*Sensors and Materials*, Vol. 38, No. 5 (2026) 2507–2523
MYU Tokyo



S & M 4450

# Debasing Tabular Data via Actionable Explanations for Fair Machine Learning

Jinchao Ge,[1,3] Tao Du,[1] Haidong Li,[2*] Nan Ju,[1] and Shuwen Zhao[1,4]

[1]Innovation Center for Smart Medical Technologies & Devices, Binjiang Institute of Zhejiang University, Hangzhou 310053, China
[2]China Telecom Ningbo Branch, Ningbo 315000, China
[3]School of Computing and Information Technology, University of Wollongong, Northfields Avenue, Wollongong, NSW 2522, Australia
[4]Mstar Technologies, Inc., Hangzhou 311100, China

(Received March 26, 2026; accepted April 16, 2026)

***Keywords:*** machine learning fairness, bias detection, feature interactions, model interpretability

As machine learning (ML) models are increasingly used in high-stakes domains, improving both predictive performance and fairness has become an important research problem. Existing explanation-based fairness analysis methods often focus on individual feature effects while paying limited attention to feature redundancy and interaction, which may lead to incomplete bias diagnosis. In this paper, we propose the Prioritizing Redundancy and Interaction through Monte Carlo Tree (PRIMCT), a fairness-oriented feature subset analysis framework based on Monte Carlo Tree Search (MCTS). The method jointly evaluates feature importance, redundancy, and interaction to identify feature subsets that are informative for model behavior and useful for bias diagnosis. We validate PRIMCT on three tabular datasets, namely, German Credit Data, Adult Income Data (AID), and Stop, Question, and Frisk Data. Experimental results show that PRIMCT consistently achieves a better balance between predictive performance and fairness-related metrics than several baseline methods. In particular, the method achieved an increase of 13.34% in accuracy and an improvement of 11.31% in fairness metrics on AID.

## 1. Introduction

Fair machine learning (FML) has emerged as a critical field in response to the integration of ML algorithms into consumer electronics (CE) and high-stake domains such as finance, healthcare, and criminal justice.[1–3] In CE, FML plays a pivotal role in ensuring that automated systems make equitable decisions, free from discrimination based on sensitive attributes such as race, gender, and age. This shift reflects the growing importance of developing transparent and interpretable FML systems to address widespread concerns about algorithmic opacity.

The increasing complexity of machine learning (ML) models has also stimulated growing interest in explainable artificial intelligence (XAI), which aims to provide human-understandable explanations for model behavior.[4,5] As regulatory bodies and the public demand more

*Corresponding author: e-mail: 15306662336@189.cn
https://doi.org/10.18494/SAM6352

ISSN 0914-4935 

transparency, explainability has become an essential aspect of trustworthy ML. However, despite recent progress in XAI, many existing techniques still provide limited support for bias diagnosis and mitigation. In particular, they often focus on feature-based explanations alone, which may not be sufficient to capture the complex mechanisms through which unfairness arises in ML systems.

The evolution of FML has been driven by the increasing demand for trustworthy technologies in consumer-facing applications, including personalized devices, smart home systems, and digital assistants. Recent work on fairness explanation has shown that feature-based approaches can help identify attributes associated with unfair predictions, but they may fail to capture more complex feature interactions and may overlook redundancy among correlated features.[6,7] As a result, these methods can produce incomplete or even misleading explanations of model unfairness. In addition, diagnostic explanations that trace discriminatory behavior back to its roots in the training data remain relatively limited.[8–10] This has motivated increasing interest in causal responsibility, which seeks to quantify the extent to which interventions on training data or model components may help resolve bias. Gopher,[9] a system that produces interpretable and causal explanations for bias by identifying coherent subsets of training data, also faces limitations. While it aims to provide compact and interpretable explanations, it may struggle with the scalability and complexity of real-world datasets. By identifying the root causes of bias, we can take a more targeted approach to mitigate its impact.[11] However, the implementation of such diagnostic tools is still in its infancy, and more research is needed to develop robust methods that can effectively address these challenges.[6]

To address this gap, there is a pressing need for diagnostic explanations that can trace unexpected or discriminatory behavior back to its roots in the training data. This necessity has led to the introduction of the concept of causal responsibility, which seeks to quantify the extent to which intervening in training data can resolve bias. By identifying the root causes of bias, we can take a more targeted approach to mitigating its impact.

In this paper, we present an approach to enhancing fairness in ML models, which we term the Prioritizing Redundancy and Interaction through Monte Carlo Tree (PRIMCT) search method. This search method is an innovative application of Monte Carlo Tree Search (MCTS) that delves into the intricacies of feature importance, redundancy, and interaction—key determinants of bias in ML models. Using an MCTS-based feature selection process, our method provides a nuanced and potent strategy for scrutinizing and refining ML models. The PRIMCT search method begins with a comprehensive initialization of the MCTS tree, setting the stage for an in-depth exploration of feature subsets. Through a series of iterative expansions, simulations, and backpropagations, the algorithm meticulously evaluates the contribution of each feature to the model's fairness and overall performance. The Upper Confidence Bound for Trees (UCT) score guides the expansion and selection process, ensuring a balanced exploration of the feature space while favoring promising areas. A distinctive aspect of the PRIMCT search method is its ability to pinpoint subsets of training data that are instrumental in perpetuating bias. This capability is achieved through a series of random walks to leaf nodes, simulating the impact of various feature combinations on model fairness. The results from these simulations are then used to update the performance estimates, which in turn refine the selection of features that contribute

to a fairer model prediction. The culmination of this process is the identification of an optimal feature subset. As shown in Fig. 1, to demonstrate the effectiveness of our method, we conduct experimental evaluations that showcase its ability to select features that are not biased. These evaluations serve as a testament to the potential of the MCTS-based method in offering a more transparent and FML model.

In this paper, we examine the challenges arising from the growing use of ML in sensitive domains and discuss the limitations of existing XAI techniques for bias diagnosis. To address these issues, we propose an MCTS-based method, PRIMCT, which provides a structured framework for fairness-oriented feature subset analysis. Unlike approaches that rely mainly on individual feature attribution or training-data subset identification, PRIMCT jointly considers feature importance, redundancy, and interaction within a unified search process, enabling the identification of more informative and non-redundant feature combinations. In this way, the proposed method supports more actionable bias diagnosis while maintaining model interpretability. The three key contributions of this work are summarized as follows.

