10-30-2023

# Search-Based Fairness Testing: An Overview

Hussaini Mamman
*Universiti Teknologi Petronas*, hussaini_21000736@utp.edu.my

Shuib Basri
*Universiti Teknologi Petronas*, shuib_basri@utp.edu.my

Abdullateef Balogun
*Universiti Teknologi Petronas*, abdullateef.ob@utp.edu.my

Abdullahi Abubakar Imam
*Universiti Brunei Darussalam*, abdullahi.imam@ubd.edu.bn

Ganesh Kumar
*Universiti Teknologi Petronas*, ganesh_21000736@utp.edu.my

*See next page for additional authors*

IEEE International Conference on Computing (ICOCO 2023), Langkawi Island, Malaysia, pp. 89-94, October 2023

# Search-Based Fairness Testing: An Overview

*Hussaini Mamman
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
hussaini_21000736@utp.edu.my

Shuib Basri
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
shuib_basri@utp.edu.my

Abdullateef O. Balogun
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
abdullateef.ob@utp.edu.my

Abdullahi Abubakar Imam
School of Digital Sciences
Universiti Brunei Darussalam
Brunei Darussalam, Brunei
abdullahi.imam@ubd.edu.bn

Ganesh Kumar
Department of Computer and Information Sciences
Universiti Teknologi PETRONAS
Seri Iskandar, Malaysia
ganesh_21000736@utp.edu.my

Luiz Fernando Capretz
Department of Electrical and Computer Engineering
Western University, London, Canada.
lcapretz@uwo.ca

***Abstract*—Artificial Intelligence (AI) has demonstrated remarkable capabilities in domains such as recruitment, finance, healthcare, and the judiciary. However, biases in AI systems raise ethical and societal concerns, emphasising the need for effective fairness testing methods. This paper reviews current research on fairness testing, particularly its application through search-based testing. Our analysis highlights progress and identifies areas of improvement in addressing AI systems' biases. Future research should focus on leveraging established search-based testing methodologies for fairness testing.**

***Keywords—fairness, fairness testing, search-based fairness testing***

## I. INTRODUCTION

Artificial Intelligence (AI)-based systems have gained popularity over time. They are now integral components of many software systems, including those used for medical diagnoses, policing, loan approvals and risk assessments [1]. However, the rapid growth of AI-based systems in areas directly involving humans has raised concerns about their adoption's potential risks and challenges. One of the significant risks associated with AI is the possibility of discrimination and bias in their decision-making processes [2], [3].

A classic example comes from an AI-based system used by US courts to make pretrial detention and release decisions [4]. An examination of the system, Correctional Offender Management Profiling for Alternative Sanctions (COMPAS), revealed racial discrimination against black Americans [5]. Similarly, an AI system used to screen job applicants has been found to favour specific candidates over others based on gender [6]. In addition, minority homebuyers are reported to face widespread lending discrimination, causing 80% of Black mortgage applicants to be denied [7]. These cases where AI-based systems are found to be discriminatory are numerous, making it imperative to ensure that decisions made by AI-based systems are fair.

Fairness refers to the ability of AI-based systems to avoid biases and prevent discrimination [8]. The aim is to ensure that these systems produce fair outputs and behaviours for all inputs relevant to the task at hand, regardless of sensitive attributes like gender, race, or age. Specifically, the software should not discriminate against certain groups or individuals based on their characteristics [9]. Fairness has become a crucial requirement for AI-based systems to ensure their trustworthiness and widespread adoption [10], [11], making it imperative to test AI-based systems for fairness.

Fairness in AI-based systems cannot be guaranteed solely by developing better machine learning (ML) algorithms [2], as the discrimination can originate from various sources [12]. These include biases in the training dataset or the algorithm, and hyperparameters used to train the ML model [13].

Software testing is essential to software development, ensuring reliability and quality, and fairness testing can help detect and fix fairness issues in AI-based systems [3]. Fairness testing is a branch of software testing that evaluates how an AI-based system treats individuals or groups fairly and without discrimination [14]. Fairness testing aims to uncover as many discriminatory inputs as possible in an AI-based system.

