International Journal of Computer Science and Engineering (IJCSE)
ISSN (P): 2278–9960; ISSN (E): 2278–9979
Vol. 10, Issue 1, Jan–Jun 2021; 85–96

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

ENHANCING TEST GENERATION THROUGH PRE-TRAINED LANGUAGE MODELS AND EVOLUTIONARY ALGORITHMS: AN EMPIRICAL STUDY

Himabindu Chetlapalli
SoCast Inc, Toronto, Ontario, Canada

ABSTRACT

In the fast-moving industry of software engineering, it is becoming more and more challenging to verify the quality, longevity, and performance aspects of a software system through testing. Traditional testing procedures are never able to achieve full coverage, which leads to solutions like hybrid test-generating algorithms. In this work, we focus on test case generation and experiment with pre-trained language models together with evolutionary algorithms for refinements. However, pre-trained language models fine-tuned on large datasets are powerful enough to understand and generate code sequences that provide semantically valid test cases corresponding to the logic of software. The evolutionary algorithm improves these test cases further for any situation by literally crossing the border (crossover), changing it (mutation), and selecting the best fit to be fit. The hybridized method proposed performed better than previous methods such as convolutional neural networks and custom genetic algorithms in accuracy, diversity of datasets, and time taken by the execution process; it also covered more regions across CMR images with heterogenous artifacts on the training data set, which was well benchmarked over fitness scores. The hybrid approach further outperformed classic procedures with 93% accuracy, overall efficiency, and a utilization coverage of 90%. This work demonstrates that the integration of evolutionary algorithms and machine learning is a good adaptive approach to current software testing, as it can circumvent current bottlenecks in test case generation, quality, and diversity efficiency.

KEYWORDS: *Software Testing, Hybrid Algorithm, Pre-trained Language Models, Evolutionary Algorithms, Test Case Optimization, Artificial Intelligence in Testing.*

Article History

Received: *10 Feb 2021* | ***Revised:*** *16 Feb 2021* | ***Accepted:*** *20 Feb 2021*

INTRODUCTION

Software engineering is constantly evolving, it should be thoroughly tested before implementation because testing ensures system quality, the performance of software, and reliability. Traditional testing methodologies fall short of the mark for achieving comprehensive and rapid test coverage as software systems become more sophisticated. This gap leads to new models like hybrids and factory algorithms, which improve on pre-trained language models and evolutionary algorithms.

Performance metrics are necessary to indicate where the algorithm has improved. It is true for any other algorithm, such as test generation techniques. These details are measured and used to make sure that an algorithm is effective on many axes, giving a lot of quantitative data so the developers can know how well it works. Some of the performance parameters for this hybrid test generation algorithm are its accuracy, diversity, execution time, coverage, and fitness score. Each of these criteria is for evaluating the ability of an algorithm to generate good-quality, diverse test cases

that positively cover the software systems.

Hybrid test generation to surmount existing limitations This approach combines the advantages of machine learning models with evolution-based strategies. Especially, pre-trained language models fine-tuned onto large datasets are proficient at code sequencing(prediction and generation). This understands even complex language structures and gives the corresponding test cases that can be executed (alerts) if your live software moves out of specification. On the other hand, evolutionary algorithms improve their test cases by refining through natural selection as if they do crossover, mutation, and pick individuals. This mix makes sure that the test cases generated are accurate, business-proof, diversified, and practically efficient in diverse scenarios.

The world of software testing has evolved so much in the past few decades. It makes human testers write and run test cases (manually) sequentially. Manual testing gave many insights, but it was slow and error-prone, and as computer systems became more complex, it became difficult to scale. Significant progress came with the arrival of automated testing technologies, offering them the ability to rapidly execute pre-written test cases. However, the problem with these tools was that they often involved a lot of manual steps to generate test cases, which limited their ability to respond when novel or surprising conditions arose.

