Robustness Certification of k-Nearest Neighbors

Nicolò Fassina*, Francesco Ranzato*, Marco Zanella*
*Dipartimento di Matematica, University of Padova, Italy

***Abstract*—We study the certification of stability properties, such as robustness and individual fairness, of the k-Nearest Neighbor algorithm (kNN). Our approach leverages abstract interpretation, a well-established program analysis technique that has been proven successful in verifying several machine learning algorithms, notably, neural networks, decision trees, and support vector machines. In this work, we put forward an abstract interpretation-based framework for designing a sound approximate version of the kNN algorithm, which is instantiated to the interval and zonotope abstractions for approximating the range of numerical features. We show how this abstraction-based method can be used for stability, robustness, and individual fairness certification of kNN. Our certification technique has been implemented and experimentally evaluated on several benchmark datasets. These experimental results show that our tool can formally prove the stability of kNN classifiers in a precise and efficient way, thus expanding the range of machine learning models amenable to robustness certification.**

***Index Terms*—k-nearest neighbors, robustness, individual fairness, formal certification**

I. Introduction

k-Nearest Neighbors (kNN) is one of the simplest supervised machine learning (ML) algorithms. Nevertheless, kNN is a popular and accurate predictive model with diverse application fields. The basic idea of kNN is to predict the outcome for an input sample $\mathbf{x} \in \mathbb{R}^n$ by inferring the k nearest neighbors of $\mathbf{x}$ ranging in a given dataset. The number $k \in \mathbb{N}$ of neighbors as well as the distance function between vectors are parameters of this model. Once the set of k nearest neighbors of an input sample is computed, the output is inferred as the most common label of these k neighbors in case of classification, or as average of the values of the k neighbors in case of regression. The diagram in Fig. 1 depicts an example of classification for a kNN model with $k = 3$ and a dataset in $\mathbb{R}^2$ with three classes: for an input vector $\mathbf{x}$ represented by a black bullet, 3NN therefore computes the 3 nearest samples in the dataset w.r.t. Manhattan distance, as depicted by the dashed lines, and then the most common label among them is inferred. In kNN the dataset is stored and entirely used at classification time, namely, kNN is a lazy learning algorithm. While this makes kNN simple to implement, it can exhibit a significant prediction time due to the computational effort required to calculate distances for the whole dataset and, correspondingly, for sorting samples, especially for high values of k (k is usually a low odd value, often below 9).

Francesco Ranzato and Marco Zanella were partially funded by the *Italian MIUR*, under the PRIN 2017 project no. 201784YSZ5. Francesco Ranzato was partially funded by: the *Italian MUR*, under the PRIN 2022 PNRR project no. P2022HXNSC; *Meta* (formerly *Facebook*) *Research*, under a "Probability and Programming Research Award" and under a *WhatsApp Research Award* on "Privacy-aware Program Analysis"; by an *Amazon Research Award* for "AWS Automated Reasoning".

Adversarial machine learning [14] studies vulnerabilities of ML in adversarial scenarios. Adversarial examples have been found in diverse application fields of ML, and the current defense techniques include adversarial model training, input validation, testing and automatic formal certification of learning algorithms. A ML classifier C is defined to be *stable* on an input $\mathbf{x}$ for a (typically very small) perturbation $P(\mathbf{x})$ of $\mathbf{x}$ which represents an adversarial attack, when C assigns the same class to all the samples in $P(\mathbf{x})$. Moreover, when such class is also the correct class of $\mathbf{x}$ with respect to ground truth, the classifier C is *robust* on $\mathbf{x}$ as it cannot be deceived by unnoticeable malicious alterations of $\mathbf{x}$. Fig. 1 depicts in grey an adversarial region $P(\mathbf{x})$ defined around the black input sample $\mathbf{x}$, which represents an (infinite) set of attacks. Here, the 3 nearest neighbors of each attack in $P(\mathbf{x})$ are labeled as *red* ($\mathbf{p}_1$, $\mathbf{p}_2$) and *green* ($\mathbf{p}_3$), making 3NN stable on $\mathbf{x}$ as 3NN classifies $\mathbf{x}$ are *red*. If *red* is the ground truth label for $\mathbf{x}$, then 3NN is robust on $\mathbf{x}$ as well.

Contributions. Our main contribution is a novel formal and automatic verification method for inferring when a kNN classifier is *provably stable* for an input sample with respect to a given perturbation. We leverage the well-established framework of abstract interpretation [5], [6], [13] for computing correct over-approximations of dynamic system behaviours, which has already been successfully applied to the formal verification of diverse machine learning models, see the surveys [1], [18]. Our approach is based on designing a sound abstract version $C^A_{\delta,k}$ of a kNN classifier based on a distance function δ, e.g. Euclidean or Manhattan distance. This approximate classifier $C^A_{\delta,k}$ is defined over a symbolic numerical abstraction A of the input space $\wp(\mathbb{R}^n)$, and leverages a sound approximation δ^A in A of the distance function. In turn, the definition of δ^A relies on sound approximations over A of its basic numerical operations such as addition, product, and modulus. Given an abstract value $a \in A$ which provides a symbolic over-approximation of an adversarial perturbation $P(\mathbf{x})$ of an input sample $\mathbf{x}$, $C^A_{\delta,k}(a)$ returns an over-approximation of the set of classes computed by kNN for all the samples in $P(\mathbf{x})$. Hence, if $C^A_{\delta,k}(a) = k\text{NN}(\mathbf{x})$ holds then we can infer that kNN is provably stable on $\mathbf{x}$ for its perturbation $P(\mathbf{x})$. We instantiate our certification method to the well-known numerical abstract domains of intervals [6] and zonotopes [15], that approximate the range of numerical features by, resp., real intervals (e.g., $\mathbf{x}_i \in [l, u]$) and affine forms (e.g., $\mathbf{x}_i = a_0 + \sum_{j=1}^{k} a_j \epsilon_j$ with $a_j \in \mathbb{R}$ and noise symbols $\epsilon_j \in [-1, 1]$). This certification framework for kNN has been implemented in Python. The corresponding tool, called NAVe (k**N**N **A**bstract **Ve**rifier; the Italian word "nave" means

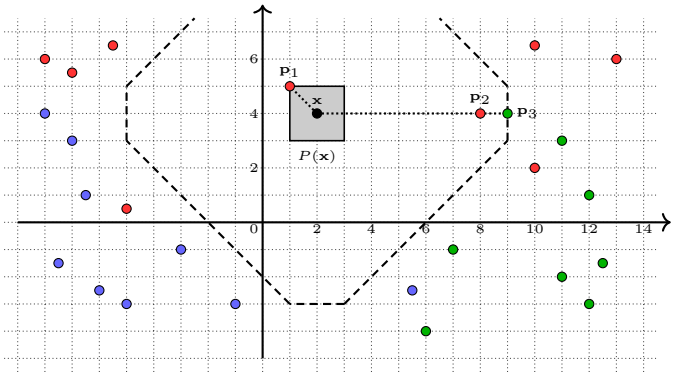

Fig. 1. kNN on a dataset with three classes *red*, *green*, *blue*.

"ship"), has been designed to be scalable both in the size of the training dataset and in the value of k, for which no upper bound is assumed. We performed an experimental evaluation of NAVe on 7 datasets commonly used in robustness certification and on 2 additional datasets for individual fairness verification. These experimental results show that NAVe is an effective tool for formally certifying the adversarial robustness of inputs to kNN, and that, in general, kNN turns out to be a quite robust prediction algorithm since for adversarial perturbations $\leq \pm 2\%$, NAVe is able to infer for several datasets more than 90% of robustness for $k \in \{1, 3, 5, 7\}$.

Illustrative Example. Let us consider the example in $\mathbb{R}^2$ depicted in Fig. 1, where $\mathbf{x} = (2, 4)$ is the input sample and $P(\mathbf{x}) \triangleq \{\mathbf{x}' \in \mathbb{R}^2 \mid \max(|\mathbf{x}'_1 - \mathbf{x}_1|, |\mathbf{x}'_2 - \mathbf{x}_2|) \leq 1\}$ is a perturbation defined as the ℓ_∞ ball of radius 1 centered in $\mathbf{x}$, which can be exactly represented through intervals as $(\mathbf{x}_1 \in [1, 3], \mathbf{x}_2 \in [3, 5])$. By leveraging the interval abstract domain $\mathcal{I}$, we compute the abstract Manhattan distance $\mu^{\mathcal{I}}$ between $P(\mathbf{x})$ and the 3 points $\mathbf{p}_1 = (1, 5)$, $\mathbf{p}_2 = (8, 4)$, $\mathbf{p}_3 = (9, 4)$:

$$\begin{aligned}
\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_1) &= |[1,3] -^{\mathcal{I}} 1|^{\mathcal{I}} +^{\mathcal{I}} |[3,5] -^{\mathcal{I}} 5|^{\mathcal{I}} \\
&= [0,2] +^{\mathcal{I}} [0,2] = [0,4]\,, \\
\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_2) &= |[1,3] -^{\mathcal{I}} 8|^{\mathcal{I}} +^{\mathcal{I}} |[3,5] -^{\mathcal{I}} 4|^{\mathcal{I}} \\
&= [5,7] +^{\mathcal{I}} [0,1] = [5,8]\,, \\
\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_3) &= |[1,3] -^{\mathcal{I}} 9|^{\mathcal{I}} +^{\mathcal{I}} |[3,5] -^{\mathcal{I}} 4|^{\mathcal{I}} \\
&= [6,8] +^{\mathcal{I}} [0,1] = [6,9]\,.
\end{aligned}$$

These abstract distances are symbolically computed in the interval abstraction $\mathcal{I}$ and provide correct lower and upper bounds for the infinite set of distances $\{\mu(\mathbf{y}, \mathbf{p}_i) \in \mathbb{R}_{\geq 0} \mid \mathbf{y} \in P(\mathbf{x})\}$. By leveraging these abstract distances, the abstract classifier $C^{\mathcal{I}}_{\mu,k}(P(\mathbf{x}))$ returns an over-approximation of the set of classes $\cup_{\mathbf{y} \in P(\mathbf{x})} k\text{NN}(\mathbf{y})$. Let us observe that $\mathbf{p}_1$ is the nearest point to $P(\mathbf{x})$, as its interval $[0, 4]$ is strictly dominated by all the others ($[l_1, u_1]$ is strictly dominated by $[l_2, u_2]$ when $u_1 < l_2$). As a consequence $C^{\mathcal{I}}_{\mu,1}(P(\mathbf{x})) = \{red\}$, so that 1NN is provably stable on $\mathbf{x}$. On the other hand, it turns out that $\mathbf{p}_2$ is closer than $\mathbf{p}_3$ to every point in $P(\mathbf{x})$, although this cannot be inferred from the corresponding abstract distances since the interval $[5, 8]$ for $\mathbf{p}_2$ is not strictly dominated by $[6, 9]$ for $\mathbf{p}_3$: this is an example of loss of precision, also called *incompleteness* of the stability certification. Consequently, if we use $k = 2$ in this scenario, then we cannot exclude $\mathbf{p}_3$ by the approximate set of neighbors, which could be either be $\{\mathbf{p}_1, \mathbf{p}_2\}$, thus resulting in *red*, or $\{\mathbf{p}_1, \mathbf{p}_3\}$, thus causing an ambiguity between *red* and *green*. This entails that $C^{\mathcal{I}}_{\mu,2}(P(\mathbf{x})) = \{red, green\}$, so that stability of 2NN on $\mathbf{x}$ cannot be proved. In this case, *green* is therefore a false positive, arising from the interval approximation. Finally, for $k = 3$, the stability verification turns out to be complete, because the three samples $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the unique points which will be taken into account, as hinted by the black dashed line in Fig. 1, so that $C^{\mathcal{I}}_{\mu,3}(P(\mathbf{x})) = \{red\}$ holds, thus allowing us to infer that 3NN is stable on $\mathbf{x}$.

