UDC 004.8
IRSTI 28.23.02

https://doi.org/10.55452/1998-6688-2026-23-2-312-325

[1]*Beishekeyev A.,
Master's student, ORCID ID: 0009-0006-0845-4551,
*e-mail: a.beishekeyev@kbtu.kz
[2]Umarov T.,
PhD, ORCID ID: 0009-0008-0044-7159,
e-mail: t.umarov@bmu.edu.kz

[1]Kazakh-British Technical University, Almaty, Kazakhstan
[2]British Management University, Tashkent, Uzbekistan

FORMAL VERIFICATION OF DECISION TREE FAIRNESS AND ROBUSTNESS VIA SMT SOLVER

Abstract

As Artificial Intelligence systems are increasingly deployed in safety-critical domains such as healthcare and finance, ensuring their trustworthiness and compliance is paramount. While Deep Neural Networks have received significant attention in formal verification, traditional models such as Decision Trees, often preferred for their interpretability, cannot inherently enforce constraints after training for fairness and stability. This paper presents a novel, comprehensive approach for the formal verification of Decision Tree classifiers using Satisfiability Modulo Theories (SMT). We propose a robust translation scheme that converts trained decision trees into logical constraints, enabling constraint inference that guarantees demographic parity and local robustness at prediction time. We implement this framework by using the z3 SMT solver and validate it on widely recognized fairness benchmarks, including UCI Adult, German Credit, and Loan Approval datasets. Experimental results demonstrate that our constrained model effectively eliminates demographic parity violations with a marginal accuracy trade-off of less than 0.2%. This approach transforms the SMT solver from a simple diagnostic tool into a provably fair inference engine suitable for regulated industries.

Keywords: Machine Learning, Decision Trees, Formal Methods, SMT.

Received April 7, 2026; accepted June 7, 2026.

Introduction

Artificial intelligence has achieved widespread adoption through advances in deep learning and machine learning. However, as these models are increasingly deployed in safety-critical domains such as healthcare and finance, the reliability of their outputs is paramount. Without appropriate intervention during training or evaluation, machine learning models are susceptible to biases against specific demographic groups and may exhibit instability when exposed to outliers [1].

While traditional validation techniques, such as empirical testing on hold-out datasets, provide a statistical measure of performance, they fail to offer guarantees of safety. To address this, formal verification provides mathematical proofs ensuring that a model's behaviour satisfies defined constraints for all possible inputs.

We adopt the concept of robustness defined in [2], where robustness was defined as relative stability to a specific value, meaning that for a model to be robust, it must maintain its performance even when certain conditions change.

Despite the growing importance of verification, current research is disproportionately focused on deep neural networks. Significant gaps remain in verifying traditional models like Decision Trees, particularly throughout the machine learning pipeline. Existing approaches often neglect the data

preprocessing stage or suffer from scalability constraints, such as the quadratic scaling observed in Coloured Petri-Net verification [3]. Furthermore, current verification methods typically prioritise local robustness over global properties, providing limited support for critical fairness metrics.

Although decision trees are traditionally viewed as "white-box" models, their interpretability does not equate to formal provability. In safety-critical or highly regulated sectors (e.g., Finance, GDPR-compliant systems), it is insufficient for a model to be "generally fair", it must be provably compliant with specific logical constraints.

Current verification methods for DTs face a "verification gap":

- Inductive Bias: Standard DT algorithms (CART/ID3) optimise for statistical heuristics like Gini Impurity, which can inadvertently encode proxies for protected attributes (e.g., race or gender) hidden in continuous data.
- Scalability of Constraints: Existing formal methods like Binary Decision Diagrams (BDDs) suffer from exponential memory growth when dealing with continuous variables, making them impractical for trees with high-cardinality features.
- The Enforcement Problem: There is currently no unified framework that allows a developer to query a decision tree for edge-case violations (Individual Fairness) or to prune biased branches without destroying the model's global structure.

The scientific novelty of this research lies in transitioning from passive model auditing to active prediction-time enforcement within an SMT-driven logic space. Unlike previous methods [3] that focus on post-hoc rule validation or monitoring, our framework treats the Decision Tree as a set of dynamic constraints that are solved at inference time. Specifically, our novelty is threefold:

- a translation scheme that maps continuous feature boundaries into the SMT Theory of Real Arithmetic, bypassing the state-explosion limits of BDDs
- the simultaneous enforcement of global fairness (Demographic Parity) and local robustness; and
- a prediction-time correction mechanism (Algorithm 2) that identifies and resolves potential violations before an output is generated.

This transforms the SMT solver from a diagnostic tool into a provably fair inference engine.

Materials and methods

Current research on the formal verification of machine learning models can be categorised into two groups: traditional machine learning approaches [3,4] and neural network verification. The most recent work has focused on verifying the robustness of neural networks [5–11].

SMT-based methods represent a prominent verification approach. [7] developed an SMT-solver for ReLU activation function verification, combining three components: a simplex algorithm for core functionality, a ReLUplex engine for search management, and an SMT-solver for constraint splitting. For data verification, [12] employed SMT solvers to validate sensor data integrity through controlled manipulation experiments.

Runtime monitoring approaches have emerged as complementary verification strategies. [8] proposed neural network monitoring using binary decision trees and Hamming distance metrics, storing activation patterns from the final layers of the network during training for runtime comparison.

Traditional machine learning verification faces unique challenges. [3] implemented Coloured Petri-Nets for decision tree rule verification, while [13] combined temporal logic with particle swarm optimisation to verify multi-layer perceptrons through reachability analysis.