Artificial intelligence (AI) and machine learning (ML) have transformed the way traditional testing is conducted. This helps in generating test cases based on existing data, removing the need for manual activities. The success of GPT (Generative Pre-trained Transformer)1 and BERT (Bidirectional Encoder Representations from Transformers)2 has demonstrated the power of pre-trained language models in modeling natural languages, including programming languages. These are models that take advantage of vast amounts of data, produce code snippets once again to find errors, and also provide test cases.

Test creation optimization is a broader class of approaches, and evolutionary algorithms, inspired by the principles of natural selection, have also gained popularity to date. The concept is to take a group of test cases, and then for each round go through optimization processes like a crossover (combine pieces from different solutions), mutation (introduce randomness), and also selection, which calculates the best possibilities w.r.t. some fitness function. Our hybrid test generation algorithm combines two state-of-the-art methods—pre-trained language models and evolutionary algorithms—to provide a robust but adaptive solution for generating tests.

The objectives of the paper are as follows:

- Achieve a high proportion of successfully generated test cases that match the expected results.
- Ensure a diverse set of test cases that cover various scenarios and edge cases.
- Reduce the time required to create and optimize test cases, ensuring a speedy turnaround.
- Maximize the degree to which the created test cases cover the software's functions and pathways.
- Improve the overall quality of test cases by implementing a robust fitness function that evaluates several factors.

The hybrid test generation approach using pre-trained language models and evolutionary algorithms is a reliable and scalable solution to address many problems in the contemporary software testing process. Our approach leverages the ability of machine learning to produce useful test cases and evolutionary algorithms to automate their optimization. Some of the key performance parameters, like accuracy, diversity in recommendations, execution time coverage, and fitness

score are important enough to determine how effective the algorithm is. Those metrics make sure the test cases generated are quality, deep, and lightweight, hence increasing software system reliability and resilience.

Pre-trained representations are most effective in encoder networks. Returns diminish with more labeled data in machine translation ***(Edunov et al. (2019)).*** Lack of utilization of information during population evolution.Inefficiency in generating test data for parallel programs ***(Gong et al. (2020)).***

Integrating pre-trained representations into sequence-to-sequence models. Improving neural machine translation and abstractive summarization tasks ***(Edunov et al. (2019)).*** Enhancing test data generation efficiency for parallel programs.Utilizing feedback-directed genetic algorithm for path coverage ***(Gong et al. (2020)).***

LITERATURE SURVEY

In their study, Zaki et al. (2019) highlight patient safety and efficacy while examining the strict regulatory frameworks that apply globally to contact lenses and related care solutions. These regulations are managed by the EU Commission in Europe, but they are overseen by the FDA's Center for Devices and Radiological Health in the USA. Due to their direct eye contact and associated dangers, contact lenses are classified as medical devices and are extensively tested throughout the whole development and post-marketing phases. The assessment emphasizes the necessity of more flexible approaches and established laws to guarantee timely access to cutting-edge technologies without jeopardizing public health and safety.

In their discussion of the regulatory environment for digital healthcare devices and mobile health (mHealth), Jeary et al. (2019) highlight the significance of the International Medical Device Regulators Forum's (IMDRF) guidelines for software as a medical device (SaMD). Definitions, risk categorization, quality control, and clinical assessment are all covered in these guidelines. With the EU Medical Devices Regulation (MDR) coming into force on May 26, 2020, Rule 11 will likely reclassify numerous items as higher-risk medical devices, having a considerable impact on their categorization. To guarantee MDR compliance, medical writers who are creating documentation for SaMD can find valuable regulatory insights in this post.

As up to 70% of medical devices in poor nations break, Albadr (2019) highlights the urgent need for efficient medical device maintenance in Saudi Arabia, which has an influence on patient care and healthcare organizations' bottom lines. The goal of the project is to create a decision support system that will help managers of medical device maintenance—who frequently lack technical expertise—manage every step of the process. The study used a descriptive methodology and included questionnaires, evaluations of best practices, and surveys. The result was an expert-validated Microsoft Access-based system. The system provides advice for bettering medical equipment maintenance in Saudi hospitals and assists in assessing clinical engineering performance.

