Abstract Interpretation-Based Feature Importance for SVMs

Abhinandan Pal
IIIT Kalyani, India

Francesco Ranzato
University of Padova, Italy

Caterina Urban
INRIA & ENS | PSL, France

Marco Zanella
University of Padova, Italy

Abstract

We propose a symbolic representation for support vector machines (SVMs) by means of abstract interpretation, a well-known and successful technique for designing and implementing static program analyses. We leverage this abstraction in two ways: (1) to enhance the *interpretability* of SVMs by deriving a novel feature importance measure, called *abstract feature importance* (AFI), that does not depend in any way on a given dataset of the accuracy of the SVM and is very fast to compute, and (2) for verifying stability, notably *individual fairness*, of SVMs and producing concrete counterexamples when the verification fails. We implemented our approach and we empirically demonstrated its effectiveness on SVMs based on linear and non-linear (polynomial and radial basis function) kernels. Our experimental results show that, independently of the accuracy of the SVM, our AFI measure correlates much more strongly with the stability of the SVM to feature perturbations than feature importance measures widely available in machine learning software such as permutation feature importance. It thus gives better insight into the trustworthiness of SVMs.

1 Introduction

Machine learning (ML) software is increasingly being employed in high stakes or sensitive applications (Chouldechova, 2017; Khandani et al., 2010, etc.). This poses important challenges for safety, privacy, and non-discrimination (Buolamwini and Gebru, 2018; Obermeyer et al., 2019). As a consequence, research in ML verification rapidly gained popularity (Liu et al., 2021; Urban and Miné, 2021) and demand for interpretable ML models is more and more pronounced (Tjoa and Guan, 2021). Notably, interpretability is needed to meet the requirements of recent legal regulations on sensitive automated decision-making applications, such as the General Data Protection Regulation (GDPR) and the Artificial Intelligence Act in the EU.

There is a tradeoff between interpretability and performance of a ML model: non-linear models generally deliver much better predictions but they do not provide explanations. The most prominent interpretability technique is *feature importance*, measuring the contribution of each input feature to a model prediction (Bhatt et al., 2020).

Permutation feature importance (PFI) (Breiman, 2001; Fisher et al., 2019), the most widely used and understood importance measure, observes the decrease in predictive performance when a feature value is randomly shuffled: an increased loss is indicative of how much that feature is important for the predictive model. PFI is easy to explain, implement, and use, making it widely available in ML software (e.g., the scikit-learn Python library, etc.). On the other hand, the result may greatly vary depending on the dataset. Second, the result depends on shuffling and must be averaged across repetitions to stabilize, thus becoming resource intensive when the number of features is large. Third, PFI yields misleading results when features are correlated (Hooker et al., 2019). Finally, and more importantly, the quality of the result heavily depends on the accuracy of the model! Notably, model variance to feature perturbations (Goodfellow et al., 2018) and PFI are strongly correlated only when the model generalizes well.

Contributions. In this work, we propose a novel feature importance measure for SVMs, called *abstract feature importance* (AFI), that: (a) does *not* depend on a given dataset or the accuracy of the model, and (b) is extremely fast to compute, independently of the number of input features. We support both linear and non-linear kernels, in particular the polynomial and the radial basis function (RBF) kernels.

We derive our importance measure from a symbolic representation of a SVM based on *abstract interpretation* (Cousot and Cousot, 1977; Cousot, 2021), a well-established framework for designing computable and correct over-approximations of model computations. Specifically, the concrete quantities being manipulated by model computations are represented using an *abstract domain*, which defines their abstract counterparts and their data-structure representations, as well as algorithms to manipulate them according to the semantics of the computations.

We leverage existing abstract domains such as hyperrectangles (Cousot and Cousot, 1977) and reduced affine forms (Messine, 2002) that we combine with a novel abstract domain tailored for precisely representing computations with one-hot encoded categorical input features. We show the effectiveness of this combination in verifying model stability against feature perturbations. In particular, we focus on verifying *individual fairness* (Dwork et al., 2012).We evaluate our approach by verifying SVMs trained on the reference datasets in the literature on ML fairness (Mehrabi et al., 2021a) and considering different similarity relations.

Our approach is *sound*, meaning that an individually fair abstract representation of a SVM implies that the SVM is also fair. Thus, the fraction of successful fairness verifications over a test dataset is a *lower bound* on individual fairness of a SVM. On the other hand, our approach is *not complete* as there are cases in which the SVM is fair but the verification of its abstract representation fails due to imprecisions introduced by the abstraction. Our third contribution in this work is a way to leverage this abstract representation to generate concrete counterexamples when unable to verify fairness, i.e., concrete similar inputs to a SVM that result in different classifications. The fraction of successful counterexample searches over a test dataset yields a lower bound on how biased an SVM is and thus, by complement, an *upper bound* on individual fairness of a SVM.

Finally, we conduct an extensive experimental comparison between our proposed importance measure AFI and the standard and popular PFI and we show that AFI is better correlated with stability of a SVM model to feature perturbations *independently of the accuracy of the model*.

Related Work. Feature importance measures can be *local*, i.e., measuring feature importance for a specific prediction, or *global*, i.e., measuring importance over the entire input space of the ML model. We also distinguish *model-agnostic* measures, which can be applied to any model, and *model-specific* measures. Finally, we classify importance measures in *performance-based*, i.e., measuring importance with respect to the predictive performance of the model (requiring knowledge of the ground truth values), and *effect-based*, measuring importance based on the magnitude of change in the predicted outcome due to changes in the feature value (requiring no knowledge of the ground truth values). PFI (Breiman, 2001; Fisher et al., 2019) is a global, model-agnostic, performance-based measure. AFI, our novel feature importance measure, is *specific for SVMs* but can be used *both* as *global and local* measure, and is *effect-based*. Several other model-agnostic importance measures have been proposed in the literature. Prominent effect-based measures are visual tools such as partial dependence (PD) (Friedman, 2001), individual conditional expectation (ICE) (Goldstein et al., 2015), and accumulated local effects (ALE) (Apley and Zhu, 2020) plots. Other notable effect-based measures are Shapley values (Strumbelj and Kononenko, 2014), and local measures such as local interpretable model-agnostic explanations (LIME) (Ribeiro et al., 2016), and SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017). Visual tools, such as individual conditional importance (ICI) and partial importance (PI) curves (Casalicchio et al., 2018), are also proposed for local performance-based measures. Casalicchio et al. (2018) additionally propose a Shapley feature importance (SFIMP) measure that allows comparing feature importances across different models. Input gradient (Hechtlinger, 2016) is a local measure that can be both effect-based and performance-based. Feature importance measures specific for SVMs are typically limited to linear SVMs or face scalability issues with the number of features, e.g. (Mladenić et al., 2004; Chang and Lin, 2008). By contrast, our AFI measure also supports non-linear kernels and has no scalability issues.

Our work generally contributes to the research ecosystem around the verification of ML models using formal methods (Liu et al., 2021; Urban and Miné, 2021). Most approaches have focused on neural networks (Roh et al., 2020; Ruoss et al., 2020; Urban et al., 2020; Yurochkin et al., 2020, etc.) while here we focus on SVMs. Our work leverages the SVM verifier SAVer (Ranzato and Zanella, 2019). In addition, we introduce here a more precise abstraction for one-hot encoded features. Our fairness analysis is closely in line with the approach by Ranzato et al. (2021), who evaluated the individual fairness of decision tree ensembles trained by a new fairness-aware learning technique. Similar works either consider a very different notion of fairness or a different "threat model", in most cases both. Xiao et al. (2015) evaluate security against flipping a few labels to maximize classification error. Ghosh et al. (2022) consider group and causal fairness metrics. Langenberg et al. (2019) deal with robustness of SVMs against adversarial attacks. Fish et al. (2016) propose a new fairness metric where they add a new feature with random values and bias individuals on this feature: the model is fair when it recovers the original labels. Park et al. (2022) put forward a protocol to protect sensitive information and train a fair model using homomorphic encryption. Mehrabi et al. (2021b) and Verma and Rubin (2018) discuss several fairness metrics used to verify a variety of ML models.

2 Background

Support Vector Machines. SVMs (Cristianini and Shawe-Taylor, 2000) are machine learning models based on separation curves that partition the input vector space into regions that best fit binary classification labels $L = \{-1, +1\}$. Separation curves are computed by maximizing their distance (margin) from the closest vectors in the training dataset. The simplest SVM is linear, which in its primal form boils down to an hyperplane $\mathbf{w} \cdot \mathbf{z} - b = 0$, where $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ are learned parameters, that determines whether an input $\mathbf{z}$ falls above/below (i.e.,

$\operatorname{sgn}(\mathbf{w}\cdot\mathbf{z}-b)=\pm1$) w.r.t. the hyperplane. This approach is extended to non-linear SVMs through a projection to a high-dimensional space via a kernel function $k:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$. Given a training dataset $T=\{(\mathbf{x}_1,y_1),....,(\mathbf{x}_n,y_n)\}\subseteq X\times\{-1,+1\}$, kernel function k and learned parameters $c_i,b\in\mathbb{R}$, $i\in[1,n]$, a non-linear SVM C_T is represented in its dual form by the function $C_T(\mathbf{z})\triangleq\operatorname{sgn}\left(\sum_{i=1}^n(c_iy_ik(\mathbf{x}_i,\mathbf{z}))-b\right)$. Most common kernels are: (i) *linear*, where $k(\mathbf{x},\mathbf{z})=\mathbf{x}\cdot\mathbf{z}$; (ii) *polynomial*, where $k(\mathbf{x},\mathbf{z})=(\mathbf{x}\cdot\mathbf{z}+c)^p$, for some hyperparameters $c\in\mathbb{R}$ and $p\in\mathbb{N}$; (iii) *radial basis function* (RBF), where $k(\mathbf{x},\mathbf{z})=e^{-\gamma\|\mathbf{x}-\mathbf{z}\|_2^2}$, for some positive hyperparameter $\gamma>0$. In multi-classification for a set of labels $L=\{y_1,...,y_m\}$, $m>2$, the standard approach is a reduction into multiple binary classification problems combined by leveraging a voting procedure over different labels.

