



Search-based fairness testing for regression-based machine learning systems

Anjana Perera¹ · Aldeida Aleti¹ · Chakkrit Tantithamthavorn¹ · Jirayus Jiarapakdee¹ · Burak Turhan² · Lisa Kuhn³ · Katie Walker³

Accepted: 29 December 2021 / Published online: 30 March 2022
© The Author(s) 2022

Abstract

Context: Machine learning (ML) software systems are permeating many aspects of our life, such as healthcare, transportation, banking, and recruitment. These systems are trained with data that is often biased, resulting in biased behaviour. To address this issue, fairness testing approaches have been proposed to test ML systems for fairness, which predominantly focus on assessing classification-based ML systems. These methods are not applicable to regression-based systems, for example, they do not quantify the magnitude of the disparity in predicted outcomes, which we identify as important in the context of regression-based ML systems.

Method: We conduct this study as design science research. We identify the problem instance in the context of emergency department (ED) wait-time prediction. In this paper, we develop an effective and efficient fairness testing approach to evaluate the fairness of regression-based ML systems. We propose fairness degree, which is a new fairness measure for regression-based ML systems, and a novel search-based fairness testing (SBFT) approach for testing regression-based machine learning systems. We apply the proposed solutions to ED wait-time prediction software.

Results: We experimentally evaluate the effectiveness and efficiency of the proposed approach with ML systems trained on real observational data from the healthcare domain. We demonstrate that SBFT significantly outperforms existing fairness testing approaches, with up to 111% and 190% increase in effectiveness and efficiency of SBFT compared to the best performing existing approaches.

Conclusion: These findings indicate that our novel fairness measure and the new approach for fairness testing of regression-based ML systems can identify the degree of fairness in predictions, which can help software teams to make data-informed decisions about whether

Communicated by: Tim Menzies and Mei Nagappan

This article belongs to the Topical Collection: *Inventing the Next Generation of Software Analytics*

✉ Anjana Perera
Anjana.Perera@monash.edu

✉ Burak Turhan
burak.turhan@oulu.fi

Extended author information available on the last page of the article.

such software systems are ready to deploy. The scientific knowledge gained from our work can be phrased as a technological rule; to measure the fairness of the regression-based ML systems in the context of emergency department wait-time prediction use fairness degree and search-based techniques to approximate it.

Keywords Fairness testing · Software testing · Search-based software testing · Software fairness · Machine learning · Bias

1 Introduction

Machine learning (ML)-based systems are an important infrastructure that enables our society's digital transformation, replacing many dangerous, difficult and tedious tasks with automation, with the catch that the resultant decisions, if unchecked, may have a profoundly negative impact on individuals and organisations. Machine learning systems impact us in very deep and personal ways, and are permeating many aspects of life, including healthcare, education, banking, employment and transportation. Such software systems decide how we purchase online (Mattioli 2012), who gets a loan (Olson 2011), what kind of medical diagnosis and treatment we receive (Strickland 2016), and are incorporated into criminal justice systems to determine who goes to jail and who is set free (Ferral).

In recent years, concerns about biased machine learning systems (i.e., software systems that have a machine learning component, and which wrongfully discriminate against certain individuals or groups of individuals in favour of others Friedman and Nissenbaum 1996) have been growing. For example, facial recognition systems often perform poorly on female and African-American faces (Klare et al. 2012). YouTube's automatic captions for men's speech are more accurate than women's (Tatman 2017). Assistant robots in aged-care centres create anxiety in older people when robots work to keep older people happy with pleasant conversations by not telling them the truth (Sharkey 2020). Natural language processing tools for detecting hate speech online are biased against African-American people (Ghafary 2019). Amazon software decided not to offer same-day delivery to predominantly minority neighbourhoods (Ingold and Soper 2016a). Staples offered online discounts to customers only in wealthier and more affluent neighbourhoods (Hardwar 2012). These examples highlight the importance of the fairness measures in order to quantify the degree of bias in the machine learning systems and fairness testing approaches in order to ensure that machine learning systems do not exhibit biases in their predictions and recommendations.

Various fairness measures have been introduced. For example, conditional statistical parity (Corbett-Davies et al. 2017), equal opportunity (Hardt et al. 2016), counterfactual fairness (Chiappa 2019) and equalised odds (Hardt et al. 2016). However, these fairness measures are only applicable to classification-based ML systems (i.e., machine learning systems that use classification techniques)—*which are not purposely designed for regression-based ML systems* (i.e., machine learning systems that use regression techniques). In addition, different fairness measures always capture different aspects of bias. Thus, their suitability is still context and domain dependent (Mehrabi et al. 2019). For example, Dieterich et al. (Dieterich et al. 2016) found that COMPAS (i.e., an ML system used by some U.S.A. courts to predict the likelihood of a defendant becoming a recidivist) is not biased when using the disparate impact as the fairness measure, but later is considered unfair when evaluated using other measures (e.g., the equality of error rates (Angwin et al. 2016)).

Similarly, various fairness testing approaches have been introduced in the software engineering literature (Zhang et al. 2020). The goal of these fairness testing approaches

is to generate discriminatory inputs, which are the test cases that expose biases (i.e., individual discrimination, group discrimination, and causal discrimination) in the ML-based system under test (SUT). To generate the discriminatory inputs, black-box fairness testing approaches have been proposed, e.g., random test generation (Themis (Galhotra et al. 2017)), directed test generation (Aequitas (Udeshi et al. 2018)), symbolic generation (Aggarwal et al. 2019), as well as white-box fairness testing approaches (Zhang et al. 2020). However, these fairness testing approaches are primarily designed for classification-based ML systems, but not for regression-based ML systems.

Regression-based ML systems use regression techniques to predict continuous values (e.g., patient wait-time estimation). Since the nature of the predicted outcomes of regression-based ML systems is different from classification-based ML systems (i.e., continuous vs binary), existing fairness measures and fairness testing approaches may not be directly applicable, and also not yet effective and efficient. In particular, existing fairness measures for classification-based ML systems are indicative of the number of test cases that exhibit different outcomes (TRUE \rightarrow FALSE or FALSE \rightarrow TRUE) for two similar inputs except the difference of a sensitive attribute such as gender. However, for regression-based ML systems, the difference of the predicted outcomes is not binary, but continuous—which is challenging to determine if two predicted continuous outcomes are different or not to be considered as unfair. Thus, one common approach is to use a threshold to determine if a test input is considered discriminatory, i.e., the difference of two predicted outcomes is greater than the threshold (Udeshi et al. 2018). Nevertheless, existing fairness testing approaches that leverage these fairness measures only indicate the number of the discriminatory inputs, without quantifying the magnitude of the difference of the two predicted outcomes.

Recently, Berk et al. (2017) introduced a fairness measure for regression-based ML systems as the average differences of predicted outcomes for two similar inputs which differ in the value of the sensitive attribute, without considering the maximum difference of the two predicted outcomes. Thus, the Berk's fairness measure may underestimate the possible worst case of unfairness of the SUT. Therefore, the extremely worst cases may go unnoticed during testing if the existing fairness measures are used, making the users of the regression-based ML systems vulnerable to unfair predictions (e.g., female patients are estimated to wait longer than male patients based on the estimation of an emergency department wait-time prediction). To the best of our knowledge, none of the existing fairness measures and testing approaches can estimate the possible worst case of unfairness of the ML-based system under test.

In this paper, we propose (1) a new fairness measure called *fairness degree*, which is defined as the maximum difference in the predictions for two inputs that are identical except for a sensitive attribute (e.g., gender); and (2) a novel Search-Based Fairness Testing (SBFT) approach to estimate the fairness degree of the ML-based system under test. In contrast to existing fairness testing approaches, SBFT is the first search-based approach designed to test fairness in any machine learning system by employing a genetic algorithm. It has an efficient fitness evaluation procedure which uses an archiving approach for values that are more likely to lead to bias revealing inputs, and a fast local search procedure that improves the search for high quality test inputs that reveal biases. Then, we evaluate the effectiveness and efficiency of our SBFT approach in terms of discovering fairness degree and compare with four existing fairness testing approaches (i.e., Aequitas Udeshi et al. (Udeshi et al. 2018), Themis (Galhotra et al. 2017), symbolic generation (Aggarwal et al. 2019) and random testing).

Our study follows a design science research approach. Fairness metrics for regression-based ML systems are rare and the existing ones ignore to quantify the magnitude of outcome difference. We identify the real problem instance in the context of emergency

department (ED) wait-time prediction. We propose a novel fairness metric, i.e., fairness degree, to quantify the degree of bias in a regression-based ML system and a novel search-based fairness testing technique to estimate the fairness degree of an ML system. The proposed solutions are implemented and applied to ED wait-time prediction software. The scientific knowledge gained from our work can be phrased as a technological rule; to measure the fairness of the regression-based ML systems in the context of ED wait-time prediction use fairness degree and search-based techniques to approximate it.

Through an experimental evaluation on 12 emergency department wait-time prediction models based on regression techniques trained from over 1.3 million patient records from 12 hospitals (i.e., the largest emergency department patient datasets to date), we address the following two research questions:

RQ1 How effective is SBFT? SBFT is more effective than the baseline approaches, yielding a statistically higher fairness degree with large effect sizes at 10, 7, and 10 hospitals for the sensitive attributes *country of birth*, *Indigenous status*, and *gender*, respectively.

RQ2 How efficient is SBFT? SBFT is significantly more efficient than the baseline approaches with large effect sizes at 11, 5, and 6 hospitals for the sensitive attributes *country of birth*, *Indigenous status*, and *gender*, respectively.

These results confirm that our SBFT approach outperforms the state-of-the-art fairness testing approaches in terms of finding fairness degree of ML software systems based on regression techniques. Thus, we expect that our SBFT approach can help software teams to identify the degree of fairness in predictions and make data-informed decisions about whether such software systems are ready to deploy.

Novelty & Contributions The novelty and contributions of this paper are as follows:

1. We propose a fairness measure called fairness degree that describes the worst case behaviour of a regression-based ML system.
2. We propose a search-based fairness testing approach that effectively and efficiently estimates the fairness degree of a regression-based ML system.
3. The results of our experimental evaluation on 12 emergency department wait-time models demonstrate that SBFT is more effective and efficient in terms of finding fairness degree compared to the state-of-the-art fairness testing approaches.

Paper Organisation The organisation of this paper following Section 1, the introduction, is as follows. Section 2 provides the context of emergency department wait-time prediction software used in this study. Section 3 describes the design science research approach undertaken for this study. Section 4 formally defines the proposed fairness measure, i.e., fairness degree. Section 5 describes SBFT, our proposed approach to estimate the fairness degree of a regression-based ML system. Section 6 experimentally evaluates SBFT. Section 7 positions our study amongst related work. Section 8 outlines the threats to validity of our study. Section 9 concludes the paper.

2 Context

AI-enabled healthcare software is being adopted by the healthcare industry, aiming for improved efficiency, reduced costs and errors, and improvements to both patient satisfaction

and experience. It has been estimated that the market value of healthcare software is projected to reach around \$29.9 billion by 2023, with a compound annual growth rate (CAGR) of 7.4% from 2018 to 2023.

With the wave of the global COVID-19 pandemic, emergency departments (ED) in many countries are currently overwhelmed and overcrowded due to an unpredictable and unscheduled influx of patients.¹ Overcrowded EDs, usually caused by overcrowded hospitals, jeopardise patient safety and are associated with increased death rates (Di Somma et al. 2015). When EDs are overcrowded, this also cascades back to community health, taking ambulances off the road whilst they are unable to offload patients into ED beds. Patients have limited access to information about ED waiting time and hospital capacity. This lack of transparency from healthcare facilities has negative consequences on patient satisfaction and can make a difficult patient journey much harder for consumers of healthcare (Walker et al. 2020). Hospitals throughout the world have started to deploy AI-enabled healthcare software to support EDs in improving patient satisfaction, healthcare management, and clinical practices. Patient waiting time for healthcare services is identified as one of the key measurements of a responsive healthcare system (Sun et al. 2017), according to the World Health Organization (WHO), and a well-designed and resourced healthcare system should not have long queues for consultations and treatments. Wait times are often highly politicised, for both acute and elective care.

In this paper, we focus on a regression-based machine learning system, which is used in EDs for patient wait-time prediction (aka. ED Software). ED Software provides patients attending the ED at hospitals with an estimated time that they would have to wait before being seen by a doctor or other provider. ED Software is currently being used in hospitals across Melbourne, Australia to estimate the wait time for new patients seeking care (Walker et al. 2021).

Machine learning to predict ED patient wait-times is useful to keep patients informed about the expected wait time before being treated by a doctor (Shah et al. 2015). Such wait-time predictions can also be used to raise awareness of the current flow of waiting patients, improve capacity planning, and identify system bottle-necks (Strobel et al. 2021). The accuracy of the patient wait-time predictions is one of the key factors used in hospitals to better manage patients' expectations. An overestimated wait-time can cause a patient to decide to delay seeking critical treatment whereas an underestimated wait-time can negatively impact patient experience (Soremekun et al. 2011).

According to the WHO's global strategy on digital health 2020–2025, AI-enabled healthcare software that are used to support decision-makers must ensure the *ethical* use of technology. In addition, ED clinicians perceive the role of the ED to be a safety net for healthcare for all patients equally, treating anyone at anytime, without judgement. Thus, ED Software should not exhibit any discrimination bias in patient wait-time predictions.

Therefore, ED Software should produce similar wait-time estimations for patients who have the same clinical urgency, arriving at the same time with the same number of patients ahead of them in the queue. This should be the case regardless of individual sensitive attributes (e.g., gender, country of birth, indigenous status, race, religion). Discrimination bias embedded in the prediction models in the ED Software could have a negative impact on patient satisfaction, regardless of the number of patients affected. Patients and families cope better with expected waits than unexpected waits and are disappointed by longer than expected waits. Hospitals that seek to attract more patients (fee-for-service funding)

¹<https://www.japantimes.co.jp/news/2021/01/09/national/overloaded-hospitals-japan-coronavirus/>

may find customers choosing alternative facilities if their wait-time is overestimated and generate low net promoter scores and ill-will when the wait time is underestimated (also losing customers). Hospitals with block funding also suffer when wait-time predictions are inaccurate, due to reduced patient satisfaction and increased aggression and complaints. Therefore, the use of existing fairness measures (Berk et al. 2017) (e.g., the average differences of predicted outcomes for two similar inputs which differ in the value of the sensitive attribute) may hinder the extreme cases of unfairness, leading to a poor understanding of the ultimate worst case of unfairness of an ML-based system under test and sub-optimal decision-making if such unfair systems are deployed. Thus, a novel fairness degree to indicate the absolute worst case of unfairness by an ML-based system is critically needed.

