2022 IEEE International Conference on
Systems, Man, and Cybernetics (SMC)

2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC) | 978-1-6654-5258-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SMC53654.2022.9945130

EADTC: An Approach to Interpretable and Accurate Crime Prediction

Yujunrong Ma, Kiminori Nakamura, Eung-Joo Lee, and Shuvra S. Bhattacharyya

***Abstract*— Machine learning applications related to high-stakes decisions are often surrounded by significant amounts of controversy. This has led to increasing interest in interpretable machine learning models. A well-known class of interpretable models is that of decision trees (DTs), which mirror a common strategy used by humans to arrive at solutions through a series of well-defined decisions. However, much of previous research on DTs for criminal justice predictions has focused primarily on collections (ensembles) of DTs whose results are aggregated together. Such DT ensembles are used to help improve accuracy; however, their increased complexity and deviation from human decision-making processes makes them much less interpretable compared to single-DT approaches. In this paper, we present a new DT model for criminal recidivism prediction that is designed with high interpretability, accuracy, and fairness as core objectives. The interpretability of the model stems from its formulation in terms of a single DT structure, while accuracy is achieved through an intensive optimization process of DT parameters that is carried out using a novel evolutionary algorithm. Through extensive experiments, we analyze the performance of our proposed EADTC (Evolutionary Algorithm Decision Tree for Crime prediction) method on relevant datasets. Our experiments show that the EADTC approach achieves competitive accuracy and fairness with respect to state-of-the-art ensemble DT models, while achieving higher interpretability due to the simpler, single-DT structure.**

I. Introduction

Machine learning techniques have been increasingly studied and deployed in high-stakes decision making contexts, where the safety and well being of the public, especially vulnerable members, can be impacted, such as medical diagnosis [1], health care [2], and criminal justice [3]. In criminal justice settings, risk assessment, which assesses the likelihood of criminal recidivism and other adverse outcomes, has been an area that has received the most attention in terms of machine learning research. While machine learning techniques can improve predictive accuracy and promote better outcomes for disadvantaged populations over human judgment [4], the techniques have also been a source of concern, especially due to their limited transparency [5]. In recent years, there has been increasing attention to the transparency and interpretability of machine learning models, precisely because limited interpretability, or the difficulty for humans to comprehend why and how predictions are made, can diminish the credibility and legitimacy of decisions based on these models [6]. Even models that provide very high accuracy can fail to be convincing if their operational process is encapsulated within a black box or is otherwise unexplainable.

There have been increasing research efforts on improving the interpretability of machine learning models. These efforts can be classified into two main types. The first type focuses on explaining black box models. Some studies have used more interpretable models, such as shallow feed-forward networks and decision trees, to mimic and explain how the black-box models work [7] [8], while others have used techniques such as guided class activation maps to analyze the models post-hoc [9]. The second type of effort focuses on directly designing models that are inherently interpretable [10] [11]. Unlike the models that mimic the output of black-box models but still do not allow for the understanding of how predictions are generated, these interpretable models provide predictions that can be easily traceable to well-defined decision rules.

In this paper, we present a new decision tree (DT) model for recidivism prediction that is designed to provide high interpretability while maintaining competitive predictive accuracy. DTs have been demonstrated to have various advantages for interpretability: 1) the hierarchical sequence of well-defined decision rules in a DT is completely transparent and similar to multi-step decision-making processes used by humans; 2) the complexity of a DT is easy to control (e.g., by setting the maximum depth of the tree); and 3) the feature selection process can be automated, which safeguards against subjective bias that can be introduced by manual filtering and configuration of features.

Prior development of DT-based models shows that ensemble methods (e.g., random forest, gradient boosting), which incorporate multiple trees and aggregate the predictions produced from the different trees, can have a better regularization performance — leading to increased predictive accuracy — compared to single-DT models (e.g., see [12], [13], [14]). However, their increased complexity and deviation from human decision-making processes makes ensemble-DT models much less interpretable compared to single-DT approaches.

The DT method that we introduce in this paper is designed as a single-DT model to provide high interpretability. To overcome the potential degradation in accuracy compared to ensemble-DT models, we incorporate a novel evolutionary algorithm approach to intensively optimize parameters of the proposed single-DT framework. We refer to our integrated

Y. Ma is with the Dept. of Electrical and Computer Engineering, University of Maryland, USA mayu1996@umd.edu.

S. S Bhattacharyya is with the Dept. of Electrical and Computer Engineering, Institute for Advanced Computer Studies, and Maryland Crime Research and Innovation Center, University of Maryland, USA ssb@umd.edu.

K. Nakamura is with the Dept. of Criminology and Criminal Justice, University of Maryland, USA knakamur@umd.edu.

E. Lee is with CAMCA, Dept. of Radiology, MGH and Harvard Medical School, USA elee66@mgh.harvard.edu.

evolutionary algorithm and DT method as the EADTC (Evolutionary Algorithm Decision Tree for Crime prediction) method.