Abstract Interpretation. A tuple $\langle A,\sqsubseteq^A,\gamma^A\rangle$ is a *numerical abstract domain* (or abstraction) when $\langle A,\sqsubseteq^A\rangle$ is an partially ordered set of abstract values and $\gamma^A:A\to\wp(\mathbb{R}^n)$ is a concretization function which maps abstract values to sets of numerical vectors and monotonically preserves the ordering relation, i.e., $a_1\sqsubseteq^A a_2$ implies $\gamma^A(a_1)\subseteq\gamma^A(a_2)$. Intuitively, an abstract domain defines a symbolic representation of sets of vectors in $\wp(\mathbb{R}^n)$.

Given a k-ary operation $f:(\mathbb{R}^d)^k\to\mathbb{R}^d$, for some $k>0$, a corresponding abstract function $f^A:A^k\to A$ is a *sound* (over-)approximation of f on $(a_1,...,a_k)\in A^k$ when $\{f(\mathbf{x}_1,...,\mathbf{x}_k)\mid\mathbf{x}_i\in\gamma^A(a_i)\}\subseteq\gamma^A(f^A(a_1,...,a_k))$ holds. Morever, f^A is a *complete* (over-)approximation of f on its input $(a_1,...,a_k)$ when equality holds.

Abstract Domains. We consider the well-known abstract domain of *hyperrectangles* (or *intervals*) (Cousot, 2021; Rival and Yi, 2020). The hyperrectangle abstract domain HR_n consists of n-dimensional vectors h of real intervals $h=\big([l_1,u_1],\dots,[l_n,u_n]\big)\in\mathrm{HR}_n$, with lower and upper bounds $l_i,u_i\in\mathbb{R}\cup\{-\infty,+\infty\}$ such that $l_i\le u_i$. Hence, the concretization function $\gamma^{\mathrm{HR}}:\mathrm{HR}_n\to\wp(\mathbb{R}^n)$ is defined by $\gamma^{\mathrm{HR}}(h)\triangleq\{\mathbf{x}\in\mathbb{R}^n\mid\forall i.\,l_i\le\mathbf{x}_i\le u_i\}$. Abstract operations are defined by extending the following abstract additions and multiplications of intervals: $[l_1,u_1]+^{\mathrm{HR}}[l_2,u_2]\triangleq[l_1+l_2,u_1+u_2]$ and $[l_1,u_1]*^{\mathrm{HR}}[l_2,u_2]\triangleq[\min(l_1l_2,l_1u_2,l_2u_1,l_2u_2),\max(l_1l_2,l_1u_2,l_2u_1,l_2u_2)]$.

It is known that a compositional abstract evaluation on HR of an expression exp can be imprecise, e.g., the evaluations of the simple expressions $x-x$ and $x\cdot x$ on an input interval $[-c,c]$, with $c>0$, yield, respectively, $[-2c,2c]$ and $[-c^2,c^2]$, rather than the exact intervals $[0,0]$ and $[0,c^2]$. This dependency problem can trigger a significant source of imprecision for the hyperrectangle abstraction of a polynomial/RBF SVM classifier. Therefore, following Ranzato and Zanella (2019), for our SVM abstract representations, we leverage the relational reduced affine form (RAF) abstraction. A RAF for vectors in $\mathbb{R}^n$ is given by an expression $a_0+\sum_{i=1}^n a_i\epsilon_i+a_r\epsilon_r$, where the ϵ_i's are symbolic variables ranging in the real interval $[-1,1]$ representing a dependence from the i-th component of the vector, while ϵ_r is a further symbolic variable in $[-1,1]$ which accumulates all the approximations introduced by non-linear operations. Thus, $\mathrm{RAF}_n\triangleq\{a_0+\sum_{i=1}^n a_i\epsilon_i+a_r\epsilon_a\mid a_0,a_1,...,a_n\in\mathbb{R},\,a_r\in\mathbb{R}_{\ge0}\}$. The concretization map $\gamma^{\mathrm{RAF}}:\mathrm{RAF}_n\to\wp(\mathbb{R})$ is defined by $\gamma^{\mathrm{RAF}}(a_0+\sum_{i=1}^n a_i\epsilon_i+a_r\epsilon_a)\triangleq\{x\in\mathbb{R}\mid a_0-\sum_{i=1}^n|a_i|-|a_r|\le x\le a_0+\sum_{i=1}^n|a_i|+|a_r|\}$. Moreover, RAF_n also has a top element $\top^{\mathrm{RAF}}$ representing the lack of information, i.e., such that $\gamma^{\mathrm{RAF}}(\top^{\mathrm{RAF}})=\mathbb{R}$. Linear operations, namely additions and scalar multiplications, admit a complete approximation on the RAF abstraction. Thus, RAFs settle the dependency problem for linear expressions. Instead, non-linear abstract operations, such as multiplication, must necessarily be approximated for RAFs. We will use an optimal abstract multiplication of RAFs defined by Skalna and Hladík (2017).

Robustness. We consider an input space $X\subseteq\mathbb{R}^d$, a set of classification labels $L=\{y_1,...,y_m\}$, and a dataset $T=\{(\mathbf{x}_1,y_1),....,(\mathbf{x}_n,y_n)\}\subseteq X\times L$. A classifier trained on the dataset T is modeled as a map $C_T:X\to L$.

An adversarial region for an input sample $\mathbf{x}\in X$ is designated by a perturbation $P(\mathbf{x})\subseteq X$ such that $\mathbf{x}\in P(\mathbf{x})$. Usually, a perturbation function $P:X\to\wp(X)$ is defined through a metric m to measure similarity between inputs as their distance w.r.t. m. The most common metric is induced by the ℓ_∞ maximum norm (Carlini and Wagner, 2017) defined as $\|\mathbf{x}\|_\infty\triangleq\max\{\mathbf{x}_1,\dots,\mathbf{x}_d\}$, so that the corresponding perturbation $P_\infty(\mathbf{x})$ includes all the vectors $\mathbf{z}\in X$ whose ℓ_∞ distance from $\mathbf{x}$ is bounded by a threshold $\epsilon\in\mathbb{R}^+$, that is, $P_\infty(\mathbf{x})\triangleq\{\mathbf{z}\in X\mid\|\mathbf{x}-\mathbf{z}\|_\infty\le\epsilon\}$.

A classifier C is robust (or stable) for a perturbation function P on an input $\mathbf{x}$, denoted by $\mathrm{robust}(C,\mathbf{x},P)$, when for all $\mathbf{z}\in P(\mathbf{x})$, $C(\mathbf{z})=C(\mathbf{x})$ holds. Robustness to a perturbation P is used as a major metric (Goodfellow et al., 2018) to assess a classifier C on a testing set $T\subseteq X\times L$ as follows: $\mathrm{rob}_T(C,P)\triangleq\frac{|\{(\mathbf{x},y)\in T\;\mid\;\mathrm{robust}(C,\mathbf{x},P)\}|}{|T|}$.

SAVer. Our work leverages SAVer (SVM Abstract Verifier), an automatic tool for robustness verification of SVMs introduced by Ranzato and Zanella (2019). Given an SVM $C:X\to L$, SAVer leverages an abstraction A_n of $\wp(\mathbb{R}^n)$ to first achieve a sound abstraction $P^\sharp(\mathbf{x})\in A_n$ of an adversarial region $P(\mathbf{x})$, i.e., $P(\mathbf{x})\subseteq\gamma^A(P^\sharp(\mathbf{x}))$, and then applies sound abstract versions of the transfer functions occurring in C to design an abstract SVM $C^\sharp:A\to\wp(L)$ that computes an over-approximation of the labels assigned to inputs in $P(\mathbf{x})$, i.e., $\{C(\mathbf{z})\in L\mid\mathbf{z}\in P(\mathbf{x})\}\subseteq C^\sharp(P^\sharp(\mathbf{x}))$. If $C^\sharp(P^\sharp(\mathbf{x}))=\{y_i\}$, then every input in $P(\mathbf{x})$ is classified as y_i, so C is proved robust over $P(\mathbf{x})$. In the binary case $L=\{-1,+1\}$,

$C^\sharp$ consists of an abstract function $\mathcal{A}^\sharp_C : A_n \to A_1$ that computes an over-approximation $\mathcal{A}^\sharp_C(P^\sharp(\mathbf{x}))$ of the set of distances between samples in $P^\sharp(\mathbf{x})$ and the separation curve, and then over-approximates the set of labels: $C^\sharp(P^\sharp(\mathbf{x})) \triangleq$ **if** $\gamma^{A_1}(\mathcal{A}^\sharp_C(P^\sharp(\mathbf{x}))) \subseteq \mathbb{R}_{<0}$ **then** $\{-1\}$ **elseif** $\gamma^{A_1}(\mathcal{A}^\sharp_C(P^\sharp(\mathbf{x}))) \subseteq \mathbb{R}_{>0}$ **then** $\{+1\}$ **else** $\{-1,+1\}$. In multi-classification, the voting also needs to be soundly approximated (Ranzato and Zanella, 2019).

3 Abstract Feature Importance

We can now define our *abstract feature importance* (AFI).

