Discover Applied Sciences

Review

Constrained adversarial learning for automated software testing: a literature review

João Vitorino[1,2] · **Tiago Dias**[1,2] · **Tiago Fonseca**[1] · **Eva Maia**[1,2] · **Isabel Praça**[1,2]

Received: 2 January 2025 / Accepted: 30 April 2025
Published online: 27 May 2025

Abstract
It is imperative to safeguard computer applications and information systems against the growing number of cyber-attacks. Automated software testing can be a promising solution to quickly analyze many lines of code and detect vulnerabilities and possible attack vectors by generating function-specific testing data. This process draws similarities to the constrained adversarial examples generated by adversarial learning methods, so there could be significant benefits to the integration of these methods in testing tools. Therefore, this literature review is focused on the current state-of-the-art of constrained data generation methods applied for adversarial learning and software testing, aiming to guide researchers and developers to enhance software testing tools with adversarial testing methods and improve the resilience and robustness of their information systems. The found constrained data generation applications were systematized, and the advantages and limitations of approaches specific for white-box, grey-box, and black-box testing were analyzed, identifying research gaps and opportunities to improve automated testing tools with data generated by adversarial attacks.

Keywords Software development · Adversarial testing · Adversarial attack · Constrained data generation · Machine learning

Abbreviations

API	Application programming interface
CGAN	Conditional GAN
CVAE	Conditional VAE
GA	Genetic algorithm
GAN	Generative adversarial network
REST	Representational state transfer
SUT	System under test
VAE	Variational autoencoder
WGAN	Wasserstein GAN

✉ João Vitorino, jpmvo@isep.ipp.pt; Tiago Dias, tiada@isep.ipp.pt; Tiago Fonseca, calof@isep.ipp.pt; Eva Maia, egm@isep.ipp.pt; Isabel Praça, icp@isep.ipp.pt | [1]ISEP, Polytechnic of Porto, 4249-015 Porto, Portugal. [2]Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), 4249-015 Porto, Portugal.

1 Introduction

From decision-making processes to electronic commerce and a variety of online services, the dependency of modern organizations on computer programs and information systems is not slowing down soon [1]. However, no computer code is flawless, so the integration of different technologies and software components can add hidden vulnerabilities ready to be exploited in novel zero-day cyber-attacks. Not even systems that rely on artificial intelligence are secure, due to the susceptibility of machine learning models to attacks with adversarial examples [2]. As organizations rush the deployment of computer applications without thoroughly testing all software components, they are increasing the risk of an attacker with malicious intents disrupting their critical business processes and impacting their business continuity [3].

To safeguard organizations and their personnel from the damage caused by an attack, it is imperative to integrate security best practices in software development workflows and ensure that all computer applications are adequately tested before being deployed. Nonetheless, despite growing concerns for software testing, it is still mostly a manual process where developers design the tests that each distinct part of the code must be validated against. This process is time-consuming, expensive, and usually incomplete, because human personnel cannot formulate tests for the entire attack surface of an information system nor tackle all possible attack vectors [4].

Automated software testing is a very promising research field because an automated and methodical approach can address the attack vectors that would otherwise be left unnoticed. By quickly analyzing thousands of lines of code, both time and monetary costs can be reduced. Nonetheless, to perform trustworthy tests and ensure that a program is functioning correctly, the generated testing data must address the different constraints of the tested functions and software components. So, in short, an automated testing tool slightly modifies the parameters of a function to check if it deviates from the expected behaviour [5]. In another pertinent research field, adversarial machine learning, a similarity can be noticed: an adversarial attack method creates slight modifications in data features to deceive a model, causing misclassifications and unexpected behaviors [6]. Therefore, both automated testing tools and adversarial learning methods serve similar purposes and there could be significant benefits to their integration.

Due to the lack of publications addressing the use of adversarial learning methods to improve software testing tools, with data perturbations generated according to the specific constraints of the tested functions and software components, this work investigates the recent scientific developments and technological advances of these fields. The main research question of this literature review was: *How can automated software testing tools be improved with constrained adversarial machine learning methods?*.

To guide the research performed in the scope of each field, software testing and adversarial learning, the main question was divided into two narrower sub-questions:

RQ1: How can adversarial learning methods be leveraged for constrained data generation?
RQ2: How can automated software testing tools benefit from constrained data generation?

By investigating the current approaches in a methodical manner, this work can guide researchers and developers to enhance automated testing tools with adversarial learning methods and improve the resilience and robustness of the System Under Test (SUT). This paper is organized into multiple sections. Section 2 describes the search methodology. Sections 3 and 4 present and discuss the found publications for RQ1 and RQ2. Finally, Sect. 5 presents the concluding remarks and future research topics.

2 Search methodology

This section describes the methodical review process employed for both RQ1 and RQ2. To achieve a transparent, replicable, and complete literature review, a part of the PRISMA methodology [7] was followed. Search terms were created to be used in reputable bibliographic databases, and several inclusion and exclusion criteria were defined. After an initial screening phase with the titles and abstracts of the found publications, their full texts were assessed for eligibility, and only then they were included in the review.

2.1 Search terms

Search terms were chosen after a careful initial analysis of the literature. Since constrained data generation is a relatively unexplored research topic without standardized keywords, the search had to cover broader terms to prevent narrowing it down too much and obtain relevant publications for the scope of the review. Therefore, some word variations were also considered to widen the search to more common concepts like `number generators` and `conditional generators`, as well as approaches that address `constraints` and `restrictions`, which are word variations represented by "*".

Considering that RQ1 and RQ2 share a common scope, constrained data generation, only their target field was different in the search queries. For RQ1, the query included the `adversarial` term to focus on approaches related to adversarial machine learning and adversarial perturbations. For RQ2, the target field was `software testing` to analyze the publications specific to this field. Table 1 presents the combined query, where the terms of each scope were aggregated with AND operators.

2.2 Search sources

The primary search source was Science Direct [8], which is a large bibliographic database of scientific journals and conference proceedings provided by the publisher Elsevier. Due to their acknowledged relevance for scientific literature of software engineering, the search also included the digital libraries of the Institute of Electrical and Electronics Engineers (IEEE) [9], the Association for Computing Machinery (ACM) [10], and the Multidisciplinary Digital Publishing Institute (MDPI) [11].

2.3 Inclusion and exclusion criteria

The review was performed in the beginning of 2024 and was focused on peer-reviewed conference papers and journal articles introducing or applying constrained generation approaches, in the English language. Considering that both adversarial learning and software testing are active areas of research and had relevant technological advances in recent years, the selected time frame was 2017 to 2023, the latest 7 whole years of scientific developments, at the time the review was started. To limit the findings to recent developments published and stored in these databases, backward snowballing was not performed in this review.

The widened search queries led to more relevant publications within this time frame, but also to many publications that were not aligned with the purpose of this review. To screen the publications and assess their eligibility, several criteria were defined. These intended to include only the publications presenting possible approaches that took constraints into account, excluding general surveys and reviews that were not focused on constraints.

Furthermore, the criteria also excluded publications that mentioned constraints or conditions but not during their data generation processes, and publications that did not detail how the utilized constraints were applied nor how they affected the data generation. Table 2 provides an overview of the defined inclusion and exclusion criteria for both RQ1 and RQ2.

Table 1 Search query for RQ1

Scope	Keywords
Constrained	Constrained* OR Conditional* OR Restricted*
Field	Adversarial learning OR Software testing
Data	Data OR Sample OR Example OR Number
Generation	Generation OR Generator

* Wildcard symbol for keyword variations

Table 2 Inclusion and exclusion criteria

Inclusion criteria	Exclusion criteria
IC1 Peer-reviewed journal article or conference paper	EC1 duplicated publication
IC2 Published from 2017 to 2023	EC2 survey or review
IC3 Available in the English language	EC3 constraints not in generation
IC4 Introduced or applied an approach for constrained data generation	EC4 constraints not detailed
	EC5 full text not available

3 Adversarial machine learning

This section presents and discusses the findings of RQ1, systematizing the constrained data generation applications for adversarial machine learning. A total of 1915 records were initially obtained by applying the query to the contents of the publications stored in the selected databases. After the screening and eligibility assessment phases, 130 publications were included in the review. This process is detailed in Fig. 1.

The included publications were systematized to identify the current applications of constrained data generation and the methods they are based on. A total of 98 applications were identified across 11 sectors and industries, although a few were more general and not specific to a single industry. The number of applications grew from 2 in 2017 to 36 in 2023, with the cybersecurity and healthcare sectors being the most prominent and having more diverse applications, followed by the aerospace and energy sectors. These are the key sectors where it was identified that constrained data generation is necessary and there is a growing number of applications.

Table 3 summarizes the reviewed applications and the methods they were based on. These base methods were either explicitly built upon or their concepts were implicitly used in new implementations. The table is divided according to the main sectors and industries: aerospace, automotive, chemical, cybersecurity, economy, energy, geological, healthcare, mechanical, music, and water. The remaining publications that were more broad are included in a general subdivision at the end of the table.