Various software testing methods have been applied for fairness testing [15]. Combinatorial testing focuses on input interactions but can face combinatorial explosion [16]. Verification-based testing relies on constraint-solving techniques but is less scalable and resource-intensive [17]. A promising approach is search-based testing (SBT), which efficiently explores input spaces to detect challenging bugs [14], [18], [19]. For this, many fairness testing approaches employed SBT techniques to detect discrimination in AI-based systems. This study provides an overview of the fairness testing approaches that used SBT.

The rest of the paper is organised as follows. Section II gives a background of AI-based systems, fairness testing, and search-based fairness testing (SBFT). Section III highlights the methodology of the study, while results and discussions are provided in Section IV. Section V highlights some potential research directions. Section VI gives a concluding remark for the paper, while section VII finally displays the references used in the study.

## II. BACKGROUND

This section introduces AI-based systems, fairness testing and fairness testing life cycle. SBFT and its workflow are also discussed.

### *A. AI-based Systems*

AI-based (or just AI) systems are software applications with at least one AI component to provide functionalities. AI-based systems include various software tools and applications that use machine learning (ML) techniques to analyse data and

develop analytical models. ML is a subfield of AI that focuses on data analysis and provides valuable insights to improve the performance of AI-based systems. Applying ML techniques enhances AI-based systems' accuracy, efficiency, and reliability[20].

AI-based systems include systems such as hiring decision support systems and healthcare systems that are used to diagnose patients. Image or speech recognition and autonomous driving are other examples [21].

### B. *Fairness in AI-Based System*

Fairness in decision-making is the absence of bias or preference toward a person or group based on their inherent or acquired attributes [9]. Two notable definitions of fairness have been employed in fairness literature: individual fairness and group fairness. Individual fairness ensures that any two people who are similar in terms of a similarity metric defined for a particular task should get the same result [22]. For example, a loan software would be considered fair when it grants a loan to two individuals with similar characteristics, irrespective of their protected attributes. Individual fairness is more frequently addressed in fairness testing [22].

On the other hand, Group fairness involves ensuring that the distribution of outputs is similar across different groups based on a particular input characteristic. This approach seeks fairness by ensuring that the same proportion of individuals within various groups receives a specific outcome [2]. For example, in group fairness, a loan software would be considered fair with respect to age if it approves loans for the same proportion of applicants under 40 and over 40 years of age.

### C. *Fairness Testing*

Fairness testing is crucial in the software engineering process as it ensures high-quality and reliable AI-based systems are developed and deployed. Fairness testing involves executing test cases to uncover discrimination in AI-based systems.

Fairness testing aims to assess the level of fairness a classifier provides by automatically generating test instances and using them to identify possible instances of discrimination [23]. Input instances that exhibit bias towards individuals with similar characteristics are called discriminatory instances [24]. These instances highlight the presence of unintended discrimination within AI systems and are a focus of concern in individual fairness testing and analysis.

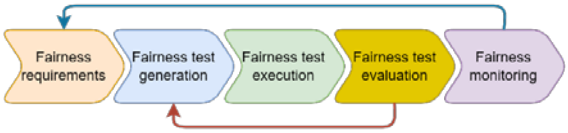


Fig. 1: Fairness testing life cycle [15]

#### 1) *Fairness Testing Life Cycle*

The process of fairness testing involves a series of testing activities that detail its implementation [14]. As depicted in Fig. 1, software engineers determine and specify the desired fairness requirements for the AI-based system under test (SUT) via requirements engineering. Then, they identify and construct test oracles in alignment with these fairness criteria and generate or sample test inputs from the available data. The engineers then execute the test inputs on the SUT to determine if the test oracles are satisfied. They assess the tests' efficacy in revealing fairness bugs and use the bug report generated from the test run results to correct and eliminate such bugs. This fairness testing procedure is repeated until an acceptable level of fairness is attained for the SUT. When the system is deployed to production, continuous monitoring is applied to ensure that fairness requirements are satisfied.