Activation function verification remains an active research area. [6] transformed Swish function verification into constraint satisfaction problems, demonstrating advantages over ReLU in classification tasks. [5] developed the MSVL language for temporal logic verification of neural networks, achieving parity with PyTorch implementations.

[14] introduces the verification and model-based formal approach that transforms neural network models into UPPAAL Timed Automata. This study is used for the formal verification of neural dynamics.

Abstraction-refinement for the verification of machine learning mainly focused on neural networks, but the [15] study applied it to Fuzzy Decision Trees. The abstraction applies constraints to lower and upper bound splits using minimum and maximum split values. Refinement means reducing errors through domain splitting and optimisation.

Current formal verification methods exhibit three principal limitations. First, most approaches focus on post-training model verification, neglecting critical data preparation and feature engineering stages [16]. Second, existing techniques face scalability constraints: BDD-based methods support only hundreds of variables, while CP-Net verification scales quadratically with decision tree nodes [3]. Third, verification currently prioritises local robustness over global model properties, with limited support for temporal constraints or fairness metrics.

Implementation challenges persist across verification paradigms. CNN verification requires solving complex optimisation problems that mirror training computational intensity [5]. Decision tree verification using CP-Nets demonstrates limited generalisability, having only been validated on single-institution educational data [3].

Table 1 – Limitations of Existing Approaches

Feature	CP-Nets / Petri-Nets [3]	BDD-Based Methods [8]	This Work (SMT-Based)
Scalability	Quadratic with node count.	Exponential with continuous variables.	Path extraction linear in node count; solver complexity NP-complete but manageable for depth ≤ 5
Variable Support	Discrete/Categorical focus.	Memory-intensive for high-cardinality.	Native Real Arithmetic for continuous data.
Constraint Type	Primarily rule verification.	Local robustness/activation patterns.	Global Fairness + Local Robustness.

While SMT solvers have been applied to neural networks to handle ReLU activations [7], their application to Decision Trees has been largely overlooked in favour of simpler heuristic checks. This work distinguishes itself by treating the Decision Tree not as a series of if-else statements, but as a disjoint union of polyhedra. This geometric interpretation allows us to apply SMT solvers to find 'counterexamples' that represent unfair or non-robust regions of the feature space that empirical testing would likely miss.

As noted by [16], the machine learning pipeline's initial stages represent critical verification gaps: "The first stages of the machine learning process (including data preparation) may be considered as the most fragile steps of the whole computation procedure [...] these initial steps are generally considerably neglected in terms of verification." Key unresolved challenges include:

- Data provenance verification and duplication detection techniques
- Global robustness verification beyond local adversarial examples
- Versatile verification frameworks supporting diverse model architectures

Current methods additionally suffer from exponential time complexity in SMT/MILP solvers and memory-intensive runtime monitoring requirements.

This work distinguishes itself from existing SMT-based verification approaches in three key aspects. First, unlike [7, 12] which focus on neural networks or sensor data, we provide a direct translation of decision tree paths into SMT-LIB formulas that handles continuous features natively using the theory of real arithmetic, avoiding the state explosion of binary decision diagrams (BDDs) used in [8]. Second, while most verification methods address either robustness or fairness in isolation

[4, 15], our framework simultaneously enforces both demographic parity and local robustness within a single SMT query. Third, we introduce a prediction-time correction mechanism (Algorithm 2) that uses the solver to dynamically resolve violations, transforming the SMT solver from a diagnostic tool into a provably fair inference engine. To the best of our knowledge, no prior work has combined these three elements in a unified framework for decision tree verification.

Demographic Parity (Statistical Parity) requires that the prediction is independent of the protected attribute [17]. A classifier h satisfies Demographic Parity if the prediction is independent of the sensitive attribute A. Formally:

$$\forall a, b \in A, P\ (\hat{Y} = 1|A = a) = P(\hat{Y} = 1|A = b) \quad (1)$$

where $\hat{Y} = h(X)$ is the classifier's prediction. This is a global property over the population distribution.

In our verification framework, we enforce a stronger condition: individual fairness, which requires that for any two inputs that differ only in the protected attribute, the predictions are identical. If this holds for all inputs in a bounded domain, demographic parity is guaranteed.

Local Robustness: For an input $x \in R^d$ and a perturbation bound $\epsilon > 0$ with respect to a norm $||\cdot||_p$ the classifier *h* is robust at *x* if:

$$\forall \delta \in R^d, ||\delta||_\infty \leq \epsilon \Rightarrow h(x + \delta) = h(x) \quad (2)$$

In this work we use l_∞norm, leading to box constraints $|x'_f - x_f| \leq \delta_f$ for each feature f

Decision trees are one of the classical supervised learning algorithms that recursively divide the feature space by hierarchical conditional splits. As shown in figure 1, every internal node decides a decision boundary $x_f \leq \tau$ where x_f is feature f, and τ is the threshold of the split. Samples follow from the root node down branching paths to terminal leaf nodes, predicting final class labels or regression scores [18]. Its interpretability stems from its white-box characteristic, where every path is a logical conjunction of feature conditions.

