. The slidSHAP values can give a visual hint to users on where the change points could potentially be located before statistically checking for their existence.

6.3.10 slidSHAPs *approximations*

The computation of Shapley values is a well-known NP-hard problem that involves evaluating the value function on each possible subset of players. The exact computation of Shapley values becomes soon unfeasible due to the exponentially growing number of evaluations involved. Several approximations appeared in the literature. In Chapter 4, we suggested that defining an "upper bound" for the subsets' size on which the value function will be evaluated achieves better results than using randomly sampled

	2	3	4	5	6	7	8	9
LED	318.9	611.0	945.8	1223.1	1283.7			
ADD	50.3	63.8	68.9					
MIX	145.4	433.2	852.4	1344.8	1725.7	1860.6	1987.6	2046.0
AND	51.8	61.5	65.9					
COE	50.4	63.0	68.5					
MUL	50.3	61.0	68.5					
OR	49.8	60.8	67.6					
XOR	51.0	60.6	65.9					

Table 6.5: Runtime. Average runtime in seconds for the slidSHAP series ($d = 100$ and $a = 70$) over 10 trials. Each column represents a different upper bound.

subsets; we implemented the same approximation for the computation of our slidSHAP series. The upper bound defined by the user influences the quality of the approximation for the Shapley values; for a N-dimensional time series, an upper bound equal to N represents the non-approximated computation of the Shapley values. We generally use the non-approximated version of the slidSHAPs, except in the two high-dimensional real-world datasets KDD and MSL, where we use the approximated version with an upper bound equal to 2.

We conducted experiments using various upper bounds on the synthetic datasets to evaluate the runtime of the slidSHAP series' computation; Table 6.5 contains the runtimes in seconds to compute the slidSHAP series for $d = 100$ and overlap $a = 70$ when using the different upper bounds and for the various synthetic datasets. All the experiments were run on Intel Xeon CPU E5-2640 v4 @ 2.40GHz with 10 cores.

Part II

ABOUT TRUSTWORTHINESS OF EVALUATIONS AND METHODS

7 INTRODUCTION AND RELATED WORK

The spread of machine learning techniques to safety-critical applications has raised major concerns about model interpretability, fairness, and trustworthiness. From an ethical perspective and socially acceptable guarantees of well-used technologies, ensuring that metrics, methods, and technologies are fair and interpretable became necessary. The use of black-box models for critical applications, e.g., health and societal impactful applications, is limited by the lack of trust and understanding of the (over)complicated machine learning models. Although pretty obsolete and often outperformed by more advanced methods, technologies such as simple linear models are still used daily because of their clarity, straightforwardness, and ease of use. Increasing trust and understandability for the general public is, therefore, essential. Furthermore, consistent evaluations are often far from achieved. The results reported by practitioners in statistics and machine learning are mostly data- and metrics-dependent; this renders the correct assessment of the methods' performances, the reproducibility, and the credibility of the reported achievements complex.

In the second part of this thesis, we will focus on the *consistency* of a specific category of metrics and interpretability methods and on proving a new intepretable method for unsupervised anomaly detection. The trustworthiness of machine learning is fundamentally based on the understanding and the reliability of methods. The Oxford dictionary defines "trustworthiness" as "the ability to be relied on as honest or truthful". Can we rely on the results of a machine learning method? (Probably) yes, if it is transparent, fairly tested, understandable, and coherent. Trust is hard to evaluate, and building trust in new technologies is often tackled collaboratively by machine learning researchers and psychologists. Assuring increased fairness, transparency, coherence, and interpretability of methods boosts the trust in black-box models.

Here, we follow the common thread on rankings from the perspective of results' consistency. We consider the consistency of rankings' evaluation metrics and the consistency among relative importance scores derived from saliency explanations for time series classification. Furthermore, we provide a new application of Shapley values for interpreting bagging models-based anomaly detectors. The following three chapters are based on published or under-review work as cited below:

- Chapter 8, "A group-theoretic perspective on ranking evaluation metrics" is based on [BMM24],
- Chapter 9, "On the consistency and robustness of saliency Explanations for time series classification methods", is based on the collaborative work [BLM23a], and
- Chapter 10, "On the efficient Explanation of Outlier Detection Ensembles through Shapley values ", extends [KBM24].

We now proceed with an overview of the related work relevant to all chapters.

7.1 RELATED WORK

Costlier, more complicated, advanced machine learning prediction models often outperform straightforward linear regression methods. This trade-off between intepretability and model performance has increased the interest in post-hoc explanations techniques, ideally model-agnostic, straightforward, and robust. Few exceptional models are inherently interpretable. Examples are shallow models; although broadly applied, they often lack accuracy compared to more sophisticated prediction models whose predictions' interpretation is far from obvious. Among explanation methods, saliency explanation recently gained traction and succeeded in various computer vision [Pil+22; SGK17] and natural language processing tasks [TYR23; Rus19]. We have already recalled SHAP [LL17], a successful attempt to introduce Shapley values [Sha+53] in machine learning aiming at assigning importance scores to features for local explanations of black-box models' predictions. On the one hand, Shapley values offer mathematical guarantees of fairness that make them an attractive choice in many contexts; on the other hand, their practical application poses a significant challenge due to the requirement of training an exponentially large number of models.

Explaining time series models and outlier detection methods still faces challenges. Time series data structure has generated interest in directly applying saliency maps to obtain meaningful explanations for classification models [Ism+20]. However, images and time series represent fundamentally different types of data. On the one hand, the temporal dependency in time series leads to time-dependent changes in feature attribution. On the other hand, explanation approaches are often not directly applicable to time series models with recurrent- or attention-based components [Cho+16; JW19]. Deep time series analysis models usually consider sliding windows as basic input units to capture temporal information. Analog to saliency maps that visualize image pixel importance, similar saliency maps are generated for time series frames with *feature-time pixel* importance. Current research on explanation approaches for time series data can be classified into two categories. The first category contains methods treating sliding windows as *frames* of images and applying classical image explanation methods, e.g., SHAP [LL17], LIME [RSG16], and DeepLIFT [SGK17]. Those methods extract local feature information while disregarding the time series data's typical time structure. The second category contains methods considering the time dimension as an additional feature for joint explanation [Ben+21b; Ism+20]. Another thread of approaches using attention-based models obtains time-dependent explanations by attention weights [Kaj+19; Son+18; Cho+16]. The acquired feature and time attribution to the prediction can be visualized in saliency maps, which are initially implemented for images [Bac+15], and are a current trend in obtaining explanations for importance scores of timestamps and features. Among them, we find gradient-based and perturbation-based feature importance scores [ZF14; Sur+17]. For anomaly detectors, making the models interpretable often goes hand in hand with determining the importance of single input features in the prediction of the anomaly scores. Also, here, feature importance analysis plays an essential role [LZV23]; an example is offered by Dissanayake et al. [Dis+20], introducing a technique based on feature attributions. Additionally, rule-based models [Mül+12], decision trees [PK21], and the usual model-agnostic techniques [RSG16; LL17] are used to explain anomaly detection methods. Visualizing the explanations is essential for both time series and anomaly detector explanations; heat maps, scatter plots and saliency maps offer a concrete boost of users' trust in complex models' scenarios [KB01].

Although progress is not neglectable, the explanations provided by the most recent works are mostly not quantitatively measurable, thus still raising trust issues in users [Zha+19]. Few recent works focus on the quality of the explanation methods; Dombrowski et al. [Dom+19] showed that explanations for image classification are non-robust against possible visually hardly detectable manipulations, and additional works focus on the disagreements among explanation methods [Kri+22]. Furthermore, the interest in choosing proper and fair evaluation metrics in computer science has grown fast in recent years. It is often the case that newly proposed methods optimize for maximizing or minimizing specific metrics, inevitably inducing a relative evaluation of the methods, eventually outperforming competitors on specific datasets or with respect to selective metrics. The first issue is partly solved by benchmarking studies, aiming to compare methods under the same and bright conditions and on various datasets. Some contemporary works in the state-of-the-art literature [Gös+21; GTP21] have started defining essential properties for metrics in specific contexts. However, the conclusion remains always the same: one metric fitting all purposes probably does not exist and trust in machine learning methods is far from being fully achieved.

8 A GROUP-THEORETIC PERSPECTIVE ON RANKING EVALUATION METRICS

Searching for better-performing machine learning techniques requires comparing them with well-established methods. While facing the challenges of finding the right metric to prove the strengths of the proposed models, choosing one over another is non-straightforward. Comparing rankings introduces its formidable challenges; additionally, metrics to compare rankings strictly meant for specific contexts spread to other areas, sometimes without a complete understanding of their inner functioning. This leads to unexpected results and misuses, and, as distinguished metrics focus on different facets of the rankings, comparisons of models' results frequently appear contradicting. We are not the first one to raise the importance of the issue; among others, Tamm et al. [TDV21] harshly criticized the comparisons' reliability of some often-used ranking evaluation metrics.

We observe a notable gap in the literature concerning the limited exploration of standard Recommender Systems (RS) and Information Retrieval (IR) evaluation metrics in contexts beyond RS and IR. The literature focusing specifically on Recommender Systems and Information Retrieval is extensive, where several works explore the relationships among the various metrics [Val+18; GSY12; Sil+19]. We recall [Her+04], where the authors propose a theoretical division of the metrics for comparing collaborative filtering Recommender systems, and [Liu+09], describing most of the metrics typically used for RS and IR techniques. Jarvelin et al. [JK02] focus on various metrics based on cumulative gain, highlighting their main advantages and drawbacks. The work by [Hoy+22] introduces a theoretical foundation for rank-based evaluation metrics, and [ASC18] defines a set of properties for IR metrics and shows that none of the existing ones satisfy all the properties proposed. Other works focus on metrics for RS and their intrinsic properties or ranking metrics for the top-*k* recommendations [BV04; Val+20].

However, the applicability of these metrics and other error metrics is not limited to Recommender Systems. Real-world applications such as the design of strategies based on customers' feedback and allocation of priorities in R&D extended the interest in defining distances among rankings in [Dwo+01; Scu07; Kim+13], where the problem statement focuses on *rank aggregation*. Examples of similarly scoped works are [CKS86; FV86]. However, no available attempts go beyond specific contexts; all literature focuses on concrete applications.

We propose to theorize rankings using the mathematical formalism of symmetric groups. We reveal that pairs of metrics often yield contradictory evaluations. We introduce the "agreement ratio" to quantify the frequency of such disagreements and formally establish essential mathematical properties for ranking evaluation metrics. Lastly, we investigate whether the metrics can be considered "distances" in the mathematical sense. In conclusion, our analysis sheds light on the causes of the inconsistencies in ranking evaluations by comparing the metrics purely based on mathematical concepts. The unique, similar approach was introduced by Diaconis [Dia88], who focuses on six metrics on symmetric groups, among them *Kendall's* τ and *Spearmann's* ρ, and studies them from a statistical and theoretical perspective. [Dia88] defines theoretical properties for the metrics. Among them, we find "interpretability", "tractability", "sensitivity" defined as the ability of one metric to range among the available counter-domain, and "theoretical availability".

To sum up, our main contributions are:

1. transferring the problem of evaluating ranking to metrics on symmetric groups,
2. defining desirable theoretical mathematical properties, desirable both from a theoretical point of view as well as for applications,
3. proving which of them are satisfied by the metrics, thus justifying the appearance of inconsistent evaluations.

8.1 RANKING EVALUATION METRICS

In the state-of-the-art literature, we find various metrics to evaluate ranking in specific contexts. This is the case for most offline Recommender Systems metrics, some evaluation metrics for prediction models, and rank aggregation approaches. Some of these metrics started spreading to other domains following the need to evaluate rankings. However, this is not always a good idea as the domain defines the evaluation's exigencies. In our work, we consider metrics evaluating full rankings that can be easily transferred to adjacent domains and cut out from the analysis all those metrics that require context-specific information, e.g., "diversity" in Recommender Systems. We refer to the group as ranking evaluation metrics; a complete list is summarized in Table 8.1.

Ranking aware metrics	**nDCG**, **DCG**, meanRank, GMR, MRR
Metrics assigning equal importance to each position	*SMAPE*, *MAPE*, *MAE*, *RMSE*, *MSE*, *R^2 score*, NDPM, Spearmann ρ, Kendall's τ
Set based metrics	markedness, PT, recall, LR+, Jaccard index, F1 score, FDR, accuracy, MCC, TNR, fallout, FNR, LR- informedness, NPV, FOR, BA, FM, precision

Table 8.1: LIST OF CONSIDERED METRICS. Bold, italic, wind, and plain text indicate **CGB**, *EB*, CMB, and CB metrics. Other metrics are blue color-coded.

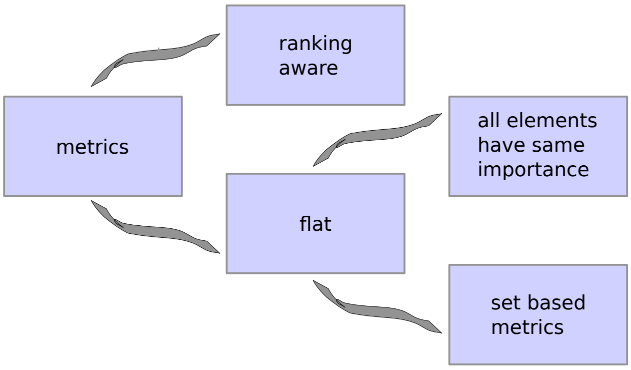


Figure 8.1: THEORETICAL SUBDIVISION OF THE METRICS.

We categorize the ranking evaluation metrics under two different theoretical aspects. One subdivision derives from their "awareness" of the position of single items in the rankings: *ranking aware metrics* satisfy this criterion, while *flat metrics* do not. In this second group, we find the *set-based metrics* and those assigning equal importance to each position. The subdivision is shown in Figure 8.1. From their theoretical definition, we individuate four main groups: *confusion matrix-based CMB metrics* focus on the number of correctly retrieved elements and are essentially set-based metrics; *correlation-based CB metrics* quantify the ordinal association between the two rankings from a statistical perspective; *error-based EB metrics* are often used to analyze the performance of predicting models and are flat metrics assigning equal importance to each position; finally, *cumulative gain-based CGB metrics* focus on the rankings of the single elements.

8.2 RANKING EVALUATION METRICS ON SYMMETRIC GROUPS

To generalize the metrics over an abstract structure, we introduce the symmetric groups S_N. Given a finite set $\mathcal{N} = \{1, \ldots, N\}$, the *symmetric group* S_N is the set of bijective functions from $\mathcal{N}$ to $\mathcal{N}$, i.e., the rankings or *permutations* of elements in $\mathcal{N}$; S_N has size

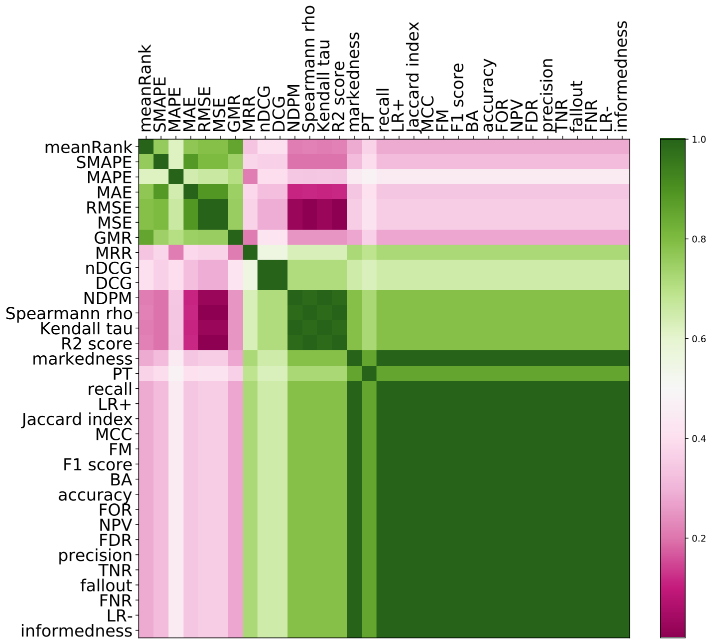


Figure 8.2: Agreement ratio among metrics. Heatmap of the disagreement ratios among pairs of ranking evaluation metrics.

$N!$. Permutations are designed with lowercase Greek letters, i.e., $\sigma \in S_N$; exceptionally, id indicates the group identity or *identity function*. $\sigma(i)$ indicates the position in which item i is sent by σ and, given $\sigma, \nu \in S_N$, $\sigma \circ \nu \in S_N$ is a new ranking defined by $\sigma \circ \nu(i) = \sigma(\nu(i)), \forall i \in \{1, \ldots, N\}$; $\circ$ is the group operation and it is not commutative, i.e., generally $\sigma \circ \nu \neq \nu \circ \sigma$. $\sigma_{|k} = (\sigma(1), \ldots, \sigma(k))$ indicates the ranking of the first k elements; metrics@k consider exclusively the first k ranked elements. Finally, a *(single) swap* is a permutation $\sigma = (j\ k) \in S_N$, swapping only the two elements j, k in $\mathcal{N}$; [HM14] refers to them as "transpositions".

8.2.1 *Clustering by agreement*

Our work is mainly justified by the lack of "consistent" evaluation of rankings when using different metrics. A *ranking evaluation metric* is a function $m : S_N \times S_N \to \mathbb{R}_+$, taking two permutations as input and returning a real number. In some cases, metrics take only one ranking as input; all the given definitions work correspondingly for one-input metrics.

Definition 8.2.1. *Two metrics m_1, m_2 are* non-consistent *if there exists $\sigma, \mu, \nu \in S_N$ such that the following two conditions hold:*

$$\begin{aligned} m_1(id, \sigma) \leq m_1(id, \mu) \ \wedge \ m_2(id, \sigma) \leq m_2(id, \mu) \\ m_1(id, \sigma) \leq m_1(id, \nu) \ \wedge \ m_2(id, \sigma) > m_2(id, \nu) \end{aligned} \tag{8.1}$$

Otherwise, we say that m_1, m_2 are consistent.

The first line of (8.1) guarantees that the reversed metric $\tilde{m}_2 = -m_2$ is still non-consistent with m_1. Proving consistency between two metrics is much trickier than finding three rankings satisfying the inconsistency condition; therefore, rather than classify them, we estimate the degree of inconsistency among pairs of metrics by introducing the agreement ratio. The coefficient provides an estimate of the extent to which two metrics disagree in the evaluation of rankings over symmetric groups.