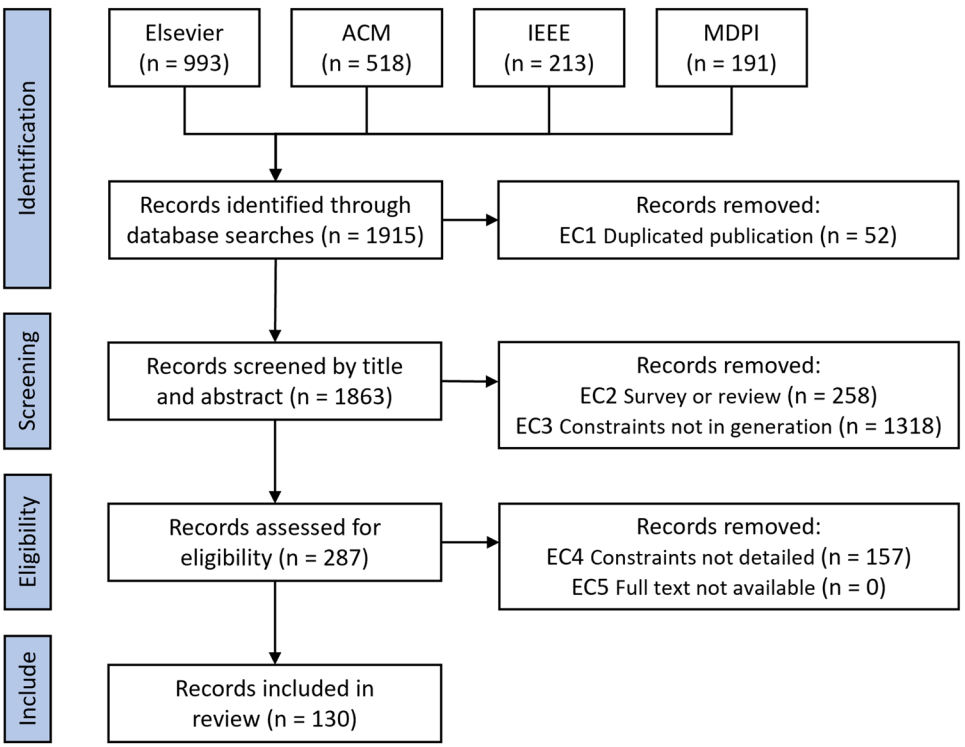

Fig. 1 PRISMA search process for RQ1

Table 3 Applications of constrained data generation

Sector/ Industry	Applications	Base Methods
Aerospace	Synthetic-Aperture Radar (SAR) image reconstruction, and hyperspectral classification [12–15]	CGAN, CVAE
	Satellite mapping of surface water, road surface area, reflectance, and radiance [16–19]	CGAN
	Precipitation estimation, aircraft detection, remote sensing, and signal separation [20–23]	CGAN
	Object detection in unmanned drone inspection tasks with Unmanned Aerial Vehicles (UAVs) [24, 25]	CGAN
Automotive	Urban traffic trajectory simulation, and parking occupancy estimation [26, 27]	CGAN, CVAE
	Motor vehicle transmission gear and Inertial Measurement Unit (IMU) signal analysis [28, 29]	CGAN
	Ultrasonic signal sensors for autonomous driving systems [30]	CGAN
	Vehicular and transportation network simulation [31, 32]	CGAN
Chemical	Chemical production process fault detection and diagnosis [33, 34]	CGAN, WGAN
	Soft sensors for multiple chemical compound analysis [35]	CGAN, WGAN
	Environmental feature extraction in aerosol optical depth sensors [36]	CGAN
Cybersecurity	Network intrusion detection, anomaly detection, and cyber-attack classification [37–42]	CGAN, evolutionary algorithms
	Illegal webpage detection, and malicious domain name detection [38, 43, 44]	CGAN, evolutionary algorithms
	Input generation for software testing, and malware detection in code blocks [45–48]	CGAN, fuzzy algorithms
	Fake user detection, and behavior and keystroke dynamics generation [49–51]	CGAN
	Speech recognition and denoising, and voice spoofing detection [52–55]	CGAN
	Data masking, obfuscation, anonymization, and compression [56–59]	CGAN
Economy	Monte Carlo simulation design for economic studies [60]	CGAN, WGAN
	Multi-period financial portfolio time-series simulation [61]	CGAN
Energy	Power distribution, economic load dispatch, and optimal power flow simulation [32, 62–66]	CGAN, graph algorithms
	Anomalous energy consumption detection, and electricity theft detection [67, 68]	CVAE, WGAN
	Charging behavior, and supply and demand forecasting [69–74]	CGAN
Sector/ Industry	Applications	Base Methods
Geological	Seismic data, sedimentary facies, and geological patterns simulation [75–78]	CGAN
	Virtual landscape and terrain authoring [79, 80]	CGAN

Table 3 (continued)

Sector/ Industry	Applications	Base Methods
Healthcare	Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) analysis [81–86]	CGAN, WGAN
	X-ray and Computed Tomography (CT) scan analysis [87–91]	CGAN, WGAN
	Electrocardiogram (ECG) and Electroencephalogram (EEG) signal analysis [92–97]	CGAN, WGAN
	Fundus image generation and Retinopathy screening [98]	CGAN
	Electrodermal Activity (EDA) generation for stress detection [99]	CGAN
	Lung and skin lesions detection [100]	CGAN
	Mammographic density segmentation and breast cancer diagnosis [101–104]	CGAN
	Gene expression profiling for cancer diagnosis [105]	CGAN
	Microscopic blood smear tests and single-cell segmentation [106, 107]	CGAN
	Vocal cord and voice disorder detection [108]	CGAN
	Connectomes segmentation for reconstruction of neural circuits [109]	CGAN
Mechanical	Rotating machinery and rolling bearing fault diagnosis [110–115]	CGAN, CVAE, WGAN
	Industrial equipment failure, conveyor belt, and joint damage detection [116–120]	CGAN, CVAE, WGAN
	Wind turbine gearbox fault diagnosis [121–123]	CGAN, CVAE
Music	Variable-length music and audio generation [124, 125]	CGAN, CVAE
	Cross-modal musical performance generation [126]	CGAN
Water	Water flow simulation for water supply and distribution systems [32, 63, 127]	CGAN, graph algorithms
	Maritime vessel trajectory prediction in waterways [128]	CGAN
General	Architectural space layout generation [129]	CGAN, graph algorithms
	Motion, stretches, and compressions simulation [130, 131]	CGAN, WGAN, skinning algorithms
	Portrait picture generation according to age and facial attributes [132–134]	CGAN
	Font generation according to different font styles [135]	CGAN
	Pathfinding for Very Large-Scale Integration (VLSI) of integrated circuits [136]	CGAN
	Wireless indoor signal denoising, positioning, and tracking [137, 138]	CGAN
	Word-gesture text input, handwritten text, and digit recognition[139–141]	CGAN

The increasing use of constraints across various sectors and industries highlights that better data can be generated when the specificities of the task at hand are considered. Nonetheless, since most publications handle image classification datasets, they employ methods with simple constraints to generate new synthetic images according to the classes of a dataset. Most approaches are based on improved versions of the Generative Adversarial Network (GAN) [142] and the Variational Autoencoder (VAE) [143], such as Conditional GAN (CGAN) [144], Wasserstein GAN (WGAN) [145], and Conditional VAE (CVAE) [146].

Even though the CGAN and CVAE improved versions of deep learning algorithms spread across the literature because they can learn to generate different data for different classes, the samples of each class are freely generated without addressing any complex constraint. For more complex tasks in sectors that require other data types like tabular data and time series, some approaches are starting to rely on methods that support more rigorous configurations and more complex constraints. For instance, some researchers employ evolutionary computation with swarm intelligence and custom-built Genetic Algorithms (GAs) [37], and others apply fuzzy logic in their algorithms to deal with the uncertainty of the information [45].

Despite the wide range of applications of constrained data generation, the few approaches that could potentially be transferable to automated software testing tools were mainly developed for cybersecurity applications related to communication networks and cyber-physical applications in energy grids and water distribution networks. Due to the low-quality data that conventional methods would generate in these complex domains, some researchers encapsulated the data generation processes in mechanisms that enforce task-specific constraints.

In [63], the authors intended to simulate electrical grids and water supply and distribution systems. Each action or event in one part of these cyber-physical systems would affect the entire energy and water networks, so the presence of a given value at a given feature would restrict the values that other features could have. Linear equality constraints were defined to fulfill the physical capacity requirements of each network and multiple simulations were performed to validate them. This optimization approach could be useful to model the relationships between different nodes of a network and different software components, but the constraints would need to be carefully redesigned for each minor change in the tested functions or to be transferred to different SUT architectures.

To provide a structure that could be adapted to changes, in [32], energy and water networks were addressed through graph theory. The graph structure provided a more rigorous representation of the electric and water flow physics between different nodes in a network, enabling the simulation of cyclic trends and acyclic congestions, as well as their effects in other nodes. Nodes and their connections could be added or removed to adapt the structure to different flow networks, although it was tailored to the specific physical constraints of those domains. Additionally, it is also relevant to highlight the mechanism introduced in [131] to perform physically accurate simulations of stretches and compressions of objects in three-dimensional spaces. Despite not being as rigorous as flow graphs and only being tested with triangulated cloth meshes of human body poses and joint rotations, this mechanism could be transferable to the perturbation of images and objects provided to the SUT and test its complex code structure.

Regarding biomedical images, in [100], the authors addressed the insufficient label granularity of most datasets. Since multiple sub-types of a disease can be aggregated into a single broader label, the data samples of a class can exhibit entirely different data distributions and feature correlations. Even if a CGAN is used to distinguish between different classes of a dataset, the generated data may mix the characteristics of multiple sub-types and therefore may not correspond to an actual disease. Conditioning vectors were used to restrict the modification of relevant disease features according to the values exhibited by other similar samples that were presumed to be of the same sub-type. Even though this mechanism is still based on a CGAN and would need to be redesigned for other data types, the concept of configuring class-specific value constraints could be useful to improve data quality and generate relevant parameter combinations in a reduced amount of time.

These sub-classes were further explored in [40], where the authors established that an adversarial example must be valid within its domain structure, which corresponds to the input of the software component being tested, and be coherent with the characteristics and purposes of each class. Constrained data perturbations were generated by analysing the value intervals of individual features and the value combinations of groups of features. Despite being developed for network intrusion detection, the method created constrained adversarial examples that fulfilled complex constraints. This method could be useful to software testing because the tested functions could be aggregated into classes according to their functionalities and function-specific value perturbations could be performed, quickly generating more adequate parameter combinations to achieve a high code coverage.

