Fairness Measures of Machine Learning Models in Judicial Penalty Prediction

Yanjun Li[1,4], Huan Huang[2], Qiang Geng[2,5*], Xinwei Guo[3], Yuyu Yuan[1]

[1] Beijing University of Posts and Telecommunications, China
[2] Shenzhen Research Institute of Nanjing University, China
[3] China Justice Big Data Institute Co.Ltd, China
[4] China National Accreditation Service for Conformity Assessment, China
[5] Population Research Institute, Business School of Nanjing University, China
liyj@cnas.org.cn, huanghuan@mooctest.com, gengq@nju.edu.cn, guoxinwei@cjbdi.com, yuanyuyu@bupt.edu.cn

Abstract

Machine learning (ML) has been widely adopted in many software applications across domains. However, accompanying the outstanding performance, the behaviors of the ML models, which are essentially a kind of black-box software, could be unfair and hard to understand in many cases. In our human-centered society, an unfair decision could potentially damage human value, even causing severe social consequences, especially in decision-critical scenarios such as legal judgment. Although some existing works investigated the ML models in terms of robustness, accuracy, security, privacy, quality, etc., the study on the fairness of ML is still in the early stage. In this paper, we first proposed a set of fairness metrics for ML models from different perspectives. Based on this, we performed a comparative study on the fairness of existing widely used classic ML and deep learning models in the domain of real-world judicial judgments. The experiment results reveal that the current state-of-the-art ML models could still raise concerns for unfair decision-making. The ML models with high accuracy and fairness are urgently demanding.

Keywords: Fairness, Machine learning, Judgment document, Big data

1 Introduction

With the boom of big data in the past decade, machine learning (ML) has been successfully applied to many application domains. ML provides an approach to solving problems from the real world and has achieved remarkable results, such as face recognition [1], machine translation [2], anomaly detection [3], and automatic driving [4-5]. Meanwhile, ML has especially been applied to the applications in judicial fields [6-8], mainly aimed at processing the decision-making tasks [9].

However, ML models are not as reliable as many may believe. Recently, several unfair decisions made by ML models have aroused wide attention. For example, a computer program sends out a score to predict each person's likelihood of committing a crime in the future. The unfairness of the machine may lead to wrong judgment. For example, a black man and a white man are jailed for theft, in which the black man is rated as high-risk and the white man is rated as low-risk. Two years later, the black man did not commit a crime, and the white man was jailed for theft again [10]. These unfair decisions could harm innocent people, threaten disadvantaged groups, and even lead to political conflicts [11]. However, scientists still have controversies and discussions on evaluating the fairness of machine learning models in judicial decision-making [12]. Since there may be biases in the raw training data set of ML models, such models may introduce unexpected social biases, violating the non-discrimination policies of the respective organizations or countries. As a human-centered society, ML fairness becomes an important concern, especially in decision-critical scenarios. Fairness is the essential requirement of the judiciary, and judicial injustice will damage the credibility of the relevant departments. Consequently, it is necessary and important to validate ML models to check for possible discrimination. Thus, measuring the ML models becomes a crucial yet challenging task.

Although some general-purpose fairness concepts are recently proposed [13-14], they mostly rely on data to calculate the fairness metrics, which barely have a practical meaning regarding the judicial trial. Only a few research [15-16] are conducted for judicial trials. LInCo [15] is a legal inconsistency coefficient, which can be applied to judge the inconsistency of data features. It aims to evaluate the inconsistency between data groups divided by specific features, such as gender, region, and ethnicity. Although LInCo has a certain effect in the field of judicial fairness, the features of the data may be a little limited due to the complexity of judicial documents. Another existing study [16] proposed two algorithms based on counterfactual fairness and causal judgment aggregation theory to ensure that the aggregation probability causal model meets the fairness standard is generated, but the verified models are minimal. Besides, existing research lacks evaluation of the relationship between fairness and accuracy of the ML models, making it difficult to balance the relationship between the two.

To overcome the aforementioned challenges and shortages, we design a set of fairness metrics for ML models from different perspectives and experimentally verify the effectiveness of the metrics in this paper. To bridge the gap in judicial fairness of ML, we propose a set of fairness measures

*Corresponding Author: Qiang Geng; E-mail: gengq@nju.edu.cn
DOI: 10.53106/160792642022092305019

from multiple portrayals, including *AmpScore*, *Fairness deviation*, *Normalized fairness*, and *Fairness Score*, specially designed for the judicial field. The proposed metrics are based on the normative characteristics of judicial data, comprehending consideration of the accuracy and fairness of the judicial model. $AmpScore$ obtains a definition of accuracy suited to the judicial field by scaling errors. $Fairness\ deviation$ can be used to define fairness, while $Normalized\ fairness$ is more suitable for evaluation as a percentage data. Eventually, $Fairness\ Score$ gives a quantitative fairness score based on accuracy.