Definition 8.2.2. *For any $\sigma \in S_N$ fixed, the σ* agreement ratio *among two ranking evaluation metrics, m_1 and m_2 is*

$$AR^{\sigma}_{m_1,m_2} = \frac{1}{|\mathcal{P}(S_N)|(|\mathcal{P}(S_N)|-1)} \sum_{\mu,\nu \in \mathcal{P}(S_N)} f^{m_1,m_2}_{\sigma}(\nu,\mu)$$

where $f^{m_1,m_2}_{\sigma}(\nu,\mu) = \mathbb{1}\{\mu,\nu \text{ are consistent w.r.t. } \sigma\}$ and $\mathbb{1}$ is the indicator function.

As the size of $\mathcal{P}(S_N)$ grows exponentially, we randomly sample a subset $\mathcal{T}$ of $\mathcal{P}(S_N)$ thus obtaining an estimate of the number of inconsistencies existing among two metrics. The agreement ratio equals 1 if m_1 and m_2 are consistent and goes to zero with increasing inconsistencies found; furthermore, the agreement ratio is a symmetric metric.

The color-code heatmap in Figure 8.2 highlights, respectively, in green and pink, the existence of a high agreement and disagreement; a partial agreement is represented in white. It is visible that theoretical similar metrics tend to have an agreement ratio closer to 1. The agreement ratio represents an estimate of the number of inconsistencies among metrics; Figure 8.2 refers to rankings in S_{100}, where $\mathcal{T}$ contains 10000 random rankings. For CMB metrics, we fixed to 30 the number of retrieved and relevant elements.

8.3 PROPERTIES FOR RANKING EVALUATION METRICS

Most pairs of metrics are affected by frequent inconsistent evaluations (cf. Section 8.2.1). We list essential mathematical properties to highlight the peculiarity of each metric and give the chance to properly select one or another based on them for a context-dependent evaluation. The properties in question are: (1) *identity of indiscernibles* (IoI); (2) *symmetry* (or *independence from a ground truth*); (3) *robustness* (Type-I and Type-II); (4) *stability* with respect to *k*; (5) *sensitivity* and *width-swap-dependency*; (6) (induced) *distance*. Some of them have been defined in other domains, e.g., [GTP21; Gös+21; HM14; CKS86; FV86], often under different names.

ranking length	relevant	baseline	σ	τ	CMB metrics	MSE	RMSE	MAE	MAPE	SMAPE	R^2 score	Kendall's τ	Spearmann ρ	DCG	nDCG	MRR	GMR	NDPM	meanRank
10	5	id	(1 2)	id	○	●	●	●	●	●	●	●	●	●	●	●	●	●	●
10	5	id	(1 2)	(3 4)	○	○	○	○	●	●	○	○	○	●	●	○	○	○	○
10	5	id	(1 2)	(2 4)	○	●	●	●	○	○	●	●	●	●	●	○	○	●	○

Table 8.2: IoI PROPERTY. Examples of rankings that metrics cannot distinguish. We compare for each evaluation metric m the values $m(\text{id}, \sigma)$ and $m(\text{id}, \tau)$. If the metric fails in distinguishing the two rankings, we impute a ○; else, a ●.

For each property, we will highlight in which context and why it is important. Table 8.3 and Table 8.4 help the reader to keep trace of the mentioned results; the code is available on GitHub[1].

8.3.1 *Identity of indiscernibles*

Ideally, a metric m quantifies how "close" or "similar" two rankings σ and τ are. However, situations may arise where σ and τ are "so" similar to be practically indistinguishable by some metrics. This effect might be undesired in some fields, such as (fair) rank aggregation, where even small differences, especially in the presence of protected groups, make the difference between fair and unfair rankings.

Definition 8.3.1. *A metric m satisfies the* identity of indiscernible (IoI) property *if, $\forall \sigma \in S_N$ fixed, the following holds*

$$m(\sigma, \tau) = m(\sigma, \nu) \Leftrightarrow \tau = \nu, \qquad \forall \tau, \nu \in S_N. \tag{8.2}$$

Up to renaming the elements, we can rewrite Equation (8.2) as $m(\text{id}, \tau) = m(\text{id}, \nu) \Leftrightarrow \nu = \tau$ where id is the usual identity of S_N.

Almost all metrics do not satisfy the IoI property; for set-based metrics and metrics@k, clear examples not satisfying (8.2) are rankings σ that can be written as a disjoint composition of cycles of permutations of elements before and after k. Table 8.2 illustrates examples for each metric where the IoI is not satisfied. It can be proven that

Proposition 8.3.2. *DCG and nDCG satisfy the Identity of Indiscernibles property.*

[1] Code available at `chiarabales/rankingsEvaluMetrics`

Proof of Proposition 8.3.2. As DCG and nDCG differ only for a constant multiplicative factor, we prove the claim only for DCG. As we deal with pure rankings on symmetric groups, we use the convention $\mathrm{rel}_i = \sigma(i)$ representing a rescaling of the relevance score to distinguished integer numbers; given $\sigma \in S_N$, we use the following definition $\mathrm{DCG}(\sigma) = \sum_{i=1}^{N} \frac{\sigma(i)}{\log_2(i+1)}$.

The goal is proving that for any $\sigma_1, \sigma_2 \in S_N$,

$$\mathrm{DCG}(\sigma_1) = \mathrm{DCG}(\sigma_2) \Leftrightarrow \sigma_1 = \sigma_2.$$

Without loss of generality, we prove that $\mathrm{DCG}(\mathrm{id}) = \mathrm{DCG}(\sigma) \Leftrightarrow \sigma = \mathrm{id}$ for any $\sigma \in S_N$, i.e.,

$$\sum_{i=1}^{N} \frac{i - \sigma(i)}{\log_2(i+1)} = 0.$$

As this is not straightforward, we prove instead a stronger version

$$\sum_{i=1}^{N} \frac{i - \sigma(i)}{\log_2(i+1)} < 0 \Leftrightarrow \sigma \neq \mathrm{id} \in S_N. \tag{8.3}$$

We base our proof on induction over N.

Base case. The base case $N = 2$ is trivial as $S_2 = \{\mathrm{id}, \sigma = (1\ 2)\}$; in particular, $\mathrm{DCG}(\mathrm{id}) = 0$ while

$$\mathrm{DCG}(\sigma) = \frac{1 - \sigma(1)}{\log_2 2} + \frac{2 - \sigma(2)}{\log_2 3} = -\frac{1}{\log_2 2} + \frac{1}{\log_2 3} < 0$$

Inductive case. The claim holds for $N - 1$ and we prove it for N; consider $\sigma \in S_N$. We distinguish two cases.

One element is fixed by σ. Up to renaming the elements, we suppose that N is fixed by σ, i.e., $\sigma(N) = N$. Given $N, k \in \mathbb{N}$, we can construct an immersion

$$i_{N,k} : \sigma \in S_N \mapsto i_{N,k}(\sigma) \in S_{N+k}$$

of S_N in S_{N+k}, such that $i_{N,k}(\sigma)(j) = \sigma(j)$ if $j \leq N$ otherwise $i_{N,k}(\sigma)(j) = j$; $i_{N,k}$ is injective and surjective on

$$A = \{\sigma \in S_{N+k} \mid \sigma(j) = j, \forall j > N + k\}$$

and σ fixes N, σ belongs to S_{N-1} (as the counter-image of $i_{N,1}$). Therefore, the claim holds.

No element is fixed by σ. It holds $\sigma(N) \neq N$ and we can rewrite

σ as the composition of two permutations, i.e., $\sigma = \tau \circ \mu$ such that $\tau = (j\ N)$ for some fixed j and μ such that $\mu(s) = \sigma(s)$ if $s \neq N, k^*$, $\mu(s) = j$ if $s = k^*$ and $\mu(s) = N$ if $s = N$ where we named

$$k^* = \mu^{-1}(j) = \sigma^{-1}(N).$$

We can now rewrite σ in terms of $\tau \circ \mu$:

$$\begin{aligned}
&\sum_{i=1}^{N} \frac{i - \sigma(i)}{\log_2(i+1)} = \\
&\sum_{i=1, i\neq k^*}^{N-1} \frac{i-\sigma(i)}{\log_2(i+1)} + \frac{k^* - \sigma(k^*)}{\log_2(k^*+1)} + \frac{N-\sigma(N)}{\log_2(N+1)} = \\
&\sum_{i=1, i\neq k^*}^{N-1} \frac{i-\mu(i)}{\log_2(i+1)} + \frac{k^* - \tau\circ\mu(k^*)}{\log_2(k^*+1)} + \\
&\qquad \frac{N-\sigma(N)}{\log_2(N+1)} + \frac{k^*-\mu(k^*)}{\log_2(k^*+1)} - \frac{k^*-\mu(k^*)}{\log_2(k^*+1)} = \\
&\sum_{i=1}^{N-1} \frac{i-\mu(i)}{\log_2(i+1)} + \frac{k^*-\tau(j)}{\log_2(k^*+1)} + \\
&\qquad \frac{N-\sigma(N)}{\log_2(N+1)} - \frac{k^*-\mu(k^*)}{\log_2(k^*+1)}
\end{aligned}$$

We note that $\sum_{i=1}^{N-1} \frac{i-\mu(i)}{\log_2(i+1)}$ is negative for the inductive hypothesis and we assume that $\mu \neq \mathrm{id} \in S_{N-1}$.

By substituting $\sigma = \tau \circ \mu$, we conclude the proof if we can upper bound their sum with 0.

$$\begin{aligned}
&\frac{k^*-\tau(j)}{\log_2(k^*+1)} + \frac{N-\sigma(N)}{\log_2(N+1)} - \frac{k^*-\mu(k^*)}{\log_2(k^*+1)} = \\
&\frac{k^*-N-(k^*-\mu(k^*))}{\log_2(k^*+1)} + \frac{N-\sigma(N)}{\log_2(N+1)} = \\
&\frac{\mu(k^*)-N}{\log_2(k^*+1)} + \frac{N-\sigma(N)}{\log_2(N+1)} < \\
&\frac{\mu(k^*)-N}{\log_2(k^*+1)} + \frac{N-\sigma(N)}{\log_2(k^*+1)} = 0
\end{aligned}$$

where we used

$$\begin{aligned}
&\log_2(N+1) > \log_2(k^*+1) \\
&\sigma(N) = \tau\circ\mu(N) = \tau(N) = j \\
&\mu(k^*) = j.
\end{aligned}$$

Thus, the claim is proved for $\mu \neq \mathrm{id}$.

	IoI	Symmetry	Rob. I	Rob. II	WSD	Sensitivity	Stability	Distance
recall	✗	✓	✓	✗	✓	✗	✓	✗
FNR	✗	✓	✓	✗	✓	✗	✓	✗
fallout	✗	✓	✓	✗	✓	✗	✗	✗
TNR	✗	✓	✓	✗	✓	✗	✗	✗
precision	✗	✓	✓	✗	✓	✗	✓	✗
FDR	✗	✓	✓	✗	✓	✗	✓	✗
NPV	✗	✓	✓	✗	✓	✗	✗	✗
FOR	✗	✓	✓	✗	✓	✗	✗	✗
alluracy	✗	✓	✓	✗	✓	✗	✗	✗
BA	✗	✓	✓	✗	✓	✗	✗	✗
F1 score	✗	✓	✓	✗	✓	✗	✓	✗
FM	✗	✓	✓	✗	✓	✗	✓	✗
Mll	✗	✓	✓	✗	✓	✗	✓	✗
Jallard index	✗	✓	✓	✗	✓	✗	✗	✗
markedness	✗	✓	✓	✗	✓	✗	✓	✗
LR-	✗	✓	✓	✗	✓	✗	✗	✗
informedness	✗	✓	✓	✗	✓	✗	✓	✗
PT	✗	✓	✗	✗	✓	✗	✓	✗
LR+	✗	✓	✗	✗	✓	✗	✓	✗
MSE	✗	✓	✗	✓	✓	✗	✓	✗
RMSE	✗	✓	✗	✓	✓	✗	✓	✓
MAE	✗	✓	✗	✓	✓	✓	✓	✗
MAPE	✗	✗	✗	✓	✓	✓	✓	✗
SMAPE	✗	✗	✗	✓	✓	✓	✓	✗
R^2 score	✗	✓	✗	✓	✓	✗	✓	✗
Kendall's τ	✗	✓	✓	✓	✗	✗	✓	✓
Spearmann ρ	✗	✓	✓	✓	✗	✗	✓	✗
NDPM	✗	✓	✓	✗	✗	✓	✓	✗
DCG	✓	✗	✗	✗	✓	✓	✓	✓
nDCG	✓	✗	✓	✗	✓	✓	✓	✓
MRR	✗	✗	✓	✗	✓	✓	✓	✗
IGMR	✗	✗	✗	✗	✓	✓	✗	✗
meanRank	✗	✗	✗	✗	✓	✓	✗	✗

Table 8.3: SUMMARY TABLE. Properties satisfied by the metrics.

In the case $\mu = \text{id}$: Then it holds $\sigma = \tau$ and $\text{DCG}(\sigma)$ reads

$$\sum_{i=1}^{N} \frac{i - \sigma(i)}{\log_2(i+1)} = \sum_{i=1}^{N} \frac{i - \tau(i)}{\log_2(i+1)} =$$

$$\frac{j - \tau(j)}{\log_2(j+1)} + \frac{N - \tau(N)}{\log_2(N+1)} =$$

$$\frac{j - N}{\log_2(j+1)} + \frac{N - j}{\log_2(N+1)} < \frac{j - N + (n - j)}{\log_2(j+1)} = 0$$

This concludes the proof. □

8.3.2 *Symmetry property*

Often, guarantees that the evaluation is symmetric with respect to input items are desirable [Gös+21; GTP21], particularly when the interest is in having a sort of mathematical distance, e.g., for rank aggregation. However, as usual, the context rules the need for a symmetric evaluation. The symmetry property studies whether the metric's evaluation is independent of the order in which the rankings are compared. In RS and IR, the common presence of a "ground truth order" makes the symmetric property impossible.

Definition 8.3.3. *A metric $m : S_N \times S_N \to \mathbb{R}$ is* symmetric *if*

$$m(\sigma, \nu) = m(\nu, \sigma), \qquad \forall \sigma, \nu \in S_N. \tag{8.4}$$

8.3.3 *Robustness*

The Identity of Indiscernibles property studies whether metrics can distinguish rankings, regardless of their similarity. On the other side, the similarities among rankings should be projected on the evaluations: small differences in rankings should result in small differences in the evaluation scores. Under the assumption that a single swap represents a small difference between two rankings, the *Type I robustness* property assesses how sensitive a ranking evaluation metric is to single swaps in the compared rankings.

Definition 8.3.4. *A metric m is* Type I Robust *if a* single swap *in one of the rankings implies small changes in its evaluation, i.e.,*

$$|m(\sigma, \nu) - m(\sigma, \nu \circ (i\ j))| < \epsilon. \tag{8.5}$$

We compute the average of the results of Equation (8.5) evaluated on 1000 different randomly drawn pairs of rankings in S_{100} and round it to two decimal numbers. We state that the metric satisfies the Type I Robustness if the resulting average is 0.

For completeness, we define a second type of robustness that studies the effect of renaming the items in the rankings. [Dia88] mentions Type II Robustness as "right-invariance" and [HM14] as "resistance to item relabeling".

Definition 8.3.5. *A metric is* Type II Robust *if it is an invariant w.r.t. the composition of permutations, i.e., it holds*

$$m(\mu, \sigma) = m(\mu \circ \nu, \sigma \circ \nu), \forall \sigma, \nu \in S_N.$$

Type II Robustness property investigates whether applying the same change in both rankings affects the evaluation. The property is essential in contexts where the numbers appearing in the rankings have to be considered as proper "items" or "items' names"; this is often the case in rank aggregation approaches, Recommender Systems, and Information Retrieval techniques. However, it does not apply when dealing with importance scores. We claim the following:

Proposition 8.3.6. *MSE, RMSE, MAE, MAPE, R^2 score, Kendall's τ score and Spearmann's ρ are the only considered metrics satisfying the Type II Robustness.*

Proof of Proposition 8.3.6. MSE, RMSE, MAE, MAPE, R^2 SCORE. Decomposing the sum in the definition of $MSE(\sigma \circ (j\ k), \nu \circ (j\ k))$ among addends involving k or j and others, it is easy to get to $MSE(\sigma, \nu)$. Similarly, for the other metrics.
KENDALL'S τ. It is enough to note that the number of discordant and concordant pairs does not change when applying a swap to both the rankings σ and ν.
SPEARMANN'S ρ. Similarly to the case of the error based metric, we decompose the sum defining the Spearmann's ρ in elements involving j and k and others; manipulating the definition, we eventually get the thesis.

UNICITY: For all the other metrics, finding pairs of rankings providing counterexamples is trivial. For cumulative gain based metrics, the swaps change the association between the position in the ranking and the relevance score. For confusion matrix based metrics, swaps change both the set of relevant and retrieved elements (but not equally); thus, the evaluation is different after applying swaps in both rankings. □

8.3.4 *Sensitivity*

The sensitivity property is valuable for a metric, particularly in the case of Recommender Systems and Information Retrieval, where high dimensional rankings may not be fully explored. Under the assumption that a full exploration of the rankings is not possible, sensitive metrics assign more weight to the first part of the rankings, considering whether the first k items are "correctly" ranked. Mathematically, we define:

Definition 8.3.7. *Given $i < j < k < l \in \{1, \dots, N\}$ and $(i\ j), (l\ k)$ having the same width. A ranking evaluation metric m is* sensitive *if the swap $(i\ j)$ has a different impact on the evaluation than $(k\ l)$.*

As the evaluation of the property is far from easy, we introduce the *width swap dependency*, formalizing a property that prevents the metrics from being sensitive.