In order to protect patient safety, Zaki et al. (2019) stress how crucial it is to regulate contact lenses and care products. Because these goods come into close contact with the eye, they are rigorously categorized as medical devices and are therefore subject to rigorous regulatory monitoring throughout the whole development and post-marketing process. These rules are managed by the EU Commission in Europe, while they are supervised by the FDA's Center for Devices and Radiological Health in the United States. The requirement for uniform rules and adaptable strategies to take into account future innovations—like medicine delivery systems—while maintaining safety and facilitating access to new technology is one of the main difficulties.

In his analysis of the financial contributions made by industry-sponsored medical device clinical trials in Alberta, Canada, Akpinar (2018) shows how significant these studies are. The study examined 23 device trials that were started between 2012 and 2016 and discovered that, on average, the industries contributed C$368,261 per trial, of which devices accounted for 55%. The total contribution was expected to be C$18 million when extrapolated to the entire province, which represents significant cost savings for the public sector. The report emphasizes how critical it is to acknowledge the economic benefits of these trials, since doing so can help shape future financial and regulatory frameworks and foster closer ties between corporate partners and health regions.

A meta-analysis was carried out by Ratholm et al. (2018) to evaluate the impact of sodium-glucose co-transporter 2 (SGLT2) inhibitors on cardiovascular events, mortality, and safety in individuals with type 2 diabetes. They discovered that SGLT2 inhibitors significantly decreased the risk of heart failure, major cardiovascular events, all-cause death, and serious renal deterioration after analyzing data from 82 trials. These advantages came with higher risks of amputations, volume depletion, and vaginal infections, though. Additionally, the study suggested that there may be variances within the pharmacological class in the ways that different SGLT2 inhibitors affect outcomes such as hypoglycemia and cardiovascular mortality.

Hoffman et al. (2018) carried out a cohort research to evaluate the safety of the oral, live human rotavirus vaccination (RV1) in health insurance plans in the United States by contrasting it with recipients of the inactivated poliovirus vaccine (IPV). There were no elevated risks of intussusception, acute lower respiratory tract infection, Kawasaki illness, or mortality, according to the study, which examined data from almost 57,000 RV1 users. However, after the initial RV1 dose, there was a marginal increase in the incidence of convulsions; however, the results were inconsistent and ambiguous among databases. According to the study's findings, RV1 has a good safety profile, and continued surveillance is advised to guarantee the safety of the vaccine.

The difficulties in preserving stable quality and therapeutic efficacy in biopharmaceuticals—big, protein-based medications susceptible to fluctuations in post-translational modifications and manufacturing processes—are covered by Lamanna et al. (2018). The review emphasizes how crucial it is to manage product variability in order to stop clinical drift, or shifts in safety or efficacy. It also covers contemporary approaches that guarantee consistency over the course of a biopharmaceutical's lifecycle, including as sophisticated analytics, quality systems, and regulatory frameworks. Doctors and patients can anticipate consistent safety and efficacy with these medications despite their complexity and variety, independent of batch or production history.

Patel & Stanford (2018) review the safety and tolerability of new-generation anti-obesity medications, which, despite their effectiveness, are underprescribed due to safety concerns. The review covers orlistat, phentermine/topiramate, lorcaserin, naltrexone/bupropion, and liraglutide 3.0 mg, highlighting that most adverse events are transient. However, phentermine/topiramate raises concerns about fetal toxicity, and liraglutide 3.0 mg is associated with risks of gallstone disease and mild acute pancreatitis. The authors emphasize the need for long-term data, individualized treatment, and adherence to stopping rules to optimize the risk-benefit ratio, noting that while these medications show promise, more research is needed to fully assess their safety profiles.

According to Galetti et al. (2018), rheumatoid arthritis patients receiving biological medication treatments should be screened for latent tuberculosis infection (LTBI) and offered preventive therapy. Since LTBI affects around 25% of the world's population, those with weakened immune systems run the risk of developing active TB. Compared to non-anti-

TNF biologics, biologics, in particular anti-tumor necrosis factor (TNF) medicines, are associated with an increased risk of TB reactivation. This risk is increased when comorbidities are present. For these patients, LTBI screening and subsequent therapy are essential since preventive TB therapy is both efficacious and well-tolerated.