To verify the effectiveness of our proposed metrics, we conducted a comparative study on seven ML models widely used in judicial trials. We performed real-world punishment prediction analysis on more than 2 million judicial documents. The tested models include classic machine learning models (SVC, LinearSVC, RandomForest) and deep learning models (CNN, Text-CNN, Attention-CNN, and ResNet). Through a series of empirical experiments, we have found the following conclusions through our experiments. First, we find that classic machine learning models (SVC, LinearSVC, RandomForest) perform well in terms of fairness. In contrast, deep learning models (CNN, Text-CNN, Attention-CNN, and ResNet) show their advantage in terms of accuracy. Second, among the ML models we studied, attention-CNN and ResNet can earn relatively fair predictions without losing accuracy.

The contributions of this paper can be summarized as follows:

- **Approach**. We propose a novel metrics framework for fairness analysis of ML justice systems. The proposed metrics are specifically for evaluating the judgment of judicial documents.
- **Dataset**. We evaluated and compared the fairness performance of the ML model widely used in the current judicial field on large-scale real judicial data.
- **Study**. After comparing classic machine learning models with deep learning models, we find that attention-CNN and ResNet obtain higher fairness without losing much accuracy.

The remainder of this paper is structured as follows. Section2 elaborates on the metrics we proposed for the fairness evaluation of ML models in the judicial domain. Section 3 illustrated our study and methodology in detail and reported the experimental results. In Section 4, we discuss the related works about ML fairness. Finally, Section 5 concludes the paper and elaborates on our future work.

2 Methodology

Figure 1 visualizes the entire workflow of our evaluation of the ML model. First, the judicial model can be trained with the raw judicial dataset and ML training algorithm as the model input. Second, through a series of data preprocessing operations, such as word segmentation, text cleaning, and standardization, the raw judicial dataset can be transformed into judicial data transformation, which is more suitable for the judicial model. Third, input the data into the trained ML model to calculate the accuracy of the model. Then, according to the accuracy results of the model, the fairness deviation can be generated. Finally, through the result of fairness deviation, we can get fairness definition and calculate normalized fairness. Meanwhile, combining the normalized fairness and model accuracy, the final model fairness score can be calculated as the final output.

2.1 Definition

In general, fairness could be rather difficult to define formally. Different domains may have different views regarding fairness. Different people from the same domain may still view fairness differently.

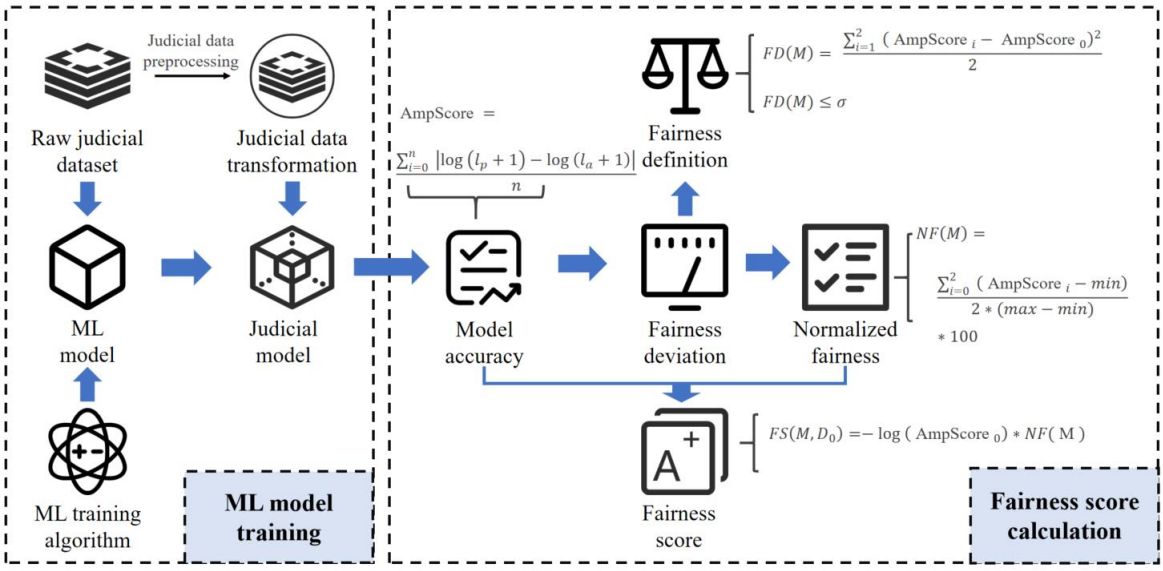

Figure 1. The workflow of ML fairness evaluation

The definition of fairness used in the information system that widely accepted recently is counterfactual fairness. When it comes to counterfactual fairness in the judicial field, it means that any two valid judicial cases that differ only in the protected or sensitive attribute are always condemned to the same penalty. If it yields different sentences for some pair of valid judicial cases, bias is said to exist [17]. Considering the application of the deep learning classification model in

judicial penalty prediction, fairness can be summarized into the following two propositions.