Evolutionary algorithms are population-based optimization algorithms that are inspired from natural selection in biological species, and are effective in their ability to escape local optima. An evolutionary algorithm is initialized with a random population of candidate solutions. This population then evolves for multiple generations until a given stopping criterion is reached. A fitness function, which is an important design component of an evolutionary algorithm, is used to assess the quality of a given candidate solution. The process of deriving the next generation G_{i+1} of an evolutionary algorithm population from the current generation G_i generally involves selecting pairs of candidate solutions (parents) from G_i, and deriving one or more candidate solutions (children) in G_{i+1} from each pair of selected parents. The fitness function is used to guide the selection of parents and other aspects in the evolution of generations. For more details on evolutionary algorithms, we refer the reader to [15].

In our prototyping and experiments presented in this paper, we focus on the task of predicting criminal recidivism, which typically refers to the return to criminal behavior or other undesirable outcomes after contact with the criminal justice system. In the remainder of this paper, we assume the context of recidivism when defining details of the EADTC approach, and presenting experiments that demonstrate the approach. We present the results on both prediction accuracy and fairness metrics. Although the developments in this paper are focused on recidivism, the underlying DT framework is flexible, and we envision that it can be adapted to other predictive applications in criminal justice.

II. Background and Related work

In this section, we first discuss defining elements of model interpretability in the literature and how they vary across most commonly used machine learning models, with special attention to applications in criminal justice settings. We then discuss the application of evolutionary algorithms to machine learning models, followed by a discussion on fairness metrics.

A. Interpretability

While the concept of interpretability can be domain specific, three distinct aspects have been identified [16]: (1) Simulatability: the ability that a model allows humans to use input data together with a model to reproduce every calculation step necessary to make the prediction. For example, this allows humans to understand changes in model parameters caused by the training data. (2) Decomposability: the feature that a model can provide intuitive explanations for all of the model parameters. (3) Algorithmic transparency: the ability to explain the working of the training algorithm that optimizes the parameters of a model. In the following section, we review several classes of machine learning models that are commonly used for recidivism prediction in light of these aspects of interpretability.

B. Commonly-Used Models

(1) Logistic Regression. Unlike the usual hard-label output of most predictive models, the probability output of logistic regression makes it easier to reason about relative risk levels between different input instances. Furthermore, the interpretability of logistic regression is considered as being in the top-tier. This is because in a logistic regression model it is inherently clear how much weight is being given to each feature in the model. An important variant of logistic regression in the context of interpretable recidivism prediction is the concept of scoring systems, such as RiskSLIM, where feature weights are constrained to be integers [10]. Such scoring systems can improve interpretability, especially decomposability, since integer-valued weights are generally easier for humans to understand compared to weights that can have arbitrary real values.

(2) Decision Trees. A decision tree (DT) is a decision support tool that uses a branching model of decisions and their possible consequences. A DT is a graphical, tree-based model in which vertices represent decisions, and edges from a parent vertex to its children correspond to different decision outcomes associated with the parent vertex [17]. For example, random forests, an ensemble-DT method, has been applied to predict domestic violence for informing pretrial release decisions [18]. When examining a prediction produced by this approach, one can see how the prediction has been arrived at by examining selected paths in the random forest that are associated with the prediction. In particular, there is one path to examine through each constituent tree whose output agrees with the prediction result of the ensemble. This property of identifying path-sets for prediction results provides good simulatability. However, compared to single-DT models, interpretability of ensemble models is significantly weaker. This is in part because the relationship between multiple paths within a given path-set is in general not clear for a given prediction result.

(3) Boosting-based Models. Models that incorporate boosting optimization are among the best performing models on tabular data (e.g., see [13]). Boosting models involve the integration of relatively low-complexity "weak learners" into higher accuracy "strong learners" that combine the results of the weak learners. The integration of weak learners into a strong learner is performed through an iterative optimization process. Despite their relatively high accuracy, the simulatability of boosting-based methods is considered to be relatively weak. This is because the iterative optimization process and the process of combining results from different weak learners introduce complexities that make the resulting models difficult to interpret.

(4) Deep Neural Networks (DNNs). DNNs and their variants are widely-used in many application areas, such as computer vision and natural language processing, to name a few. In recent years, DNNs have been applied to crime prediction (e.g., see [19]). However, the underlying mechanisms of DNNs are usually considered as being non-interpretable. This is due in part to the uncertainty and ambiguity introduced

by non-linearities in the employed computational process. Non-linear layers, such as Rectified Linear Unit (ReLU) layers, can significantly increase the expressive power of DNNs, but the introduction of such components comes with a corresponding decrease in interpretability. Moreover, DNNs typically involve very large numbers of parameters, which further degrades their interpretability — both in terms of decomposability and simulatability.