Overall, the reviewed adversarial machine learning approaches present valuable insights for automated software testing applications, although they are tailored to the specificities of cyber-physical systems. These mechanisms could

be redesigned to consider the value constraints of the parameters of each tested function and generate high-quality data to test different software components. Therefore, it is also pertinent to explore the recent advances in automated software testing tools and analyze the ways that constrained data generation could be used to improve them.

4 Automated software testing

This section presents and discusses the findings of RQ2, thoroughly analyzing the advantages and limitations of approaches specific to software testing. A total of 537 records were identified in the selected databases. After the screening of their titles and abstracts and the assessment of their full texts for eligibility, 17 publications were included in the review. This process is detailed in Fig. 2.

Software testing approaches can be divided into a three-fold: (i) white-box, (ii) black-box, and (iii) grey-box. The first is mostly executed from a developer's point of view, where all the domain constraints and the system's inner workings are known and leveraged. The black-box approach is characterized by its lack of knowledge and access to the source code [147]. In a black-box testing setting, the tester has only information about what the system is supposed to do and how it can be interacted with. However, further intricacies of the SUT are unknown [147]. Lastly, grey-box testing combines both approaches. However, the implementation of the SUT and its domain is only partially known [148].

Software testing automation can be extremely challenging, depending on the approach and levels of testing considered. The key aspect to achieve automation relies on the construction of the test cases, which are a reflection of the SUT's functions parameter values and their execution sequence [149]. When considering the generation of input values to test the system, one must deal with the difficulties related with finding the right input values along with their combination and the right execution sequence, since the prior execution of system functionality may alter its state, which leads to different system execution paths [149].

Depending on the information available to the automated testing tool these can have enormous search spaces, which is not ideal when considering the possible execution time [150]. Depending on the software and the levels of testing that the automated tool is trying to achieve, a white-box automated testing tool with access to more domain information might perform better and more efficient testing of the SUT's domain. Nonetheless, testing a system from a user perspective using black-box techniques may sometimes raise different exceptions and errors that are not usually expected nor visually understandable in the source code [151].

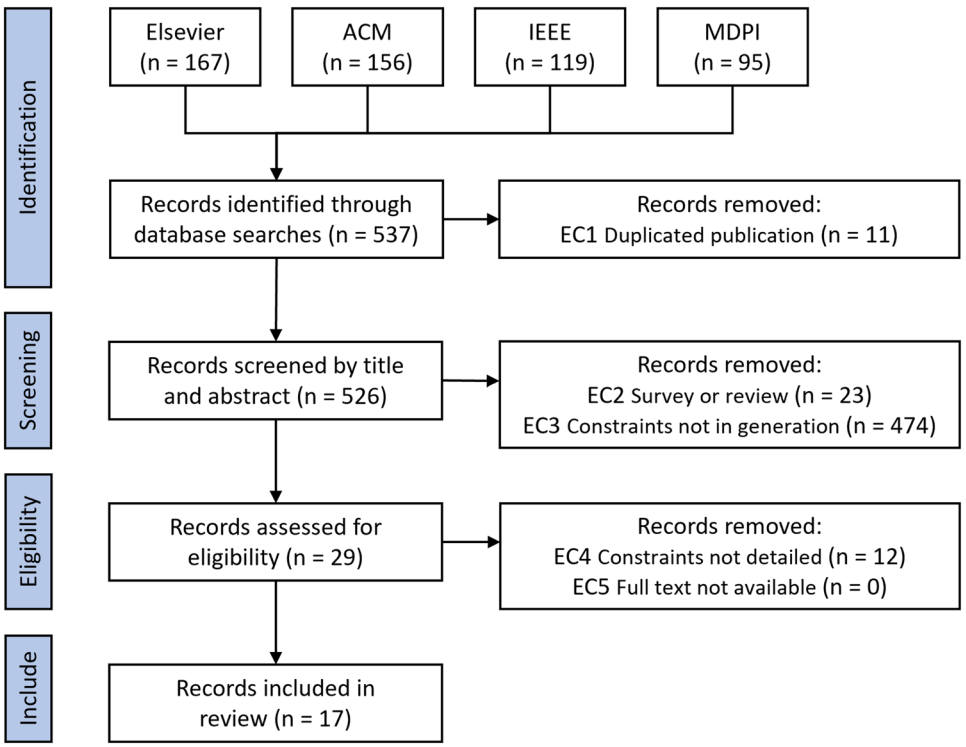

Fig. 2 PRISMA search process for RQ2

Considering the different domain constraints that different systems have, this review has shown that although random generation of data may be capable of achieving reasonable test path coverage of simpler paths, it is highly unpractical on more complex ones. Complex functionalities usually contain more dependencies, making the combination of the right parameters extremely complex. Moreover, the literature shows that in a black-box context, the input space for certain data types is continuous, and therefore different techniques such as providing boundary values [152–154], leveraging system feedback [152, 155, 156] and applying search techniques can reduce computational complexity [157].

In a grey-box setting, the authors have also attempted to reduce the input space by resorting to specifications that reflect source code knowledge to constraint the input [158] and techniques for quick exploration of the input space [159]. In a white-box setting, automated testing can be interpreted as a search problem, with most existing methods focusing on metaheuristics and their optimization to find more diverse test cases [160–165]. One publication has attempted to use GANs for software testing [46], but like many white-box approaches, it only focuses on numerical input generation, which may be insufficient to fully test all the inputs of modern software applications and information systems.

The gathered publications were analyzed to identify the test scope, the main input generation methods, the covered data types and the utilized data sources. This information is summarized in Table 4. The keyword"All"refers to numerical, textual, byte, boolean, and composite data types.

Regarding the methods for input generation from a black-box testing perspective, since there is no knowledge regarding the SUT, authors have found ways of constraining the input generation using other available data. For instance, Martin-Lopez et al. [152] decided to test a Representational State Transfer (REST) Application Programming Interface (API). Constrained input was produced via three different methods: (i) OpenAPI Specifications (OAS), which establishes the baseline information to interact with an API, and may include request examples, (ii) custom data generators which produce constrained input, and (iii) public knowledge bases. Similarly, in [154], the authors propose using the data collected from a public API and mutate faulty input to produce test cases. Even though both approaches are capable of producing realistic test data, they rely on publicly available data, which might not be sufficient to cover the entirety of the SUT test paths.

Recent works have also considered the use of model-based approaches to solve the search problem by producing a model of the SUT capable of generating constrained input. For instance, Martin-Lopez [153] resorts to three different methods for testing web APIs: (i) search-based methods, (ii) mutating API JSON input and output, and (iii) computing metamorphic testing. The goal of the authors is to not diverge from the initial input data, preserving its quality by using artificial intelligence and generative algorithms to produce constrained test input. Walkinshaw et al. [155] propose a method that combines Learning-based Testing (LBT) and Query By Committee (QBC). QBC is used in this context to circumvent LBT's dependence between model inference algorithm and the test-generation algorithm. This mechanism

Table 4 Characteristics of constrained testing methods

Test scope	Method	Data types	Data sources
Black-box	Custom Data Generators [152]	All	OpenAPI specifications and knowledge
	Metamorphic and Search-based Testing [153]	All	Input/Output data and API configuration
	Custom Mutators [154]	All	Faulty real user input
	Test By Committee [155]	All	Custom specifications
	Reinforcement Learning [156]	Numerical	Random data
	Input Distance [157]	Numerical	Navigational model
Grey-box	Pairwise hybrid Artificial Bee Colony [158]	Numerical	Constraint specifications
	Locally Spreading, Points Algorithm, and Adaptive Random Test [159]	Numerical	Defect-free confidence interval
White-box	Genetic Algorithm and Reinforcement Learning [160]	Numerical	Control flow graph
	Genetic Algorithm [161–163]	Numerical	Random data
	Particle Swarm Optimization [164]	Numerical	Control flow graph
	D-Algorithm [166]	Numerical	Control flow graph and define-use chains
	Multi-objective Optimization Algorithm [167]	Numerical	Random data
	Coevolutionary Genetic Algorithm [165]	Numerical	Random data
	Generative Adversarial Networks [46]	Numerical	Gcov file

allows LBT to select inputs based on the combined uncertainty of the models, inferring a behavioural model of the SUT and selecting the test that it is least certain about.

Additionally, Kim et al. [156] reinterpreted this search problem as a reinforcement learning one. Despite achieving 100% branch coverage during testing, for unforeseen arbitrary functions it only achieved 60.06% branch coverage. Moreover, the algorithm only considers numeric input which depending on the SUT may render this approach unpractical for software testing. These approaches are as good as the amount of knowledge one has of the SUT, but as modern software and digital systems grow in size and complexity, so do the domain constraints and the number of branches that should be included in the configurations.

M. Biagiola et al. [157] produce a navigational model of a web application by crawling it to discover possible test cases. The presented method's first test is generated randomly and is added to a set of executed tests. The subsequent tests and concrete input vectors are selected depending on the distance between each candidate and the current set of executed test cases, where only the farthest case is computed. Their goal is to generate a set of test cases that diversify the coverage of the navigational graph. Even though they present a method that does not rely on exhaustive testing of the API that requires in-browser executions, they still rely on random input which dictates how long the system will take to find the next valuable input data to cover different test paths, since it is calculated by the distance formula. From a grey-box perspective, Alazzawi et al. [158] also tackle the same problem using a Pairwise hybrid Artificial Bee Colony algorithm, which takes random input and a configuration file with the constraints to find the most diversified test cases. However, depending on the manual configuration, the test cases can lack quality and diversity.