Definition 8.3.8. *Given a swap $(i\ j) \in S_N$ and $|i-j|$ its* width*, m is* width swap dependent *(WSD) if it evaluates swaps with the same width equally; otherwise, it is called* non-width swap dependent*.*

The WSD property cuts out some of the metrics from being sensitive. From their definitions, it can be proven that

Lemma 8.3.9. *Kendall's τ, Spearmann ρ, NDPM are* width swap dependent*.*

Proof of Proposition 8.3.9. SPEARMANN'S ρ. It has an equivalent formulation that depends only on the differences $d_i = \sigma(i) - \nu(i)$; the fact that the elements appearing in the ranking are all distinct implies the WSD property directly.
KENDALL'S τ AND NDPM. To prove the claim for Kendall's τ (NDPM is similar), we fix an arbitrary N and a swap $(i\, j) \in S_N$ of width d. We proceed by induction on d and prove that Kendall's τ is based only on d, independently from i and j. If $d = 1$, then the swap is of the form $(i\ \ i+1)$; in this case, the number of concordant pairs is $\binom{N}{2} - 1$, and the only discordant pair is given by $(i\, i+1)$. Recalling the definition of Kendall's τ, we want to prove that

$$K_\tau = \frac{|\{\text{concordant pairs}\}| - |\{\text{discordant pairs}\}|}{\binom{N}{2};} = \frac{\binom{N}{2} - 4|i-j| + 2}{\binom{N}{2}}.$$

This holds for $d = 1$ as $K_\tau(id, (i\, j)) = \frac{\binom{N}{2}-1+(\binom{N}{2}-(\binom{N}{2}-1))}{\binom{N}{2}} = \frac{\binom{N}{2}-2}{\binom{N}{2}}$. We now suppose that it holds for d and prove it for $d+1$; the number of discordant pairs in a swap of length $d+1$ equals the number of elements that are not anymore concordant with i, i.e., $d+1$, plus the number of elements that are not anymore concordant with j minus 1, i.e., d. Summing up we get

$$K_\tau(id, (i\, j)) = \frac{\binom{N}{2} - (2d+1) + \left(\binom{N}{2} - \left(\binom{N}{2} - (2d+1)\right)\right)}{\binom{N}{2}} = \frac{\binom{N}{2} - 4(d+1) + 2}{\binom{N}{2}}.$$

We conclude that Kendall's τ is width-swap-dependent. □

For the other metrics, we evaluate if pairs of disjoint swaps had different effects in the final evaluation when happening at various positions within the rankings.

8.3.5 *Stability*

We introduce the stability property for those metrics that can be applied on "rankings@k". We recall that a ranking@k is the ranking of the items in the first k positions. To evaluate rankings@k, it is essential that the difference between evaluations "@k" and "@$(k+1)$" is not significant, i.e., that the choice of k does not highly impact the result; this guarantees a trustworthy evaluation.

Definition 8.3.10. *A ranking evaluation metric m is* stable *if, for any two rankings $\sigma, \nu \in S_N$, it holds*

$$|m_{@k}(\sigma,\nu) - m_{@k+1}(\sigma,\nu)| < \epsilon_k \tag{8.6}$$

with ϵ_k small. Moreover, the sequence $\{\epsilon_k\}_k$ satisfies $\lim_{k\to n} \epsilon_k = 0$.

The property is again essential for extremely long rankings and for contexts where rankings are not fully explored. We evaluate the stability by randomly drawing 1000 pairs of rankings in S_{100}, computing the absolute differences of Equation (8.6), and counting the number of times that Equation (8.6) holds with $\epsilon_k = \frac{1}{k}$.

We state that a metric is stable if the criterion is satisfied in at least 97.5% of the cases.

8.3.6 *Distance*

In mathematics, the terms metric and distance are synonyms. However, when it comes to evaluation metrics, most of them are not "distances" on S_N in the mathematical sense. Whether a metric is a mathematical distance or not is often insignificant for the final evaluations; however, being aware of this fundamental mathematical difference can avoid incomprehension and misuses.

Definition 8.3.11. *A* distance *on a set X is a function $f_m : X \times X \to [0,\infty) : (x,y) \mapsto f_m(x,y) \in \mathbb{R}_+$ that, for all $x,y,z \in X$, satisfies:*

1. $f_m(x,y) = 0 \Leftrightarrow x = y$,
2. *the* positive definiteness, *i.e., $f_m(\sigma,\nu) \geq 0, \forall \sigma,\nu \in X$,*
3. *the* symmetry *property and*
4. *the* triangle inequality, *i.e., $f_m(x,y) \leq f_m(x,z) + f_m(z,y)$.*

Some ranking evaluation metrics are distances; in [HM14; Dia88], it is proven that Kendall's τ is a distance. However, a ranking evaluation metric that does not satisfy some of the properties mentioned in Definition 8.3.11 is not a distance.

We investigate if we can induce distances from single input metrics. Given a metric $m : S_N \to \mathbb{R}$, we consider two options as potential induced distances, i.e., $f_m(\sigma, \nu) = m(\sigma) - m(\nu)$ or $\tilde{f}_m(\sigma, \nu) = |m(\sigma) - m(\nu)|$. DCG and nDCG are the only two metrics satisfying the Identity of Indiscernibles property that, for metrics with one unique argument, is equivalent to Property (1) for f_m. We can easily prove that

Proposition 8.3.12. *f_m is* not *a distance while $\tilde{f}_m$ is a distance with the Identity of Indiscernibles property, where m is either DCG or nDCG.*

Proof of Proposition 8.3.12. We must prove the three properties defining a distance for $m =$ DCG (similar to nDCG, that differs only by a multiplicative factor).
Identity Property. Proposition 8.3.2 states that DCG satisfies the IoI property. Furthermore, it follows that $f_m(\sigma, \nu) = 0 \Leftrightarrow \sigma = \nu$; Similarly, $\tilde{f}_m(\sigma, \nu) = 0 \Leftrightarrow \nu = \sigma$.
Symmetry property. It is easy to find pairs of permutations $\sigma, \nu \in S_N$ such that $f_{DCG}(\nu, \sigma) = f_{DCG}(\sigma, \nu)$; in particular, f_{DCG} satisfy the anti-symmetric property, i.e.,

$$\begin{aligned} f_{DCG}(\nu, \sigma) = DCG(\nu) - DCG(\sigma) = \\ -\left[DCG(\sigma) - DCG(\nu)\right] = -f_{DCG}(\sigma, \nu). \end{aligned}$$

On the other hand, $\tilde{f}_{DCG}$ satisfies the symmetry property.
Triangle inequality. The triangle inequality property is satisfied if $\forall \nu, \sigma, \mu \in S_N$ holds $f_{DCG}(\sigma, \mu) \leq f_{DCG}(\sigma, \nu) + f_{DCG}(\nu, \mu)$. Expanding the formula of DCG we get

$$\begin{aligned} f_{DCG}(\mu, \sigma) = {} & DCG(\mu) - DCG(\sigma) = \\ & DCG(\mu) - DCG(\nu) + DCG(\nu) - DCG(\sigma) = \\ & f_{DCG}(\mu, \nu) + f_{DCG}(\nu, \sigma). \end{aligned}$$

The equality holds $\forall \nu, \sigma, \mu \in S_N$; for $\tilde{f}_{DCG}$, the property still holds with the inequality:

$$\begin{aligned} & \tilde{f}_{DCG}(\mu, \sigma) = \ |DCG(\mu) - DCG(\sigma)| = \\ & |DCG(\mu) - DCG(\nu) + DCG(\nu) - DCG(\sigma)| \leq \\ & |DCG(\mu) - DCG(\nu)| + |DCG(\nu) - DCG(\sigma)| = \\ & \tilde{f}_{DCG}(\mu, \nu) + \tilde{f}_{DCG}(\nu, \sigma). \end{aligned}$$

Positive definiteness. $\tilde{f}_{DCG}$ is defined as an absolute value; the claim obviously holds. Instead, f_{DCG} can assume both positive and negative values.

This concludes the proof. □

8.4 ARE THE METRICS INTERPRETABLE?

Given the importance of trust, fairness, and explainability for machine learning methods, one could then ask how "interpretable" the scores assigned by the metrics are. We first need some definitions.

Definition 8.4.1. *A ranking evaluation metric m is said to satisfy the* maximal agreement property *if*

(a) $m(\sigma,\sigma) = m_{\max}, \forall\sigma \in S_N$ *and*

(b) $m(\sigma,\nu) \leq m_{\max}, \forall\nu,\sigma \in S_N$.

We say that m is lower-bounded *if it exists a real number $m_{\min}$ such that $m(\sigma,\nu) \geq m_{\min}, \forall\nu,\sigma \in S_N$. An evaluation metric that admits a lower bound is said to satisfy the* minimal agreement *property.*

For a metric to be "interpretable" we expect that

1. each ranking is maximally similar to itself and, given $N \in \mathbb{N}$, this value is constant, i.e., $m(\sigma,\sigma) = m_{\max}, \forall\sigma \in S_N$ and $\forall N$;
2. m satisfies the maximal agreement property;
3. there exists a lower bound $m_{\min}$ for any possible pair of rankings, i.e., $m(\sigma,\mu) \geq m_{\min}, \forall\sigma,\mu \in S_N$.

The maximal agreement property says that each ranking is maximally similar to itself, and no other ranking can achieve a higher score than $m_{\max}$; furthermore, ideally, $m_{\max}$ is independent of the length of the rankings. Properties 1 and 2 imply that a ranking evaluation metric is a monotone increasing function of the similarity of two rankings: the more similar two rankings are, the higher the score they get when evaluated using an "interpretable" metric. If $m_{\max}$ is independent of N, the evaluation of rankings is independent of N. However, this is hardly satisfied by any metrics, and only after introducing a normalization score do the metrics satisfy the requirement. Furthermore, the lowest scores are assigned by some metrics to maximally similar pairs of rankings, e.g., error-based metrics. The only metrics, among the ones considered in this paper, automatically satisfying this property are Kendall's τ score and Spearmann ρ.

A ranking evaluation metric satisfying the maximal agreement property is also *upper-bounded*. For the sake of interpretability, we could check whether a metric m satisfies $m(\rho^{-1},\rho) = m_{\min}$ where ρ^{-1} indicates the inverse ranking. How do we define the "inverse of a ranking"? Kendall's τ satisfies this property, given that the inverse of one ranking σ is the ranking τ assigning the highest

	description	domain
Identity of Indiscernibles	in highly sensitive evaluations, where detecting tiny differences among rankings is essential	(Fair) rank aggregation Recommender Systems Feature ranking/selection
Symmetry	ensures that the input rankings have an equal role in the evaluations	Rank aggregation Contexts independent from ground truths
Robustness I	ensures that small changes influence proportionately the evaluations	Information Retrieval Rank aggregation
Robustness II	ensures independence from items renaming	Information Retrieval Feature ranking Rank aggregation
Sensitivity	for not fully explored rankings, when the interest is on to the top part of the rankings	Information Retrieval Recommender Systems
Stability	ensures trustworthiness in evaluations @k	Information Retrieval Recommender Systems
Distance	ensures that the metric in questions respect the definition of distance on S_N	(Fair) rank aggregation

Table 8.4: SUMMARY.

position to the last element of the ranking σ; however, this does not correspond with the inverse of the ranking in the symmetric group. Assessing whether metrics for permutations are humanly interpretable is not new and has already been discussed in [Dia88]. However, then, as well as now, the concept of interpretability lacks a unified definition. Thus, we leave this section open and do not argue further on the interpretability of the considered metrics.

8.5 DISCUSSION

We explored metrics for comparing and evaluating rankings and analyzed their theoretical properties. All the mentioned metrics are widely used in the literature to evaluate Recommender Systems, Information Retrieval, feature ranking, rank aggregation methods, and items' score assignments. Each property is highly desirable in some contexts and less in others. The IoI property is desirable in highly sensitive evaluations, where detecting tiny differences among rankings is essential; fair ranking aggregation is an example, where swapping items can make the difference between fair and unfair rankings. Conversely, robustness ensures that small changes influence the evaluations proportionately in a one-to-one fashion. A metric that satisfies both the IoI and the robustness properties ensures contemporaneously that small changes are not overlooked but do not significantly impact the evaluations. The symmetry property ensures that the input rankings have an equal role in the evaluations. This is essential in most domains unless ground truth ranking is available. Note that non-symmetric metrics are also not distances. Rank aggregation is again an example of use for the symmetry property, where the consensus ranking is directly compared with the original rankings provided. Sensitivity is crucial when rankings are not fully explored. This is often the case for Recommender Systems and Information Retrieval techniques' evaluations. With the same applicability, the stability property ensures trustworthiness in evaluations @k, which is again highly relevant for Recommender Systems and Information Retrieval techniques. To assure stable evaluations, we recommend considering evaluating the impact @k and @$(k+i)$ with i arbitrarily chosen, in particular when $k << N$. Finally, the distance property is defined to complete the proposed analysis and highlights the chance that mathematical terms are misused in machine learning contexts. Table 8.4 summarizes the properties' descriptions and application domains.

9 ON THE CONSISTENCY AND ROBUSTNESS OF SALIENCY EXPLANATIONS FOR TIME SERIES CLASSIFICATION

Intrinsic and post-hoc explanations have become essential in many applications and a core topic for machine learning research. Time series represents one niche data type where most implemented methods still lack explanations. One of the reasons for the poor literature on the explainability of time series data is the additional challenges brought by the time-dependent structure. Explanations often reduce to applications of model-agnostic post-hoc explanations for general data samples to time-dependent data; the time structure is often disregarded, and the timestamps are treated as independent samples on which the model is learned [Ben+21b; Sch+19]. Overall, explanation methods on time series data consider the time dimension either jointly with the other features [Ben+21a] or separately [Ism+20] by sequentially considering time and features. In both cases, the explanation is limited to one single "frame", i.e., a sliding window. Hence, both categories are insufficient for interpreting the overall temporal information over a long time span. Another thread of approaches using attention-based models obtains time-dependent explanations by attention weights [Kaj+19; Son+18; Cho+16]. The acquired feature and time attribution to the prediction can be visualized in saliency maps, which are initially implemented for images [Bac+15] and are a current trend in obtaining explanations for importance scores of timestamps and features; saliency explanations rely on the creation of heatmaps, representing the importance of pixel-like features that can be easily interpreted, visually identifying the most relevant areas for the given task. The quality of explanations derived from treating windows of time series as images and applying vision explanation methods is often questionable. Among typical image explanation approaches, we find gradient-based [Bae+10; STY17; Smi+17; SGK17] and perturbation-based feature importance scores [ZF14; Sur+17]. Gradient (GRAD) was introduced by Baehrens et al. [Bae+10] in 2010 as a post-hoc model agnostic interpretation tool that can explain nonlinear classifiers at a local level. The explanations measure how each data point must be moved to change the predicted label. The local scores derive from the direct computation of the local gradients (or their estimations) for the given model. Similarly, Integrated Gradients [STY17] is also a gradient-based feature im-

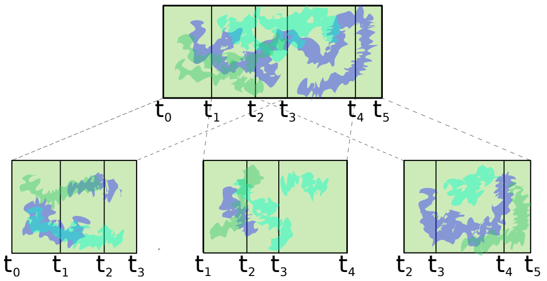


Figure 9.1: Representation of inconsistencies among saliency maps of adjacent sliding windows (x-axis: time, y-axis: features, color: importance attribution).

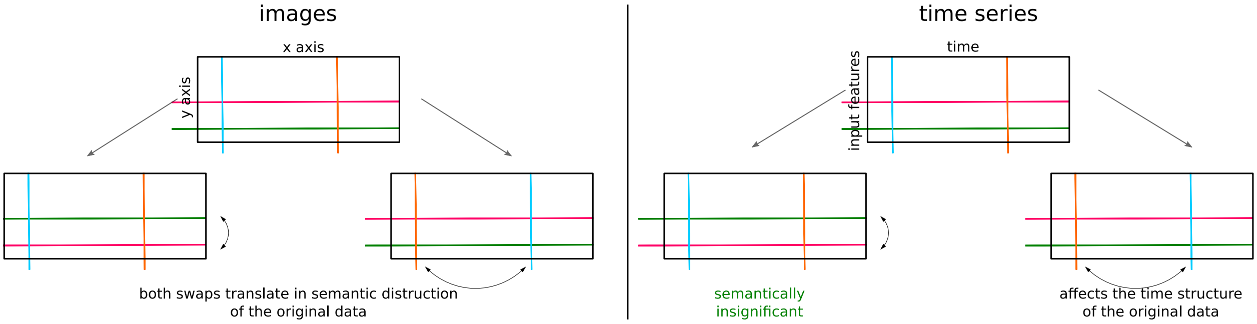


Figure 9.2: Representative drawings of time series data as 2D data frames.

portance attribution method and builds up on [Bae+10] and two axioms, i.e., the "sensitivity" and "implementation invariance". Ismail et al. [Ism+20] pointed out how these methods often lack understanding of the time-feature structures, allowing them to achieve only good performances at the time or at the features level. The authors propose an alternative two-step approach to saliency explanations for time series, where the time structure is considered first, and the importance of the features is considered only in the second step. The explanation quality of such a method is still understudied in the time series domain.

We study saliency maps to explain time series data predictions by introducing two main concerns, i.e., the saliency maps' *consistency* and *robustness*. We report experiments showing that the typical attribution approaches used for time series are neither robust nor consistent. We claim that explanations over the intersection of sliding windows should exhibit "consistent" behaviors, and we identify such a flaw in current time series saliency explanations. We admit that in adjacent sliding windows, different temporal contexts may lead to different absolute feature attributions; hence, we pursue "relative consistent attributions" in local sub-windows. Figure 9.1 illustrates the meaning of "consistency" among overlapping time windows. In addition to the saliency explanation consistency, the "robustness" of saliency maps against feature perturbation is another essential factor in ensuring explanation quality. In images, the semantic meaning of columns and rows is equivalent, while in time series, the time structure makes time series data semantically different and introduces dependence among the observations in the various timestamps. In images, swaps of rows and columns of pixels affect the semantic structure of the original data. In contrast, in time series, only the swaps affecting the temporal orders of the observations are semantically meaningful. In contrast, the order in which input values are collected has no effects. The phenomenon is illustrated in Figure 9.2 where the x-axis corresponds to the time, and the y-axis to the input variables. When saliency maps are applied to time series, the salient features should be insensitive to the order of the input features. When feature columns in the time series frame are swapped, essential areas in the saliency map should stay salient in the corresponding swapped areas. We call this the *robustness* of saliency explanations.

We examine saliency explanations from popularly used approaches on multiple deep classification models [HS97; Lea+17; Vas+17] and show on five real-world datasets that the studied saliency explanation suffers from inconsistency and non-robustness issues. These preliminary results underline the encountered problems as a motivating example of further research on developing robust and consistent saliency explanations for time series.