3 Research Methodology

We conduct this study as design science research. Runeson et al. (2020) designed a visual abstract template to help identify the design science constructs in software engineering research. We use this template to communicate our research methodology as shown in Fig. 1. According to Runeson et al. (2020), there are three main constructs of design science research; i) technological rule, ii) its instantiation in terms of a real problem–solution pair, and iii) the empirical or theoretical support for problem conceptualisation and the solution design.

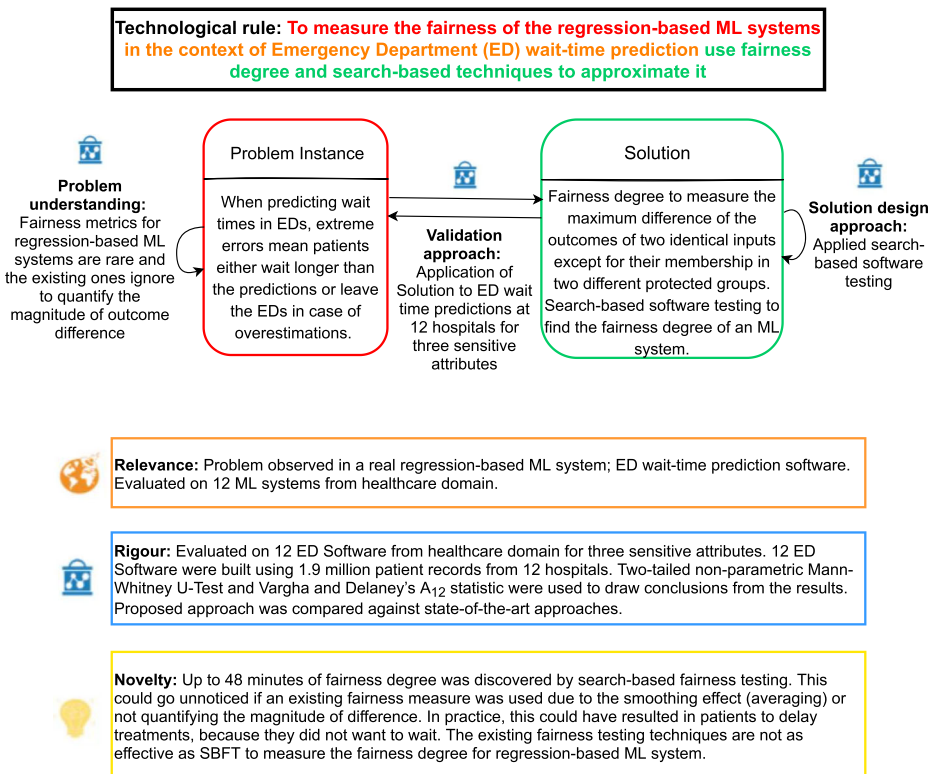


Fig. 1 Visual abstract for the paper

The scientific knowledge gained from our work can be phrased as a technological rule; to measure the fairness of the regression-based ML systems in the context of ED wait-time prediction use fairness degree and search-based techniques to approximate it.

We understand through the literature that fairness metrics for regression-based ML systems are rare and the existing ones ignore to quantify the magnitude of outcome difference. The real problem instance is identified in the context of emergency department wait time prediction, where extreme errors mean patients either wait longer than the predictions or leave the EDs in case of over-estimations. We propose a novel fairness metric called fairness degree to measure the maximum difference of the outcomes of two identical inputs except for their membership in two different protected groups (Section 4). To find the fairness degree of an ML system, we propose a novel search-based fairness testing technique (Section 5). The proposed solutions are implemented and applied to ED wait-time prediction software.

The relevance of the research, the rigour of the research activities, and the novelty of the technological rule in relation to the underpinning research will be discussed in the Conclusion (Section 9).

4 Fairness Degree for Regression-Based Machine Learning Systems

A regression-based machine learning system can be formally defined as:

Definition 1 A regression-based machine learning system is a software system that has a machine learning component where the explanatory variables (or instances) are denoted by $\mathbf{x} \in X$, the target variables (or labels) by $y \in Y = [-\infty, +\infty]$, and the predicted variables are denoted by $\hat{y} \in \hat{Y} = [-\infty, +\infty]$, where y and \hat{y} are continuous.

We consider the case where each instance \mathbf{x} contains a sensitive feature s and its value is denoted as x^s . An example of a sensitive feature is gender, and its value can be female, male, or non-binary.

In regression-based machine learning systems, the target and predicted values are continuous, and the same value is not likely to occur even twice in the data, which makes it hard to determine whether the system is fair or not.

Berk et al. (2017) introduced the first measure for fairness of the regression-based ML systems. The measure is called individual fairness, and estimates how similarly a model treats two similarly labelled instances, which differ in the value of the sensitive feature.

For every cross pair $(x_i, y_i) \in g_1$ and $(x_j, y_j) \in g_2$, where g_1 and g_2 are groups from the same population that are different in terms of the sensitive attribute s (e.g., women, men), *individual fairness* measures how differently the machine learning model treats x_i and x_j , weighted by a function of $|y_i - y_j|$.

$$f = \frac{1}{|g_1||g_2|} \sum_{(x_i, y_i) \in g_1, (x_j, y_j) \in g_2} d(y_i, y_j)(x_i - x_j)^2 \quad (1)$$

where d is a fixed non-negative function decreasing in $|y_i - y_j|$, which takes care of cancellation issues.

More specifically, individual fairness states that the fairness penalty for overestimating several of one group's labels cannot be mitigated by overestimating several of the other group's labels. However, this averaging over the instances has a smoothing effect, and hides the impact of outliers.

Previous research shows that assuring machine learning fairness depends more on how fairness is defined than on how it is implemented (Corbett-Davies and Goel 2018), which explains why the issue that has received the most attention in the ML fairness literature is the definition of fairness (Selbst et al. 2019). Indeed, several definitions of fairness have been introduced, such as conditional statistical parity (Corbett-Davies et al. 2017), equal opportunity (Hardt et al. 2016), counterfactual fairness (Chiappa 2019), equalised odds (Hardt et al. 2016), etc.

Feldman et al. (2015), formalise the Equal Employment Opportunity Commission's 80% rule into a formal measure of fairness called disparate impact, which is used in U.S. law to encode unintentional bias. Disparate impact measures if a decision making process has widely different outcomes for different groups, even as it appears to be neutral. In the well known debate about the COMPAS risk score, the creators argued that COMPAS was fair because the test accuracy was equalised across groups (Dieterich et al. 2016), which was estimated by the disparate impact measure. Instead, ProPublica journalists demonstrated that COMPAS was not fair because another important measure of fairness, known as equality of error rates was violated (Angwin et al. 2016).

This example highlights that existing fairness definitions are simplifications that cannot fully capture the range of existing and overlapping notions of fairness in all philosophical, legal, and sociological contexts. Hence, the suitability of a fairness measure is context dependent (Mehrabi et al. 2019) and must be carefully selected.

Without discounting the importance of avoiding a systemic bias on the whole population, outliers, where the predictions deviate by a large margin from reality, deserve special attention. In other words, minimising extreme deviations in predictions is as important as avoiding systemic bias.

In the context of ED Software, this could mean that the wait-time was grossly overestimated, so the patient may decide to delay treatment or go to another clinic, or underestimated, where the patient may decide to wait and not receive timely treatment (Strobel et al. 2021). Deciding to go to another clinic as a result of overestimation may have negative impacts if the selected clinic is not a specialist one. Not receiving timely treatment due to underestimation can also be detrimental to health. Existing fairness measures do not address this issue, hence we propose a new measure of fairness, which we call *fairness degree*, and a search-based fairness testing (SBFT) approach that serves for approximating the fairness degree of a machine learning system.

Fairness degree describes the worst-case behaviour of a regression-based machine learning system. It indicates that the machine learning system is more biased (i.e., less fair) if the fairness degree is higher. Formally,

Definition 2 Given a machine learning system, the **fairness degree** is measured by the maximum difference in the predicted values by the machine learning system for all pairs of instances (x_i, x_j) that are identical apart from the sensitive attribute, i.e., $x_i^s \neq x_j^s$.

$$D = \max_{\forall i, j} |y_i - y_j|; x_i^s \neq x_j^s \text{ and } x_i^k = x_j^k, \forall k \neq s \quad (2)$$

In the case of the ED Software, we would expect the machine learning model to predict similar wait-times for two patients who are identical apart from the sensitive attribute; e.g., it is reasonable to assume that a woman would not need to wait longer than a man to be seen by a doctor, given that all relevant circumstances (urgency, age, etc.) are the same. A larger value of D indicates that the ED Software is more biased. More specifically, this means that there are two patients who are identical except they are members of two different

groups (e.g., male and female), and one of them is overestimated a wait-time of D minutes compared to the other.

Identifying the fairness degree of a machine learning system can be computationally expensive for large input spaces. This calls for a more effective method for testing such systems, which provides motivation for our Search-Based Fairness Testing approach introduced in the next section.

5 Search-Based Fairness Testing

Search-Based Fairness Testing (SBFT) is the first approach for testing regression-based machine learning systems. The key research challenge SBFT addresses is the identification of test inputs that reveal large bias for regression-based machine learning systems in order to estimate the fairness degree of the system in an efficient way. The optimisation problem is formulated as:

$$\text{Maximise } |y_i - y_j| \tag{3}$$

$$s.t. \quad x_i^k = x_j^k, \forall k \neq s \tag{4}$$

$$x_i^s \neq x_j^s. \tag{5}$$

where x_i^s and x_j^s are the values of the sensitive attribute. Existing test generation techniques for fairness use no guidance to find test inputs that can expose the fairness degree of the machine learning system. For example, Aequitas (Udeshi et al. 2018) uses a predefined threshold to identify test inputs that are discriminatory, i.e., if a test input reveals a fairness degree above the threshold it is considered discriminatory. Aequitas considers all test inputs that are discriminatory as equally important, hence this approach is not suitable for finding inputs that maximise the fairness degree.

The proposed approach SBFT is presented in Algorithm 1. It is based on a genetic algorithm, and introduces new enhancements to make the search more efficient and effective in identifying test inputs that reveal the fairness degree. In the next subsections, first we introduce the solution representation, and discuss the fitness evaluation and caching of sensitive variables which helps speed up the evaluation of the fairness degree of a machine learning system (Section 5.1). Next, we discuss the search steps with introducing the genetic operators used to create the new offspring from parent solutions and select individuals to the next population (Section 5.2).

5.1 Solution Representation and Fitness Evaluation

In SBFT, a solution to the optimisation problem is a test input, which is defined as $x_i = \{x_i^1, x_i^2, x_i^s, \dots, x_i^N\}$, where each variable $x_i^k; k \in [1, N]$ can be an integer, categorical variable or real number. The variables representing the sensitive attributes are defined with the superscript letter s (i.e., x_i^s).

The fitness of a test input x_i is measured by the individual fairness degree, defined as

$$d_i = |y_i - y_j|; x_i^k = x_j^k, \forall k \neq s \text{ and } x_i^s \neq x_j^s \tag{6}$$

where y_i and y_j are the outputs of the inputs x_i and x_j , which are only different in terms of their sensitive attributes.

Algorithm 1 Search-based fairness testing (SBFT).

```

1: procedure SBFT
2:   Input:  $MLS, s, N$   $\triangleright$   $MLS$  is the machine learning system under test,  $s$  is the
   index of the sensitive attribute, and  $N$  is the total number of attributes.
3:   Input:  $p_c, r_i$   $\triangleright$   $p_c$  is probability of using cache, and  $r_i$  is the rate of random
   test insertion in every iteration.
4:   Initialise  $F : \{x_i\} \rightarrow \mathbb{R}^+$   $\triangleright$   $F$  is a map between test inputs and their fitness.
5:    $X \leftarrow \text{INITIALISE}(M)$   $\triangleright$   $M$  is the population size.
6:   for  $x_i \in X$  do
7:      $F(x_i) \leftarrow \text{EVALUATEFITNESS}(x_i, s)$ 
8:   while !terminationCriteria do
9:      $X' \leftarrow \text{GENERATEOFFSPRING}(X, M)$ 
10:     $X'' \leftarrow X \cup X'$ 
11:    for  $x_i \in X''$  do
12:       $F(x_i) \leftarrow \text{EVALUATEFITNESS}(x_i, s)$ 
13:     $X \leftarrow \text{SELECTION}(X'')$   $\triangleright$  Select using the elitism strategy.
14:     $x'_B \leftarrow \text{LOCALSEARCH}(x_B)$   $\triangleright$  Perform local search on the best solution
     $x_B \in X$ .
15:     $X \leftarrow X \cup \{x'_B\}$ 
16:    return  $\text{MAX}(F)$   $\triangleright$  Return the maximum fitness (fairness degree).
17: procedure EVALUATEFITNESS( $x_i, s$ )
18:   if  $\text{RANDOM}(0, 1) \leq p_c$  then
19:      $x'_i \leftarrow \{x_i^1, x'^s_c, \dots, x_i^N\}$   $\triangleright$   $x'^s_c$  is a cached value of the sensitive variable.
20:      $x''_i \leftarrow \{x_i^1, x''_c, \dots, x_i^N\}$   $\triangleright$   $x''_c$  is a cached value of the sensitive variable.
21:   else
22:      $x'_i \leftarrow \{x_i^1, x'^s_i, \dots, x_i^N\}$   $\triangleright$   $x'^s_i$  is a random value of the sensitive variable.
23:      $x''_i \leftarrow \{x_i^1, x''_i, \dots, x_i^N\}$   $\triangleright$   $x''_i$  is a random value of the sensitive variable.
24:    $f = |y'_i - y''_i|$ 
25:   if  $f > \text{MAX}(F)$  then
26:     Update  $x'^s_c$  and  $x''_c$ 
27:   return  $f$ 
28: procedure GENERATEOFFSPRING( $X, M$ )
29:    $X' \leftarrow \emptyset$ 
30:   for  $u \leftarrow 0; u < M/2; u++$  do
31:      $x_i, x_j \leftarrow \text{SELECTPARENTS}(X)$   $\triangleright$  Roulette wheel selection
32:      $x'_i, x'_j \leftarrow \text{CROSSOVER}(x_i, x_j)$   $\triangleright$  Uniform crossover
33:      $x''_i \leftarrow \text{MUTATE}(x'_i)$ 
34:      $x''_j \leftarrow \text{MUTATEMIDDLE}(x'_j, x_i, x_j)$ 
35:      $X' \leftarrow X' \cup \{x''_i, x''_j\}$ 
36:    $X_R \leftarrow r_i * M$  number of random test inputs
37:    $X' \leftarrow X' \cup X_R$ 
38:   return  $X'$ 
39: procedure LOCALSEARCH( $x_B$ )
40:   for  $u \leftarrow 1; u \leq N; u++$  do
41:     if  $u \neq s$  then
42:        $x'_B \leftarrow x_B$ 
43:        $x''_B \leftarrow x''_B + \delta$ 
44:        $F(x'_B) \leftarrow \text{EVALUATEFITNESS}(x'_B, s)$ 
45:       if  $F(x'_B) > F(x_B)$  then
46:          $x_B \leftarrow x'_B$ 
47:   return  $x_B$ 