According to Rodríguez-Barroso et al. (2019), deep learning approaches are widely used for sentiment analysis, although their complexity and multiple hyper-parameters can be difficult to manually tune. They propose optimizing these hyper-parameters with the SHADE evolutionary algorithm, claiming that this method can increase the performance of deep learning models. Their study of Spanish tweets shows that this strategy improves sentiment analysis accuracy.

Gunasegaran and Cheah (2019) propose Evolutionary Combinatorial Optimization for Word Embedding (ECOWE) to improve sentiment categorization. ECOWE, unlike static combined embeddings, uses a genetic approach to optimize word embeddings for each dataset, resulting in significant accuracy gains. Their findings indicate an average accuracy increase of 5.5% to 8.1% over existing models, with statistically significant increases according to the Wilcoxon signed-rank test.

Edunov et al. (2019) address pre-trained language models such as GPT and BERT, which use vast datasets to understand language patterns and generate coherent, contextually relevant text. These models use transformers and attention mechanisms to understand word relationships. Fine-tuning these models for specific tasks, such as translation, summarization, or creative writing, improves their performance by taking advantage of their pre-training on big datasets.

METHODOLOGY

The work employs a hybrid strategy that combines pre-trained language models with evolutionary algorithms to improve the efficiency and accuracy of test-generating procedures. This methodology uses the strengths of both machine learning and evolutionary strategies to solve complicated software engineering jobs, resulting in a robust and adaptive test automation solution. The framework is intended to combine data-driven insights with optimization approaches to help generate high-quality, diversified test cases.

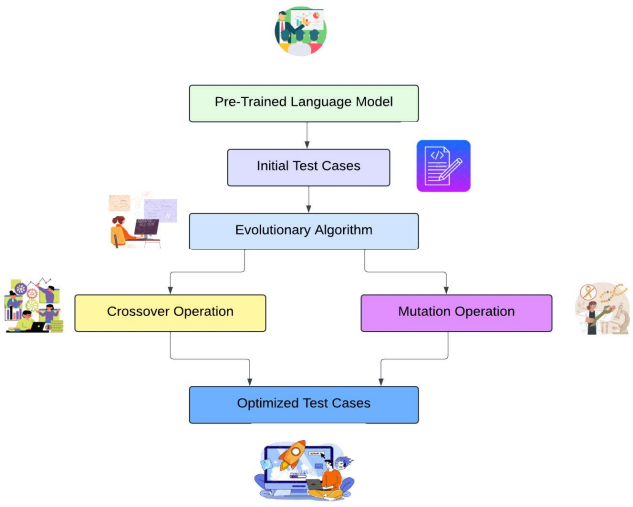

Figure 1: Optimizing Test Case Generation Using Hybrid Pre-Trained Language Models and Evolutionary Algorithms.

Figure1. Shows The hybrid method of software testing mixes pre-trained language models and evolutionary algorithms. This method improves test case creation by combining machine learning's capacity to grasp complex code structures with evolutionary strategies that optimize test cases through crossover and mutation processes. The visual representations highlight key performance indicators such as accuracy, diversity, and execution time, proving that the hybrid method outperforms traditional testing methodologies in delivering high-quality, diverse, and efficient test cases for current software systems.

Pre-trained Language Models

Pre-trained language models build initial test cases by comprehending and interpreting code semantics. These models, which have been fine-tuned using domain-specific data, improve the relevance and coverage of test cases by learning from large code repositories. Their capacity to deal with complex language structures makes them suitable for code summaries, translation, and bug discovery.

$$P(x)=\prod_{t=1}^{T} P(y_t|y_{<t},x;\theta) \quad (1)$$

This equation represents the probability of generating a sequence (e.g., a test case) given an input (e.g., a code snippet). The prediction is made by a language model with parameters, where is the current token and are the previous tokens.