Related Work. Formal verification methods in adversarial machine learning have been thoroughly investigated for (deep) neural networks, while different ML models have been much less studied. In particular, adversarial attacks on k-nearest neighbor algorithms have been studied only recently [2], [9], [16], [17], [30]–[34]. We do not have enough space for discussing all these works. Let us mention [30], where the authors put forward an algorithm, called GeoAdEx, based on higher-order Voronoi diagrams, that aims at finding the smallest perturbation that moves an input sample to an adversarial cell. However, finding this smallest perturbation, or a certified lower bound for it, may often need a long time, essentially due to a combinatorial complexity, so that in most cases GeoAdEx outputs exact results, i.e. without approximations, only for $k = 1$. Also, Fan et al. [9]'s approach is orthogonal to ours: their notion of robustness is different, since an input $\mathbf{x}$ is considered to be robust w.r.t. a set of datasets $\mathcal{I}$, when there exists a label l such that for all $D \in \mathcal{I}$, $k\text{NN}_D(\mathbf{x}) = l$. Let us finally mention that [17] proves robustness of kNN to adversarial poisonings of the dataset by leveraging an over-approximated kNN classifier. Abstract interpretation techniques have been applied for designing precise and scalable robustness verification algorithms and adversarial training techniques for a range of ML models [3], [11], [19], [22]–[26], [28], [29]. To the best of our knowledge, no prior work applied abstract interpretation for the robustness certification of k-nearest neighbors.

II. Background

Numerical Abstract Domains. A numerical abstract domain (or abstraction) [21] A symbolically represents sets of real vectors through a concretization map $\gamma^A : A \to \wp(\mathbb{R}^n)$ providing the meaning of its abstract (i.e., symbolic) values. A subset of vectors $S \in \wp(\mathbb{R}^n)$ is over-approximated by $a \in A$ when $S \subseteq \gamma^A(a)$, while S is exactly represented by a when $S = \gamma^A(a)$ holds. An abstract domain A may also admit an abstraction function $\alpha^A : \wp(\mathbb{R}^n) \to A$ such that $\alpha^A(S)$ is the best abstraction in A of the set S, where the notion of best means least (or minimal) w.r.t. the preorder relation $a \sqsubseteq^A a' \Leftrightarrow \gamma^A(a) \subseteq \gamma^A(a')$.

Given a k-ary operation on vectors $f : (\mathbb{R}^n)^k \to \mathbb{R}^n$, for some $k \geq 1$, an abstract function $f^A : A^k \to A$ is a sound (or

correct) approximation of f when for all $(a_1, ..., a_k) \in A^k$, $\{f(\mathbf{x}_1, ..., \mathbf{x}_k) \mid \forall i.\, \mathbf{x}_i \in \gamma^A(a_i)\} \subseteq \gamma^A(f^A(a_1, ..., a_k))$ holds, while f^A is defined to be exact (or complete) when equality holds. In words, soundness holds when $f^A(a_1, ..., a_k)$ never misses a concrete computation of f on some input $(\mathbf{x}_1, ..., \mathbf{x}_k)$ which is abstractly represented by $(a_1, ..., a_k)$, while exactness means that each abstract computation $f^A(a_1, ..., a_k)$ is an exact abstract representation of the set of concrete computations of f on all the inputs that are abstractly represented by $(a_1, ..., a_k)$. The abstract domain of real intervals $\mathcal{I}$ is one of the simplest and most used abstractions in ML verification. The interval domain abstracts the values of a real variable by a (possibly unbounded) real interval $[l, u]$, where $l, u \in \mathbb{R} \cup \{-\infty, +\infty\}$ and $l \leq u$ (with $-\infty \leq x \leq +\infty$ for all $x \in \mathbb{R}$). The concretization $\gamma^{\mathcal{I}} \colon \mathcal{I} \to \wp(\mathbb{R})$ is defined as $\gamma^{\mathcal{I}}([l, u]) \triangleq \{x \in \mathbb{R} \mid l \leq x \leq u\}$. The product interval abstraction $\mathcal{I}^n$, with $n \geq 1$, is also called the box domain and its concretization map $\gamma^{\mathcal{I}^n} : \mathcal{I}^n \to \wp(\mathbb{R}^n)$ is defined by a straightforward componentwise product of $\gamma^{\mathcal{I}}$. The interval domain can be imprecise as it is nonrelational, i.e., $\mathcal{I}$ does not represent information on how values of different variables are related: for example, the most precise interval approximation of the set $T = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 1, x = y\}$ is $\langle x \in [0, 1], y \in [0, 1]\rangle$, thus losing the information that $x - y = 0$. The zonotope abstract domain $\mathcal{Z}$ [12] is based on affine arithmetic [7] and consists of abstract values $\hat{a} = a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathcal{Z}$, where $a_j \in \mathbb{R}$ are coefficients and ϵ_j are noise symbols whose values range in the interval $[-1, 1]$, and when these ϵ_j are shared between different variables/features encode a relation between them. The concretization of a zonotope $\hat{a}$ is given by $\gamma^{\mathcal{Z}}(\hat{a}) \triangleq \{a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathbb{R} \mid \forall j.\, \epsilon_j \in [-1, 1]\}$, i.e., the zonotope $\hat{a}$ represents the real interval $\left[a_0 - \sum_{j=1}^{m} |a_j|, a_0 + \sum_{j=1}^{m} |a_j|\right]$. The product zonotope abstraction $\mathcal{Z}^n$, with $n \geq 1$, may share noise symbols between different components, thus enabling to represent relational information between features. For example, the above set $T \subseteq \mathbb{R}^2$ can be exactly represented by the zonotope $(\hat{x} = 0.5 + 0.5\epsilon_1, \hat{y} = 0.5 + 0.5\epsilon_1)$, so that we can infer that $\hat{x} - \hat{y} = 0$ holds. A fundamental property of zonotopes is that linear functions, such as vector addition and constant multiplication, admit corresponding exact abstract operations on $\mathcal{Z}$, while nonaffine functions, such as multiplications and modulus, must necessarily be approximated. The basic abstract operations on intervals and zonotopes for computing abstract distances are recalled below, where the exponential function boils down to iterated multiplications.

The most precise abstract operations (also called best correct approximations) on $\mathcal{I}$ are well-known [21] and given below:

addition: $[l_1, u_1] +^{\mathcal{I}} [l_2, u_2] \triangleq [l_1 + l_2, u_1 + u_2]$

multiplication: $[l_1, u_1] \cdot^{\mathcal{I}} [l_2, u_2] \triangleq$
$[\min(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2), \max(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2)]$

modulus: $|[l, u]|^{\mathcal{I}} \triangleq \begin{cases} [\min(|l|, |u|), \max(|l|, |u|)] & \text{if } lu \geq 0 \\ [0, \max(|l|, |u|)] & \text{otherwise} \end{cases}$

dominance test: $[l_1, u_1] <^{\mathcal{I}} [l_2, u_2] \triangleq u_1 < l_2$

Zonotopes are exact for linear operations, namely addition, while for nonlinear operations, in particular multiplication, the result, in general, cannot be exactly represented by a zonotope, so that the multiplication of zonotopes approximates the precise result by adding a fresh noise symbol ϵ_{f} whose coefficient is typically computed by a Taylor approximation of the nonlinear part of the multiplication (see [15, Section 2.1.5]). If $\hat{a} = a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathcal{Z}$ and $\hat{b} = b_0 + \sum_{j=1}^{m} b_j \epsilon_j \in \mathcal{Z}$ then:

addition: $\hat{a} +^{\mathcal{Z}} \hat{b} \triangleq (a_0 + b_0) + \sum_{j=1}^{m} (a_j + b_j)\epsilon_j$

multiplication: $\hat{a} \cdot^{\mathcal{Z}} \hat{b} \triangleq \big(a_0 b_0 + \frac{1}{2} \sum_{j=1}^{m} |a_j b_j|\big) +$
$\sum_{j=1}^{m} (a_j b_0 + b_j a_0)\epsilon_j + \big(\frac{1}{2} \sum_{j=1}^{m} |a_j b_j| + \sum_{1 \leq i < j \leq m} |a_i b_j + a_j b_i|\big)\epsilon_{\mathrm{f}}$

dominance test: $\hat{a} <^{\mathcal{Z}} \hat{b} \triangleq a_0 - b_0 + \sum_{j=1}^{m} |a_j - b_j| < 0$

An Abstract Modulus on Zonotopes. To the best of our knowledge, no algorithm implementing a sound abstraction of the modulus on zonotopes is available in literature. Thus, we designed a novel abstract function on $\mathcal{Z}$ that approximates the generic modulus operation by a hyperplane and introduces a nonlinear noise contribution to guarantee soundness. Due to lack of space, we omit this general definition. On the other hand, we observe that in our algorithm for the stability certification of kNN the modulus function always has a specific form that can be exploited to enhance the time complexity of its abstract version on $\mathcal{Z}$. This instance of the modulus boils down to $|a_0 + a_j \epsilon_j|^{\mathcal{Z}}$, for some $j \in [1, m]$, namely, to the modulus of a line with unknown ϵ_j on a plane. Hence, we derived an abstract modulus by computing the line on that plane including the two extremal points $(-1, a_0 - a_j)$ and $(+1, a_0 + a_j)$ as a correct upper bound for that modulus. Additionally, we determine the parallel line passing through the point $(-\frac{a_0}{a_j}, 0)$ as a correct lower bound. Finally, we compute the line $y = px + q$ parallel to these two lines and at the same distance $d > 0$ from them. This allows us to define as abstract modulus $|a_0 + a_j \epsilon_j|^{\mathcal{Z}} \triangleq q + p\epsilon_j + d\epsilon_{\mathrm{f}}$, where ϵ_{f} is a fresh noise symbol.

kNN Classifiers. Consider a ground truth dataset $D \subseteq X \times L$, where $X \subseteq \mathbb{R}^n$ is an input space and L is a set of classification labels, and a distance function $\delta : X \times X \to \mathbb{R}_{\geq 0}$. Given $k \in \mathbb{N}^* \triangleq \mathbb{N} \setminus \{0\}$, a kNN classifier is modeled as a total function $C_{\delta,k} \colon X \to \wp(L)$, which maps an input sample $\mathbf{x} \in X$ into a nonempty set of labels, by first selecting in D the k nearest samples to $\mathbf{x}$ according to δ, and then returning the set of their most frequent labels. Hence, an output set including more than one label means a tie vote, and this explains why we consider sets of labels as codomain of classifiers.