### D. *Search-Based Fairness Testing (SBFT)*

Search-based testing (SBT) methods typically involve systematically exploring the input space of a system to identify specific inputs that trigger certain behaviours or properties. SBT combines automatic test case generation and search techniques to optimise software testing [25]. A test case is an input of variables or conditions a tester uses to confirm that the SUT functions appropriately and meets a given requirement under review [25]. SBT can reduce the time and effort required to create test cases while increasing their effectiveness. It can also enable identifying more defects in the software in less time by selecting the best possible test cases to ensure the software is thoroughly tested [18].

Search-based fairness testing (SBFT) uses SBT techniques to evaluate the degree of discrimination in AI-based systems. It involves employing search algorithms and optimisation techniques to systematically explore the input space of a model and identify potential instances where unfair outcomes occur. The goal is to uncover and address discrimination cases in AI systems that may disproportionately affect specific individuals or groups based on race, gender, or other sensitive characteristics.

## III. METHODOLOGY

In this section, we outline the methodology adopted for conducting the review. We initiated a thorough review of relevant papers published in the ACM & IEEE journals from 2017 to 2023.

### A. *Keyword Search*

We intend to capture emerging trends within the realms of fairness testing research. The search criteria were devised using the keywords in Fig. 2 to ensure a more comprehensive search.

```
("bias" OR "discriminat*" OR "fair*") AND
("test*" OR "detect*" OR "audit" OR "evaluat*" OR "assess*"
OR "verif*" OR "discover*" OR "uncover*" OR "Investigat*" OR
"Examin*" OR "Inspect*" OR "check*") AND
("learn*" OR "software" OR "Artificial Intelligence" OR "AI" OR
"system*" OR "application" OR "Natural language processing"
OR "NLP" OR "Neural networks" OR "Algorithm" OR "Data
mining" OR "computer vision" OR "big data" OR "data-driven"
OR "decision making")
```

Fig. 2: Database search keywords

### B. *The Review Process*

Fig. 3 illustrates the utilised flowchart in the research, which was adapted from Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [26]. After removing duplicate papers and screening based on title and abstract, our review commenced with 53 articles. By applying our predetermined exclusion criteria, 33 articles were eliminated from consideration. As a result, this review is composed of a total of 20 articles. The exclusion criteria used in this study are as follows:

- Irrelevant to fairness testing in machine learning
- Absence of SBT methods for fairness evaluation

- Lack of emphasis on fairness test generation
- Duplicate publications
- Non-English papers

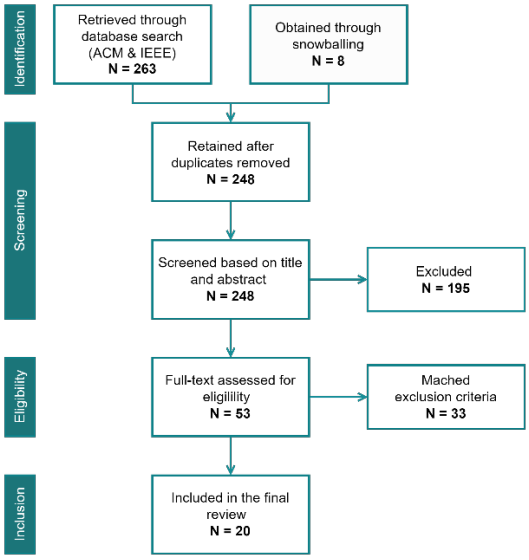


Fig. 3: Study PRISMA diagram

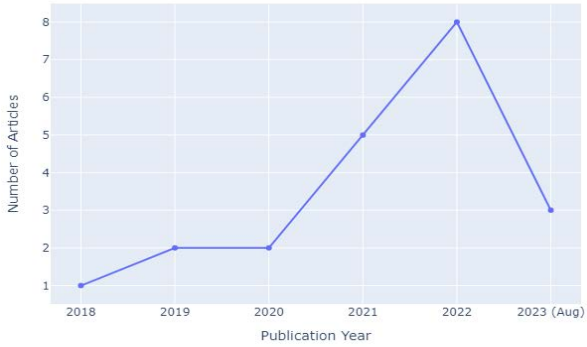


Fig. 4: Distribution of published studies over the years

### C. Data Extraction

A data extraction form is designed to gather information from the selected articles on SBFT. The form has categories encompassing the author, publication year, research problem, core techniques, ML access level, ML task, data type, and evaluation metrics. The extracted data underwent thematic analysis to facilitate evaluation and interpretation, enabling insights from the data to form the research questions.