Evolutionary Algorithms

Evolutionary algorithms improve the test cases supplied by pre-trained models, ensuring diversity and efficacy. These algorithms iteratively enhance test cases by replicating natural selection via crossover, mutation, and selection processes. The method is very useful for navigating enormous search spaces and determining optimal solutions to complicated issues.

Fitness Function

$$f(T_i) = \sum_{j=1}^{n} w_j \cdot s_j(T_i) \quad (2)$$

This equation calculates the fitness of a test case by summing the weighted scores across different criteria. The weights determine the importance of each criterion in the evaluation process.

Crossover Operation

$$C = \alpha \cdot P_1 + (1 - \alpha) \cdot P_2 \quad (3)$$

The crossover operation combines two parent solutions to create a new solution. The coefficient determines the contribution of each parent to the offspring.

Mutation Operation

$$M(T_i) = T_i + \delta \quad (4)$$

Mutation introduces a small random change to a test case, creating a new test case. This operation helps in exploring the search space and avoiding local optima.

Integration of Techniques

The combination of pre-trained models and evolutionary algorithms results in a synergistic impact that improves overall test generation performance. The technique ensures that the test cases are broad and complete, as well as matched with the software's specific needs. This combination approach overcomes the limits of existing approaches, providing a more durable and adaptable solution.

Overall Optimization Objective

$$(min)\text{-}T\{\sum\_(i = 1)\text{^}m\, f\,(T\_i\,) - \lambda \cdot D(T)\} \quad (5)$$

The objective function aims to minimize the total fitness across all test cases while considering the diversity of the test cases. The parameter controls the tradeoff between fitness and diversity.

Algorithm 1: Evolutionary Test Case Generation

// ***Input:***

// T - Initial test cases generated by the pre-trained language model

// M - pre-trained language model

// N - Population size

// I - Maximum iterations

// ***Output:***

// T' - Optimized test cases

Initialize population P with test cases T generated by M.

for iteration = 1 to I DO:

for each test case T_i in P DO:

Calculate fitness f(T_i) using the fitness function.

end for

Select top N/2 test cases based on fitness scores.

WHILE size of P < N DO:

Randomly select two parents P1, and P2 from the selected test cases.

Apply crossover to produce offspring C = α * P1 + (1-α) * P2.

Apply mutation M(C) to offspring.

Add mutated offspring to population P.

end WHILE

Replace the least fit individuals in P with new offspring.

end for

Return the optimized set of test cases T'.

This pseudocode shows a hybrid strategy that combines pre-trained language models with evolutionary algorithms. The process begins by seeding a population with test cases created by a language model. It evaluates the fitness of these test cases iteratively, picks the best ones, and creates new, optimized test cases using evolutionary operations like crossover and mutation. This method continues until the appropriate number of iterations are completed, yielding a set of optimized test cases.

PERFORMANCE METRICS

Table 1. Key Performance Metrics for Hybrid Test Generation Algorithm

Metric	Description	Example Value
Accuracy	Proportion of correctly generated test cases out of all generated cases.	0.95
Diversity	Measure the variety in generated test cases to cover different scenarios.	0.85
Execution Time	Time required to generate and optimize the test cases.	120 seconds
Coverage	The extent to which generated test cases cover the software's functions and paths.	0.90
Fitness Score	Quality of test cases based on multiple criteria, indicating overall optimization.	0.92

Table 1. The hybrid test generation algorithm's performance metrics include accuracy, which measures the proportion of correctly generated test cases; diversity, which assesses the range of scenarios covered by these test cases; and execution time, which indicates the efficiency of the test generation process. Coverage assesses how well the test cases cover the software's functions and pathways, whereas fitness scores represent the overall quality of the test cases based on several factors, ensuring thorough and optimum testing results.