Proposition 1. *If a model M is fair of the attribute set A on dataset D, the absence of A will not affect the performance of M.*

Proposition 1 provides an ideal application scenario. In this scenario, we can ensure that the set of attributes unrelated to the final decision is isolated from the decision system. But this approach has certain limitations since we can't guarantee that there are no attributes unrelated in the judgment documents. Proposition 1 successfully shielded the unfairness but did not solve the unfairness; that is what 2 tries to consider.

Proposition 2. *If a model M is fair of the attribute set A on dataset D, the specific value of the attribute set of A will not affect the performance of M, even the counterfactual values.*

Proposition 2 attempts to ensure the fairness of the ML system by restoring the randomness of irrelevant sensitive attributes. This proposition more realistically reproduces the requirements of fairness in the judicial field.

Satisfying Proposition 1 can significantly improve fairness but is not in line with the actual situation. The charge of judicial personnel is to keep a complete record of the cases. Proposition 2 trades the ideal situation at the expense of fairness. Once these sensitive attributes appear in the dataset, we cannot guarantee that the ML will not extract features from it. Therefore, the investigation of the fairness of ML justice needs to take both of them into consideration instead of only one of them.

Based on the propositions above, we propose a quantitative definition of the fairness for judicial ML systems. With the same dataset and different ML algorithms, we can train the models, respectively. To study the fairness performance of each model, we apply each model on the same testing sets, namely the testing set D_0 obtained from the original dataset, the D_1 was modified by the original data set according to Proposition 1, and the D_2 was modified by the original data set according to Proposition 2. We examine each model and explore the potential relationships between them regarding fairness.

2.2 Metrics

To amplify the deviation on the model, we propose an indicator named $AmpScore$ with logarithmic function transformation.

Definition 1. $\boldsymbol{AmpScore}$. *$AmpScore$ evaluates the accuracy of the model by zooming in or out on the impact of difference on the results. An $AmpScore$ of each model is defined as:*

$$AmpScore = \frac{\sum_{i=0}^{n} |log(l_p+1) - log(l_a+1)|}{n} \quad \textbf{(1)}$$

in which l_p denotes the predicted terms of the penalty, l_a denotes the anticipated terms of the penalty, and n denotes the amounts of the cases.

We use Propositions 1 and 2 to initialize the test set D_0 and obtain the unbiased test set D_1, D_2. In test set D_1, the sensitive attributes are eliminated. In test set D_2, the sensitive attributes are distributed randomly. Through $AmpScore$ calculation on D_0, D_1, D_2, we obtain $AmpScore_0$, $AmpScore_1$, $AmpScore_2$, respectively. To further investigate whether the ML model is fair, we adopt the fairness deviation function defined as follows.

Definition 2. $\boldsymbol{Fairness\ deviation.}$ *Fairness deviation measures to what extent the fair deviation is for an ML model M. which is defined as:*

$$FD(M) = \frac{\sum_{i=1}^{2} (AmpScore_i - AmpScore_0)^2}{2} \quad \textbf{(2)}$$

So far, we have defined a quantitative function to describe the fairness of the ML model. Based on this, we can define ML judicial fairness under the threshold of an FD value. Our next question is how to use this function. We usually use other metrics like a gap to describe it in detail. At the same time, the definition of the limit tells us that when a gap is small enough, we can approximate it as if they are equal. From this, we can define ML judicial fairness based on the judgment document.

Definition 3. $\boldsymbol{Fairness}$. *Given a very small constant σ, we say an ML model is fair if its fairness deviation (FD) satisfies: $FD(M) \leq \sigma$.*

In particular, when D_2 is generated by counterfactual augmentation, this fairness can meet counterfactual fairness. The specific value of σ depends on the requirements of the judicial system to which it is applied. The Fairness deviation function gives a range that defines whether the model is fair. There are only two outcomes of the function, namely, fair or unfair. However, we need to investigate the relative fairness of multiple models for ML model selection instead of only determining whether a model is fair or not.

Therefore, we propose another metric, the Normalized fairness function, to normalize the results of fairness deviation and obtain continuous results among multiple models.