C. Application of Evolutionary Algorithms to Machine Learning

Evolutionary algorithms have been applied to crime prediction independently of machine learning methods. For example, Wu and Grubesic used a multi objective evolutionary algorithm to identify crime hot-spots [20]. On the other hand, evolutionary algorithms have been used to construct DNN models, such as recurrent neural networks (RNNs) [21], convolutional neural networks (CNNs) [22] and hybrid networks [23]. Evolutionary algorithms have also been used to construct rule-based models [24]. Due to characteristics of their structure, tree-based models require special consideration when designing encodings, and crossover and mutation operators for integration with evolutionary algorithms. Issues in the design of encoding and operators for EA-integrated tree models, such as deleting nodes and subtrees, were discussed in [25], and modified operators, such as subtree transplanting, were introduced in [26].

To the best of our knowledge, the EADTC method is the first method for recidivism prediction that integrates evolutionary algorithms and single-DT machine learning models. In particular, the framework of evolutionary algorithms is applied to provide powerful optimization capability that enhances the quality of single-DT crime prediction. The result is a novel method that achieves accuracy and fairness levels that are competitive with state-of-the-art prediction methods based on ensemble-DT models, while providing the significantly enhanced interpretability of single-DT models. Our work leads to a new way of studying machine learning for recidivism prediction in terms of evolutionary algorithms and single DT modeling.

D. Fairness

In addition to accuracy and interpretability, *fairness* is another important criterion for developing predictive models in many application areas, including criminal justice. There is concern that being heavily dependent on machine learning can cause so-called run-away loops, where imbalanced data causes models to produce imbalanced results, which leads to decisions that exacerbate imbalances in the data, which in turn leads to more imbalanced models, and so on. There are different definitions of fairness in the literature. Corbett-Davies et al. [27] describe three popular definitions of algorithmic fairness: statistical parity, conditional statistical parity and predictive equality.

III. Methods

The proposed evolutionary algorithm approach, EADTC, provides an alternative to conventional approaches for single-DT optimization, such as the well-known top-down induction of decision trees (TDIDT) approach [17]. Most of the concepts in genetic algorithms, which is the most popular type of EA, are adopted in the proposed method. EADTC enables exploration of a large search space of possible DT configurations, which is especially useful when considering applications such as crime prediction, where the features involved can be diverse and can be interrelated in complex ways. Indeed, the ability of EADTC to explore a large search space, by leveraging the randomized search process of the general evolutionary algorithm framework, enables intensive optimization of single-DT structures. As we demonstrate in the next section, the optimized single-DT structures produced by EADTC are competitive in performance with more complex (less interpretable) ensemble-DT structures.

A. Decision Tree Modeling in EADTC

When deriving a DT for recidivism using the EADTC approach, we are given a dataset $D = I_1, I_2, \ldots, I_n$ consisting of n instances, where each instance I_i is assigned a binary classification label $L(I_i) \in \{0,1\}$. Intuitively, each I_i corresponds to a unique individual (person) and $L(I_i)$ indicates whether or not the individual experienced recidivism — $L(I_i) = 1$ indicates that individual I_i experienced recidivism, whereas $L(I_i) = 0$ indicates the absence of recidivism. Part of the design of the DT is the definition of a set of features $\{f_1, f_2, \ldots, f_n\}$, where each feature f_i is a mapping of instances: $f_i : D \to \mathscr{R}$, where $\mathscr{R}$ denotes the set of real numbers. For a given instance $I_i \in D$, the vector $F(I_i)$, defined by $F(I_i) = (f_1(I_i), f_2(I_i), \ldots, f_m(I_i))$, is referred to as the *feature vector* associated with I_i. Examples of features that are of relevance in recidivism prediction are an individual's current age, first-crime age, employment status, features related to any history of drug abuse, and violent-crime flag.

Following common practice in the design and implementation of DTs, we restrict our attention to binary DTs — that is, trees in which each internal vertex has exactly two child vertices. For each internal vertex, one of the child vertices is designated as the *left* child, while the other is designated as the *right* child. In the EADTC approach, each internal vertex v in a DT is associated with a unique feature index $k(v) \in \{1,2,\ldots,m\}$, and a decision function $\phi_v : f_{k(v)}(D) \to \{0,1\}$, which effectively maps feature vectors (using only the $k(v)$th component) into decision paths that follow the edge to the the left (0 value) or right (1 value) child vertex, respectively.