In a grey-box setting, Huang et al. [159] tackle the broad input space and constraint it by resorting to the combination of a Locally Spreading Points algorithm along with a code defect-free confidence interval. The algorithm receives multiple generated test cases and evolves them to improve the minimum distance between points, achieving a better spread set of test cases. The approach improves effectiveness when used as an add-on to Adaptive Random Testing (ART). However, the method requires manual labeling of the test cases, regarding the confidence interval.

Even though black-box and grey-box methods can be quite useful to test a system regardless of its characteristics, the findings show that most research works attempt to automate software testing from a white-box perspective interpreting input generation as a search-based problem and resorting to metaheuristics to perform the search. As such many authors have also used GA to constraint the test cases generation [160–163]. Esnaashari et al. [160] use the GA to solve the search-based along with reinforcement learning as a local search step within the GA. Their work is capable of achieving results faster, but it does not produce better coverage than other methods. In particular, Avdenkoo et al. [161] and Zhu et al. [162] focus on improving the genetic operators. Zhu et al. improve: (i) selection by resorting to symbolic execution technique to select test cases with more useful heuristic information, (ii) crossover by analyzing the coverage of the combination of test cases, and (iii) mutation by enforcing the modification of values according to specific constraints.

On the other hand, Avdeenko et al. focused on improving the fitness function of the algorithm and were able to achieve 100% code coverage after multiple generations. Despite the quality of their work, Zhu et al. only consider the generation of numeric input in an extremely simple code sample. Avdeenko et al. do not provide much detail regarding the generated input, but the tool was tested in a more complex code structure. Yao et al. [163] tackled the problem that is testing software with randomness and used GA along with a mathematical model capable of generating data according to a certain criterion, by observing the impact of random behavior on the software when given random input numbers and modifying them. In the white-box testing context, GAs are very popular. However, most approaches rely on generating random input and evolve it by testing iteratively the new input, using metrics such as the code coverage as fitness. Depending on the size of the software that is being tested, this approach may be just as unfeasible as the random approach.

Other authors have also approached the problem without necessarily resorting to the most traditional metaheuristic methods. For instance, Jaiswal et al. [164] decided to approach the problem using particle swarm optimization algorithm along with an improved fitness function that uses Control Flow Graph (CFG) to generate constrained input. This technique led to authors to achieve 100% code coverage in only two iterations. However, the method only considers numeric input and the code analysed lacks complexity. Therefore, further tests should be made to understand if this is indeed a viable solution for constrained input generation. Zhang et al. [166] propose a version of the D-algorithm for software capable of constraining the input generation process to directed local search. Their method requires a CFG of the code, as well as define-use chains, which represent where a variable is defined in the node and when it is used in the CFG.

Feng et al. [167] developed their own multi-objective optimization algorithm for path coverage-oriented test data generation. They start by randomly generating data and use it to discover the easy-to-cover paths. Then they perform mutations on the input data generated to discover the more complex paths. Dang et al. [165] attempt to automate and

reduce the search domain in the context of mutation testing, which is a fault-oriented software testing technique. Their work consists of a method for generating test data applying a dimensionality reduction on the search domain based on a mutant stubbornness. The authors approach this problem as an optimization one and decide to use a coevolutionary GA to perform the task of generating test data to kill mutants.

Guo et al. [46] took one step further and decided to use WGAN + Gradient Penalty (WGAN-GP) to produce constrained test data input. The authors focus on performing unit and integration testing. The WGAN-GP is trained on the execution path information to learn the behaviour of the SUT. The trained GAN is then capable of producing and selecting new test data to increase branch coverage. The GAN is trained with the execution path of test input using the Gcov tool. Gcov is a source code coverage analysis tool capable of counting the number of times each statement is a program is executed. The algorithm then produces new data that is used along with Gcov to generate execution path information, which is then fed into the GAN again. This way, the authors are capable of maximizing branch coverage. Compared to random testing, their work was capable of improving test coverage of 5 out of 7 functions in the unit testing.

Overall, the white-box testing approaches mostly resort to heuristics and perform iterative testing of the generated input against the SUT to check if the coverage increases. Exhaustive evaluation of all possibilities should be avoided, for it is very expensive when considering the complexity of modern systems. From a grey-box and black-box perspective, the approaches usually lack knowledge regarding the SUT and its constraints, which can lead to more time-consuming and resource-intensive testing. Automated testing tools could leverage adversarial methods like GANs capable of constraining the input more efficiently and reducing the search space. By testing specific parameter combinations tailored to the characteristics of each function, constrained adversarial data generation approaches could present significant improvements to these tools. Therefore, it is pertinent to perform further research and develop new methods and tools to improve the quality of the data utilized in automated software testing tools.

5 Conclusions

This work reviewed the recent scientific advances of constrained data generation in a methodical manner. The state-of-the-art methods for constrained data generation in the adversarial machine learning and automated software testing fields were analyzed, identifying research gaps and opportunities to transfer knowledge from adversarial attacks to enhance testing tools and improve the resilience and robustness of information systems.

Regarding the current testing tools, there are several challenges that hinder full automation. Recent works attempt to constraint the input that is generated by analyzing the SUT empirically or resorting to specifications, although most perform exhaustive search by mutating parameters in order to find different branches of the SUT that are yet to be tested, which is unpractical. This suggests it may be difficult to use adversarial learning for black-box and grey-box testing, which have more complex constraints that are not directly provided to the tool.

Since white-box testing is commonly performed during the development of a system to make it more secure, when developers have knowledge of the utilized APIs and architectures, this is the key aspect where constrained adversarial learning could be valuable. Even though most approaches were developed for general applications, they can potentially be adapted to generate high-quality data according to the constraints of the SUT. Adversarial examples could be used to further improve testing tools with different parameter combinations specifically crafted to deceive the tested functions and software components, reducing the time and the computational resources required to achieve a high test coverage in complex systems.

In the future, the use of adversarial machine learning methods and other artificial intelligence techniques to enhance software development and testing should be further explored. For instance, to perform white-box unit testing, natural language processing could be leveraged to analyze the code and extract its constraints to quickly adapt the adversarial attack methods for the relevant test cases and improve computer code coverage in a more efficient way. From a black-box perspective, it may be harder to have a fully accurate code coverage analysis, but fuzzy logic could be combined with adversarial reinforcement learning as an API behaviour learning technique to generate faulty test input according to the characteristics of different APIs, providing more adaptive software testing tools.

Author contributions Conceptualization, J.V. and I.P.; methodology, J.V., T.D. and T.F.; validation, E.M. and I.P.; investigation, J.V., T.D. and T.F.; writing, J.V., T.D. and T.F.; supervision, E.M.; project administration, I.P.; funding acquisition, I.P. All authors have read and agreed to the published version of the manuscript.

Funding This work was done and funded in the scope of the BEHAVIOR project (NORTE2030-FEDER-00576300 no. 14391). This work was also supported by UIDB/00760/2020.

Data availability Not applicable.

Declarations

Ethics approval and consent to participate Not applicable.

Consent for publication Not applicable.

Competing interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Commission European. Directorate-General for Communications Networks Content and Technology. A path to the digital decade. 2021. https://doi.org/10.2759/027045.
2. Vitorino J, Praça I, Maia E. Sok: Realistic adversarial attacks and defenses for intelligent network intrusion detection. Comput Secur. 2023. https://doi.org/10.1016/j.cose.2023.103433.
3. Agency European Union, for Cybersecurity, et al. ENISA Threat Landscape 2022. Tech rep. 2022. https://doi.org/10.2824/764318.
4. European Union Agency for Cybersecurity. Cyber Europe 2022: After Action Report. Tech Rep. 2022. https://doi.org/10.2824/397622.
5. Myers GJ, Sandler C, Badgett T. The art of software testing. 2012. https://doi.org/10.1002/9781119202486.
6. Vitorino J, Praça I, Maia E. Towards Adversarial Realism and Robust Learning for IoT Intrusion Detection and Classification. Ann Telecommun. 2023. https://doi.org/10.1007/s12243-023-00953-y.
7. Moher D, et al. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. Syst Rev. 2015;4:1. https://doi.org/10.1186/2046-4053-4-1.
8. Elsevier ScienceDirect (1 2024). https://www.sciencedirect.com/search
9. Institute of Electrical and Electronics Engineers Xplore Advanced (1 2024). https://ieeexplore.ieee.org/search/advanced
10. Association for Computing Machinery Digital Library (1 2024). https://dl.acm.org/search/advanced
11. Multidisciplinary Digital Publishing Institute (1 2024). https://www.mdpi.com/search
12. Fuentes Reyes M, Auer S, Merkle N, Henry C, Schmitt M. SAR-to-Optical Image Translation Based on Conditional Generative Adversarial Networks-Optimization Opportunities Limits. Remote Sens. 2019. https://doi.org/10.3390/rs11172067.
13. Wang X, Tan K, Du Q, Chen Y, Du P. CVA2E: A Conditional Variational Autoencoder With an Adversarial Training Process for Hyperspectral Imagery Classification. IEEE Trans Geosci Remote Sens. 2020. https://doi.org/10.1109/TGRS.2020.2968304.
14. Chen Z, Tong L, Qian B, Yu J, Xiao C. Self-Attention-Based Conditional Variational Auto-Encoder Generative Adversarial Networks for Hyperspectral Classification. Remote Sens. 2021. https://doi.org/10.3390/rs13163316.
15. Costa CJ, Tiwari S, Bhagat K, Verlekar A, Kumar KMC, Aswale S. Three-Dimensional Reconstruction of Satellite images using Generative Adversarial Networks, in: 2021 Int. Conf. on Technological Advancements and Innovations, 2021. https://doi.org/10.1109/ICTAI53825.2021.9673457.
16. Kim Y, Hong S. Deep Learning-Generated Nighttime Reflectance and Daytime Radiance of the Midwave Infrared Band of a Geostationary Satellite. Remote Sens. 2019. https://doi.org/10.3390/rs11222713.
17. Han W, Wang L, Feng R, Gao L, Chen X, Deng Z, Chen J, Liu P. Sample generation based on a supervised Wasserstein Generative Adversarial Network for high-resolution remote-sensing scene classification. Inf Sci. 2020. https://doi.org/10.1016/j.ins.2020.06.018.
18. Mizuochi H, Iijima Y, Nagano H, Kotani A, Hiyama T. Dynamic Mapping of Subarctic Surface Water by Fusion of Microwave and Optical Satellite Data Using Conditional Adversarial Networks. Remote Sens. 2021. https://doi.org/10.3390/rs13020175.
19. Cira C-I, Kada M, Manso-Callejo M-Á, Alcarria R, Bordel Sanchez B. Improving Road Surface Area Extraction via Semantic Segmentation with Conditional Generative Learning for Deep Inpainting Operations. ISPRS Int J Geo-Inf. 2022;11:1. https://doi.org/10.3390/ijgi11010043.
20. Zhang Y, Sun H, Zuo J, Wang H, Xu G, Sun X. Aircraft Type Recognition in Remote Sensing Images Based on Feature Learning with Conditional Generative Adversarial Networks. Remote Sens. 2018. https://doi.org/10.3390/rs10071123.