## IV. RESULT

This section showcases the results derived from the conducted review. The results are structured according to carefully formulated research questions, providing a guided framework for presenting the findings. The research questions are defined and discussed as follows.

***RQ 1:*** *How has the field of search-based fairness testing evolved over the years?*

Fairness testing, specifically SBFT, is still a growing research area. In 2018, one article was published indicating the foundational stage. The subsequent years, 2019 and 2020, witnessed a consistent rise, with two articles showcasing growing interest and maturity. The trend continued in 2021 when five articles were published. Notably, 2022 experienced a significant surge with eight articles, highlighting a substantial boost in research activity. The trend continues in 2023 (until August), with three published articles indicating a dynamic, expanding research area with a growing impact. This trend is depicted in Fig. 4.

***RQ 2:*** *What are the techniques employed for search-based fairness testing?*

An increasing number of fairness testing techniques used SBT to examine the input space of the AI-based system under test. These methods mainly detect individual discrimination and are based on ML classification tasks. For example, AEQUITAS [27] generates random inputs to find discriminatory instances and then applies perturbation to the non-protected attributes of those instances, using probabilistic search to discover neighbouring discriminatory samples. CGFT [16] enhances AEQUITAS by replacing the random search with combinatorial testing to generate a diverse test suite. Similarly, KOSEI [28] replaces the probabilistic search of AEQUITAS with sequential perturbation. SG [29] uses local explainability and symbolic execution to identify and generate discriminatory inputs based on the decision boundaries of SUT. Then, these inputs are perturbed for more discriminatory inputs. RULER [30] simultaneously perturbs sensitive and non-sensitive attributes to identify additional discriminatory instances outside the strict causal relations.

ExpGA [31] employs local interpretability to identify seed instances that could induce discrimination upon flipping protected attributes, efficiently utilising these seeds to generate many discriminatory offspring through a genetic algorithm (GA). Xie and Wu [32] employed reinforcement learning (RL) to create test inputs for fairness testing by treating the ML model under test (MUT) as part of the RL environment. The RL agent alters the environment to produce discriminatory inputs, gauges the environment's state, and offers feedback through rewards. This iteration uncovered optimal strategies for generating effective discriminatory inputs.

In deep learning models, ADF [33], [34] uses gradient-guided search to generate discriminatory instances. Diverse seed instances are selected from clustered samples, and the area around each identified discriminatory instance is harnessed to generate more instances. EIDIG [35] improves ADF by introducing momentum terms to aid escape from local optima for higher success in detecting individual discriminatory instances and utilising prior gradient knowledge to optimise the generation of more discriminatory instances. NeuronFair [36] identifies biased neurons through analysis, generating instances to boost their Activation Difference (ActDif) values and further expanding the set by perturbing identified instances through nearby seed searches. DeepFAIT [37] utilises Generative Adversarial Networks (GANs) to transform images across sensitive domains and identifies fairness-related neurons through ActDif. Then, it generates test samples using image processing strategies.

Some methods address specific issues or different types of bias in addition to generating discriminatory instances. For example, Ma et al. [24] constructed initial discriminatory instances for fairness testing using an interpretable method. In contrast, LIMI [38] centred on creating test inputs based on naturalness. LIMI uses GANs to mimic the target model's decision boundary in a latent space, approximating data distribution with a surrogate linear limit and identifying potential discriminatory instances closer to the actual decision boundary through vector manipulations and calculations. The

*fAux* [39] method is designed to identify historical bias [12], and it achieves this goal by comparing the derivatives of the prediction model with those of an auxiliary model. The auxiliary model estimates the protected variable based on observed data, which helps avoid the need to generate counterexamples and prevents the inclusion of out-of-distribution inputs.