Definition 3.1 (Abstract Feature Importance). Let $C : \mathbb{R}^n \to L$ be a SVM classifier and let $\mathcal{A}^{\mathrm{RAF}}_C : (\mathrm{RAF}_n)^n \to \mathrm{RAF}_n$ be its abstraction in the RAF abstract domain. Let $\mathcal{A}^{\mathrm{RAF}}_C(f_1,\ldots,f_n) \triangleq a_0 + \sum_{i=1}^n a_i\epsilon_i + a_r\epsilon_r$ be the abstract computation output for an abstract input $(f_1,\ldots,f_n)$, $f_i \in \mathrm{RAF}_n$. The importance of every input feature $i \in [1,n]$ is defined as the absolute value $|a_i| \geq 0$. □

The definition purposely approximates by ignoring the accumulative error due to the approximations of all non-linear operations performed by C, i.e., the term $a_r\epsilon_r$, influenced by all input features. When $(f_1,\ldots,f_n)$, $f_i \in \mathrm{RAF}_n$, abstracts the whole input space $X \subseteq \mathbb{R}^n$, AFI measures the *global* feature importance. Otherwise, AFI measures the *local* importance on the output label.

Example 3.2. Let us consider a toy linear SVM C over a space $X \subseteq \mathbb{R}^2$ of values normalized to $[-1,+1]$, thus $X = \{\mathbf{x} \in \mathbb{R}^2 \mid -1 \leq x_1, x_2 \leq +1\}$. We consider two support vectors $\mathbf{v}_1 = (-0.5, 1), \mathbf{v}_2 = (0.5, -1) \in X$ labeled respectively as $-1, +1$ with weights $w_1 = w_2 = 0.5$ and bias $b = 0$, so that $C^{\mathrm{sgn}}(\mathbf{x}) = -0.5(\mathbf{v}_1 \cdot \mathbf{x}) + 0.5(\mathbf{v}_2 \cdot \mathbf{x})$. We express X as the RAF $a = (0 \pm \epsilon_1, 0 \pm \epsilon_2) \in (\mathrm{RAF}_2)^2$. By performing the abstract computations of $\mathcal{A}^{\mathrm{RAF}}_C$ on this input we obtain: $\mathcal{A}^{\mathrm{RAF}}_C(a) = -0.5(\mathbf{v}_1 \cdot^{\mathrm{RAF}} a) + 0.5(\mathbf{v}_2 \cdot^{\mathrm{RAF}} a) = -0.5(-0.5(0 \pm \epsilon_1) + 1(0 \pm \epsilon_2)) + 0.5(0.5(0 \pm \epsilon_1) - 1(0 \pm \epsilon_2)) = -0.5(0 \pm (-0.5)\epsilon_1 \pm \epsilon_2) + 0.5(0 \pm 0.5\epsilon_1 \pm (-1)\epsilon_2) = 0 \pm 0.5\epsilon_1 \pm (-1)\epsilon_2$. We therefore infer the importance indices $|a_1| = 0.5$ and $|a_2| = 1$ for, respectively, x_1 and x_2, and conclude that x_2 is twice as important as x_1. Note that, since we considered a linear SVM, it can be rewritten in primal form as: $C^{\mathrm{sgn}}(\mathbf{x}) = -0.5(\mathbf{v_1} \cdot \mathbf{x}) + 0.5(\mathbf{v_2} \cdot \mathbf{x}) = -0.5(-0.5x_1 + x_2) + 0.5(0.5x_1 - x_2) = 0.5x_1 - x_2 = (0.5, -1) \cdot \mathbf{x}$, thus obtaining an explicit weight $\mathbf{w} = (0.5, -1)$ for the input features, whose absolute values $0.5, 1$ exactly match our importance indices. □

Our importance indices depend on the size of the input, which can make results harder to read and interpret, especially when the number of features is high. We use a simple clustering strategy to assign them a score: we consider the distribution of feature importances and compute its mean μ and standard deviation σ, and we assign to each feature x_i the score $score_i \in \mathbb{Z}$ such that $\mu + score_i\sigma \leq \frac{a_i - \mu}{\sigma} < \mu + (score_i + 1)\sigma$, which has the same effect of standardizing the distribution into a normal $\mathcal{N}(0,1)$ and slicing the distribution at every unit, labeling every slice with a progressive number. By doing so, features moderately influencing the result will have a score close to zero, while relevant feature will have higher scores, and those not influencing the outcome will have a negative score. Last, we suggest to shift and clip such distribution in order to obtain grades, e.g., in $[3,10]$, which can be achieved by a simple transformation: $grade_i = \max(\min(10, score_i + 6), 3)$. For instance, let us consider a distribution of indices a_i given by $(1,6,2,5,6,1,6,7,8,9)$, where $\mu = 5.1$ and $\sigma = 2.85$: we compute $gradei_i$ as $(5,7,5,6,7,5,7,7,8,8)$. In this case it becomes easy to see that x_1 and x_3 have similar impact on the classification, although having different scores.

Last, we emphasize the fact that AFI does not require any knowledge on the ground truth values, nor the actual output of the classifier, as it focuses on the computation process performed by the classifier, rather than how the result of such computation is used to assign a label to a point, thus making this approach feasible in scenarios where the correct output is not known in advance.

4 An Abstraction for One-Hot Encoding

ML algorithms need a way to represent categorical data in numeric form. Let $F = \{c_1, c_2, \ldots, c_k\}$ be the set of values of a categorical feature f. Assigning a number to each value in F introduces an unwanted ordering relation among features. A better approach is *one-hot encoding*, that is, replacing f with k binary features $(x^f_1, x^f_2, \ldots, x^f_k) \in \{0,1\}^k$ such that $\forall i \in [1,k].\, x^f_i = 1 \Leftrightarrow f = c_i$. This sequence of bits is also referred to as a *tier* of f.

Abstractions such as the hyperrectangle and RAF abstract domains, are likely to suffer from a significant loss of precision when dealing with one-hot encoded features, as they are not able to keep track of the relationship existing between the binary features resulting from the encoding.

Example 4.1. Let us consider a categorical feature $f \in F = \{\mathit{red}, \mathit{green}, \mathit{blue}\}$ and let $(x_r, x_g, x_b) \in \{0,1\}^3$ be the corresponding one-hot encoded tiers. Consider the set $\{\mathit{red}, \mathit{green}\}$, represented by the set of tiers $X = \{(1,0,0),(0,1,0)\}$. The most precise hyperrectangle abstraction of X is $h = (x_r \in [0,1], x_g \in [0,1], x_b \in [0,0]) \in \mathrm{HR}_3$. Observe that h also represents infinitely many vectors in $\mathbb{R}^3$ that do not belong to X and are illegal encodings, such as $(0.3, 0.7, 0)$, $(1,1,0)$ or $(0,0,0)$. □

To hinder this loss of precision, we define the *One-Hot abstraction* OH, a novel family of numerical abstractions tailored for one-hot encoded values.

Let us first recall the *constant propagation* abstract domain $\mathrm{CP} \triangleq \mathbb{R} \cup \{\bot^{\mathrm{CP}}, \top^{\mathrm{CP}}\}$, ordinarily used by modern compilers (Aho et al., 2006; Wegman and Zadeck, 1991). CP is a flat domain whose partial order $\sqsubseteq^{\mathrm{CP}}$ is defined by $\bot^{\mathrm{CP}} \sqsubseteq^{\mathrm{CP}} z \sqsubseteq^{\mathrm{CP}} \top^{\mathrm{CP}}$, for all $z \in \mathbb{R}$. The concretization $\gamma^{\mathrm{CP}} : \mathrm{CP} \to \wp(\mathbb{R})$ is: $\gamma^{\mathrm{CP}}(z) \triangleq \{z\}$, for all $z \in \mathbb{R}$, meaning that a given numerical feature can only assume a constant value z, $\gamma^{\mathrm{CP}}(\top^{\mathrm{CP}}) \triangleq \mathbb{R}$ representing no constancy information, while $\gamma^{\mathrm{CP}}(\bot^{\mathrm{CP}}) \triangleq \varnothing$ encodes unfeasibility. CP also has an abstraction map $\alpha^{\mathrm{CP}} : \wp(\mathbb{R}) \to \mathrm{CP}$ that provides the best approximation in CP, i.e. least w.r.t. the order $\sqsubseteq^{\mathrm{CP}}$, of a set of values, which is: $\alpha^{\mathrm{CP}}(\varnothing) \triangleq \bot^{\mathrm{CP}}$, $\alpha^{\mathrm{CP}}(\{z\}) \triangleq z$, and $\alpha^{\mathrm{CP}}(X) \triangleq \top^{\mathrm{CP}}$ otherwise.

The One-Hot abstract domain for a k-dimensional one-hot encoded feature space is: $\mathrm{OH}_k \triangleq (\mathrm{CP} \times \mathrm{CP})^k$. Thus, abstract values are k-tuples of pairs of values in CP, that keep track of the numerical information originated from a single one-hot k-encoded feature, both when this was originally false, i.e. 0, or true, i.e. 1. Given $a \in \mathrm{OH}_k$ and a component $i \in [1,k]$, let $a_{i,f/t} \in \mathrm{CP}$ denote, resp., the first/second element of the i-th pair in a. The partial order $\sqsubseteq$ of OH_k is induced componentwise by $\sqsubseteq^{\mathrm{CP}}$, i.e., for all $a, b \in \mathrm{OH}_k$, $a \sqsubseteq b \Leftrightarrow \forall i \in [1,k].\, a_{i,f} \sqsubseteq^{\mathrm{CP}} b_{i,f} \,\&\, a_{i,t} \sqsubseteq^{\mathrm{CP}} b_{i,t}$. Then, for each component $i \in [1,k]$, the map $\hat{\gamma}_i : \mathrm{OH}_k \to \wp(\mathbb{R}^k)$ is defined as: $\hat{\gamma}_i(a) \triangleq \{\mathbf{x} \in \mathbb{R}^k \mid \mathbf{x}_i \in \gamma^{\mathrm{CP}}(a_{i,t}),\, \forall j \neq i.\, \mathbf{x}_j \in \gamma^{\mathrm{CP}}(a_{j,f})\}$. Thus, $a \in \mathrm{OH}_k$ represents through $\hat{\gamma}_i$ the set of tiers whose i-th component was originally set to true. Note that if, for some $i \in [1,k]$, either $a_{i,f} = \bot^{\mathrm{CP}}$ or $a_{i,t} = \bot^{\mathrm{CP}}$, then $\hat{\gamma}_i(a) = \varnothing$. A value $a \in \mathrm{OH}_k$ such that, for all $i \in [1,k]$ and $u \in \{f,t\}$, $a_{i,u} \in \mathrm{CP} \smallsetminus \{\top^{\mathrm{CP}}\}$ is called *top-less*. To retrieve all the concrete vectors represented by a, we collect all the vectors obtained by assuming that any component of the tier was originally set to true, namely, the concretization map $\gamma^{\mathrm{OH}_k} : \mathrm{OH}_k \to \wp(\mathbb{R}^k)$ is: $\gamma^{\mathrm{OH}_k}(a) \triangleq \cup_{i=1}^{k} \hat{\gamma}_i(a)$.