Stability and Robustness. A perturbation of an input sample $\mathbf{x} \in X$ is a variation of its feature values defining a potential adversarial region $R \subseteq X$. A very common instance [4] is given by perturbations for the maximum norm ℓ_∞: given $\mathbf{x} \in \mathbb{R}^n$ and a magnitude $\tau > 0$, the ℓ_∞-perturbation is $P^{\tau}_{\infty}(\mathbf{x}) \triangleq \{\mathbf{w} \in \mathbb{R}^n \mid \max(|\mathbf{w}_1 - \mathbf{x}_1|, ..., |\mathbf{w}_n - \mathbf{x}_n|) \leq \tau\}$. This perturbation can be exactly represented through intervals and zonotopes, that is, $P^{\tau}_{\infty}(\mathbf{x}) = \gamma^{\mathcal{I}^n}(\langle [\mathbf{x}_1 - \tau, \mathbf{x}_1 + \tau], ..., [\mathbf{x}_n - \tau, \mathbf{x}_n + \tau]\rangle) = \gamma^{\mathcal{Z}^n}(\langle \frac{\mathbf{x}_1}{2} + \tau\epsilon_1, ..., \frac{\mathbf{x}_n}{2} + \tau\epsilon_n\rangle)$.
A classifier $C \colon X \to \wp(L)$ is *accurate* on an input $(\mathbf{x}, l_{\mathbf{x}}) \in X \times L$ when $C(\mathbf{x}) = \{l_{\mathbf{x}}\}$. Moreover, C is *stable* over a

region $R \subseteq X$, when $\cup_{\mathbf{w} \in R} C(\mathbf{w}) = \{l\}$ holds, for some $l \in L$. Stability means that a classifier does not change its output on a region of similar inputs, and is an orthogonal notion with respect to accuracy, as it does not require to know the ground truth labels. If a classifier C is both accurate on an input $(\mathbf{x}, l_{\mathbf{x}})$ and stable over a perturbation $P(\mathbf{x})$ of $\mathbf{x}$, then C is *robust* on input $(\mathbf{x}, l_{\mathbf{x}})$ for $P(\mathbf{x})$, i.e., for all $\mathbf{w} \in P(\mathbf{x})$, $C(\mathbf{w}) = \{l_{\mathbf{x}}\}$ holds. Accordingly, stability and robustness metrics for a classifier C on some test set $T \subseteq X \times L$ are defined as the percentage of test samples $\mathbf{x} \in T$ for which C is stable/robust over a perturbation $P(\mathbf{x})$.

Individual Fairness. Our method can be also applied to certify *individual fairness* [8], that intuitively encodes the principle that "two individuals who are similar with respect to a particular task should be classified similarly". The similarity relation on the input space X is expressed in terms of a distance δ and a threshold $\tau > 0$ by considering $S_{\delta,\tau} \triangleq \{(\mathbf{x}, \mathbf{y}) \in X \times X \mid \delta(\mathbf{x}, \mathbf{y}) \leq \tau\}$. Then, given an individual $\mathbf{x} \in X$, a classifier $C : X \to \wp(L)$ is *individually fair* on $\mathbf{x}$ with respect to $S_{\delta,\tau}$ when:

$$\forall \mathbf{y} \in X.\, (\mathbf{x}, \mathbf{y}) \in S_{\delta,\tau} \Rightarrow C(\mathbf{x}) = C(\mathbf{y}).$$

Thus, individual fairness for $\mathbf{x}$ holds if and only if for all $\mathbf{y} \in P^{\tau}_{\delta}(\mathbf{x})$, $C(\mathbf{x}) = C(\mathbf{y})$, where $P^{\tau}_{\delta} : X \to \wp(X)$ is the perturbation defined as $P^{\tau}_{\delta}(\mathbf{x}) \triangleq \{\mathbf{y} \in X \mid \delta(\mathbf{x}, \mathbf{y}) \leq \tau\}$. Hence, by leveraging this simple observation, individual fairness boils down to stability, so that their metrics coincide.

III. Abstract Verification of kNN

Given a classifier $C\colon X \to \wp(L)$, a *sound abstraction* of C on a numerical abstraction $\langle A, \gamma^A \rangle$ is an algorithm $C^A\colon A \to \wp(L)$, which is sound, i.e., for all $a \in A$, $\cup_{\mathbf{x} \in \gamma^A(a)} C(\mathbf{x}) \subseteq C^A(a)$ holds. Thus, soundness means that $C^A(a)$ over-approximates all the output labels of C on inputs abstractly represented by a. If this over-approximation is indeed a singleton then C is *provably stable* over the region $\gamma^A(a)$, i.e., this approach provides a formal *stability certification*.

Theorem III.1 (Abstract Stability Certification). *Let C^A be a sound abstraction of C and assume that a region $R \subseteq X$ is over-approximated by some $a \in A$. If $|C^A(a)| = 1$ then C is stable over R.*

It is worth remarking that the converse of Theorem III.1, in general, does not hold, meaning that this stability certification method can be incomplete. This incompleteness may depend on an input abstract value $a \in A$ which does not represent exactly the adversarial region R or by a loss of precision in the abstract computations of C^A.

A. Abstract Distance

The kNN algorithm relies on a distance $\delta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ for determining the k nearest vectors to a given input sample. Although kNN is parametric on δ, Minkowski distance is the standard choice: given $p \in \mathbb{N}^*$, $\delta_p(\mathbf{x}, \mathbf{y}) \triangleq \sqrt[p]{\sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|^p}$. In particular, the two most common instances are for $p = 1, 2$, namely, Manhattan distance: $\mu(\mathbf{x}, \mathbf{y}) \triangleq \delta_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|$; Euclidean distance: $\eta(\mathbf{x}, \mathbf{y}) \triangleq \delta_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}_i)^2}$. Observe that kNN relies on the distance for relative comparisons only, so we can safely discharge the p-th root $\sqrt[p]{\cdot}$ in δ_p to simplify the computations. We recalled in Section II the definitions of the abstract operations on intervals $\mathcal{I}$ and zonotopes $\mathcal{Z}$. Let us remark that:

(1) We need a sound *abstract dominance relation* to be used for comparing abstract distances, i.e., an algorithm $(\cdot <^A \cdot) : A \times A \to \{\textbf{true}, \textbf{?}\}$ such that if $a_1 <^A a_2 = \textbf{true}$ then for all $x \in \gamma^A(a_1)$ and $y \in \gamma^A(a_2)$, $x < y$ holds. It is worth noticing that the dominance relation $<^{\mathcal{I}}$ for intervals boils down to the so-called interval order, while the relation $<^{\mathcal{Z}}$ for zonotopes may exploit their relational information as encoded by shared noise symbols: e.g., a comparison between zonotopes such as $-2 + 2\epsilon_1 <^{\mathcal{Z}} 1 + \epsilon_1 + \epsilon_2$ reduces to $-3 + \epsilon_1 - \epsilon_2 <^{\mathcal{Z}} 0$, which clearly holds.

(2) A sound and precise enough approximation for zonotopes of the modulus function $|\cdot|$ was not available in literature, hence we designed a novel algorithm for the abstract modulus of zonotopes as described in Section II.

(3) The abstract operations on the product domains $\mathcal{I}^n$ and $\mathcal{Z}^n$ are defined by a straightforward componentwise extension of their unary versions on $\mathcal{I}$ and $\mathcal{Z}$.

It turns out that the abstract Minkowski distance $\delta_p^{\mathcal{I}^n}$, without the p-th root, on intervals does not lose precision, i.e., it is an exact approximation.

Theorem III.2 (Minkowski Distance on Intervals is Exact). *Given $\mathbf{a}, \mathbf{b} \in \mathcal{I}^n$, $\{\delta_p(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \gamma^{\mathcal{I}^n}(\mathbf{a}), \mathbf{y} \in \gamma^{\mathcal{I}^n}(\mathbf{b})\} = \gamma^{\mathcal{I}}(\delta_p^{\mathcal{I}^n}(\mathbf{a}, \mathbf{b}))$, where $\delta_p^{\mathcal{I}}(\mathbf{a}, \mathbf{b}) = (+^{\mathcal{I}})_{i=1}^{n}(|\mathbf{a}_i -^{\mathcal{I}} \mathbf{b}_i|^{\mathcal{I}})^{p^{\mathcal{I}}}$.*

By contrast, we show that the abstract Minkowski distance on zonotopes cannot be guaranteed to be exact: this is expected as the modulus and exponential operations are not linear and, therefore, are necessarily approximated on zonotopes.

Example III.3 (Minkowski Distance on Zonotopes is not Exact). Consider two zonotopes $\hat{a} = 4 + \epsilon_1 + 2\epsilon_2$ and $\hat{b} = 2 + \epsilon_1 + \epsilon_2$, representing some feature in $\mathbb{R}$, that share two noise symbols. Consider the abstract Euclidean distance $\eta^{\mathcal{Z}}(\hat{a}, \hat{b}) = (\hat{a} -^{\mathcal{Z}} \hat{b})^{2^{\mathcal{Z}}}$. Thus, we have that:

$$\begin{aligned}\eta^{\mathcal{Z}}(\hat{a}, \hat{b}) &= \big((4 + \epsilon_1 + 2\epsilon_2) -^{\mathcal{Z}} (2 + \epsilon_1 + \epsilon_2)\big)^{2^{\mathcal{Z}}} = (2 + \epsilon_2)^{2^{\mathcal{Z}}} \\ &= (2 + \epsilon_2) \cdot^{\mathcal{Z}} (2 + \epsilon_2) = \tfrac{9}{2} + 4\epsilon_2 + \tfrac{1}{2}\epsilon_{\text{fresh}}\end{aligned}$$

with $\epsilon_{\text{fresh}} \in [0, 1]$ because this nonlinear noise symbol approximates a square which is always positive (with $\epsilon_{\text{fresh}} \in [-1, 1]$ the approximation would be even worse). Thus, we have that $\gamma^{\mathcal{Z}}(\frac{9}{2} + 4\epsilon_2 + \frac{1}{2}\epsilon_{\text{fresh}}) = [0.5, 9]$. However, observe that the square operation $(2 + \epsilon_2)^{2^{\mathcal{Z}}}$ is sound but not exact, because the range of values of $(2 + \epsilon_2)^2$ is the interval $[1, 3]^2 = [1, 9]$. Thus, $\{\eta(x, y) \in \mathbb{R} \mid x \in \gamma^{\mathcal{Z}}(\hat{a}), y \in \gamma^{\mathcal{Z}}(\hat{b})\} \subsetneq \gamma^{\mathcal{Z}}(\eta^{\mathcal{Z}}(\hat{a}, \hat{b}))$, as $[1, 9] \subsetneq [0.5, 9]$. □

Exactness of the distance function is not enough to achieve completeness of the abstract kNN classifier on intervals, as shown by the following example.