(1) We propose an MCTS-based framework that jointly considers feature importance, feature interaction, and feature redundancy for fairness-oriented analysis in tabular ML models.
(2) We develop a feature subset search strategy that prioritizes informative and non–redundant feature combinations, thereby providing more actionable support for bias diagnosis than methods based only on individual feature attribution.

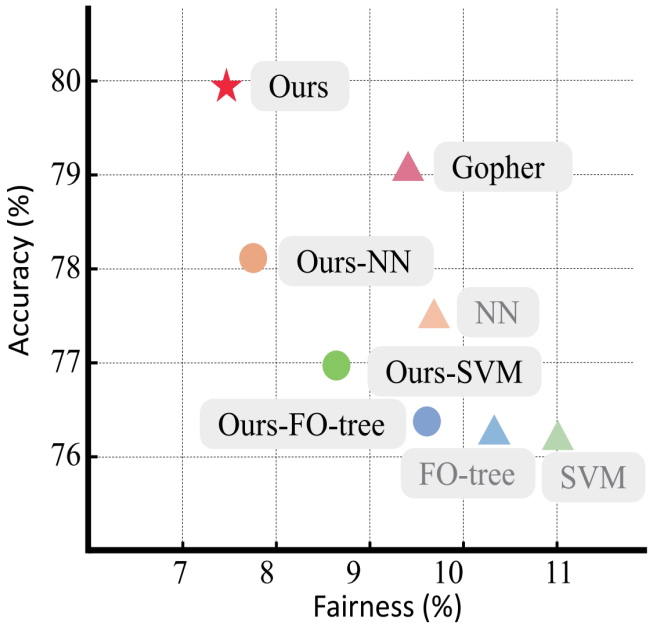


Fig. 1. (Color online) Results of comparative analysis of our methods on GCD. We illustrate the comparative performance of various models in achieving fairness, as measured by the statistical parity (SP) of fairness. The models evaluated include the standard support vector machine (SVM), neural network (NN), First Order tree (FO–tree), and Gopher, alongside our proposed models: Ours, Ours–FO–tree, Ours–NN, and Ours–SVM. The results demonstrate a consistent trend where the Ours–FO–tree model outperforms others, suggesting its enhanced capability in ensuring fairness in predictions.

(3) We evaluate the proposed method on three benchmark datasets, namely, German Credit Data (GCD), Adult Income Data (AID), and Stop, Question, and Frisk Data (SQFD). Experimental results show that PRIMCT achieves improved fairness–performance trade–offs compared with several baseline approaches.

## 2. Related Work

Pursuit of fairness in ML is a multifaceted challenge that has gained significant attention in recent years. The field has evolved from focusing on algorithmic transparency to a more nuanced understanding of fairness, encompassing various ethical dimensions.

### 2.1 ML fairness

The journey towards fairness in ML reflects the dynamic nature of the field. Initially, the emphasis was on transparency and accountability, but the discourse has matured to include a broader spectrum of fairness definitions. The seminal work of Dwork *et al*.[12] laid the foundation for individual fairness, and the literature has since expanded beyond early group and individual level notions to a broader spectrum of definitions, including causal perspectives on fairness.[13,14] Huang *et al*.[15] emphasized the importance of integrating fairness considerations throughout the entire ML pipeline, from data collection to model deployment, with a focus on continuous monitoring and adjustment.

Algorithmic fairness encompasses a variety of concepts to address ethical concerns. Calders *et al*.[16] introduced statistical parity, ensuring equal outcome distribution across groups. Hardt *et al*.[17] proposed equal opportunity, aiming for equal true positive rates among groups. Zhang *et al*.[18] explored predictive parity, which seeks to equalize predictive outcomes without regard to true labels. These concepts, while distinct, are often mutually incompatible under realistic assumptions and therefore must be balanced in practice to build an FML system.[19]

### 2.2 XAI and feature-based explanation

XAI is a critical field aimed at demystifying ML model decisions.[20] Feature-based explanations, including feature importance scoring and selection methods, have been foundational.[4] Nguyen *et al*.[21] and Mersha *et al*.[4] provided recent surveys of these techniques and their human–centered evaluation. However, as Carloni *et al*.[22] noted, these methods may not always reveal the causal relationships underlying biased outcomes.

To overcome this, the field has developed more sophisticated methods that consider the causal mechanisms of model decisions. Makhlouf *et al*.[14] emphasized the importance of causal explanations in understanding and mitigating bias. Diagnostic explanations, as highlighted by Surve and Pradhan,[10] identify the root causes of bias, providing a deeper understanding of fairness.

In recent studies, the integration of fairness into deep learning architectures,[23] the role of fairness in federated learning scenarios,[24] and the development of fairness–aware data augmentation techniques[25] have been explored. Additionally, there has been a surge in research on the explainability of AI systems, focusing on the interpretability of neural networks and the creation of new metrics for evaluating explanation quality.[26] Much of XAI research focuses on explaining ML models in terms of patterns and dependences between input features and their outcomes. Feature attribution methods, such as Shapley–style additive attributions, quantify the responsibility of input features for model predictions.[27] Methods based on surrogate explainability approximate ML models using a simple, interpretable model (e.g., local surrogate explainers).[21,28] Contrastive and causal methods explain ML model predictions in terms of minimal interventions or perturbations on input features that change the prediction.[29,30] Logic-based methods use tools from logic-based diagnosis and constraint solving to compute minimal sets of features that are sufficient and necessary for ML model predictions.[31] These approaches can still fall short in generating diagnostic explanations that help users trace an ML model's unexpected or discriminatory behavior back to its training data.[8]

The quest for fairness and explainability in ML is an ongoing journey, with each year bringing new insights and methodologies that enrich our understanding and capabilities in the field.

### 2.3 FML in consumer credit evaluation

FML in consumer credit evaluation has become a critical area of study as financial institutions aim to provide equitable access to credit while reducing biases in decision–making processes. CE devices, including smartphones and laptops, play a significant role as interfaces between consumers and credit systems. These devices facilitate real–time interactions and host AI–driven algorithms that enhance the efficiency and fairness of credit evaluations. Their integration bridges the gap between advanced ML models and end–users, ensuring seamless access to credit systems.