Example 4.2. Let us continue Example 4.1 by considering $a = \big((0,1),(0,1),(0,\bot^{\mathrm{CP}})\big) \in \mathrm{OH}_3$ as an abstraction of $X = \{(1,0,0),(0,1,0)\}$. Its concretization is: $\gamma^{\mathrm{OH}_3}(a) = \hat{\gamma}_1(a) \cup \hat{\gamma}_2(a) \cup \hat{\gamma}_3(a) = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{1\}, \mathbf{x}_2 \in \{0\}, \mathbf{x}_3 \in \{0\}\} \cup \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{0\}, \mathbf{x}_2 \in \{1\}, \mathbf{x}_3 \in \{0\}\} \cup \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{0\}, \mathbf{x}_2 \in \{0\}, \mathbf{x}_3 \in \varnothing\} = \{(1,0,0),(0,1,0)\}$. Thus a precisely represents X. $\square$

Example 4.2 is not fortuitous. In fact, for any set X of one-hot encoded tiers there always exists an abstract value a in OH which precisely represents this set, i.e., $\gamma^{\mathrm{OH}}(a) = X$.

Theorem 4.3. *If $X \in \wp(\mathbb{R}^k)$ is such that every vector of X is a one-hot encoded tier $(0,...,0,1,0,...0)$, then the abstract value $a^X \in \mathrm{OH}_k$ defined as $a_i^X \triangleq \big(\alpha^{\mathrm{CP}}(\{\mathbf{x}_i \mid \mathbf{x} \in X, \mathbf{x}_i = 0\}), \alpha^{\mathrm{CP}}(\{\mathbf{x}_i \mid \mathbf{x} \in X, \mathbf{x}_i = 1\})\big)$, for all $i \in [1,k]$, precisely represents X.*

Remark 4.4. a^X is always top-less because the components of each pair a_i^X range in $\{0, 1, \bot^{\mathrm{CP}}\}$. $\square$

Given a function $f : \mathbb{R} \to \mathbb{R}$, its abstract counterpart $f^{\mathrm{CP}} : \mathrm{CP} \to \mathrm{CP}$ on the CP domain is (Wegman and Zadeck, 1991): $f^{\mathrm{CP}}(z) \triangleq f(z)$, for all $z \in \mathbb{R}$, $f^{\mathrm{CP}}(\bot^{\mathrm{CP}}) \triangleq \bot^{\mathrm{CP}}$, and $f^{\mathrm{CP}}(\top^{\mathrm{CP}}) \triangleq \top^{\mathrm{CP}}$. In turn, f^{CP} allows us to define a sound abstract counterpart of f on our OH abstraction:

Theorem 4.5. *A sound approximation of f on OH_k is $f^{\mathrm{OH}} : \mathrm{OH}_k \to \mathrm{OH}_k$ defined, for all $i \in [1,k]$, as $(f^{\mathrm{OH}}(a))_i \triangleq \big(f^{\mathrm{CP}}(a_{i,f}), f^{\mathrm{CP}}(a_{i,t})\big)$.*

Example 4.6. Let us carry on Example 4.2. We apply $f(x) \triangleq x^2 - 3x + 1$ to every component in $\gamma^{\mathrm{OH}_3}(a)$: $f(\gamma^{\mathrm{OH}_3}(a)) = \{(f(1), f(0), f(0)), (f(0), f(1), f(0))\} = \{(-1,1,1),(1,-1,1)\}$. By applying Theorem 4.5 to a, we obtain $a' \triangleq f^{\mathrm{OH}}(a) = \big((f^{\mathrm{CP}}(0), f^{\mathrm{CP}}(1)), (f^{\mathrm{CP}}(0), f^{\mathrm{CP}}(1)), (f^{\mathrm{CP}}(0), f^{\mathrm{CP}}(\bot^{\mathrm{CP}})\big) = \big((1,-1), (1,-1), (1,\bot^{\mathrm{CP}})\big)$, whose concretization is: $\gamma^{\mathrm{OH}_3}(a') = \hat{\gamma}_1(a') \cup \hat{\gamma}_2(a') \cup \hat{\gamma}_3(a') = \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{-1\}, \mathbf{x}_2 \in \{1\}, \mathbf{x}_3 \in \{1\}\} \cup \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{1\}, \mathbf{x}_2 \in \{-1\}, \mathbf{x}_3 \in \{1\}\} \cup \{\mathbf{x} \in \mathbb{R}^3 \mid \mathbf{x}_1 \in \{1\}, \mathbf{x}_2 \in \{1\}, \mathbf{x}_3 \in \varnothing\} = \{(-1,1,1),(1,-1,1)\}$. Then, notice that soundness holds because $f(\gamma^{\mathrm{OH}_3}(a)) \subseteq \gamma^{\mathrm{OH}_3}(f^{\mathrm{OH}}(a))$. $\square$

Note that in Example 4.6, $f(\gamma^{\mathrm{OH}_3}(a)) = \gamma^{\mathrm{OH}_3}(f^{\mathrm{OH}}(a))$ also holds, i.e., f^{OH} is a complete approximation of f on a. This is as a consequence of the following general result.

Corollary 4.7. *Let $a \in \mathrm{OH}_k$ be top-less. Then (i) f^{OH} is a complete abstraction of f on a; and (ii) given $f_1, f_2, \ldots, f_p : \mathbb{R} \to \mathbb{R}$, $f_1^{\mathrm{OH}} \circ f_2^{\mathrm{OH}} \circ \cdots \circ f_p^{\mathrm{OH}}$ is a complete abstraction of $f_1 \circ f_2 \circ \cdots \circ f_p$ on a.*

We implemented this OH abstraction on top of (the interval and) RAF abstraction in SAVer. Given a categorical feature f we first perform one-hot encoding, obtaining $(x_1^f, x_2^f, \ldots, x_k^f) \in \{0,1\}^k$. When perturbing one-hot encoded values we allow every binary feature in the encoding to be either 0 or 1, so their abstract value is always in the shape $(0^{CP}, 1^{CP})^k$. As a consequence each abstract binary feature a_i can be represented as the RAF $0.5 \pm 0.5\epsilon_i$. We keep track of the relation between features and tiers using a global lookup table. After computing the abstract kernel, the resulting RAF a contains information both from the regular and the OH analysis. For verification purposes, we condense the latter to a single interval representing the behavior of the categorical feature f: we compute γ^{OH_k} over each tier and we select the values minimizing and maximizing the RAF expression a, thus computing a sound approximation of the interval of variation induced by f. Then, we build a RAF a' by transferring numerical features as they are, and replacing tier information with the newly-computed intervals, expressed in RAF form. Finally, the resulting a' allows to compute the superset of output labels as before. Both the RAF a or the condensed RAF a' are suitable for our importance measure AFI: in the former case, unlike PFI, we are additionally able to measure the importance of tiers of a one-hot encoded feature.

5 Individual Fairness

Several formal models of fairness have been investigated in the literature. Dwork et al. (2012) point out several weaknesses of group fairness and therefore study *individual fairness* defined as "the principle that two individuals who are similar with respect to a particular task should be classified similarly" (Dwork et al., 2012, Section 1.1). This is formalized as a Lipschitz condition of the classifier, that is, by requiring that two individuals $\mathbf{x}, \mathbf{y} \in X$ whose distance is $\delta(\mathbf{x}, \mathbf{y})$, are mapped, respectively, to distributions $D_\mathbf{x}$ and $D_\mathbf{y}$ whose distance is at most $\delta(\mathbf{x}, \mathbf{y})$. Intuitively, the output distributions for $\mathbf{x}$ and $\mathbf{y}$ are indistinguishable up to their distance. Several distance metrics $\delta\colon X \times X \to \mathbb{R}_{\geq 0}$ can be used in this context, where (Dwork et al., 2012, Section 2) studies the total variation or relative ℓ_∞ distances.

Following Dwork et al. (2012), a classifier $C\colon X \to L$ is (individually) *fair* when C outputs the same label for all pairs of individuals $\mathbf{x}, \mathbf{y} \in X$ satisfying a similarity relation $S \subseteq X \times X$ between input samples. This relation S can be derived from a distance δ as follows: $(\mathbf{x}, \mathbf{y}) \in S \Leftrightarrow \delta(\mathbf{x}, \mathbf{y}) \leq \epsilon$, where $\epsilon \in \mathbb{R}$ is a similarity threshold.

