Efficient Online Testing for DNN-Enabled Systems using Surrogate-Assisted and Many-Objective Optimization

Authors: Fitash Ul Haq, Donghwan Shin, Lionel Briand

ICSE 2022

Introduction

- Deep Neural Networks (DNNs) have revolutionized various fields by enabling machines to learn from data. They are a subset of artificial intelligence algorithms inspired by the structure and function of the human brain.
- When we say a system is "DNN-enabled," we mean that DNNs play a central role in its operation. These systems excel in tasks such as image recognition, natural language processing, and decision-making
- Examples of DNN-enabled systems include image recognition systems, autonomous vehicles, natural language processing applications, and many more.

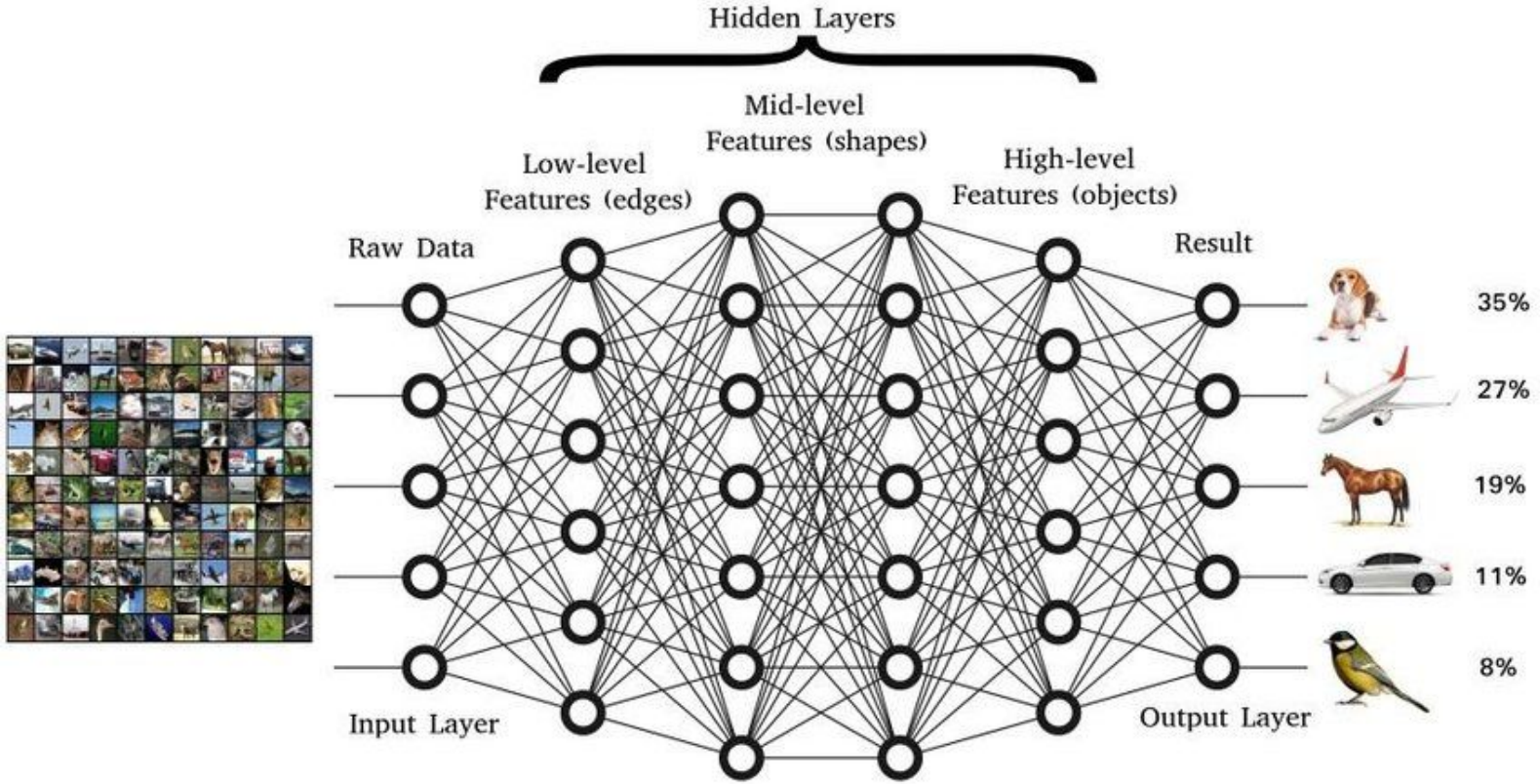


Image source: Sugiarto, Indar & Pasila, Felix. (2018). Understanding a Deep Learning Technique through a Neuromorphic System a Case Study with SpiNNaker Neuromorphic Platform. MATEC Web of Conferences. 164. 01015. 10.1051/matecconf/201816401015.

Motivation

The motivation behind the paper lies in the necessity to address the challenges of testing DNN-enabled systems in critical applications, with a focus on developing a more efficient and effective testing approach that can uncover safety violations and improve the overall reliability of these systems.

Testing DNN Enabled Systems (DES)

Offline Testing