21. Hayatbini N, Kong B, Hsu K-L, Nguyen P, Sorooshian S, Stephens G, Fowlkes C, Nemani R, Ganguly S. Conditional Generative Adversarial Networks (cGANs) for Near Real-Time Precipitation Estimation from Multispectral GOES-16 Satellite Imageries-PERSIANN-cGAN. Remote Sens. 2019. https://doi.org/10.3390/rs11192193.
22. Pan X, Zhao J, Xu J. Conditional Generative Adversarial Network-Based Training Sample Set Improvement Model for the Semantic Segmentation of High-Resolution Remote Sensing Images. IEEE Trans Geosci Remote Sens. 2021. https://doi.org/10.1109/TGRS.2020.3033816.
23. Heller-Kaikov B, Pail R, Werner M. Signal Separation in Global, Temporal Gravity Data, in: Proc. of the 6th ACM SIGSPATIAL Int. Workshop on AI for Geographic Knowledge Discovery, New York, NY, USA. 2023. https://doi.org/10.1145/3615886.3627743.
24. Chen W, Li Y, Zhao Z. InsulatorGAN: A Transmission Line Insulator Detection Model Using Multi-Granularity Conditional Generative Adversarial Nets for UAV Inspection. Remote Sens. 2021. https://doi.org/10.3390/rs13193971.
25. Munawar HS, Ullah F, Heravi A, Thaheem MJ, Maqsoom A. Inspecting Buildings Using Drones and Computer Vision: A Machine Learning Approach to Detect Cracks and Damages. Drones. 2022. https://doi.org/10.3390/drones6010005.
26. Zhang J, Zhu M, Peng L, Generation Customized Parking Data, based on Multi-conditional GAN, in,. IEEE 23rd Int. Conf on Intell Transp Syst. 2020;2020. https://doi.org/10.1109/ITSC45102.2020.9294436.
27. Cheng H, Liao W, Yang MY, Rosenhahn B, Sester M. AMENet: Attentive Maps Encoder Network for trajectory prediction. ISPRS J Photogramm Remote Sens. 2021. https://doi.org/10.1016/j.isprsjprs.2020.12.004.
28. Jaafer A, Nilsson G, Como G, Augmentation Data, of IMU Signals and Evaluation via a Semi-Supervised Classification of Driving Behavior, in,. IEEE 23rd Int. Conf on Intell Transp Syst. 2020. https://doi.org/10.1109/ITSC45102.2020.9294496.
29. Li J, Zhao B, Wu K, Dong Z, Zhang X, Zheng Z. A Representation Generation Approach of Transmission Gear Based on Conditional Generative Adversarial Network. Actuators. 2021. https://doi.org/10.3390/act10050086.
30. Pöpperli M, Gulagundi R, Yogamani S, Milz S, Networks Realistic Ultrasonic Environment Simulation Using Conditional Generative Adversarial, in,. IEEE Intell. Vehicles Symp. 2019. https://doi.org/10.1109/IVS.2019.8814091.
31. Falahatraftar F, Pierre S, Chamberland S. A Conditional Generative Adversarial Network Based Approach for Network Slicing in Heterogeneous Vehicular Networks. Telecom. 2021. https://doi.org/10.3390/telecom2010009.
32. Kocayusufoglu F, Silva A, Singh AK. FlowGEN: A Generative Model for Flow Graphs, in: Proc. of the 28th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining, 2022. https://doi.org/10.1145/3534678.3539406.
33. Zhu Q-X, Hou K-R, Chen Z-S, Xu Y, He Y-L. Research and Application of Virtual Sample Generation Method Based on Conditional Generative Adversarial Network, in: 2021 China Automation Congress, 2021. https://doi.org/10.1109/CAC53003.2021.9728144.
34. Qin R, Zhao J. High-Efficiency Generative Adversarial Network Model for Chemical Process Fault Diagnosis. IFAC-PapersOnLine. 2022. https://doi.org/10.1016/j.ifacol.2022.07.531.
35. He Y-L, Li X-Y, Ma J-H, Lu S, Zhu Q-X. A novel virtual sample generation method based on a modified conditional Wasserstein GAN to address the small sample size problem in soft sensing. J Process Control. 2022. https://doi.org/10.1016/j.jprocont.2022.03.008.
36. Zhang L, Liu P, Wang L, Liu J, Song B, Zhang Y, He G, Zhang H. Improved 1-km-Resolution Hourly Estimates of Aerosol Optical Depth Using Conditional Generative Adversarial Networks. Remote Sens. 2021. https://doi.org/10.3390/rs13193834.
37. Alhajjar E, Maxwell P, Bastian N. Adversarial machine learning in Network Intrusion Detection Systems. Expert Syst with Appl. 2021. https://doi.org/10.1016/j.eswa.2021.115782.
38. Chernikova A, Oprea A. FENCE: Feasible Evasion Attacks on Neural Networks in Constrained Environments. ACM Trans: Priv. Secur; 2022. https://doi.org/10.1145/3544746.
39. Dina AS, Siddique AB, Manivannan D. Effect of Balancing Data Using Synthetic Data on the Performance of Machine Learning Classifiers for Intrusion Detection in Computer Networks. IEEE Access. 2022. https://doi.org/10.1109/ACCESS.2022.3205337.
40. Vitorino J, Oliveira N, Praça I. Adaptative Perturbation Patterns: Realistic Adversarial Learning for Robust Intrusion Detection. Future Int. 2022. https://doi.org/10.3390/fi14040108.
41. Wang J, Yan X, Liu L, Li L, Yu Y. CTTGAN: Traffic Data Synthesizing Scheme Based on Conditional GAN. Sensors. 2022. https://doi.org/10.3390/s22145243.
42. Kumar V, Sinha D. Synthetic attack data generation model applying generative adversarial network for intrusion detection. Comput Secur. 2023. https://doi.org/10.1016/j.cose.2022.103054.
43. Wan M, Yao H, Yan X. Generation of malicious webpage samples based on GAN, in: 2020 IEEE 19th Int. Conf. on Trust, Security and Privacy in Computing and Commun. 2020. https://doi.org/10.1109/TrustCom50675.2020.00116.
44. Liu Q, Yu G, Wang Y, Yi Z. A Novel DGA Domain Adversarial Sample Generation Method By Geometric Perturbation, in: 2021 3rd Int. Conf. on Advanced Inf. Sci. and System, 2021. https://doi.org/10.1145/3503047.3503080.
45. Gao X, Saha RK, Prasad MR, Roychoudhury A. Fuzz Testing Based Data Augmentation to Improve Robustness of Deep Neural Networks, in: Proc. of the ACM/IEEE 42nd Int. Conf. on Software Eng., 2020. https://doi.org/10.1145/3377811.3380415.
46. Guo X, Okamura H, Dohi T. Automated Software Test Data Generation With Generative Adversarial Networks. IEEE Access. 2022. https://doi.org/10.1109/ACCESS.2022.3153347.
47. Kasarapu S, Shukla S, Hassan R, Sasan A, Homayoun H, PD SM. CAD-FSL: Code-Aware Data Generation Based Few-Shot Learning for Efficient Malware Detection, in: Proc. of the Great Lakes Symp. on VLSI 2022, 2022. https://doi.org/10.1145/3526241.3530825.
48. Amrith V, S D, S SK, Vajipayajula S, Srinivasan K, Thangavel SK, Kumar TG. An early malware threat detection model using Conditional Tabular Generative Adversarial Network, in: 2023 14th Int. Conf. on Computing Commun. and Networking Technologies, 2023. https://doi.org/10.1109/ICCCNT56998.2023.10307903.
49. Chonwiharnphan P, Thienprapasith P, Chuangsuwanich E. Generating Realistic Users Using Generative Adversarial Network With Recommendation-Based Embedding. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.2976491.
50. Esmaili A, Farzi S. Effective synthetic data generation for fake user detection, in: 2021 26th Int. Computer Conf., Computer Society of Iran, 2021. https://doi.org/10.1109/CSICC52343.2021.9420570.
51. Eizagirre I, Segurola L, Zola F, Orduna R. Keystroke Presentation Attack: Generative Adversarial Networks for Replacing User Behaviour, in: Proc. of the 2022 European Symp. on Software Eng., New York, NY, USA, 2023. https://doi.org/10.1145/3571697.3571714.

