AI For Bias Detection: Investigating the Existence of Racial Bias in Police Killings

Anaiy Somalwar, Chinmay Bansal, Nathan Lintu, Rishab Shah, and Phil Mui
Department of Computer Science & Engineering
Aspiring Scholars Directed Research Program
Fremont, CA, USA
e-mails: {anaiysomalwar, chinmaybansalsc, natelin, shahrishabn, thephilmui}@gmail.com

***Abstract*—A recent paper in the Journal of Politics applied machine learning models to analyze racial disparities in police shooting fatalities. It suggested building generalizable binary classifiers simply based on those social and demographic factors. This recent research in 2019 was not able to establish a convincing machine learning model account for the observed racial bias in fatalities resulting from police shootings. This paper proposes key improvements made from the previous machine learning research through the use of class weights, random oversampling, and a more expansive dataset. We are able to show generalizable relationships between factors related to a police killing and the race of the fatality with over 80 percent accuracy based on the US national police shooting fatality data in recent years. This research suggests areas for further investigations of racial bias in police killings with supervised machine learning as a methodology for algorithmic and data bias detection.**

***Index Terms*—Artificial Intelligence, Bias Detection, Machine Learning, Police Killings, Racial Bias**

I. Introduction

While supervised machine learning is commonly used with the simple goal of maximizing the accuracy of the prediction of output labels given certain input features, it can also be used to explore whether connections can be created between the features and labels, which has tremendous applications in detecting bias in situations where connections between a set of inputs and outputs should not exist. While this application of supervised machine learning could be used in a variety of different fields, we focus on the topical problem of exploring whether racial bias in police killings exists in the United States, a problem which has great implications in both racial justice and society in general.

The investigation of the existence of racial police bias has been traditionally conducted through the use of statistics, where the proportion of a certain descriptor of police killings is compared between races [1]–[5]. However, in a recent paper published in the Journal of Politics in 2019 [6], supervised machine learning models were suggested for the binary classification of White American and African American police killing fatalities to explore what factors were relevant in making such a classification and what those factors may suggest of racial disparities and police bias. This was simply an improvement to traditional statistical methods of investigating the existence of racial bias in police killings as the relationships between multiple variables and the race of a fatality can be determined from machine learning models.

However, the accuracy of the machine learning models applied in the previous research were comparable that of a "no-information" model that always outputs the majority class, which means that the models were unable to "learn" some set of general, accurate parameters or rules that would be of particular use in the classification task. So, the study concluded that there was no evidence of racial bias in police killings due to the inability of complex machine learning models to find connections between the input descriptors of police killings and the fatality's race that would allow it to make accurate classifications, a theoretically impossible task. Also, even if a machine learning model could be significantly more accurate than a no-information model, it would be important for there to be relatively similar accuracies across both classes, which could be measured by an F1 score, to demonstrate that the model learned anything particularly applicable, which is another challenge.

However, the databases available for the tracking of police killings have improved, with more data categories or descriptions of each police killing and unfortunately, a greater number of police killings or data points, which is crucial for accurate machine learning and deep learning model training. In this research, we utilize a more expansive dataset and improved machine learning training techniques to correct for dataset imbalances to build upon the previous research to explore whether machine learning models could become significantly more accurate and to analyze the rules or parameters of such models to investigate their implications on racial disparities and police bias.

II. Methods

In this research, we train a variety of machine learning models on an expansive, frequently-updated database to simulate how the binary classification of an African American or non-African American fatality of a police killing would be conducted to build upon previous research by determining whether such a classification could be done with reasonable

accuracy and to analyze the rules or weights of such a model to determine their implications on racial bias in police killings. However, it is important to note that our classification task is slightly different than that of the previous work as our majority class of non-African American fatalities is slightly larger as it includes non-white, non-African Americans, which creates a more difficult task in maintaining a similar precision and recall.