Offline testing occurs in isolation without real-time interaction. The system is trained using a dataset with correct answers, adjusting parameters to minimize errors. It's then evaluated on a separate dataset to assess performance.

Online Testing

Online testing happens in real-time with continuous interaction. The system receives and processes inputs on-the-fly, and its performance is evaluated based on real-time responses, ensuring consistent accuracy over time.

Challenges of Online Testing

1. Too many safety requirements to consider at the same time
2. Computationally intensive to run high-fidelity simulator
3. Large test data

Proposed Solution

To achieve this, the paper introduces a new method called Surrogate-Assisted Many-Objective Testing Approach (SAMOTA).

This method combines different techniques to help generate a small set of diverse test scenarios that can reveal safety problems in the DNN-based systems. By using special models and advanced search methods, SAMOTA aims to make the testing process more effective and efficient.

Surrogate Models

- A surrogate model is a simplified or approximate version of a more complex model. It's like having a stand-in or proxy for the real thing.
- Surrogate models are often used in situations where the real model is too complex or time-consuming to use directly. They provide a way to make predictions or decisions quickly and efficiently, even if they're not as precise as the original model.

Image courtesy of CARLA

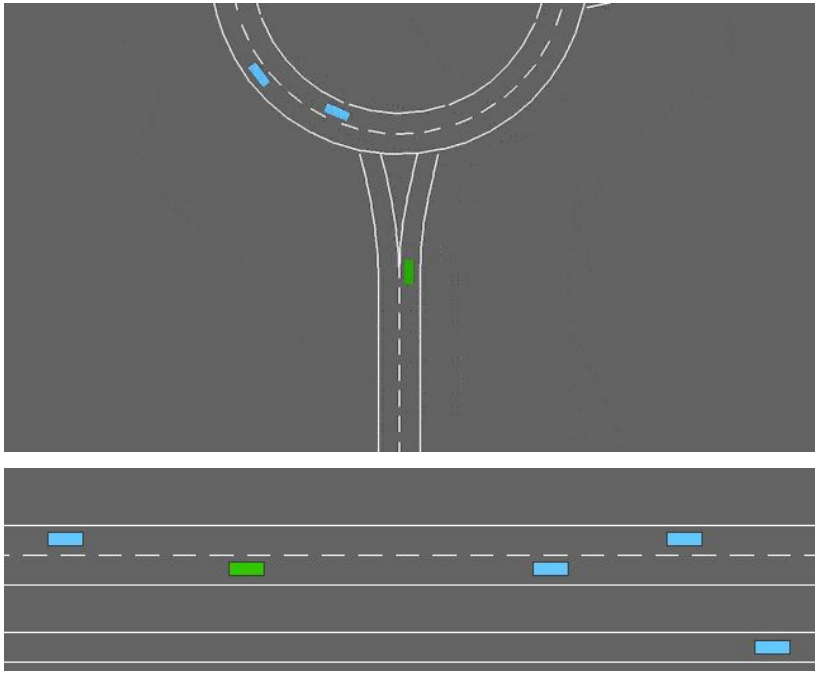


Image courtesy of Farama-Foundation from HighwayEnv (GitHub)