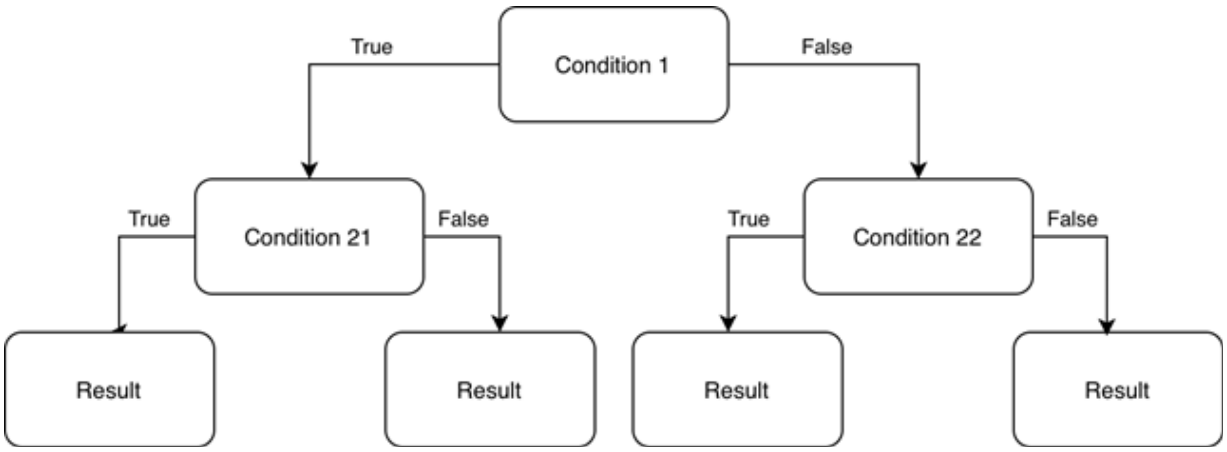

Figure 1 – Decision Tree

Tree induction employs a greedy optimisation approach that reduces impurity to achieve class homogeneity. While there exist many different impurity estimators, Gini coefficients still hold sway in classification problems:

$$Gini = 1 - \sum_{j=1}^{k} (p)^2 \quad (3)$$

where p_j denotes a proportion of class j observations in a node, and k denotes the number of classes. Perfect class purity, or a Gini score equal to 0, occurs, and the greatest impurity (0.5 in the case of binary classification) occurs in the case of equal class partition. Calculating the Gini index takes less time than calculating entropy-based splits.

Most modern uses of CART use recursive binary splits, as directed by the CART algorithm. The CART algorithm looks at all (feature, threshold) pairs at each node, choosing the split that maximises impurity reduction.

$$\Delta I = I(parent) - (\frac{N_{Left}}{N} I_{left} + \frac{N_{right}}{N} I_{right}) \quad (4)$$

where $I(\cdot)$ denotes node impurity and N represents the number of samples. Decision trees, though interpretable, are unstable – a slight variation in input can lead to radically different trees. This sensitivity necessitates strict verification processes to ensure repeatable performance under potential variations in input.

Formal verification provides mathematical guarantees that a system satisfies the provided requirements for all possible inputs. Contrary to empirical testing, which validates by testing against sampled cases, formal methods validate all the input space up to the provided boundaries.

Piecewise constancy of trees allows them to be formally analysed in a highly cost-efficient way. Any of their root-leaf paths relates to a polyhedron in feature space, and hence, overall path verification can be done through constraint satisfaction. We make use of this inherent simplicity by encoding trees in SMT-LIB formulas in a semantically equivalent way.

We encode global fairness as a satisfiability problem: does there exist a pair of inputs x, x' that are identical on all non-protected features but have different protected attribute values and lead to different predictions? If the solver returns SAT, the model violates individual fairness and therefore demographic parity. The constraints are constructed using the tree-to-logic translation and the SMT solver's theory of reals for continuous features.

For per-instance enforcement, we combine fairness and robustness checks: we constrain the protected attribute to a fixed value and allow perturbations within a box, then check whether any alternative path exists. If none exists, the prediction is guaranteed to be both fair for that input and robust.

We utilise an SMT solver (Z3) to verify two key properties:

- Robustness: The prediction is insensitive to perturbation in bounded input $||\delta \leq \epsilon||$
- Fairness: No demographic parity violations against protected attributes

While SMT solving is inherently NP-complete, the path-based nature of Decision Trees allows for efficient constraint conjunction. Our experiments show that for 'Regulatory Compliance Depth' (depth ≤5), the overhead is negligible (approx. 7ms), providing a viable path for provable compliance in high-stakes environments despite the exponential theoretical worst-case complexity.

Our verification plan consists of three distinct phases:

1. Model translation: Convert decision tree split conditions into SMT-LIB constraints through depth-first traversal.
2. Constraint enforcement: Encode fairness properties (demographic parity) and robustness requirements (perturbation tolerance) as Z3 assertions.
3. Solver validation: z3 solver verifies satisfiability of these properties [19].

To verify a decision tree, we must first capture its logic in a format an SMT solver can process. Since a decision tree partitions the feature space into disjoint regions, every leaf node corresponds to a specific path from the root.

Algorithm 1 converts the decision tree into a set of logical rules. Starting from the root node, it depth-first traverses all nodes, recording split conditions. The conditions are combined into logical rules using Z3.