To summarize, this chapter contributes to several major points:

1. we provide a theoretical, well-founded definition of *consistency*
2. and a definition of *robustness* among saliency explanations for time series classification methods;
3. we empirically show how the blind use of saliency explanations for time series classification methods does not satisfy any of the defined desirable properties.

9.1 OPEN ISSUES ON SALIENCY EXPLANATIONS

This section formally defines the consistency and robustness of the saliency explanation for time series classification. We indicate with $X = (X_1, \ldots, X_N)$ a multivariate N-dimensional discrete time series where X_i is the i-th univariate dimension; t_0 is the first timestamp on which the time series is defined. For each timestamp $t_k > t_0$, $X(t_k)$ is a N-dimensional vector of real values, i.e., $X(t_k) \in \mathbb{R}^N$. We study the consistency and robustness of saliency explanations for classification models trained on time series data. We draw upon the concept of consistency proposed by Pillai et al. [Pil+22], and define *consistency* of saliency explanations over adjacent sliding windows of time series. Additionally, regarding *robustness*, we consider the influence of swaps of features, i.e., of input variables observations, in explanations using saliency maps.

9.1.1 *Consistency*

As in Chapter 6, we define time windows $\{w_s^d\}_{s\in\mathbb{N}}$ dependent on the window length $d \in \mathbb{N}$ and the starting timestamp t_s, i.e.,

$$w_s^d = \{t_s, \ldots, t_{s+d-1}\}. \tag{9.1}$$

For each time window and given a fixed saliency map method assignation of importance score S, we get

$$S(w_s^d) = S_s^d$$

a matrix in $\mathbb{R}^{N\times d}$ such that $(S_s^d)_{n,t}$ is the importance scores assigned to the input variable X_n at time t. Saliency maps are transposed from image (pre)processing applications to explain time series classification predictions.

We examine the consistency of saliency maps defined over overlapping windows. Given two windows w_s^d and $w_{\bar{s}}^{\bar{d}}$ such that $|w_s^d \cap w_{\bar{s}}^{\bar{d}}| \neq \emptyset$ and the respective saliency maps S_s^d and $s_{\bar{s}}^{\bar{d}}$, the saliency explanations are *inconsistent* at timestamp t, if $t, \bar{t} \in w_s^d \cap w_{\bar{S}}^{\bar{d}}$ such that

$$(S_s^d)_{n,t} > (S_{\bar{s}}^{\bar{d}})_{n,t} \quad \text{and} \quad (S_s^d)_{n,\bar{t}} < (S_{\bar{s}}^{\bar{d}})_{n,\bar{t}}, \tag{9.2}$$

i.e., the importance scores assigned to features and timestamps are "relatively" inconsistent among overlapping time windows. The phenomenon is illustrated in Figure 9.1: the saliency map in the top shows the overall attribution from t_0 to t_5 where colors' distribution represents the importance of the timestamps and input variables. The bottom row shows the saliency maps relative to three adjacent sliding windows (from left to right, in the interval $[t_0, t_3]$, $[t_1, t_4]$ and $[t_2, t_5]$). It is easy to spot the different cuts of the time windows are characterized by different color distributions than the saliency map in the first row; in particular, the colors' distributions in the overlapping window $[t_2, t_3]$ in the four cases are different.

9.1.2 *Robustness*

Although similarly structured, we mentioned that images and time series intrinsically include a different semantic meaning due to the time dependency. However, the time series explanation should be insensitive to the feature ordering. A saliency explanation is considered *robust* if the saliency scores change accordingly when the features are swapped. We define the feature swapping operation on data window w^d_s and observe the effect in the corresponding saliency explanation S^d_s. Concretely, we swap random pair of features X_i and X_j $(i \neq j)$ in w^d_s for all timestamps from t_s to t_{s+d-1}. Their feature attribution in S^d_s are $(S^d_s)_i$ and $(S^d_s)_i$. After features swapping, the data window is denoted by w^{*d}_{s}, and the newly learned saliency explanation is S^{*d}_{s}. $(S^d_s)_i$ corresponds to $(S^{*d}_{s})_j$ while $(S^d_s)_j$ corresponds to $(S^{*d}_{s})_i$. The saliency explanations are robust if $\forall t_1, t_2 \in w^d_s \cap w^d_s$, it holds

$$(S^d_s)_{i,t_1} > (S^d_s)_{i,t_2} \Longrightarrow (S^{*d}_{s})_{j,t_1} > (S^{*d}_{s})_{j,t_2}, \tag{9.3}$$

i.e., important feature-time pixels maintain relative importance after swapping the feature of the data window. The phenomenon is illustrated in Figure 9.2. The position of a pixel in an image is defined by row and column numbers, while in time series data, it is defined by input variables and timestamps. Swaps of rows and columns of pixels in images may affect the semantic meaning of the entire data frame. For time series, this may happen only when swapping observations at different timestamps.

9.2 EXPERIMENTS

We perform experiments on time series classification on real-world datasets. We generate various types of explanations in the form of saliency maps for the predictions made by the model to examine their consistency and robustness. We incorporate artificial padding into the input sequences to precisely control the feature's importance and simulate the sliding window mechanism commonly used in time series analysis tasks. This section presents our findings on identifying inconsistency and non-robust saliency explanations across multiple datasets.

9.2.1 *Datasets*

We consider five real-world univariate time series datasets: Power Demand (PD), Wine (WIN), Italy Power Demand (IPD), Two Lead ECG (ECG) and Mote Strain (MS). PD derives from Keogh et al. [KLF05], while the others are available online in the UCR Archive [Che+15]. We preprocess all datasets by dividing them into non-overlapping windows a priori; the class labels of each window are available. However, the ground truth data does not include the attribution of the prediction. In Sections 9.2.3 and 9.2.4, we introduce artificial padding with random noise to each input window and assign equal importance to the area of the original input.

9.2.2 *Experimental setup*

For our experiments, we select three representatives from the common saliency explanation approaches [Ism+20] for time series data. We employ Feature Permutation (FP) and Feature Ablation (FA) [Sur+17], which are perturbation-based methods, and Integrated Gardients (IG) [STY17], which is a gradient-based method. We use the implementation provided by Ismail et al. [Ism+20].

We investigate the behavior of saliency explanations on three types of network structures: Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and attention-based networks. To this end, we picked three implementations commonly used for time series data classification: LSTM [HS97], TCN [Lea+17], and Transformer [Vas+17]. We configure these models with a Softmax output layer for classification and train the models on all the padded variants of the input windows, including top, middle and bottom padding. During the test phase, we generate saliency maps and analyze the effect of each group of padding variants separately.

9.2.3 *Consistency evaluation*

We apply artificial padding to each univariate time window to evaluate the explanation consistency over sliding windows. Specifically, we expand each univariate data window $w_s^d \in \mathbb{R}^{n \times d}$ to a matrix

$$m \in \mathbb{R}^{\alpha \times \lfloor \beta \cdot d \rfloor} (\alpha > n, 1 < \beta < 3).$$

The data window w_s^d is placed on d consecutive dimensions of m, and the other dimensions are filled with randomly sampled noise from a normal distribution. The effect of a sliding window can be simulated by placing w_s^d at different rows in m. Specifically, we allocate w_s^d at the top, middle, and bottom third of m to generate three overlapping sliding windows, i.e., three variants of each input window. We call the area in the saliency map corresponding to the input window w_s^d , the "area of interest". We show the experimental results by setting $\alpha = 4$ and $\beta = \frac{5}{3}$. An example of the padded data window is shown in Figure 9.3. In the proposed construction, we get that each padding variant group (top/middle/bottom) contains

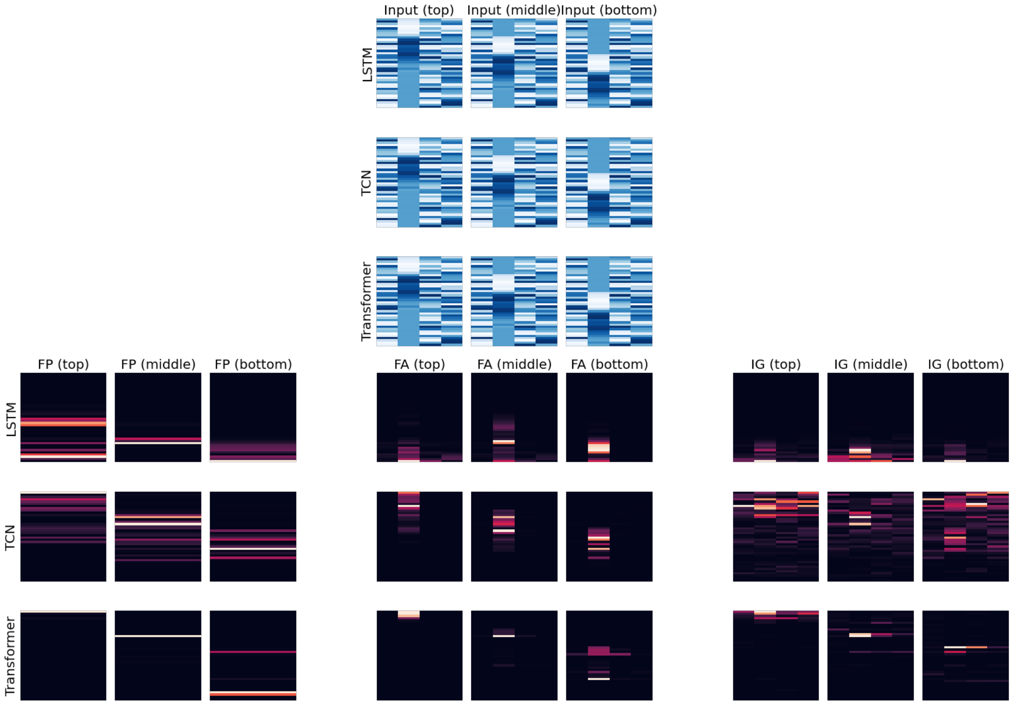
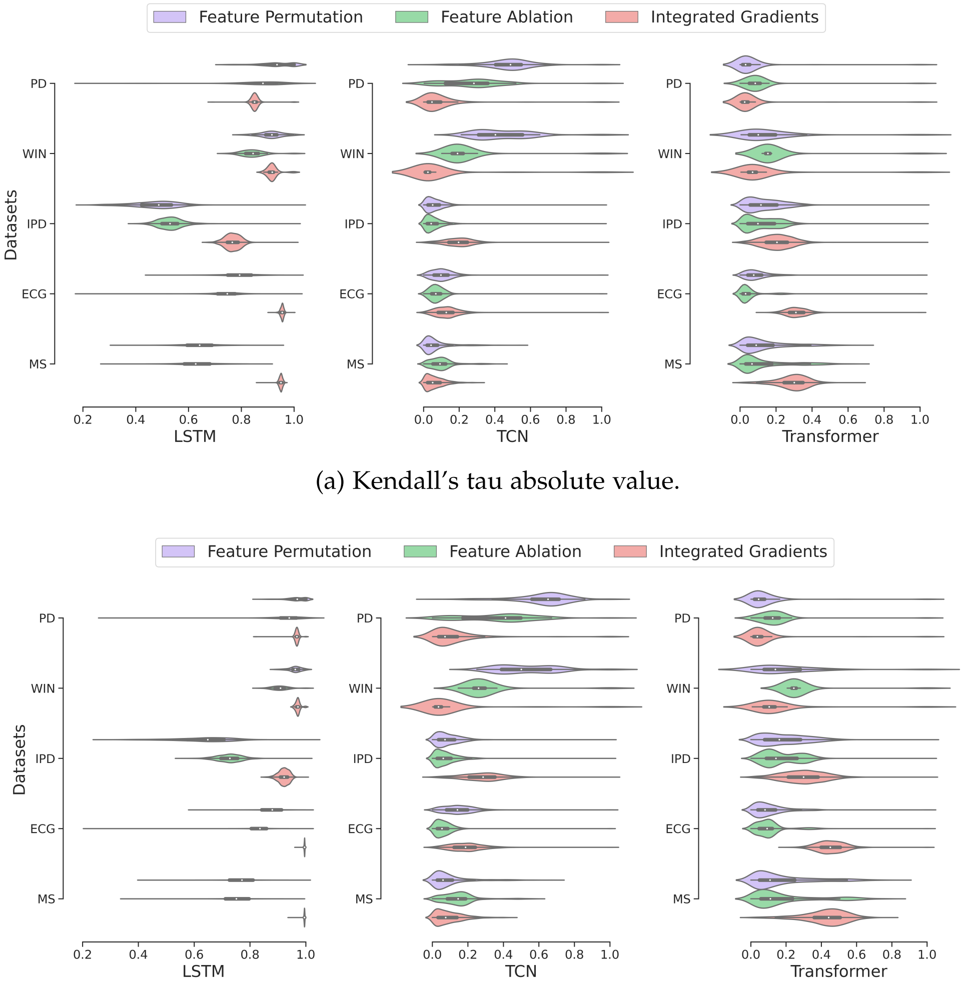


Figure 9.3: IPD DATASET, SALIENCY EXPLANATIONS. In the first line, the blue heatmaps denotes the variants of one input frame, where rows are timestamps and columns are features; the real data window is located in the frame's top, middle, and bottom third. In the second line, the remaining elements are random noise. The saliency maps are the attributions respectively by FP, FA, and IG.

the same input window, only located differently. To examine the consistency of the saliency explanations, we compare the feature ranking of the obtained attributions in corresponding locations in each padding variant. As a showcase, we visualize the result of one window from the IPD dataset in Figure 9.3. The left three columns of Figure 9.3 represent the saliency explanations in the various padded input windows, the y-axis being the time and the x-axis being the input features. Only the second feature contains essential information to be learned by the classifiers (cf. first column in Figure 9.3). The saliency maps on the right side correspond to the three explanation models FP, FA, and IG. We expect the second feature column's top, middle, and bottom third to be marked as salient. However, as Ismail et al. [Ism+20] have already shown, classical saliency methods might fail on time series data due to the temporal feature, and our experiments confirm their results; Figure 9.3 also nicely underlies that the latest timestamps play more important roles in the prediction. The various explainers can



(a) Kendall’s tau absolute value.

(b) Pearson correlation absolute value.

Figure 9.4: Violin plots. Each row contains three plots, i.e., from left to right, LSTM, TCN, and Transformer architecture; the colors code for the various explanation methods.

detect the important timestamps and suffer from distinguishing important features for TCN and Transformers.

Despite the sub-optimal saliency explanations, we analyze the consistency between the padding variants. We evaluated the disagreement empirically on the saliency explanations using Kendall’s τ and Pearson correlation; Kendall’s τ measures the smallest number of swaps of adjacent elements that transform one ranking into the other while the Pearson correlation coefficient measures the covariance of the two random variables divided by the product of their standard deviations (cf. Chapter 8). All quantities can be estimated using finite samples.

We calculate the importance scores for each timestamp and input feature, obtaining the importance ordering of the “area of

		Feature Permutation			Feature Ablation			Integrated Gradients		
		top	middle	bottom	top	middle	bottom	top	middle	bottom
PD	LSTM	0.041	0.000	0.000	0.072	0.000	0.000	0.165	0.268	0.268
	TCN	0.227	0.216	0.216	0.454	0.320	0.320	0.103	0.206	0.206
	Transf.	0.258	0.258	0.258	0.825	0.928	0.928	0.351	0.330	0.330
WIN	LSTM	0.064	0.051	0.051	0.103	0.060	0.060	0.244	0.248	0.248
	TCN	0.256	0.269	0.269	0.500	0.487	0.487	0.286	0.295	0.295
	Transf.	0.333	0.812	0.812	0.949	0.962	0.962	0.389	0.385	0.385
IPD	LSTM	0.250	0.250	0.250	0.625	0.708	0.708	0.375	0.458	0.458
	TCN	0.250	0.250	0.250	0.875	0.958	0.958	0.292	0.375	0.375
	Transf.	0.250	0.250	0.250	0.750	0.708	0.708	0.167	0.292	0.292
ECG	LSTM	0.171	0.171	0.171	0.341	0.244	0.244	0.256	0.268	0.268
	TCN	0.256	0.256	0.256	0.634	0.634	0.634	0.329	0.305	0.305
	Transf.	0.256	0.256	0.256	0.780	0.829	0.829	0.244	0.293	0.293
MS	LSTM	0.250	0.262	0.262	0.536	0.560	0.560	0.357	0.321	0.321
	TCN	0.250	0.262	0.262	0.750	0.798	0.798	0.298	0.381	0.381
	Transf.	0.250	0.250	0.250	0.845	0.821	0.821	0.274	0.369	0.369

Table 9.1: Consistency. Recall@k.

interest". For each pair of ranking from the three padding variants, we analyzed the pairwise comparisons among rankings of feature-time pixels in the saliency explanations FP, FA, and IG. The average Kendall's τ and Pearson correlation (ρ) are summarized in Table 9.2 and the absolute values are visualized in Figure 9.4.

Table 9.2 contains, for each data set, neural network architecture, and saliency map, the average Kendall's τ and Pearson correlation coefficients with the respective variance. From the table, it is easy to spot how the importance scores rankings provided vary in ranges below 1. Kendall's τ and Pearson correlation coefficients range between 1 and -1, where 1 indicates complete agreement among the rankings, while values close to zero suggest non-constant and independent orderings. From Figure 9.4a and Figure 9.4b, we observe that the coefficients are, in most cases, crowded at low values, and episodes of perfect agreement among the obtained rankings in the different windows are rare (although non-anomalies). Although the two metrics measure something essentially different, the behavior observed in Figure 9.4a and Figure 9.4b is similar.