Definition 4. $\boldsymbol{Normalized\ fairness}$. *Normalized fairness quantifies the fairness of a model and scales it to a value between 0 and 100. The NF (normalized fairness) of a model M can be defined as*:

$$NF(M) = \frac{\sum_{i=0}^{2} (AmpScore_i - min)}{2 * (max - min)} * 100 \quad \textbf{(3)}$$

where max is the maximum value among $AmpScore_0$, $AmpScore_1$, and $AmpScore_2$ and min is the minimum value among them.

From Definition 1, we can see that the smaller the $AmpScore$ is, the higher the model's accuracy. By Definition 2 and 4, FD and NF reflect the model's fairness by eliminating the irrelevant effects. Unlike FD, the NF is also related to the overall deviation of the model.

However, the definition mentioned above separates the accuracy and fairness of the model. Trading fairness at the cost of significantly reduced accuracy is not a satisfactory solution. It causes the consequences of violating the original intention of the ML judicial system. The damage caused by the wrong, fair trial will not be less than the correct unfair trial. Therefore, we further define a metric to reflect the fairness score based on accuracy.

Definition 5. $\boldsymbol{Fairness\ Score}$**.** *The* $Fairness\ Score$ *evaluates the fairness of model M on the original data set* D_0 *with the accuracy as the premise.*

$$FS(M,\ D_0) = -log(AmpScore_0) * NF(M) \qquad \textbf{(4)}$$

Through the above method, we can obtain a fairness score on different ML models. Combined with the analysis of the models trained by different ML algorithms, we will further explore the reasons for the fairness differences of each model.

3 Experiments

In this section, we first describe the experiment's settings, including dataset, experiment design, and setup, followed by the discussion of experimental results on our approach.

3.1 Dataset

We use a public large-scale dataset from CAIL2018 of over 2 million judicial documents [18], which will be the inputs to the machine learning model. Therefore, the input data is a textual description of the judicial cases, and the output value represents the time of sentencing, an integer ranging from 0 to 240 months. Specifically, 0 represents acquittal, and 240 represents the maximum sentence specified by law, i.e., twenty years (240 months). The model prediction is regarded as correct when the predicted sentencing time output by the model is consistent with the label; otherwise, the model prediction is incorrect.

After fractionation and vectorization on the documents, we perform a comparative study on the ML models trained from seven widely used training algorithms by evaluating the terms of the penalty. Our task is to test the fairness of several models using different ML algorithms by evaluating the terms of the penalty. We consider that judicial documents with similar terms are suitable for the recommendation. Each document comes from objective real judicial facts, split into training and testing sets. We evaluate $AmpScore$, FD, NF, and FS on testing sets D_0, D_1, and D_2, leading to a credible method of testing fairness.

3.2 Setup

We use eight groups of counterfactual tokens to modify the original testing set D_0 into D_1 and D_2. We evaluate our methods for several classic machine learning and deep learning models in terms of the penalty prediction task. It's noteworthy that CNNs are widely considered to show optimal performance in this prediction, including CNN, Text-CNN, Attention-CNN, and ResNet. Therefore, we apply a variety of CNN variant models to evaluate their accuracy and fairness, respectively.

As for classic machine learning models, we consider several classification models, including SVC, LinearSVC, and RandomForest. We apply word2vec [19] to pre-train the natural language and set the length of the word embedding as 512. Each classification model is trained for 15 epochs as well. As a result, we get the models for evaluation on test sets D_0, D_1, and D_2. The description of the ML models are listed as follows:

- **CNN**: The Basic CNN model usually replaces each word in the sentence with a vector representation and creates a sentencing matrix.
- **Text-CNN**: The convolutional neural network can automatically combine and filter Text local features to obtain semantic information of different abstract levels.
- **Attention-CNN**: Attention-CNN introduces an attention layer between the input layer and the convolutional layer, creating a context vector for each word to be spliced with the word vector as a new representation of the word.
- **ResNet:** Compared with the ordinary network, ResNet adds a short-circuit mechanism between every two layers, which forms residual learning, where the dotted line indicates that the number of feature maps has changed.
- **SVC**: SVC is a two-class classification model. Its basic model is defined as the linear classifier with the largest interval in the feature space.
- **LinearSVC**: LinearSVC is another implementation of support vector machines, mainly used in the case of linear kernel functions.
- **RandomForest**: RandomForest is a classifier that can well predict the effects of up to thousands of explanatory variables.

3.3 Result

Figure 2 depicts the relationship between the accuracy and fairness of different studied models. Table 1 summarizes the $Fairness\ deviation$ under different models. Figure 3 the fairness scores of several models evaluated in this paper regarding accuracy in the form of a tree diagram. The larger the area, the larger the model score.