52. Qian Y, Hu H, Tan T. Data augmentation using generative adversarial networks for robust speech recognition. Speech Commun. 2019. https://doi.org/10.1016/j.specom.2019.08.006.
53. Chen L, Liu Y, Xiao W, Wang Y, Xie H. SpeakerGAN: Speaker identification with conditional generative adversarial network. Neurocomputing. 2020. https://doi.org/10.1016/j.neucom.2020.08.040.
54. Ram SR, M VK, Subramanian B, Bacanin N, Zivkovic M, Strumberger I. Speech enhancement through improvised conditional generative adversarial networks. Microprocess Microsyst. 2020. https://doi.org/10.1016/j.micpro.2020.103281.
55. Ding Y-Y, Lin H-J, Liu L-J, Ling Z-H, Hu Y. Robustness of Speech Spoofing Detectors Against Adversarial Post-Processing of Voice Conversion. IEEE/ACM Trans Audio Speech Lang Process. 2021. https://doi.org/10.1109/TASLP.2021.3124420.
56. Yoon J, Drumright LN, van der Schaar M. Anonymization Through Data Synthesis Using Generative Adversarial Networks (ADS-GAN). IEEE J Biomed Health Inf. 2020. https://doi.org/10.1109/JBHI.2020.2980262.
57. Liu Z, Meng L, Tan Y, Zhang J, Zhang H. Image compression based on octave convolution and semantic segmentation. Knowl-Based Syst. 2021. https://doi.org/10.1016/j.knosys.2021.107254.
58. Khwaja AS, Anpalagan A, Venkatesh B. Smart Meter Data Masking Using Conditional Generative Adversarial Networks. Electr Power Syst Res. 2022. https://doi.org/10.1016/j.epsr.2022.108033.
59. Sun C, van Soest J, Dumontier M. Generating synthetic personal health data using conditional generative adversarial networks combining with differential privacy. J Biomed Inf. 2023. https://doi.org/10.1016/j.jbi.2023.104404.
60. Athey S, Imbens GW, Metzger J, Munro E. Using Wasserstein Generative Adversarial Networks for the design of Monte Carlo simulations. J Econom. 2021. https://doi.org/10.1016/j.jeconom.2020.09.013.
61. Sun H, Deng Z, Chen H, Parkes D. Decision-Aware Conditional GANs for Time Series Data, in: Proc. of the Fourth ACM Int. Conf. on AI in Finance, New York, NY, USA, 2023. https://doi.org/10.1145/3604237.3626855.
62. Fang D, Guan X, Hu B, Peng Y, Chen M, Hwang K. Deep Reinforcement Learning for Scenario-Based Robust Economic Dispatch Strategy in Internet of Energy. IEEE Int Things J. 2021. https://doi.org/10.1109/JIOT.2020.3040294.
63. Li J, Yang Y, Sun JS, Tomsovic K, Qi H. ConAML: Constrained Adversarial Machine Learning for Cyber-Physical Systems, in: Proc. of the 2021 ACM Asia Conf. on Computer and Commun. Security, 2021. https://doi.org/10.1145/3433210.3437513.
64. Song Z, Huang Y, Xie H, Li X. Generation Method of Multi-Regional Photovoltaic Output Scenarios-Set Using Conditional Generative Adversarial Networks. IEEE J Emerg Sel Topics Circ Syst. 2023. https://doi.org/10.1109/JETCAS.2023.3291145.
65. Zeng L, Sun M, Wan X, Zhang Z, Deng R, Xu Y. Physics-Constrained Vulnerability Assessment of Deep Reinforcement Learning-Based SCOPF. IEEE Trans Power Syst. 2023. https://doi.org/10.1109/TPWRS.2022.3192558.
66. Yin L, Zhao W. Graph attention-based U-net conditional generative adversarial networks for the identification of synchronous generation unit parameters. Eng Appl Artif Intell. 2023. https://doi.org/10.1016/j.engappai.2023.106896.
67. Gong X, Tang B, Zhu R, Liao W, Song L. Data Augmentation for Electricity Theft Detection Using Conditional Variational Auto-Encoder. Energies. 2020. https://doi.org/10.3390/en13174291.
68. Liu H, Li Z, Li Y. Noise Reduction Power Stealing Detection Model Based on Self-Balanced Data Set. Energies. 2020. https://doi.org/10.3390/en13071763.
69. Moon J, Jung S, Park S, Hwang E. Conditional Tabular GAN-Based Two-Stage Data Generation Scheme for Short-Term Load Forecasting. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.3037063.
70. Qin P, Wang X, Qiao Z, Li X, Hu Q, Shu W. GAN-based Residential Load Data Generation Model Considering Users' Privacy, in: 2022 7th Asia Conf. on Power and Electrical Eng., 2022. https://doi.org/10.1109/ACPEE53904.2022.9783676.
71. Zhang Y, Deng X, Zhang Y, Zhang Y. Generation of sub-item load profiles for public buildings based on the conditional generative adversarial network and moving average method. Energy Build. 2022. https://doi.org/10.1016/j.enbuild.2022.112185.
72. Zhang S, Ji Z, Zhang J, Bao Y, Wang W. Multi-dimensional Data Generation Method of Electric Vehicle Charging Behaviors Based on Improved Generative Adversarial Network, in: 2022 IEEE/IAS Industrial and Commercial Power System Asia, 2022. https://doi.org/10.1109/ICPSAsia55496.2022.9949855.
73. Bu X, Wu Q, Zhou B, Li C. Hybrid short-term load forecasting using CGAN with CNN and semi-supervised regression. Appl Energy. 2023. https://doi.org/10.1016/j.apenergy.2023.120920.
74. Ye Y, Huang Y, Li G, Zhang L, Rong X, Bie Z. Supply-demand scenario generation of active distribution systems using conditional generative adversarial networks. Energy Rep. 2023. https://doi.org/10.1016/j.egyr.2023.05.172.
75. Chang D, Yang W, Yong X, Zhang G, Wang W, Li H, Wang Y. Seismic Data Interpolation Using Dual-Domain Conditional Generative Adversarial Networks. IEEE Geosci Remote Sens Lett. 2021. https://doi.org/10.1109/LGRS.2020.3008478.
76. Hu F, Wu C, Shang J, Yan Y, Wang L, Zhang H. Multi-condition controlled sedimentary facies modeling based on generative adversarial network. Comput Geosci. 2023. https://doi.org/10.1016/j.cageo.2022.105290.
77. Xu R, Puzyrev V, Elders C, Fathi Salmi E, Sellers E. Deep semi-supervised learning using generative adversarial networks for automated seismic facies classification of mass transport complex. Comput Geosci. 2023. https://doi.org/10.1016/j.cageo.2023.105450.
78. Sun C, Demyanov V, Arnold D. A conditional GAN-based approach to build 3D facies models sequentially upwards. Comput Geosci. 2023. https://doi.org/10.1016/j.cageo.2023.105460.
79. Guérin É, Digne J, Galin É, Peytavie A, Wolf C, Benes B, Martinez B. Interactive Example-Based Terrain Authoring with Conditional Generative Adversarial Networks. ACM Trans Graph. 2017;36:6. https://doi.org/10.1145/3130800.3130804.
80. Zhang J, Li C, Zhou P, Wang C, He G, Qin H. Authoring multi-style terrain with global-to-local control. Gr Models. 2022. https://doi.org/10.1016/j.gmod.2021.101122.
81. Silva G, Domingues I, Duarte H, Santos JAM. Automatic Generation of Lymphoma Post-Treatment PETs using Conditional-GANs, in: 2019 Digital Image Computing: Techniques and Appl., 2019. https://doi.org/10.1109/DICTA47822.2019.8945835.
82. Teixeira JF, Dias M, Batista E, Costa J, Teixeira LF, Oliveira HP. Adversarial Data Augmentation on Breast MRI Segmentation. Appl Sci. 2021. https://doi.org/10.3390/app11104554.
83. Amirrajab S, Al Khalil Y, Lorenz C, Weese J, Pluim J, Breeuwer M. Label-informed cardiac magnetic resonance image synthesis through conditional generative adversarial networks. Comput Med Imag Gr. 2022. https://doi.org/10.1016/j.compmedimag.2022.102123.