A. Dataset

We used police killing data from the "2013-2020 Police Killings" sheet from the Mapping Police Violence database [7] to train various machine learning models. This dataset contained 8210 "data points", or police killings, when we conducted our research, which is nearly six times the amount of data used in the previous study. The "2013-2020 Police Killings" sheet contained several categories describing each police killing in the United States since 2013 and is updated regularly. The categories it contained include "Victim's Name", "Street Address of Incident", "City", "State", "Zipcode", "County", "Agency Responsible for Death", "Cause of Death", "Brief Description", "Official Disposition", "Criminal Charges", "Symptoms of Mental Illness", "Unarmed/Did Not Have A Weapon", "Alleged Weapon", "Alleged Threat Level", "Fleeing", "Body Camera", "WaPo ID", "Off Duty Killing?", and "Geography", and we manually validated 300 fatalities by comparing them to data in other public databases such as that of the Washington Post [8] to ensure its accuracy.

B. Model Selection

We chose to apply multiple machine learning models discussed in the previous research to establish a baseline for accuracy comparisons. However, we also chose to add additional benchmark and state-of-the-art models. We implemented a support vector machine [9], a multilayer perceptron neural network [10], a white box decision tree model [11], and a random forest [12] from the previous work and added a logistic regression [13] and gradient boosting tree [14] for the classification task.

1) Logistic Regression: A logistic regression is a supervised machine learning model that improves the accuracy of its classifications by minimizing a loss function through gradient descent [15]. The loss function of a logistic regression is $-log(1 - h_0(x) - y)$, where y is the binary label of the data point and $h_0(x)$, the hypothesis function, is the output of the sigmoid function at the sum from i to n of $\beta_0 + \beta_i X_i$, where β_0 is a bias, β_i is a weight, X_i is the feature at i, and n is the length of features given. Since the logistic regression will be given an input array that consists of a large number of complex features, we can expect the model to be blackbox, or difficult to interpret.

2) Support Vector Machine: Support vector machines (SVM) are generally blackbox, supervised machine learning models that maximize the difference between classes across a multidimensional space through the use of a linear hyperplane. When a support vector machine is given data that is not initially linearly separable, which occurs frequently in complex problems, the data is transformed into a space with higher dimensions through a kernel function [16] until a linear hyperplane can be constructed, which makes the resulting model difficult to interpret.

3) Decision Tree: A decision tree is a supervised machine learning model based on making a limited amount of decisions that allow for the classification of input features. Decision trees are a non-parametric, top-down model that make decisions, or splits in the tree, by maximizing information gain by decreasing entropy, the negative sum of $p(x) \log_2 p(x)$, where $p(x)$ is the fraction of examples in a given class. Since decision tree models fundamentally make classifications based on multiple, binary decisions, they are generally extremely interpretable and have the potential to explain patterns in the data. However, although decision trees are interpretable machine learning models, as they become more complex, and often more accurate, by making more splits in the tree, they increasingly lose their explainability.

4) Random Forest: A random forest is an ensemble machine learning model that makes classifications by constructing multiple decision tree models and outputting the mode of the outputs of those decision trees. Random forests generally perform better than decision trees as they correct for a decision tree's tendency to overfit on the training set. Although random forests, as a whole, are considered to be a blackbox, or uninterpretable model, it is possible to interpret each individual decision tree in a random forest. However, when attempting to interpret a random forest, it should be noted that each individual decision tree within it is generally considered to be a weak classifier as compared to a pure decision tree model.

5) Gradient Boosting Tree: A Gradient Boosting Tree (GBT) is a supervised machine learning model that relies on the creation of several weak classifiers, or decision trees. While the gradient boosting tree and random forest are similar in that they both are ensemble models based on multiple decision trees, their key difference lies in the way the ensemble is produced. While random forests generate decision trees through bagging, gradient boosting trees create decision trees recursively by improving the current ensemble, which is also known as boosting. However, like random forests, gradient boosting trees are considered to be a blackbox model overall, but it is possible to interpret each individual decision tree in the ensemble.