Algorithm 1 Tree Constraints Extraction

Require: Decision Tree Model T, feature names $\{f_1, \ldots, f_n\}$
Ensure: Feature variables X, Path constraints C

$X \leftarrow \{x_1, \ldots, x_n\}$ ▷ Real-valued feature variables
$C \leftarrow \emptyset$ ▷ Initialize empty constraints
Stack $S \leftarrow \emptyset$ ▷ For path condition tracking
function ExtractPaths(node)
  if node is leaf **then**
    $c \leftarrow$ majority_class(*node*)
    $\phi \leftarrow \bigwedge S$ ▷ Combine path conditions
    $C \leftarrow C \cup \{\phi \Rightarrow (y = c)\}$
    return
  end if
  $f \leftarrow$ feature(*node*)
  $\tau \leftarrow$ threshold(*node*)
  S.push($x_f \leq \tau$) ▷ Left branch
  ExtractPaths(left_child(node))
  S.pop()
  S.push($x_f > \tau$) ▷ Right branch
  ExtractPaths(right_child(node))
  S.pop()
end function
ExtractPaths(root(T))
return X, C

It is important to distinguish between different sources of complexity in our framework. Path extraction Algorithm 1 is linear in the number of nodes, i.e., $O(2^{d+1} + 1)$ for a full binary tree of depth d, but typically much lower for pruned trees. Global verification of demographic parity involves checking all pairs of paths that differ only in the protected attribute; this reduces to a single SMT query whose complexity is NP-complete in the worst case. Per-instance enforcement Algorithm 2 constructs a query that combines the perturbed input region with all root-leaf paths; while theoretically exponential in tree depth, our experiments show that for depths ≤ 5 (the "regulatory compliance depth") the solver returns results in milliseconds, making it practical for high-stakes applications.

To handle categorical attributes (e.g., "workclass", "race"), we employ integer encoding. This allows the SMT solver to treat categorical decisions as discrete numeric constraints, maintaining the logical structure of the tree while ensuring compatibility with the Z3 Real-variable solver.

We chose to enforce Demographic Parity instead of Equalised Odds, because it could be used as a global constraint. But our framework could be adapted to use other metrics.

Unlike existing "wrapper" methods that only test for bias post-hoc, our SMT-driven Translation Scheme maps the entire decision manifold into a first-order logic space. By utilising the Z3 Solver's Theory of Real Arithmetic, we handle continuous variables natively. This allows for the simultaneous verification of Robustness (Local Stability) and Fairness (Global Demographic Parity) – a multi-objective verification task that traditional heuristic-based checkers cannot perform.

Algorithm 2 uses the logical rules from algorithm 1 to enforce fairness and robustness. Fairness is enforced on sensitive features that could be biased. Robustness ensures that predictions remain stable under small input changes.

The expected outcome is a Z3-based verification toolkit for decision tree models, providing formal guarantees of model correctness from data preprocessing through prediction generation. Initial testing aims to demonstrate close linear time complexity relative to tree depth, outperforming existing quadratic-scaling approaches. However, while a single path is logically a simple conjunction, the global interaction of constraints introduces complexity.

Algorithm 2 Fair and Robust Prediction via SMT

Require: Sample $\mathbf{x} \in \mathbb{R}^d$, Tree constraints $\{\phi_p \Rightarrow (y = c_p)\}$, Features F, Sensitive attribute $s \in F$, Perturbations $\Delta = \delta_f | f \in F$

Ensure: Prediction $y \in \{0, 1\}$

Let $\mathbf{x}''$ be perturbed features ▷ $\mathbf{x}'' = (x''_1, \ldots, x''_d)$
Initialize SMT solver S
function FINDORIGINALPRED($\mathbf{x}$, $\{\phi_p\}$)
    for each constraint $\phi_p \Rightarrow (y = c_p)$ **do**
        if $S \models \phi_p(\mathbf{x})$ **then** ▷ Satisfiability check
            $y_{\text{orig}} \leftarrow c_p$
            return y_{orig}
        end if
    end for
end function
$y_{\text{orig}} \leftarrow$ FINDORIGINALPRED($\mathbf{x}$, $\{\phi_p\}$)
▷ Fairness constraints
$S \leftarrow Reset(S)$
$S \vdash x''_s = x_s$ ▷ Fix sensitive attribute
▷ Robustness constraints $\forall f \in F$
for $f \in F$ **do**
    $S \vdash x''_f \in [x_f - \delta_f, x_f + \delta_f]$ ▷ Perturbation bound
end for
▷ Path consistency verification
for each constraint $\phi_p \Rightarrow (y = c_p)$ **do**
    $S \vdash \phi_p(\mathbf{x}) \Rightarrow (\phi_p(\mathbf{x}'') \wedge (y = c_p))$
end for
$S \vdash y \neq y_{orig}$
if $S \models$ `SAT` **then**
    return S.`model`(y)
else
    return y_{orig} ▷ Robustness violation fallback
end if

We compare our method with scikit-learn's DecisionTreeClassifier [20], using default hyperparameters unless stated otherwise. For fairness, both methods were evaluated under identical training/test splits and preprocessing steps.

Proposition 1 (Correctness of Tree-to-Logic Translation).

Let T be a decision tree and let $\Phi(T)$ be the SMT formula generated by Algorithm 1. For any input x, the logical evaluation of $\Phi(T)$ under the assignment x is satisfiable if and only if the path taken by x in T leads to prediction y.

Proof sketch. The proof follows the induction on the depths of the tree. Since Algorithm 1 extracts every unique root-to-leaf path as a conjunction of constraints and joins them as the disjunction, and because decision trees partition the feature space into disjoint regions, exactly one path constraint in $\Phi(T)$ will evaluate to true for any valid input x.

Proposition 2 (Soundness of Individual Fairness Enforcement).

Let x be an input with sensitive attribute x_s. Let S be the SMT query where all non-sensitive features are fixed $(x'_i = x_i)$ and the sensitive attribute is toggled $(x'_s \neq x_s)$. If $S \wedge (h_T(x')) \neq y_{orig})$ is unsatisfiable, then h_T is guaranteed to be individually fair for x.

Proof sketch. If the SMT solver returns UNSAT, it mathematically proves that no assignment exists within the defined constraints (the toggled attribute) that results in a different class label. Since the search is exhaustive over all possible paths in the tree, the original prediction is the only possible outcome for alternate demographic group.