84. Qiang N, Dong Q, Liang H, Li J, Zhang S, Zhang C, Ge B, Sun Y, Gao J, Liu T, Yue H, Zhao S. Learning brain representation using recurrent Wasserstein generative adversarial net. Comput Methods Progr Biomed. 2022. https://doi.org/10.1016/j.cmpb.2022.106979.
85. Jafaritadi M, Duan H, Chinn G, Moradi F, Levin CS. Multi-Tracer PET Image Synthesis in Prostate Cancer using Generative Deep Learning, in: 2023 IEEE Nuclear Sci. Symp., Medical Imaging Conf. and Int. Symp. on Room-Temperature Semiconductor Detectors, 2023. https://doi.org/10.1109/NSSMICRTSD49126.2023.10338614.
86. Jung E, Luna M, Park SH. Conditional GAN with 3D discriminator for MRI generation of Alzheimer's disease progression. Pattern Recognit. 2023. https://doi.org/10.1016/j.patcog.2022.109061.
87. Liu M, Zou W, Wang W, Jin C-B, Chen J, Piao C. Multi-Conditional Constraint Generative Adversarial Network-Based MR Imaging from CT Scan Data. Sensors. 2022. https://doi.org/10.3390/s22114043.
88. Jin Y, Chang W, Ko B. Generating Chest X-Ray Progression of Pneumonia Using Conditional Cycle Generative Adversarial Networks. IEEE Access. 2023. https://doi.org/10.1109/ACCESS.2023.3305994.
89. Garcia Hernandez A, Fau P, Wojak J, Mailleux H, Benkreira M, Rapacchi S, Adel M. Synthetic computed tomography generation for abdominal adaptive radiotherapy using low-field magnetic resonance imaging. Phys Imag Radiat Oncol. 2023. https://doi.org/10.1016/j.phro.2023.100425.
90. Mendes J, Pereira T, Silva F, Frade J, Morgado J, Freitas C, Negrão E, de Lima BF, da Silva MC, Madureira AJ, Ramos I, Costa JL, Hespanhol V, Cunha A, Oliveira HP. Lung CT image synthesis using GANs. Expert Syst Appl. 2023. https://doi.org/10.1016/j.eswa.2022.119350.
91. Li X, Zhang R, Wang Q, Duan X, Sun Y, Wang J. SAR-CGAN: Improved generative adversarial network for EIT reconstruction of lung diseases. Biomed Signal Process Control. 2023. https://doi.org/10.1016/j.bspc.2022.104421.
92. Hagad JL, Kimura T, Fukui K-I, Numao M. Learning Subject-Generalized Topographical EEG Embeddings Using Deep Variational Autoencoders and Domain-Adversarial Regularization. Sensors. 2021. https://doi.org/10.3390/s21051792.
93. Zhou X, Zhu X, Nakamura K, Noro M. Electrocardiogram Quality Assessment with a Generalized Deep Learning Model Assisted by Conditional Generative Adversarial Networks. Life. 2021. https://doi.org/10.3390/life11101013.
94. Karabulut SE, Khorasani MM, Pantanowitz A. Neurocartographer: CC-WGAN Based SSVEP Data Generation to Produce a Model toward Symmetrical Behaviour to the Human Brain. Symmetry. 2022. https://doi.org/10.3390/sym14081600.
95. Xia Y, Wang W, Wang K. ECG signal generation based on conditional generative models. Biomed Signal Process Control. 2023. https://doi.org/10.1016/j.bspc.2023.104587.
96. Qu Z, Shi W, Tiwari P. Quantum conditional generative adversarial network based on patch method for abnormal electrocardiogram generation. Comput Biol Med. 2023. https://doi.org/10.1016/j.compbiomed.2023.107549.
97. Biswas S, Chand P, Mathur A, Sinha R. Characterization of the event-related potentials during GAN-based generation of EEG signals and their data augmented subject classification, in: 2023 Int. Conf. on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies, 2023. https://doi.org/10.1109/REEDCON57544.2023.10151321.
98. Xie Y, Wan Q, Xie H, Xu Y, Wang T, Wang S, Lei B. Fundus Image-Label Pairs Synthesis and Retinopathy Screening via GANs With Class-Imbalanced Semi-Supervised Learning. IEEE Trans Med Imag. 2023. https://doi.org/10.1109/TMI.2023.3263216.
99. Ehrhart M, Resch B, Havas C, Niederseer D. A Conditional GAN for Generating Time Series Data for Stress Detection in Wearable Physiological Sensor Data. Sensors. 2022. https://doi.org/10.3390/s22165969.
100. Havaei M, Mao X, Wang Y, Lao Q. Conditional generation of medical images via disentangled adversarial inference. Med Image Anal. 2021. https://doi.org/10.1016/j.media.2021.102106.
101. Saffari N, Rashwan HA, Abdel-Nasser M, Kumar Singh V, Arenas M, Mangina E, Herrera B, Puig D. Fully Automated Breast Density Segmentation and Classification Using Deep Learning. Diagnostics. 2020. https://doi.org/10.3390/diagnostics10110988.
102. Gür A. Deep Feature Synthesis for Accurate Breast Cancer Prediction, in: 2022 Medical Technologies Congress, 2022. https://doi.org/10.1109/TIPTEKNO56568.2022.9960237.
103. Zhang F, Zhang Y, Zhu X, Chen X, Du H, Zhang X. PregGAN: A prognosis prediction model for breast cancer based on conditional generative adversarial networks. Comput Methods Programs Biomed. 2022. https://doi.org/10.1016/j.cmpb.2022.107026.
104. Inan MSK, Hossain S, Uddin MN. Data augmentation guided breast cancer diagnosis and prognosis using an integrated deep-generative framework based on breast tumor's morphological information. Inf Med Unlocked. 2023. https://doi.org/10.1016/j.imu.2023.101171.
105. Wu D, Zhang Y. A Data Enhancement Method for Gene Expression Profile Based on Improved Conditional SN-GAN, in: Proc. of the 2022 4th Int. Conf. on Robotics, Intelligent Control and Artificial Intelligence, New York, NY, USA, 2023. https://doi.org/10.1145/3584376.3584565.
106. Bailo O, Ham D, Shin YM. Red Blood Cell Image Generation for Data Augmentation Using Conditional Generative Adversarial Networks, in: 2019 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops, 2019. https://doi.org/10.1109/CVPRW.2019.00136.
107. Tasnadi E, Sliz-Nagy A, Horvath P. Structure preserving adversarial generation of labeled training samples for single-cell segmentation. Cell Rep Methods. 2023. https://doi.org/10.1016/j.crmeth.2023.100592.
108. Chui KT, Lytras MD, Vasant P. Combined Generative Adversarial Network and Fuzzy C-Means Clustering for Multi-Class Voice Disorder Detection with an Imbalanced Dataset. Appl Sci. 2020. https://doi.org/10.3390/app10134571.
109. Chen K, Zhu D, Lu J, Luo Y. An Adversarial and Densely Dilated Network for Connectomes Segmentation. Symmetry. 2018. https://doi.org/10.3390/sym10100467.
110. Yu Y, Tang B, Lin R, Han S, Tang T, Chen M, CWGAN: Conditional Wasserstein Generative Adversarial Nets for Fault Data Generation, in,. IEEE Int. Conf on Robotics and Biomimetics. 2019. https://doi.org/10.1109/ROBIO49542.2019.8961501.
111. Dixit S, Verma NK, Ghosh AK. Intelligent Fault Diagnosis of Rotary Machines: Conditional Auxiliary Classifier GAN Coupled With Meta Learning Using Limited Data. IEEE Trans Instrum Meas. 2021. https://doi.org/10.1109/TIM.2021.3082264.
112. Ahang M, Jalayer M, Shojaeinasab A, Ogunfowora O, Charter T, Najjaran H. Synthesizing Rolling Bearing Fault Samples in New Conditions: A Framework Based on a Modified CGAN. Sensors. 2022. https://doi.org/10.3390/s22145413.
113. Li Y, Zou W, Jiang L. Fault diagnosis of rotating machinery based on combination of Wasserstein generative adversarial networks and long short term memory fully convolutional network. Measurement. 2022. https://doi.org/10.1016/j.measurement.2022.110826.
114. Liu Y, Jiang H, Wang Y, Wu Z, Liu S. A conditional variational autoencoding generative adversarial networks with self-modulation for rolling bearing fault diagnosis. Measurement. 2022. https://doi.org/10.1016/j.measurement.2022.110888.