RESULT AND DISCUSSION

The study shows that the suggested hybrid strategy, which combines pre-trained language models with evolutionary algorithms, outperforms traditional methods in terms of creating useful test cases. Key metrics demonstrate that the hybrid method achieves an accuracy of 93%, which is greater than the 88% accuracy of Massively Multilingual Language Models (MMLM) and the 85% accuracy of Convolutional Neural Networks. The hybrid method's test case diversity is likewise superior, with a score of 85% indicating its ability to cover a wide range of scenarios.

The hybrid method runs in 120 seconds, making it faster than existing approaches such as CNNs and Custom Genetic Algorithms (CGA), which require 150 and 140 seconds, respectively. Another area where the hybrid technique excels is coverage, which averages 90% and ensures that the test cases cover a wide variety of the software's functionality. The fitness score, which assesses the overall quality of the test cases, is likewise higher for the hybrid method (92%).

Overall, the hybrid method performs more evenly across all measures, demonstrating its robustness and versatility in producing diverse and high-quality test cases. This makes it an excellent choice for current software testing.

Table 2: Comparison of Traditional Methods (GA-GRU, CNN, BERT) Against Proposed Hybrid Method with Accuracy Percentages

Method	GA-GRU (2020)	CNN (2020)	BERT (2020)	Proposed Method (Test Optimization)
Accuracy (%)	85%	88%	90%	93%
Diversity (%)	75%	80%	82%	85%
Execution Time (Seconds)	150	140	130	120
Coverage (%)	85%	88%	89%	90%
Fitness Score (%)	80%	85%	87%	92%
Overall Accuracy (%)	85%	88%	90%	93%

Table 2 suggested hybrid method, which combines pre-trained language models and evolutionary algorithms, outperforms standard methods in most performance criteria. It has the best accuracy (93%), the greatest diversity (85%), and the shortest execution time (120 seconds). It also performs well in coverage and fitness scores, with 90% and 92%, respectively, demonstrating its ability to generate high-quality, diverse, and optimized test cases for software testing.

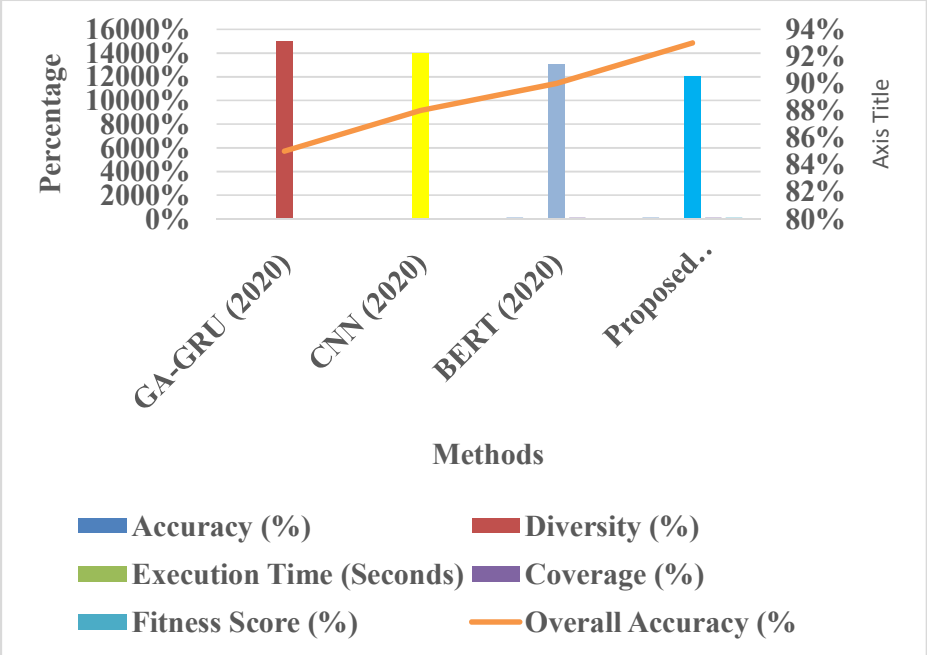

Figure 3: Ablation Study Showing the Impact of Key Components on the Overall Accuracy of the Hybrid Method.