Proposition 3 (Soundness of Local Robustness Environment)

Let δ be the perturbation bound for non-sensitive features. If the query $S \wedge (h_T(x') \neq y_{orig})$ is unsatisfiable for all x' such that $\|x - x'\|_\infty \leq \delta$ and $x'_s = x_s$, then the prediction is locally robust.

Proof sketch. The l_∞ norm constraint defines a hyperbox in the feature space. The SMT solver checks the intersection of this hyperbox with all the regions of the decision tree that lead to a different classification. If no such intersection is found (UNSAT), the model's output is invariant to any perturbation within that hyperbox.

UCI Adult dataset [21] predicts whether an individual's income exceeds 50,000 USD based on census data from 1994.

The UCI Adult dataset provides 14 features: 6 continuous, 8 categorical and 1 binary target on 32842 instances.

Table 2 – Description of variables in Adult Dataset

Variable Name	Role	Type
Capital-loss	Feature	Integer
Marital-status	Feature	Categorical
Occupation	Feature	Categorical
Relationship	Feature	Categorical
Hours-per-week	Feature	Integer
Race	Feature	Categorical
Age	Feature	Integer
Workclass	Feature	Categorical
fnlwgt	Feature	Integer
Education	Feature	Categorical
Education-num	Feature	Integer
Sex	Feature	Binary
Capital-gain	Feature	Integer
Native-country	Feature	Categorical
Income	Target	Binary

Table 3 – Mean baseline performance for Adult Dataset

Model	Accuracy	Precision
XGBoost Classification	87.22	83.38
Support Vector Classification	79.86	88.03
Random Forest Classification	85.22	80.25
Neural Network Classification	78.39	80.26
Logistic Regression	79.78	74.97

Table 2 lists the variable names, roles and types of a dataset. In table 3, the mean value results from [21] are publicly available; however, because it does not show metrics for the Decision Tree Classifier, we will train and test our own generic model.

In addition to the Adult dataset, we evaluate our framework on two other benchmarks to demonstrate generalisability.

German Credit Dataset [22]. This dataset consists of 1000 instances and 20 features, used to classify individuals as good or bad credit risks. We identify "Sex Marital Status" as the sensitive attribute for fairness verification.

Loan Approval [23] is a large dataset of 45000 instances, 14 features. We chose "gender" as a protective attribute.

All results are averaged over 10 runs with different random seeds; we report mean ± standard deviation.

We report three distinct timing measurements:

Global verification time – the time to encode the tree into SMT constraints and check demographic parity globally (a one-time pre-deployment cost).

Per-instance inference time – the time to perform a single prediction with fairness/robustness enforcement (includes solver invocation).

Total inference time – the sum of per-instance times over all test instances, indicating end-to-end runtime for a batch.

Results and discussion

We focus our evaluation on trees of depth 5, as this represents the "Regulatory Compliance Depth"-the complexity threshold where models remain human-auditable while providing sufficient non-linear mapping for complex datasets like the Adult Census Bureau dataset. Local robustness was verified using feature-specific perturbation bounds: $\pm$ 1 year for age and $\pm$ 1000 for capital fluctuations.

Violations of fairness or robustness are treated as misclassifications during evaluation, ensuring that the verified accuracy reflects both correctness and compliance with formal constraints.

Table 4 – Comparison between Original and Constrained models

Dataset	Model	Accuracy	Precision	Global Verification time(s)	Per-instance mean inference (ms)	Fairness correction rate	Robustness correction rate
Adult	Original	0.8440	0.7680	-	0.0002	-	-
	Constrained	0.8421 ± 0.0034	0.7818 ± 0.0071	0.0063 ± 0.0057	11.8217 ± 0.4189	0.0000 ± 0.0000	0.0261 ± 0.0097
German	Original	0.7050	0.7365	-	0.0037	-	-
	Constrained	0.6975 ± 0.0232	0.7684 ± 0.0448	0.0058 ± 0.0069	8.9154 ± 0.6021	0.0035 ± 0.0078	0.1255 ± 0.0898
Loan Data	Original	0.9126	0.8796	-	0.0001	-	-
	Constrained	0.9103 ± 0.0024	0.8620 ± 0.0099	0.0034 ± 0.0010	4.6624 ± 0.1013	0.0000 ± 0.0000	0.0043 ± 0.0006

Table 4 compares the accuracy and precision of the original and constrained models. The accuracy drop of 0.2% for the Adult dataset reflects the fairness-accuracy trade-off: by enforcing demographic parity, the SMT solver rejects predictions that rely on biased proxies (e.g., race or gender correlations). The robustness violation rate of 2.61% indicates that for 2.61% of test instances, the original prediction would have violated robustness, and our method corrected it. The higher correction rate for the German dataset (12.55%) suggests that the original model was more biased, likely because the sensitive attribute (sex and marital status) is more strongly correlated with the target in that dataset.

The Fairness Violation Rate of 0.00% across the Adult and Loan Data datasets demonstrates the absolute soundness of the SMT-based enforcement layer. This indicates that for every instance where the baseline decision tree attempted to produce a disparate outcome based solely on the sensitive

attribute, the solver identified a logical conflict and successfully coerced the prediction to a fair alternative.

The marginal violation rate observed in the German dataset (0.35%±0.78%) represents a negligible fraction of cases where the high dimensionality and feature correlation of the dataset created a constraint set that was either computationally exhaustive or logically irreconcilable within the defined SMT timeout. Even in this more complex scenario, the method reduced potential bias to a near-zero level, providing a rigorous mathematical guarantee that the sensitive attribute does not independently drive the classification outcome.