Definition 5.1 (Individual Fairness). A classifier $C\colon X \to L$ is *fair* on an individual $\mathbf{x} \in X$ with respect to a similarity relation $S \subseteq X \times X$, denoted by $\mathrm{fair}(C, \mathbf{x}, S)$, when $\forall \mathbf{z} \in X.\, (\mathbf{x}, \mathbf{z}) \in S \Rightarrow C(\mathbf{z}) = C(\mathbf{x})$. $\square$

To define a fairness metric for a classifier C, we compute how often C is fair on sets of similar individuals in a test set $T \subseteq X \times L$: $\mathrm{fair}_{T,S}(C) \triangleq \frac{|\{(\mathbf{x},y) \in T \mid \mathrm{fair}(C,\mathbf{x},S)\}|}{|T|}$. Hence, individual fairness for a similarity relation S boils down to robustness on the perturbation $P_S(\mathbf{x}) \triangleq \{\mathbf{z} \in X \mid (\mathbf{x}, \mathbf{z}) \in S\}$ induced by S, i.e., for all $\mathbf{x} \in X$, $\mathrm{fair}(C, \mathbf{x}, S) \Leftrightarrow \mathrm{robust}(C, \mathbf{x}, P_S)$.

6 Mitigating Incompleteness

The abstract framework described in Sec. 2 and 4 is *sound*, thus a classifier C verified as *robust* over a region $P(\mathbf{x})$ guarantees that every point $\mathbf{x}' \in P(\mathbf{x})$ receives the same label. The converse is generally not true for non-linear kernels, due to lack of *completeness*: when the abstract verification is not able to assert robustness, it may be either due to a loss of precision or an actual point in $P(\mathbf{x})$ which receives a different label. We refer to the latter as a *counterexample*. In case of an inconclusive analysis we can mitigate the effect of incompleteness by searching for counterexamples: if at least one is found the classifier can be marked as *not robust*. Finding a counterexample within a possibly infinite set of points however is a daunting task. Let $a \in \mathrm{RAF}_n$ be a sound abstraction for $P(\mathbf{x})$, C a classifier, and $\mathcal{A}_C^{\mathrm{RAF}}$ its abstraction. We define an informed heuristic search approach which leverages our AFI measure:

1. let $a_{out} = a_0 + \sum_{i=1}^{n} a_i \epsilon_i + a_r \epsilon_r$ be the output of the abstract computation on RAF (cf. Sec. 3);
2. if $C(\mathbf{x}) < 0$, we look for a potential counterexample $\mathbf{x}^*$ by maximizing a_{out}, i.e., selecting the maximum possible value for every x_i when $a_i > 0$, and the minimum when $a_i < 0$ (the converse if $C(\mathbf{x}) > 0$);
3. if $C(\mathbf{x}^*) \neq C(\mathbf{x})$, then $\mathbf{x}^*$ is a counterexample for $\mathbf{x}$, and the classifier is not robust;
4. otherwise we select the most influential feature x_M, and its mean value in $P(\mathbf{x})$ given by $m = \frac{\min\{x_M \mid \mathbf{x} \in P(\mathbf{x})\} + \max\{x_M \mid \mathbf{x} \in P(\mathbf{x})\}}{2}$, and we partition $P(\mathbf{x})$ using the cutting hyperplane $x_M \leq m$, obtaining left and right sets $P_l(\mathbf{x}), P_r(\mathbf{x}) \subseteq P(\mathbf{x})$;
5. we consider $a_l, a_r \in \mathrm{RAF}$ abstracting $P_l(\mathbf{x}), P_r(\mathbf{x})$, respectively, and we recursively repeat from step 1 until a counterexample is found, or a user-defined timeout is met.

Maximization in step 2 requires additional care for features obtained through one-hot encoding, as exactly one must be set to 1: we set to 1 the most influential feature only. We also observe that computing a_l, a_r during step 5 does not introduce any loss of precision, as RAF represent hyperrectangles, and partitioning one using a cutting hyperplane of the form $x_i \leq k$ yields two smaller hyperrectangles.

Step 1 and 2 correspond to looking for a counterexample in the vertices of the hyperrectangle represented by a, which have the greatest distance from the center and are therefore intuitively more likely to exhibit different labels. Since a hyperrectangle has 2^n vertices it is not feasible to check them all, and we thus use our feature importance analysis to infer a gradient pointing towards the most promising one. If no counterexample is found, it may be due to the separation curve of C crossing $P(\mathbf{x})$ while leaving all the vertices on the same side. We therefore proceed to step 4 and 5 partitioning $P(\mathbf{x})$ into two smaller components $P_l(\mathbf{x})$ and $P_r(\mathbf{x})$ by cutting the former space in half along the axis of the most influential feature, and recursively repeating the process on the two components. Should a counterexample be found, C can be definitively marked as not robust. Otherwise, we set a timeout mechanism, such as a limit on the recursion depth, to avoid non-termination.

Example 6.1. Let $\mathbf{s} = (0, -\sqrt{2}), \mathbf{t} = (-1, 1), \mathbf{v} = (1, 1)$ be the support vectors of an SVM with polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^2$, $-\alpha_\mathbf{s} = \alpha_\mathbf{t} = \alpha_\mathbf{v} = 1$ their weights, and $b = 0$ the bias. Classifier C is therefore given by $C(\mathbf{x}) = -(\mathbf{s} \cdot \mathbf{x} + 1)^2 + (\mathbf{t} \cdot \mathbf{x} + 1)^2 + (\mathbf{v} \cdot \mathbf{x} + 1)^2$ and can be rewritten to its primal form $2x_1^2 + 2(2 + \sqrt{2})x_2 + 1$, allowing to see the separation plane as the parabola $\Gamma : x_2 = -\frac{1}{2+\sqrt{2}} x_1^2 - \frac{1}{2(2+\sqrt{2})}$. We now consider $\mathbf{x}' = (0.5, -0.5)$ and $P(\mathbf{x}')$ the hyperrectangle of radius 0.5 centered in $\mathbf{x}'$: $P(\mathbf{x}) = \{\mathbf{x} \in \mathbb{R}^2 \mid 0 \leq x_1 \leq 1, -1 \leq x_2 \leq 0\}$. We observe that $C(\mathbf{x}') \approx -1.91$, but every point $\mathbf{x} \in P(\mathbf{x}')$ having $x_2 = 0$ will evaluate to the positive expression $2x_1^2 + 1$, hence C is not robust over $P(\mathbf{x})$. Visually, Γ crosses $P(\mathbf{x})$ leaving some vertices on different sides of the space.

We consider $a = (0.5 \pm 0.5\epsilon_1, -0.5 \pm 0.5\epsilon_2) \in \mathrm{RAF}_2^2$ abstracting $P(\mathbf{x}')$, and compute $a_{out} = c \pm 0.5\epsilon_1 \pm (1 + \sqrt{2})\epsilon_2 \pm d\epsilon_r$, where the values of the center $c \in \mathbb{R}$ and the non linear accumulation term $d \in \mathbb{R}$ are omitted as not relevant for this example. As a result we obtain the feature importance gradient $(0.5, 1+\sqrt{2})$ having both components positive. Since $C(\mathbf{x}') < 0$ we are looking for positive counterexamples and, following step 3, we build $\mathbf{x}^* \in P(\mathbf{x}')$ by selecting the maximum values for x_1, x_2 in $P(\mathbf{x}')$, which yields $\mathbf{x}^* = (1, 0)$. Then compute $C(\mathbf{x}^*) = 2(1)^2 + 2(2+\sqrt{(}2)(0) + 1 = 3 > 0$, hence a counterexample is found and C can be marked as not robust. □

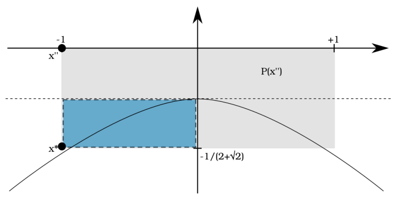

Figure 1: Recursion while searching for counterexamples.

Example 6.2. Continuing Example 6.1, we consider $\mathbf{x}'' = (-1, 0) \in \mathbb{R}^2$, and the region $P(\mathbf{x}'') = \{\mathbf{x} \in \mathbb{R}^2 \mid -1 \leq x_1 \leq +1, -\frac{1}{2+\sqrt{2}} \leq x_2 \leq 0\}$, as depicted by Fig. 1. It is easy to observe that every vertex of $P(\mathbf{x}'')$ is on the same side of Γ and thus receives the same label, while a small region in the bottom lays on the other side making C not robust. We repeat the counterexample search process obtaining the importance gradient $(0, \frac{3+2\sqrt{2}}{(2+\sqrt{2})^2})$, this time looking for a negative counterexample hence moving in the opposite direction with respect to the gradient. By doing so we compute $\mathbf{x}^* = (-1, -\frac{1}{2+\sqrt{2}})$ as the bottom-left corner, but $C(\mathbf{x}^*) = 1$ which does not produce a counterexample. We therefore follow steps 4 and 5 partitioning $P(\mathbf{x}'')$ with $x_2 < -\frac{1}{2(2+\sqrt{2})}$ as shown in the picture, and starting the recursion. After the first recursive step, none of the rectangles have counterexamples in their vertices, so another recursive call is started on the lowest one after partitioning on $x_1 < 0$. Both the resulting rectangles have now counterexamples in their vertices, which can be found by step 1 and 2. C can thus be marked as non robust. □

7 Experimental Evaluation

We consider datasets standard to the fairness literature (Mehrabi et al., 2021a) (a) **Adult** (Dua and Graff, 2017), which labels yearly incomes (above or below 50K US$) based on personal attributes. (b) **Compas** (Angwin et al., 2016), which labels recidivism risk based on personal attributes and criminal history. (c) **Crime** (Dua and Graff, 2017), which labels communities below or above the median per capita violent crime rate based on socio-economic, law enforcement, and crime data. (d) **German** (Dua and Graff, 2017), which labels (good or bad) credit scores (Dua and Graff, 2017). (e) **Health** (`https://www.kaggle.com/c/hhp`), which labels ten-year mortality (above or below the median Charlson index) for a patient based on physician records and insurance claims.