A special case is the LSTM algorithm that provides consistent saliency maps among the various windows. However, observing the explanation over the LSTM model, we see that both Kendall's τ and Pearson correlation coefficients tend to accumulate to high scores (≈ 1) as the LSTM method tends to accumulate the learning

		FP		FA		IG		
		τ	ρ	τ	ρ	τ	ρ	
PD	LSTM	0.936 ± 0.063	0.967 ± 0.035	0.868 ± 0.114	0.921 ± 0.092	0.852 ± 0.024	0.968 ± 0.012	consistency
	TCN	0.475 ± 0.143	0.625 ± 0.155	0.247 ± 0.200	0.334 ± 0.256	0.060 ± 0.148	0.080 ± 0.168	
	Transf.	0.040 ± 0.136	0.049 ± 0.145	−0.049 ± 0.149	−0.086 ± 0.161	0.026 ± 0.138	0.030 ± 0.145	
WIN	LSTM	0.914 ± 0.042	0.961 ± 0.023	0.843 ± 0.044	0.908 ± 0.030	0.918 ± 0.022	0.971 ± 0.009	
	TCN	0.445 ± 0.164	0.530 ± 0.169	0.220 ± 0.159	0.287 ± 0.149	0.052 ± 0.200	0.057 ± 0.206	
	Transf.	0.150 ± 0.192	0.196 ± 0.207	−0.11 ± 0.218	−0.198 ± 0.236	0.107 ± 0.180	0.140 ± 0.178	
IPD	LSTM	0.475 ± 0.086	0.633 ± 0.101	0.531 ± 0.047	0.725 ± 0.045	0.767 ± 0.030	0.921 ± 0.019	
	TCN	0.001 ± 0.083	−0.002 ± 0.112	−0.004 ± 0.075	−0.031 ± 0.095	0.192 ± 0.083	0.277 ± 0.114	
	Transf.	0.124 ± 0.115	0.167 ± 0.157	0.054 ± 0.141	0.042 ± 0.199	0.204 ± 0.088	0.295 ± 0.121	
ECG	LSTM	0.789 ± 0.070	0.873 ± 0.056	0.738 ± 0.062	0.827 ± 0.055	0.954 ± 0.007	0.995 ± 0.001	
	TCN	0.102 ± 0.074	0.143 ± 0.096	0.072 ± 0.062	0.054 ± 0.070	0.129 ± 0.078	0.189 ± 0.101	
	Transf.	0.089 ± 0.082	0.098 ± 0.110	0.020 ± 0.092	−0.038 ± 0.137	0.315 ± 0.066	0.453 ± 0.081	
MS	LSTM	0.642 ± 0.072	0.768 ± 0.066	0.632 ± 0.078	0.753 ± 0.070	0.950 ± 0.007	0.995 ± 0.002	
	TCN	0.038 ± 0.096	0.053 ± 0.134	−0.031 ± 0.116	−0.058 ± 0.167	0.055 ± 0.061	0.081 ± 0.089	
	Transf.	0.124 ± 0.136	0.156 ± 0.189	0.093 ± 0.165	0.076 ± 0.244	0.291 ± 0.089	0.423 ± 0.125	
PD	LSTM	0.972 ± 0.064	0.985 ± 0.036	0.973 ± 0.112	0.982 ± 0.102	0.807 ± 0.064	0.942 ± 0.066	robustness
	TCN	0.680 ± 0.168	0.803 ± 0.157	0.597 ± 0.259	0.706 ± 0.285	0.221 ± 0.148	0.307 ± 0.175	
	Transf.	0.042 ± 0.141	0.052 ± 0.153	0.129 ± 0.148	0.164 ± 0.161	0.123 ± 0.133	0.171 ± 0.143	
WIN	LSTM	0.932 ± 0.047	0.971 ± 0.021	0.924 ± 0.056	0.967 ± 0.029	0.929 ± 0.075	0.972 ± 0.057	
	TCN	0.598 ± 0.107	0.696 ± 0.083	0.553 ± 0.094	0.676 ± 0.073	0.083 ± 0.186	0.105 ± 0.189	
	Transf.	0.267 ± 0.241	0.347 ± 0.283	0.466 ± 0.153	0.587 ± 0.167	0.345 ± 0.149	0.473 ± 0.143	
IPD	LSTM	0.446 ± 0.124	0.603 ± 0.147	0.579 ± 0.075	0.732 ± 0.076	0.699 ± 0.062	0.864 ± 0.053	
	TCN	0.114 ± 0.141	0.157 ± 0.188	0.277 ± 0.159	0.358 ± 0.194	0.254 ± 0.137	0.361 ± 0.183	
	Transf.	0.369 ± 0.158	0.480 ± 0.178	0.364 ± 0.215	0.458 ± 0.244	0.322 ± 0.148	0.449 ± 0.192	
ECG	LSTM	0.821 ± 0.100	0.898 ± 0.073	0.841 ± 0.133	0.905 ± 0.103	0.967 ± 0.029	0.996 ± 0.028	
	TCN	0.382 ± 0.119	0.515 ± 0.138	0.517 ± 0.087	0.657 ± 0.101	0.207 ± 0.125	0.300 ± 0.172	
	Transf.	0.276 ± 0.126	0.365 ± 0.153	0.535 ± 0.141	0.673 ± 0.141	0.415 ± 0.081	0.582 ± 0.101	
	LSTM	0.747 ± 0.112	0.857 ± 0.089	0.694 ± 0.107	0.811 ± 0.090	0.968 ± 0.027	0.996 ± 0.024	
	TCN	0.147 ± 0.140	0.201 ± 0.187	0.184 ± 0.144	0.248 ± 0.189	0.068 ± 0.087	0.100 ± 0.126	
	Transf.	0.473 ± 0.220	0.575 ± 0.238	0.513 ± 0.215	0.627 ± 0.239	0.453 ± 0.105	0.621 ± 0.125	

Table 9.2: Consistency and robustness.

		Feature Permutation			Feature Ablation			Integrated Gradients		
		top	middle	bottom	top	middle	bottom	top	middle	bottom
PD	LSTM	0.031	0.000	0.000	0.072	0.000	0.000	0.134	0.237	0.237
	TCN	0.227	0.258	0.258	0.299	0.443	0.443	0.155	0.216	0.216
	Transf.	0.268	0.309	0.309	0.742	0.876	0.876	0.268	0.299	0.299
WIN	LSTM	0.056	0.030	0.030	0.103	0.051	0.051	0.231	0.248	0.248
	TCN	0.274	0.265	0.265	0.389	0.355	0.355	0.295	0.282	0.282
	Transf.	0.303	0.252	0.252	0.466	0.568	0.568	0.179	0.372	0.372
IPD	LSTM	0.250	0.250	0.250	0.458	0.417	0.417	0.375	0.458	0.458
	TCN	0.250	0.250	0.250	0.458	0.500	0.500	0.333	0.375	0.375
	Transf.	0.250	0.250	0.250	0.458	0.542	0.542	0.292	0.333	0.333
ECG	LSTM	0.244	0.134	0.134	0.341	0.232	0.232	0.256	0.280	0.280
	TCN	0.256	0.256	0.256	0.476	0.476	0.476	0.232	0.256	0.256
	Transf.	0.256	0.256	0.256	0.634	0.622	0.622	0.390	0.305	0.305
MS	LSTM	0.250	0.250	0.250	0.381	0.417	0.417	0.333	0.262	0.262
	TCN	0.250	0.250	0.250	0.500	0.500	0.500	0.405	0.381	0.381
	Transf.	0.250	0.250	0.250	0.357	0.405	0.405	0.190	0.226	0.226

Table 9.3: Robustness Recall@k.

in the last timestamps, thus implying that the explanation methods assign high importance only to the last timestamps. We further observe FA correctly finds the relevant timestamps but cannot distinguish between noisy and relevant features.

In addition to the relative ranking, we also check the quality of the saliency explanation using recall@k. Table 9.1 contains the recall@k obtained among the importance rankings of timestamps in the "areas of interest". Recall@k measures the ratio among correctly relevant and retrieved elements and the number of relevant elements and ranges in $[0, 1]$. High recall (≈ 1) indicates that the highly ranked feature-time pixels are concentrated in the area of interest, while low recall@k indicates the inability to find relevant elements correctly.

9.2.4 *Robustness evaluation*

To evaluate the robustness of the saliency explanation, we apply the feature swapping depicted in Figure 9.2. Specifically, we continue using the padded input matrix $m \in \mathbb{R}^{\alpha \times \lfloor \beta \cdot d \rfloor}$ from Section 9.2.3 and swap the feature dimensions containing the original input data window, i.e., the "area of interest", with noise dimensions. We train different classification models with the swapped and not swapped data. For simplicity, we always locate the original window in the middle of the selected feature dimension in this experiment. We compare the ranking of feature-time pixel explanations in the "areas of interest" of swapped and not swapped pairs. The Kendall's τ and Pearson correlation ρ are summarized in Table 9.2 and Table 9.3. The absolute values of Kendall's τ and Pearson correlation for TCN and Transformers indicate a significant difference in saliency maps after the swapping. In other words, when the important feature is switched with a noisy feature, the feature attribution in the saliency map is not switched correspondingly. An exception is the LSTM classifier, which robustly explains all datasets except IPD. However, the explanation quality is limited.

10 ON THE EFFICIENT EXPLANATION OF OUTLIER DETECTION ENSEMBLES THROUGH SHAPLEY VALUES

Feature bagging models have revealed their practical usability in various contexts. For outlier detection, where they build ensembles to assign outlier scores to data samples reliably, they perform exceptionally well [KM23]. However, the interpretability of so-obtained outlier detection methods is far from achieved. Shapley values [LL17; SK10; RSG16] are used as one of the many black-box models' interpretability approaches. By providing valuable insights into the importance of features in identifying anomalies, Shapley values attribute the contributions of the individual features to the anomaly scores and reveal helpful for outlier detection as well [TI14; TC20]. However, Shapley values are characterized by high computational runtimes that make them useful only in low-dimensional setups. The exponential blow-up in computational cost soon renders their use for high-dimensional contexts infeasible, while approximation techniques have been implemented to make Shapley values more accessible [Cam+18; CGT09; LL17; BC21; SK10].

Ensemble approaches comprehend various techniques such as bagging [Bre96], boosting [Sch+99], and stacking [San17]. Bagging involves training multiple base models on possibly bootstrapped data samples and aggregating their predictions; examples are k-nearest neighbors, Support Vector Machines, and neural networks. Adapted to outlier detection, the ensemble's collective decision provides more robust results [AP96]. Homogeneity among the base models' types characterizes "homogeneous" outlier ensembles, where the models usually differ only by a different initialization. DEAN [BKM22] and Isolation Forest (IForest) [LTZ08] are prime examples of outlier detection methods employing such homogeneous ensembles; DEAN is based on multiple neural networks, while IForest relies on a collection of isolation trees. On the other hand, all ensemble methods are hardly interpretable and pretty complex.

We propose bagged Shapley values, a method to achieve interpretability for feature bagging ensembles for outlier detection. Interpreting anomaly detection methods is essential for understanding *why* single data points are considered anomalous, particularly in safety-critical applications; feature importance analysis plays an essential role here [LZV23; Dis+20]. Our method not only assigns local importance scores to each feature of the initial space, helping to increase the interpretability but also solves the computational issue; specifically, the BAGGED SHAPLEY VALUES can be exactly computed in polynomial time. We visualize the interpretation as heatmaps [KB01], which have historically been useful in enhancing trust in complex models' predictions (cf. Chapters 6,9).

To conclude, our main contribution can be summarized as

1. we solve the computational challenge of exactly computing Shapley values by introducing the BAGGED SHAPLEY VALUES and

2. we use the BAGGED SHAPLEY VALUES to interpret through saliency maps the result of feature-bagging ensembles for anomaly detection.

10.1 OUTLIER DETECTION ENSEMBLES

In our context, a set $X \subseteq \mathbb{R}^N$ of data points can be parted into two subsets: the set of "normal observations" indicated with X_{nor}, and the set of "abnormal observations", indicated with X_{abn}. In unlabeled data, distinguishing normal from anomalous data is not always straightforward. We consider a model for outlier detection g, that aims at classifying each data point $x \in X$ as either *normal* or *anomalous*. Among the various anomaly detection methods, we focus on methods that provide to each data point a score measuring its "outlierness".

Definition 10.1.1. *Given a set of data points X, we call* model *a function $a : X \mapsto \mathbb{R}$ where $a(x)$ represents the outlier score assigned by a to the sample x.*

The higher the value $a(x)$, the more likely x is considered to be an anomaly compared to the set X. On the same set X, various outlier detection models can be constructed. We indicate with $\mathcal{M}_X$ the set of models constructed on X.

Definition 10.1.2. *Given a set of (sub)models $\mathcal{M}_X$, an* ensemble *is a function $A_{\mathcal{M}_X} : X \mapsto \mathbb{R}$ that assigns to each $x \in X$ its average outlier score, i.e.,*

$$A_{\mathcal{M}_X}(x) = \frac{1}{\|\mathcal{M}_X\|} \sum_{a \in \mathcal{M}_X} a(x). \tag{10.1}$$

The ensemble prediction is the average submodel prediction in the set $\mathcal{M}_X$.

Using the trick of projected data points in lower dimensional spaces, we reach the definition of bagging. We indicate with $\mathcal{N}$

the set of coordinates of X and with X_I the set of data points in X projected only on the $I \subseteq \mathcal{N}$ coordinates (or *features*), i.e., given $x \in X$ the corresponding point $x_I = (x_i)_{i \in I}$ and $I \subset \mathcal{N}$. Now we can define a subset $\mathcal{M}_{X_I} \subseteq \mathcal{M}_X$ as the set of submodels that belongs to $\mathcal{M}_X$ trained only on X_I.

The bagging procedure is meant to randomly cover the information in X, considering only the projection of X in smaller-sized subsets. We refer to the size of the data points in the projection as BAG. Having $X \subseteq \mathbb{R}^N$ fixed and BAG $\leq N$, we can get $\binom{N}{\text{BAG}}$ different subsets of size BAG from the N features.

Definition 10.1.3. *After fixing the* BAG*, the* bagging procedure *consists in defining the model $b_{S,a} \in \mathcal{M}_S$ such that $b_{S,a}(x)$ is the result of a model a when trained on the data set X_S and S is a subset of $\mathcal{N}$ whose size is $|S| =$* BAG.

The bagging procedure does not fix either the model a from $\mathcal{M}_X$ or the set $S \subseteq \mathcal{N}$, thus potentially covering, using sufficiently many random seeds, all the information contained in X. We write $b_{S,a|\text{seed}}$ for the specific *bagging submodel* resulting after we fixed the seed for the random sampling of S and the model a. Finally, we can construct the so-called *feature bagging* ensemble based on the bagging technique.

Definition 10.1.4. *Given a dataset X and a set of models $\mathcal{M}_X$, we define the function $g_{\mathcal{M}_X} : X \mapsto \mathbb{R}$ such that it assign to each $x \in X$ the score defined as*

$$g_{\mathcal{M}_X}(x) = \lim_{n \to \infty} \frac{1}{n} \sum_{j=0}^{n} b_{S,a|seed[j]}(x). \tag{10.2}$$

where seed is an eventually infinite vector of randomly drawn seeds.

A similar definition could also be made for non-outlier detection ensembles as long as the output is a linear combination of the submodel predictions. Still, feature bagging is most commonly used in outlier detection.

10.2 THE BAGGED SHAPLEY VALUES

We defined cooperative games as pairs $(\mathcal{N}, f)$ where $\mathcal{N}$ is the set of players, and f is the value function in Section 2.2. The Shapley values are a "fair" assignment of weights to the single players that consider the role of the single players in any single coalition. Given the game $(\mathcal{N}, f)$ and a player $i \in \mathcal{N}$, the *Shapley value of i* is defined by Equation (2.1) , i.e.,

$$\phi_f(i) = \sum_{\mathcal{A} \subseteq \mathcal{N} \setminus i} \frac{1}{N\binom{N-1}{|\mathcal{A}|}} \left[f(\mathcal{A} \cup i) - f(\mathcal{A}) \right].$$

We mentioned the exponential complexity intrinsic of Shapley values; we show that the exact computation of Shapley value-similar scores for feature bagging ensembles can be easily reduced to a polynomial time.

We introduce the BAGGED SHAPLEY VALUES; their definition perfectly aligns with the impossibility of training an ensemble method with less than BAG features. We rewrite the definition of Shapley values from their definition (2.1) for feature bagging ensembles as $\phi_{g_{\mathcal{M}_X}(x)}(i)$, where $x \in X \subseteq \mathbb{R}^N$ is a data point, $g_{\mathcal{M}_X}$ is the feature bagging model and we are interested in assigning to the coordinate i of X an importance score in predicting the overall outlier score $g_{\mathcal{M}_X}(x)$. We define the BAGGED SHAPLEY VALUES:

Definition 10.2.1. *Given a set of data points $X \subseteq \mathbb{R}^N$, a set of (sub)models $\mathcal{M}_X$ and a feature bagging model $g_{\mathcal{M}_X}$ defined over $\mathcal{M}_X$, the* BAGGED SHAPLEY VALUES *are the values*

$$\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i) = \sum_{S \subseteq \mathcal{N}, i \notin S, s \geq \text{BAG}} \frac{N \cdot (s! \cdot (N-s-1)!)}{(N - \text{BAG}) \cdot N!} \cdot \left[g_{\mathcal{M}_{X_{S \cup \{i\}}}}(x) - g_{\mathcal{M}_{X_S}}(x) \right] \tag{10.3}$$

This equation removes terms with magnitude $\propto \frac{\text{BAG}}{N}$, a necessary step, as defining an ensemble model with less than BAG features is not possible. Notice that the higher the dimension of the data points in X is, the smaller the difference between $\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i)$ and $\phi_{g_{\mathcal{M}_X}(x)}(i)$. To somewhat correct for this difference, we add a factor $\frac{N}{N-\text{BAG}}$ to compensate that we are summing over fewer subsets of $\mathcal{N}$.

10.3 THEORETICAL GUARANTEES FOR THE APPROXIMATION

The main result of our study regards the chance to express Shapley values with a limited number of selected bagging submodels, thus avoiding the exponential computational costs of Shapley values.

Theorem 10.3.1. *The* BAGGED SHAPLEY VALUES *can be expressed using a selection of submodels involved in the feature bagging ensemble $g_{\mathcal{M}_X}$. In particular, it holds*

$$\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i) \propto g_{\mathcal{M}_X}(x) - g_{\mathcal{M}_{X_{\mathcal{N} \setminus i}}}(x).$$

Proof. To increase readability, we use the notation

$$k(S, N) = \frac{N}{N - \text{BAG}} \frac{s!(N-s-1)!}{N!}$$

where $s = |S|$ and $N = |\mathcal{N}|$. For abuse of notation and readability, we write S instead of X_S throughout the whole proof. Now, we can rewrite the BAGGED SHAPLEY VALUES in the following way $b_{S,a|\text{seed}}$ and substitute it with $b_{|\text{seed}} \in \mathcal{M}_S$

$$\begin{aligned}
\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i) &= \sum_{S\subseteq\mathcal{N}, i\notin S, s\geq\text{BAG}} k(S,N)\left[g_{S\cup\{i\}}(x) - g_S(x)\right] \\
&= \lim_{n\to\infty} \sum_{S\subseteq\mathcal{N}, i\notin S, s\geq\text{BAG}} k(S,N) \\
&\cdot \left(\frac{\sum_{j=0,\dots,n, b\in\mathcal{M}_{S\cup\{i\}}} b_{|\text{seed}}(x)}{\|\mathcal{M}_{S\cup\{i\}}\|} - \frac{\sum_{j=0,\dots,n, b\in\mathcal{M}_S} b_{|\text{seed}}(x)}{\|\mathcal{M}_S\|}\right)
\end{aligned}$$

where $\mathcal{M}_K = \{a \in \mathcal{M}_X \mid a \text{ restricted to features in } K\}$ is the subset of models that contain only features included in K.