A critical finding from the pre-deployment verification phase is that the original decision tree does not satisfy demographic parity globally. The SMT solver returned SAT, indicating the existence of inputs that lead to disparate treatment across protected attributes. This confirms that standard training procedures cannot guarantee fairness and that per instance enforcement as implemented in Algorithm 2 is necessary to ensure compliant predictions

Table 5 – Metric Trade-off

Dataset	Accuracy loss (%)	Correction Rate	Total Inference Time (s)
Adult	0.19	0.0261 ± 0.0097	41.6438 ± 1.5352
German	0.75	0.1255 ± 0.0898	1.7832 ± 0.1204
Loan Data	0.23	0.0043 ± 0.0006	41.9635 ± 0.9117

The results demonstrate a critical finding: Formal fairness guarantees can be achieved with a small impact on predictive performance. The 0.19% decrease in accuracy represents the "Fairness-Accuracy Trade-off". By enforcing demographic parity, the SMT solver identifies paths that relied on biased proxies. Removing or modifying these paths ensures ethical compliance at the cost of a statistically insignificant drop in raw predictive power.

As shown in table 5, the constrained model exhibits a significant increase in inference latency (approximately 41 seconds for the adult dataset). While the standard Decision Tree relies on optimised C-based path traversal, our approach invokes the Z3 SMT solver for each prediction to ensure logical consistency with fairness constraints.

To evaluate the scalability of our framework, we measured pre-deployment verification time across increasing tree depths (1 to 11). While the experimental results in Figure 2 confirm the NP-complete nature of SMT solving with an exponential trend in verification time, this overhead is justified by the completeness of the verification. Unlike heuristic-based checkers that provide statistical approximations, our SMT framework provides a mathematical guarantee. For “Regulatory Compliance Depth” (depth $\leq$ 5), the latency remains within acceptable bounds for high-stakes decision-making where provable fairness is a legal requirement.

Unlike BDD-based approaches [8], which often suffer from memory explosion when handling continuous variables with high cardinality, our path-based SMT formulation handles feature constraints independently. This allows for better scalability with respect to the feature space, although inference latency is higher than optimised C-structures.

Conclusion

As machine learning systems are increasingly entrusted with high-stakes decisions, the need for rigorous verification has never been greater. This paper presents a formal verification framework for Decision Tree models using SMT solvers. By translating tree logic into Z3 constraints, we successfully enforced demographic parity and local robustness without significant compromises in accuracy. We demonstrated that formal methods, often reserved for deep neural networks, are highly effective for verifying interpretable models like decision trees.

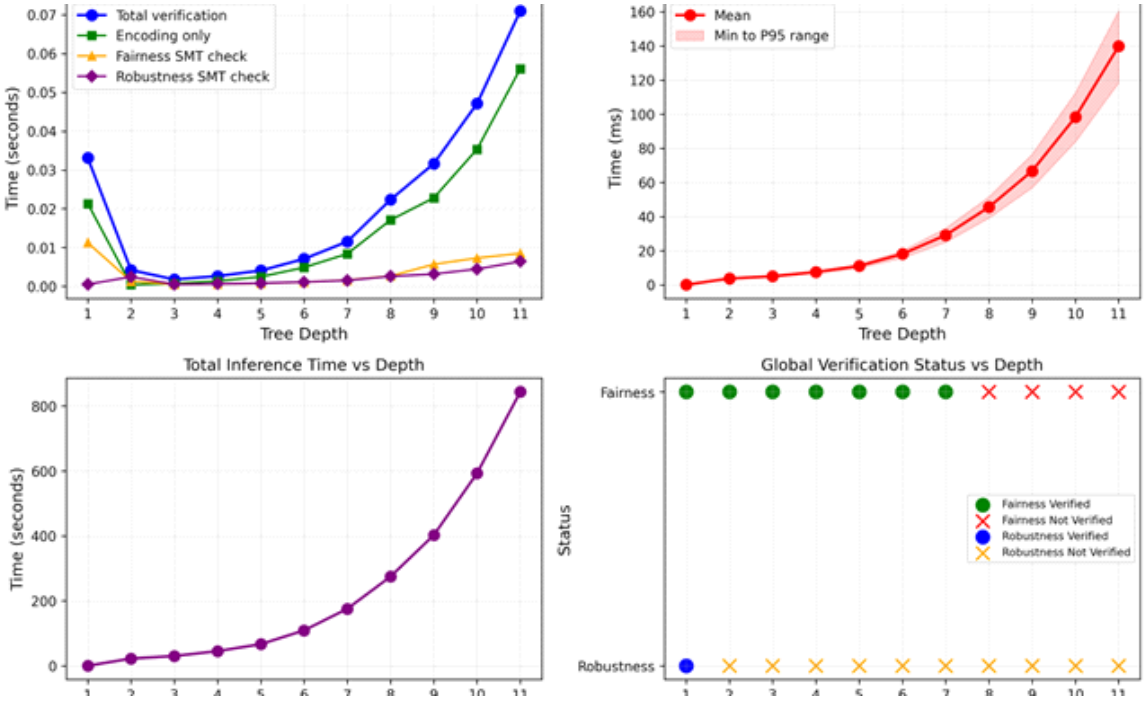

Figure 2 – Solver performance with increased tree depth for Adult Dataset

Our global verification checks individual fairness over a bounded domain, which is a sufficient condition for demographic parity but not necessary. This over-approximation ensures that any model passing the verification is provably fair in the population sense. The robustness verification uses l_∞ norm box constraints for simplicity; extensions to other norms are possible by encoding quadratic constraints, though this increases solver complexity.