In the EADTC approach, each leaf vertex w in a DT is associated with a predicted value $p(w) \in \{0,1\}$. Intuitively, to compute the recidivism prediction result for a given instance I_i, we start at the root vertex $v[0]$ of the DT, compute $z_0 = \phi_{v[0]}(f_{k(v[0])}(I_i))$, and then set the next vertex $v[1]$ to visit by selecting the left or right child of $v[0]$ depending on whether z_0 is 0 or 1, respectively. We then compute $z_1 = \phi_{v[1]}(f_{k(v[1])}(I_i))$, and similarly, set the next vertex $v[2]$ depending on the value of z_1. This process is repeated $(d-3)$ more times, where d is the depth of the tree until we arrive at a leaf vertex $v[d-1]$. The value predicted by the DT model for $I(i)$ is then derived as $p(v[d-1])$.

In our experiments, the DT is trained with some subset of D and tested with the remaining elements of D. We randomly select 75% of the instances for training. More details on our experiments are discussed in Section IV.

B. Evolutionary Algorithm

The design of an evolutionary algorithm can be divided into several key components: the representation, parent selection process, crossover operator, mutation operator, and termination condition. In the remainder of this subsection, we present the design of these components in EADTC. For background on how these components are used in the overall operation of an evolutionary algorithm, we refer the reader to [15].

(1) Representation: Representation concepts include the genotype, phenotype and representation mapping. For example, we may represent an integer (phenotype) as a binary array (genotype): 37 as 00100111 and 116 as 01110100. In EADTC, we adopt a representation that is specifically geared toward decision trees. For a given decision tree $T = \{n_1, n_2, \ldots, n_k\}$, the representation encodes the set of nodes in T, where each node is a tuple of the form $n(d, f, t, l, r)$. Here, d is the depth of n within T, f is the feature associated with n that determines the decision outcome associated with the node, t is the threshold associated with the decision, and l and r are the left and right children of n.

(2) Parent Selection: The parent selection process is used to select parent individuals for crossover operations, where the suitability of a parent individual is quantified by its fitness. In EADTC, we use accuracy as the fitness function and proportional fitness as the selection metric. If $Fit(T_i)$ indicates the fitness of tree T_i, $N_{correct}$ and N_{all} represent the number of correctly classified records and the total number of records, respectively, and $P(T_i)$ is the probability for tree T_i to be selected as a parent, then we can write:

$$Fit(T_i) = \frac{N_{correct}}{N_{all}} \tag{1}$$

$$P(T_i) = \frac{Fit(T_i)}{\sum_{j=1}^{n} Fit(T_j)} \tag{2}$$

(3) Crossover Operators: In addition to determining how pairs of parents are selected for crossover, we need to determine how the characteristics of the parents are used to derive offspring through the crossover process. Typically, the crossover operator merges the components of two parents in some way to construct a new individual (candidate solution). For example, the individual parent representations may be split into parts, and then selected parts from the two parents may be mixed to derive a new individual. For the tree representation in EADTC, the crossover operator exchanges a subtree of one parent with the corresponding subtree in the other parent. Fig. 1 illustrates this type of crossover operation.

(4) Mutation Operators: Mutation operators simulate the biological mechanism of mutation, where certain parts of a genome may undergo random changes in nature. Mutation operators for tree-based models in EAs include: (a) replacing the decision feature or threshold within a node — $M(n(d, f, t, l, r)) = n(d, \hat{f}, \hat{t}, l, r)$, where M is the mutation operator and $\hat{f}, \hat{t}$ give the new feature and threshold values; and (b) replacing a subtree by a leaf $n(d, class, None, None, None)$. As shown in Fig. 1, the right subtree of the individual can mutate into a leaf node.

(5) Termination Condition: The termination condition specifies when an evolutionary algorithm should end. Commonly, the stopping condition is specified in terms of a fixed number of generations through which the EA should execute. In EADTC, we incorporate a tolerance term, which stops evaluating new generations when a certain number of consecutive generations does not produce any improvement. This allows the EA to continue until it converges, at least in a local sense of convergence.

C. Fairness and Equity

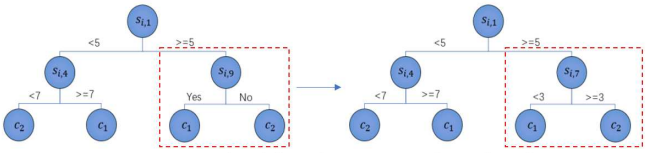
In fairness-integrated approaches to machine learning, some features may be designated as being associated with *protected groups*, which means that the features should not be directly used as input to the models. However, even when direct indicators of protected group membership, such as race and gender, are not included as input, some correlations between these measures and legitimate predictors can result in unfairness. This enlightens us that only discarding sensitive information is not enough to eliminate bias; we also need to define fairness and design corresponding metrics. As seen in the recent literature on fairness and recidivism prediction, it is not possible to maximize accuracy and all types of fairness at the same time [27] (see also [28] [29]).