From the previous equation, we see that $\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i)$ is a sum over the same bagging models multiple times, as they are part of various subsets. We can simplify the writing to evaluate each model only once but weight them using some constant factors α_b and β_b:

$$\begin{aligned}
\tilde{\phi}_{g_{\mathcal{M}_X}(x)}(i) = \lim_{n\to\infty} \frac{1}{\|\mathcal{M}_X\|} &\sum_{b\in\mathcal{M}_X} \alpha_b \cdot b_{|\text{seed}}(x) \\
&- \frac{1}{\|\mathcal{M}_{\mathcal{N}\setminus i}\|} \sum_{b\in\mathcal{M}_{\mathcal{N}\setminus i}} \beta_b \cdot b_{|\text{seed}}(x).
\end{aligned} \tag{10.4}$$

We can shuffle our feature labels without changing Equation 10.4, $\alpha_b = \alpha$ and $\beta_b = \beta$ have to be independent on the specific model $b_{|\text{seed}}$. By the same argument, α and β can not depend on the model outputs $b_{|\text{seed}}(x)$. This allows us to choose any model $b(x)$ to compute them; we pick here

$$b(x) = \begin{cases} 1 & \text{if model } b \text{ considers feature i} \\ 0 & \text{otherwise} \end{cases}. \tag{10.5}$$

Using the proposed $b(x)$, the β term disappears, thus we can write α as

$$\begin{aligned}\alpha &= \lim_{n\to\infty} \frac{\sum_{S\subseteq\mathcal{N}, i\notin S, |S|\geq\textsc{bag}} k(S,N) \frac{\|\mathcal{M}_X\|}{\|\mathcal{M}_{X_{S\cup\{i\}}}\|} \sum_{b\in\mathcal{M}_{X_{S\cup\{i\}}}} b(x)}{\sum_{b\in\mathcal{M}_X} b(x)} \\ &= \lim_{n\to\infty} \frac{\sum_{S\subseteq\mathcal{N}, i\notin S, |S|\geq\textsc{bag}} k(S,N) \cdot \frac{\text{count}(\mathcal{M}_{X_{S\cup\{i\}}})}{\|\mathcal{M}_{X_{S\cup\{i\}}}\|}}{\frac{\text{count}(\mathcal{M}_X)}{\|\mathcal{M}_X\|}}\end{aligned}$$

where $\text{count}(\mathcal{M}_{X_K})$ is the number of models in $\mathcal{M}_{X_K}$ that contain one specific feature in K. We can use

$$\lim_{n\to\infty} \frac{\text{count}(\mathcal{M}_{X_K})}{\|\mathcal{M}_{X_K}\|} = \frac{\binom{|K|-1}{\textsc{bag}-1}}{\binom{|K|}{\textsc{bag}}} = \frac{\textsc{bag}}{|K|}$$

thus getting

$$\begin{aligned}\alpha =& N\frac{N}{N-\textsc{bag}} \sum_{s=\textsc{bag}}^{N-1} \binom{N}{s} \cdot \frac{s!(N-s-1)!}{N!} \cdot \frac{1}{s+1} \\ =& \frac{N}{N-\textsc{bag}} \sum_{s=\textsc{bag}}^{N-1} \cdot \frac{1}{s+1} \\ =& \frac{N}{N-\textsc{bag}} \cdot (\psi^0(N+1) - \psi^0(\textsc{bag}+1))\end{aligned}$$

with the digamma function ψ^0.

When instead of choosing $b(x)$ to be independent of i, we find that $\tilde{\phi}_{f_{\mathcal{M}_X}(x)}(i) \propto (\alpha - \beta)$. But since the feature is designed not to have any effect, we know that $\tilde{\phi}_{f_{\mathcal{M}_X}(x)}(i) = 0$ and thus $\alpha = \beta$. This concludes the proof. □

The results not only show that the bagged Shapley value is proportional to the difference of two feature bagging, respectively defined on $\mathcal{M}_X$ and $\mathcal{M}_{X_{\mathcal{N}\setminus i}}$, but also that when using bagging models, we can estimate the BAGGED SHAPLEY VALUES in polynomial time. This is because for deterministic submodels, instead of using ∞ of them, we only need to train $\binom{N}{\textsc{bag}} < N^{\textsc{bag}}$ submodels. The code is available online[1].

10.4 EXPERIMENTS

We evaluate our approach on various freely available real-world datasets with varying numbers of features [Tri+17; Den12; Liu+15].

[1] Code available at `chiarabales/ensembleShapley`

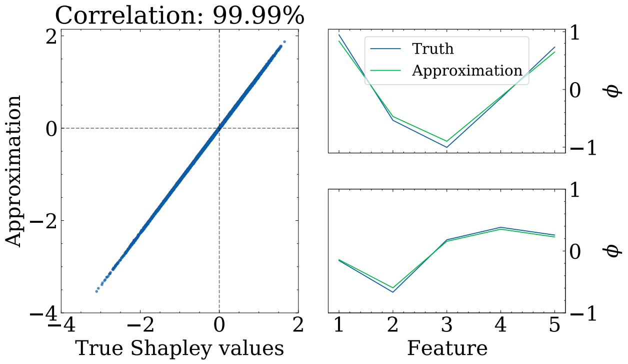


Figure 10.1: Approximation accuracy. On the left, plot of the bagged Shapley values against the exact Shapley values for each data sample in the phoneme dataset. On the right, Shapley values and their approximation for two example samples.

We conduct experiments on the correctness of the approximation (Section 10.4.1), the effectiveness (Section 10.4.2), and the scalability (Section 10.4.3) of our approach.

10.4.1 *Quality of the Approximation*

To fairly investigate the approximation accuracy of the bagged Shapley values, we use a low-dimensional dataset, i.e., the five-dimensional phoneme dataset [Tri+17] that requires the training of feature bagging ensemble models only 2^5 times. The low dimensionality of the dataset allows us to compute the non-approximated version of Shapley values without incurring extremely long runtimes. We train isolation trees from [LTZ08] with a bagging size of 2 and simplify the obtained anomaly score to fit our methodology by using the negative average path length over all trees as an indicator of anomalies. We train one million submodels and average the obtained results to guarantee consistent and robust results. The total training takes about 70min of CPU time[2]. The ROC-AUC score is 0.733.

We separate the trained models into ensembles for each subset and compute the exact Shapley values and the bagged Shapley values. We combine the values obtained into Figure 10.1. As the mapping lies on the diagonal line, we conclude that the approximation works well on all data points.

[2] All experiments were performed on Intel Xeon E5 CPUs. In the paper, we also stick to CPUs over GPUs when we use neural network submodels; the choice is justified by the higher amount of parallelization they allow.

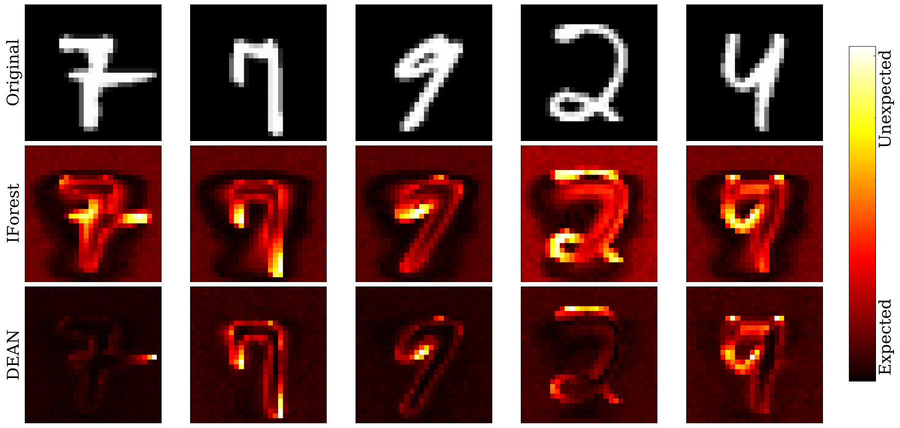


Figure 10.2: MNIST DATASET. The original images are in the top row. The bottom rows contain the derived BAGGED SHAPLEY VALUES heatmap for ISOR and DEAN. We rescaled the color legend to each plot's Shapley values' upper and lower bounds.

10.4.2 *Effectiveness*

We can compute the BAGGED SHAPLEY VALUES for datasets whose dimensions are too high for an exact computation. We focus on the MNIST dataset [Den12], a collection of images of hand-written digits usually used to train image-recognition models. Following the approach of [Ruf+18], we consider normal all images representing a handwritten "seven", and anomalous the images representing other digits. Each image has a resolution of 28×28, i.e., we handle 784 features in each image. Computing the exact Shapley values for the single pixels requires $2^{784} \approx 10^{237}$ evaluations, a number significantly larger than the computational power available.

For the BAGGED SHAPLEY VALUES, we use the bagging size $\text{BAG} = 32$. We train two models: we use DEAN, a deep learning model-based ensemble, and a shallow isolation forest [LTZ08]. We choose DEAN [BKM22] because of its inherited feature bagging and relatively low training time per submodel. The training time is significantly longer than using IForest[3]. Note that we do not only train a model on each possible subset, as the number of subsets is still $\binom{768}{32} \approx 4 \cdot 10^{32}$. Instead, we train on random subsets until the result converges. This also helps deal with the random nature of our algorithms.

Figure 10.2 represents the plots of the Shapley values for five representative samples in the form of heatmaps; bright colors represent high scores, i.e., features highly increasing the outlier

[3] The isolation forest takes about 220min of CPU time. DEAN requires about 113days; however, the independent ensembles are easy to parallelize, and less accurate results can already be achieved with ten thousand submodels (27 hours).

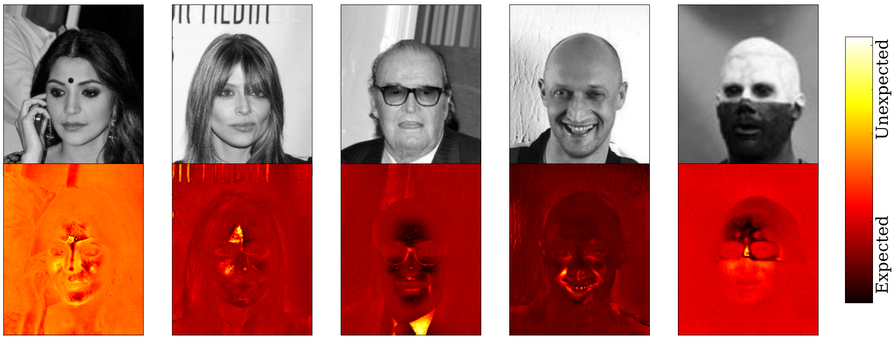


Figure 10.3: CELEBA DATASET. The heatmaps show the BAGGED SHAPLEY VALUES; brighter colors indicate features having a strong influence on the anomaly score's prediction.

score. Each heatmap, both for DEAN and IForest, highlights the changes to the original input that would make it closer to a normal observation by highlighting the erroneous regions. From the left to the right side, the first two input images are labeled as normal; however, they still contain features that are not expected, e.g., the middle horizontal line in the first image. These unexpected features are highlighted in bright red and yellow. Similarly, the other three images obtain high outlier scores, although they contain typical features for normal input images. These features are also unexpected by the model and thus result in high Shapley values. Examples are represented by the "nine" and the "four"; removing the lower line from the circle would make the "nine" more similar to a normal observation while adding a horizontal line to the top would make the "four" more similar to a "seven".

Comparing DEAN and IForest, we see how the understanding of the "normal" concept, i.e., the digit "seven", of the isolation forest, is too simple to explain the predictions entirely. In the second column of Figure 10.2, we see that the isolation forest expects the tail of the "seven" to bend instead of going straight down. On the other hand, based on a deep learning method, DEAN has less difficulty learning a broader concept of "seven". This is also reflected in the outlier detection performance: while DEAN reaches a ROC-AUC of 0.9698 on the dataset, the isolation forest only reaches a lower 0.9118 score. We strongly believe that the BAGGED SHAPLEY VALUES's saliency maps provide useful insights into what the model understood and learned from the training data, additionally to better performance measured by the ROC-AUC metric.

10.4.3 *Scalability*

We select the celebA dataset [Liu+15] to study how the approach scales to larger datasets. celebA contains images with $218 \times 178 = 38804$ pixels, which we convert to grayscale to simplify the plotting. In the previous section, we showed how complex patterns can overwhelm outlier detection ensembles that struggle to learn a proper schema for normal and abnormal data points. Thus, we aim to maximize the separation between normal and abnormal classes to simplify the learning task. We divide the dataset into normal and anomalous instances, where we characterize a normal observation being labeled with the attributes "female", "young", "attractive", and "not bald". The inverse attributes characterize an abnormal observation. Here, the choice of attributes was only guided by the distribution of attributes in the dataset, and similar results would likely have followed any other choices for the anomalous and normal classes. We only trained the DEAN ensemble on the dataset, as the model proved to handle complicated attributed data better. We represent the obtained BAGGED SHAPLEY VALUES as heatmaps on five images in Figure 10.3. The first row is the input image, while the second contains the corresponding Shapley values.

The images resulting from the BAGGED SHAPLEY VALUES plotting have high resolution and show some features as more anomalous; however, the designed features do not match the designed separation in normal and abnormal images. This can also be seen in the ROC-AUC score of 0.6184. From left to right, the most anomalous features seem to be the bindi, the partially covered forehead, the shirt collar, the laugh lines, and the skin paint transition. These are rare features in the images of young women in celebA, thus considered "anomalous" by the model. Still, the complexity of the separation is likely too big for the available samples (≈ 72000), and thus, the learning, as shown by the ROC-AUC, is inaccurate. Although the features outlined are not the expected ones from our understanding of the separation between the two classes, it is worth noticing how the maps of BAGGED SHAPLEY VALUES can be used to understand and improve the outlier detection models. The runtime of the training procedure for one million DEAN submodels is ≈ 468 days; training 500 submodels simultaneously requires about 4 days of CPU time. Under parallelization assumption, we use 4 millions of submodels in our training setup and set up the bagging size to be $\text{BAG} = 32$. A different bagging size might have achieved more accurate results, but we did not optimize it since, in most contexts, the outlier detection task sets the bagging size. We finally want to characterize the minimum number of submodels

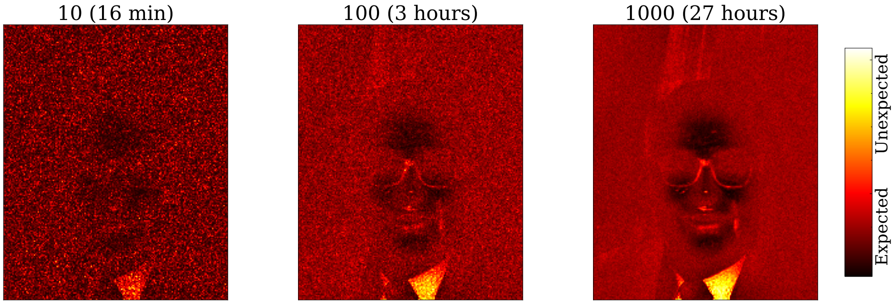


Figure 10.4: Influence of the number of submodels on the bagged Shapley values. Here we use 12127, 121263, and 1212625 submodels so that each feature is approximately sampled 10, 100, and 1000 times. The times stated assume a parallelization with 500 CPUs.

needed for our methodology to perform well. For this, we calculate the bagged Shapley values maps so that each feature is used 10, 100, and 1000 times. The corresponding maps for the central image of Figure 10.3 are shown in Figure 10.4. While some features are already visible at about 12000 submodels, the noise level being still very high, facial features are undetectable; with about 10, those become visible while extensively the number of submodels to about 100 times more, they have become clear.

As a rule of thumb, we suggest training $10 \cdot N$ features to visualize the basic features and to train $10 \cdot N^{\frac{3}{2}}$ for clear images.

Part III

SUMMARY

11 SUMMARY AND FUTURE WORK

Several solutions involving rankings and importance scores have been proposed in this thesis, focusing on unlabelled data mining applications, trustworthiness, and consistency of evaluations. Part I focused on unlabeled tabular data and unlabeled time series. We explored Shapley values to evaluate and measure the correlation structure among the data features and track eventual changes over time. In particular, we used the obtained importance scores to rank features in unlabelled data sets and pathways in collections of gene sets, respectively, in Chapter 4 and Chapter 5. Using a label-independent value function enabled us to bridge the gap between unsupervised feature selection and cooperative game theory. In Chapter 4 and Chapter 6, the total correlation-based Shapley values summarize the correlation structure of the datasets in feature importance scores; these scores aggregate the measured correlations from an exponential amount of feature subsets in a unique score per feature. They allow (a) ranking and (b) selecting features preserving the correlation structure of unlabeled datasets, as shown in Chapter 4, and (c) keeping track of correlations among univariate dimensions of unlabeled time series, and detecting happening drifts, as shown in Chapter 6. In Chapter 4, the two introduced algorithms, SVFS and SVFR, integrate Shapley values with redundancy awareness. In particular, we introduced SVFS as an unsupervised redundancy-aware feature selection method and SVFR as an unsupervised redundancy-aware feature ranking approach; the experiments confirm the low redundancy rate retained in the selected features on numerous data sets in comparison to other unsupervised feature selection methods [CZH10; Zhu+19; ZL07]. Chapter 6 proposed a new unsupervised change point visualization and detection method for time series with discrete values. SLIDSHAPs allow us to visualize change points in the correlation structure of the time series and provide a method to relocate identified change points in the multivariate original time series; the method resulted in being more effective in detecting changing points related to the correlation structure than other unsupervised approaches [BG07; DP11].