```

To measure the individual fairness degree of a test input, SBFT changes the value of the variable representing the sensitive attribute, keeping everything else the same (for example, flipping the value of the variable representing gender from female to male). Not all sensitive attributes have only two values. For instance, country of birth can take 293 values in the ED software system. It would be inefficient to exhaustively sample every value of the sensitive attribute to evaluate the fitness like in Aequitas. Therefore, SBFT maintains a cache of two values of the sensitive attribute which expose the current fairness degree of the system. To evaluate the fitness of a test input, SBFT either uses the cache with a probability of p_c (lines 18-20) or chooses two random values for the sensitive attribute with a $(1 - p_c)$ probability (lines 21-23). If the fitness of the test input is greater than the current fairness degree, it updates the cache with the new values (lines 25-26).

5.2 The Search Steps and Genetic Operators

SBFT starts with an initial M number of random test inputs as the initial population (line 5). Solutions are then evaluated using the fitness function defined by procedure EVALUATE-FITNESS in line 17, and explained in Section 5.1. SBFT applies the crossover, mutation and selection operators to evolve the population of test cases until a termination criteria is met (lines 9 to 15). Test cases with the highest fitness, as defined in (6) are more likely to reproduce and survive to the next generation.

SBFT generates an offspring population (line 9) by calling the procedure GENERATEOFFSPRING in line 28. Parents are selected using the roulette wheel selection strategy (line 21) (McMinn 2004). This way test cases with higher fitness are more likely to be selected as parents, hence to reproduce. Next, the selected parents are crossed-over to produce two children with a certain probability p_{cr} (line 32).

Formally, given two parents

$$x_i = \{x_i^1, x_i^s, \dots, x_i^N\}$$

$$x_j = \{x_j^1, x_j^s, \dots, x_j^N\}$$

the uniform crossover operator may produce two children

$$x'_i = \{x_j^1, x_i^s, \dots, x_j^N\}$$

$$x'_j = \{x_i^1, x_j^s, \dots, x_i^N\}$$

where each variable is swapped with a probability of 0.5 except for the sensitive variables, which remain the same. The crossover operator helps the search focus on high-quality areas of the search, exploiting existing information in the current population.

SBFT uses a uniform mutation operator. It mutates each variable of one of the two children x'_i with a certain probability p_μ (line 33), by randomly choosing a new value for each variable in the range of all possible values. Given

$$x'_i = \{x_j^1, x_i^s, \dots, x_j^N\}$$

the uniform mutation operator creates a new solution

$$x''_i = \{x'_j^1, x_i^s, \dots, x'_j^N\}$$

This mutation operator helps the search algorithm explore new areas of the search space, thus preventing the search from becoming trapped in local optima.

Each variable of the second child x'_j is mutated with a probability p_μ by randomly choosing a new value in the range of the values of its parents (line 34). Given

$$x'_j = \{x_i^1, x_j^s, \dots, x_i^N\}$$

the mutated solution is

$$x''_j = \{x_i^1, x_j^s, \dots, x_i^N\}; x''_j^k \in [\min(x_i^k, x_j^k), \max(x_i^k, x_j^k)], \forall k \neq s$$

This mutation operator helps the search to further exploit the information of the selected parents, which are already one of the best solutions in the current population. Similar to the crossover operator, both mutation operators do not change the sensitive variables.

In the preliminary experiments, we find that the search is likely to get trapped in local optima, hence produce sub-optimal solutions. Therefore, SBFT generates a small number of random test inputs every iteration to include as a seed in the population (line 36). This is done with the focus on increasing the exploration capabilities of SBFT.

Once the offspring population is generated, SBFT merges it with the current population (line 10) and selects the best M test inputs to the next population (*elitism*) (line 13). Finally, SBFT performs a local search on the best solution of the selected population by calling the procedure `LOCALSEARCH` in line 39 to find a better solution in the neighbourhood of the current best solution (line 14). For each variable in the best solution (line 40), the local search changes the value of it by $\delta \in \{-1, 1\}$ (line 43), and updates the best solution if the fitness of the new solution is better than the current best solution (lines 44-46). The local search explores the neighbourhood of the best solution in a more systematic way than the crossover operator to find the local optimum. Thus, it adds another layer of exploitation of the best solutions for the search in SBFT.

6 Experimental Evaluation

We design a set of experiments to evaluate the effectiveness and efficiency of SBFT. Effectiveness is assessed based on the fairness degree exposed of the system under test (SUT) at the end of the search process, which is the maximum fitness as found during the search. The approach that discovers higher fairness degree of the SUT compared to other approaches is considered more effective. Efficiency, on the other hand is the rate at which the approaches discover fairness degree of the SUT. The baseline approaches are presented in Section 6.1, details of the ED Software, i.e., emergency department wait-time prediction models, used as the experimental subjects are given in Section 6.2, experimental settings are described in Section 6.3, and the results are presented in Section 6.4. In essence, with these experiments, we aim to answer the following research questions:

RQ1 How effective is SBFT? The effectiveness of SBFT in discovering fairness degree of the SUT is compared to the baseline approaches described in Section 6.1. All approaches are executed against 12 ED Software built for 12 hospitals (Section 6.2) for the same amount of execution time, and the final fitness, as measured by (6) is reported for each hospital and sensitive attribute. We perform statistical tests (as described in Section 6.3) to determine whether differences in performance are statistically significant.

RQ2 How efficient is SBFT? We measure efficiency using the Area Under Curve (AUC) of the fairness degree over the execution time for each testing approach. Figure 2 illustrates this.

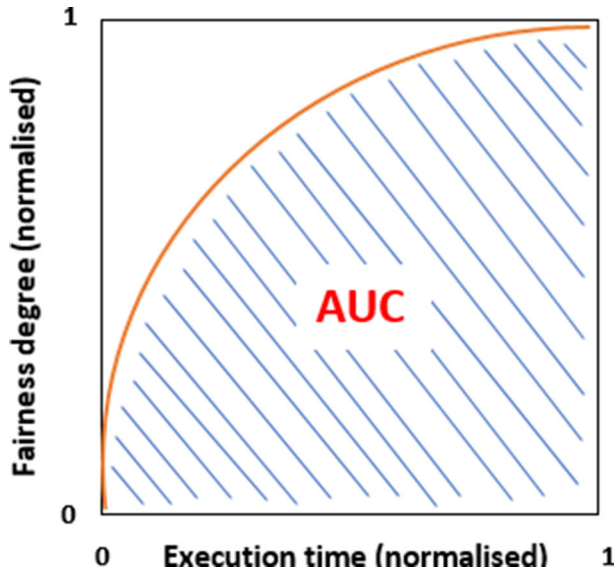


Fig. 2 Area Under the Curve (AUC)

A large AUC value indicates that the approach is fast at finding a higher fairness degree. Both the x (execution time) and y (fairness degree) axes of the curve are normalised in the range 0 to 1.

6.1 Baseline Approaches

We use the state-of-the-art Aequitas fully directed approach (Udeshi et al. 2018), Themis (Galhotra et al. 2017), symbolic generation (SG) (Aggarwal et al. 2019) and random testing as baselines for comparison.

We employ a threshold of 10 minutes for Aequitas, which is a parameter used to determine if a test input is discriminatory. The global search of Aequitas is run until a discriminatory input is found, then a local search is run on that input. These steps are repeated until the allocated execution time runs out.

In Themis, causal discrimination detection technique is used as it is appropriate to our context. Both Themis and SG are developed with the focus on classification-based machine learning systems. Therefore, we utilise a similar method as in Aequitas to determine if a test input is discriminatory, i.e., a threshold of 10 minutes.

Since the tool for symbolic generation approach is not publicly available, we re-use the tool developed by Zhang et al. (2020) for their experimental evaluation, which can be found at <https://github.com/pxzhang94/ADF>. The authors (Aggarwal et al. 2019) suggest to use a random seed in the symbolic generation approach in the absence of training data, which is the case in our ED models. Thus, we allocate the first 10% of the execution time, i.e., 12 minutes, to generate a random seed in every run of SG. In addition, we replace the decision tree classifier with a decision tree regressor to generate the decision tree as the ED models are regression-based ML systems. For all the other parameters, we use the same values as used in Zhang et al. (2020), which are from the best performance setting according to Aggarwal et al. (2019).

6.2 ED Patient Wait-Time Estimation Models

In our experiments, we use the twelve ED patient wait-time estimation models provided by Walker et al. (2021). The 12 ED models are built for 12 hospitals. Due to the highly-sensitive nature of the patient datasets, we do not have access to the datasets and the implementation source code. Instead, we only have access to the model objects as Python pickle files. Next, we describe the implementation details as provided by Walker et al. (2021).

A total of 1,930,609 patient records from 12 hospitals is collected from each hospital. The patient records for each hospital were sorted by their arrival time in chronological order. The 1,388,509 patient records from 2017-18 are used to build ED models using a random forest regression technique, while the 542,100 patient records in 2019 are used to evaluate the ED models using a time-wise hold-out validation approach. Table 1 presents a summary of the 13 variables based on the Victorian Minimum Emergency Dataset (VEMD) and 6 additional variables to approximate the resource and capacity in real-time (i.e., age, patients in triage queue, patients awaiting a provider, admitted patients awaiting departure, ambulance offload queue, and average wait-time of the last k patients). The 19 variables in Table 1 are considered as independent variables (\mathcal{X}), while the patient wait-time is considered as dependent variable (\mathcal{Y}). Due to the ethical concerns, we anonymise the hospital names by using H1 to H12 to denote the twelve hospitals in our study.

To avoid any administrative data entry errors, Walker et al. (2021) removed the wait time outliers from the datasets according to the following criteria: (1) the wait-time exceeding the maximum of 360 minutes; and (2) the wait-time exceeding the predefined statistical outlier threshold value (defined as 1.5 times the interquartile range ($IQR = Q3 - Q1$) over $Q3$) ($n = 13,612$ (0.7%)).

6.3 Experimental Settings

We consider three sensitive attributes: *Country of birth*, *Indigenous status*, and *Gender*. Country of birth can take 293 values (e.g., Australia), and Indigenous status (e.g., Aboriginal but not Torres Strait Islander origin) and gender (e.g., Male) can take 6 and 4 values respectively. We evaluate SBFT against the baselines for the three attributes separately. Each of the five approaches are given an execution time of 120 minutes. To account for the randomness of SBFT and the baseline approaches, we repeat the experiments 20 times. Then, we conduct non-parametric Mann-Whitney U-Test with the significance level (α) 0.05 (Arcuri and Briand 2014) to check for statistical significance of the differences. If p-value < 0.05 , then the differences are statistically significant.

In addition, we conduct Vargha and Delaney's \hat{A}_{12} statistical test (Vargha and Delaney 2000) to compute the effect size of such differences. The \hat{A}_{12} statistic indicates the probability of one algorithm producing a larger value than another algorithm. We consider there is a small effect size if $0.58 \leq \hat{A}_{12} < 0.65$, medium effect size if $0.65 \leq \hat{A}_{12} < 0.75$, and large effect size if $\hat{A}_{12} \geq 0.75$. If $\hat{A}_{12} = 0.50$, then the two algorithms are equivalent, and the effect size is negligible if $0.50 < \hat{A}_{12} < 0.58$ (Panichella et al. 2015).

SBFT has various parameters to be configured. We use *irace*, iterated racing for automatic algorithm configuration, to automatically tune the parameters of SBFT (López-Ibáñez et al. 2016). The following parameter settings produced the best results: population size $M = 100$, crossover probability $p_{cr} = 0.75$, mutation probability $p_{\mu} = 0.70$, rate of random test insertion $r_i = 0.10$, probability of using cache $p_c = 0.50$.

Table 1 A summary of the studied independent variables

Variables	Type of variables
Advanced care directive alert	Categorical
Arrival time	Date/Time (Integer)
Arrival transport mode	Categorical
Campus code	Categorical
Insurance status	Categorical
Preferred language	Categorical
Referred by	Categorical
Triage category (according to Australasian Triage Scale)	Categorical
Type of usual accommodation	Categorical
Type of visit	Categorical
Age (used instead of date of birth to preserve privacy)	Continuous (Integer)
Patients in triage queue	Continuous (Integer)
Patients awaiting a provider	Continuous (Integer)
Admitted patients awaiting departure	Continuous (Integer)
Ambulance offload queue	Continuous (Integer)
Average wait-time of the last k-patients	Continuous (Real)
Sensitive variables	
Country of birth	Categorical
Indigenous status	Categorical
Gender	Categorical

We implement SBFT, Aequitas, random testing and Themis in a prototype tool in order to experimentally evaluate them. The prototype tool is available at the online appendix, <https://github.com/search-based-fairness-testing>.