Research in consumer credit risk assessment categorizes algorithms into traditional single classifiers, intelligent single classifiers, and hybrid or ensemble classifiers.[32] These studies examine how model interpretability, bias detection, multi–pattern capabilities, and fairness contribute to improving the reliability of these systems. In their review of credit scoring models, Valdrighi *et al.*[33] and Ayari *et al.*[34] focused on key challenges such as feature selection, evaluation metrics, data imbalance, transparency, and model limitations. An example is the GF–LRP framework,[35] which integrates layer–wise relevance propagation with the encoder–decoder branch–specific architecture of variational graph autoencoders to address the interpretability deficiency of graph models and enhance model transparency.

These works are essential in understanding the intersection of fairness, bias detection, feature interactions, and model interpretability, which are critical to addressing the challenges in developing equitable and reliable ML systems for consumer credit evaluation.

## 3. Methodology

### 3.1 Task formulation

Our methodology begins with dataset preparation and the definition of the feature search space. Let the dataset be denoted as $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where $x_i \in \mathbb{R}^m$ represents the input feature vector of the $i$-th sample, $y_i$ denotes the corresponding target variable, $N$ is the total number of samples, and is the number of candidate features. We define the candidate feature set as $\mathcal{F} = \{f_1, f_2, \cdots f_m\}$, where $f_j$ denotes the $j$-th feature.

Our objective is to identify an optimal feature subset $F \subseteq \mathcal{F}$ that improves both predictive performance and fairness-related behavior. To efficiently explore the combinatorial feature space, we formulate feature selection as a search problem and solve it using an MCTS framework.

### 3.2 MCTS structure

We represent subsets of features as nodes within an MCTS tree. Each node corresponds to a feature subset, denoted as $v \equiv F_v \subseteq F$. The root node corresponds to the empty feature set, i.e., $F_{root} = \varnothing$. Each edge in the tree represents the addition of a new feature, and thus, a path from the root to a node represents a feature subset.

In the selection phase, we identify nodes for expansion on the basis of their statistical data. We employ the UCT algorithm to balance the trade–off between exploration and exploitation. The UCT formula for a node is given by

$$UCT(V) = \frac{Q(v)}{n(v)} + c \cdot \sqrt{\frac{\ln(n(parent(v)))}{n(v)}}, \tag{1}$$

where $Q(v)$ signifies the average performance of node $v$, $n(v)$ denotes the number of visits to node $v$, $n(parent(v))$ is the number of visits to the parent node of $v$, and $C$ is the exploration parameter that governs the extent of exploration relative to exploitation.

### 3.3 Performance of feature subsets

In this phase, we augment the tree with new nodes representing previously unconsidered combinations of features. The node with the highest UCT score is selected for expansion, thereby ensuring that the most promising areas of the feature space are explored.

We assess the performance of the feature subsets by running models on a holdout dataset and calculating the performance metrics. The performance of a feature subset $F$ is evaluated using the following formula:

$$\mathcal{P}(F) = \sum_{f_i \in F} \mathcal{J}(f_i) + \mathcal{P}_{int}(F) + \mathcal{P}_{red}(S). \tag{2}$$

Here, $\mathcal{J}(f_i)$ represents the importance of a specific feature $f_i$, $\mathcal{P}_{int}(F)$ denotes the interaction-performance of the feature subset $F$ and $\mathcal{P}_{red}(S)$ is a measure of the redundancy performance within the subset. The importance of an individual feature $f_i$ within the feature set $F$ is evaluated by calculating the average performance gain across all possible subsets $S$ excluding $f_i$. The importance is measured as follows.

$$\mathcal{J}(f_i) = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\left(|F| - |S| - 1\right)!}{|F|!} \times \left(\mathcal{M}(S \cup \{f_i\}) - \mathcal{M}(S)\right) \tag{3}$$

This summation iterates over all possible subsets $S$ of $F$ excluding $f_i$. The term $\frac{|S|!\left(|F| - |S| - 1\right)!}{|F|!}$ represents the number of ways to choose a subset $S$ from $F$, serving as a weight in the calculation. $\mathcal{M}(S \cup \{f_i\})$ and $\mathcal{M}(S)$ denote the predictive performance of the model with and without $f_i$ in the subset $S$, respectively. The difference between these two terms indicates the performance gain upon adding $f_i$ to the subset $S$.

The interaction performance of a feature subset $F$ is assessed using a distance metric $dist(F, x)$ between the feature subset $F$ and samples $X$ from the same and different classes.

$$\mathcal{P}_{int}(F) = -\frac{1}{k} \sum_{i \in same\ class} \mathcal{D}(F, x_i) + \frac{1}{k} \sum_{j \in different\ class} \mathcal{D}(F, x_j) \tag{4}$$

A lower $\mathcal{P}(F)$ indicates better performance, with $\mathcal{D}(F, x_i)$ and $\mathcal{D}(F, x_j)$ representing distances in feature space for intra–class and inter–class nearest neighbors, respectively.

The redundancy performance balances the proportion of the feature subset relative to the total feature set and the model's classification error on that subset.

$$\mathcal{P}_{red}(S) = w \cdot \frac{|S|}{|F|} + (1 - w) \cdot Error(\mathcal{M}(S)), \tag{5}$$

where $w$ is a weight parameter that trades off the importance of subset size against classification error, with values ranging from 0 to 1, $\frac{|S|}{|F|}$ is a measure of the redundancy of the subset size, and $Error(\mathcal{M}(S))$ represents the classification error of the model on the subset $S$. The optimization seeks a subset $S$ that minimizes redundancy while maintaining low classification error, thus achieving a balance between feature selection and model performance.

Following the simulation results, we update the statistical data of all nodes along the selected path. This process refines the quality metric of each node, reflecting the updated performance of the feature subsets. The update rule for the $Q$ value of node $v$ is given by

$$Q(v) \leftarrow \frac{n(v) \cdot Q(v) + \mathcal{P}(F)}{n(v) + 1}, \tag{6}$$

where $Q(v)$ is the updated quality metric of node $v$, ($n(v)$) is the number of times node $v$ has been visited, and $\mathcal{P}(F)$ is the performance of the feature subset $F$ as evaluated during the simulation phase. The visit count $n(v)$ is updated accordingly to reflect this new information.

The MCTS process is terminated on the basis of predefined conditions, which may include reaching a set number of iterations, encountering computational resource limitations, or observing that performance improvements are no longer significant. These criteria ensure the efficient completion of the search process.

The final step in our methodology involves determining the most important feature subset from the search results. Typically, this involves selecting the feature subset corresponding to the node with the highest Q value. This selection represents the optimal balance of feature representation and model performance, as indicated by the statistical data accumulated from the MCTS process.