Distinct research focuses on various ML tasks. For instance, in regression-based tasks, Perera et al. [19] employed GA to generate test cases that could potentially unveil biases by assessing the maximum difference between predictions, referred to as the fairness degree. FairRec [40], [41] adopted a dual-particle swarm technique to gauge the maximum gap between user groups in recommendation tasks. This method employs separate swarms: one aimed at the most advantaged group and the other at the most disadvantaged group within the multidimensional search space. In addition, other studies focus on detecting group discrimination. For example, TestSGD [42] uses a rule-based method to measure group discrimination. It applies slight, uniform changes to a randomly sampled input to generate more samples. Then, the samples are used to estimate the statistical parity score between demographic groups. FAIRVIS [43] provides an interactive visual analytics tool that facilitates fairness auditing of SUT by integrating subgroup discovery techniques and performance comparison, aiding detailed investigation through coordinated views.

***RQ 3:*** *Which fairness categories are examined by search-based fairness testing studies?*

From the work of Galhotra et al. [2] in 2017, the domain of SBFT has been primarily characterised by a focus on individual fairness testing. This emphasis is driven mainly by the claim that testing for individual discrimination leads to identifying more discriminatory instances [2]. As can be seen in Fig. 5, Group fairness testing in SBFT has received comparatively less attention (10%), with only two studies [41], [42] dedicated to its exploration.

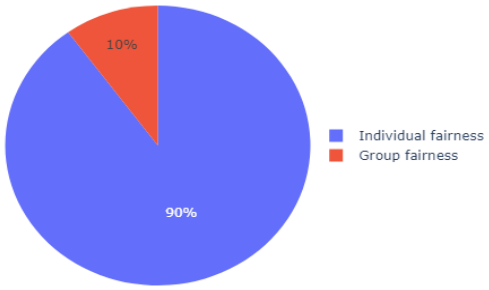


Figure 5: Studies distribution based on fairness category

***RQ 4:*** *Which fairness evaluation metrics are used in search-based fairness testing studies?*

Numerous assessment metrics have been devised in fairness testing, grounded in the foundational concepts of individual and group fairness. When evaluating individual discrimination, research predominantly employs three benchmarks: 1) the quantity of generated test cases, 2) the proportion of detected discriminatory instances, and 3) the time required for test case generation. In studies concentrating on group fairness, the Calders & Verwer (CV) score is commonly utilised to quantify the disparity in misclassification rates between the two evaluated groups.

## V. Recommendation for Future Studies

This section highlights potential research directions that need further exploration in the field of SBFT as follows.

*1) Using metaheuristic algorithms*

SBT techniques systematically explore a system's input space to uncover specific inputs that trigger desired behaviours, often employing established metaheuristic algorithms like genetic GA, particle swarm optimisation (PSO), and simulated annealing. These algorithms use an objective function to guide an efficient search for optimal solutions from a large search space. While a few studies [19], [31], [40], [41] have used GA and PSO for fairness testing, there's a need for more exploration of other well-known metaheuristics in assessing AI systems for fairness.

*2) Multi-objectives testing*

Existing SBFT studies focus on generating test cases that reveal bias. However, ensuring fairness in AI systems often involves balancing multiple objectives, such as fairness metrics, accuracy, and interpretability [44]. Multi-objective optimisation techniques seek to find a set of solutions that represent a trade-off between multiple competing objectives. In fairness testing, this could mean generating inputs that simultaneously expose biases while maintaining model performance, an area that needs further investigation.

*3) Initial seed selection*

The efficiency of generating discriminatory instances is affected by the quality of initial seeds [33], [35]. However, existing studies used random or clustering-based sampling, which may not be optimal. Notably, only one study [24] has explored initial seed selection in fairness testing, highlighting a potential avenue for future research.