As can be seen from the above, $AmpScore$ reflects the accuracy of the model. The larger the $AmpScore$, the lower the accuracy of the model. For easy observation, we plot the value of one minus $AmpScore$ on the chart in Figure 2. The larger the histogram value, the higher the accuracy of the model. The NF value describes the fairness of the model. The larger the NF value, the higher the accuracy of the model. We use different colors to mark different dataset results. It is not difficult to find that between the same model, the accuracy obtained by using the initial data set is slightly lower than that obtained by using the modified data sets D1 and D2 after definition. The accuracy of the first four algorithms (CNN, text CNN, attention CNN, and RESNET) used the D2 is higher than that used D0 and D1. When reducing the deviation of fairness caused by decision-independent attributes, Proposition 1 erases the existence of these attributes, thus avoiding the dependence of the algorithm on irrelevant data. Although Proposition 2 does not completely erase attribute set A, it reduces the bias of the original data set by random and uniform data generation to achieve better accuracy. The difference between the accuracy of the three data sets of the classic ML models (SVC, LiearSVC, RandomForest) is not as obvious as that of deep learning models (CNN, text CNN, attention CNN, and RESNET), because the dependence of the ML on data is not as deep as the DL. The value of NF reflects the fairness and overall deviation of the model. It can be seen that the fairness of the ML is lower than that of the DL, but the overall deviation is small.

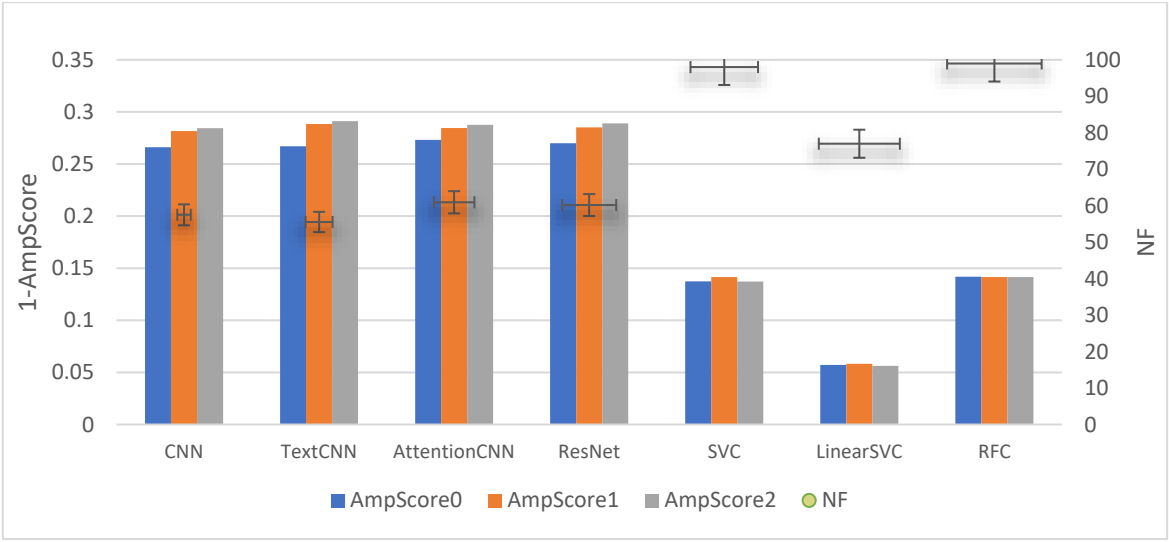

Figure 2. Distribution of accuracy (Ampscore) and fairness (NF)

Table 1. The fairness deviation under different models

ID	Model	Fairness deviation
1	CNN	0.00028752
2	Test-CNN	0.00052197
3	Attention-CNN	0.00017000
4	ResNet	0.00030039
5	SVC	0.00000862
6	LinearSVC	0.00000090
7	RandomForest	0.00000010

Table 1 summarizes the *Fairness deviation* under different models. The larger the FD value, the lower fairness of the model. We can see that models obtained by the classic ML models (SVC, LiearSVC, RandomForest) have a smaller offset. By Definition 3, the FD value entirely depends on the requirements of the actual application scenario. When the requirements are broad enough, such as 0.0003, most models can meet fairness requirements. In terms of FD values, the performance of LinearSVC is better than SVC. However, due to the larger overall deviation of LinearSVC, SVC performs better on NF values.

In Figure 3, the area of the space denotes the fairness score of each model. From Figure 2, we can see that CNN and Text-CNN do well in terms of accuracy but worse in fairness. However, we need to consider accuracy as a prerequisite to consider fairness, which is the purpose of the FS indicator. In Figure 3, they (CNN and Text-CNN) get the lower FS value.