### 3.4 Algorithm

The MCTS Algorithm1 for feature selection initializes a tree with a root node representing an empty feature set and expands it by adding unvisited features. Simulations are performed to evaluate feature subsets, and results are backpropagated to update nodes, with the process repeated for a set number of iterations to find the optimal feature subset.

## 4. Experiments

### 4.1 Datasets

We explore three distinct datasets to address unique prediction challenges in the realms of finance, socioeconomics, and law enforcement. The GCD provides a comprehensive profile of 1000 bank account holders, encompassing 20 attributes that discern creditworthiness. We employ this dataset to predict good and bad credit risks, offering insights into financial reliability. AID contains 45222 instances with demographic and socioeconomic attributes. Our task is to forecast annual incomes surpassing $50000, unveiling patterns in income disparity. Lastly, SQFD presents a poignant analysis of 72548 encounters with the New York Police Department, where we predict the likelihood of a frisk based on a myriad of individual and procedural factors. Together, these datasets form a tapestry of contemporary societal issues, ripe for data-driven storytelling and impactful analysis.

### 4.2 Setup

We utilized a decision tree regressor, NN, and SVM, implementing these algorithms using the PyTorch library. Each dataset was divided into training and test data. We trained an ML model on the training data and generated explanations using our algorithm. The top explanations for each dataset are reported. For more detailed results on the feature sets, refer to the experiment

Algorithm 1
Monte Carlo Tree Search for Feature Selection.

**Input:** Dataset $\mathcal{D}$ ; Feature set $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$; Exploration parameter $C$; Weight parameter $w$; Maximum iterations $T$
**Output:** Optimal feature subset $F^*$
**Procedure** *Initialize*(·)
  Initialize root node $v_0$with feature subset $F(v_0)=\varnothing$
  **For All** nodes $v$ in $T$ **do**
    $Q(v)\leftarrow 0$, $n(v)\leftarrow 0$
  **End for**
**End Procedure**
Procedure *Expand* ($v$)
  Select the node $v$ with the highest UCT score for expansion
  **For All** unvisited features $f$ in Feature Candidates **do**
    Create child node $v'$ with feature set $F(v') = F(v) \cup \{f\}$
    Attach $v'$ to $v$
  **End For**
**End Procedure**
**Procedure** *Simulate* ($v$)
  Starting from node $v$, perform random walks to a leaf node $l$
  Evaluate the performance of the feature subset $F(l)$ on the holdout dataset
  Update Performance ($F(l)$) using the simulation results
**End Procedure**
**Procedure** *Backpropagate* ($v$,*result*)
  **For All** nodes $u$ on the path from $v$ to the root **do**
    $$Q(u) \leftarrow \frac{n(u)\cdot Q(u) + result}{n(u)+1}$$
    $n(u)\leftarrow n(u)+1$
  **End For**
**End Procedure**
($F^*\leftarrow \varnothing$ )
**For** ($t$) from 1 to $T$ **do**
  $v$ = root of the MCTS tree
  **While** $v$ has unvisited children **do**
    $v$ = child of $v$ selected according to UCT
  **End While** Expand($v$)
  $l$ = result of Simulate($v$) Backpropagate ($v$, Performance($F(l)$))
  **If** $Q(v) > Q(F^*)$ **then**
    $F^*\leftarrow F(v)$
  **End If**
**End For**

sections. For a fair comparison, we used Gopher as our baseline. Additionally, we trained the FO–tree (decision tree regressor), an NN, and an SVM using our method for comparison.

In this study, we introduce a novel approach to feature subset selection, tailored for MCTS algorithms. We establish an analogy between feature and node subsets, leading to the development of refined evaluation metrics that better capture the impact of feature selection on model performance. Here, we redefine three key metrics: feature subset sparsity (FSS), harmonic fidelity (HF), and fairness (EO, SP) to better suit the feature selection process.

### 4.2.1 FSS

The FSS metric quantifies the compactness of a feature subset, adapted from graph theory.

$$FSS(S) = 1 - \frac{|S|}{|F_{total}|}, \tag{7}$$

where $|S|$ is the number of features in the subset $S$ and $|F_{total}|$ is the total number of features.

### 4.2.2 HF

The feature subset fidelity (FSF) metric quantifies the impact of a feature subset on the model's predictive performance. It contrasts with node removal by assessing the effect of adding a feature subset to the current set. The formula is given by

$$Fidelity(F,S) = \left[f(F)\right]_c^* - \left[f(F \cup S)\right]_c^*. \tag{8}$$

Here, $F$ represents the current feature set, $S$ is the feature subset under evaluation, and $f$ is the model's prediction function. The notation $\left[x\right]_c^*$ denotes the optimal performance of the model under configuration $c$. Similar to FSF, the feature subset inverse fidelity (FSIF) metric evaluates the impact on model performance when the feature subset $S$ is integrated into the current feature set $F$. The formula is

$$FidelityInv(F,S) = \left[f(F)\right]_c^* - \left[f(S)\right]. \tag{9}$$

The HF of a feature subset is a composite metric that integrates FSF in a harmonious manner.

$$N - FSN(F,S):m1 = FSF(F,S) \times FSS(F,S) \tag{10}$$

$$N - FSIF(F,S):m2 = FSIF(F,S) \times (1 - FSS(F,S)) \tag{11}$$

$$HF(F,S) = \frac{(1+m1)(1-m2)}{(2+m1-m2)} \tag{12}$$

This metric provides a holistic measure of a feature subset's performance, considering both the standalone and combined effects of feature subsets on model fidelity.

### 4.2.3 Fairness (EO, SP)

We incorporate fairness metrics to ensure equitable feature importance: SP is quantified as the difference in the probability of a positive outcome across groups, aiming for zero disparity.

$$SP = \left| Pr(Outcome_{+} \left| Group_i \right.) - Pr(Outcome_{+} \left| Group_i \right.) \right| \tag{13}$$

Equal opportunity (EO) ensures equal true positive rates for all groups.

$$EO = TPER(Group_i) = TPR(Group_i) \tag{14}$$

These metrics prioritize features that minimize SP and EO disparities, fostering a fair decision–making process and steering towards models that are not only effective but also just.

### 4.3 Experiment

The results of the experimental analysis are presented in Table 1. Table 1 enables a comprehensive comparison of our proposed methods against several baselines across three different datasets: GCD, AID, and SQFD. The metrics used for evaluation include accuracy (ACC), HF, and two variations of the difference in performance metrics, Δ*SP* and Δ*EO*, where lower values are preferable.