6) Multilayer Perceptron: A multilayer perceptron (MLP), or a feed-forward artificial neural network, is a type of deep learning model that was inspired by the way neurons work in human brains. Multilayer perceptrons consist of an input layer, one ore multiple hidden layers, and an output layer, which

are connected through optimized functions. Each non-input layer is connected to a non-linear activation function, and all layers are trained through backpropogation [17], a supervised machine learning technique. Based on the massive number of parameters, most of which are weights in the functions that connect each layer, multilayer perceptrons and neural networks are known to perform at state-of-the-art accuracy, but are generally uninterpretable.

C. Feature Selection

We used 9 relevant categories with sufficient recorded instances and without missing values: the fatality's gender, the state in which the killing occurred, the cause of death, the resulting criminal charges, whether the fatality had a mental illness, whether the fatality was armed, the fatality's alleged weapon, whether the fatality was fleeing, and whether the police officer's body camera was on to classify the fatality's race. To create inputs for the model, we dropped null values in the dataset, removed duplicate values, and one-hot encode and horizontally stacked the 9 categories to create a list of 182-vectors. After dropping null and duplicate values, there remained approximately 6800 police killings to serve as input to our models. We one-hot encoded the labels so that [1,0] denoted an African American killing and [0,1] denoted a non-African American killing and created a 60-20-20 training, validation, and test set respectively.

D. Class Imbalance Corrections

Since there is an under-representation of the African American class as around 70 percent of the inputs were non-African Americans, even if a machine learning model performed with an accuracy of around 70 percent on the testing data, it would be unclear whether the model "learned" any useful information that suggests the existence of police bias. Thus, a higher level of model accuracy needed to be achieved, so we attempted several different methods to improve the accuracy of the models. In the previous work, this class imbalance was not accounted for during training, so we used class weights to increase the penalization of an incorrect prediction on the minority class and also implemented simple random oversampling before training. Both techniques were used separately.

E. Model Creation

We used a multilayer perceptron with one hidden layer, where the input layer had 2000 units, the hidden layer had 200 units, and the output layer had one unit. We used the "relu" [18] activation function after each dense layer, a dropout [19] of 0.3 after the input and hidden layers, and trained it with a batch size of 16, the default Adam optimizer [20], and 20 epochs using binary crossentropy loss. We trained the logistic regression using a learning rate of 0.01, the decision tree using the gini coefficient and a maximum depth of 5000, the support vector machine using stochastic gradient descent, and the random forest and the gradient boosting tree with 300 estimators. All of the machine learning models used were constructed with the scikit-learn [21] library, and any model parameters not specified were set to default values. All hyperparameters were determined through grid search [22] and all of the code for the optimized models has been posted on GitHub.

III. Results

We evaluated the performance of the models' using two metrics: simple testing accuracy and the F1-Score. The F1-score represents the average accuracies of the predictions across the testing data, and its formula can be seen below, where TP represents the true positives, FP represents the false positives, and FN represents the false negatives. We monitor the F1-score to ensure that the models do not simply always output the majority class, which may have been a problem in the previous research and would signify that the models were not truly "learning", and thus do not actually show correlations between the social and demographic input features and the race of a police killing fatality.

$$F_1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{1}$$

The testing accuracies of the logistic regression, support vector machine, multilayer perceptron, decision tree, gradient boosting tree, and random forest using class weights during training are 61.9%, 65.1%, 67.2%, 62.7%, 66.4%,and 68.3% respectively, and their F1-scores are 55.0%, 62.2%, 67.1%, 62.0%, 64.3%, and 64.7% respectively. The testing accuracies of the logistic regression, support vector machine, multilayer perceptron, decision tree, gradient boosting tree, and random forest using oversampling are 70.8%, 73.2%, 76.2%, 78.5%, 79.9%, and 81.3% respectively, and their F1-scores are 70.8%, 73.2%, 76.2%, 78.3%, 78.4%, and 81.2% respectively. This is summarized in Table I below, where LR, SVM, MLP, DT, GBT, and RF represent logistic regression, support vector machine, multilayer perceptron, decision tree, gradient boosting tree, and random forest, respectively, and AccOversampling, AccCW, F1Oversampling, F1CW, represent the testing accuracies of the models using oversampling, the testing accuracies of the models using class weights during training, the F1-scores of the models using oversampling, and the F1-scores of the models using class weights during training respectively.