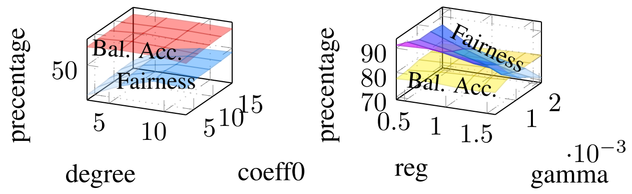

Figure 2: Trends on hyperparameters for SVMs with polynomial (left) and RBF (right) kernels trained on Crime.

The data is preprocessed following (Ruoss et al., 2020, Section 5). Some of these datasets exhibit a highly unbalanced label distribution, leading to high accuracy and 100% individual fairness for a constant classifier like $C(\mathbf{x}) = 1$. Thus, following (Ruoss et al., 2020) we also report the *balanced accuracy*, i.e., $\frac{1}{2}\left(\frac{\textit{truePos}}{\textit{truePos}+\textit{falseNeg}} + \frac{\textit{trueNeg}}{\textit{trueNeg}+\textit{falsePos}}\right)$.

Similarity Relations. Let $I \subseteq \mathbb{N}$ denote the set of features after one hot encoding and $\mathbf{x}, \mathbf{y} \in X$ be two individuals. Following (Ruoss et al., 2020, Section 5.1), we consider three similarity relations.

NOISE: $S_{\textit{noise}}(x, y)$ iff $|\mathbf{x}_i - \mathbf{y}_i| \leq \epsilon$ for all $i \in I'$, and $\mathbf{x}_i = \mathbf{y}_i$ for all $i \in I \setminus I'$, where $I' \subseteq I$ is a subset of numerical features. For our experiments, we consider $\epsilon = 0.05$ which leads upto a 10% perturbation for data normalised to [0,1].

CAT: $S_{\textit{cat}}(x, y)$ iff $\mathbf{x}_i = \mathbf{y}_i$ for all $i \in I \setminus I'$, where $I' \subseteq I$ represent sensitive categorical attributes. For Adult and German, we select the gender attribute. For Compas, its race. For Crime, we consider state. Lastly, for Health, we consider gender and age group.

NOISE-CAT: $S_{\textit{noise-cat}}(x, y) = S_{\textit{noise}} \cup S_{\textit{cat}}(x, y)$

Further domain-specific similarities can be defined and handled by our approach by simply instantiating the underlying static analyzer with a suitable abstract domain.

Setup. We trained the SVMs used in our experiments with scikit-learn (Pedregosa et al., 2011). Hyperparameters were chosen by hit-and-trial and observing trends. For instance, Fig. 2 shows trends in balanced accuracy and individual fairness with different hyperparameters for SVMs with polynomial and RBF kernels trained on the Crime dataset. Individual fairness is calculated with respect to the NOISE-CAT similarity relation using the RAF+OH abstract domain. The plots show that balanced accuracy and individual fairness are inversely correlated and the change in fairness is steeper than accuracy. Similar trends occur for SVMs trained on the other datasets. The final chosen hyperparameters for each kernel for each dataset were those that lead to SVMs with high balanced accuracy. Our implementation, the datasets with preprocessing pipelines, our

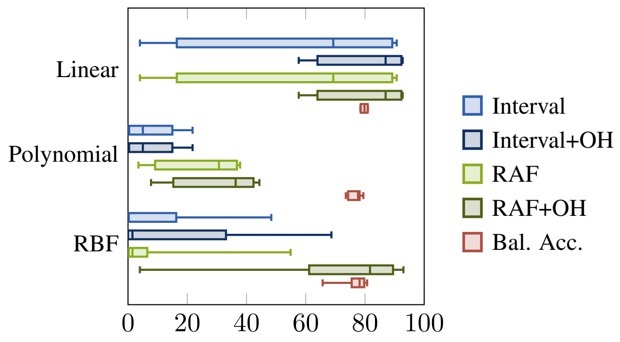

Figure 3: Comparison of individual fairness using different abstract domains for SVMs trained on Crime.

Table 1: Bounds on fairness with respect to the NOISE-CAT similarity relation using the RAF+OH abstraction.

Dataset	Linear		Polynomial		RBF	
	LB	UB	LB	UB	LB	UB
Crime	29.6	58.4	39.1	72.9	91.0	92.5
Health	95.5	99.6	0.02	98.7	29.4	97.4
Compas	94.9	94.9	0.09	71.4	89.3	93.0
German	81.5	81.5	10.0	76.0	0.0	84.0
Adult	91.6	91.6	0.03	89.5	92.2	95.4

SVM models and experiment scripts will be made available on GitHub upon publication of this work.

Individual Fairness. We show a summary of individual fairness and balanced accuracy scores obtained for SVMs trained on the Crime dataset in Fig. 3. We can see that the RAF abstraction typically outperforms intervals, and our OH abstraction always yields equal or higher individual fairness score. The full raw data for each dataset shows the same trend. It is shown in Table 3 in Appendix A.2.

In Table 1, for each dataset, we show bounds on individual fairness with respect to the NOISE-CAT similarity relation using the RAF+OH abstraction: the lower bound is the verified individual fairness score (cf. Section 5) and the upper bound is the estimate obtained with our counterexample search without input partitioning (cf. Section 6), e.g., an upper bound of 76% indicates that we found concrete counterexamples to individual fairness for 48 over 200 test data points. The gap between bounds is quite narrow for linear SVMs as well as for SVMs with RBF kernels trained on the Crime, Compas, and Adult datasets: in these cases, our RAF+OH abstraction is precise and our counterexample search heuristic is strong. On the other hand, the gap is much wider in the other cases, notably for SVMs with polynomial kernels, mostly due to a lower precision of the abstraction. Using partitioning up to 3.125% of the original input size, yields similar upper bound values, which indicates the presence of few additional counterexamples. Only partitioning up to 0.1% of the original size input size, we could find substantially more counterexamples.

Table 2: Comparison of our AFI and PFI on Compas.

Linear	LB	87.2	93.3	97	97.1	97.7	99	100
	AFI (0.225s)	1	2	3	4	5	6	7
	PFI (1976s)	1	2	5	4	3	7	6
Polynomial	LB	1.23	1.32	1.32	12.5	16.2	34.5	48.7
	AFI (0.199s)	4	6	5	3	2	1	7
	PFI (5166s)	3	1	5	7	4	2	6
RBF	LB	69	71.5	73.7	74	75.4	76	76.7
	AFI (0.267s)	1	2	4	3	5	6	7
	PFI (11776s)	2	1	4	3	5	7	6

Feature Importance. We now compare our feature importance measure AFI with PFI (as implemented in `sklearn.inspection` with `n_repeat` = 10). As a representative example, we show in Table 2, a comparison on SVMs trained on the Compas dataset. For each SVM model, in line 'LB', we order the numerical features in the datasets (i.e., 'priors_count', 'juv_fel_count', etc.) based on the verified individual fairness score with respect to the NOISE perturbation (with $\epsilon = 0.3$ in order to amplify the difference in score between the features). In lines 'AFI' and 'PFI' we show the order of these features based on the importance measured by AFI and PFI, respectively. We also indicate in parenthesis the time it took to compute these measures. We used the RAF+OH abstraction for AFI. We can see that AFI better correlates with model variance to feature perturbations. In fact, the correlation is perfect in the linear SVM case. In the polynomial SVM case, the lower individual fairness scores indicate that the abstraction loses precision and this explains why AFI is less accurate (but still better than PFI, at least in identifying the least important feature). Note also that AFI is computed in a small fraction of time compared to PFI. We show similar or better results for other datasets in Appendix A.2.

Finally, we recall that AFI can also measure the importance of tiers of a feature (cf. Section 4). Thus, it offers another way to detect potential bias issues starting from discrepancies between tier importances (e.g., we observed that the tier 'wife' of the 'relationship' feature of the Adult dataset results has a different importance from the tier 'husband').

8 Conclusion

We put forward a novel abstract feature importance measure based on an abstract interpretation for SVMs tailored for achieving a precise symbolic representation of one-hot encoded features. We showed that our abstraction is effective for verifying robustness properties —notably, individual fairness— of SVMs and that our abstract feature importance measure outperforms the state-of-the-art.

As future work, we plan to extend our approach to verify alternative or stronger fairness notions. We also aim to design quantitative verification methods to provide probabilistic guarantees on the behavior of SVM models.

Acknowledgements

Francesco Ranzato has been partially funded by the *Italian MIUR*, under the PRIN2017 project no. 201784YSZ5 "AnalysiS of PRogram Analyses (ASPRA)", by *Facebook Research*, under a "Probability and Programming Research Award", by an *Amazon Research Award* for "AWS Automated Reasoning", and by a *WhatsApp Research Award* for "Privacy aware program analysis".

References

Aho, A. V., Lam, M. S., Sethi, R., and Ullman, J. D. (2006). *Compilers: Principles, Techniques, and Tools (2nd Edition)*. Addison-Wesley Longman Publishing Co., Inc., USA.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine Bias. *ProPublica, May*, 23:2016.

Apley, D. W. and Zhu, J. (2020). Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. *Journal of the Royal Statistical Society Series B*, 82(4):1059–1086.

Bhatt, U., Xiang, A., Sharma, S., Weller, A., Taly, A., Jia, Y., Ghosh, J., Puri, R., Moura, J. M. F., and Eckersley, P. (2020). Explainable Machine Learning in Deployment. In *FAT* 2020*, pages 648–657.

Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1):5–32.

Buolamwini, J. and Gebru, T. (2018). Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on fairness, accountability and transparency*, pages 77–91. PMLR.

Carlini, N. and Wagner, D. A. (2017). Towards Evaluating the Robustness of Neural Networks. In *Proc. of 38th IEEE Symposium on Security and Privacy (S & P 2017)*, pages 39–57.

Casalicchio, G., Molnar, C., and Bischl, B. (2018). Visualizing the Feature Importance for Black Box Models. In *ECML/PKDD*, pages 655–670.