Two useful metrics for assessing algorithmic fairness in recidivism prediction are conditional procedure error CPE and conditional use error CUE associated with a given protected group (e.g., a race category).

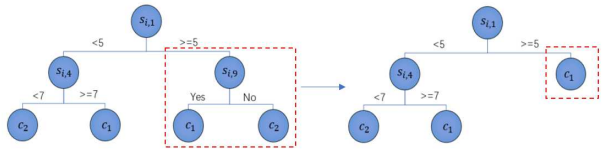
Suppose that we have a predictive model Z that is tested on a dataset D that includes members from multiple protected groups, and the underlying prediction problem involves discrimination between two classes, which we refer to as positive and negative classes. For example, in our experiments in Section IV, we consider the positive and negative classes to correspond to high-risk and low-risk members, respectively, of the population being studied.

The CPE and CUE each have two components. For CPE and a given protected group g, these components are defined as follows: $\alpha_1(g) = FN(g)/(TP(g) + FN(g))$, $\alpha_2(g) = FP(g)/(FP(g) + TN(g))$, where $TP(g)$, $FP(g)$, $TN(g)$, and $FN(g)$ respectively represent the counts of true positives, false positives, true negatives, and false negatives in the given testing experiment for Z across all instances in the given dataset δ that are associated with protected group g. Similarly, for CUE, there are two components, which we can express as $\beta_1(g) = FP(g)/(TP(g) + FP(g))$, and $\beta_2(g) = FN(g)/(FN(g) + TN(g))$.

Given two protected groups g_1 and g_2, we can define disparity metrics associated with their CPE and CUE values



(a) Crossover example, right subtree comes from the other parent.



(b) Mutation example, right subtree mutates to final class c_1.

Fig. 1: Illustrations of crossover and mutation operators in EADTC. Here, c_1 and c_2 represent the two classes for classification, and $s_{i,j}$ represents the jth feature of the ith sample.

as follows:

$$\Delta_\alpha(g_1,g_2) = \frac{abs(\alpha_1(g_1)-\alpha_1(g_2)) + abs(\alpha_2(g_1)-\alpha_2(g_2))}{2},$$

and similarly,

$$\Delta_\beta(g_1,g_2) = \frac{abs(\beta_1(g_1)-\beta_1(g_2)) + abs(\beta_2(g_1)-\beta_2(g_2))}{2},$$

where $abs(x)$ represents the absolute value of the given number x. Observe from this formulation that Δ_α and Δ_β always lie in the interval $[0,1]$.

The concepts of CPE and CUE are used in two forms of fairness assessment, which are referred to as conditional procedure accuracy equality (CPAE) and conditional use accuracy equality, respectively (CUAE). CPAE is said to be achieved across a given set $G = \{g_1, g_2, \ldots, g_n\}$ of protected groups when $\Delta_\alpha(g_i, g_j) = 0$ for all $i, j \in \{1,2,\ldots,n\}$. Similarly, CUAE is said to be achieved across G when $\Delta_\beta(g_i, g_j) = 0$ for all $i, j \in \{1,2,\ldots,n\}$.

IV. Experiments

We developed a prototype implementation of the EADTC method using Python 3.7 along with sci-kit of version 0.23.2. In this section, we report on experimental results that we derived from this implementation. The evolutionary algorithm in EADTC follows a typical procedure imitating the natural selection process: representing individuals, parent selection, crossover and mutation (e.g. see [15]). For EADTC, most of the steps are similar to other EA applications, but due to the special tree representation of each individual, we made adjustments to traditional bit-wise crossover and mutation and directly exchanged nodes between parents as shown in Fig. 1.

Key hyperparameters involved in the evolutionary algorithm are summarized in Section IV-B.

A. Data Preparation

We used two data sets for the experiments. The first one, called *Recidivism of Prisoners Released in 1994*, is provided by ICPSR (Inter-University Consortium for Political and Social Research), and includes 38,624 release records from 1994. The records were randomly sampled from 302,309 prisoners in 15 states (Arizona, California, Delaware, Florida, Illinois, Maryland, Michigan, Minnesota, New Jersey, New York, North Carolina, Ohio, Oregon, Texas, and Virginia) [30]. The released individuals were tracked for three years following their release. The features include demographic information, such as date of birth, sex, and race. The features also include arrest information, such as the date of arrest, prior arrests, and an out-of-state indicator; as well as sentencing and prison term information, such as the length of sentence, crime type, and time served.

We first determined a set of features that we expected to be especially useful for purposes of prediction. The selected features are summarized in Table I. The age variables are calculated by simply subtracting the birth date from the arrest/release date. Non-categorical features were scaled to a form with zero mean and unit variance, which helps to avoid overfitting on features that have larger magnitudes.

TABLE I: Variables from the ICPSR dataset that were used in our experiments.