6.4 Results

6.4.1 RQ1: How effective is SBFT?

To address RQ1, we use the fairness degree measure (Definition 2) to quantify the effectiveness of SBFT when comparing to the baseline approaches, i.e., Aequitas, random testing, Themis and SG. We compute the fairness degree discovered by each studied approach given an execution time of 2 hours. Since the studied approaches are non-deterministic, we repeat the experiments 20 times prior to conducting statistical tests to determine whether such differences in results are statistically significant.

Table 2 shows a statistical summary and results of statistical tests of fairness degree produced by the studied approaches for the 12 hospitals and the three sensitive attributes. We observe that overall, SBFT is more effective than the baseline approaches, yielding a statistically higher fairness degree with large effect sizes at 10, 7, and 10 hospitals for *country of birth*, *Indigenous status*, and *gender*, respectively.

In particular, for *country of birth*, SBFT discovers that the fairness degree of the ED Software at hospital H5 is 44.94 minutes on average, which is 38.01 (+548.5%), 23.64

Table 2 (RQ1) A statistical summary and results of statistical tests of fairness degree produced by the studied approaches for the 12 hospitals and the three sensitive attributes

		SBFT vs.												
		Median (minutes)						Aequitas						
Site	Country of Birth	SBFT	Aequitas	Random	Themis	SG	Aequitas		Random		Themis		SG	
							p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}
H1		2.51	1.01	0.87	1.02	0.52	0.062	0.67	0.083	0.66	0.015	0.73	< 0.001	0.97
H2		27.53	15.74	19.49	14.87	5.88	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H3		10.52	8.28	8.80	8.18	3.71	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H4		2.63	2.02	2.15	1.98	1.52	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H5		44.81	20.64	27.86	23.16	2.28	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H6		13.26	8.21	7.96	7.98	5.29	< 0.001	0.98	< 0.001	0.93	< 0.001	0.93	< 0.001	0.99
H7		43.87	34.36	37.58	34.53	3.27	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H8		1.26	0.43	0.53	0.48	0.08	< 0.001	0.93	< 0.001	0.85	< 0.001	0.92	< 0.001	0.96
H9		0.37	0.04	0.19	0.04	<0.01	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H10		8.73	8.71	8.73	8.71	8.55	< 0.001	0.99	< 0.001	1.00	< 0.001	0.99	< 0.001	1.00
H11		3.98	3.35	3.28	3.16	<0.01	0.035	0.70	0.002	0.78	0.010	0.74	< 0.001	1.00
H12		1.85	1.32	1.29	1.21	0.67	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H1	Indigenous	<0.01	<0.01	<0.01	<0.01	<0.01	0.544	0.45	0.002	0.77	< 0.001	1.00	< 0.001	1.00
H2		0.18	0.17	0.17	0.17	0.11	0.031	0.70	< 0.001	0.86	0.051	0.68	< 0.001	1.00
H3		29.09	27.92	25.84	27.10	2.30	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H4		<0.01	<0.01	<0.01	<0.01	<0.01	0.195	0.61	< 0.001	0.94	< 0.001	1.00	< 0.001	1.00
H5		0.38	0.37	0.37	0.37	0.06	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H6		3.76	3.56	3.53	3.58	1.08	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H7		48.85	46.85	45.45	37.75	<0.01	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00
H8		2.01	2.01	1.99	2.01	<0.01	0.034	0.70	0.001	0.81	0.001	0.80	< 0.001	1.00
H9		16.92	16.92	16.92	16.92	<0.01	0.020	0.66	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00

Table 2 (continued)

Site	Median (minutes)						SBFT vs.							
	SBFT	Aequitas	Random	Themis	SG		Aequitas		Random		Themis		SG	
							p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}
H10	10.25	9.97	9.97	9.93	9.54		<0.001	0.99	<0.001	1.00	<0.001	1.00	<0.001	1.00
H11	3.07	2.75	2.64	2.51	<0.01		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H12	1.61	1.61	1.61	1.61	0.32		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
Gender	20.41	20.08	3.53	19.99	0.02		0.189	0.62	0.013	0.73	0.070	0.67	<0.001	1.00
H2	0.71	0.67	0.67	0.67	0.11		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H3	0.36	0.36	0.36	0.36	0.02		<0.001	0.88	<0.001	0.95	<0.001	1.00	<0.001	1.00
H4	0.16	0.13	0.13	0.13	0.05		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H5	2.72	2.30	2.13	2.38	0.19		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H6	5.11	5.11	5.11	5.11	0.01		<0.001	0.89	<0.001	0.99	<0.001	0.98	<0.001	1.00
H7	2.89	1.92	1.97	1.93	0.23		<0.001	0.98	<0.001	0.97	<0.001	0.95	<0.001	1.00
H8	0.51	0.51	0.51	0.51	0.01		0.036	0.69	0.004	0.76	0.009	0.74	<0.001	1.00
H9	0.74	0.74	0.74	0.74	0.01		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H10	1.20	0.96	0.94	0.96	0.03		<0.001	0.99	<0.001	1.00	<0.001	1.00	<0.001	1.00
H11	4.41	4.10	4.11	4.03	0.01		<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00
H12	0.40	0.40	0.40	0.40	0.07		<0.001	0.96	<0.001	1.00	<0.001	1.00	<0.001	1.00

The statistically significant values (p-value < 0.05) and large effect sizes ($\hat{A}_{12} \geq 0.75$ and $\hat{A}_{12} \leq 0.25$) are highlighted in bold

(+111.0%), 22.55 (+100.7%) and 15.46 (+52.4%) minutes higher than what symbolic generation, Aequitas, Themis and random testing found. SBFT finds the highest fairness degree of the ED Software at hospital H7, which is 48.85 minutes on average, for the sensitive attribute, *Indigenous status*. On the other hand, there is no hospital where SBFT performs worse than the baselines in terms of effectiveness for any sensitive attribute.

6.4.2 RQ2: How efficient is SBFT?

To address RQ2, we quantify the efficiency of each studied approach using the area under curve (AUC) of fairness degree that is discovered over time, as described under RQ2 in Section 6. Similar to RQ1, we allocate 2 hours of execution time for all of the studied approaches. We compute the fairness degree of each studied approach at every second. For each studied approach, we draw a line plot of fairness degree (y-axis) and execution time (x-axis), and compute the area under curve. A large AUC value indicates that an approach can discover large fairness degree early. Finally, we use Mann-Whitney U-Test and A_{12} statistic to statistically determine the most efficient studied approach.

Table 3 shows a statistical summary and results of statistical tests of AUC for the studied approaches for the 12 hospitals and the 3 sensitive attributes. We observe that overall, SBFT is more efficient than the baseline approaches, yielding statistically higher AUC values with large effect sizes at 11, 5, and 6 hospitals for *country of birth*, *Indigenous status*, and *gender*, respectively.

In particular, for *country of birth*, SBFT has an AUC value of 0.90 on average at hospital H5, which is 0.85 (+1700.0%), 0.59 (+190.3%), 0.55 (+157.1%) and 0.36 (+66.7%) higher than symbolic generation, Aequitas, Themis and Random Testing. In some cases, SBFT achieves AUC values closer to 1.0, e.g., at H3 and H7 for *Indigenous status*, suggesting that they converge to the final fairness degree of the ED Software very early in the search. We observe that SBFT discovers the highest fairness degree of 48.85 minutes at H7 for *indigenous status*. These findings show that SBFT can effectively find the largest fairness degree in a very efficient manner (i.e., discovered in the early stage of the search).

Symbolic generation (SG) has the worst performance compared to all the other approaches. In particular, it discovers the fairness degree of the ED Software only up to 7.91 minutes on average for any sensitive attribute at any hospital. We find that SG generates only 8 test inputs on average for an execution time of 2 hours across the three sensitive attributes and the 12 hospitals. This is significantly lower than the number of test inputs generated by the other four approaches, for example, SBFT and random testing generate around 14000 test inputs. The symbolic generation approach is significantly slowed down at its local explainer. SG uses LIME (Ribeiro et al. 2016) as the local explainer and LIME produces an execution path for each test input generated. To produce the path, LIME randomly samples a large number of inputs in the neighbourhood of the generated test input (i.e., 5000 inputs is the default value), and executes the ED Software for each of these inputs, which is computationally expensive.

Figure 3 shows the fairness degree improvements of the five approaches over execution time at hospital H5 for the sensitive attribute *country of birth*. We observe that SBFT finds better test cases compared to the baseline methods early in the search, and keeps this advantage throughout the execution time, with a steady improvement in solution quality. Aequitas, Themis and random testing, on the other hand, have long plateaus where fitness does not improve for many iterations and there are sudden large increments in the fairness degree. This behaviour is expected for Themis and random testing because both of them are based on random search techniques. The local search procedure in Aequitas considers all the

Table 3 A summary of the AUC values of fairness degree over execution time by the studied approaches for the 12 hospitals and the three sensitive attributes

		SBFT vs.												
		Median						SBFT vs.						
Site	Country of Birth	Acquitas			Random			Themis			SG			
		SBFT	Acquitas	Random	Themis	SG	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value
H1	H1	0.48	0.31	0.32	0.27	0.07	0.72	0.204	0.62	0.003	0.77	<0.001	1.00	<0.001
H2	H2	0.91	0.42	0.63	0.35	0.12	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H3	H3	0.88	0.59	0.73	0.56	0.17	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H4	H4	0.94	0.61	0.74	0.62	0.28	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H5	H5	0.91	0.26	0.54	0.37	0.02	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H6	H6	0.99	0.50	0.61	0.51	0.09	0.95	<0.001	0.96	<0.001	0.97	<0.001	1.00	<0.001
H7	H7	0.98	0.54	0.78	0.52	0.02	0.98	<0.001	0.92	<0.001	0.98	<0.001	1.00	<0.001
H8	H8	0.81	0.31	0.38	0.30	0.03	0.85	0.005	0.76	<0.001	0.93	<0.001	0.98	<0.001
H9	H9	0.93	0.09	0.27	0.08	0.01	0.99	<0.001	0.99	<0.001	1.00	<0.001	1.00	<0.001
H10	H10	1.00	0.97	1.00	0.96	0.35	1.00	0.001	0.82	<0.001	1.00	<0.001	1.00	<0.001
H11	H11	0.86	0.53	0.64	0.52	<0.01	0.97	<0.001	0.95	<0.001	0.97	<0.001	1.00	<0.001
H12	H12	0.93	0.58	0.57	0.54	0.13	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H1	Indigenous	0.76	0.77	0.66	0.40	0.13	0.41	<0.001	0.88	<0.001	1.00	<0.001	1.00	<0.001
H2	H2	0.93	0.90	0.89	0.90	0.27	0.63	0.006	0.76	0.144	0.64	<0.001	1.00	<0.001
H3	H3	0.97	0.89	0.80	0.88	<0.01	1.00	<0.001	1.00	<0.001	0.98	<0.001	1.00	<0.001
H4	H4	1.00	1.00	1.00	1.00	<0.01	0.51	0.482	0.57	0.387	0.58	<0.001	1.00	<0.001
H5	H5	0.98	0.97	0.97	0.97	0.04	0.79	0.001	0.80	0.009	0.74	<0.001	1.00	<0.001
H6	H6	0.97	0.89	0.87	0.89	0.10	0.98	<0.001	0.97	<0.001	0.95	<0.001	1.00	<0.001
H7	H7	0.99	0.90	0.86	0.68	<0.01	1.00	<0.001	1.00	<0.001	1.00	<0.001	1.00	<0.001
H8	H8	0.97	0.99	0.98	0.98	<0.01	0.433	0.935	0.49	0.626	0.55	<0.001	1.00	<0.001
H9	H9	1.00	1.00	1.00	1.00	<0.01	0.007	<0.001	0.85	0.009	0.74	<0.001	1.00	<0.001

Table 3 (continued)

		SBFT vs.																			
		Median				Aequitas				Random				Themis				SG			
Site	SBFT	Aequitas	Random	Themis	SG	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value	\hat{A}_{12}	p-value
H10	0.99	0.96	0.96	0.96	0.35	1.00	< 0.001	1.00	< 0.001	0.97	< 0.001	0.97	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001
H11	0.92	0.70	0.74	0.72	<0.01	0.97	< 0.001	0.93	< 0.001	0.93	< 0.001	0.93	< 0.001	0.99	< 0.001	0.99	< 0.001	1.00	< 0.001	1.00	< 0.001
H12	1.00	0.99	0.99	0.99	0.09	0.91	< 0.001	0.68	0.048	0.68	0.030	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001
Gender	H1	0.15	0.43	0.24	<0.01	0.41	0.330	0.41	0.482	0.57	0.871	0.49	< 0.001	0.49	< 0.001	0.49	< 0.001	0.49	< 0.001	0.49	< 0.001
	H2	0.91	0.83	0.80	0.05	0.87	< 0.001	0.90	< 0.001	0.90	< 0.001	0.96	< 0.001	0.96	< 0.001	0.96	< 0.001	1.00	< 0.001	1.00	< 0.001
	H3	1.00	1.00	1.00	0.02	0.52	0.829	0.52	0.062	0.67	0.766	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001
	H4	0.92	0.83	0.83	0.12	< 0.001	< 0.001	0.87	< 0.001	0.90	< 0.001	0.91	< 0.001	0.91	< 0.001	0.91	< 0.001	1.00	< 0.001	1.00	< 0.001
	H5	0.87	0.76	0.73	0.04	< 0.001	< 0.001	0.89	< 0.001	0.97	< 0.001	0.94	< 0.001	0.94	< 0.001	0.94	< 0.001	1.00	< 0.001	1.00	< 0.001
	H6	1.00	1.00	1.00	<0.01	0.47	0.705	0.47	0.005	0.76	0.766	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001	0.47	< 0.001
	H7	0.75	0.52	0.56	0.02	< 0.001	< 0.001	0.94	< 0.001	0.89	< 0.001	0.92	< 0.001	0.92	< 0.001	0.92	< 0.001	1.00	< 0.001	1.00	< 0.001
	H8	0.96	0.95	0.96	0.01	0.49	0.914	0.49	0.194	0.62	0.957	0.50	< 0.001	0.50	< 0.001	0.50	< 0.001	0.50	< 0.001	0.50	< 0.001
	H9	0.94	0.86	0.85	0.01	0.005	< 0.001	0.76	< 0.001	0.83	0.035	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001	0.70	< 0.001
	H10	0.91	0.69	0.73	0.02	< 0.001	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001
	H11	0.96	0.89	0.88	<0.01	< 0.001	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001	1.00	< 0.001
	H12	0.85	0.83	0.78	0.06	0.66	0.079	0.66	0.002	0.79	0.001	0.82	< 0.001	0.82	< 0.001	0.82	< 0.001	0.82	< 0.001	0.82	< 0.001