Upon examining Table 1, it is evident that our proposed method, Ours, consistently outperforms the baselines across all datasets. Specifically, Ours achieves the highest ACC and HF scores, indicating superior classification accuracy and the ability to identify a high–fidelity subset of the data. Additionally, Ours demonstrates the lowest values for Δ*SP* and Δ*EO*, suggesting that our method is more robust in maintaining performance across different subgroups and overall, respectively.

For GCD, Ours attains the highest ACC (79.98) and HF (0.49), while also achieving the lowest Δ*SP* (7.49) and Δ*EO* (5.38), indicating the best combined performance among all compared methods. In AID, Ours again produces the highest ACC (76.66) and HF (0.50), together with the lowest fairness disparities (Δ*SP* =6.15, Δ*EO* = 4.47). In SQFD, Ours yields the

Table 1
Results of our proposed method, Ours, with the baselines (%). This table enables a detailed comparative analysis of the performance metrics between "Ours" and several baseline methods across three distinct datasets: GCD, AID, and SQFD. The metrics include the ACC percentage, which indicates the model's ability to distinguish between classes; HF, which measures the frequency of correct predictions; and the changes in SP and EO, which reflect the model's precision and error rate, respectively. The table is organized to facilitate the comparison of these key performance indicators, highlighting the relative strengths and potential areas for improvement in each method.

| | GCD | | | | AID | | | | SQFD | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Method | *ACC*% (↑) | *HF* (↑) | Δ*SP* (↓) | Δ*EO* (↓) | *ACC*% (↑) | *HF* (↑) | Δ*SP* (↓) | Δ*EO* (↓) | *ACC*% (↑) | *HF* (↑) | Δ*SP* (↓) | Δ*EO* (↓) |
| FO–tree | 76.22 | 0.46 | 10.24 | 8.20 | 62.67 | 0.46 | 19.36 | 16.67 | 58.61 | 0.43 | 7.50 | 6.89 |
| NN | 77.51 | 0.46 | 9.63 | 7.70 | 62.71 | 0.46 | 18.81 | 16.15 | 60.35 | 0.44 | 6.18 | 5.75 |
| SVM | 76.18 | 0.45 | 10.97 | 8.54 | 62.53 | 0.46 | 21.51 | 19.85 | 56.28 | 0.43 | 8.14 | 7.42 |
| Gopher | 79.04 | 0.46 | 9.42 | 8.17 | 63.32 | 0.47 | 17.46 | 15.38 | 60.89 | 0.44 | 7.20 | 6.15 |
| Ours–FO–tree | 76.39 | 0.47 | 9.67 | 7.96 | 72.16 | 0.47 | 9.26 | 9.14 | 60.03 | 0.44 | 5.94 | 5.03 |
| Ours–NN | 78.07 | 0.48 | 7.91 | 6.24 | 75.29 | 0.48 | 7.53 | 5.97 | 62.42 | 0.46 | 5.21 | 4.37 |
| Ours–SVM | 76.95 | 0.46 | 8.85 | 7.64 | 73.71 | 0.47 | 9.28 | 8.51 | 60.16 | 0.44 | 5.57 | 4.39 |
| Ours | 79.98 | 0.49 | 7.49 | 5.38 | 76.66 | 0.50 | 6.15 | 4.47 | 63.52 | 0.47 | 5.06 | 4.15 |

best ACC (63.52), the highest HF (0.47), and the lowest $\Delta SP$ and $\Delta EO$ (5.06 and 4.15, respectively). Furthermore, the variants of our method, Ours–FO–tree, Ours–NN, and Ours–SVM, also show competitive results, suggesting that the integration of our proposed techniques into different ML models can enhance their performance across various metrics.

In summary, the results presented in Table 1 demonstrate the effectiveness of our proposed method in achieving higher accuracy and fairness in classification tasks. The consistent outperformance across different datasets and metrics highlights the robustness and generalizability of our approach.

### 4.4 Feature set result

We evaluated our proposed method on three datasets: AID, GCD, and SQFD, each addressing different aspects of model bias.

For AID, which focuses on gender bias, our method demonstrated superior performance compared with the Gopher baseline. The sensitive attribute "gender" exhibited high importance in the feature set (Fig. 2). Our approach, leveraging influence functions, selected a model near the optimal parameters. While Gopher identified some patterns that reduced model bias to a certain extent, our method returned four attributes with higher accuracy and fairer SP values (76.66, 6.15%). Notably, our method's single predicate ([gender = female]) aligned with Gopher's selection, while in choosing additional attributes on the basis of our method's interactivity and redundancy criteria, our method outperformed Gopher, as further explored in our ablation study.

Regarding GCD, which exhibits age-related bias in credit risk assessment, Fig. 2 shows the top explanations generated by our method, containing up to four predicates each (including their support and true influence), ranked by importance scores. While Gopher reduced overall model bias by removing training data points of older individuals marked as low credit risk, our method did not remove biased data. Consequently, our explanations for this dataset included predicates of the sensitive attribute, highlighting its importance in bias reduction. The resulting (accuracy, SP) values were (78.98, 7.49%). For SQFD, which highlights the NYPD's disproportionate stop, question, and frisk practices towards Black individuals, our top three explanations (Fig. 2) identified patterns of protected groups being searched and privileged groups not being searched. Our method's ability to find associations between different attributes and remove redundant ones led to a reduced correlation between privileged groups and "no search" outcomes, thereby

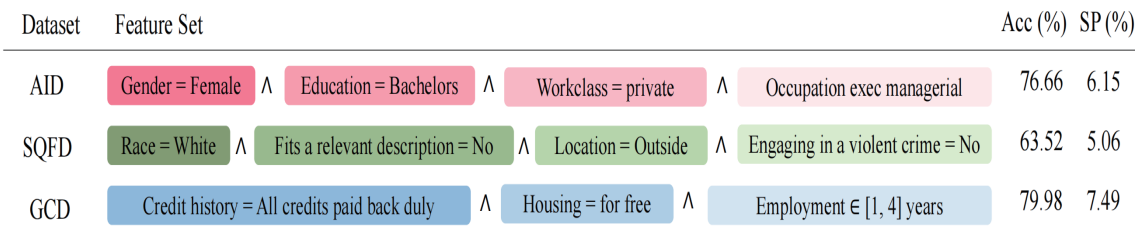


Fig. 2. (Color online) Summary of three datasets, highlighting the optimal feature sets and their corresponding accuracies and SP.

mitigating model bias. The most important explanation generated by our method, (race=white) ∧ (fits a relevant description=no) ∧ (location=outside) ∧ (engaging in a violent crime=no), achieved comparable accuracy (63.52%) and fairness (5.06%) to our top explanation.