Variable	Dtype	Description
SEX1	int64	Sex of prisoner
LENG_SEN	int64	Sentence length
DRUGAB	bool	Drug abuser
ALCABUS	bool	Alcohol abuser
DRUGTRT	bool	Drug treatment
ALCTRT	bool	Alcohol treatment
DFM	bool	Felony or misdemeanor
DDMV	bool	Domestic violence involved
DFIR	bool	Firearms involved
PRIR	int64	Number of prior arrests
RELTYP	category	Release type
NFRCTNS	int64	Number of infractions
TMSRV	int64	Length of prison term served
AGE_1stARR	float64	Age at first arrest
AGE_1stRLS	float64	Age at release

The second recidivism dataset that we used represents individuals in Pennsylvania state prisons. The individuals represented in this dataset were released into parole supervision between 2006 and 2008. The set of features that we used for this second dataset, which we refer to as the PA dataset, is summarized as shown in Table II.

For both datasets, the label we targeted on is whether the released prisoner experienced a rearrest within 3 years of release.

B. Evolutionary Algorithm Hyperparameters

In our experiments, the settings for key hyperparameters in the evolutionary algorithm engine underlying EADTC are selected from a dense grid search, and the finalized set is as follows.

- `n_trees = 100` (number of decision trees in each generation).

TABLE II: Variables from the PA dataset that were used in our experiments.

Variable	Dtype	Description
SEX	int64	Sex of prisoner
AGE	int64	Age at release
VIO	bool	Violent crime
PRO	bool	Property crime
DRUG	bool	Drug crime
PUBLIC	bool	Public order crime
OTHER	bool	Other crime
CRT_ADM	bool	Court admission
TM_SERVE	bool	Length of prison term served
FIRST	bool	First-time incarceration
LSIR	int64	Risk Score

- `max_depth = 6` (maximum depth of each decision tree).
- `max_iter = 4000` (maximum number of iterations to run the evolutionary algorithm).
- `early_stopping = 200` (number of iterations without improvement that triggers early termination of the evolutionary algorithm).
- `selection = proportional_fitness` (use fitness-proportional selection to select parents for recombination).
- `mutation_prob = 0.1` (probability of mutating an individual).
- `selected_parent = 10` (the number of parents to be kept for the next generation).

These parameters were determined empirically through iterative experimentation.

C. Model Performance

All experiments were carried out with 4-fold cross validation. We first compared the prediction accuracy of EADTC with several relevant models from the literature: logistic regression (LR), which uses a linear combination of variables to estimate a probability for each class [31]; ridge regression, which is a variation of linear regression [32]; the classification and regression tree (CART) method, which is a modern variation on the single decision tree approach, as described in [17]; the random forest (RF), which is an ensemble of decision trees, and provides the foundation for the approach of Berk and Bleich (see Section II-B); gradient boosting decision trees (GBDTs), which use gradient descent techniques for loss minimization that are generalized to trees [13].

The results of our comparison between EADTC and the previously-developed models are summarized in Table III. The accuracy values are in the range of $[0,1]$, where an accuracy of 1 represents 100% accuracy.

TABLE III: Comparison between EADTC and several reference models in terms of prediction accuracy

	LR	Ridge	CART	RF	GBDT	EADTC
ICPSR	0.693	0.675	0.688	0.701	0.708	0.697
PA	0.772	0.765	0.769	0.786	0.788	0.776

TABLE IV: AUC comparison of all models

	LR	Ridge	CART	RF	GBDT	EADTC
ICPSR	0.757	0.755	0.757	0.762	0.769	0.760
PA	0.701	0.695	0.696	0.715	0.718	0.707

Given that recidivism data can be highly unbalanced, the Area Under the Receiver Operator Characteristic Curve (AUC) can provide better insight, compared to prediction accuracy, into the relative performance of different prediction methods. Table IV compares the measured performance of EADTC with the reference models in terms of the AUC metric.

From the results of Table III and Table IV, we can see that EADTC shows improvement both in terms of accuracy and AUC over all of other interpretable models, including CART, logistic regression and ridge regression. This indicates that EADTC is competitive even when handling unbalanced data sets. EADTC also helps to the narrow gap of performance between interpretable models and ensemble models such as RF and GBDT, which lack interpretability.