The statistically significant values (p-value < 0.05) and large effect sizes ($\hat{A}_{12} \geq 0.75$ and $\hat{A}_{12} \leq 0.25$) are highlighted in bold

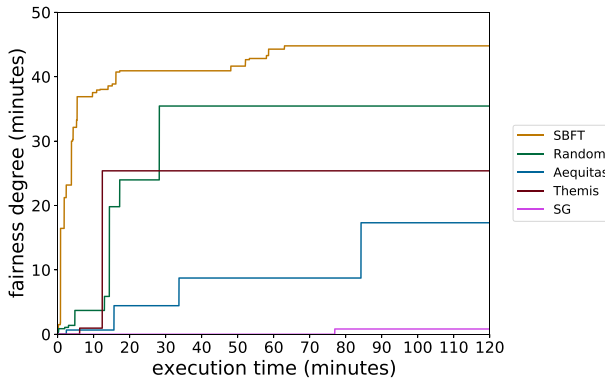


Fig. 3 Fairness degree improvement over execution time at hospital H5 for *country of birth*

test inputs with individual fairness degree above the threshold, i.e., 10 minutes, as equally important. It searches the neighbourhood of test inputs that have individual fairness degree of more than the threshold in the hope of finding more similar test inputs in terms of their fairness degree. This results in the local search discovering new test inputs that have a similar individual fairness degree to the ones found in the global search, which is responsible for exploration.

We observe that SBFT and random testing generate 14,455 and 14,341 test inputs on average for an execution time of 2 hours for the sensitive attribute *country of birth*. On the other hand, Aequitas only generates 100 test inputs on average. Such significant differences in the number of generated test inputs is due to the exhaustive nature of the sampling process of Aequitas. The fitness evaluation in Aequitas is slower than SBFT and random testing because of its exhaustive nature. Aequitas samples every possible value of the sensitive variable to evaluate the fitness of a test input. Thus, when a sensitive attribute has many possible values, e.g., *country of birth* which has a total of 293 possible values, Aequitas is much slower than SBFT and random testing. We observe the same problem in the fitness evaluation of Themis, which only generates 96 test inputs on average for *country of birth*. The fitness evaluation in Themis is similar to Aequitas such that Themis samples every possible value of the sensitive attribute until the input is determined as discriminatory, i.e., difference of predicted outputs is greater than the threshold.

Further analysis into the results show that SBFT is not only better at finding a higher fairness degree of ED Software, but also at generating test cases with higher individual fairness degree as measured by (6). Figures 4, 5 and 6 show the distributions of the individual fairness degree of the test cases generated by the studied approaches for each hospital as violin plots for the sensitive attributes *country of birth*, *Indigenous status* and *gender*, respectively. We exclude the test cases with individual fairness degree less than one minute in the analysis. According to the Mann-Whitney U-Test, SBFT generates test cases with significantly higher individual fairness degree compared to the baseline approaches at 7, 6 and 3 hospitals for *country of birth*, *Indigenous status* and *gender*, respectively. Descriptive statistics are provided in the online appendix, <https://github.com/search-based-fairness-testing>. We can also clearly see in the violin plots that SBFT has a higher third quartile and there are more test cases above the third quartile at hospitals H2, H3, H5 and H7 for *country of birth*, H3 and H7 for *Indigenous status*, and H11 for *gender*. This suggests that the test cases generated by SBFT have higher individual fairness degree than the ones generated by the baseline approaches.

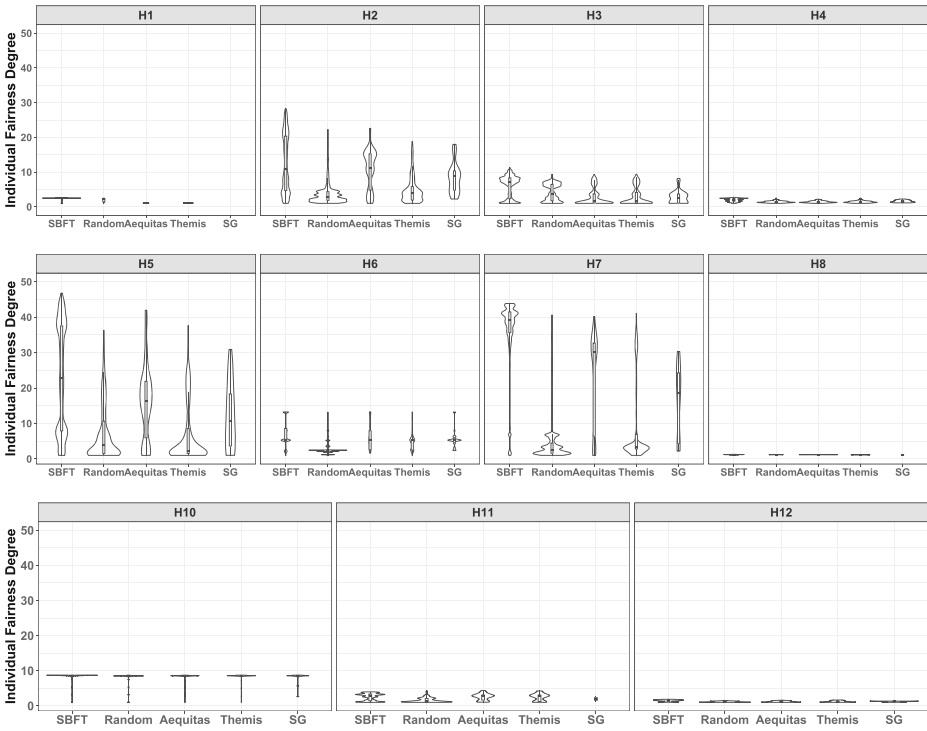


Fig. 4 Individual fairness degree distributions as discovered by each approach for *country of birth*

6.5 Discussion

Healthcare delivery is biased in favour of certain population groups. These favoured groups will experience better outcomes for the same diseases, when compared to other populations

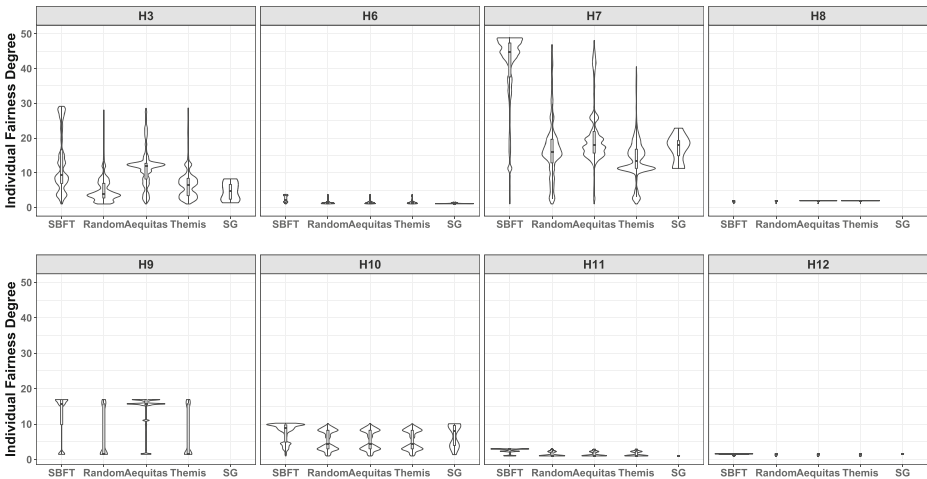


Fig. 5 Individual fairness degree distributions as discovered by each approach for *Indigenous status*

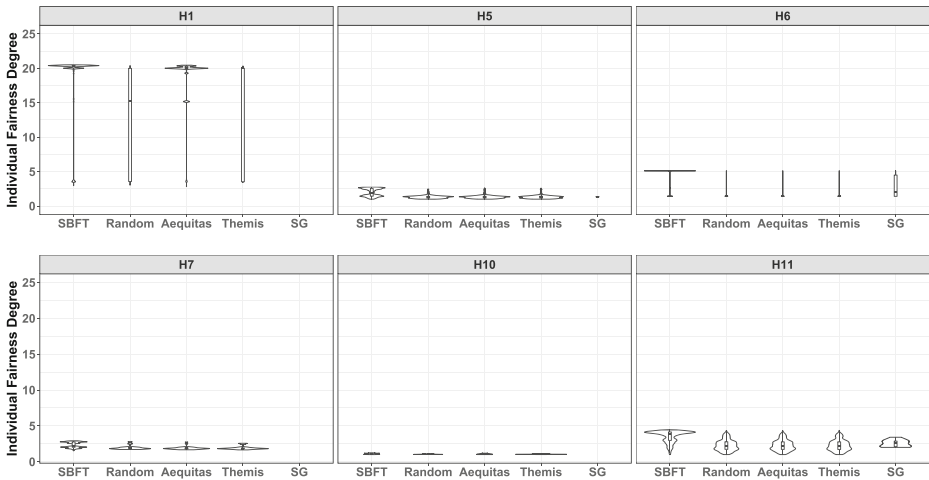


Fig. 6 Individual fairness degree distributions as discovered by each approach for *gender*

(higher survival rates and less long-term disability). The reasons for this are many, and bias exists at decision points the whole way along some patients' journeys through illness, from recognising the first symptoms to final long-term treatment. This creates biased derivation health datasets. Machine learning algorithms may well learn these biases and produce biased outputs when applied to health questions or scenarios. There is a risk of not only continuing to propagate these endemic biases, but also amplifying them as machine learning algorithms are incorporated into healthcare decision making.

In health, a woman with heart attack symptoms might not be advised to seek immediate medical care (Bairey Merz et al. 2017) whereas a man might be encouraged to call an ambulance and attend a heart hospital for immediate care (McSweeney et al. 2016; Udell et al. 2018). In a time-critical condition such as a heart attack, this will increase the woman's risk of dying or having a larger area of damage to her heart compared to the outcome had she received aggressive, immediate treatment (Mehta et al. 2016; Stehli et al. 2021).

Favoured groups in healthcare tend to be middle-aged men, who speak the main language of their country, are of the majority racial group, and have high affluence and health literacy (Juergens et al. 2016; Wechkunanukul et al. 2016). Groups who fare less well include women, older people, those with cultural and linguistic diversity compared to the majority of the country, and those with low affluence and low health literacy (Vogel et al. 2021).

In order to understand if machine learning is generating biased outputs prior to deploying them in the real world, it is important to be able to detect the biases. Once the biases are detectable, machine learning could have the potential to mitigate against biases rather than amplifying the inequities.

Fairness is recognised as a critical non-functional attribute of machine learning systems (Horkoff 2019). Recent work has focused on improving ML fairness, finding that removing sensitive features such as gender and ethnicity is not sufficient to ensure fair outcomes (Kamishima et al. 2011), which highlights the importance of fairness testing approaches. Our novel approach for fairness testing in regression-based machine learning systems can help software teams to identify the degree of fairness in predictions and make data-informed decisions about whether such software systems are ready to deploy.

In this study, we demonstrate the use of fairness degree to measure the biases in regression-based ML systems from the healthcare domain, i.e., emergency department wait-time prediction. We can draw some parallels between the new measure of fairness degree and existing work on worst-case analysis pertaining to quality attributes such as reliability (Bishop and Bloomfield 2002) and performance (Puschner and Burns 2000; Cortellessa et al. 2005; Ramamoorthy and Ho 1980). For example, Cortellessa et al. (2005) devise a methodology for estimating performance-based risk factors for software systems, which originate from violations of performance requirements, i.e., performance failures. The work introduces annotated UML diagrams to estimate the performance failure probability, which is combined with the failure severity estimate, thus enabling the determination of risky scenarios. The bias degree has a similar goal as it aims at identifying the most risky scenarios with respect to fairness.

Puschner and Burns (Puschner and Burns 2000) present an overview of worst case execution time analysis for safety critical real-time systems. Such analysis forms the basis for establishing confidence into the timely operation of a real-time system. When it comes to fairness, the fairness degree can be considered as a worst case analysis, as it aims at determining the highest bias that a system exhibits. Worst-case performance analysis is relevant specifically for safety critical real-time systems, such as robotics or cars (e.g., response time of the software component controlling the break pedal in a car), and may not be as relevant in software systems that do not need perform safety critical and real-time actions. Similarly, the fairness degree is an important metric that should be considered when developing software systems that are used in life-critical scenarios, such as the problem domain considered in this paper.

Fairness degree can also be used to measure the biases in other ML systems, such as the crime prediction algorithms, e.g., the risk of an individual re-offending (recidivism) (Angwin et al. 2016), personalised price prediction algorithms (Mahdawi 2018), patient risk-score predicting algorithms (Ledford 2019), and face detection algorithms, e.g., to crop a human face in an image (Hern 2020). For example, face detection algorithms detect the specific position of human faces in an image using a bounding box, i.e., drawing a rectangular box surrounding the human face in the image. The output is a bounding box marked by the (x, y) coordinates of the centre of the box along with its height and width. Given a face detection algorithm, it is reasonable to assume that it outputs the same bounding box ((x, y) , height and width) for two exactly identical faces except for a sensitive attribute like the skin colour (e.g., light skin and dark skin), if not that means the face detection algorithm is biased against individuals based on their skin colour. The concept of fairness degree can be leveraged to measure the maximum difference of (x, y) coordinates of the detected boxes (also height and width) for two identical faces which are different only from a sensitive attribute, like skin colour.