Example III.4 (Incompleteness of Abstract kNN on Intervals). Let us consider a dataset $D = \{(\mathbf{v} = 2, l_1), (\mathbf{w} = 3, l_2)\}$ in the one-dimensional input space $\mathbb{R}$, and the 1NN classifier $C_{\mu,1}$ for the Manhattan distance μ. Consider a region $R = P^1_\infty(0) = \{\mathbf{x} \in \mathbb{R} \mid -1 \leq \mathbf{x} \leq 1\} \in \wp(\mathbb{R})$. The distances of a generic adversarial vector $\mathbf{x} \in R$ from $\mathbf{v}$ and $\mathbf{w}$ are: $\mu(\mathbf{x}, \mathbf{v}) = |\mathbf{x} - 2| = 2 - \mathbf{x}$, and $\mu(\mathbf{x}, \mathbf{w}) = |\mathbf{x} - 3| = 3 - \mathbf{x}$. Hence, the comparison $\mu(\mathbf{x}, \mathbf{v}) <^? \mu(\mathbf{x}, \mathbf{w})$ boils down to $2 - \mathbf{x} <^? 3 - \mathbf{x}$, which always holds. Thus, $\mathbf{v}$ is always the nearest neighbor to R, and, in turn, every sample in R is classified by $C_{\mu,1}$ as l_1, so that stability holds.
Let us perform the abstract stability certification on $\mathcal{I}$, where R is exactly represented by the interval $a \triangleq [-1, 1]$. The abstract Manhattan distances are as follows:

$$\mu^{\mathcal{I}}(a, \mathbf{v}) = |[-1, 1] -^{\mathcal{I}} 2|^{\mathcal{I}} = |[-3, -1]|^{\mathcal{I}} = [1, 3],$$
$$\mu^{\mathcal{I}}(a, \mathbf{w}) = |[-1, 1] -^{\mathcal{I}} 3]|^{\mathcal{I}} = |[-4, -2]|^{\mathcal{I}} = [2, 4].$$

These abstract distances do not allow us to infer the nearest vector to a because $\mu^{\mathcal{I}}(a, \mathbf{v}^{\mathcal{I}}) \not<^{\mathcal{I}} \mu^{\mathcal{I}}(a, \mathbf{w}^{\mathcal{I}})$ and $\mu^{\mathcal{I}}(a, \mathbf{w}^{\mathcal{I}}) \not<^{\mathcal{I}} \mu^{\mathcal{I}}(a, \mathbf{v}^{\mathcal{I}})$. This lack of precision is rooted in the interval abstraction that does not keep track of multiple occurrences of the same $\mathbf{x}$ in different abstract distances. More refined relational abstractions such as octagons or even convex polyhedra [21] would also fail: for instance, with the convex polyhedra abstraction $\mathcal{P}$ we would still have an inconclusive comparison $\mu^{\mathcal{P}}(a, \mathbf{v}) = 1 \leq \mathbf{x} \leq 3 \not<^{\mathcal{P}} 2 \leq \mathbf{x} \leq 4 = \mu^{\mathcal{P}}(a, \mathbf{w})$. On a positive side, the relational information of the zonotope abstraction $\mathcal{Z}$ in this case allows us to prove stability. In fact, the zonotope $\hat{a} \triangleq 0 + \epsilon_1 \in \mathcal{Z}$ exactly represents the region R by keeping track of the dependence on $\mathbf{x}$ through the noise symbol ϵ_1, so we have that:

$$\mu^{\mathcal{Z}}(\hat{a}, \mathbf{v}) = |0 + \epsilon_1 \ -^{\mathcal{Z}} 2|^{\mathcal{Z}} = |-2 + \epsilon_1|^{\mathcal{Z}} = 2 + \epsilon_1,$$
$$\mu^{\mathcal{Z}}(\hat{a}, \mathbf{w}) = |0 + \epsilon_1 \ -^{\mathcal{Z}} 3|^{\mathcal{Z}} = |-3 + \epsilon_1|^{\mathcal{Z}} = 3 + \epsilon_1.$$

Thus, $\mu^{\mathcal{Z}}(\hat{a}, \mathbf{v}) <^{\mathcal{Z}} \mu^{\mathcal{Z}}(\hat{a}, \mathbf{w})$ iff $2 + \epsilon_1 <^{\mathcal{Z}} 3 + \epsilon_1$, which clearly holds. □

The following example shows that even if zonotopes are more precise than intervals, it may happen that intervals prove the stability of some input sample whereas zonotopes fail.

Example III.5 (Intervals vs Zonotopes for Proving Stability). Consider the dataset $D = \{(\mathbf{v} = 0, \ell_1), (\mathbf{w} = 4.1, \ell_2)\}$, a region $R = \{\mathbf{x} \mid 0 \leq \mathbf{x} \leq 2\}$, and the 1NN classifier for the Euclidean distance η (w.l.o.g. we consider the square of η). The region R is exactly represented by the interval $a = [0, 2] \in \mathcal{I}$ and by the zonotope $\hat{a} = 1 + \epsilon_1 \in \mathcal{Z}$. The abstract Euclidean distances turn out to be as follows:

$$\eta^{\mathcal{I}}(a, \mathbf{v}) = ([0, 2] -^{\mathcal{I}} 0)^{2^{\mathcal{I}}} = [0, 4],$$
$$\eta^{\mathcal{I}}(a, \mathbf{w}) = ([0, 2] -^{\mathcal{I}} 4.1)^{2^{\mathcal{I}}} = [4.41, 16.81],$$
$$\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v}) = (1{+}1\epsilon_1 -^{\mathcal{Z}} 0)^{2^{\mathcal{Z}}} = 1 + 2\epsilon_1 + \epsilon_{\mathrm{f}_1} \text{ with } \epsilon_{\mathrm{f}_1} \in [0, 1],$$
$$\eta^{\mathcal{Z}}(\hat{a}, \mathbf{w}) = (1{+}1\epsilon_1 -^{\mathcal{Z}} 4.1)^{2^{\mathcal{Z}}} = 9.61 - 6.2\epsilon_1 + \epsilon_{\mathrm{f}_2} \text{ with } \epsilon_{\mathrm{f}_2} \in [0, 1].$$

Thus, for intervals, we have that $\eta^{\mathcal{I}}(a, \mathbf{v}) <^{\mathcal{I}} \eta^{\mathcal{I}}(a, \mathbf{w})$ iff $[0, 4] <^{\mathcal{I}} [4.41, 16.81]$, which holds and, therefore, entails stability. For zonotopes, $\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v}) <^{\mathcal{Z}} \eta^{\mathcal{Z}}(\hat{a}, \mathbf{w})$ iff $1 + 2\epsilon_1 + \epsilon_{\mathrm{f}_1} <^{\mathcal{Z}} 9.61 - 6.2\epsilon_1 + \epsilon_{\mathrm{f}_2}$ iff $-8.61 + 8.2\epsilon_1 + \epsilon_{\mathrm{f}_1} - \epsilon_{\mathrm{f}_2} <^{\mathcal{Z}} 0$, which does not hold for, e.g., $\epsilon_2 = 1$ and $\epsilon_3 = 0$. Thus, stability cannot be proved with $\mathcal{Z}$. Let us stress that zonotopes here fail because $\mathcal{Z}$ needs to introduce two different fresh nonlinear noise symbols ϵ_{f_1} and ϵ_{f_2} for computing, resp., $\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v})$ and $\eta^{\mathcal{Z}}(\hat{a}, \mathbf{w})$, while both would represent the same square ϵ_1^2. □

Example III.5 arises because zonotopes do not keep track precisely of all nonlinear terms, as for the p-th Minkowski distance in $\mathbb{R}^n$ this would require storing and computing n^p nonlinear terms, thus making abstract computations for practical datasets unfeasible (see [15] for further details on the approximations and practical limitations of zonotopes).

B. Abstract kNN Classification

Given a ground truth dataset D, we describe an algorithm for computing the sound abstract kNN classifier $C^A_{\delta,k}$ on a numerical abstract domain A, which is parametric on a distance function δ, provided that A is endowed with the abstract functions (cf. Section II) to be used for designing a sound abstract distance $\delta^A : A \times A \to A$, where, by a slight abuse of notation, A used in the domain $A \times A$ of δ^A is meant to be an abstraction of sets of vectors in $\wp(\mathbb{R}^n)$ while A used as codomain of δ^A is an abstraction of sets of numbers in $\wp(\mathbb{R})$—in this latter case, for each $a \in A$, we assume that $\mathrm{lb}(a), \mathrm{ub}(a) \in \mathbb{R} \cup \{-\infty, +\infty\}$ provide, resp., a sound lower and upper bound for $\gamma^A(a) \in \wp(\mathbb{R})$. The pseudocode for $C^A_{\delta,k}$ is given as Algorithm 1.

STEP$_1$: Computing and Ordering Abstract Distances. Given a kNN classifier $C_{\delta,k}$, an input $(\mathbf{x}, l_\mathbf{x}) \in X \times L$, and a perturbation function $P : X \to \wp(X)$, we first need a sound abstraction $a_{P(\mathbf{x})} \in A$ for the region $P(\mathbf{x})$, and an abstract representation $\mathbf{y}^A \in A$ for every vector $\mathbf{y}$ occurring in the dataset as $(\mathbf{y}, l_\mathbf{y}) \in D$. For abstract domains that admit an abstraction function $\alpha^A : \wp(\mathbb{R}^n) \to A$, we typically define $a_{P(\mathbf{x})} \triangleq \alpha^A(P(\mathbf{x}))$: this can always be done for intervals where $\alpha^{\mathcal{I}}(S) \triangleq [\inf S, \sup S]$, whereas zonotopes in general do not admit an abstraction function. On the other hand, let us recall that both intervals and zonotopes provide exact abstract representations for ℓ_∞ perturbations $P^\epsilon_\infty(\mathbf{x})$. For each sample $(\mathbf{y}, l_\mathbf{y}) \in D$, we compute its abstract distance $d^A_\mathbf{y} \triangleq \delta^A(a_{P(\mathbf{x})}, \mathbf{y}^A) \in A$ from the abstract value $a_{P(\mathbf{x})}$ representing the perturbation $P(\mathbf{x})$. Each abstract distance is paired with its corresponding label, thus constructing the set of pairs $\{(d^A_\mathbf{y}, l_\mathbf{y})\}_{(\mathbf{y},l_\mathbf{y}) \in D}$. The abstract dominance relation $<^A$ on A is extended to $A \times L$ by disregarding labels, i.e., $(d^A_\mathbf{y}, l_\mathbf{y}) <^{A \times L} (d^A_\mathbf{z}, l_\mathbf{z})$ when $d^A_\mathbf{y} <^A d^A_\mathbf{z}$. This relation $<^{A \times L}$ is weakened by the following total order relation:

$$(d^A_\mathbf{y}, l_\mathbf{y}) \preceq (d^A_\mathbf{z}, l_\mathbf{z}) \overset{\triangle}{\iff}$$
$$\mathrm{lb}(d^A_\mathbf{y}) < \mathrm{lb}(d^A_\mathbf{z}) \text{ or } \big(\mathrm{lb}(d^A_\mathbf{y}) = \mathrm{lb}(d^A_\mathbf{z}) \,\&\, \mathrm{ub}(d^A_\mathbf{y}) \leq \mathrm{ub}(d^A_\mathbf{z})\big).$$

This allows us to sort the set $\{(d^A_\mathbf{y}, l_\mathbf{y})\}_{(\mathbf{y},l_\mathbf{y}) \in D}$ into a totally ordered set $\langle O, \preceq \rangle$ ($\prec$ denotes the corresponding strict order relation). By a slight abuse of notation, we refer to $O[i]$, with $i \in [1, |D|]$, as the i-th smallest element of the total order

$\langle O, \preceq \rangle$, so that $O[1]$ is the smallest element, $O[2]$ the second smallest, and so forth. Firstly, let us observe that $\preceq$ weakens $<^{A\times L}$, because if $O[i] <^{A\times L} O[j]$ holds then $\mathrm{lb}(O[i]) \leq \mathrm{ub}(O[i]) < \mathrm{lb}(O[j])$, so that $O[i] \prec O[j]$ holds, meaning that $i < j$. Moreover, a second property of $\langle O, \preceq \rangle$ is that if $O[j]$ dominates $O[i]$, then any entry $O[k]$ with index $k \geq j$ also dominates $O[i]$, i.e., $O[i] <^{A\times L} O[j]$ implies $\forall k \geq j$, $O[i] <^{A\times L} O[k]$. In fact, $k > j$ implies $O[j] \preceq O[k]$, so that $\mathrm{lb}(O[j]) \leq \mathrm{lb}(O[k])$, and, in turn, since $O[i] <^{A\times L} O[j]$, we have that $\mathrm{ub}(O[i]) < \mathrm{lb}(O[j]) \leq \mathrm{lb}(O[k])$, hence entailing that $O[i] <^{A\times L} O[k]$ holds. In the best case scenario, $\langle O, \preceq \rangle$ may result to be a total order for $<^{A\times L}$, meaning that for all $i, j \in [1, |D|]$, if $i < j$ then $O[i] <^{A\times L} O[j]$. In this optimal case, the abstract computation of the k nearest neighbors of $a_{P(\mathbf{x})}$ boils down to extracting the first k elements from the sequence O. However, in general, $\langle O, \preceq \rangle$ will not be totally ordered for $<^{A\times L}$ because abstract distances may "overlap", as illustrated in Example III.4 for intervals $[1, 3]$ and $[2, 4]$. In our NAVe tool, O has been implemented as a min heap for the total order $\preceq$ (cf. line 6 of Algorithm 1) to leverage its logarithmic cost for building heaps and extracting its i-th smallest element.