Furthermore, we introduced in Chapter 5 an application-oriented method for reducing the dimension and redundancy of collections of sets. We gave a practice application and a straightforward motivation based on a case study on collections of gene sets. We used the Shapley values as importance scores that consider the distribution of elements within the family of sets and their overlap. We added various pruning techniques to obtain overlapping free rankings and aimed to satisfy various properties when selecting sets. Interestingly, using microarray games [MPB07] allows for efficient computing of Shapley value, even in cases where the number of players is large. During our experiments on collections of gene sets, we studied the effect of the rankings of pathways under several aspects: we observed a reduction in overlapping of sets, high coverage of the genes, and similar statistical significance to one of the original collections using much smaller ones. However, the range of potential applications is much broader. The proposed ranking methods can be applied in any family of sets or where the relationship can be expressed as a binary matrix B.

Part II focuses on theoretical aspects and trustworthiness issues, again focusing on rankings and importance scores. The objective is to critically examine some commonly used interpretability methods and metrics and introduce a new interpretable method for bagging-based anomaly detection. Relevant to the trustworthiness of machine learning, we focus on the consistency of evaluations of machine learning methods and saliency map interpretations [Kri+22]. Chapter 8 considered ranking evaluation metrics, primarily derived from Recommender Systems and Information Retrieval techniques and used to evaluate rankings in applications-independent contexts; Chapter 9 focuses on saliency explanations for time series classification methods and reveals how these explanations are mostly inconsistent on the invariant time-windows structure of time series.

In Chapter 8, we provided theoretical and experimental insights on the necessity of careful choices for ranking evaluation metrics on symmetric groups. We showed that non-consistent evaluations appear when using ranking evaluation metrics and proposed theoretical properties to understand these metrics better. Our investigation illustrates how most metrics do not distinguish small changes, how single swaps and slides of the rankings influence their evaluation, and how robust the metrics are. We additionally gave insights into the interplay among these properties and attempted to define distances on symmetric groups.

In Chapter 9, we studied yet another context where consistency of methods plays a role. While explanations based on saliency maps have succeeded in vision and natural language domains, they remain challenging for time series data, where explanations struggle with the time-features structure [Ism+20]. We analyzed the use of saliency explanations initially thought for computer vision applications [HS97; Lea+17; Vas+17] to interpret time series methods; there, we identified issues related to *inconsistencies* rising in saliency explanations over overlapping time windows and *non-robustness* when swapping features in time series windows. As an exploratory analysis, we aimed to raise awareness of the described problems [Kri+22] and motivate further development of saliency methods that address the existing flaws.

Finally, we work on explaining anomaly detection approaches, another typical unsupervised application. Some explanation methods are already available for anomaly detection [TI14; TC20]; however, the computational complexity of Shapley values does not allow for large-scale implementations. We combine Shapley values with ensemble techniques [BKM22], explicitly focusing on feature-bagging ensembles for outlier detection. The BAGGED SHAPLEY VALUES offer an advantageous reduction of the computational costs, giving a chance to compute importance scores for settings with tens of thousands of features. Furthermore, we showed the value of highlighting anomalous features in images to obtain insights into the features learned by the outlier detection method. Our experiments showed that, combined with ensemble methods, the computation of Shapley values-based explanations has a polynomial runtime. We believe that combining Shapley values with ensemble methods can boost the use of Shapley values in the machine learning community, showing advantages from a computational and interpretability point of view, and leading to better, more reliable, outlier detection models.

11.0.1 *Future research directions*

The thesis is the starting point of several research directions. The NP-hardness of the Shapley values computation is a considerable obstacle that still needs to be overcome. The number of features and the dimension of the multivariate time series had to fit within the framework, thus barely allowing for consideration of real-world problems, where we often deal with higher-dimensional data sets. The literature overflows out of approximation approaches [CGT09; LL17]. Choosing one of them is, again, a challenge; it is context-dependent and not uniquely solved. In our experiments, we often referred to Castro et al. [CGT09] as an approximation technique that, with its simplicity, offers an easy-implementing solution. We believe in the necessity of summarizing and revealing the theoretical properties and benchmarking the available approximations; the result will eventually offer an understanding of whether it is necessary to implement additional methods. It is worth noting the possibility of using non-approximated Shapley values under highly specific conditions; Chapter 5 and Chapter 10 are examples of this. We further believe they are not the only two cases in which an exact computation is feasible in polynomial time.

The data type represents a limitation of Chapter 4 and Chapter 6. Both chapters focus on the case of categorical or discrete data sets, a direct consequence of using total correlation as a value function [Cov99]. We implemented several techniques to overcome the mentioned limitation, mainly exploring approximation techniques, acting on the data as a preprocessing step, substituting the value function, approximating the total correlation, and applying encoding techniques, e.g., [KLF05]. The differential Shannon entropy for the total correlation computation was considered a starting point. However, research into overcoming the issue has not yet come to a full stop.

An interesting aspect not tackled in the thesis regards the societal impact of rankings and the assignment of importance scores in critical applications. Feature importance scores are ubiquitous and created from the most disparate settings; the induced orderings result in recommendations, information retrieval, selections, and deselections of items and individuals. In context-critical real-world applications, it is sometimes unclear if their use is legitimate or induces or propagates biases. Biases – against races, sexes, sexual orientations, and minorities – represent a threat to societal applications where the models' predictions can discriminate against underrepresented and protected communities. We often mentioned how Shapley values represent a "individual fair" assignment of importance scores; however, there is no trace of "group fairness" in their definition. Adding fairness to importance scores, rank aggregation methods, and Recommender Systems represents a tiny step toward addressing this significant challenge. However, how do we address the bias problem when people interact with Recommender Systems? How do we ensure that biases in rankings are not propagated through rank aggregation strategies? The last two questions are fascinating and challenging. For all contexts where fairness concerns represent a danger, additional research to add "fairness constraints" to existing methods is essential to spread trustworthy machine learning methodologies to critical applications.

Finally, we highlight the need for trustworthy and interpretable machine-learning techniques. In this direction, the state-of-the-art literature needs more methods for unsupervised applications and unlabeled data, particularly for time series data, where methods based on unlabeled data streams are hardly interpretable. We plan to extend the slidSHAP series to increase the interpretability of the detected change points, using the shifts in the slidSHAPs in the neighbor of the change points to predict how they influence the input variable correlation in future timestamps. Furthermore, Chapter 9 has left us with a problem to solve and without a solution. The inconsistency issue could be solved using Siamese networks, simultaneously minimizing the consistency of explanations and accuracy. In Chapter 9, we focused on our time series classification setup; however, the inconsistency among interpretability methods is a much broader issue [Kri+22]. Furthermore, the disagreement among interpretability methods is fragile with respect to "fairwashing" [Aïv+19] attacks. Fairwashing refers to the malicious use of saliency maps, importance scores, and explanation methods to voluntarily induce a false perception in the user, mainly regarding the fairness of the machine learning models. An increased awareness of eventual inconsistencies among metrics, explanations, and methods is needed to develop new solutions and guarantee an informed use of Machine Learning methods.

11.0.2 *Conclusions*

This thesis proposes methodological solutions in Part I while focusing on a theoretical analysis of fundamental aspects of methods' trustworthiness in Part II. We conclude by using the exact words we used when we started it. Rankings and importance scores offer an easy, interpretable solution to many Machine Learning challenges. By studying the data structure from a "correlation" point of view, we explored unlabelled data, offering solutions on reducing their dimensions when predictions' goals are unclear or absent; using Shapley values-based importance scores in unlabeled setups, we detect correlational concept drifts in unlabeled data streams, decrease the size of a collection of gene sets, and introduce a redundancy-free ranking of features. Additionally, we highlighted the risks of mindlessly using ranking evaluation metrics across different contexts and transposing saliency explanation methods from computer vision to interpret time series classification models; we concluded with the introduction of a new Shapley values-based anomaly detector explanation approach.

Future challenges also involve rankings and important scores when formulating problems or solutions. A handful of questions remained unanswered, and many others keep us interested in consequential and non-strictly related future research directions. We hope we have convinced the readers about both the potential and the risks of using rankings and importance scores in Machine Learning solutions and that the proposed advances will have a positive future impact.

BIBLIOGRAPHY

[AT05] Gediminas Adomavicius and Alexander Tuzhilin. "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions." In: *IEEE Transactions on Knowledge and Data Engineering* 17.6 (2005).

[Agr18] Alan Agresti. *An Introduction to Categorical Data Analysis*. 2018.

[AK18] Zahra Ahmadi and Stefan Kramer. "Modeling recurring concepts in data streams: a graph-based framework." In: *Knowledge and Information Systems* (2018).

[Aïv+19] Ulrich Aïvodji, Hiromi Arai, Olivier Fortineau, Sébastien Gambs, Satoshi Hara, and Alain Tapp. "Fairwashing: the risk of rationalization." In: *ICML*. 2019.

[AP96] Kamal M Ali and Michael J Pazzani. "Error reduction through learning multiple descriptions." In: *Machine Learning* 24 (1996).

[ASC18] Enrique Amigó, Damiano Spina, and Jorge Carrillo-de-Albornoz. "An axiomatic analysis of diversity evaluation metrics: Introducing the rank-biased utility metric." In: *SIGIR*. 2018.

[Ant+21] Liat Antwarg, Ronnie Mindlin Miller, Bracha Shapira, and Lior Rokach. "Explaining anomalies detected by autoencoders using Shapley Additive Explanations." In: *Expert Systems with Applications* (2021).

[Bac+15] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." In: *PloS ONE* 10.7 (2015).

[Bae+10] David Baehrens, Timon Schroeter, Stefan Harmeling, Motoaki Kawanabe, Katja Hansen, and Klaus-Robert Müller. "How to explain individual classification decisions." In: *The Journal of Machine Learning Research* 11 (2010).

[Bae+06] Manuel Baena-Garcıa, José del Campo-Ávila, Raúl Fidalgo, Albert Bifet, R Gavalda, and Rafael Morales-Bueno. "Early drift detection method." In: *Workshop on Knowledge Discovery from Data Streams*. 2006.

[Bal+22] Chiara Balestra, Florian Huber, Andreas Mayr, and Emmanuel Müller. "Unsupervised Features Ranking via Coalitional Game Theory for Categorical Data." In: *DaWaK*. 2022.

[BLM23a] Chiara Balestra, Bin Li, and Emmanuel Müller. "On the Consistency and Robustness of Saliency Explanations for Time Series Classification." In: *arXiv preprint arXiv:2309.01457* (2023).

[BLM23b] Chiara Balestra, Bin Li, and Emmanuel Müller. "slidSHAPs – sliding Shapley Values for correlation-based change detection in time series." In: *DSAA*. 2023.

[Bal+23] Chiara Balestra, Carlo Maj, Emmanuel Müller, and Andreas Mayr. "Redundancy-aware unsupervised ranking based on game theory: Ranking pathways in collections of gene sets." In: *Plos one* 18.3 (2023), e0282699.

[BMM24] Chiara Balestra, Andreas Mayr, and Emmanuel Müller. "Ranking evaluation metrics from a group-theoretic perspective." In: *under review*. 2024.

[Bel+15] Frida Belinky, Noam Nativ, Gil Stelzer, Shahar Zimmerman, Tsippi Iny Stein, Marilyn Safran, and Doron Lancet. "PathCards: multi-source consolidation of human biological pathways." In: *Database* 2015 (2015).

[BH95] Yoav Benjamini and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 57.1 (1995).

[BY01] Yoav Benjamini and Daniel Yekutieli. "The control of the false discovery rate in multiple testing under dependency." In: *Annals of Statistics* (2001).

[Ben+21a] João Bento, Pedro Saleiro, André F Cruz, Mário AT Figueiredo, and Pedro Bizarro. "Timeshap: Explaining recurrent models through sequence perturbations." In: *KDD*. 2021.

[Ben+21b] João Bento, Pedro Saleiro, André F. Cruz, Mário A. T. Figueiredo, and Pedro Bizarro. "TimeSHAP: Explaining Recurrent Models through Sequence Perturbations." In: *KDD*. 2021.

[BG07] Albert Bifet and Ricard Gavalda. "Learning from time-changing data with adaptive windowing." In: *SDM*. 2007.

[Bif+13] Albert Bifet, Bernhard Pfahringer, Jesse Read, and Geoff Holmes. "Efficient data stream classification via probabilistic adaptive windows." In: *ACM Symposium on Applied Computing*. 2013.

[BA95] J Martin Bland and Douglas G Altman. "Multiple significance tests: the Bonferroni method." In: *Bmj* (1995).

[BKM22] Benedikt Böing, Simon Klüttermann, and Emmanuel Müller. "Post-Robustifying Deep Anomaly Detection Ensembles by Model Selection." In: *ICDM*. 2022.

[Bre96] Leo Breiman. "Bagging predictors." In: *Machine Learning* 24 (1996).

[BV04] Chris Buckley and Ellen M Voorhees. "Retrieval evaluation with incomplete information." In: *SIGIR*. 2004.

[BC21] Mark Alexander Burgess and Archie C Chapman. "Approximating the Shapley Value Using Stratified Empirical Bernstein Sampling." In: *IJCAI*. 2021.

[CZH10] Deng Cai, Chiyuan Zhang, and Xiaofei He. "Unsupervised feature selection for multi-cluster data." In: *KDD*. 2010.

[Cam+18] Tjeerd van Campen, Herbert Hamers, Bart Husslage, and Roy Lindelauf. "A new approximation method for the Shapley value applied to the WTC 9/11 terrorist attack." In: *Social Network Analysis and Mining* (2018).

[CGT09] Javier Castro, Daniel Gómez, and Juan Tejada. "Polynomial calculation of the Shapley value based on sampling." In: *Computers & Operations Research* 36.5 (2009).

[Cat+21] Amnon Catav, Boyang Fu, Yazeed Zoabi, Ahuva Libi Weiss Meilik, Noam Shomron, Jason Ernst, Sriram Sankararaman, and Ran Gilad-Bachrach. "Marginal contribution feature importance-an axiomatic approach for explaining data." In: *ICML*. 2021.

[CMO16] Rodolfo C Cavalcante, Leandro L Minku, and Adriano LI Oliveira. "Fedd: Feature extraction for explicit concept drift detection in time series." In: *IJCNN*. 2016.

[Che+13] Edward Y Chen, Christopher M Tan, Yan Kou, Qiaonan Duan, Zichen Wang, Gabriela Vaz Meirelles, Neil R Clark, and Avi Ma'ayan. "Enrichr: interactive and collaborative HTML5 gene list enrichment analysis tool." In: *BMC Bioinformatics* 14.1 (2013).

[Che+15] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. *The UCR Time Series Classification Archive*. 2015.

[Cho+16] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. "Retain: an interpretable predictive model for healthcare using reverse time attention mechanism." In: *NeurIPS* (2016).

[CDR05] Shay Cohen, Gideon Dror, and Eytan Ruppin. "Feature selection based on the Shapley value." In: *IJCAI* (2005).

[Con99] William Jay Conover. *Practical nonparametric statistics*. Vol. 350. 1999.

[CKS86] Wade D Cook, Moshe Kress, and Lawrence M Seiford. "An axiomatic approach to distance on partial orderings." In: *RAIRO-Operations Research* 20.2 (1986).

[Cos+17] Fausto G da Costa, Felipe SLG Duarte, Rosane MM Vallim, and Rodrigo F de Mello. "Multidimensional surrogate stability to detect data stream concept drift." In: *Expert Systems with Applications* (2017).

[Cov99] Thomas M Cover. *Elements of information theory*. 1999.

[CKL22] Ian Covert, Chanwoo Kim, and Su-In Lee. *Learning to Estimate Shapley Values with Vision Transformers*. 2022.

[Das+06] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. "An information-theoretic approach to detecting changes in multi-dimensional data streams." In: *Interface of Statistics, Computing Science, and Applications*. 2006.

[Den12] Li Deng. "The mnist database of handwritten digit images for machine learning research." In: *IEEE Signal Processing Magazine* 29 (2012).

[Dia88] Persi Diaconis. "Group representations in probability and statistics." In: *Lecture notes-monograph series* 11 (1988).

[Dis+20] Theekshana Dissanayake, Tharindu Fernando, Simon Denman, Sridha Sridharan, Houman Ghaemmaghami, and Clinton Fookes. "A robust interpretable deep learning classifier for heart anomaly detection without segmentation." In: *IEEE Journal of Biomedical and Health Informatics* 25.6 (2020).

[DP11] Gregory Ditzler and Robi Polikar. "Hellinger distance based drift detection for nonstationary environments." In: *CIDUE*. 2011.

[Dod+12] Mark S Doderer, Zachry Anguiano, Uthra Suresh, Ravi Dashnamoorthy, Alexander JR Bishop, and Yidong Chen. "Pathway Distiller-multisource biological pathway consolidation." In: *BMC Genomics* 13 (2012).

[Dom+19] Ann-Kathrin Dombrowski, Maximillian Alber, Christopher Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. "Explanations can be manipulated and geometry is to blame." In: 2019.

[DG17] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017.

[DVL08] Sandrine Dudoit, Mark J Van Der Laan, and Mark J van der Laan. *Multiple testing procedures with applications to genomics*. 2008.

[Dur08] John R Durbin. *Modern algebra: an introduction*. 2008.

[DM21] Alexandre Duval and Fragkiskos D Malliaros. "Graphsvx: Shapley value explanations for graph neural networks." In: *ECML PKDD*. 2021.

[Dwo+01] Cynthia Dwork, Ravi Kumar, Moni Naor, and Dandapani Sivakumar. "Rank aggregation methods for the web." In: *WWW*. 2001.

[Fis35] R. A. Fisher. "The Logic of Inductive Inference." In: *Journal of the Royal Statistical Society* 98.1 (1935).

[FV86] Michael A Fligner and Joseph S Verducci. "Distance based ranking models." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 48.3 (1986).

[Fri+14] Isvani Frias-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jimenez, Rafael Morales-Bueno, Agustin Ortiz-Diaz, and Yaile Caballero-Mota. "Online and non-parametric drift detection methods based on Hoeffding's bounds." In: *IEEE Transactions on Knowledge and Data Engineering* (2014).

[Gam+04] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. "Learning with drift detection." In: *Brazilian Symposium on Artificial Intelligence*. 2004.

[Gem+20] Rosana Noronha Gemaque, Albert França Josuá Costa, Rafael Giusti, and Eulanda Miranda Dos Santos. "An overview of unsupervised drift detection methods." In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* (2020).

[Gös+21] Martijn Gösgens, Anton Zhiyanov, Aleksey Tikhonov, and Liudmila Prokhorenkova. "Good classification measures and how to find them." In: *NeurIPS*. 2021.

[GTP21] Martijn M Gösgens, Alexey Tikhonov, and Liudmila Prokhorenkova. "Systematic analysis of cluster similarity indices: How to validate validation measures." In: *ICML*. 2021.