*4) Re-using test data*

Re-using test data is essential in software testing for efficiency and quality, but in studies related to SBFT, generating new test data for each protected attribute can lead to redundant cycles and unnecessary waste of time and effort. The possibility of using test cases designed for one attribute to assess another emphasises the need for efficient strategies like test case recycling, including sound pruning [2], to optimise resource utilisation.

*5) Test cost reduction*

Methods for reducing costs, such as test selection, prioritisation, and minimisation, have been explored in SBT [45]. High test costs hinder fairness testing as it involves model retraining, repeated predictions, or extensive data generation. Exploring cost-reduction techniques for more efficient fairness testing is required.

*6) More fairness testing options*

Most SBFT research has concentrated on classification tasks, with limited exploration of regression and recommendation tasks. Other ML tasks like reinforcement learning and unsupervised learning remain under-explored, offering potential for future investigation. Additionally, while existing SGFT methods prioritise individual fairness, group fairness, especially subtle discrimination [42], is less studied. Novel testing strategies for group fairness are needed.

## VI. Conclusions

Ensuring the fairness of AI-based systems is of utmost importance to uphold justice and equity in society. As AI

continues to expand, the need for fairness testing remains a critical area of research. In this context, search-based testing methods will continue to have a significant role. This paper reviews contemporary fairness testing approaches utilising SBT to evaluate AI-based systems. Additionally, we have outlined potential avenues for future research, offering valuable insights for further exploration.

## VII. REFERENCES


[1] B. Johnson and J. Smith, "Towards Ethical Data-Driven Software: Filling the Gaps in Ethics Research Practice," *Proceedings - 2021 IEEE/ACM 2nd International Workshop on Ethics in Software Engineering Research and Practice, SEthics 2021*, pp. 18–25, 2021, doi: 10.1109/SEthics52569.2021.00011.

[2] S. Galhotra, Y. Brun, and A. Meliou, "Fairness testing: Testing software for discrimination," *Proceedings of the ACM SIGSOFT Symposium on the Foundations of Software Engineering*, vol. Part F1301, pp. 498–510, 2017, doi: 10.1145/3106237.3106277.

[3] Y. Brun and A. Meliou, "Software fairness," *ESEC/FSE 2018 - Proceedings of the 2018 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 754–759, 2018, doi: 10.1145/3236024.3264838.

[4] J. Angwin, J. Larson, S. Mattu, and L. Kirchner, "Machine Bias: There's software used across the country to predict future criminals. And it's biased against blacks.," *ProPublica*, 2016. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

[5] Md. A. Malek, "Criminal courts' artificial intelligence: the way it reinforces bias and discrimination," *AI and Ethics*, vol. 2, no. 1, pp. 233–245, Feb. 2022, doi: 10.1007/s43681-022-00137-9.

[6] J. Dastin, "Amazon scraps secret AI recruiting tool that showed bias against women," *Reuters*, Oct. 11, 2018. https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G (accessed May 16, 2022).

[7] C. Counts, "Minority homebuyers face widespread statistical lending discrimination," *Phys.org*, Nov. 15, 2018. https://phys.org/news/2018-11-minority-homebuyers-widespread-statistical-discrimination.html#google_vignette (accessed May 16, 2022).

[8] S. Verma and J. Rubin, "Fairness definitions explained," *Proceedings - International Conference on Software Engineering*, pp. 1–7, 2018, doi: 10.1145/3194770.3194776.

[9] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A Survey on Bias and Fairness in Machine Learning," *ACM Comput Surv*, vol. 54, no. 6, 2021, doi: 10.1145/3457607.

[10] B. Li *et al.*, "Trustworthy AI: From Principles to Practices," vol. 1, no. 1, 2021, [Online]. Available: http://arxiv.org/abs/2110.01167

[11] T. P. Quinn, M. Senadeera, S. Jacobs, S. Coghlan, and V. Le, "Trust and medical AI: the challenges we face and the expertise needed to overcome them," *J Am Med Inform Assoc*, vol. 28, no. 4, pp. 890–894, Apr. 2021, doi: 10.1093/jamia/ocaa268.