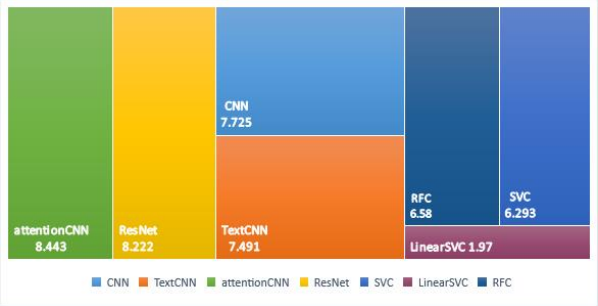

Figure 3. Fairness score of the models

In contrast, in terms of fairness performance, the performance of classic machine learning methods (SVC, LiearSVC, RandomForest) is significantly better than deep learning models (CNN, Text-CNN, Attention-CNN, ResNet), but the accuracy is relatively disappointing. Combining accuracy with fairness, the ideal model is attention-CNN and ResNet.

3.4 Discussion

It can be seen from the experimental results that the accuracy of the deep learning model is more advantageous, and the classic machine learning model does better in fairness. Tracing back to the reason, the classic machine learning model was born to solve the needs of machines to automatically extract features and learn [20-21]. But in the foreseeable era of big data, when faced with massive amounts of data, the advantages of deep learning will become prominent.

Surprisingly, classic machine learning models have a smaller impact on fairness than deep learning models. The reason for this may be that the classic machine learning model has fewer training layers than the deep learning model. The more layers of training, the higher the accuracy of the model, the more likely it is to incorporate sensitive information into the decision process. Deep learning is the type of machine learning model on complex artificial neural networks. The basic idea is to stack multiple layers, which means that the output of this layer serves as the input to the next layer. But in the process of layer-by-layer learning, the model learns through surface clues, and the model believes that these clues help to accomplish a specific task. This led to the loss of fairness. Deep learning often "learns inappropriate knowledge" compared to classic machine learning. Once there is a deviation in the distribution of the data in the original training set, a deep learning model has a higher probability of amplifying these deviations.

4 Related Work

In this section, we compare our study to prior work from two aspects of fairness evaluation and model testing of ML models.

4.1 Fairness of Machine Learning Models

This work stems from our early study on judgment prediction [9]. Many researchers tried to define the fairness of the ML model from a more general perspective. Kusner et al. [22] proposed a set of definitions of fairness, individual fairness, counterfactual fairness and elaborated on them in detail. Berk et al. [23] use discrete data, cost ratios to highlight the different effects of misjudgment of the same value in different degrees of justice in the judicial-related ML system. Kilbertus et al. [24] provide a solution to examine sensitive attributes and avoid disparate impact and treatment when it comes to transforming attribute sets. Du et al. [25] calibrated the prediction of the model during the inference process by forcing the prediction distribution to be close to the training distribution or a specific fairness index. In practice, the definition of fairness does not depend entirely on technologists and ethicists. Fairness originates from the public and eventually applies to the public. Saxena et al. [26] found that calibrated fairness and "ratio" decision win more favor among the public, which is a reason we use the ratio to normalize the fairness score.

However, ML justice has a unique need for group fairness, which is also a motivation for our work. Unlike the above-mentioned existing fairness analysis approaches, our proposed approach aims at the task of judicial penalty prediction. We design a fairness analysis method that conforms to the features of fairness in the judicial field.

4.2 Testing the Machine Learning Models

The fairness evaluation of the ML model is essentially a test dimension of the model. DeepXplore [27], a white box differential testing algorithm, is used to systematically find inconsistent inputs that may trigger multiple deep neural networks (DNNs). Neuron coverage is used as a system indicator to measure how much internal logic of the DNN has been tested. DeepTest [28] used deformation relations to identify wrong behaviors in DNNs. The use of the deformation relationship solves the limitation of the different tests to some extent, especially the requirement of having multiple DNNs that achieve the same function. The black box testing approach proposed by Zafar et al. [29] can be used to verify the security of deep neural networks. The fairness evaluation proposed by Sakshi et al. [13] uses their approach to evaluate the robustness of neural networks in applications where safety is critical, such as traffic sign recognition. Zhang et al. [30] proposed Adversarial Discrimination Finder (ADF) to generate examples by adversarial sampling and search unfairness based on a specific probability distribution.