Many Objective Search

A many-objective search algorithm is a type of algorithm used to solve optimization problems that involve multiple conflicting objectives or criteria.

These algorithms search for solutions that represent the best possible trade-offs between the conflicting objectives.

For example, in designing a car, engineers may want to minimize fuel consumption, maximize speed, and minimize emissions all at the same time.

Local Search, Global Search

Local search is looking for something nearby without exploring the entire area. It focuses on finding the best solution in the immediate neighborhood of a given solution.

Global search is exploring the entire area to find the best solution. It aims to search the entire solution space systematically to find the globally optimal solution, which is the best possible solution across the entire search space.

Surrogate Model Types

1. **Kriging (KR)**

   Kriging is a statistical method that predicts values at unknown locations based on the values of nearby known locations, often used in spatial analysis and interpolation.

2. **Polynomial Regression with Clustering (PR):**

   Polynomial Regression with Clustering is a regression technique that fits a polynomial curve to the data points, with the addition of clustering to group similar data points before regression, often used in data analysis and predictive modeling.

3. **Radial basis Function network with Clustering (RBF):**

   Radial Basis Function network with Clustering is a type of neural network that uses radial basis functions as activation functions, along with clustering to group similar data points, often used in function approximation and pattern recognition.

4. **Ensemble**

   Ensemble methods combine multiple individual models to improve predictive performance, often used in machine learning to reduce variance, improve accuracy, and enhance robustness. Examples include Random Forest, Gradient Boosting, and Bagging.

Overview of SAMOTA

- SAMOTA employs global and local search phases using surrogate models to identify critical test cases for DNN-enabled systems efficiently.
- Global search explores the search space broadly, while local search refines promising regions for detailed analysis.
- Alternating between global and local searches, SAMOTA iteratively finds critical test cases.
- Test cases are evaluated using the actual DNN system to update surrogate models and guide subsequent searches.
- SAMOTA balances efficiency and effectiveness by combining surrogate models and iterative refinement."

RESEARCH QUESTIONS

RQ1: What is the best configuration for LS ?

RQ2: How do alternative approaches fare in terms of test effectiveness?

RQ3: How do alternative approaches fare in terms of test efficiency?

RQ1: Configuration for Local Search

The study, addressing Research Question 1 (RQ1), focuses on identifying the optimal configuration for Local Search (LS) in generating test cases. Local Search Effectiveness (LSE) is measured by actual fitness scores for all objectives. Different LS configurations, including surrogate model types and generation approaches, are considered for optimizing test case generation effectiveness.

The paper employs a specific formula to calculate LSE, aiming to assess the effectiveness of various LS configurations in generating test cases for specified search objectives, with a particular focus on DNN-enabled systems.

$$LSE(T) = \frac{1}{|O|} \sum_{o \in O} \frac{\max_{t \in T} f(t,o)}{|O|}$$

Where:

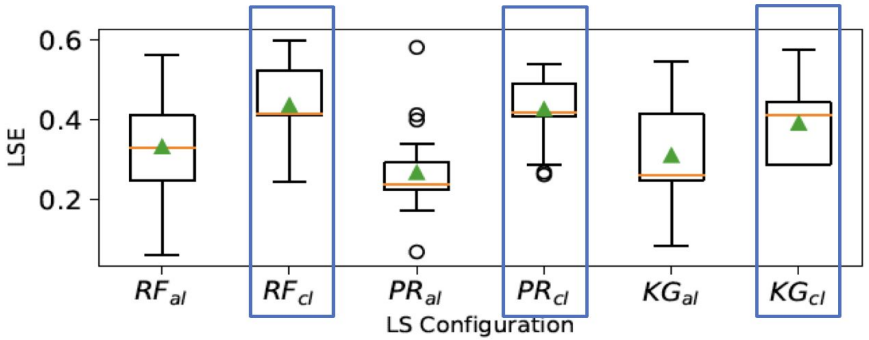
t - single test case

o - single objective

O - set of objectives

f(t, o) - fitness value of o in t

RQ1 : Results



The Figure shows the distribution of the LSE values for the six LS configurations.

The orange bar and the green triangle in the middle of each box represent the median and the average, respectively.

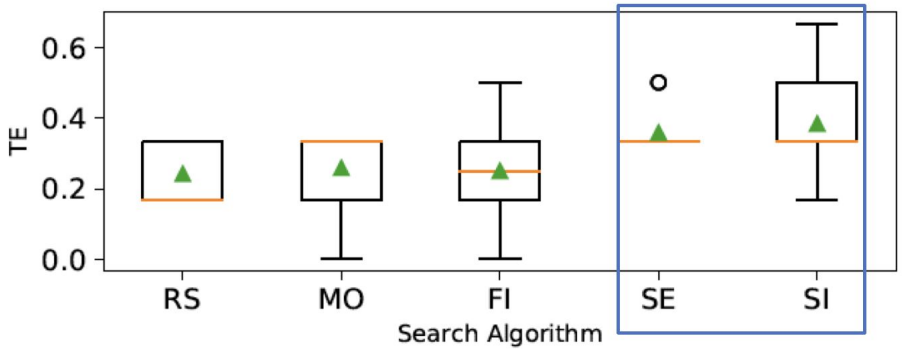
- Using our clustering-based approach (*cl*) for surrogate model generation is significantly better than the existing approach (*al*) in all cases.
- There is no significant difference overall between different surrogate model types.
- On average, RF_{cl} performs better than other surrogate model types.

RQ2: Test Effectiveness

Research Question 2 (RQ2) evaluates different test suite generation approaches, comparing SAMOTA with MOSA and FITEST. The focus is on assessing their effectiveness in uncovering safety violations within a specified time. Test Effectiveness (TE) measures the proportion of violated safety requirements, with higher TE values indicating better performance.

The study also explores SAMOTA's performance with and without an initial database (SAMOTA-I and SAMOTA-E) to understand how leveraging historical data influences accuracy and efficiency in detecting safety violations. This investigation highlights the significance of using historical data to enhance the overall effectiveness of test case generation.

RQ2: Results



The Figure shows the distribution of TE values achieved by RS, MOSA (MO), FITEST (FI), SAMOTA-E (SE), and SAMOTA-I (SI) over 20 runs. Again, the orange bar and the green triangle in the middle of each box represent the median and average, respectively.

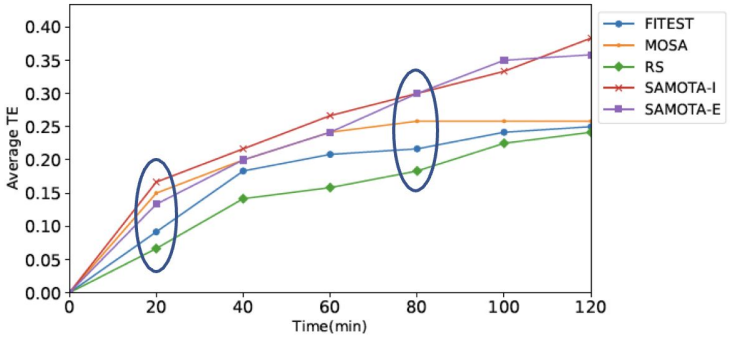
- SAMOTA variants are significantly more effective than other many-objective search algorithms tailored for test suite generation andrandom search with archive.
- Furthermore, SAMOTA can achieve acceptable test effectiveness without an initial database.