These results demonstrate the effectiveness of our proposed method in identifying and mitigating biases across diverse datasets, and show that our method outperforms the Gopher baseline in terms of both accuracy and fairness metrics.

We report the average time expenditure across eight distinct methodologies, as depicted in Fig. 3. The FO–tree method generally lags behind the other methods by a factor of 5 to 10 times in terms of efficiency. We observe that, while the FO–tree method provides a valid estimation in single–step updates for assessing the efficiency impact of model parameters on predictive outcomes, it is significantly slower than the NN and SVM analysis methods when estimating the effect of parameter subsets on model performance. It is noteworthy that the initial model parameters were employed to expedite the retraining process, thus reducing the time cost associated with retraining to a level faster than that of single–step gradient descent. We conclude that when retraining the model is not a viable option, the FO–tree, NN, SVM, and Gopher methods offer alternative solutions. These methods are capable of providing effective estimations of the impact of model parameters on specific optimization problems across various scenarios, albeit with the potential need for additional adjustments and computational resources.

### 4.5 Analysis and explanations

In our comprehensive ablation study, we meticulously dissected the model to evaluate the significance of each of its components. By incrementally removing or modifying key elements, we aimed to isolate the effects of individual features on the overall performance. Ours served as the baseline, establishing a benchmark against which the performance of the modified models was compared. The Ours–FO–tree variant incorporated a feature optimization technique designed to enhance the selection of relevant variables. The Ours–NN model integrated neural network components, leveraging their capacity for capturing complex patterns within the data. Lastly, the Ours–SVM model combined the strengths of SVMs, known for their effectiveness in high–dimensional spaces. Through a series of controlled experiments, we systematically varied

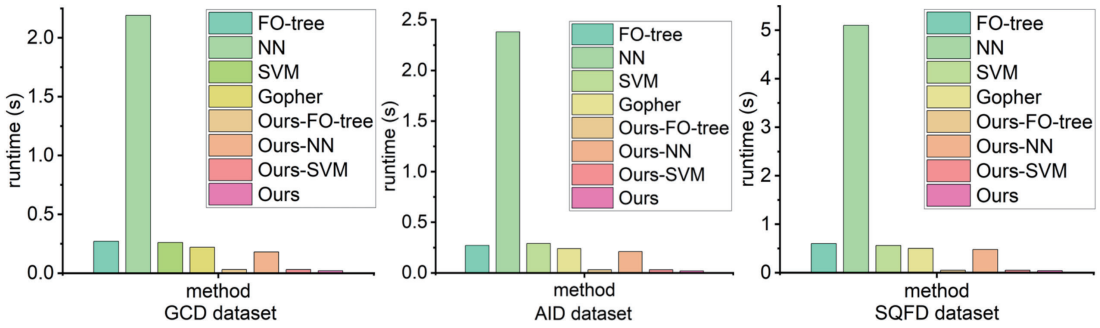


Fig. 3. (Color online) Runtime for computing feature subsets of three different datasets. Our method approximations over 30 runs are significantly faster than baseline methods.

to observe how the model's performance fluctuated with changes in dataset size or composition. The results, as depicted in Fig. 4, underscore the nuanced interplay between model architecture and dataset characteristics, providing valuable insights into the factors that drive the model's predictive power.

We also perform an ablation analysis on the three datasets, GCD, AID, and SQFD, to evaluate the impact of different feature sets on model fairness and accuracy. The objective is to identify the most influential features that contribute to bias and to test the effectiveness of our update mechanism in reducing such bias. The ablation study revealed significant differences in the performance of models when specific features are altered. Notably, our update mechanism was able to reduce bias in all datasets, with varying degrees of success. Our update mechanism involves adjusting the values of specific features within the dataset to reflect a more equitable representation. This process is aimed at minimizing the bias without significantly compromising the model's predictive accuracy.

The updates led to a notable reduction in bias across all datasets. For instance, in GCD, updating the Credit history feature to reflect all credits paid back duly resulted in a bias reduction of 7.49%, as shown in Fig. 5. When comparing the updated models with their baseline