Different from the above-mentioned neuron coverage calculation metrics, the metrics proposed in this paper mainly aim at the fairness analysis of the machine learning models. To analyze and evaluate the current mainstream machine learning models, we separately evaluated the accuracy and fairness of each model. The results of the comparative experiment show that the classic machine learning models (SVC, LinearSVC, RandomForest) perform well in terms of fairness, while deep learning models (CNN, Text-CNN, Attention-CNN, and ResNet) show their advantage in the accuracy. Besides, among the ML models we studied, Attention-CNN and ResNet can earn fair prediction without the loss of accuracy.

5 Conclusion

The trend of ML justice is urgently demanding, and judicial justice is the core point of whether the judicial system can be recognized. To be able to evaluate the fairness of the ML model, we first proposed some metrics to measure the fairness of the model. We study some models using ML algorithms with higher recognition and popularity and compare their fairness using these metrics. Through experiments, we find that deep learning models perform better in terms of accuracy, while classic machine learning performs better in terms of fairness. To better meet actual needs, we try to achieve higher fairness while maintaining high accuracy. In this regard, we recommend Attention-CNN and ResNet, which earn the highest score of 8.443 and 8.222.

Judicial egalitarianism makes sense in our proposed fairness measures. In future work, we will explore ways of other performance of discrimination. Besides, based on our current research on model fairness, we will try to enhance the model to achieve higher accuracy and improve the current ML judicial fairness.

Acknowledgments

The work is supported by Science, Technology and Innovation Commission of Shenzhen Municipality (CJGJZD20200617103001003).

References

[1] W. Zhao, R. Chellappa, P. J. Phillips, A. Rosenfeld, Face Recognition: A Literature Survey, *ACM computing surveys (CSUR)*, Vol. 35, No. 4, pp. 399-458, December, 2003.

[2] P. F. Brown, J. Cocke, S. D. Pietra, V. J. D. Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, P. S. Roossin, A Statistical Approach to Machine Translation, *Computational linguistics*, Vol. 16, No. 2, pp. 79-85, June, 1990.

[3] D. Li, W. E. Wong, W. Wang, Y. Yao, M. Chau, Detection and Mitigation of Label-flipping Attacks in Federated Learning Systems with KPCA and K-means, *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, Yinchuan, China, 2021, pp. 551-559.

[4] M. Maurer, J. C. Gerdes, B. Lenz, H. Winner, *Autonomous driving: technical, legal and social aspects*, Springer Nature, 2016.

[5] D. D. Hema, K. A. Kumar, An Optimized Intelligent Driver's Aggressive Behaviour Prediction Model Using GA-LSTM, *International Journal of Performability Engineering*, Vol. 17, No. 10, pp. 880-888, October, 2021.

[6] Z. Guo, J. Liu, T. He, Z. Li, P. Zhangzhu, TauJud: Test Augmentation of Machine Learning in Judicial Documents, *Proceedings of the 29th ACM SIGSOFT International Symposium on Software Testing and Analysis*, Virtual Event, USA, 2020, pp. 549-552.

[7] Z. Liu, H. Chen, A Predictive Performance Comparison of Machine Learning Models for Judicial Cases, *2017 IEEE Symposium series on computational intelligence (SSCI)*, Honolulu, HI, USA, 2017, pp.1-6.

[8] J. Yang, Analysis and Application of Judicial Data Based on Machine Learning, *Journal of Physics: Conference Series*, Vol. 1846, No. 1, Article No. 012027, March, 2021.

[9] T. He, H. Lian, Z. Qin, Z. Chen, B. Luo, PTM: A Topic Model for the Inferring of the Penalty, *Journal of Computer Science and Technology*, Vol. 33, No. 4, pp. 756-767, July, 2018.

[10] J. Angwin, J. Larson, S. Mattu, L. Kirchner, *Machine Bias*, ProPublica, May, 2016.

[11] Z. Obermeyer, B. Powers, C. Vogeli, S. Mullainathan, Dissecting racial bias in an algorithm used to manage the health of populations, *Science*, Vol. 366, No. 6464, pp. 447-453, October, 2019.

[12] R. Courtland, Bias detectives: the researchers striving to make algorithms fair, *Nature*, No. 558, No. 7710, pp. 357-360, June, 2018.

[13] S. Udeshi, P. Arora, S. Chattopadhyay, Automated Directed Fairness Testing, *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, Montpellier, France, 2018, pp. 98-108.

[14] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A Survey on Bias and Fairness in Machine Learning, *ACM Computing Surveys*, Vol. 54, No. 6, pp. 115:1-115:35, July, 2022.

[15] Y. Wang, C. Xiao, S. Ma, H. Zhong, C. Tu, T. Zhang, Z. Liu, M. Sun, Equality before the Law: Legal Judgment Consistency Analysis for Fairness, *Computing Research Repository (CoRR)*, Vol. abs/2103.13868, 2021.