Chang, Y.-W. and Lin, C.-J. (2008). Feature ranking using linear svm. In *Causation and prediction challenge*, pages 53–64. PMLR.

Chouldechova, A. (2017). Fair Prediction with Disparate Impact: A Study of Bias in Recidivism Prediction Instruments. *Big Data*, 5(2):153–163.

Cousot, P. (2021). *Principles of Abstract Interpretation*. MIT Press.

Cousot, P. and Cousot, R. (1977). Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proc. 4th ACM Symposium on Principles of Programming Languages (POPL 1977)*, pages 238–252.

Cristianini, N. and Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press.

Dua, D. and Graff, C. (2017). Uci machine learning repository.

Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. S. (2012). Fairness through awareness. In *Innovations in Theoretical Computer Science 2012*, pages 214–226. ACM.

Fish, B., Kun, J., and Lelkes, Á. D. (2016). A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM international conference on data mining*, pages 144–152. SIAM.

Fisher, A., Rudin, C., and Dominici, F. (2019). All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. *Journal of Machine Learning Research*, 20(177):1–81.

Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics*, pages 1189–1232.

Ghosh, B., Basu, D., and Meel, K. S. (2022). Algorithmic fairness verification with graphical models. In *Proceedings of AAAI*, volume 2.

Goldstein, A., Kapelner, A., Bleich, J., and Pitkin, E. (2015). Peeking Inside the Black Box: Visualizing Statistical Learning With Plots of Individual Conditional Expectation. *Journal of Computational and Graphical Statistics*, 24(1):44–65.

Goodfellow, I., McDaniel, P., and Papernot, N. (2018). Making Machine Learning Robust Against Adversarial Inputs. *Commun. ACM*, 61(7):56–66.

Hechtlinger, Y. (2016). Interpretation of Prediction Models Using the Input Gradient. *CoRR*, abs/1611.07634.

Hooker, G., Mentch, L., and Zhou, S. (2019). Unrestricted permutation forces extrapolation: Variable importance requires at least one more model, or there is no free variable importance.

Khandani, A. E., Kim, A. J., and Lo, A. W. (2010). Consumer Credit-Risk Models via Machine-Learning Algorithms. *Journal of Banking & Finance*, 34(11):2767–2787.

Langenberg, P., Balda, E., Behboodi, A., and Mathar, R. (2019). On the robustness of support vector machines against adversarial examples. In *2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–6. IEEE.

Liu, C., Arnon, T., Lazarus, C., Strong, C., Barrett, C., Kochenderfer, M. J., et al. (2021). Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404.

Lundberg, S. M. and Lee, S. (2017). A Unified Approach to Interpreting Model Predictions. In *NeurIPS*, pages 4765–4774.

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021a). A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6).

Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., and Galstyan, A. (2021b). A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6):1–35.

Messine, F. (2002). Extentions of Affine Arithmetic: Application to Unconstrained Global Optimization. *The Journal of Universal Computer Science*, 8(11):992–1015.

Mladenić, D., Brank, J., Grobelnik, M., and Milic-Frayling, N. (2004). Feature selection using linear classifier weights: interaction with classification models. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 234–241.

Obermeyer, Z., Powers, B., Vogeli, C., and Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453.

Park, S., Byun, J., and Lee, J. (2022). Privacy-preserving fair learning of support vector machine with homomorphic encryption. In *Proceedings of the ACM Web Conference 2022*, pages 3572–3583.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Ranzato, F., Urban, C., and Zanella, M. (2021). Fairness-aware training of decision trees by abstract interpretation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 1508–1517.

Ranzato, F. and Zanella, M. (2019). Robustness verification of support vector machines. In *International Static Analysis Symposium*, pages 271–295. Springer.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *KDD*, pages 1135–1144.

Rival, X. and Yi, K. (2020). *Introduction to Static Analysis: An Abstract Interpretation Perspective*. The MIT Press.

Roh, Y., Lee, K., Whang, S., and Suh, C. (2020). Fr-train: A mutual information-based approach to fair and robust training. In *Proc. of the 37th Int. Conf. on Machine Learning (ICML 2020)*, volume 119 of *Proceedings of Machine Learning Research*, pages 8147–8157. PMLR.

Ruoss, A., Balunovic, M., Fischer, M., and Vechev, M. T. (2020). Learning certified individually fair representations. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems (NeurIPS 2020)*.

Skalna, I. and Hladík, M. (2017). A new algorithm for Chebyshev minimum-error multiplication of reduced affine forms. *Numerical Algorithms*, 76(4):1131–1152.

Strumbelj, E. and Kononenko, I. (2014). Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowledge and Information Systems*, 41(3):647–665.

Tjoa, E. and Guan, C. (2021). A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Transactions on Neural Networks and Learning Systems*, 32(11):4793–4813.

Urban, C., Christakis, M., Wüstholz, V., and Zhang, F. (2020). Perfectly parallel fairness certification of neural networks. *Proc. ACM Program. Lang.*, 4(OOPSLA):185:1–185:30.

Urban, C. and Miné, A. (2021). A Review of Formal Methods applied to Machine Learning. *CoRR*, abs/2104.02466.

Verma, S. and Rubin, J. (2018). Fairness definitions explained. In *2018 ieee/acm international workshop on software fairness (fairware)*, pages 1–7. IEEE.

Wegman, M. N. and Zadeck, F. K. (1991). Constant propagation with conditional branches. *ACM Trans. Program. Lang. Syst.*, 13(2):181–210.

Xiao, H., Biggio, B., Nelson, B., Xiao, H., Eckert, C., and Roli, F. (2015). Support vector machines under adversarial label contamination. *Neurocomputing*, 160:53–62.

Yurochkin, M., Bower, A., and Sun, Y. (2020). Training individually fair ML models with sensitive subspace robustness. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

A Appendix

A.1 Proofs

Proof of Theorem 4.3. We must show that $X = \gamma^{\mathrm{OH}_k}(a)$. Since we required values in X to be the result of a one-hot encoding, values other than f, t cannot occur. We start by calling $F_i = \{x_i | \mathbf{x} \in X \wedge x_i = f\}$ and $T_i = \{x_i | \mathbf{x} \in X \wedge x_i = t\}$ the sets of false and true values occurring the i-esim component, for every $1 \leq i \leq k$. We observe that F_i can be either the singleton of f or the empty set, the latter if and only if the i-esim member of the tier is never false in X. The same is true for T_i being either the singleton of t or the empty set.

We now show that $X \subseteq \gamma^{\mathrm{OH}_k}(a)$, that is, every element of the former also belongs to the latter. If X is the empty set condition is trivially satisfied. Otherwise, we consider a generic element $\mathbf{x}' \in X$. Since $\mathbf{x}'$ is the result of a one-hot encoding procedure, there exists $1 \leq i \leq k$ such that $x'_i = t$, while $x'_j = f$ for every $1 \leq j \leq k, j \neq i$. As a consequence, T_i cannot be the empty set, and must be $T_i = \{t\}$, similarily we obtain $F_j = \{f\}$ for every $j \neq i$. This implies that $a_{i,t} = t$ and $a_{j,f} = f$, allowing to explicitly compute

$$\begin{aligned}\hat{\gamma}_i(a) =& \{\mathbf{x} \in \mathbb{R}^k | x_i \in \gamma^{\mathrm{CP}}(a_{i,t}) \wedge \\ & \forall j \neq i \colon x_j \in \gamma^{\mathrm{CP}}(a_{j,f})\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_i \in \gamma^{\mathrm{CP}}(t) \wedge \forall j \neq i \colon x_j \in \gamma^{\mathrm{CP}}(f)\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_i \in \{t\} \wedge \forall j \neq i \colon x_j \in \{f\}\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_i = t \wedge \forall j \neq i \colon x_j = f\} \\ =& \{\mathbf{x}'\}\end{aligned}$$

which in turns implies $\mathbf{x}' \in \hat{\gamma}_i(a) \subseteq \bigcup_{j=1}^{k} \hat{\gamma}_j(a) = \gamma^{\mathrm{OH}_k}(a)$, hence $X \subseteq \gamma^{\mathrm{OH}_k}(a)$.

Last, we show that $\gamma^{\mathrm{OH}_k}(a) \subseteq X$ by arguing that every element of the former also belongs to the latter. If $\gamma^{\mathrm{OH}_k}(a)$ is the empty set implication is trivially satisfied, otherwise we consider a generic element $\mathbf{x}' \in \gamma^{\mathrm{OH}_k}(a)$. Since $\gamma^{\mathrm{OH}_k}(a) = \bigcup_{i=1}^{k} \hat{\gamma}_i(a)$, there must exists some $1 \leq z \leq k$ such that $\mathbf{x}' \in \hat{\gamma}_z(a)$, which can be written as

$$\begin{aligned}\hat{\gamma}_z(a) =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \gamma^{\mathrm{CP}}(a_{z,t}) \\ & \wedge \forall j \neq z \colon x_j \in \gamma^{\mathrm{CP}}(a_{j,f})\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(T_z)) \\ & \wedge \forall j \neq z \colon x_j \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(F_j))\}\end{aligned}$$

if $T_z = \varnothing$, or $\exists j \neq z \colon F_j = \varnothing$ then $\hat{\gamma}_z(a) = \varnothing$, which is not acceptable and thus must not be considered. As a consequence, it must be $T_z \neq \varnothing$ and $\forall j \neq z \colon F_j \neq \varnothing$, which in turns implies $T_z = \{t\}$ and $\forall j \neq z \colon F_j = \{f\}$. This allows to rewrite

$$\begin{aligned}\hat{\gamma}_z(a) =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(T_z)) \\ & \wedge \forall j \neq z \colon x_j \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(F_j))\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(\{t\})) \\ & \wedge \forall j \neq z \colon x_j \in \gamma^{\mathrm{CP}}(\alpha^{\mathrm{CP}}(\{f\}))\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \gamma^{\mathrm{CP}}(t) \\ & \wedge \forall j \neq z \colon x_j \in \gamma^{\mathrm{CP}}(f)\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_z \in \{t\} \wedge \forall j \neq z \colon x_j \in \{f\}\} \\ =& \{\mathbf{x} \in \mathbb{R}^k | x_z = t \wedge \forall j \neq z \colon x_j = f\}\end{aligned}$$

which is a singleton. Since $\mathbf{x}' \in \hat{\gamma}_z(a)$, we can write $\hat{\gamma}_z(a) = \{\mathbf{x}'\}$. On the other hand, $T_z = \{t\}$ also implies $\exists \mathbf{x}'' \in X \colon x''_z = t$. Moreover, it must be $x''_j = f$ for all $j \neq i$, as $\mathbf{x}''$ is the result of a one-hot encoding, hence $\mathbf{x}'' = \mathbf{x}'$ and $\mathbf{x}' \in X$, which leads to the conclusion that every $\hat{\gamma}_i(a) \subseteq X$, hence $\gamma^{\mathrm{OH}_k}(a) \subseteq X$.