STEP$_2$: Computing Score Bounds for Labels. We compute the *abstract score intervals* $\mathsf{s}[l] \in \mathcal{I}$, for all the labels $l \in L$, namely, an integer interval $\mathsf{s}[l] = [\mathrm{lb}(l), \mathrm{ub}(l)]$, with $\mathrm{lb}(l), \mathrm{ub}(l) \in \mathbb{N}$, that provides a lower bound $\mathrm{lb}(l) \geq 0$ and an upper bound $\mathrm{ub}(l) \geq \mathrm{lb}(l)$ to the number of votes that a label l receives from the k nearest neighbors of $a_{P(\mathbf{x})}$. We initialize $\mathsf{s}[l] = [0, 0]$, for each label $l \in L$, then we extract the first k pairs from the indexed sequence O of STEP$_1$. For each extracted pair $(d^A_{\mathbf{z}}, l_{\mathbf{z}})$, we check whether O still includes a pair $(d^A_{\mathbf{y}}, l_{\mathbf{y}})$ having a different label and not dominating $d^A_{\mathbf{z}}$, i.e., such that $l_{\mathbf{y}} \neq l_{\mathbf{z}}$ and $d^A_{\mathbf{z}} \not<^A d^A_{\mathbf{y}}$. If such pair does not exist, then all the pairs $(d^A_{\mathbf{y}}, l_{\mathbf{z}})$ left in O are such that $d^A_{\mathbf{z}} <^A d^A_{\mathbf{y}}$, thus meaning that $l_{\mathbf{z}}$ will certainly get a vote from $\mathbf{z}$, which has been proved to be a k-nearest neighbor of $a_{P(\mathbf{x})}$. If this is the case, it is therefore correct to increase by 1 both the lower and the upper bound of the interval of scores $\mathsf{s}[l_{\mathbf{z}}]$. Otherwise, it is not possible to infer that $l_{\mathbf{z}}$ will certainly get a vote from $\mathbf{z}$, so that the lower bound $\mathrm{lb}(l_{\mathbf{z}})$ cannot be increased, while to preserve the soundness of $\mathsf{s}[l_{\mathbf{z}}]$ we must increase its upper bound $\mathrm{ub}(l_{\mathbf{z}})$ by 1, meaning that it is possible that $l_{\mathbf{z}}$ will get an additional vote from $\mathbf{z}$. After this computation of the score intervals $[\mathrm{lb}(l), \mathrm{ub}(l)]_{l\in L}$ that processed the first k pairs extracted from the sequence O, the sum $\sigma_k \triangleq \sum_{l\in L} \mathrm{lb}(l)$ of the current lower bounds may be less than k, meaning that still no sound inference on the set of most voted labels for kNN can be drawn from the current status of the score intervals. Then, if $\sigma_k < k$ and there exist unprocessed pairs $(d^A_{\mathbf{z}}, l_{\mathbf{z}})$ left in O whose distance $d^A_{\mathbf{z}}$ does not dominate all the distances of the first k pairs extracted from O, then we check whether $\mathrm{ub}(l_{\mathbf{z}}) < k - \sum_{l\in L\setminus\{l_{\mathbf{z}}\}} \mathrm{lb}(l)$ holds. If this is the case then $\mathrm{ub}(l_{\mathbf{z}})$ is increased by 1.

STEP$_3$: Refining Lower Bounds. Following STEP$_2$, we try to refine the lower bounds of $\mathsf{s}[l]$ as sketched by the following example. Let us consider a binary classification with $k = 7$ and two labels l_1 and l_2 whose current score intervals are, resp., $\mathsf{s}[l_1] = [2, 4]$ and $\mathsf{s}[l_2] = [1, 3]$. We observe that this information allows us to make a sound increment of the lower bounds of both l_1 and l_2. In fact, since the sum of the two labels must be $k = 7$, this can happen just when $\mathsf{s}[l_1] = [4, 4]$ and $\mathsf{s}[l_2] = [3, 3]$. Therefore, in this case, we can infer that l_1 is the most voted label. A precise pseudocode of this refinement step is given at lines 17-19 of Algorithm 1.

STEP$_4$: Abstract Classification. After the refinement of STEP$_3$, we return the set of labels whose score intervals are numerically significant, i.e. different from $[0, 0]$, and maximal for the dominance relation $<^{\mathcal{I}}$ between score intervals, that is, $C^A_{\delta,k}(a_{P(\mathbf{x})})$ outputs the following set of labels:

$$\{l \in L \mid \mathrm{ub}(l) \geq \lceil \tfrac{k}{\min(k,|L|)} \rceil,\ \forall l' \neq l \colon \mathsf{s}[l] \not<^{\mathcal{I}} \mathsf{s}[l']\}.$$

Here, we are thus excluding from the output set only those labels l whose score interval either has an upper bound strictly less than $\lceil \frac{k}{\min(k,|L|)} \rceil$ or is not maximal, i.e., there exists a different label l' with a dominant score $\mathsf{s}[l'] >^{\mathcal{I}} \mathsf{s}[l]$, meaning that the number of votes for l is surely less than the votes of l'. This definition is sound because no real classification label given as output by $C_{\delta,k}(\mathbf{y})$ for some adversarial attack $\mathbf{y} \in \gamma^A(a_{P(\mathbf{x})})$ is forgot, while the lack of precision in computing the abstract distances—this cannot happen with intervals but it may be the case of zonotopes, cf. Theorem III.2 and Example III.3—and, in turn, the score intervals may lead to an over-approximation that includes some spurious labels.

Algorithm 1 Pseudocode of $C^A_{\delta,k}(a_{P(\mathbf{x})})$

```
1:  M, O ← ∅                                              ▷ M, O indexing starts at 1
2:  for all (y, l_y) ∈ D do                               ▷ STEP_1
3:      y^A ← α(y)
4:      d^A_y ← δ^A(a_{P(x)}, y^A)
5:      Insert(O, (d^A_y, l_y))
6:  MakeTotalOrder(O, ⪯)                                  ▷ MinHeapify(O, ⪯)
7:  for all l ∈ L do {lb(l) ← 0; ub(l) ← 0}                ▷ STEP_2
8:  for all i ∈ [1, k] do M[i] ← Extract(O[i])
9:  for all (d^A_z, l_z) ∈ M do
10:     ub(l_z)++
11:     if ∀(d^A_y, l_y) ∈ O, l_y = l_z ⇒ d^A_z <^A d^A_y then lb(l_z)++
12: σ_k ← Σ_{l∈L} lb(l)
13: if σ_k < k then
14:     for all (d^A_z, l_z) ∈ O do
15:         if ∃(d^A_y, l_y) ∈ M s.t. l_y ≠ l_z ∧ d^A_z ≮^A d^A_y then
16:             if ub(l_z) < k − (σ_k − lb(l_z)) then ub(l_z)++
17: for all l ∈ L do                                      ▷ STEP_3
18:     μ ← min(k, Σ_{l'≠l} ub(l'))
19:     lb(l) ← max(lb(l), k − μ)
20: for all l ∈ L do                                      ▷ STEP_4
21:     if ub(l) = 0 then L ← L ∖ {l}
22: if (|L| = 1 or k = 1) then return L
23: τ ← ⌈k / min(k,|L|)⌉
24: return {l ∈ L | ub(l) ≥ τ, ∀l' ∈ L ∖ {l}. s[l] ≮^I s[l']}
```

Theorem III.6 (Soundness of Abstract kNN). *The abstract classifier $C^A_{\delta,k}$ is a sound approximation of $C_{\delta,k}$, namely, for all $a \in A$, $\cup_{\mathbf{y}\in\gamma^A(a)} C_{\delta,k}(\mathbf{y}) \subseteq C^A_{\delta,k}(a)$.*

Remarks. In STEP_1, the first k pairs of the total order $\langle O, \preceq \rangle$ are intuitively the k most likely candidates to be the k nearest neighbors of the abstract adversarial region $a_{P(\mathbf{x})}$. If their distances from $a_{P(\mathbf{x})}$ are all strictly dominated by the other pairs in O then these first k samples in O are indeed the k nearest neighbors of $a_{P(\mathbf{x})}$, and therefore we can assign a sure vote to their labels, i.e., we increment both the lower and upper bounds of the score intervals for their labels. If, instead, this is not the case, namely, there exist $O[i] = (d_{\mathbf{z}}, l_{\mathbf{z}})$, for some $i \in [1, k]$, and $O[j] = (d_{\mathbf{y}}, l_{\mathbf{y}})$ with $j > k$, such that $d_{\mathbf{z}} \not<^A d_{\mathbf{y}}$, then we increment the upper bound $\text{ub}(l_{\mathbf{z}})$ just when $l_{\mathbf{z}} \neq l_{\mathbf{y}}$: in fact, if $l_{\mathbf{z}} = l_{\mathbf{y}}$ then neglecting the contribution of the sample $\mathbf{z}$ among the k nearest neighbors does not change the score for that same label $l_{\mathbf{z}}$. Moreover, if some $O[j] = (d_{\mathbf{y}}, l_{\mathbf{y}})$, with $j > k$, strictly dominates all the first k pairs of O, then all the pairs $O[m]$ with $m \geq k$ do the same, so that we do not need to consider them in computing the score intervals. The same reasoning applies to any pair $O[j] = (d_{\mathbf{y}}, l_{\mathbf{y}})$ w.r.t. a generic sample: if there exists some labeled sample $(\mathbf{u}, l_{\mathbf{u}})$ such that $l_{\mathbf{u}} \neq l_{\mathbf{y}}$ and $d_{\mathbf{u}} \not<^A d_{\mathbf{y}}$, then the upper bound $\text{ub}(l_{\mathbf{y}})$ can be safely incremented by 1, as this label $l_{\mathbf{y}}$ could potentially be considered, although we do not know this for sure due to incompleteness. Increasing an upper bound of a score by some positive integer is always sound. However, while the computation of the abstract distance $\delta^A(a_{P(\mathbf{x})}, \mathbf{y}^A)$ may be exact (cf. Theorem III.2), that of the score intervals, in general, is not exact. This is due to the fact that score intervals for labels cannot represent relations between different scores. For example, mutual exclusion is a relational property which cannot be expressed by score intervals: the property "if a label $l_{\mathbf{x}}$ gets n votes, then a different label $l_{\mathbf{y}}$ gets $m - n$ votes" cannot be represented through intervals, that do not keep track of the fact that the score of $l_{\mathbf{y}}$ depends on that of $l_{\mathbf{x}}$.

IV. Dealing with Categorical Features

One-hot encoding is a *de facto* standard strategy for numerical encoding of categorical features that consists in replacing a feature having k categories with k binary numerical features. More precisely, if $F = \{c_1, c_2, ..., c_q\}$ is the set of values for a categorical feature $f \in F$, one-hot encoding replaces f with q binary numerical features $(x_1^f, x_2^f, ..., x_q^f) \in \{0, 1\}^q$ in such a way that $\forall i \in [1, q]$: $x_i^f = 1 \Leftrightarrow f = c_i$. Therefore, one-hot encoding implicitly introduces the constraint $\sum_{i=1}^{q} x_i^f = 1$, which prevents a one-hot encoded sample from having more than one categorical value. If these relational constraints between one-hot encoded numerical features cannot be represented by an abstraction A, then an abstract classifier defined on A may exhibit a significant loss of precision, as illustrated by the following example for intervals.