TABLE I
ACCURACIES AND F1-SCORES OF MODELS

	AccOversampling	AccCW	F1Oversampling	F1CW
LR	70.8%	61.9%	70.8%	55.0%
SVM	73.2%	65.1%	73.2%	62.2%
MLP	76.2%	67.2%	76.2%	67.1%
DT	78.5%	62.7%	78.3%	62.0%
GBT	79.9%	66.4%	78.4%	64.3%
RF	81.3%	68.3%	81.2%	64.7%

IV. Discussion

While the accuracies of the models using class weights while training did not improve upon the accuracy of the previous work, it should be noted that the F1-scores of the

models are relatively close to the true accuracies of the models, suggesting that the use of class weights prevented the model from outputting only one class. However, the F1-scores from the previous work were not reported, which makes the effects of the improvement of using class weights during training difficult to determine. However, the use of random oversampling made clear improvements, with the random forest performing with over 11 percent higher accuracy than the best model from the previous work and the best model created with class weights during training in this work. So, it is clear from the 81.3% accuracy of the random forest that it did indeed learn some set of accurate and generalizable rules about the data that theoretically should perhaps not exist, which could indicate the existence of racial police bias. To ensure that the model's high accuracy was not the result of an irregular split of the dataset that resulted in a simpler test set, we ran the model 100 times and graphed the distribution of testing accuracies, a distribution with a range of less than 0.02%. This demonstrates that the model is consistently accurate and generalizable, and the exact distribution can be seen below.

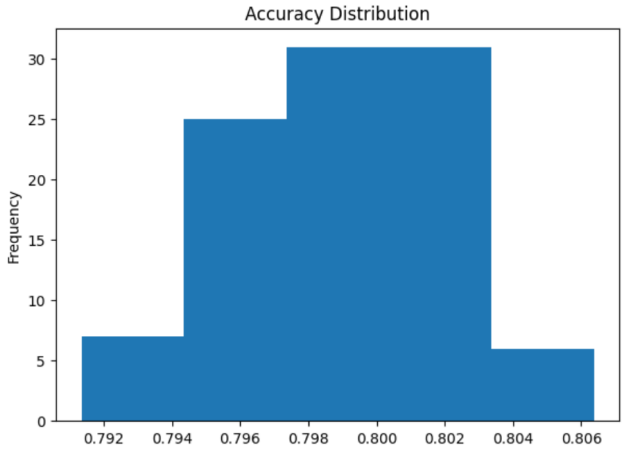

Fig. 1. A visualization of the distribution of accuracies of the random forest. This graph demonstrates the consistency of the random forest's accuracy.

Although there is evidence that there exists some relationships between certain descriptors of a police killing that make it possible to predict the fatality's race accurately, to understand what and whether police bias exists, the relationships between those descriptors must be interpreted to ensure that those relationships do not have any other explanation. However, this is not a trivial task; the random forest model that performed with over 80% testing accuracy was over 500 splits deep, which makes it extremely difficult to visualize and interpret. Although we attempted to place a maximum depth of splits on the random forest while training to make it more understandable, over 30 splits were necessary to consistently perform with a validation accuracy of over 70 percent, which is still quite complex, as depicted in Figure 2.