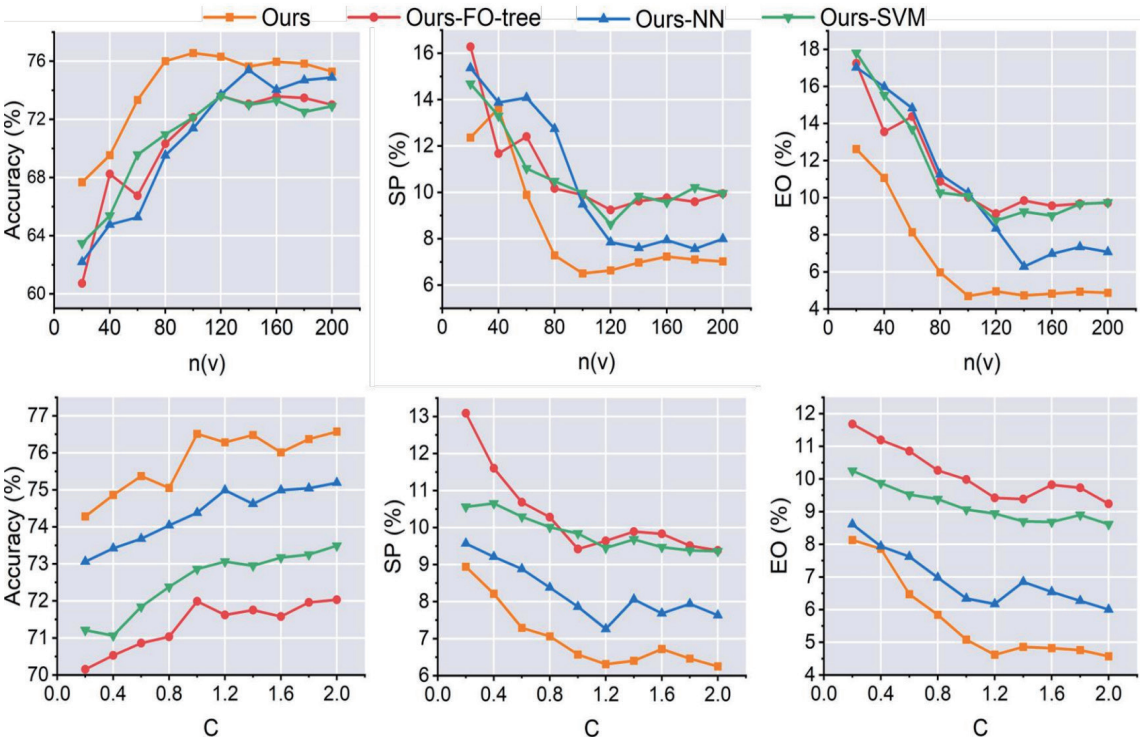


Fig. 4. (Color online) Results of an ablation study examining the impact of individual components on the performance of our proposed model. The x–axis represents the variable n(v) or C, which represents variables or instances in the dataset. The y–axes display performance accuracy. The curves correspond to different model configurations: Ours (base model), Ours–FO–tree (model with feature optimization), Ours–NN (model with NN enhancements), and Ours–SVM (model with SVM integration).

counterparts, our approach demonstrated a consistent improvement in fairness without any substantial loss in accuracy. For example, Fig. 6 shows a bias reduction of 12.74% for our model compared with the Gopher model.

In GCD, updating the Employment feature to reflect a range of 1 to 4 years, along with Residence and Credit history, improved accuracy to 79.98% with a bias reduction of 7.49%. In AID, focusing on Gender and Education as influential features resulted in a bias reduction of 6.15%.

In Fig. 7, the results of SQFD experiments further reinforced the effectiveness of our method. When considering race, criminal activity, location, and descriptive fit, we achieved a fairness score of 63.52% with a minimal bias of 5.06%. This surpassed Gopher's performance, which yielded a fairness score of 60.89% and a bias of 7.20% when evaluating similar attributes but replacing location and descriptive fit with victim casing and proximity to the offense scene. Our method's robustness was evident as it maintained strong performance (fairness: 61.09%, bias: 6.72%) even when adjusting for victim casing.

| Methods | Feature Set | Acc (%) | SP (%) |
|---|---|---|---|
| Gopher | Employment ∈ [1, 4] years ∧ Residence = 2 years ∧ Installment rate = 4% ∧ Debtors = None | 79.04 | 9.42 |
| Ours | Employment ∈ [1, 4] years ∧ Residence = 2 years ∧ Installment rate = 4% ∧ Credit history = All credits paid back duly | 79.26 | 8.84 |
| Ours | Employment ∈ [1, 4] years ∧ Residence = 2 years ∧ Credit history = All credits paid back duly ∧ Housing= for free | 79.98 | 7.49 |

Fig. 5. (Color online) Results of ablation study on GCD.

| Methods | Feature Set | Acc (%) | SP (%) |
|---|---|---|---|
| Gopher | Gender = Female ∧ Marital = Never married ∧ Relationship ∈/ [Wife, Husband] | 63.32 | 17.46 |
| Ours | Gender = Female ∧ Marital = Never married ∧ Education = Bachelors | 68.61 | 12.74 |
| Ours | Gender= Female ∧ Education = Bachelors ∧ Workclass=private | 76.66 | 6.15 |

Fig. 6. (Color online) Results of ablation study on AID.

| Methods | Feature Set | Acc (%) | SP (%) |
|---|---|---|---|
| Gopher | Race = White ∧ Engaging in a violent crime=No ∧ Casing a victim = Yes ∧ Proximity to scene of offense = No | 60.89 | 7.20 |
| Ours | Race = White ∧ Engaging in a violent crime=No ∧ Casing a victim = Yes ∧ Fits a relevant description = No | 61.09 | 6.72 |
| Ours | Race = White ∧ Engaging in a violent crime=No ∧ Location = Outside ∧ Fits a relevant description = No | 63.52 | 5.06 |

Fig. 7. (Color online) Results of ablation study on SQFD.

## 5. Conclusions

In this paper, we introduced PRIMCT, a method that significantly advances the field of fair and explainable ML. By integrating feature importance, redundancy, and interaction, PRIMCT offers a nuanced comprehension of bias, supported by a robust framework for its mitigation. Our experimental evaluations on GCD, AID, and SQFD not only validate the effectiveness of PRIMCT in enhancing model fairness and interpretability but also demonstrate its superiority over existing XAI techniques. On AID, PRIMCT improved ACC by 13.34% and reduced $\Delta SP$ by 11.31%, while also achieving consistent improvements in fairness-related metrics across all three datasets. PRIMCT proves its mettle in real–world applications. The method's ability to provide actionable insights for model enhancement, coupled with its rigorous empirical validation, solidifies its potential to redefine fairness and interpretability in ML.

## Acknowledgments

This research was supported in part by the "Lingyan" R&D Program of Zhejiang Province under Grant 2026C02A1248, in part by the Natural Science Foundation of Hangzhou under Grant 2025SZRJJ2340.

## Contribution

Jinchao Ge: Writing—review & editing; Writing—original draft; Visualization; Software; Methodology; Formal analysis; Conceptualization. Tao Du: Writing—review & editing; Resources; Visualization; Software; Data curation. Haidong Li: Resources; Data curation; Project support; Technical consultation. Nan Ju: Writing—review & editing; Data curation; Experimental validation. Shuwen Zhao: Review & supervision; Project administration; Funding acquisition; Project conceptualization; Team coordination.

## References

1 I. A. Zahid, S. Garfan, M. A. Chyad, A. S. Albahri, O. S. Albahri, A. H. Alamoodi, M. Deveci, R. Z. Homod, and L. Alzubaidi: IEEE Trans. Emerg. Top. Comput. Intell. **9** (2025) 3728. https://doi.org/10.1109/TETCI.2025.3567604
2 L. Capogrosso, F. Cunico, D. S. Cheng, F. Fummi, and M. Cristani: IEEE Access **12** (2024) 23406. https://doi.org/10.1109/ACCESS.2024.3365349
3 Financial Stability Board. The Financial Stability Implications of Artificial Intelligence. Financial Stability Board (FSB) 2024.
4 M. Mersha, K. Lam, J. Wood, A. K. AlShami, and J. Kalita: Neurocomputing **599** (2024) 128111. https://doi.org/10.1016/j.neucom.2024.128111
5 B. Cravens, A. Lensen, P. Maddigan, and B. Xue: IEEE Trans. Emerg. Top. Comput. Intell. **10** (2026) 676. https://doi.org/10.1109/TETCI.2025.3561666
6 L. Deck, J. Schoeffer, M. De-Arteaga, and N. Kühl: Proc. 2024 ACM Conf. Fairness, Accountability, and Transparency (FAccT '24) (ACM, New York, 2024) 1579–1595. https://doi.org/10.1145/3630106.3658990
7 H. Hwang, A. Bell, J. Fonseca, V. Pliatsika, J. Stoyanovich, and S. E. Whang: Proc. 2025 ACM Conf. Fairness, Accountability, and Transparency (FAccT '25) (ACM, New York, 2025) 1588–1601. https://doi.org/10.1145/3715275.3732105