7 Related Work

7.1 Fairness of Machine Learning Systems

“Humans are inscrutable in a way that algorithms are not” (Mullainathan 2019). As software is created by people, it is inevitable that these biases will end up in code, resulting in biased software. Indeed, it is becoming increasingly evident that machine learning systems are vulnerable to bias, which render their decisions “unfair”. In the context of decision-making, fairness is the absence of any favouritism toward an individual or a group based

on their inherent or acquired characteristics (Mehrabi et al. 2019). In other words, an unfair machine learning system is one whose decisions are skewed toward a particular group of people (Binns 2018).

Unlike humans, however, software can be tested, creating the potential for new forms of transparency and hence opportunities to detect biases that are otherwise unavailable (Galhotra et al. 2017). For example, in 2016, Amazon.com, Inc. used software to determine which parts of the United States it would offer free same-day delivery. The decisions made by the software prevented minority neighbourhoods from benefiting from this offer, often when every surrounding neighbourhood could participate (Ingold and Soper 2016b).

7.2 Bias Detection and Mitigation Techniques

The machine learning community has been working on developing techniques for bias detection and mitigation for ML-based systems (Kamiran and Calders 2012; Calmon et al. 2017; Biswas and Rajan 2020). For example, IBM AI Fairness 360 toolkit (Bellamy et al. 2019) provides state-of-the-art bias detection techniques with 71 fairness metrics, such as disparate impact, which is the ratio between the probability of unprivileged group getting a favourable prediction and the probability of privileged group getting a favourable prediction. Similar to IBM AI Fairness 360, fairkit-learn (Johnson et al. 2020) is a toolkit that helps practitioners to reason about and understand fairness. In addition, it can also evaluate multiple models by computing the optimal trade-offs between fairness and performance of models. However, such toolkits only detect bias using the existing datasets, but are not able to detect bias by discovering discriminatory inputs via exploring the search space of all possible inputs.

Bias mitigation algorithms can be classified into *pre-processing* (fix the data), *in-processing* (fix the classifier), and *post-processing* (fix the predictions) techniques. *Pre-processing techniques* do not change the model, but only work on the dataset before training so that models can produce fairer predictions. For example, a reweighing technique (Kamiran and Calders 2012) and a disparate impact remover technique (Feldman et al. 2015). Zhang and Harman (2021) recommend that an enlarged feature set substantially improves both model accuracy and fairness, while extending the training dataset when the feature set is small could result in more unfairness by the trained model. *In-processing techniques* modify the ML model to mitigate the bias in the original model prediction. For example, an adversarial debiasing technique (Zhang et al. 2018). *Post-processing techniques* aim to modify the prediction results instead of the ML models or the input data. Fairway (Chakraborty et al. 2020) is an example of a technique that combines *pre-processing* and *in-processing* bias mitigation techniques. Although various techniques have been proposed in the machine learning community, such techniques are not designed for bias discovery in ML models.

7.3 Fairness Testing

Fairness testing of ML-based systems is still at an early stage. The existing fairness testing techniques are primarily focused on classification-based ML systems (Galhotra et al. 2017; Udeshi et al. 2018; Aggarwal et al. 2019; Zhang et al. 2020). Among those techniques, only Aequitas (Udeshi et al. 2018) is designed to work with regression-based ML systems. In addition, Galhotra et al. (2017) suggest ways of extending Themis to work with continuous variables as outputs, however, both papers limit their experimental evaluations only to systems with binary outputs.

Udeshi et al. (2018) proposed Aequitas, a directed test generation technique focusing on the individual fairness of the SUT. Aequitas starts with a global search and then performs a local search on the discriminatory inputs identified in the global search to find further discriminatory inputs. Their approach considers a discrimination threshold for the difference of two outputs to determine discrimination, thereby making it compatible with regression-based ML systems. Aequitas directs the local search only by considering whether the recent inputs are discriminatory or not, and it considers all the inputs that are discriminatory as equally important. This is opposite to the concept of fairness degree, which considers the extreme deviations in predictions are important.

Aequitas exploits the inherent robustness property of machine learning models, i.e., the output of the model should have low variation for small perturbations in the input (Udeshi et al. 2018). While this works for its intended task, i.e., maximising the number of discriminatory inputs found, we argue the Aequitas approach is ineffective in terms of discovering fairness degree. The local search in Aequitas does small perturbations on the inputs identified in the global search. According to the robustness property, the inputs identified in the local search should behave similarly to the ones identified in the global search, hence, the local search fails to make use of the inputs provided by the global search in order to maximise the observed fairness degree. In contrast, SBFT employs a genetic algorithm to incrementally maximise the fairness degree. In particular, SBFT evolves a population of test inputs through exploration and exploitation via mutation, crossover and a fast local search procedure.

Themis (Galhotra et al. 2017) is a random test generation technique that targets group and causal discrimination of the SUT. Its causal discrimination detection technique can be used to detect violations of individual fairness, hence, to find fairness degree. Unlike SBFT, Themis searches for test inputs randomly without any guidance to maximise the fairness degree of the SUT.

Aggarwal et al. (2019) propose a black-box testing technique called symbolic generation which focuses on individual fairness of classification-based ML systems. Their approach uses symbolic execution, which systematically generates test inputs, together with local explanation, which approximates the path in the model for a corresponding input. The symbolic generation approach is very slow at the local explainer. When LIME (Ribeiro et al. 2016) produces an execution path for a corresponding input in each iteration, it randomly samples a large number of inputs in the neighbourhood, e.g., 5000, and executes the model for each of these inputs. This is a time consuming task compared to the fast searching procedures used in SBFT which only executes the model to evaluate fairness degree. As a result, symbolic generation spends a large percentage of its allocated execution time to build decision tree paths, while SBFT utilises a large part of the execution time to explore the search space for high quality test inputs that reveal biases.

Zhang et al. (2020) proposed a white-box testing technique called adversarial discrimination finder (ADF) to detect individual discriminatory instances in classification-based ML systems. It was specifically designed for systems built with deep neural networks (DNNs). In contrast, SBFT is a black-box testing technique that does not require the internal details of the model under test, which enables testing of third party ML systems where the models are not accessible (Angwin et al. 2016). Unlike ADF, the concept of SBFT is not model dependent and can be applied to any regression-based ML system.

FairTest (Tramer et al. 2017) is a testing tool that detects statistically significant associations between sensitive attributes and an output of a model and generates an interpret-able bug report. These associations are called unwarranted associations and characterised as fairness bugs in the paper. FairTest needs an existing dataset as input to generate the bug report.

The datasets used to train and test ML models often contain highly sensitive information, e.g., in healthcare domain, and are rarely available to practitioners at the time of testing. Unlike FairTest, SBFT does not require ground-truth datasets, and it evolves an initial population of random test inputs to maximise the identified fairness degree of the system under test.

Unlike other testing approaches (Galhotra et al. 2017; Udeshi et al. 2018; Aggarwal et al. 2019; Zhang et al. 2020; Tramer et al. 2017), TILE (Sharma and Wehrheim 2019) tests fairness of ML algorithms at the learning stage. It applies four metamorphic transformations on training data; i) permutation of training data instances, ii) permutation of feature ordering, iii) shuffling of feature names and iv) renaming of feature values, and claims the algorithm is fair when the application of transformations results in equivalent predictors. In contrast to TILE, SBFT is intended to use at the prediction stage where the models are already built.

7.4 Situation Testing

The concept of fairness degree relates to situation testing. Situation testing is a systematic research procedure used in the legal field to analyse discriminatory treatment on an individual (Bendick 2007). In situation testing, pairs of individuals who are similar to each other except for their membership in two different protected groups are sent out to decision makers. Then, the decisions each pair receive are used to analyse the discriminatory behaviour. Fairness degree also checks the difference of outcomes for two individuals who are identical to each other except for their membership in two protected groups. In contrast to situation testing, fairness degree focuses on the maximum difference in the outcomes.

Luong et al. (2011) first used situation testing for discrimination discovery of a dataset in a classification setting. They used k-NN approach to find out similar individuals in a dataset. Zhang et al. (2016) followed a similar approach for discrimination discovery using Causal Bayesian Networks to find out similar individuals. Chakraborty et al. (2021) proposed to train a logistic regression model on the dataset and then flip the value of the sensitive attribute for every data point to find out if the outcome changes. The original data input was considered biased if the outcome was changed. As opposed to these works, SBFT does not require a dataset and explores the search space of all valid test inputs to find out the fairness degree of the ML system under test. In contrast to Luong et al. (2011) and Zhang et al. (2016), SBFT considers two individuals are similar only if they are identical except for a sensitive attribute. These discrimination discovery techniques based on situation testing are designed for classification-based ML systems (Luong et al. 2011; Zhang et al. 2016; Chakraborty et al. 2021). However, it is difficult to determine if two outcomes are different enough to be considered biased in a regression setting. We address this problem by introducing a novel fairness measure, fairness degree.

7.5 Fairness Measures for Machine Learning Systems

Underpinning the efforts for fair ML systems are an ever increasing array of fairness measures which aim to quantify fairness. Any measure, however, will undoubtedly have limitations. Indeed, the implication of “measurement” is problematic as it implies a straightforward process (Barocas et al. 2018).

Several studies have developed and applied fairness measures to evaluate ML systems. The most well known example is the report by ProPublica (Angwin et al. 2016) on the COMPAS algorithm. COMPAS (Dieterich et al. 2016) was used for recidivism prediction, and it was shown that it had higher false positive rates for African American defendants

than European American groups, which was interpreted as the tool was biased (Angwin et al. 2016). The false positive metric has also been used to measure the fairness of credit scoring algorithms (Hardt et al. 2016) and child welfare services (Chouldechova et al. 2018). Other studies have focused on Area Under the Curve (AUC) as a measure of fairness (Siegel 2003).

Aside from the philosophical and ethical debates on defining fairness, creating generalised definitions of fairness is difficult. Metrics usually either emphasise individual (e.g., everyone is treated equal), or group fairness. Existing fairness measures can be broadly categorised into group fairness and individual fairness (Caton and Haas 2020), with the following examples standing out in the literature.

- **Demographic parity** (Dwork et al. 2012) - group - the likelihood of a positive outcome should be the same independent of the value of the protected attribute.
- **Conditional statistical parity** (Corbett-Davies et al. 2017) - group - people in both protected and unprotected groups should have equal probability of being assigned to a positive outcome given a set of legitimate factors.
- **Equalised odds** (Hardt et al. 2016) - group - the probability of being correctly assigned a positive outcome and the probability of incorrectly being assigned a positive outcome should be the same independent of the value of the protected attribute.
- **Equal opportunity** (Hardt et al. 2016) - group - the protected and unprotected groups should have equal true positive rates.
- **Fairness through unawareness** (Grgic-Hlaca et al. 2016) - individual - an algorithm is fair as long as the sensitive attributes are not explicitly used in the decision-making process.
- **Fairness through awareness** (Dwork et al. 2012) - individual - any two individuals who are similar should receive a similar outcome.
- **Counterfactual fairness** (Chiappa 2019) - individual - a decision is fair towards an individual if it is the same in both the actual world and a counterfactual world where the individual belonged to a different demographic group.

All of these measures were developed and applied to classification-based machine learning systems, and cannot be directly applied to regression-based machine learning systems.

7.6 Search-Based Software Testing

Search-based software testing techniques have been applied to a plethora of testing problems such as unit testing (e.g., EvoSuite (Fraser and Arcuri 2013), AUSTIN Lakhota et al. 2013), end-to-end testing of Android apps (e.g., Sapienz (Mao et al. 2016)), functional testing (Wegener and Bühler 2004), security testing (Del Grosso et al. 2005), etc. EvoSuite is a state-of-the-art unit test generation tool which generates JUnit test suites for Java programs to optimise code coverage by employing metaheuristic search methods like genetic algorithms (Panichella et al. 2015). EvoSuite has been shown to be effective at not only achieving high code coverage (Panichella et al. 2018), but also finding bugs (Perera et al. 2020). Sapienz is an automated Android testing tool that optimises test sequences, maximising coverage and fault revelation, which has been successfully deployed in production at Facebook (Alshahwan et al. 2018). Prior to our work, search-based methods have not been applied in the area of fairness testing. SBFT is the first approach that generates bias

revealing test inputs to optimise the observed fairness degree of a machine learning system by employing a search-based technique.

8 Threats to Validity

Construct Validity The new measure introduced in this paper tackles a particular context of regression-based machine learning systems. While we do not claim that the fairness degree covers all fairness concerns of such systems, a threat to the validity of the results is the appropriateness of inferences made on the basis of such measure.

The SBFT approach is an approximate method. It effectively and efficiently searches the search space of potential test inputs via exploration and exploitation techniques to find the fairness degree of the SUT. However, it does not provide a guarantee that the final identified fairness degree is the actual fairness degree, i.e., absolute worst case, of the SUT, which is always greater than or equal to the fairness degree discovered by SBFT. Our experimental evaluation on an ML system in the healthcare domain demonstrates that SBFT is more effective and efficient than the existing approaches at finding fairness degree.

Another construct threat to validity relates to the definition of our fairness degree. Our proposed fairness degree measures the maximum difference in the predicted values by the machine learning system, while the existing fairness measure by Berk et al. (2017) considers the average differences. However, these two measures have their own advantages and disadvantages. The advantage of the maximum difference is to highlight the ultimate worst case of the unfairness of an ML-based system under test, but the maximum difference may be biased due to the outlier data. However this threat is mitigated in our experimental evaluation since the outliers in the patient datasets were removed prior to building the ED wait-time models (as mentioned in Section 6.2). On the other hand, while the average difference (Berk et al. 2017) is not sensitive to the outliers, it does not reveal what is the worst case of the unfairness of the ML-based system under test, which is the primary focus of our paper.