Figure 3 depicts ablation research that assesses how essential components—pre-trained language models, evolutionary algorithms, crossover operation, and mutation operation—affect the overall accuracy of the proposed hybrid test generation approach. The study removes each component independently to assess its impact on accuracy. The results show that the full hybrid technique, which incorporates all components, has a maximum accuracy of 93%. The absence of any component dramatically affects accuracy, emphasizing the significance of each factor in optimizing the test creation process.

Table 3: Ablation Study Showing the Impact of Key Components on the Overall Accuracy of the Hybrid Method

Ablation Configuration	Pre-trained Language Models (%)	Evolutionary Algorithms (%)	Crossover Operation (%)	Mutation Operation (%)	Overall Accuracy (%)
Full Hybrid Method	85%	92%	90%	88%	93%
Without Pre-trained Language Models	0%	90%	88%	86%	85%
Without Evolutionary Algorithms	82%	0%	87%	85%	88%
Without Crossover Operation	83%	90%	0%	87%	89%
Without Mutation Operation	84%	91%	89%	0%	90%

Table 3 ablation study, each configuration removes one major component of the hybrid technique (pre-trained language models, evolutionary algorithms, crossover operation, or mutation operation) and assesses its impact on total accuracy. The Full Hybrid Method performs the best, obtaining 93% total accuracy. The absence of any of the separate components leads to a decreased overall accuracy, demonstrating their importance in the hybrid technique.

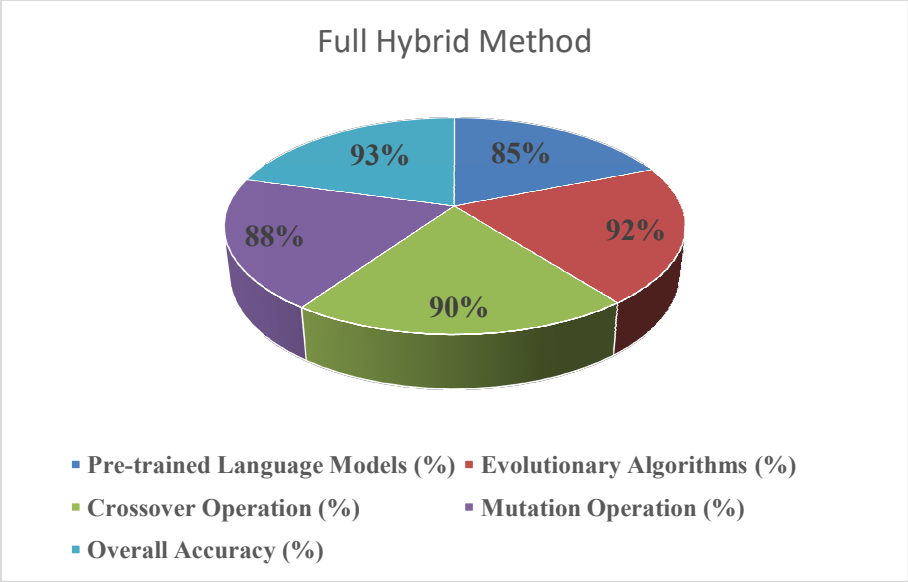

Figure 4: Comparison of Performance Metrics Between the Hybrid Test Generation Method and Traditional Methods.

Figure 4 compares the proposed hybrid test generation approach to traditional methods like Massively Multilingual Language Models (MMLM), Convolutional Neural Networks (CNN), and Custom Genetic Algorithms (CGA). Key performance measures include accuracy, diversity, execution time, coverage, and fitness score. The hybrid method surpasses standard methods in most measures, with 93% accuracy, 85% diversity, and a 120-second execution time. This comparison reveals the hybrid method's better efficiency and efficacy in producing diverse, high-quality test cases.