Example IV.1 (Loss of Precision due to One-Hot Encoding). Consider data samples with a categorical $color \in \{red, green, blue\}$ and a numerical $size \in \mathbb{R}_{\geq 0}$. Let $\mathbf{a}' \triangleq (red, 1)$, $\mathbf{b}' \triangleq (red, 3)$, and consider a dataset $D = \{(\mathbf{a}', l_1), (\mathbf{b}', l_2)\}$. By one-hot encoding, $color$ is replaced by $(isRed, isGreen, isBlue) \in \{0, 1\}^3$, so that $\mathbf{a}'$ and $\mathbf{b}'$ are encoded as $\mathbf{a} \triangleq (1, 0, 0, 1)$ and $\mathbf{b} \triangleq (1, 0, 0, 3)$. Consider an adversarial region $R \triangleq \{(r, g, b, size) \mid r, g, b \in \{0, 1\}, size \in [0, 1]\}$. We observe that $\mathbf{a}$ is always closer than $\mathbf{b}$ to any vector $\mathbf{x} \in R$, for any Minkowski distance δ_p: in fact, we have that

$$\delta_p(\mathbf{a}, \mathbf{x}) < \delta_p(\mathbf{b}, \mathbf{x}) \Leftrightarrow \sqrt[p]{|1 - \mathbf{x}_1|^p + \mathbf{x}_2^p + \mathbf{x}_3^p + |1 - \mathbf{x}_4|^p} < \sqrt[p]{|1 - \mathbf{x}_1|^p + \mathbf{x}_2^p + \mathbf{x}_3^p + |3 - \mathbf{x}_4|^p} \Leftrightarrow |1 - \mathbf{x}_4| < |3 - \mathbf{x}_4|$$

which always holds for $\mathbf{x}_4 = \mathbf{x}_{size} \in [0, 1]$. Hence, 1NN classifies any vector in R as l_1.
Consider the abstract 1NN classifier on the interval abstraction $\mathcal{I}$, as defined in Section III-B, and the Manhattan distance δ_1. Therefore, R is abstracted as $\alpha^{\mathcal{I}^4}(R) = r = \langle [0, 1], [0, 1], [0, 1], [0, 1] \rangle \in \mathcal{I}^4$, and the abstract distances are as follows: $\delta_1^{\mathcal{I}}(r, \mathbf{a}) = [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] = [0, 4]$ and $\delta_1^{\mathcal{I}}(r, \mathbf{b}) = [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [2, 3] = [2, 6]$. Since the intervals $[0, 4]$ and $[2, 6]$ overlap, we cannot infer which of the two samples $\mathbf{a}$ and $\mathbf{b}$ is the nearest to R, so that the abstract 1NN classifier returns $\{l_1, l_2\}$, i.e., no information at all. This loss of precision depends on the interval abstraction, which is not able to represent the constraint $isRed, isGreen, isBlue \in \{0, 1\}$ and $isRed + isGreen + isBlue = 1$. □

This further loss of precision due to one-hot encoding could happen for zonotopes as well, although this phenomenon is mitigated by the chance that zonotopes represent some relational information between different one-hot encoded features through shared noise symbols.

To avoid the loss of precision due to one-hot encoding, we can partition the original adversarial region R, abstractly represented by some $a \in A$, into q subregions $R_i \subseteq R$, each of them abstractly represented by some $a_i \in A$, where q is the overall number of values of the categorical features perturbed in the adversarial region R. Then, we execute the abstract classifier $C^A(a_i)$ for each abstract subregion a_i, and for each of them we therefore compute a sound output set of labels. If, by repeatedly applying $C^A(a_i)$, it happens that the union of their output sets of labels is the whole set L, then we stop and output L. This splitting process will be such that every categorical feature of every subregion R_i will have exactly one possible categorical value, so that within each subregion R_i there is no need for abstracting the one-hot encoded categorical features. The final output will be obtained by collecting all the labels for each a_i, namely: $C^A(a) \triangleq \cup_{i \in [1,q]} C^A(a_i)$. This simple splitting strategy over categorical features reduces false negatives generated by one-hot encoding at the price of a higher certification time, since this procedure generates a new sub-problem for every possible combination of categorical values. Let us remark that if the perturbation of an input sample concerns categorical values only (i.e., numerical values are not perturbated)—this can happen in individual fairness certification—then this partitioning approach boils down to a concrete (and, therefore, trivially exact) verification, at the cost of an exponential number of sub-problems. More precisely, if m is the maximum number of different categories and p is the number of perturbed categorical features then we need

to check $O(m^p)$ sub-problems. This exponential blow-up is somehow expected for an exact stability certification procedure with no false negatives. To balance cost and precision, one could allow only certain features to be split (unsplit features behave as numerical ones, and soundness still holds).

V. Experimental Evaluation

We implemented our abstraction framework for kNN classifiers in Python in a verification tool called NAVe, and we instantiated it with the interval and zonotope abstractions. The source code of NAVe together with datasets and scripts for reproducing our experimental results is available on GitHub [10]. For our experiments, we considered some standard datasets used in robustness certification of kNN [30] and fairness verification of deep neural networks [20] (these fairness datasets are highlighted in grey in the tables). Following [27], the datasets are preprocessed as follows: (1) rows/columns with missing values are dropped; (2) when needed (Letter, Pendigits and Satimage already have explicit test sets), datasets are split into training ($\approx$ 70-80%) and test ($\approx$ 20-30%) sets, resp., D and T; (3) categorical features are one-hot encoded; (4) numerical features are scaled to $[0, 1]$. The details of these datasets, together with the accuracy of kNN on their test sets, are summarized by Table I. In our individual fairness experiments, we consider the Noise-Cat similarity relation as defined by Ruoss et al. [27], where two individuals $\mathbf{x}, \mathbf{y} \in X$ are similar when: (1) given the subset Noise $\subseteq \mathbb{N}$ of indexes of all numerical features and a threshold $\epsilon \geq 0$, for all $i \in$ Noise, $|\mathbf{x}_i - \mathbf{y}_i| \leq \epsilon$; (2) given a subset Cat $\subseteq \mathbb{N}$ of indexes of "sensitive" categorical features, both $\mathbf{x}$ and $\mathbf{y}$ are allowed to have any category for features with indexes in Cat; (3) every other categorical feature of $\mathbf{x}$ and $\mathbf{y}$, i.e. with index not in Cat, must be the same, namely, for any index $i \notin$ (Noise $\cup$ Cat), $\mathbf{x}_i = \mathbf{y}_i$ holds. Fairness experiments with $\epsilon = 0$ represent a pure Cat perturbation to sensitive categorical features only, leaving numerical features unaltered: in this case, our certification method is complete, i.e., the percentages of individual fairness for $\epsilon = 0$ turn out to be exact (i.e., not a lower bound).

We instantiated our parametric abstract kNN classifier of Theorem III.6 to both intervals $\mathcal{I}$ and zonotopes $\mathcal{Z}$, and we evaluated both the Manhattan δ_1 and Euclidean δ_2 distances. We considered the standard ℓ_∞-perturbation P^ϵ_∞ for our stability experiments, with the magnitude ϵ ranging in $[0.001, 0.1]$ for stability experiments ($[0.001, 0.05]$ for the dataset Letter), i.e., numerical features can be altered from $\pm 0.1\%$ to $\pm 10\%$. In the individual fairness experiments, we considered the following Noise-Cat perturbations: for Noise, the numerical attributes were perturbed with P^ϵ_∞ with $\epsilon \in [0, 0.05]$; for Cat, the sensitive categorical attributes were *race* for Compas and *gender* for German; when $\epsilon = 0$, this boils down to a pure Cat perturbation. The parameter k ranges in $\{1, 3, 5, 7\}$, where, following the standard practice for kNN, we avoided even values of k as they are more likely to introduce tie votes in the classification. We conducted all our experiments on a low-cost AWS virtual machine *t2.micro* instance, that provides a baseline level of CPU performance through a single 2.5 GHz CPU and 1 GB of RAM. Throughout the experiments, we mostly observed consistent time behaviours.

TABLE I
Summary of datasets.

Dataset	$\lvert D\rvert$ training	$\lvert T\rvert$ test	#feat.	#feat. with one-hot	#labels	accuracy % $k=1$	$k=3$	$k=5$	$k=7$
Stability									
Australian	483	207	14	39	2	77.8	80.2	82.6	82.6
BreastCancer	479	204	10	10	2	92.6	94.6	93.6	93.6
Diabetes	556	230	8	8	2	70.9	72.2	70.0	71.3
Fourclass	604	258	2	2	2	100	100	100	100
Letter	15000	5000	16	16	26	95.7	94.6	94.2	94.3
Pendigits	7494	3498	16	16	10	97.7	97.8	97.5	97.5
Satimage	4435	2000	36	46	6	88.8	90.3	89.5	90.1
Fairness									
Compas	4222	1056	10	370	2	58.4	59.1	60.2	61.1
German	800	200	20	56	2	73.0	71.5	74.5	77.0

Table III reports the percentages of test samples in T for which our NAVe tool proves that the kNN classifier is stable, i.e., for all k and ϵ, we provide the following metric:

$$\text{ProvableStability}_{k,\epsilon} \triangleq |\{(\mathbf{x}, \_) \in T \mid |C^A_{\delta_i,k}(P^\epsilon_\infty(\mathbf{x}))| = 1\}|/|T|$$

where $A \in \{\mathcal{I}, \mathcal{Z}\}$ and $i = 1, 2$. As shown in Section II, for fairness datasets, provable stability means provable individual fairness. For each distance δ_1 and δ_2, and for each dataset and perturbation magnitude ϵ, we highlight in bold the percentage corresponding to the most provably stable/fair kNN classifier. Due to incompleteness of the abstract kNN classification (cf. Example III.4), it is worth remarking that $\text{ProvableStability}_{k,\epsilon}$ is a lowerbound of the real stability of kNN on the test set T.

As expected, the zonotope abstraction allows us to have a certification technique that is generally more precise, and often much more precise, than that using the intervals, the only exception being the case of German with $\epsilon = 0.02$ for $k = 1$ where intervals infer one more stable sample than zonotopes (overall, 85% vs 84.5% of provable stability; this may happen as shown in Example III.5). Our NAVe tool infers with the zonotope abstraction more than 80% of stability, independently of k and of distance δ_i, of: (i) Australian for all $\epsilon \leq 0.1$; (ii) BreastCancer for all $\epsilon \leq 0.05$; (iii) Fourclass and Pendigits for all $\epsilon \leq 0.03$; (iv) Diabetes, Letter and Satimages for $\epsilon \leq 0.005$. Of course, provable stability decreases with higher values of ϵ since stronger perturbations are more likely to produce unstable behaviours, as well as more false positives among the approximate output sets of labels. In particular, we observe that Diabetes exhibits the worst stability scores, that together with a low accuracy ($\approx 70\%$) hints that a diagnosis of diabetes may be a hard task for which kNN does not perform well. On the other hand, the provable stability of Letter seems to be negatively affected by the size of its training set D, as more samples and more features are more likely to introduce ties between abstract distances. The fairness experiments show that kNN predictions on German are rather fair on the *gender* attribute (average for all k with $\epsilon = 0$ is 83.8%), while on Compas are rather unfair on the *race* category (average for all k with $\epsilon = 0$ is 64.7%).