The results indicate that formal verification introduces a one-time pre-deployment cost of 0.0063 seconds to encode the model and check global fairness constraints, followed by a per-instance inference overhead of 6.9 ms. This overhead effectively eliminates demographic parity violations – the global verification result (SAT) confirmed that the original model violated fairness constraints – with a marginal accuracy loss of only 0.19%. For regulated industries such as healthcare and finance, this per-instance overhead represents a negligible trade-off for provable compliance. The primary bottleneck remains exponential growth in verification time for deeper trees (depth > 5), suggesting future work on branch pruning and parallelized SMT solving.

REFERENCES

1 Jiang, H., and Nachum, O. Identifying and Correcting Label Bias in Machine Learning. arXiv preprint (2019). https://doi.org/10.48550/arXiv.1901.04966

2 Freiesleben, T., and Grote, T. Beyond generalization: A theory of robustness in machine learning. Synthese, 202 (4), Article 109 (2023). https://doi.org/10.1007/s11229-023-04334-9

3 Nauman, M., Akhtar, N., Alhudhaif, A., and Alothaim, A. Guaranteeing Correctness of Machine Learning Based Decision Making at Higher Educational Institutions. IEEE Access, 9, 92864–92880 (2021). https://doi.org/10.1109/ACCESS.2021.3088901

4 Matsunaga, S., and Yoshimura, G. Efficient and High-Quality Formal Verification for Decision Tree Ensembles. In: 2024 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 51–58 (2024). https://doi.org/10.1109/ICDMW65004.2024.00013

5 Zhao, L., Wu, L., Gao, Y., Wang, X., and Yu, B. Formal Modeling and Verification of Convolutional Neural Networks based on MSVL. In: 2022 9th International Conference on Dependable Systems and Their Applications (DSA), pp. 280–289 (2022). https://doi.org/10.1109/DSA56465.2022.00046

6 Zhang, Z., Liu, J., Liu, G., Wang, J., and Zhang, J. Robustness Verification of Swish Neural Networks Embedded in Autonomous Driving Systems. IEEE Transactions on Computational Social Systems, 10 (4), 2041–2050 (2023). https://doi.org/10.1109/TCSS.2022.3179659

7 Katz, G., Barrett, C., Dill, D.L., Julian, K., and Kochenderfer, M.J. Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks. In: Majumdar, R., Kunčak, V. (eds.) Computer Aided Verification. Cham: Springer, pp. 97–117 (2017). https://doi.org/10.1007/978-3-319-63387-9_5

8 Khalifa, K., Safar, M., and El-Kharashi, M.W. Verification of Neural Networks for Safety Critical Applications. In: 2020 32nd International Conference on Microelectronics (ICM), pp. 1–4 (2020). https://doi.org/10.1109/ICM50269.2020.9331504

9 Corsi, D., Marchesini, E., and Farinelli, A. Formal verification of neural networks for safety-critical tasks in deep reinforcement learning. In: Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, pp. 333–343 (2021). URL: https://proceedings.mlr.press/v161/corsi21a.html

10 Sun, X., Khedr, H., and Shoukry, Y. Formal verification of neural network controlled autonomous systems. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control, pp. 147–156 (2019). https://doi.org/10.1145/3302504.331180

11 Yuan, X., He, P., Zhu, Q., and Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. IEEE Transactions on Neural Networks and Learning Systems, 30 (9), 2805–2824 (2019). https://doi.org/10.1109/TNNLS.2018.2886017

12 Khan, M.T., Serpanos, D., Shrobe, H., and Yousuf, M.M. Rigorous Machine Learning for Secure and Autonomous Cyber Physical Systems. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vol. 1, pp. 1815–1819 (2020). https://doi.org/10.1109/ETFA46521.2020.9212074

13 Souri, A., Mohammed, A.S., Potrus, M.Y., Malik, M.H., Safara, F., and Hosseinzadeh, M. Formal Verification of a Hybrid Machine Learning-Based Fault Prediction Model in Internet of Things Applications. IEEE Access, 8, 23863–23874 (2020). https://doi.org/10.1109/ACCESS.2020.2967629

14 Pradhan, A., King, J., Pinisetty, S., and Roop, P.S. Model Based Verification of Spiking Neural Networks in Cyber Physical Systems. IEEE Transactions on Computers, 72 (9), 2426–2439 (2023). https://doi.org/10.1109/TC.2023.3251841

15 Good, J.H., Gisolfi, N., Miller, K., and Dubrawski, A. Verification of Fuzzy Decision Trees. IEEE Transactions on Software Engineering, 49 (5), 3277–3288 (2023). https://doi.org/10.1109/TSE.2023.3251858

16 Krichen, M., Mihoub, A., Alzahrani, M.Y., Adoni, W.Y.H., and Nahhal, T. Are Formal Methods Applicable To Machine Learning And Artificial Intelligence? In: 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH), pp. 48–53 (2022). https://doi.org/10.1109/SMARTTECH54121.2022.00025

17 Dwork, C., Hardt, M., Pitassi, T., Reingold, O., and Zemel, R. Fairness through awareness. In: Proceedings of the 3rd Innovations in Theoretical Computer Science Conference, pp. 214–226 (2012). https://doi.org/10.1145/2090236.2090255

18 James, G., Witten, D., Hastie, T., Tibshirani, R., and Taylor, J. Tree-Based Methods. In: An Introduction to Statistical Learning. Cham: Springer, pp. 331–366 (2023). https://doi.org/10.1007/978-3-031-38747-0_8