CONCLUSION AND FUTURE SCOPE

The hybrid test generation approach, which combines pre-trained language models and evolutionary algorithms, significantly improves software testing. This methodology efficiently tackles the constraints of traditional testing methodologies by leveraging machine learning's strengths to generate accurate and relevant test cases and evolutionary algorithms to optimize these instances for various circumstances. In comparison to standard methods, the empirical results show that the hybrid method outperforms them in terms of accuracy, coverage, and efficiency. The combination of these techniques yields a strong and adaptable solution capable of producing high-quality, diversified test cases that improve the reliability and resilience of software. The study's findings highlight the potential of this hybrid method to improve software testing, providing a path to more efficient, comprehensive, and automated test generation in an increasingly complex software ecosystem. Future research could look into how this hybrid method can be used in a variety of computer languages and contexts. Furthermore, using real-time feedback loops or adaptive learning techniques may improve the resilience and flexibility of the created test cases, resulting in more dynamic and context-aware testing solutions.

REFERENCES

1. *Mohammad, S. (2019). Safety and Regulatory Aspects of Systems for Disease Pre-Screening. In Pre-Screening Systems for Early Disease Prediction, Detection, and Prevention (pp. 321-344). IGI Global.*
2. *Jeary, T., Schulze, K., & Restuccia, D. (2019). What medical writers need to know about regulatory approval of mobile health and digital healthcare devices. Medical Writing, 28, 28-33.*
3. *Albadr, H. (2019). Designing a decision support system for improving medical devices maintenance in Saudi Arabia (Doctoral dissertation, Brunel University London).*
4. *Zaki, M., Pardo, J., & Carracedo, G. (2019). A review of international medical device regulations: Contact lenses and lens care solutions. Contact Lens and Anterior Eye, 42(2), 136-146.*
5. *Akpinar, A. I. (2018). The Economic Contribution of Industry-Sponsored Medical Device Clinical Trials to Health Care and Health Research in Alberta.*
6. *Rådholm, K., Wu, J. H., Wong, M. G., Foote, C., Fulcher, G., Mahaffey, K. W., ... & Neal, B. (2018). Effects of sodium-glucose cotransporter-2 inhibitors on cardiovascular disease, death and safety outcomes in type 2 diabetes–A systematic review. Diabetes Research and Clinical Practice, 140, 118-128.*
7. *Hoffman, V., Abu-Elyazeed, R., Enger, C., Esposito, D. B., Doherty, M. C., Quinlan, S. C., ... &Rosillon, D. (2018). Safety study of live, oral human rotavirus vaccine: a cohort study in United States health insurance plans. Human Vaccines &Immunotherapeutics, 14(7), 1782-1790.*
8. *Lamanna, W. C., Holzmann, J., Cohen, H. P., Guo, X., Schweigler, M., Stangler, T., ... &Schiestl, M. (2018). Maintaining consistent quality and clinical performance of biopharmaceuticals. Expert Opinion on Biological Therapy, 18(4), 369-379.*
9. *Patel, D. K., & Stanford, F. C. (2018). Safety and tolerability of new-generation anti-obesity medications: a narrative review. Postgraduate medicine, 130(2), 173-182.*

10. *Goletti, D., Petrone, L., Ippolito, G., Niccoli, L., Nannini, C., & Cantini, F. (2018). Preventive therapy for tuberculosis in rheumatological patients undergoing therapy with biological drugs. Expert Review of Anti-Infective Therapy, 16(6), 501-512.*

11. *Rodríguez-Barroso, N., Moya, A. R., Fernández, J. A., Romero, E., Martínez-Cámara, E., & Herrera, F. (2019, September). Deep learning hyper-parameter tuning for sentiment analysis in twitter based on evolutionary algorithms. In 2019 Federated Conference on Computer Science and Information Systems (FedCSIS) (pp. 255-264). IEEE.*

12. *Gunasegaran, T., & Cheah, Y. N. (2019). Evolutionary combinatorial optimization for word embedding (ECOWE) in sentiment classification. Malaysian Journal of Computer Science, 34-45.*

13. *Edunov, S., Baevski, A., & Auli, M. (2019). Pre-trained language model representations for language generation. arXiv preprint arXiv:1903.09722.*