8 X. Li, G. Ma, H. Chen, L. Zhang, J.–H. He, S. Liu, N. Wang, S. Wang, L. Wang, and G. Liu: Heart Mind **9** (2025) 328. https://doi.org/10.4103/hm.HM-D-24-00068
9 R. Pradhan, J. Zhu, B. Glavic, and B. Salimi: Proc. 2022 Int. Conf. Manage. Data (ACM, New York, 2022) 247–261. https://doi.org/10.1145/3514221.3517886
10 T. Surve and R. Pradhan: Proc. 28th Int. Conf. Extending Database Technol. (EDBT 2025) (OpenProceedings.org, 2025) 623–635. https://doi.org/10.48786/EDBT.2025.50
11 S. Nathawat, R. Sharma, A. Rani, and M. Mahadevaswamy: Heart Mind **9** (2025) 482. https://doi.org/10.4103/hm.HM-D-25-00023
12 C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel: Proc. 3rd Innovations Theor. Comput. Sci. Conf. (ACM, New York, 2012) 214–226. https://doi.org/10.1145/2090236.2090255
13 S. Caton and C. Haas: ACM Comput. Surv. **56** (2024) Article 166, 1–38. https://doi.org/10.1145/3616865
14 K. Makhlouf, S. Zhioua, and C. Palamidessi: J. Log. Algebr. Methods Program. **141** (2024) 101000. https://doi.org/10.1016/j.jlamp.2024.101000
15 Y. Huang, Z. Tang, and X. Chang: AdapFair: arXiv (2024) arXiv:2409.15088. https://doi.org/10.48550/arXiv.2409.15088
16 T. Calders, F. Kamiran, and M. Pechenizkiy: Proc. 2009 IEEE Int. Conf. Data Min. Workshops (IEEE, Piscataway, 2009) 13–18. https://doi.org/10.1109/ICDMW.2009.83
17 M. Hardt, E. Price, and N. Srebro: Adv. Neural Inf. Process. Syst. **29** (2016) 3315
18 W. Zhang, T. Hernandez–Boussard, and J. Weiss: Proc. AAAI Conf. Artif. Intell. **37** (2023) 14611–14619. https://doi.org/10.1609/aaai.v37i12.26708
19 M. Zehlike, A. Loosley, H. Jonsson, E. Wiedemann, and P. Hacker: Artif. Intell. **340** (2025) 104280. https://doi.org/10.1016/j.artint.2024.104280
20 N. Ullah, J. A. Khan, I. De Falco, and G. Sannino: ACM Comput. Surv. **57** (2024) Article 94, 1–36. https://doi.org/10.1145/3705724
21 T. Nguyen, A. Canossa, and J. Zhu: arXiv (2024) arXiv:2403.14496. https://doi.org/10.48550/arXiv.2403.14496
22 G. Carloni, A. Berti, and S. Colantonio: Data Min. Knowl. Discov. **15** (2025) e70015. https://doi.org/10.1002/widm.70015
23 P. Liu, L. Jiang, H. Lin, J. Hu, S. Garg, and M. Alrashoud: IEEE Trans. Consum. Electron. **70** (2024) 4564. https://doi.org/10.1109/TCE.2023.3339630
24 J. Lu, Y. Sheng, S. Cao, S. Elnaffar, M. M. Saad, A. M. Seid, and A. Erbad: IEEE Trans. Consum. Electron. **70** (2024) 7334. https://doi.org/10.1109/TCE.2024.3397863
25 K. Li, Z. Dai, X. Wang, Y. Song, and G. Jeon: IEEE Trans. Consum. Electron. **70** (2024) 6174. https://doi.org/10.1109/TCE.2024.3387557
26 M. F. Saiyed, I. S. Al–Anbagi, and M. S. Hossain: IEEE Trans. Consum. Electron. **71** (2025) 6839. https://doi.org/10.1109/TCE.2024.3482092
27 D. L. Sinclair and S. Sussman: Heart Mind **9** (2025) 426. https://doi.org/10.4103/hm.HM-D-24-00080
28 P. Knab, S. Marton, U. Schlegel, and C. Bartelt: Which LIME Should I Trust? Concepts, Challenges, and Solutions. Communications in Computer and Information Science (2025) Vol. 2577. https://doi.org/10.1007/978-3-032-08324–1_2
29 J. Jiang, F. Leofante, A. Rago, and F. Toni: Proc. 33rd Int. Joint Conf. Artif. Intell. (IJCAI–24) (2024) 8086–8094. https://doi.org/10.24963/ijcai.2024/894
30 S. Upadhyay, H. Lakkaraju, and K. Z. Gajos: Proc. 30th Int. Conf. Intell. User Interfaces (IUI '25) (ACM, New York, 2025) 446–462. https://doi.org/10.1145/3708359.3712095
31 J. Marques–Silva: arXiv (2024) arXiv:2406.11873. https://doi.org/10.48550/arXiv.2406.11873
32 X. Zhang and L. Yu: Expert Syst. Appl. **237** (2024) 121484. https://doi.org/10.1016/j.eswa.2023.121484
33 G. Valdrighi, A. M. Ribeiro, J. S. B. Pereira, V. Guardieiro, A. Hendricks, D. Miranda Filho, J. D. Nieto Garcia, F. F. Bocca, T. B. Veronese, L. Wanner, and M. M. Raimundo: Neural Comput. Appl. **37** (2025) 20781. https://doi.org/10.1007/s00521-025-11520-y
34 H. Ayari, R. Guetari, and N. Kraïem: Artif. Intell. Rev. **59** (2026) Article 13. https://doi.org/10.1007/s10462-025-11416-2
35 E. Rodrigo-Bonet and N. Deligiannis: IEEE Trans. Emerg. Top. Comput. Intell. **9** (2025) 281. https://doi.org/10.1109/TETCI.2024.3419714