Table II shows the average certification time, in seconds, per input sample $\mathbf{x}$ and per magnitude ϵ. This is computed as the

TABLE II
AVERAGE CERTIFICATION TIME PER SAMPLE IN SECONDS.

	Intervals $\mathcal{I}$		Zonotopes $\mathcal{Z}$	
Dataset	δ_1 (s)	δ_2 (s)	δ_1 (s)	δ_2 (s)
Australian	0.01	0.01	0.11	0.22
BreastCancer	0.01	0.01	0.06	0.05
Diabetes	0.11	0.07	0.55	0.58
Fourclass	0.04	0.04	0.35	0.40
Letter	5.44	4.97	21.89	22.64
Pendigits	0.26	0.57	9.99	9.70
Satimage	0.33	2.90	11.91	4.29
Compas	15.30	18.48	140.82	239.99
German	0.75	0.74	5.08	7.67

average time for executing NAVe for all $k \in \{1, 3, 5, 7\}$ on a given input sample (i.e., average on the whole test set T) and for a given magnitude ϵ (i.e., average on the 8 magnitudes ϵ). Our certification technique turns out to be quite fast, where the peak average time of about 4 minutes is reached for certifying the individual fairness of Compas samples with Euclidean distance through zonotopes, very likely due to one-hot encoding that explodes the number of features from 10 to 370.

VI. CONCLUSION

We have shown how to design an abstract interpretation of k-nearest neighbor classifiers and how this technique defines, to the best of our knowledge, the first robustness certification framework for this popular ML algorithm. We implemented and experimentally evaluated our verification technique. The experiments show that our approach is effective and precise, and that kNN classification is generally robust for numerical perturbations less than $\pm 3\%$. As future work, we plan to design a new numerical abstraction that can precisely track the role of different features when comparing abstract distances between two samples. Ideally, we would aim to achieve a *complete stability certification* of kNN.

REFERENCES

[1] A. Albarghouthi. Introduction to neural network verification. *Found. Trends Program. Lang.*, 7(1-2):1–157, 2021.

[2] L. Amsaleg, J. Bailey, A. Barbe, S. M. Erfani, T. Furon, M. E. Houle, M. Radovanovic, and X. V. Nguyen. High intrinsic dimensionality facilitates adversarial attack: Theoretical evidence. *IEEE Trans. Inf. Forensics Secur.*, 16:854–865, 2021.

[3] S. Calzavara, P. Ferrara, and C. Lucchese. Certifying decision trees against evasion attacks by program analysis. In *Proc. 25th European Symp. on Research in Computer Security, ESORICS 2020*, volume 12309 of *LNCS*, pages 421–438. Springer, 2020.

[4] N. Carlini and D. A. Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE Symposium on Security and Privacy, IEEE S&P*, pages 39–57, 2017.

[5] P. Cousot. *Principles of Abstract Interpretation*. MIT Press, 2021.

[6] P. Cousot and R. Cousot. 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, 1977.

[7] L. H. de Figueiredo and J. Stolfi. Affine arithmetic: Concepts and applications. *Numer. Algorithms*, 37(1-4):147–158, 2004.

[8] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel. Fairness Through Awareness. In *Proc. 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.

[9] A. Z. Fan and P. Koutris. Certifiable robustness for nearest neighbor classifiers. In D. Olteanu and N. Vortmeier, editors, *Proc. 25th Int. Conf. on Database Theory, ICDT 2022*, LIPIcs vol. 220, pages 6:1–6:20, 2022.

[10] N. Fassina, F. Ranzato, and M. Zanella. NAVe: kNN Abstract Verifier. https://github.com/abstract-machine-learning/NAVe, 2023.

[11] T. Gehr, M. Mirman, D. Drachsler-Cohen, P. Tsankov, S. Chaudhuri, and M. T. Vechev. AI2: Safety and robustness certification of neural networks with abstract interpretation. In *Proc. 2018 IEEE Symposium on Security and Privacy, IEEE S&P 2018*, pages 3–18, 2018.

[12] K. Ghorbal, E. Goubault, and S. Putot. The zonotope abstract domain Taylor1+. In *Proc. 21st Int. Conf. on Automated Verification, CAV 2009*, LNCS vol. 5643, pages 627–633. Springer, 2009.

[13] R. Giacobazzi and F. Ranzato. History of abstract interpretation. *IEEE Ann. Hist. Comput.*, 44(2):33–43, 2022.

[14] I. Goodfellow, P. McDaniel, and N. Papernot. Making machine learning robust against adversarial inputs. *Commun. ACM*, 61(7):56–66, 2018.

[15] E. Goubault and S. Putot. A zonotopic framework for functional abstractions. *Formal Methods Syst. Des.*, 47(3):302–360, 2015.

[16] J. Jia, Y. Liu, X. Cao, and N. Z. Gong. Certified robustness of nearest neighbors against data poisoning and backdoor attacks. In *Proc. 36th AAAI Conf. on Artificial Intelligence*, pages 9575–9583, 2022.

[17] Y. Li, J. Wang, and C. Wang. Proving robustness of KNN against adversarial data poisoning. In *Proc. 22nd Int. Conf. on Formal Methods in Computer-Aided Design, FMCAD 2022*, pages 7–16. IEEE, 2022.

[18] C. Liu, T. Arnon, C. Lazarus, C. A. Strong, C. W. Barrett, and M. J. Kochenderfer. Algorithms for verifying deep neural networks. *Found. Trends Optim.*, 4(3-4):244–404, 2021.

[19] Y. Liu, J. Peng, L. Chen, and Z. Zheng. Abstract interpretation based robustness certification for graph convolutional networks. In *Proc. 24th Eur. Conf. on Artificial Intelligence, ECAI 2020*, pages 1309–1315, 2020.

[20] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2022.

[21] A. Miné. Tutorial on static inference of numeric invariants by abstract interpretation. *Found. Trends Program. Lang.*, 4(3-4):120–372, 2017.

[22] M. Mirman, T. Gehr, and M. T. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *Proc. 35th Int. Conf. on Machine Learning, ICML 2018*, pages 3575–3583, 2018.

[23] F. Ranzato, C. Urban, and M. Zanella. Fairness-aware training of decision trees by abstract interpretation. In *Proc. 30th ACM Int. Conf. on Information and Knowledge Manag., CIKM*, pages 1508–1517, 2021.

[24] F. Ranzato and M. Zanella. Robustness Verification of Support Vector Machines. In *Proc. 26th International Static Analysis Symposium, SAS 2019*, LNCS vol. 11822, pages 271–295, 2019.

[25] F. Ranzato and M. Zanella. Abstract interpretation of decision tree ensemble classifiers. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, pages 5478–5486, 2020.

[26] F. Ranzato and M. Zanella. Genetic adversarial training of decision trees. In *Proceedings of the 2021 Genetic and Evolutionary Computation Conference, GECCO 2021*, pages 358–367. ACM, 2021.

[27] A. Ruoss, M. Balunovic, M. Fischer, and M. Vechev. Learning certified individually fair representations. In *Proc. 34th Ann. Conf. on Advances in Neural Information Processing Systems, NeurIPS 2020*, 2020.

[28] G. Singh, T. Gehr, M. Püschel, and M. Vechev. An abstract domain for certifying neural networks. *Proc. ACM Program. Lang.*, 3(POPL 2019):41:1–41:30, Jan. 2019.

[29] G. Singh, T. Gehr, M. Püschel, and M. T. Vechev. Boosting robustness certification of neural networks. In *Proc. of the 7th Int. Conf. on Learning Representations, ICLR*, 2019.

[30] C. Sitawarin, E. M. Kornaropoulos, D. Song, and D. A. Wagner. Adversarial examples for k-nearest neighbor classifiers based on higher-order Voronoi diagrams. In *Proc. Annual Conference on Neural Information Processing Systems, NeurIPS*, pages 15486–15497, 2021.

[31] C. Sitawarin and D. A. Wagner. Minimum-norm adversarial examples on KNN and KNN based models. In *Proc. IEEE Security and Privacy Workshops, SP Workshops, 2020*, pages 34–40. IEEE, 2020.

[32] L. Wang, X. Liu, J. Yi, Z.-H. Zhou, and C.-J. Hsieh. Evaluating the robustness of nearest neighbor classifiers: A primal-dual perspective. *ArXiv*, abs/1906.03972, 2019.

[33] Y. Wang, S. Jha, and K. Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proc. 35th Int. Conf. on Machine Learning, ICML*, volume 80 of *PMLR*, pages 5120–5129, 2018.

[34] Y. Yang, C. Rashtchian, Y. Wang, and K. Chaudhuri. Robustness for non-parametric classification: A generic attack and defense. In *Proc. 23rd Int. Conf. on Artificial Intelligence and Statistics, AISTATS*, PMLR vol. 108, pages 941–951, 2020.

TABLE III

PERCENTAGES OF PROVABLE STABILITY OR INDIVIDUAL FAIRNESS FOR **INTERVALS** $\mathcal{I}$ WITH **MANHATTAN** DISTANCE δ_1 ON THE WHOLE TEST SETS T.

	Australian				BreastCancer				Diabetes				Fourclass				Letter			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	98.5	99.5	99.0	99.0	**100**	98.5	99.5	**100**	93.4	94.3	93.4	95.2	**100**	99.6	**100**	**100**	**96.9**	94.4	93.4	93.5
0.005	96.6	96.1	93.2	95.1	97.5	98.5	98.5	97.5	73.4	69.1	63.9	65.2	99.6	99.6	99.2	99.2	90.5	86.7	83.6	81.2
0.01	92.2	93.2	91.3	94.2	93.6	95.5	97.0	96.5	45.2	39.1	36.9	34.3	99.2	98.8	96.1	95.7	75.3	67.2	60.8	56.1
0.02	88.8	88.4	86.9	89.8	85.7	86.2	86.7	88.7	14.3	12.6	10.8	9.13	87.6	86.0	81.7	81.0	40.2	32.6	27.9	25.0
0.03	84.0	83.0	85.9	85.5	78.9	83.3	84.8	86.2	4.7	3.0	1.7	1.3	70.1	68.6	67.0	62.4	15.4	12.0	9.84	8.60
0.05	79.7	80.1	83.5	82.6	66.6	68.6	75.4	75.0	0.8	0.8	0	0	34.1	34.5	29.8	28.6	1.2	1.1	1.1	1.0
0.07	78.2	78.2	77.7	78.2	34.8	45.1	51.4	58.8	0.4	0	0	0	15.1	15.5	14.3	13.5	–	–	–	–
0.10	69.0	64.2	65.7	66.6	5.3	5.8	6.3	19.1	0	0	0	0	5.4	5.0	5.0	4.2	–	–	–	–

	Pendigits				Satimages					Compas				German			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	99.5	99.1	99.2	99.1	93.6	91.8	92.0	91.7	0	56.4	62.0	68.0	**70.5**	**85.5**	83.5	82.5	85.0
0.005	96.5	96.1	95.8	95.4	68.3	65.5	64.2	63.3	0.001	46.5	52.3	57.7	60.0	84.5	83.5	81.5	84.5
0.01	92.1	91.8	91.1	90.7	44.4	43.5	43.7	43.9	0.002	40.5	45.9	49.3	51.7	84.0	82.5	81.0	84.0
0.02	79.0	78.4	77.9	77.5	20.8	19.9	20.0	20.4	0.005	26.3	31.8	33.7	36.3	83.5	79.5	80.5	82.0
0.03	62.6	63.6	63.2	63.0	12.1	12.1	11.8	12.2	0.01	16.9	20.8	22.7	26.0	82.0	76.0	78.0	80.0
0.05	28.3	29.5	29.2	28.6	8.4	8.4	8.4	8.4	0.02	11.0	13.7	14.5	16.9	78.0	74.0	74.0	73.0
0.07	8.1	8.9	9.1	9.1	6.3	6.4	6.2	6.4	0.03	9.1	10.6	11.5	13.6	73.5	69.5	70.0	68.0
0.10	0.2	0.1	0.06	0.03	2.9	3.0	2.9	2.9	0.05	5.8	7.0	7.4	8.9	67.5	62.5	59.5	58.0

PERCENTAGES OF PROVABLE STABILITY OR INDIVIDUAL FAIRNESS FOR **ZONOTOPES** $\mathcal{Z}$ WITH **MANHATTAN** DISTANCE δ_1 ON THE WHOLE TEST SETS T.