[16] F. M. Zennaro, M. Ivanovska, Pooling of Causal Models under Counterfactual Fairness via Causal Judgement Aggregation, *Computing Research Repository (CoRR)*, abs/1805.09866, 2018.

[17] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, D. Saha, Black Box Fairness Testing of Machine Learning Models, *Proceedings of the ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/SIGSOFT) (FSE)*, Tallinn, Estonia, 2019, pp. 625-635.

[18] C. Xiao, H. Zhong, Z. Guo, C. Tu, Z. Liu, M. Sun, Y. Feng, X. Han, Z. Hu, H. Wang, J. Xu, Cail2018: A Large-scale Legal Dataset for Judgment Prediction, *Computing Research Repository (CoRR)*, abs/1807.02478, 2018.

[19] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient Estimation of Word Representations in Vector Space, *the 1st International Conference on Learning Representations (ICLR)*, Scottsdale, Arizona, USA, 2013, pp. 1301.3781.

[20] J. Devlin, M. Chang, K. Lee, K. Toutanova, Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, MN, USA, 2019, pp. 4171-4186.

[21] M. A. Beghoura, Software Engineering Teamwork Data Understanding using an Embedded Feature Selection, *International Journal of Performability Engineering*, Vol. 17, No. 5, pp. 464-472, May, 2021.

[22] M. J. Kusner, J. R. Loftus, C. Russell, R. Silva, Counterfactual fairness, *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017, pp. 4066-4076.

[23] R. Berk, Accuracy and Fairness for Juvenile Justice Risk Assessments, *Journal of Empirical Legal Studies*, Vol. 16, No. 1, pp. 175-194, March, 2019.

[24] N. Kilbertus, A. Gascón, M. J. Kusner, M. Veale, K. P. Gummadi, A. Weller, Blind justice: Fairness with Encrypted Sensitive Attributes, *Proceedings of the 35th International Conference on Machine Learning, (ICML)*, Stockholm, Sweden, 2018, pp. 2635-2644.

[25] M. Du, F. Yang, N. Zou, X. Hu, Fairness in Deep Learning: A Computational Perspective, *IEEE Intelligent Systems*, Vol. 36, No. 4, pp. 25-34, July-August, 2021.

[26] N. A. Saxena, K. Huang, E. DeFilippis, G. Radanovic, D. C. Parkes, Y. Liu, How do Fairness Definitions fare? Examining Public Attitudes towards Algorithmic Definitions of Fairness, *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, Honolulu, HI, USA, 2019, pp. 99-106.

[27] K. Pei, Y. Cao, J. Yang, S. Jana, DeepXplore: Automated Whitebox Testing of Deep Learning Systems, *Communications of the ACM*, Vol. 62, No. 11, pp. 137-145, November, 2019.

[28] Y. Tian, K. Pei, S. Jana, B. Ray, Deeptest: Automated Testing of Deep-neural-network-driven Autonomous Cars, *Proceedings of the 40th international conference on software engineering*, Gothenburg, Sweden, 2018, pp. 303-314.

[29] M. B. Zafar, I. Valera, M. G. Rodriguez, K.P. Gummadi, Fairness Constraints: Mechanisms for Fair Classification, *Artificial Intelligence and Statistics*, Fort Lauderdale, Florida, USA, 2017, pp. 962-970.

[30] P. Zhang, J. Wang, J. Sun, G. Dong, X. Wang, X. Wang, J. S. Dong, T. Dai, White-box Fairness Testing through Adversarial Sampling, *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, Seoul, South Korea, 2020, pp. 949-960.

Biographies

Yanjun Li engages in software testing and evaluation research at Beijing University of Posts and Telecommunications. His research interests include software testing and evaluation, software testing laboratory accreditation management, intelligent justice, and system testing and evaluation in the field of judicial big data.

Huan Huang received her bachelor's degree in engineering from Central South University, in 2020. She is currently a Research Assistant with the Shenzhen Research Institute of Nanjing University. Her research interests include software engineering, data analysis, and artificial Intelligence.

Qiang Geng received his Ph.D. in Economics from Nanjing University in 2004. Now he is engaged in teaching and scientific research at the Population Research Institute of Nanjing University. His research interests include national economics, the economics of transition and development.

Xinwei Guo received his master's degree in software engineering from the School of Software Engineering, Beijing University of Technology, in 2016. His research interests include artificial intelligence testing, big data testing, other testing technologies, and the research of quality standards of court information technology.

Yuyu Yuan received her M.S. degree from the University of Electronic Science and Technology of China and a Ph.D. degree from the Research Institute for Fiscal Science, Ministry of Finance. She is a Professor at the Beijing University of Posts and Telecommunications. Her research interests include software quality, software testing.