In addition to the reference models that we experimented with for the comparisons reported in Table III and Table IV, we also compared EADTC with the RiskSLIM approach [10] discussed in Section II-B. However, we did not include RiskSLIM in our comparison tables. We omitted RiskSLIM from the tables because it strictly requires all features to be Boolean-valued, which can make it more difficult to apply compared to the methods that are included in the tables. In particular, the restriction to Boolean data types requires manual bagging to handle non-categorical features. The need for such bagging can introduce two problems: (1) models can be difficult to adapt to new datasets, and (2) information can be lost in the bagging process that is useful for our prediction purposes. Although we did not include RiskSLIM in Table III and Table IV, we did experiment with the approach using the ICPSR dataset. We found that the best performance levels we measured on RiskSLIM, after hand-tuning its hyperparameters, are an accuracy of 0.674 and an AUC of 0.731, which are both lower than the corresponding results produced by EADTC. Due to the limitations described above associated with the bagging requirement for RiskSLIM, we did not carry out experiments on the PA dataset using RiskSLIM. Regardless of the performance of RiskSLIM, its use brings about limitations in connection with the objectives of this paper — in particular, the objective of being efficiently adaptable to different datasets — and warrants further comparative research.

D. Fairness Metrics

We evaluated the prediction model derived by EADTC in term of fairness metrics defined in Section III-C. We selected the *Race* attribute as our protected attribute, and examined the protected groups corresponding to white and black members of the released prison population, respectively. We employed the ICPSR dataset in this experiment.

TABLE V: Confusion matrix for white prisoners.

White	High risk prediction	Low risk prediction	CPE
Rearrest	737	265	0.267
Non-rearrest	371	668	0.361
CUE	0.338	0.288	

TABLE VI: Confusion matrix for black prisoners.

Black	High risk prediction	Low risk prediction	CPE
Rearrest	1263	361	0.171
Non-rearrest	334	420	0.491
CUE	0.225	0.407	

Table V and Table VI show the confusion matrices resulting from application of our predictive model on each of the two protected groups. The tables also provide CPE and CUE values. From these tables, we see that our model exhibits similar false rates on the protected groups, where the false rate is simply the sum of the false positive and false negative rates.

An illustration of the prediction results on protected groups in the form of pie charts is shown in Fig. 2.

From the results in Table V and Table VI, we can calculate the disparity metrics we defined in Section III-C by

$$\begin{aligned}\Delta_\alpha(g_w,g_b) &= \frac{abs(\alpha_1(g_w)-\alpha_1(g_b))+abs(\alpha_2(g_w)-\alpha_2(g_b))}{2}\\ &= \frac{abs(0.267-0.171)+abs(0.361-0.491)}{2}\\ &= 0.113\end{aligned}$$

and

$$\begin{aligned}\Delta_\beta(g_w,g_b) &= \frac{abs(\beta_1(g_w)-\beta_1(g_b))+abs(\beta_2(g_w)-\beta_2(g_b))}{2}\\ &= \frac{abs(0.338-0.225)+abs(0.288-0.407)}{2}\\ &= 0.116.\end{aligned}$$

Here, g_b and g_w denote the black and white protected groups, respectively.

V. Conclusions

In summary, our work makes the following contributions:

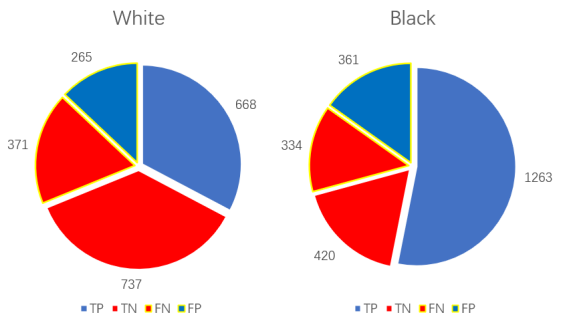


Fig. 2: Illustration of prediction results on protected groups.

- Instead of introducing more complexity by using a traditional greedy induction strategy, we propose a new evolutionary algorithm algorithm approach, called EADTC (Evolutionary Algorithm Decision Tree for Crime prediction). EADTC incorporates optimization to increase the effectiveness of the decision tree model without losing interpretability.
- Unlike most of the existing risk assessment models that adopt scoring systems, models produced by EADTC do not depend heavily on domain specific feature engineering, such as age bagging. Additionally, hyper-parameter tuning and data processing are relatively straightforward for tabular data to be compatible with the model.
- We report on comprehensive experiments to analyze models produced by EADTC in terms of relevant metrics of predictive performance and fairness, and to provide comparison against several alternative prediction approaches. The results demonstrate the effectiveness of the EADTC approach in terms of both predictive performance and fairness.

Overall, the EADTC method that we designed in this paper outperforms single CART while maintaining the same high level of interpretability. While this paper makes significant contributions in the area of interpretability and transparency in machine learning, future research is warranted to formalize model interpretability and systematically improve key aspects of interpretability that help foster legitimacy in machine learning-supported decision making.

References

[1] Z. Xu, J. Zhang, Q. Zhang, and P. S. F. Yip, "Explainable learning for disease risk prediction based on comorbidity networks," in *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019, pp. 814–818.

[2] A. L. Beam and I. S. Kohane, "Big data and machine learning in health care," *Jama*, vol. 319, no. 13, pp. 1317–1318, 2018.