[Gui+20] Riccardo Guidotti, Anna Monreale, Francesco Spinnato, Dino Pedreschi, and Fosca Giannotti. "Explaining Any Time Series Classifier." In: *CogMI*. 2020.

[GSY12] Asela Gunawardana, Guy Shani, and Sivan Yogev. "Evaluating recommender systems." In: *Recommender Systems Handbook*. 2012.

[Hal+21] Ben Halstead, Yun Sing Koh, Patricia Riddle, Mykola Pechenizkiy, Albert Bifet, and Russel Pears. "Fingerprinting concepts in data streams with supervised and unsupervised meta-information." In: *ICDE*. 2021.

[HM14] Farzad Farnoud Hassanzadeh and Olgica Milenkovic. "An axiomatic approach to constructing distances for rank comparison and aggregation." In: *IEEE Transactions on Information Theory* 60.10 (2014).

[HCN05] Xiaofei He, Deng Cai, and Partha Niyogi. "Laplacian score for feature selection." In: *NeurIPS* (2005).

[Her+04] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. "Evaluating collaborative filtering recommender systems." In: *TOIS* 22.1 (2004).

[Ho05] Shen-Shyang Ho. "A martingale framework for concept change detection in time-varying data streams." In: *ICML*. 2005.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural Computation* 9.8 (1997).

[Hoy+22] Charles Tapley Hoyt, Max Berrendorf, Mikhail Galkin, Volker Tresp, and Benjamin M Gyori. "A unified framework for rank-based evaluation metrics for link prediction in knowledge graphs." In: *arXiv preprint arXiv:2203.07544* (2022).

[Hun+18] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding." In: *SIGKDD*. 2018.

[Ier+08] Martijn P van Iersel, Thomas Kelder, Alexander R Pico, Kristina Hanspers, Susan Coort, Bruce R Conklin, and Chris Evelo. "Presenting and exploring biological pathways with PathVisio." In: *BMC Bioinformatics* 9.1 (2008).

[Ism+20] Aya Abdelsalam Ismail, Mohamed Gunady, Hector Corrada Bravo, and Soheil Feizi. "Benchmarking deep learning interpretability in time series predictions." In: *Advances in Neural Information Processing Systems* 33 (2020).

[Jac01] Paul Jaccard. "Etude de la distribution florale dans une portion des Alpes et du Jura." In: *Bulletin de la Societe Vaudoise des Sciences Naturelles* 37 (1901).

[JW19] Sarthak Jain and Byron C Wallace. "Attention is not explanation." In: *arXiv preprint arXiv:1902.10186* (2019).

[JK02] Kalervo Järvelin and Jaana Kekäläinen. "Cumulated gain-based evaluation of IR techniques." In: *TOIS* 20.4 (2002).

[KB01] Timor Kadir and Michael Brady. "Saliency, scale and image description." In: *International Journal of Computer Vision* 45.2 (2001).

[Kaj+19] Deepak A Kaji, John R Zech, Jun S Kim, Samuel K Cho, Neha S Dangayach, Anthony B Costa, and Eric K Oermann. "An attention based deep learning model of clinical events in the intensive care unit." In: *PloS ONE* 14.2 (2019).

[KMB12] Fabian Keller, Emmanuel Müller, and Klemens Böhm. "HiCS: High Contrast Subspaces for Density-Based Outlier Ranking." In: *ICDE*. 2012.

[KLF05] Eamonn Keogh, Jessica Lin, and Ada Fu. "Hot sax: Efficiently finding the most unusual time series subsequence." In: *ICDM*. 2005.

[KD22] Utkarsh Mahadeo Khaire and R Dhanalakshmi. "Stability of feature selection algorithm: a review." In: *Journal of King Saud University-Computer and Information Sciences* 34.4 (2022).

[Kim+13] Minji Kim, Fardad Raisali, Farzad Farnoud, and Olgica Milenkovic. "Gene prioritization via weighted Kendall rank aggregation." In: *CAMSAP*. 2013.

[KBM24] Simon Klüttermann, Chiara Balestra, and Emmanuel Müller. "On the efficient Explanation of Outlier Detection Ensembles through Shapley Values." In: *PAKDD*. 2024.

[KM23] Simon Klüttermann and Emmanuel Müller. "Evaluating and Comparing Heterogeneous Ensemble Methods for Unsupervised Anomaly Detection." In: *IJCNN*. 2023.

[Kri+22] Satyapriya Krishna, Tessa Han, Alex Gu, Javin Pombra, Shahin Jabbari, Steven Wu, and Himabindu Lakkaraju. "The disagreement problem in explainable machine learning: a practitioner's perspective." In: *arXiv preprint arXiv:2202.01602* (2022).

[Kul+16] Maxim V Kuleshov, Matthew R Jones, Andrew D Rouillard, Nicolas F Fernandez, Qiaonan Duan, Zichen Wang, Simon Koplev, Sherry L Jenkins, Kathleen M Jagodnik, Alexander Lachmann, et al. "Enrichr: a comprehensive gene set enrichment analysis web server 2016 update." In: *Nucleic Acids Research* 44.W1 (2016).

[LR78] A Verdugo Lazo and P Rathie. "On the entropy of continuous probability distributions (corresp.)" In: *IEEE Transactions on Information Theory* 24.1 (1978).

[Lea+17] Colin Lea, Michael D Flynn, Rene Vidal, Austin Reiter, and Gregory D Hager. "Temporal convolutional networks for action segmentation and detection." In: *CVPR*. 2017.

[LBM22] Bin Li, Chiara Balestra, and Emmanuel Müller. "Enabling the visualization of distributional shift using Shapley values." In: *distSHIFT@NIPS*. 2022.

[Li+17] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. "Feature selection: a data perspective." In: *ACM Computing Surveys (CSUR)* 50.6 (2017).

[Li+12] Zechao Li, Yi Yang, Jing Liu, Xiaofang Zhou, and Hanqing Lu. "Unsupervised feature selection using nonnegative spectral analysis." In: *AAAI*. 2012.

[LZV23] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. "A survey on explainable anomaly detection." In: *ACM Transactions on Knowledge Discovery from Data* 18.1 (2023).

[Lib+15] Arthur Liberzon, Chet Birger, Helga Thorvaldsdóttir, Mahmoud Ghandi, Jill P Mesirov, and Pablo Tamayo. "The molecular signatures database hallmark gene set collection." In: *Cell Systems* 1.6 (2015).

[Lin10] Shili Lin. "Rank aggregation methods." In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.5 (2010).

[LTZ08] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. “Isolation forest.” In: *ICDM*. 2008.

[Liu+09] Tie-Yan Liu et al. “Learning to rank for information retrieval.” In: *Foundations and Trends in Information Retrieval* 3.3 (2009).

[Liu+15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. “Deep Learning Face Attributes in the Wild.” In: *ICCV*. 2015.

[Lu+18] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. “Learning under concept drift: a review.” In: *IEEE Transactions on Knowledge and Data Engineering* (2018).

[Lu+07] Yijuan Lu, Ira Cohen, Xiang Sean Zhou, and Qi Tian. “Feature selection using principal feature analysis.” In: *MM*. 2007.

[Luc+10] Roberto Lucchetti, Stefano Moretti, Fioravante Patrone, and Paola Radrizzani. “The Shapley and Banzhaf values in microarray games.” In: *Computers & Operations Research* 37.8 (2010).

[LL17] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions.” In: *Advances in Neural Information Processing Systems* 30 (2017).

[Mal+15] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. “Long Short Term Memory Networks for Anomaly Detection in Time Series.” In: *Esann*. 2015.

[Mat+18] Ravi Mathur, Daniel Rotroff, Jun Ma, Ali Shojaie, and Alison Motsinger-Reif. “Gene set analysis methods: a systematic comparison.” In: *BioData Mining* 11.1 (2018).

[Mit+22] Rory Mitchell, Joshua Cooper, Eibe Frank, and Geoffrey Holmes. “Sampling permutations for shapley value estimation.” In: *Journal of Machine Learning Research* 23.43 (2022).

[MJK15] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. 2015.

[Mor+08] Stefano Moretti, Danitsja van Leeuwen, Hans Gmuender, Stefano Bonassi, Joost Van Delft, Jos Kleinjans, Fioravante Patrone, and Domenico Franco Merlo. “Combining Shapley value and statistics to the analysis of gene expression data in children exposed to air pollution.” In: *BMC Bioinformatics* 9.1 (2008).

[MPB07] Stefano Moretti, Fioravante Patrone, and Stefano Bonassi. “The class of microarray games and the relevance index for genes.” In: *Top* 15 (2007).

[Mül+12] Emmanuel Müller, Fabian Keller, Sebastian Blanc, and Klemens Böhm. “OutRules: a framework for outlier descriptions in multiple context spaces.” In: *ECML PKDD*. 2012.

[Ngu+19] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. “GEE: A Gradient-based Explainable Variational Autoencoder for Network Anomaly Detection.” In: *CNS*. 2019.

[Nob09] William S Noble. “How does multiple testing correction work?” In: *Nature Biotechnology* 27.12 (2009).

[PK21] Cheong Hee Park and Jiil Kim. “An explainable outlier detection method using region-partition trees.” In: *The Journal of Supercomputing* 77 (2021).

[PV16] Ali Pesaranghader and Herna L Viktor. "Fast hoeffding drift detection method for evolving data streams." In: *ECML PKDD*. 2016.

[PVP18] Ali Pesaranghader, Herna L Viktor, and Eric Paquet. "McDiarmid drift detection methods for evolving data streams." In: *IJCNN*. 2018.

[Pfa+16] Karlson Pfannschmidt, Eyke Hüllermeier, Susanne Held, and Reto Neiger. "Evaluating tests in medical diagnosis: combining machine learning with game-theoretical concepts." In: *IPMU*. 2016.

[Pil+22] Vipin Pillai, Soroush Abbasi Koohpayegani, Ashley Ouligian, Dennis Fong, and Hamed Pirsiavash. "Consistent Explanations by Contrastive Learning." In: *CVPR*. 2022.

[Qah+15] Abdulhakim A Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. "A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams." In: *KDD*. 2015.

[RGL19] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. "Failing loudly: an empirical study of methods for detecting dataset shift." In: *Advances in Neural Information Processing Systems* (2019).

[Rei+16] Denis Moreira dos Reis, Peter Flach, Stan Matwin, and Gustavo Batista. "Fast unsupervised online drift detection using incremental kolmogorov-smirnov test." In: *KDD*. 2016.

[RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "" Why should I trust you?" Explaining the predictions of any classifier." In: *KDD*. 2016.

[Roz+22] Benedek Rozemberczki, Lauren Watson, Péter Bayer, Hao-Tsung Yang, Olivér Kiss, Sebastian Nilsson, and Rik Sarkar. "The shapley value in machine learning." In: *arXiv preprint arXiv:2202.05594* (2022).

[Ruf+18] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. "Deep One-Class Classification." In: *ICML*. 2018.

[Rus19] Chris Russell. "Efficient search for diverse coherent explanations." In: *FAccT*. 2019.

[Sal+21] Rohit Saluja, Avleen Kaur Malhi, Samanta Knapic, Kary Främling, and Cicek Cavdar. *Towards a Rigorous Evaluation of Explainability for Multivariate Time Series*. Tech. rep. 2021.

[San17] Miguel Oliveira Sandim. "Using stacked generalization for anomaly detection." MA thesis. Universidade do Porto, 2017.

[Sch+99] Robert E Schapire et al. "A brief introduction to boosting." In: *IJCAI*. 1999.

[Sch+19] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A Keim. "Towards a rigorous evaluation of XAI methods on time series." In: *ICCVW*. 2019.

[SMR08] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. 2008.

[Scu07] D Sculley. "Rank aggregation for similar items." In: *SDM*. 2007.

[Sha+53] Lloyd S Shapley et al. "A value for n-person games." In: (1953).

[She+17] Arvind Shekar, Tom Bocklisch, Patricia Sánchez, Christoph Straehle, and Emmanuel Müller. "Including Multi-feature Interactions and Redundancy for Feature Ranking in Mixed Datasets." In: *ECML PKDD*. 2017.

[SGK17] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. "Learning important features through propagating activation differences." In: *ICML*. 2017.

[Sil+19] Thiago Silveira, Min Zhang, Xiao Lin, Yiqun Liu, and Shaoping Ma. "How good your recommender system is? A survey on evaluations in recommendation." In: *International Journal of Machine Learning and Cybernetics* 10 (2019).

[Smi+17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. "Smoothgrad: removing noise by adding noise." In: *arXiv preprint arXiv:1706.03825* (2017).

[SCM20] Saúl Solorio-Fernández, J Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. "A review of unsupervised feature selection methods." In: *Artificial Intelligence Review* 53.2 (2020).

[Son+18] Huan Song, Deepta Rajan, Jayaraman Thiagarajan, and Andreas Spanias. "Attend and diagnose: Clinical time series analysis using attention models." In: *AAAI*. 2018.

[Sto+18] Ruth Alexandra Stoney, Jean-Marc Schwartz, David L Robertson, and Goran Nenadic. "Using set theory to reduce redundancy in pathway sets." In: *BMC Bioinformatics* 19.1 (2018).

[SK10] Erik Strumbelj and Igor Kononenko. "An efficient explanation of individual classifications using game theory." In: *The Journal of Machine Learning Research* 11 (2010).

[Sub+05] Aravind Subramanian, Pablo Tamayo, Vamsi K Mootha, Sayan Mukherjee, Benjamin L Ebert, Michael A Gillette, Amanda Paulovich, Scott L Pomeroy, Todd R Golub, Eric S Lander, et al. "Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles." In: *Proceedings of the National Academy of Sciences* 102.43 (2005).

[Sun+20] Min Woo Sun, Stefano Moretti, Kelley M Paskov, Nate T Stockham, Maya Varma, Brianna S Chrisman, Peter Y Washington, Jae-Yoon Jung, and Dennis P Wall. "Game theoretic centrality: a novel approach to prioritize disease candidate genes by combining biological networks with the Shapley value." In: *BMC Bioinformatics* 21.1 (2020).

[STY17] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." In: *ICML*. 2017.

[Sur+17] Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. "Clinical intervention prediction and understanding with deep neural networks." In: *Machine Learning for Healthcare Conference*. 2017.

[TI14] Toru Takahashi and Rui Ishiyama. "FIBAR: Fingerprint Imaging by Binary Angular Reflection for Individual Identification of Metal Parts." In: *EST*. 2014.

[Tak19a] Naoya Takeishi. "Shapley Values of Reconstruction Errors of PCA for Explaining Anomaly Detection." In: *ICDMW*. 2019.

[Tak19b] Naoya Takeishi. "Shapley values of reconstruction errors of pca for explaining anomaly detection." In: *ICDMW*. 2019.

[TC20] A Tallón-Ballesteros and C Chen. "Explainable AI: Using Shapley value to explain complex anomaly detection ML-based systems." In: *Machine learning and artificial intelligence* 332 (2020).

[TDV21] Yan-Martin Tamm, Rinchin Damdinov, and Alexey Vasilev. "Quality metrics in recommender systems: Do we calculate metrics consistently?" In: *RecSys*. 2021.

[Tri+17] Isaac Triguero et al. "KEEL 3.0: an Open Source Software for Multi-Stage Analysis in Data Mining." In: *International Journal of Computational Intelligence Systems* 10 (2017).

[TYR23] Che-Ping Tsai, Chih-Kuan Yeh, and Pradeep Ravikumar. "Faithshap: The faithful shapley interaction index." In: *Journal of Machine Learning Research* 24.94 (2023).

[Val+18] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. "On the robustness and discriminative power of information retrieval metrics for top-N recommendation." In: *RecSys*. 2018.

[Val+20] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. "Assessing ranking metrics in top-N recommendation." In: *Information Retrieval Journal* 23 (2020).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In: 2017.

[VE14a] Jorge Vergara and Pablo Estevez. "A Review of Feature Selection Methods Based on Mutual Information." In: *Neural Computing and Applications* (2014).

[VE14b] Jorge R Vergara and Pablo A Estévez. "A review of feature selection methods based on mutual information." In: *Neural Computing and Applications* 24 (2014).

[WTL15] Suhang Wang, Jiliang Tang, and Huan Liu. "Embedded unsupervised feature selection." In: *AAAI*. 2015.

[Xie+21] Zhuorui Xie, Allison Bailey, Maxim V Kuleshov, Daniel JB Clarke, John E Evangelista, Sherry L Jenkins, Alexander Lachmann, Megan L Wojciechowicz, Eryk Kropiwnicki, Kathleen M Jagodnik, et al. "Gene set knowledge discovery with Enrichr." In: *Current protocols* 1.3 (2021).

[Yan+11] Yi Yang, Heng Tao Shen, Zhigang Ma, Zi Huang, and Xiaofang Zhou. "$l_{2,1}$-norm regularized discriminative feature selection for unsupervised learning." In: *IJCAI*. 2011.

[YWP18] Shujian Yu, Xiaoyang Wang, and José C Príncipe. "Request-and-reverify: Hierarchical hypothesis testing for concept drift detection with expensive labels." In: *arXiv preprint arXiv:1806.10131* (2018).

[ZF14] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks." In: *ECCV*. 2014.

[Zha+19] Yujia Zhang, Kuangyan Song, Yiming Sun, Sarah Tan, and Madeleine Udell. “" Why Should You Trust My Explanation?" Understanding Uncertainty in LIME Explanations.” In: *arXiv preprint arXiv:1904.12991* (2019).

[ZK20] Di Zhao and Yun Sing Koh. “Feature Drift Detection in Evolving Data Streams.” In: *DEXA*. 2020.

[ZL07] Zheng Zhao and Huan Liu. “Spectral feature selection for supervised and unsupervised learning.” In: *ICML*. 2007.

[Zhe+19] Shihao Zheng, Simon B van der Zon, Mykola Pechenizkiy, Cassio P de Campos, Werner van Ipenburg, Hennie de Harder, and Rabobank Nederland. “Labelless concept drift detection and explanation.” In: *NeurIPS Workshop on Robust AI in Financial Services*. 2019.

[Zhu+19] Xiaoyan Zhu, Yu Wang, Yingbin Li, Yonghui Tan, Guangtao Wang, and Qinbao Song. “A new unsupervised feature selection algorithm using similarity-based feature clustering.” In: *Computational Intelligence* 35.1 (2019).

[ZS88] Matjaz Zwitter and Milan Soklic. *Breast Cancer*. UCI Machine Learning Repository. 1988.