Internal Validity To account for the non-deterministic behaviour of the five algorithms, we repeat the experiments 20 runs and carry out rigorous statistical analysis, i.e., two-tailed non-parametric Mann-Whitney U-Test (Arcuri and Briand 2014) and Vargha and Delaney's \hat{A}_{12} statistic (Vargha and Delaney 2000), to draw conclusions from the results.

All algorithms are implemented in the same tool. The baseline approaches, Aequitas, Themis and symbolic generation, are implemented as described in the original papers (Galhotra et al. 2017; Udeshi et al. 2018; Aggarwal et al. 2019). Thus, any confounding effects due to different implementations or use of tools are mitigated in our experimental evaluation.

External Validity We evaluate SBFT using 12 ED wait-time prediction models for 12 hospitals with three different sensitive attributes. While the results cannot be generalised to all regression-based machine learning systems, the concept behind our proposed SBFT approach does not depend on the ED Software used in our experimental evaluation. Thus, future work can explore if our SBFT is effective and efficient in other regression-based machine learning systems.

SBFT supports three variable types for the test inputs: integer, real and categorical. While other input types, such as strings, images, sounds or videos are not directly handled in the current version of SBFT, the core concepts of our algorithm are independent of the input variable types. SBFT will be able to handle these types of variables by incorporating additional techniques to create valid inputs.

9 Conclusion

We present a new approach for evaluating the fairness of regression-based machine learning systems, which includes a new measure called *fairness degree* and a new automated testing approach for testing regression-based machine learning systems for fairness, called Search-Based Fairness Testing. Our approach is motivated by a machine learning system from the healthcare domain, which is used to predict wait times in an emergency department (ED Software). Existing fairness testing approaches are not designed for regression-based machine learning systems, and SBFT is the first approach that tackles this important problem. Such systems are being used in life critical situations such as the prediction of wait times in emergency departments. In our experimental evaluation, we observed differences of up to 48 minutes in the prediction of wait-times for patients that are identical apart from the sensitive attribute. Such errors in predictions may result in patients delaying critical treatment when the wait time is overestimated, hence it is important that we test such machine learning systems for the fairness degree, which analyses the worst case behaviour.

Our study follows a design science research approach. Runeson et al. (Runeson et al. 2020) proposed that the contributions of a design science research be assessed with respect to relevance, rigour and novelty. In regard to relevance of the research, we observe the problem in a real-world regression-based ML system; ED wait-time prediction software and evaluate the proposed solutions on 12 ML systems from healthcare domain. In Section 6.5, we discussed other regression-based ML systems that our proposed solutions can be applied to. In regard to the rigour of the research activities, we evaluate the proposed solutions on 12 ED Software from healthcare domain, which were built using nearly 1.9 million patient records from 12 hospitals. Each ED Software was evaluated for three sensitive attributes. The proposed SBFT technique was compared against four alternative solutions. We use two-tailed non-parametric Mann-Whitney U-Test and Vargha and Delaney's \hat{A}_{12} statistic to draw conclusions from the results. In regard to the novelty of the technological rule, the proposed SBFT technique discover up to 48 minutes of fairness degree, which could go unnoticed if an existing fairness measure was used due to the smoothing effect (averaging) or not quantifying the magnitude of outcome difference. In the context of EDs, this could mean patients delaying treatments because they did not want to wait. The alternative fairness testing techniques are not as effective and efficient as SBFT to measure the fairness degree for regression-based ML system.

The context is subtle and the impacts are nuanced. Overall, the continued exposure to bias adds up and influences health outcomes - more like a thousand cuts rather than one big event. In this scenario, our approach searches for test cases that expose the maximum difference in wait time prediction by the ED Software for patients that are identical apart from the sensitive attribute. We demonstrate that SBFT discovers discriminatory inputs of larger margins in terms of fairness degree compared to the baseline approaches. SBFT is also more efficient than the baseline approaches for discovering discriminatory inputs. In the future, we plan to investigate a new study from ambulance wait-time prediction software and work on other measures of fairness for regression-based machine learning systems.

Acknowledgements *Contributors:* Mrs Anne Loupis, Cabrini Institute, Melbourne; A/Prof Keith Joe, Monash Art, Design and Architecture, Monash Uni.; A/Prof Michael Ben-Meir, Austin and Cabrini Hospitals, Melbourne; Dr Hamed Akhlaghi, St Vincent's and Werribee Hospitals, Melbourne; Dr Jennie Hutton, St Vincent's Hospital, Melbourne; Dr Gabriel Blecher, Monash Medical Centre, Monash Health, Melbourne; Dr Paul Buntine, Box Hill Hospital, Eastern Health, Melbourne; Mrs Amy Sweeny, Gold Coast University Hospital, Bond University.

Collaborative Group Author: Australasian College for Emergency Medicine, Clinical Trial Network (ACEM CTN)

Contributor Statement: Funding: KW, MBM, KJ, BT; Ethics AL, KW; Clinical site investigators: KW, GB, PB, JH, HA, AS; Data acquisition: AL; Data cleaning: JJ; Project concept: AP, AA, BT, CT, JJ, KW, LK; Methods: AA, AP, BT, CT, JJ; Implementation: AP; Data analysis: AA, AP, JJ; Manuscript Draft: AP, AA, CT, BT, JJ, LK, KW; Manuscript revisions: All authors; AA takes responsibility for the overall manuscript.

Funding Open Access funding provided by University of Oulu including Oulu University Hospital. The Australian government, Medical Research Future Fund, via Monash Partners, funded this study. Researchers contributed in-kind donations of time. The Cabrini Institute and Monash University provided research infrastructure support. Chakkrit Tantithamthavorn was partially supported by the Australian Research Council's Discovery Early Career Researcher Award (DECRA) funding scheme (DE200100941).

Declarations

Ethics Approval The study received Monash Health ethics committee approval (RES-19-0000-763A).

Conflict of Interests The authors have no conflicts of interest to declare.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Aggarwal A, Lohia P, Nagar S, Dey K, Saha D (2019) Black box fairness testing of machine learning models. In: Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 625–635
- Alshahwan N, Gao X, Harman M, Jia Y, Mao K, Mols A, Tei T, Zorin I (2018) Deploying search based software engineering with sapienz at facebook. In: International Symposium on Search Based Software Engineering. Springer, pp 3–45
- Angwin J, Larson J, Mattu S, Kirchner L (2016) Machine bias. ProPublica
- Arcuri A, Briand L (2014) A hitchhiker's guide to statistical tests for assessing randomized algorithms in software engineering. *Softw Test Verif Reliab* 24(3):219–250
- Bairey Merz CN, Andersen H, Sprague E, Burns A, Keida M, Walsh MN, Greenberger P, Campbell S, Pollin I, McCullough C, Brown N, Jenkins M, Redberg R, Johnson P, Robinson B (2017) Knowledge, attitudes, and beliefs regarding cardiovascular disease in women: The women's heart alliance. *J Am Coll Cardiol* 70(2):123–132. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0735109717374077>
- Barocas S, Hardt M, Narayanan A (2018) Fairness and machine learning. fairmlbook.org
- Bellamy RKE, Dey K, Hind M, Hoffman SC, Houde S, Kannan K, Lohia P, Martino J, Mehta S, Mojsilović A et al (2019) Ai fairness 360: An extensible toolkit for detecting and mitigating algorithmic bias. *IBM J Res Dev* 63(4/5):4–1

- Bendick M (2007) Situation Testing for Employment Discrimination in the United States of America. [Online; accessed 29-November-2021]. [Online]. Available: <https://www.cairn.info/revue-horizons-strategiques-2007-3-page-17.htm>
- Berk R, Heidari H, Jabbari S, Joseph M, Kearns M, Morgenstern J, Neel S, Roth A (2017) A convex framework for fair regression. arXiv:1706.02409
- Binns R (2018) Fairness in machine learning: Lessons from political philosophy. In: Conference on Fairness, Accountability and Transparency. PMLR, pp 149–159
- Bishop PG, Bloomfield RE (2002) Worst case reliability prediction based on a prior estimate of residual defects. In: 13th International Symposium on Software Reliability Engineering, 2002. Proceedings. IEEE, pp 295–303
- Biswas S, Rajan H (2020) Do the machine learning models on a crowd sourced platform exhibit bias? an empirical study on model fairness. arXiv:2005.12379
- Calmon F, Wei D, Vinzamuri B, Ramamurthy KN, Varshney KR (2017) Optimized pre-processing for discrimination prevention. In: Advances in Neural Information Processing Systems, pp 3992–4001
- Caton S, Haas C (2020) Fairness in machine learning: A survey
- Chakraborty J, Majumder S, Menzies T (2021) Bias in machine learning software: Why? how? what to do? arXiv:2105.12195
- Chakraborty J, Majumder S, Yu Z, Menzies T (2020) Fairway: A way to build fair ml software. In: Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering, pp 654–665
- Chiappa S (2019) Path-specific counterfactual fairness. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol 33, pp 7801–7808
- Chouldechova A, Benavides-Prado D, Fialko O, Vaithianathan R (2018) A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In: Conference on Fairness, Accountability and Transparency. PMLR, pp 134–148
- Corbett-Davies S, Goel S (2018) The measure and mismeasure of fairness: A critical review of fair machine learning. arXiv:1808.00023
- Corbett-Davies S, Pierson E, Feller A, Goel S, Huq A (2017) Algorithmic decision making and the cost of fairness. In: Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining, pp 797–806
- Cortellessa V, Goseva-Popstojanova K, Appukkutty K, Guedem AR, Hassan A, Elnaggar R, Abdelmoez W, Ammar HH (2005) Model-based performance risk analysis. IEEE Trans Softw Eng 31(1):3–20
- Del Grosso C, Antoniol G, Di Penta M, Galinier P, Merlo E (2005) Improving network applications security: a new heuristic to generate stress testing data. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation, pp 1037–1043
- Di Somma S, Paladino L, Vaughan L, Lalle I, Magrini L, Magnanti M (2015) Overcrowding in emergency department: an international issue. Internal Emerg Med 10(2):171–175. [Online]. Available: <https://doi.org/10.1007/s11739-014-1154-8>
- Dieterich W, Mendoza C, Brennan T (2016) Compas risk scales: Demonstrating accuracy equity and predictive parity. Northpoint Inc 7(7.4):1
- Dwork C, Hardt M, Pitassi T, Reingold O, Zemel R (2012) Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference, pp 214–226
- Feldman M, Friedler SA, Moeller J, Scheidegger C, Venkatasubramanian S (2015) Certifying and removing disparate impact. In: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 259–268
- Ferral K Wisconsin supreme court allows state to continue using computer program to assist in sentencing the capital times. [Online; accessed 9-January- 2020]. [Online]. Available: <http://host.madison.com/ct/news/local/govt-and-politics/wisconsin-supreme-court-allows-state-to-continue-using-computer-program/article7eb67874-bf40-59e3-b62a-923d1626fa0f.html>
- Fraser G, Arcuri A (2013) Evosuite: On the challenges of test case generation in the real world. In: 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation. IEEE, pp 362–369
- Friedman B, Nissenbaum H (1996) Bias in computer systems. ACM Trans Inf Syst 14(3):330–347
- Galhotra S, Brun Y, Meliou A (2017) Fairness testing: testing software for discrimination. In: Joint Meeting on Foundations of Software Engineering (FSE). ACM, pp 498–510
- Ghaffary S (2019) The algorithms that detect hate speech online are biased against black people. [Online; accessed 14-October-2020]. [Online]. Available: <https://www.vox.com/recode/2019/8/15/20806384/social-media-hate-speech-bias-black-african-american-facebook-twitter>
- Grgic-Hlaca N, Zafar MB, Gummadi KP, Weller A (2016) The case for process fairness in learning: Feature selection for fair decision making. In: NIPS Symposium on Machine Learning and the Law, vol 1, p 2