	Australian				BreastCancer				Diabetes				Fourclass				Letter			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	99.5	99.5	99.5	**100**	99.0	99.5	**100**	95.2	96.5	96.5	96.0	**100**	99.6	**100**	**100**	**96.9**	94.4	93.4	93.5
0.005	**98.5**	97.5	94.6	95.6	98.0	98.5	99.0	**99.5**	**85.2**	82.1	83.4	83.9	99.6	**100**	**100**	99.6	**92.3**	89.7	88.2	87.7
0.01	**94.6**	**94.6**	92.7	**94.6**	96.0	96.5	**98.5**	97.5	72.1	**73.0**	**73.0**	72.1	**99.6**	**99.6**	**99.6**	99.2	**84.9**	82.2	81.0	79.6
0.02	**93.7**	90.8	91.3	92.2	91.6	92.6	**97.0**	94.6	60.4	56.5	64.3	**65.6**	94.5	96.5	97.6	**98.0**	**66.7**	64.1	63.9	61.7
0.03	91.3	87.9	88.4	**92.2**	89.7	91.1	92.6	**94.6**	50.4	54.7	**57.3**	52.6	90.3	89.9	**91.0**	89.9	**54.0**	52.6	52.8	51.7
0.05	85.0	85.0	**89.8**	87.9	88.2	92.1	92.1	**94.6**	23.9	28.7	28.7	**29.5**	74.0	79.4	80.6	**84.5**	**37.7**	28.3	26.8	25.9
0.07	81.6	81.6	85.9	**86.9**	78.9	84.8	87.7	**89.7**	10.4	7.3	**20.4**	10.4	**67.4**	58.5	59.6	58.1	–	–	–	–
0.10	84.0	85.5	85.5	**86.9**	73.5	**83.8**	78.4	79.9	3.4	0.4	**19.5**	6.9	37.9	37.6	46.9	**51.9**	–	–	–	–

	Pendigits				Satimages					Compas				German			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**99.6**	99.2	99.3	99.2	**95.5**	93.8	94.7	**95.5**	0	56.4	62.0	68.0	**70.5**	**85.5**	83.5	82.5	85.0
0.005	**98.4**	98.0	98.3	97.9	84.8	84.7	84.5	**85.2**	0.001	47.9	53.6	58.7	**61.9**	**85.0**	83.5	81.5	84.5
0.01	96.4	96.8	**97.1**	96.8	75.4	77.6	78.9	**81.1**	0.002	44.0	49.5	55.3	**56.8**	**85.0**	82.5	81.5	84.0
0.02	92.7	**93.9**	93.7	93.4	74.9	75.9	79.5	**80.2**	0.005	33.2	38.5	43.7	**47.1**	**84.0**	81.0	81.0	83.5
0.03	88.4	91.5	91.4	**91.9**	66.7	68.3	70.8	**73.3**	0.01	25.1	30.2	31.9	**37.1**	**82.5**	76.5	78.5	82.0
0.05	77.2	83.4	**84.1**	83.7	55.6	57.7	57.5	**60.1**	0.02	17.6	21.5	24.5	**27.1**	**80.0**	76.5	76.0	78.5
0.07	61.8	67.7	**68.4**	**68.4**	49.7	41.0	42.7	**51.5**	0.03	13.5	17.2	19.3	**21.3**	**75.0**	**75.0**	73.0	74.0
0.10	38.4	41.2	49.0	**52.5**	**41.8**	35.0	35.7	35.9	0.05	9.1	12.9	16.1	**16.8**	**70.5**	69.5	63.5	69.0

PERCENTAGES OF PROVABLE STABILITY OR INDIVIDUAL FAIRNESS FOR **INTERVALS** $\mathcal{I}$ WITH **EUCLIDEAN** DISTANCE δ_2 ON THE WHOLE TEST SETS T.

	Australian				BreastCancer				Diabetes				Fourclass				Letter			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	**100**	**100**	**100**	99.5	**100**	99.0	99.5	95.2	96.0	**98.7**	**98.7**	**100**	**100**	**100**	**100**	**97.9**	96.3	95.2	95.6
0.005	97.1	98.5	97.5	**99.0**	96.5	**99.5**	98.0	98.5	78.7	80.4	80.0	77.3	99.6	99.6	99.2	98.8	91.7	88.6	85.4	83.0
0.01	95.1	96.6	96.1	97.1	96.5	99.0	96.5	97.0	62.6	59.5	56.5	54.3	99.2	**99.6**	98.4	97.6	82.6	75.1	69.4	65.2
0.02	92.7	91.3	91.7	94.6	92.1	93.1	91.6	93.1	31.3	25.6	23.9	23.0	93.4	91.4	86.8	85.2	54.6	45.3	40.2	36.9
0.03	91.3	88.8	89.3	92.2	86.7	88.7	87.7	88.2	13.0	9.13	7.83	6.09	78.2	75.5	75.5	72.8	31.0	24.8	21.2	19.2
0.05	85.9	84.0	85.5	85.9	75.0	77.9	81.3	83.8	3.0	1.7	0.8	0.8	44.9	42.6	40.3	37.6	7.7	6.6	5.7	5.2
0.07	84.5	82.1	85.5	83.5	65.2	65.6	67.6	74.5	1.3	1.3	0	0	24.0	23.6	22.8	19.3	–	–	–	–
0.10	82.1	80.1	85.0	83.0	37.7	38.7	42.6	52.9	0.8	0.8	0	0	9.30	8.53	8.14	8.14	–	–	–	–

	Pendigits				Satimages					Compas				German			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	99.6	99.4	99.4	99.6	93.8	93.8	93.7	94.3	0	58.2	62.6	69.1	**71.0**	**85.0**	83.0	83.5	82.5
0.005	98.3	98.0	98.1	97.8	73.7	72.8	72.4	72.3	0.001	48.2	54.5	61.1	62.4	**85.0**	83.0	83.0	82.0
0.01	96.6	96.1	95.7	95.0	56.0	54.9	54.0	54.2	0.002	45.5	50.9	56.7	57.6	**85.0**	82.5	82.5	82.0
0.02	91.4	90.5	89.5	89.0	31.4	31.0	31.8	32.7	0.005	35.2	41.4	45.4	46.0	**85.0**	81.5	82.5	81.5
0.03	82.0	81.9	81.0	79.9	19.3	18.5	18.3	18.6	0.01	25.6	30.7	32.8	36.4	83.0	78.0	81.0	81.5
0.05	58.9	60.6	59.7	58.9	9.6	9.6	9.6	9.8	0.02	17.8	20.8	23.8	27.4	80.0	75.5	78.0	78.5
0.07	36.3	38.4	37.5	36.8	8.2	8.1	7.9	7.8	0.03	13.5	16.6	19.5	21.7	76.0	72.5	75.0	73.5
0.10	13.5	13.4	13.2	13.2	4.7	4.6	4.6	4.5	0.05	10.2	12.7	14.3	16.4	72.0	68.5	69.0	69.5

PERCENTAGES OF PROVABLE STABILITY OR INDIVIDUAL FAIRNESS FOR **ZONOTOPES** $\mathcal{I}$ WITH **EUCLIDEAN** DISTANCE δ_2 ON THE WHOLE TEST SETS T.

	Australian				BreastCancer				Diabetes				Fourclass				Letter			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	**100**	**100**	**100**	99.5	**100**	99.0	99.5	96.5	96.0	**98.7**	**98.7**	**100**	**100**	**100**	**100**	**97.9**	96.3	95.2	95.7
0.005	98.0	**99.0**	98.5	**99.0**	97.5	**99.5**	99.0	99.0	85.2	87.8	**88.2**	87.8	99.6	99.6	**100**	**100**	**94.3**	92.1	91.0	90.1
0.01	97.1	97.1	96.6	**98.0**	96.5	98.5	**99.0**	98.0	79.1	**81.3**	77.3	79.1	99.2	**99.6**	**99.6**	99.2	**88.0**	86.0	84.2	82.8
0.02	95.1	92.2	95.1	**96.6**	94.1	95.1	94.6	**96.5**	66.5	**66.9**	65.2	64.7	90.7	94.5	94.5	**96.5**	**68.3**	66.6	65.0	63.3
0.03	93.7	89.3	93.7	**94.6**	92.6	**97.0**	95.1	94.6	50.0	52.6	55.2	**57.3**	81.4	83.3	84.5	**86.0**	**51.4**	51.3	50.5	49.1
0.05	89.3	87.4	**89.8**	88.8	89.2	93.1	92.6	**95.5**	25.6	30.0	31.7	**38.7**	56.2	64.7	**69.7**	68.9	**28.7**	27.8	27.1	25.8
0.07	85.0	83.0	**87.9**	85.5	77.4	86.7	91.6	**95.1**	10.8	8.2	**11.3**	10.0	36.0	37.2	44.1	**49.2**	–	–	–	–
0.10	82.6	80.1	**85.5**	84.5	66.6	75.9	75.9	**78.9**	4.3	1.7	**14.3**	2.1	19.7	22.0	31.0	**32.1**	–	–	–	–

	Pendigits				Satimages					Compas				German			
ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	ϵ	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	99.6	99.6	99.4	**99.7**	95.4	95.0	95.2	**96.0**	0	58.2	62.6	69.1	**71.0**	**85.0**	83.0	83.5	82.5
0.005	**98.8**	98.5	98.6	98.7	84.8	85.1	85.5	**86.8**	0.001	49.0	55.5	61.9	**63.4**	**85.0**	83.0	83.0	82.5
0.01	97.6	97.8	**97.9**	97.8	76.1	77.9	78.7	**79.4**	0.002	46.8	52.6	58.7	**59.9**	**85.0**	83.0	82.5	82.0
0.02	94.9	95.4	95.4	**95.5**	68.2	70.2	74.4	**76.0**	0.005	38.2	45.1	49.6	**52.4**	84.5	83.0	82.5	82.0
0.03	91.1	92.7	93.0	**93.1**	60.4	63.0	64.9	**67.8**	0.01	29.8	36.4	40.8	**45.8**	**83.5**	80.0	81.0	81.5
0.05	79.7	85.3	**86.7**	85.8	44.4	49.4	52.6	**56.9**	0.02	22.9	28.3	32.3	**35.4**	79.5	77.5	78.5	**80.5**
0.07	62.6	71.1	**73.7**	73.5	25.5	31.6	34.8	**42.5**	0.03	17.9	23.1	25.7	**28.6**	76.0	75.0	**77.5**	**77.5**
0.10	38.7	48.3	51.3	**53.1**	9.7	26.6	24.9	**36.2**	0.05	12.3	16.7	18.7	**22.0**	72.5	**74.0**	**74.0**	73.0