Since both $X \subseteq \gamma^{\mathrm{OH}_k}(a)$ and $\gamma^{\mathrm{OH}_k}(a) \subseteq X$ hold at the same time, we can conclude that $X = \gamma^{\mathrm{OH}_k}(a)$ and a precisely X without loss of precision. □

Proof of Theorem 4.5. It must be shown that, for any $f : \mathbb{R} \rightarrow \mathbb{R}$ and for any $a \in \mathrm{OH}_k$, $f(\gamma^{\mathrm{OH}_k}(a)) \subseteq \gamma^{\mathrm{OH}_k}(f^{\mathrm{OH}}(a))$.

$$\begin{aligned} f(\gamma^{\mathrm{OH}_k}(a)) =& f\left(\bigcup_{i=1}^{k} \hat{\gamma}_i(a)\right) \\ =& \bigcup_{i=1}^{k} f(\hat{\gamma}_i(a)) \\ =& \bigcup_{i=1}^{k} \{f(\mathbf{x}) | x_i \in \gamma^{\mathrm{CP}}(a_{i,t}) \wedge \\ & \forall j \neq i \colon x_j \in \gamma^{\mathrm{CP}}(a_{j,f})\} \\ =& \bigcup_{i=1}^{k} \{\mathbf{x} | x_i \in f(\gamma^{\mathrm{CP}}(a_{i,t})) \wedge \\ & \forall j \neq i \colon x_j \in f(\gamma^{\mathrm{CP}}(a_{j,f}))\} \\ \subseteq& \bigcup_{i=1}^{k} \{\mathbf{x} | x_i \in \gamma^{\mathrm{CP}}(f^{\mathrm{CP}}(a_{i,t})) \wedge \\ & \forall j \neq i \colon x_j \in \gamma^{\mathrm{CP}}(f^{\mathrm{CP}}(a_{j,f}))\} \\ =& \bigcup_{i=1}^{k} \hat{\gamma}_i(f^{\mathrm{OH}}(a)) = \gamma^{\mathrm{OH}_k}(f^{\mathrm{OH}}(a))\end{aligned}$$

□

Proof of Corollary 4.7. (i) By assuming no $\top^{\mathrm{CP}}$ value occurs in a, then $f(\gamma^{BC}(a_{i,z})) = \gamma^{BC}(f^{BC}(a_{i,z}))$ for any $1 \leq i \leq k$ and $z \in \{f, t\}$. This equivalence can be exploited in proof of Th. 4.5, yielding $f(\gamma^{\mathrm{OH}_k}(a)) = \gamma^{\mathrm{OH}_k}(f^{\mathrm{OH}}(a))$.

(ii) We will prove the stronger property that analysis is complete and no $\top^{\mathrm{CP}}$ values can be generated. Proof is by induction on the number $n \in \mathbb{N}$ of applications of functions f_i.

$P(0)$ Due to Th.4.3, it is possible to abstract X with some $a_0 \in \mathrm{OH}_k$ without loss of precision. Remark 4.4 also applies and guarantees that no $\top^{\mathrm{CP}}$ can occur in a_0.

$P(n) \Rightarrow P(n+1)$ By inductive hypotheses we have an intermediate abstract value $a_n \in \mathrm{OH}_k$ which represents its concrete counterpart without loss of precision, and we also know that a_n does not contain any $\top^{\mathrm{CP}}$ values. We can therefore compute $a_{n+1} = f_{n+1}^{\mathrm{OH}_k}(a_n)$ and, due to Corollary 4.7, no loss of precision can occur in a_{n+1}. Also, by definition of f^{OH} in Th.4.5, $\top^{\mathrm{CP}}$ cannot be introduced unless already present in a_n, which is not the case thanks to inductive hypotheses.

Alternatively, it is possible to consider $f = f_1 \circ f_2 \circ \ldots \circ f_n$ and directly apply Corollary 4.7, although this appears more restrictive as it does not allow for composition with other complete transfer functions in between. □

A.2 Additional Experimental Data

Table 3 shows accuracy, balanced accuracy, and individual fairness scores for SVMs with linear kernels (represented as L(reg. param.)), polynomial kernels (represented as P(reg. param, degree, base)), and RBF kernels (represented as R(reg. param, γ)). Individual fairness scores are computed with respect to the NOISE (N), CAT (C), and NOISE-CAT (NC) similarity relations. The table also compares results with and without the OH abstraction.

We show further comparisons between our proposed AFI and PFI in Table 4 and Table 5. We compute individual fairness scores with respect to the NOISE perturbation. We choose different ϵ values to amplify the difference in score between the features. Note the much better correlation of AFI with fairness scores than PFI. We do not report results for the Crime dataset as it has dozens of numerical features. PFI timed out after executing for several hours on Health.

Table 3: Accuracy, balanced accuracy, and individual fairness scores for SVMs trained on each dataset.

Dataset	**Kernel**	**Acc.**	**Bal. Acc**	**Interval**			**Interval OH**			**RAF**			**RAF OH**		
				C	NC	N	C	NC	N	C	NC	N	C	NC	N
Adult	L(1)	84.6	75.6	95.2	91.6	96.5	95.2	91.6	96.5	95.2	91.6	96.5	95.2	91.6	96.5
	R(0.05,0.01)	83.8	72.0	2.8	0.0	2.4	2.8	0.0	2.4	42.4	37.2	94.8	97.9	95.4	97.5
	P(0.01,3,3)	83.9	76.7	0.0	0.0	0.0	0.0	0.0	0.0	0.5	.03	0.5	0.5	0.03	0.5
Compas	L(1)	64.7	64.1	99.5	94.9	95.5	99.5	94.9	95.5	99.5	94.9	95.5	99.5	94.9	95.5
	R(0.05,0.01)	64.5	63.1	42.5	1.0	54.3	42.5	1.0	54.3	71.6	66.9	91.8	97.5	89.3	94.4
	P(0.01,3,3)	64.3	63.9	0.0	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.5	0.6	0.1	0.5
Crime	L(1)	82.0	82.0	0.75	0.25	67.2	60.9	29.6	67.2	0.75	0.25	67.2	60.9	29.6	67.2
	R(1,10^{-3})	77.7	77.7	1.0	0.0	37.8	92.5	29.3	37.8	22.1	16.0	90.2	100	91.0	90.2
	P(1,9,0)	74.2	74.1	71.9	9.0	13.3	71.9	9.0	13.3	83.0	31.3	55.4	93.5	39.1	55.4
German	L(1)	79.0	70.8	94.5	81.5	87.5	94.5	81.5	87.5	94.5	81.5	87.5	94.5	81.5	87.5
	R(10,0.05)	79.5	74.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	82.0	0.0	2.0
	P(0.01,6,6)	75.5	71.8	0.0	0.0	0.0	0.0	0.0	0.0	76.0	10.0	78.5	76.0	10.0	78.5
Health	L(0.01)	77.8	69.5	91.3	90.6	99.4	96.1	95.5	99.4	91.3	90.6	99.4	96.1	95.5	99.4
	R(0.1,0.01)	82.3	76.4	0.68	0.65	1.26	3.38	2.87	4.71	0.82	0.81	80.1	94.1	29.4	82.0
	P(0.1,3,0.01)	71.0	61.0	0.0	0.0	.005	0.0	0.0	.005	0.03	0.02	6.62	0.03	0.02	6.62

Table 4: Comparison of our AFI and PFI on Adult.

Linear ($\epsilon = 0.3$)	LB	83.4	96	98.37	98.37	98.44	99.2
	AFI	1	2	3	4	5	6
	PFI	1	2	6	4	5	3
Polynomial ($\epsilon = 0.1$)	LB	23.2	23.9	24.5	28.3	29.5	95.2
	AFI	2	5	6	3	4	1
	PFI	6	5	4	2	3	1
RBF ($\epsilon = 0.3$)	LB	50.3	53.5	56.4	56.6	57.7	57.9
	AFI	1	2	3	4	5	6
	PFI	2	1	4	5	6	3

Table 5: Comparison of our AFI and PFI on German.

Linear ($\epsilon = 0.3$)	LB	78	83.5	87	89.5	90	90	98.5	98.5	99.5
	AFI	1	2	3	4	5	6	7	8	9
	PFI	3	7	1	6	4	2	8	5	9
Polynomial ($\epsilon = 0.3$)	LB	83	86	86.5	87	92.5	92.5	93	94.5	94.5
	AFI	1	6	5	3	2	9	4	7	8
	PFI	2	3	4	5	9	1	6	7	8
RBF ($\epsilon = 0.1$)	LB	54	54	54.5	55	56.5	57	57.5	58	58.5
	AFI	1	2	3	4	5	6	8	7	9
	PFI	1	3	7	9	4	5	8	2	6