- Hardawar D (2012) Staples, home depot, and other online stores change prices based on your location. [Online; accessed 14-October-2020]. [Online]. Available: <https://venturebeat.com/2012/12/24/staples-online-stores-price-changes/>
- Hardt M, Price E, Srebro N (2016) Equality of opportunity in supervised learning. *Adv Neural Inf Process Syst* 29:3315–3323
- Hern A (2020) Twitter apologises for 'racist' image-cropping algorithm. [Online; accessed 7-August-2021]. [Online]. Available: <https://www.theguardian.com/technology/2020/sep/21/twitter-apologises-for-racist-image-cropping-algorithm>
- Horkoff J (2019) Non-functional requirements for machine learning: Challenges and new directions. In: 2019 IEEE 27th International Requirements Engineering Conference (RE). IEEE, pp 386–391
- Ingold D, Soper S (2016) Amazon doesn't consider the race of its customers. should it? [Online; accessed 14-October-2020]. [Online]. Available: <https://www.bloomberg.com/graphics/2016-amazon-same-day/>
- Ingold D, Soper S (2016) Amazon doesn't consider the race of its customers. should it? *Bloomberg News*
- Johnson B, Bartola J, Angell R, Keith K, Witty S, Giguere SJ, Brun Y (2020) Fairkit, fairkit, on the wall, who's the fairest of them all? supporting data scientists in training fair models. arXiv:2012.09951
- Juergens CP, Dabin B, French JK, Kritharides L, Hyun K, Kilian J, Chew DerekPB, Brieger D (2016) English as a second language and outcomes of patients presenting with acute coronary syndromes: results from the concordance registry. *Med J Aust* 204(6):239–239. [Online]. Available: <https://doi.org/https://doi.org/10.5694/mja15.00812>
- Kamiran F, Calders T (2012) Data preprocessing techniques for classification without discrimination. *Knowl Inf Syst* 33(1):1–33
- Kamishima T, Akaho S, Sakuma J (2011) Fairness-aware learning through regularization approach. In: 2011 IEEE 11th International Conference on Data Mining Workshops. IEEE, pp 643–650
- Klare BF, Burge MJ, Klontz JC, Bruegge RWV, Jain AK (2012) Face recognition performance: Role of demographic information. *IEEE Trans Inf Forensic Secur* 7(6):1789–1801
- Lakhoria K, Harman M, Gross H (2013) Austin: An open source tool for search based software testing of c programs. *Inf Softw Technol* 55(1):112–125
- Ledford H (2019) Millions of black people affected by racial bias in health-care algorithms. [Online; accessed 7-August-2021]. [Online]. Available: <https://www.nature.com/articles/d41586-019-03228-6>
- López-Ibáñez M, Dubois-Lacoste J, Cáceres LP, Birattari M, Stützle T (2016) The irace package: Iterated racing for automatic algorithm configuration. *Oper Res Perspect* 3:43–58
- Luong BT, Ruggieri S, Turini F (2011) k-nn as an implementation of situation testing for discrimination discovery and prevention. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp 502–510
- Mahdawi A (2018) Is your friend getting a cheaper Uber fare than you are? [Online; accessed 7-August-2021]. [Online]. Available: <https://www.theguardian.com/commentisfree/2018/apr/13/uber-lyft-prices-personalized-data>
- Mao K, Harman M, Jia Y (2016) Sapienz: Multi-objective automated testing for android applications. In: Proceedings of the 25th International Symposium on Software Testing and Analysis, pp 94–105
- Mattioli D (2012) On Orbitz, Mac Users Steered to Pricier Hotels. [Online; accessed 9-January-2020]. [Online]. Available: <http://www.wsj.com/articles/SB10001424052702304458604577488822667325882>
- McMinn P (2004) Search-based software test data generation: a survey. *Softw Test Verif Reliab* 14(2):105–156
- McSweeney JC, Rosenfeld AG, Abel WM, Braun LT, Burke LE, Daugherty SL, Fletcher GF, Gulati M, Mehta LS, Pettey C, Reckelhoff JF (2016) Preventing and experiencing ischemic heart disease as a woman: State of the science. *Circulation* 133(13):1302–1331. [Online]. Available: <https://doi.org/10.1161/CIR.0000000000000381>
- Mehrabi N, Morstatter F, Saxena N, Lerman K, Galstyan A (2019) A survey on bias and fairness in machine learning. arXiv:1908.09635
- Mehta LS, Beckie TM, DeVon HA, Grines CL, Krumholz HM, Johnson MN, Lindley KJ, Vaccarino V, Wang TY, Watson KE, Wenger NK (2016) Acute myocardial infarction in women. *Circulation* 133(9):916–947. [Online]. Available: <https://doi.org/10.1161/CIR.0000000000000351>
- Mullainathan S (2019) Biased algorithms are easier to fix than biased people. www.nytimes.com/2019/12/06/business/algorithm-bias-fix.html, accessed: 10/12/2019.
- Olson P (2011) CNN Money, The algorithm that beats your bank manager. <http://www.forbes.com/sites/parmyolson/2011/03/15/the-algorithm-that-beats-your-bank-manager/#cd84e4f77ca8>, [Accessed 9/11/2020]
- Panichella A, Kifetew FM, Tonella P (2015) Reformulating branch coverage as a many-objective optimization problem. In: 2015 IEEE 8th international conference on software testing, verification and validation (ICST). IEEE, pp 1–10

- Panichella A, Kifetew FM, Tonella P (2018) A large scale empirical comparison of state-of-the-art search-based test case generators. *Inf Softw Technol* 104:236–256
- Perera A, Aleti A, Böhme M, Turhan B (2020) Defect prediction guided search-based software testing. In: Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. ACM
- Puschner P, Burns A (2000) Guest editorial: A review of worst-case execution-time analysis. *Real-Time Syst* 18(2-3):115–128
- Ramamoorthy CV, Ho GS (1980) Performance evaluation of asynchronous concurrent systems using petri nets. *IEEE Trans Softw Eng* 5:440–449
- Ribeiro MT, Singh S, Guestrin C (2016) "why should i trust you?" explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 1135–1144
- Runeson P, Engström E, Storey M-A (2020) The design science paradigm as a frame for empirical software engineering. In: Contemporary empirical methods in software engineering. Springer, pp 127–147
- Selbst AD, Boyd D, Friedler SA, Venkatasubramanian S, Vertesi J (2019) Fairness and abstraction in sociotechnical systems. In: Proceedings of the conference on fairness, accountability, and transparency, pp 59–68
- Shah S, Patel A, Rumoro DP, Hohmann S, Fullam F (2015) Managing patient expectations at emergency department triage. *Patient Exper J* 2(2):31–44
- Sharkey A (2020) Care robots for the elderly are dangerous. [Online; accessed 14-October-2020]. [Online]. Available: <https://www.telegraph.co.uk/science/2016/05/30/care-bots-for-the-elderly-are-dangerous-warns-artificial-intelli>
- Sharma A, Wehrheim H (2019) Testing machine learning algorithms for balanced data usage. In: 2019 12th IEEE Conference on Software Testing, Validation and Verification (ICST). IEEE, pp 125–135
- Siegel RB (2003) Equality talk: Antisubordination and anticlassification values in constitutional struggles over brown. *Harv L Rev* 117:1470
- Soremekun OA, Takayesu JK, Bohan SJ (2011) Framework for analyzing wait times and other factors that impact patient satisfaction in the emergency department. *J Emerg Med* 41(6):686–692
- Stehli J, Duffy SJ, Burgess S, Kuhn L, Gulati M, Chow C, Zaman S (2021) Sex disparities in myocardial infarction: biology or bias? *Heart Lung Circul* 30(1):18–26
- Strickland E (2016) Doc bot preps for the or. *IEEE Spectr* 53(6):32–60
- Strobel S, Ren KY, Dragoman A, Pettit C, Stancati A, Kallergis D, Smith M, Sidhu K, Rutledge G, Mondoux S (2021) Do patients respond to posted emergency department wait times: Time-series evidence from the implementation of a wait time publication system in hamilton, canada. *Ann Emerg Med*
- Sun J, Lin Q, Zhao P, Zhang Q, Xu K, Chen H, Hu CJ, Stuntz M, Li H, Liu Y (2017) Reducing waiting time and raising outpatient satisfaction in a chinese public tertiary general hospital-an interrupted time series study. *BMC Public Health* 17(1):1–11
- Tatman R (2017) Gender and dialect bias in youtube's automatic captions. In: Proceedings of the First ACL Workshop on Ethics in Natural Language Processing, pp 53–59
- Tramer F, Atlidakis V, Geambasu R, Hsu D, Hubaux J-P, Humbert M, Juels A, Lin H (2017) Fairtest: Discovering unwarranted associations in data-driven applications. In: 2017 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp 401–416
- Udell JA, Fonarow GC, Maddox TM, Cannon CP, Frank Peacock W, Laskey WK, Grau-Sepulveda MV, Smith EE, Hernandez AF, Peterson ED et al (2018) Sustained sex-based treatment differences in acute coronary syndrome care: insights from the american heart association get with the guidelines coronary artery disease registry. *Clin Cardiol* 41(6):758–768
- Udeshi S, Arora P, Chattopadhyay S (2018) Automated directed fairness testing. In: Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering, pp 98–108
- Vargha A, Delaney HD (2000) A critique and improvement of the cl common language effect size statistics of mcgraw and wong. *J Educ Behav Stat* 25(2):101–132
- Vogel B, Acevedo M, Appelman Y, Merz CNB, Chieffo A, Figtree GA, Guerrero M, Kunadian V, Lam CSP, Maas AHEM et al (2021) The lancet women and cardiovascular disease commission: reducing the global burden by 2030. *The Lancet*
- Walker K, Jiarpakdee J, Loupis A, Tantithamthavorn C, Joe K, Ben-Meir M, Akhlaghi H, Hutton J, Wang W, Stephenson M, Blecher G, Buntine P, Sweeny A, Turhan B (2021) On behalf of the Australasian College for Emergency Medicine, Emergency medicine patient wait time multivariable prediction models: a multicentre derivation and validation study. medRxiv, [Online]. Available: <https://www.medrxiv.org/content/early/2021/03/24/2021.03.19.21253921>

- Walker K, Stephenson M, Loupis A, Ben-Meir M, Joe K, Stephenson M, Lowthian J, Yip B, Wu E, Hansen K et al (2020) Displaying emergency patient estimated wait times: A multi-centre, qualitative study of patient, community, paramedic and health administrator perspectives. *Emergency Medicine Australasia*
- Wechkunanukul K, Grantham H, Teubner D, Hyun KK, Clark RA (2016) Presenting characteristics and processing times for culturally and linguistically diverse (cald) patients with chest pain in an emergency department: Time, ethnicity, and delay (ted) study ii. *Int J Cardiol* 220:901–908
- Wegener J, Bühler O (2004) Evaluation of different fitness functions for the evolutionary testing of an autonomous parking system. In: *Genetic and Evolutionary Computation Conference*. Springer, pp 1400–1412
- Zhang BH, Lemoine B, Mitchell M (2018) Mitigating unwanted biases with adversarial learning. In: *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp 335–340
- Zhang JM, Harman M (2021) 'ignorance and prejudice' in software fairness. In: 2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE). IEEE, pp 1436–1447
- Zhang JM, Harman M, Ma L, Liu Y (2020) Machine learning testing: Survey, landscapes and horizons. *IEEE Trans Softw Eng*
- Zhang L, Wu Y, Wu X (2016) Situation testing-based discrimination discovery: A causal inference approach
- Zhang P, Wang J, Sun J, Dong G, Wang X, Wang X, Dong JS, Dai T (2020) White-box fairness testing through adversarial sampling. In: *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pp 949–960

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Anjana Perera is a PhD student in Software Engineering at Monash University, Australia. His research interests include search-based software engineering, automated test generation and fairness testing of AI-based systems. Anjana received his BSc. (Hons) of Engineering Degree, specialising in Electronics and Telecommunication Engineering, from University of Moratuwa, Sri Lanka, in 2017. He was a software engineer, developing a latency critical, highly scalable and reliable electronic trading system for London Stock Exchange Group (LSEG) at LSEG Technology (formerly known as Millennium IT) until 2018. The goal of his PhD is to use defect prediction to improve the bug detection capability of search-based software testing. Anjana has served as PC member at SSBSE 2020 and Artefact Evaluation PC member at ASE 2021 and ICSME 2021. For more information, please visit: <https://anjana-perera.github.io>.

Aldeida Aleti is an Associate Professor and Associate Dean of Engagement and Impact at the Faculty of Information Technology, Monash University. Aldeida received her PhD in 2012 from Swinburne University of Technology, and now works in the area of Search-Based Software Engineering (SBSE), with a particular focus on the methodological aspects of how to assess the effectiveness of these techniques. This includes fitness landscape characterisation to analyse how hard SBSE problems are for search techniques, and algorithm selection, which is about identifying which SBSE technique works in what scenario. Aldeida has published more than 50 papers in top AI, optimisation and software engineering venues, served as PC member and organising committee at both SE and optimisation conferences, such as ASE 2020, 21, ICSE 2019, GECCO 2017, SSBSE 2018, 19. Aldeida has attracted more than \$2,500,000 in competitive funding and was awarded the prestigious Discovery Early Career Researcher (DECRA) Award from the Australian Research Council.

Chakkrit Tantithamthavorn is a 2020 ARC DECRA Fellow and a Lecturer in Software Engineering in the Faculty of Information Technology, Monash University, Melbourne, Australia. His current fellowship is focusing on the development of "Practical and Explainable Analytics to Prevent Future Software Defects". His work has been published at several top-tier software engineering venues, such as the IEEE Transactions on Software Engineering (TSE), the Springer Journal of Empirical Software Engineering (EMSE) and the International Conference on Software Engineering (ICSE). More about Chakkrit and his work is available online at <http://chakkrit.com>.


Jirayus Jiarpakdee is a Ph.D. candidate at Monash University, Australia. His research interests include empirical software engineering and mining software repositories (MSR). The goal of his Ph.D. is to apply the knowledge of statistical modelling, experimental design, and software engineering to improve the explainability of defect prediction models.

Burak Turhan *PhD (Boğaziçi University)*, is a Professor of Software Engineering at the University of Oulu and an Adjunct Professor (Research) in the Faculty of IT at Monash University. His research focuses on empirical software engineering, software analytics, quality assurance and testing, human factors, and (agile) development processes. He is a Senior Associate Editor of Journal of Systems and Software, an Associate Editor of ACM Transactions on Software Engineering and Methodology and Automated Software Engineering, an Editorial Board Member of Empirical Software Engineering, Information and Software Technology, and Software Quality Journal, and a Senior Member of ACM and IEEE. For more information, please visit: <https://turhanb.net>.

Lisa Kuhn is a health services researcher, having commenced her career as an emergency nurse 30 years ago. Her main research interests involve ensuring equity in cardiovascular healthcare for women and underserved groups through generating and translating evidence into practice. She uses quantitative, qualitative and mixed methods research designs, and systematic reviews to investigate a broad range of issues including heart disease, emergency care and mental health. Lisa is a member of the Editorial Board of Australian Critical Care and has served on a health service Ethics Committee. She is Chair (Nursing) of the Monash Emergency Research Collaborative, is Co-lead of the Women's Heart Health Grand Challenge at the Victorian Heart Institute, and is a member of the Research Leadership Committee for the soon to open Victorian Heart Hospital.

Katie Walker is an emergency physician at Monash Health, Melbourne, Australia and a Clinical Professor at the School of Clinical Sciences, Monash University. Her main interests are in health services research and evaluating and integrating novel technologies. Katie is a member of several editorial boards, including Annals of Emergency Medicine and the Emergency Medicine Journal. She is chair of the Australasian College for Emergency Medicine.

Affiliations

Anjana Perera¹  · **Aldeida Aleti**¹ · **Chakkrit Tantithamthavorn**¹ · **Jirayus Jiarpakdee**¹ · **Burak Turhan**² · **Lisa Kuhn**³ · **Katie Walker**³

Aldeida Aleti
Aldeida.Aleti@monash.edu

Chakkrit Tantithamthavorn
Chakkrit@monash.edu

Jirayus Jiarpakdee
Jirayus.Jiarpakdee@monash.edu

Lisa Kuhn
Lisa.Kuhn@monash.edu

Katie Walker
Katie.Walker@monash.edu

¹ Faculty of Information Technology, Monash University, Melbourne, Australia

² Faculty of Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland

³ Faculty of Medicine, Nursing and Health Sciences, Monash University, Melbourne, Australia