[12] H. Suresh and J. Guttag, "A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle," *ACM International Conference Proceeding Series*, 2021, doi: 10.1145/3465416.3483305.

[13] A. R. Patel, J. Chandrasekaran, Y. Lei, R. N. Kacker, and D. R. Kuhn, "A Combinatorial Approach to Fairness Testing of Machine Learning Models," 2022.

[14] Z. Chen, J. M. Zhang, M. Hort, F. Sarro, and M. Harman, "Fairness Testing: A Comprehensive Survey and Analysis of Trends," Jul. 2022, [Online]. Available: http://arxiv.org/abs/2207.10223

[15] J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine Learning Testing: Survey, Landscapes and Horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1–36, 2022, doi: 10.1109/TSE.2019.2962027.

[16] D. P. Morales, T. Kitamura, and S. Takada, "Coverage-Guided Fairness Testing," *Studies in Computational Intelligence*, vol. 985, pp. 183–199, 2021, doi: 10.1007/978-3-030-79474-3_13.

[17] Z. Zhao, T. Toda, and T. Kitamura, "Efficient Fairness Testing Through Hash-Based Sampling," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Science and Business Media Deutschland GmbH, 2022, pp. 35–50. doi: 10.1007/978-3-031-21251-2_3.

[18] P. McMinn, "Search-based software testing: Past, present and future," in *Proceedings - 4th IEEE International Conference on Software Testing, Verification, and Validation Workshops, ICSTW 2011*, 2011, pp. 153–163. doi: 10.1109/ICSTW.2011.100.

[19] A. Perera *et al.*, "Search-based fairness testing for regression-based machine learning systems," *Empir Softw Eng*, vol. 27, no. 3, 2022, doi: 10.1007/s10664-022-10116-7.

[20] J. Siebert *et al.*, "Construction of a quality model for machine learning systems," *Software Quality Journal*, pp. 1–29, 2021.

[21] S. Martínez-Fernández *et al.*, "Software Engineering for AI-Based Systems: A Survey," May 2021, doi: 10.1145/3487043.

[22] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel, "Fairness through awareness," *ITCS 2012 - Innovations in Theoretical Computer Science Conference*, pp. 214–226, 2012, doi: 10.1145/2090236.2090255.

[23] S. Calzavara, L. Cazzaro, C. Lucchese, and F. Marcuzzi, "Explainable Global Fairness Verification of Tree-Based Classifiers," Sep. 2022, [Online]. Available: http://arxiv.org/abs/2209.13179

[24] M. Ma *et al.*, "Enhanced Fairness Testing via Generating Effective Initial Individual Discriminatory Instances," *arXiv preprint arXiv:2209.08321*, 2022.

[25] M. Khari and P. Kumar, “An extensive evaluation of search-based software testing: a review,” *Soft Computing*, vol. 23, no. 6. Springer Verlag, pp. 1933–1946, Mar. 29, 2019. doi: 10.1007/s00500-017-2906-y.

[26] D. Moher, A. Liberati, J. Tetzlaff, D. G. Altman, and P. Group*, “Preferred reporting items for systematic reviews and meta-analyses: the PRISMA statement,” *Ann Intern Med*, vol. 151, no. 4, pp. 264–269, 2009.

[27] S. Udeshi, P. Arora, and S. Chattopadhyay, “Automated directed fairness testing,” *ASE 2018 - Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pp. 98–108, 2018, doi: 10.1145/3238147.3238165.

[28] S. Sano, T. Kitamura, and S. Takada, “An efficient discrimination discovery method for fairness testing,” in *Proceedings of the International Conference on Software Engineering and Knowledge Engineering, SEKE*, Knowledge Systems Institute Graduate School, 2022, pp. 200–205. doi: 10.18293/SEKE2022-064.

[29] A. Aggarwal, P. Lohia, S. Nagar, K. Dey, and D. Saha, “Black box fairness testing of machine learning models,” *ESEC/FSE 2019 - Proceedings of the 2019 27th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pp. 625–635, 2019, doi: 10.1145/3338906.3338937.