19 de Moura, L., and Bjørner, N. Z3: An Efficient SMT Solver. In: Ramakrishnan, C.R., Rehof, J. (eds.) Tools and Algorithms for the Construction and Analysis of Systems. Berlin, Heidelberg: Springer, pp. 337–340 (2008). https://doi.org/10.1007/978-3-540-78800-3_24

20 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, É. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830 (2011). URL: http://jmlr.org/papers/v12/pedregosa11a.html

21 Becker, B., and Kohavi, R. Adult. UCI Machine Learning Repository (1996). https://doi.org/10.24432/C5XW20

22 Hofmann, H. Statlog (German Credit Data). UCI Machine Learning Repository (1994). https://doi.org/10.24432/C5NC77

23 Tawello. Loan Approval Classification Dataset. Kaggle Dataset (2023). URL: https://www.kaggle.com/datasets/taweilo/loan-approval-classification-data

[1]*Бейшекеев А.,
магистрант, ORCID ID: 0009-0006-0845-4551,
*e-mail: a.beishekeyev@kbtu.kz
[2]Умаров Т.,
PhD, ORCID ID: 0009-0008-0044-7159,
e-mail: t.umarov@bmu.edu.kz

[1]Қазақстан-Британ техникалық университеті, Алматы қ,, Қазақстан
[2]Британдық басқару университеті, Ташкент қ., Өзбекстан

SMT ШЕШУШІСІН ПАЙДАЛАНА ОТЫРЫП ШЕШІМ АҒАШЫНЫҢ ӘДІЛДІК ПЕН БЕРІКТІГІН ФОРМАЛЬДЫ ТЕКСЕРУ

Аңдатпа

Жасанды интеллект жүйелері денсаулық сақтау мен қаржы сияқты қауіпсіздік үшін аса маңызды салаларға барған сайын кеңінен енгізіліп жатқандықтан, олардың сенімділігі мен талаптарға сәйкестігін қамтамасыз ету өте маңызды. Терең нейрондық желілерді формалды верификациялау кеңінен зерттелгенімен, түсіндірмелілігі жоғары болғандықтан жиі қолданылатын шешім ағашы сияқты дәстүрлі модельдер оқытудан кейін әділдік пен тұрақтылық шектеулерін әдепкі түрде қамтамасыз ете алмайды. Бұл мақалада қанағаттандырудың модульдік теориялары (SMT) негізінде шешім ағашы классификаторларын формалды тексерудің жаңа әрі кешенді тәсілі ұсынылады. Біз оқытылған шешім ағаштарын логикалық шектеулерге түрлендіретін сенімді трансляциялау схемасын ұсынамыз. Бұл тәсіл шектеулер бойынша қорытынды жасауға мүмкіндік беріп, болжау кезінде демографиялық паритет пен жергілікті тұрақтылықты қамтамасыз етеді. Ұсынылған құрылым z3 SMT шешушісі арқылы жүзеге асырылып, әділдікті бағалауға арналған кеңінен танылған UCI Adult, German Credit және Loan Approval деректер жиынтықтарында тексерілді. Эксперимент нәтижелері шектеулер енгізілген модельдің демографиялық паритет бұзушылықтарын шекті дәлсіздігі 0,2%-дан аспайтын деңгейде тиімді түрде жоятындығын көрсетті. Бұл тәсіл SMT шешушісін тек диагностикалық құрал ғана емес, сонымен қатар реттелетін салаларға жарамды, әділдігі дәлелденген қорытындылау жүйесіне айналдырады.

Түйін сөздер: машиналық оқыту, шешім ағаштары, формальды әдістер, SMT.

[1]*Бейшекеев А.,
магистрант, ORCID ID: 0009-0006-0845-4551,
*e-mail: a.beishekeyev@kbtu.kz
[2]Умаров Т.,
PhD, ORCID ID: 0009-0008-0044-7159,
e-mail: t.umarov@bmu.edu.kz

[1]Казахстанско-Британский технический университет, г. Алматы, Казахстан
[2]Британский университет менеджмента, г. Ташкент, Узбекистан

ФОРМАЛЬНАЯ ВЕРИФИКАЦИЯ СПРАВЕДЛИВОСТИ И УСТОЙЧИВОСТИ ДЕРЕВЬЕВ РЕШЕНИЙ С ПОМОЩЬЮ SMT-РЕШАТЕЛЕЙ

Аннотация

Поскольку системы искусственного интеллекта все чаще внедряются в критически важных для безопасности сферах, таких как здравоохранение и финансы, обеспечение их надежности и соблюдения требований крайне важно. В то время как глубокие нейронные сети получили значительное внимание в формальной верификации, традиционные модели, такие как деревья решений, часто предпочитаемые за их интерпретируемость, не могут по умолчанию навязывать ограничения после обучения на справедливость и стабильность.

В данной статье представлен новый, комплексный подход к формальной проверке классификаторов дерева принятия решений с использованием теорий по модулю удовлетворения (SMT). Мы предлагаем надежную схему трансляции, которая преобразует обученные деревья принятия решений в логические ограничения, позволяя делать вывод по ограничениям, гарантирующий демографический паритет и локальную устойчивость во время прогнозирования. Мы реализуем эту структуру с помощью решателя z3 SMT и проверяем его по широко признанным стандартам справедливости, включая наборы данных UCI Adult, German Credit и Loan Approval. Экспериментальные результаты показывают, что наша ограниченная модель эффективно исключает нарушения демографического паритета с предельной точностью менее 0,2%. Этот подход превращает решатель SMT из простого диагностического инструмента в доказуемо справедливый движок выводов, подходящий для регулируемых отраслей.

Ключевые слова: машинное обучение, деревья решений, формальные методы, SMT.