While we are still working towards determining what specific descriptors of a police killing make it possible for the

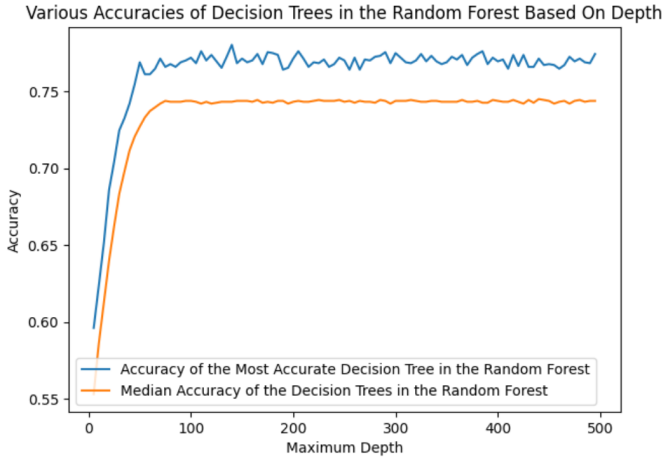

Fig. 2. A visualization of the highest and median accuracies of decision trees in the random forest based on their maximum depth. This graph demonstrates the large number of splits necessary to attain a high classification accuracy.

accuracy of such a model to be so consistently high, the attempted methodologies of having a larger, more expansive dataset as well as simple random oversampling and class weights during model training greatly improved the accuracy and generalizability of the models from previous research and may suggest that some form of racial bias in police killings may exist. However, it is extremely important to note that while there exist correlations between the social and demographic factors and the race of a fatality in a police killing, which is suggested by an AI model's ability to predict the one given the other with high accuracy, this is not definite evidence that there is racial bias in police killings as these correlations may be at least partly explained based on population demographics, natural variance, and the different nature of police encounters between different race populations. However, the increased accuracy of the models for this theoretically impossible classification task is an important step forward for understanding what set of complex, social and demographic factors may be correlated with a police killing of a certain race and whether racial bias exists in police killings.

V. Conclusion

Overall, this research shows that there is a strong correlation between social and demographic factors and the race of a fatality in a police killing. These relationships between social and demographic factors and the race of the fatality were found with over 80 percent classification precision and recall by interpretable machine learning models - a level which has not been attained previously. This research not only serves as an important improvement and stepping stone for further investigations in determining whether racial bias exists in police killings, but also as an example of supervised machine learning as a methodology for bias detection, which could be applied in a variety of different fields.

REFERENCES

[1] A. Gelman, J. Fagan, and A. Kiss, "An analysis of the new york city police department's "stop-and-frisk" policy in the context of claims of racial bias," *Journal of the American statistical association*, vol. 102, no. 479, pp. 813–823, 2007.

[2] C. T. Ross, "A multi-level bayesian analysis of racial bias in police shootings at the county-level in the united states, 2011–2014," *PloS one*, vol. 10, no. 11, p. e0141854, 2015.

[3] I. Cano, "Racial bias in police use of lethal force in brazil," *Police Practice and Research: An International Journal*, vol. 11, no. 1, pp. 31–43, 2010.

[4] J. A. Shjarback and J. Nix, "Considering violence against police by citizen race/ethnicity to contextualize representation in officer-involved shootings," *Journal of criminal justice*, vol. 66, p. 101653, 2020.

[5] J. L. Worrall, S. A. Bishopp, and W. Terrill, "The effect of suspect race on police officers' decisions to draw their weapons," *Justice Quarterly*, pp. 1–20, 2020.

[6] S. Streeter, "Lethal force in black and white: Assessing racial disparities in the circumstances of police killings," *The Journal of Politics*, vol. 81, no. 3, pp. 1124–1132, 2019.

[7] M. P. Violence, "Mapping police violence," *Mapping Police Violence*, 2017.

[8] W. Post, "Police shootings database 2015-2020," *Washington Post*, 2016.

[9] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[10] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[11] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[12] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1, pp. 278–282, IEEE, 1995.

[13] J. A. Nelder and R. W. Wedderburn, "Generalized linear models," *Journal of the Royal Statistical Society: Series A (General)*, vol. 135, no. 3, pp. 370–384, 1972.

[14] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[15] J. Kiefer, J. Wolfowitz, *et al.*, "Stochastic estimation of the maximum of a regression function," *The Annals of Mathematical Statistics*, vol. 23, no. 3, pp. 462–466, 1952.

[16] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[17] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, pp. 65–93, Elsevier, 1992.

[18] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[19] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[22] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.