[30] G. Tao, W. Sun, T. Han, C. Fang, and X. Zhang, “RULER: discriminative and iterative adversarial training for deep neural network fairness,” in *ESEC/FSE 2022 - Proceedings of the 30th ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Association for Computing Machinery, Inc, Nov. 2022, pp. 1173–1184. doi: 10.1145/3540250.3549169.

[31] M. Fan, W. Wei, W. Jin, Z. Yang, and T. Liu, “Explanation-Guided Fairness Testing through Genetic Algorithm,” in *2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE)*, 2022, pp. 871–882. doi: https://doi.org/10.1145/3510003.3510137.

[32] W. Xie and P. Wu, “Fairness testing of machine learning models using deep reinforcement learning,” *Proceedings - 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2020*, pp. 121–128, 2020, doi: 10.1109/TrustCom50675.2020.00029.

[33] P. Zhang *et al.*, “White-box fairness testing through adversarial sampling,” *Proceedings - International Conference on Software Engineering*, pp. 949–960, 2020, doi: 10.1145/3377811.3380331.

[34] P. Zhang *et al.*, “Automatic Fairness Testing of Neural Classifiers through Adversarial Sampling,” *IEEE Transactions on Software Engineering*, vol. 5589, no. c, pp. 1–20, 2021, doi: 10.1109/TSE.2021.3101478.

[35] L. Zhang, Y. Zhang, and M. Zhang, “Efficient white-box fairness testing through gradient search,” *ISSTA 2021 - Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis*, pp. 103–114, 2021, doi: 10.1145/3460319.3464820.

[36] H. Zheng *et al.*, “NeuronFair: Interpretable White-Box Fairness Testing through Biased Neuron Identification,” in *Proceedings of The 44th International Conference on Software Engineering (ICSE 2022)*, Association for Computing Machinery, 2021, pp. 1519–1531.

[37] P. Zhang, J. Wang, J. Sun, and X. Wang, “Fairness Testing of Deep Image Classification with Adequacy Metrics,” no. July 2017, 2021, [Online]. Available: http://arxiv.org/abs/2111.08856

[38] Y. Xiao, A. Liu, T. Li, and X. Liu, “Latent Imitator: Generating Natural Individual Discriminatory Instances for Black-Box Fairness Testing,” in *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, New York, NY, USA: ACM, Jul. 2023, pp. 829–841. doi: 10.1145/3597926.3598099.

[39] G. Castiglione, G. Wu, C. Srinivasa, and S. Prince, “fAux: Testing Individual Fairness via Gradient Alignment,” Oct. 2022, [Online]. Available: http://arxiv.org/abs/2210.06288

[40] H. Guo, “Fairness Testing for Recommender Systems,” in *Proceedings of the 32nd ACM SIGSOFT International Symposium on Software Testing and Analysis*, 2023, pp. 1546–1548.

[41] H. Guo *et al.*, “FairRec: Fairness Testing for Deep Recommender Systems,” *arXiv preprint arXiv:2304.07030*, 2023.

[42] M. Zhang, J. Sun, J. Wang, and B. Sun, “TESTSGD: Interpretable Testing of Neural Networks Against Subtle Group Discrimination,” *ACM Transactions on Software Engineering and Methodology*, 2022.

[43] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau, “FairVis: Visual analytics for discovering intersectional bias in machine learning,” in *2019 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 2019, pp. 46–56.

[44] S. Liu and L. N. Vicente, “Accuracy and fairness trade-offs in machine learning: a stochastic multi-objective approach,” *Computational Management Science*, vol. 19, no. 3, pp. 513–537, 2022, doi: 10.1007/s10287-022-00425-z.

[45] A. S. Habib, S. U. R. Khan, and E. A. Felix, “A systematic review on search-based test suite reduction: State-of-the-art, taxonomy, and future directions,” *IET Software*, vol. 17, no. 2. John Wiley and Sons Inc, pp. 93–136, Apr. 01, 2023. doi: 10.1049/sfw2.12104.