[3] R. Berk, D. Berk, and Drougas, *Machine learning risk assessments in criminal justice settings*. Springer, 2019.

[4] J. Kleinberg, H. Lakkaraju, J. Leskovec, J. Ludwig, and S. Mullainathan, "Human decisions and machine predictions," *The quarterly journal of economics*, vol. 133, no. 1, pp. 237–293, 2018.

[5] C. Metz and A. Satariano, "An algorithm that grants freedom, or takes it away," *The New York Times*, vol. 6, 2020.

[6] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[7] L. J. Ba and R. Caruana, "Do deep nets really need to be deep?" *arXiv preprint arXiv:1312.6184*, 2013.

[8] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[9] A. S. Ravindran, M. Cestari, C. Malaya, I. John, G. E. Francisco, C. Layne, and J. L. C. Vidal, "Interpretable deep learning models for single trial prediction of balance loss," in *2020 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2020, pp. 268–273.

[10] B. Ustun, S. Traca, and C. Rudin, "Supersparse linear integer models for interpretable classification," *arXiv preprint arXiv:1306.6677*, 2013.

[11] J. Zeng, B. Ustun, and C. Rudin, "Interpretable classification models for recidivism prediction," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 180, no. 3, pp. 689–722, 2017.

[12] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[13] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

[14] R. Berk and J. Bleich, "Forecasts of violence to inform sentencing decisions," *Journal of Quantitative Criminology*, vol. 30, no. 1, pp. 79–96, 2014.
[15] T. Back, U. Hammel, and H. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 3–17, 1997.
[16] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao *et al.*, "Interpretability of deep learning models: A survey of results," in *2017 IEEE smartworld, ubiquitous intelligence & computing, advanced & trusted computed, scalable computing & communications, cloud & big data computing, Internet of people and smart city innovation (smartworld/SCALCOM/UIC/ATC/CBDcom/IOP/SCI)*. IEEE, 2017, pp. 1–6.
[17] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. Routledge, 2017.
[18] R. A. Berk, S. B. Sorenson, and G. Barnes, "Forecasting domestic violence: A machine learning approach to help inform arraignment decisions," *Journal of empirical legal studies*, vol. 13, no. 1, pp. 94–115, 2016.
[19] A. Traoré and M. A. Akhloufi, "Violence detection in videos using deep recurrent and convolutional neural networks," in *2020 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2020, pp. 154–159.
[20] X. Wu and T. H. Grubesic, "Identifying irregularly shaped crime hot-spots using a multiobjective evolutionary algorithm," *Journal of geographical systems*, vol. 12, no. 4, pp. 409–433, 2010.
[21] P. J. Angeline, G. M. Saunders, and J. B. Pollack, "An evolutionary algorithm that constructs recurrent neural networks," *IEEE transactions on Neural Networks*, vol. 5, no. 1, pp. 54–65, 1994.
[22] T.-C. Lu, "Cnn convolutional layer optimisation based on quantum evolutionary algorithm," *Connection Science*, pp. 1–13, 2020.
[23] Z. R. K. Rostam and S. A. Mahmood, "Classification of brainwave signals based on hybrid deep learning and an evolutionary algorithm," *arXiv preprint arXiv:1912.07361*, 2019.
[24] B. S. Neysiani, N. Soltani, R. Mofidi, and M. H. Nadimi-Shahraki, "Improve performance of association rule-based collaborative filtering recommendation systems using genetic algorithm," *International Journal of Information Technology and Computer Science*, vol. 11, no. 2, pp. 48–55, 2019.
[25] Q. Zhao and M. Shirasaka, "A study on evolutionary design of binary decision trees," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3, 1999, pp. 1988–1993.
[26] D. Jankowski and K. Jackowski, "Evolutionary algorithm for decision tree induction," in *IFIP International Conference on Computer Information Systems and Industrial Management*, 2015, pp. 23–32.
[27] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq, "Algorithmic decision making and the cost of fairness," in *Proceedings of the 23rd acm sigkdd international conference on know·ledge discovery and data mining*, 2017, pp. 797–806.
[28] A. Chouldechova, "Fair prediction with disparate impact: A study of bias in recidivism prediction instruments," *Big data*, vol. 5, no. 2, pp. 153–163, 2017.
[29] J. Kleinberg, S. Mullainathan, and M. Raghavan, "Inherent trade-offs in the fair determination of risk scores," *arXiv preprint arXiv:1609.05807*, 2016.
[30] P. A. Langan and D. J. Levin, *Recidivism of prisoners released in 1994*. US Department of Justice, Office of Justice Programs, Bureau of Justice …, 2002.
[31] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013.
[32] S. Le Cessie and J. C. Van Houwelingen, "Ridge estimators in logistic regression," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 41, no. 1, pp. 191–201, 1992.