115. Peng Y, Wang Y, Shao Y. A novel bearing imbalance Fault-diagnosis method based on a Wasserstein conditional generative adversarial network. Measurement. 2022. https://doi.org/10.1016/j.measurement.2022.110924.
116. Wang X, Liu H. Data supplement for a soft sensor using a new generative model based on a variational autoencoder and Wasserstein GAN. J Process Control. 2020. https://doi.org/10.1016/j.jprocont.2019.11.004.
117. Chen C-H, Tsung C-K, Yu S-S. Designing a Hybrid Equipment-Failure Diagnosis Mechanism under Mixed-Type Data with Limited Failure Samples. Appl Sci. 2022. https://doi.org/10.3390/app12189286.
118. Guo X, Liu X, Królczyk G, Sulowicz M, Glowacz A, Gardoni P, Li Z. Damage Detection for Conveyor Belt Surface Based on Conditional Cycle Generative Adversarial Network. Sensors. 2022. https://doi.org/10.3390/s22093485.
119. Chen S, Jin D, He H, Yang F, Yang J. Deep Learning Based Online Nondestructive Defect Detection for Self-Piercing Riveted Joints in Automotive Body Manufacturing. IEEE Trans Ind Inf. 2023. https://doi.org/10.1109/TII.2022.3226246.
120. Zhao H, Zhang Z, Yang Y, Xiao J, Chen J. A Dynamic Monitoring Method of Temperature Distribution for Cable Joints Based on Thermal Knowledge and Conditional Generative Adversarial Network. IEEE Trans Instrum Meas. 2023. https://doi.org/10.1109/TIM.2023.3317485.
121. Wang Y-R, Sun G-D, Jin Q. Imbalanced sample fault diagnosis of rotating machinery using conditional variational auto-encoder generative adversarial network. Appl Soft Comput. 2020. https://doi.org/10.1016/j.asoc.2020.106333.
122. Liu X, Ma H, Liu Y. A Novel Transfer Learning Method Based on Conditional Variational Generative Adversarial Networks for Fault Diagnosis of Wind Turbine Gearboxes under Variable Working Conditions. Sustainability. 2022. https://doi.org/10.3390/su14095441.
123. Zhang L, Zhang H, Cai G. The Multiclass Fault Diagnosis of Wind Turbine Bearing Based on Multisource Signal Fusion and Deep Learning Generative Model. IEEE Trans Instrum Meas. 2022. https://doi.org/10.1109/TIM.2022.3178483.
124. Haque KN, Rana R, Schuller BW. High-Fidelity Audio Generation and Representation Learning With Guided Adversarial Autoencoder. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.3040797.
125. Li S, Sung Y. INCO-GAN: Variable-Length Music Generation Method Based on Inception Model-Based Conditional GAN. Mathematics. 2021. https://doi.org/10.3390/math9040387.
126. Chen L, Srivastava S, Duan Z, Xu C. Deep Cross-Modal Audio-Visual Generation, in: Proc. of the on Thematic Workshops of ACM Multimedia 2017, 2017. https://doi.org/10.1145/3126686.3126723.
127. Rajabi MM, Komeilian P, Wan X, Farmani R. Leak detection and localization in water distribution networks using conditional deep convolutional generative adversarial networks. Water Res. 2023. https://doi.org/10.1016/j.watres.2023.120012.
128. Jia C, Ma J. Conditional temporal GAN for intent-aware vessel trajectory prediction in the precautionary area. Eng Appl Artif Intell. 2023. https://doi.org/10.1016/j.engappai.2023.106776.
129. Aalaei M, Saadi M, Rahbar M, Ekhlassi A. Architectural layout generation using a graph-constrained conditional Generative Adversarial Network (GAN). Autom Constr. 2023. https://doi.org/10.1016/j.autcon.2023.105053.
130. Barsoum E, Kender J, Liu Z. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN, in: 2018 IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops, 2018. https://doi.org/10.1109/CVPRW.2018.00191.
131. Geng Z, Johnson D, Fedkiw R. Coercing machine learning to output physically accurate results. J Comput Phys. 2020. https://doi.org/10.1016/j.jcp.2019.109099.
132. Li Y, Zhang T, Duan L, Xu C. A Unified Generative Adversarial Framework for Image Generation and Person Re-Identification, in: Proc. of the 26th ACM Int. Conf. on Multimedia, 2018. https://doi.org/10.1145/3240508.3240573.
133. Marzouk A, Barros P, Eppe M, Wermter S. The Conditional Boundary Equilibrium Generative Adversarial Network and its Application to Facial Attributes, in: 2019 Int. Joint Conf. on Neural Networks, 2019. https://doi.org/10.1109/IJCNN.2019.8852164.
134. Chen X, Sun Y, Shu X, Li Q. Attention-aware conditional generative adversarial networks for facial age synthesis. Neurocomputing. 2021. https://doi.org/10.1016/j.neucom.2021.04.068.
135. Hassan AU, Memon I, Choi J. Real-time high quality font generation with Conditional Font GAN. Exp Syst Appl. 2023. https://doi.org/10.1016/j.eswa.2022.118907.
136. Utyamishev D, Partin-Vaisband I. Multiterminal Pathfinding in Practical VLSI Systems with Deep Neural Networks. ACM Trans Des Autom Electron Syst. 2023. https://doi.org/10.1145/3564930.
137. Belmonte-Hernández A, Hernández-Peñaloza G, Martín Gutiérrez D, Álvarez F. Recurrent Model for Wireless Indoor Tracking and Positioning Recovering Using Generative Networks. IEEE Sens J. 2020. https://doi.org/10.1109/JSEN.2019.2958201.
138. Tang H, Zhao Y, Wang G, Luo C, Wang W. Wireless Signal Denoising Using Conditional Generative Adversarial Networks, in: IEEE INFOCOM 2023 - IEEE Conf. on Computer Commun. Workshops, 2023. https://doi.org/10.1109/INFOCOMWKSHPS57453.2023.10225927.
139. Jahić B, Guelfi N, Ries B. Software Eng. for Dataset Augmentation using Generative Adversarial Networks, in: 2019 IEEE 10th Int. Conf. on Software Eng. and Service Sci., 2019. https://doi.org/10.1109/ICSESS47205.2019.9040806.
140. Kang L, Riba P, Rusiñol M, Fornés A, Villegas M. Content and Style Aware Generation of Text-Line Images for Handwriting Recognition. IEEE Trans Pattern Anal Mach Intell. 2022. https://doi.org/10.1109/TPAMI.2021.3122572.
141. Chu J, An D, Ma Y, Cui W, Zhai S, Gu XD, Bi X. WordGesture-GAN: Modeling Word-Gesture Movement with Generative Adversarial Network, in: Proc. of the 2023 CHI Conf. on Human Factors in Computing Syst., New York, NY, USA, 2023. https://doi.org/10.1145/3544548.3581279.
142. Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y. Generative Adversarial Nets. Adv Neural Inf Process Syst. 2014;27:1.
143. Kingma DP, Welling M. Auto-Encoding Variational Bayes, in: 2nd Int. Conf. on Learning Representations, 2014. https://doi.org/10.48550/ARXIV.1312.6114.
144. Mirza M, Osindero S. Conditional Generative Adversarial Nets, arXiv (2014). https://doi.org/10.48550/ARXIV.1411.1784.
145. Arjovsky M, Chintala S, Bottou L. Wasserstein Generative Adversarial Networks, in: D. Precup, Y. W. Teh (Eds.), Proc. of the 34th Int. Conf. on Machine Learning, Vol. 70, 2017.
146. Sohn K, Lee H, Yan X. Learning Structured Output Representation using Deep Conditional Generative Models. Adv Neural Inf Process Syst. 2015;28:3.

147. Martin-Lopez A, Arcuri A, Segura S, Ruiz-Cortés A. Black-box and white-box test case generation for restful apis: Enemies or allies?, 2021. https://doi.org/10.1109/ISSRE52982.2021.00034.
148. Rajamanickam L, Saat NABM, Daud SNB. Software testing The generation tools. Int J Adv Trends Comput Sci Eng. 2019. https://doi.org/10.30534/ijatcse/2019/20822019.
149. Zheng Y, Liu Y, Xie X, Liu Y, Ma L, Hao J, Liu Y. Automatic web testing using curiosity-driven reinforcement learning. 2021. https://doi.org/10.1109/ICSE43902.2021.00048.
150. Khan ME, Khan F. A comparative study of white box, black box and grey box testing techniques. Int J Adv Comput Sci Appl. 2012;3:6.
151. Anwar N, Kar S. Review paper on various software testing techniques. Glob J Comput Sci Technol. 2019;19:2.
152. Martin-Lopez A, Segura S, Ruiz-Cortés A. RESTest: Automated Black-Box Testing of RESTful Web APIs, in: Proc. of the 30th ACM SIGSOFT Int. Symp. on Software Testing and Analysis, 2021. https://doi.org/10.1145/3460319.3469082.
153. Martin-Lopez A. AI-Driven Web API Testing, in: Proc. of the ACM/IEEE 42nd Int. Conf. on Software Eng., 2020. https://doi.org/10.1145/3377812.3381388.
154. Haraldsson SO, Woodward JR, Brownlee AIE. The Use of Automatic Test Data Generation for Genetic Improvement in a Live System, in: Proc. of the 10th Int. Workshop on Search-Based Software Testing, 2017.
155. Walkinshaw N, Fraser G. Uncertainty-Driven Black-Box Test Data Generation, in: 2017 IEEE Int. Conf. on Software Testing, Verification and Validation, 2017. https://doi.org/10.1109/ICST.2017.30.
156. Kim J, Kwon M, Yoo S. Generating Test Input with Deep Reinforcement Learning, in: Proc. of the 11th Int. Workshop on Search-Based Software Testing, 2018. https://doi.org/10.1145/3194718.3194720.
157. Biagiola M, Stocco A, Ricca F, Tonella P. Diversity-Based Web Test Generation, in: Proc. of the 2019 27th ACM Joint Meeting on European Software Eng. Conf. and Symp. on the Foundations of Software Eng., 2019. https://doi.org/10.1145/3338906.3338970.
158. Alazzawi AK, Rais HM, Basri S, Alsariera YA. PhABC: A Hybrid Artificial Bee Colony Strategy for Pairwise test suite Generation with Constraints Support, in: 2019 IEEE Student Conf. on Research and Development, 2019. https://doi.org/10.1109/SCORED.2019.8896324.
159. Huang X, Huang L, Zhang S, Zhou L, Wu M, Chen M. Improving Random Test Sets Using a Locally Spreading Approach, in: 2017 IEEE Int. Conf. on Software Quality, Reliability and Security, 2017. https://doi.org/10.1109/QRS.2017.13.
160. Esnaashari M, Damia AH. Automation of software test data generation using genetic algorithm and reinforcement learning. Exp Syst Appl. 2021. https://doi.org/10.1016/j.eswa.2021.115446.
161. Avdeenko TV, Serdyukov KE, Tsydenov ZB. Formulation and research of new fitness function in the genetic algorithm for maximum code coverage. Proced Comput Sci. 2021. https://doi.org/10.1016/j.procs.2021.04.194.
162. Zhu Z, Jiao L. Improving Search-Based Software Testing by Constraint-Based Genetic Operators, in: Proc. of the Genetic and Evolutionary Computation Conf., 2019. https://doi.org/10.1145/3321707.3321720.
163. Yao X, Gong D, Li B, Dang X, Zhang G. Testing Method for Software With Randomness Using Genetic Algorithm. IEEE Access. 2020. https://doi.org/10.1109/ACCESS.2020.2983762.
164. Jaiswal U, Prajapati A, Generation Optimized Test Case, for Basis Path Testing Using Improved Fitness Function with PSO, in,. Thirteenth Int. Conf on Contemporary Computing. 2021;2021. https://doi.org/10.1145/3474124.3474197.
165. Dang X, Yao X, Gong D, Tian T. Efficiently Generating Test Data to Kill Stubborn Mutants by Dynamically Reducing the Search Domain. IEEE Trans Reliab. 2020. https://doi.org/10.1109/TR.2019.2922684.
166. Zhang J, Gupta SK, Halfond WGJ. A New Method for Software Test Data Generation Inspired by D-algorithm, in: 2019 IEEE 37th VLSI Test Symp., 2019. https://doi.org/10.1109/VTS.2019.8758641.
167. Feng X, Ding R, Chai B, Huo T. Multi-Objective Heuristic Information Optimization Algorithm for Path Coverage-Oriented Test Data Generation, in: 2021 3rd Int. Conf. on Artificial Intelligence and Advanced Manufacture, 2021. https://doi.org/10.1145/3495018.3495135.