RQ3 : Test Efficiency

Research Question 3 (RQ3) assesses the efficiency of test suite generation approaches (SAMOTA, MOSA, FITEST, and Random Search). The study measures each approach's time to achieve specific levels of Test Effectiveness (TE) in detecting safety violations. TE values range from 1/6 to 6/6 (representing safety requirements).

Additionally, the impact of an initial database on SAMOTA's efficiency is explored by comparing SAMOTA with an empty database (SAMOTA-E) and SAMOTA with an initial database (SAMOTA-I).
This analysis aims to understand how historical data influences the speed of safety violation detection, highlighting the efficiency and effectiveness of SAMOTA compared to other methods within practical time constraints.

RQ3 : Results



The Figure shows the relationship between the execution time and the average TE values for 20 runs across all approaches.

For example, RS is always at the bottom, meaning that, on average, RS achieves the lowest
TE values compared to the others over the same time period.

- SAMOTA is more efficient than alternative test suite generation approaches as soon as its SMs become sufficiently accurate.
- An initial database can boost the efficiency of SAMOTA in the initial search phase and allow it to surpass other techniques right from the start.

Novelty of the Proposed Solution

- Novel Approach: SAMOTA integrates surrogate-assisted optimization with many-objective search algorithms for efficient test suite generation in DNN systems.
- Adaptive Surrogate Models: SAMOTA dynamically adjusts surrogate models to enhance scalability and adaptability across varying computational resources and complexities.
- Clustering-Based Local Surrogates: SAMOTA employs a novel clustering approach for local surrogate model generation, improving accuracy while minimizing computational overhead.
- Real-World Validation: Empirical evaluation on Pylot using CARLA demonstrates SAMOTA's effectiveness in detecting safety violations, validating its relevance in DNN system testing.

Assumptions made

- Surrogate Model Accuracy: Assumes accuracy of models like KriGing, Polynomial Regression, and Radial Basis Function network to guide search effectively towards safety violation test cases.
- SAMOTA Effectiveness: Assumes SAMOTA's superiority over alternative algorithms for DNN-based system testing.
- Simulation Environment: Assumes reliability of simulation environments like Pylot and CARLA for online testing, including accurate representation of real-world scenarios.
- Generalizability: Assumes findings from Pylot and CARLA case study can be extended to similar DNN-based systems and simulation environments.

Limitations of the Proposed Solution

- Limited Scope of Evaluation: Primarily evaluated in a specific case study with Pylot and CARLA, caution needed when applying findings to other domains or systems.
- Computational Overhead: Despite using surrogate models to reduce costs, SAMOTA may still demand significant computational resources, especially for complex systems or large-scale optimizations.
- Interpretability and Transparency: SAMOTA's black-box approach may hinder understanding and diagnosing issues in generated test data.
- Potential Bias in Test Data: Automated test data generation in SAMOTA may introduce biases if training data is not fully representative.

Practical Significance of the Proposed Solution

- Enhanced Safety Assurance: SAMOTA enhances safety assurance in real-world applications, especially in critical domains like automated driving, by efficiently identifying safety violations and improving system reliability.
- Cost Reduction in Testing: SAMOTA reduces testing costs by leveraging surrogate models to minimize computational resources, making it suitable for organizations with budget constraints.
- Accelerated Development Cycles: SAMOTA accelerates development cycles by automating test scenario generation and optimizing test effectiveness, facilitating faster time-to-market for new technologies.

Discussion Points

1. In the context of the paper's focus on online testing, how could SAMOTA be adapted to dynamically adjust its parameters and strategies based on real-time feedback from the system under test?
2. Given the rapid evolution of DNN technologies, what challenges might SAMOTA face in adapting to large-scale systems with intricate architectures and evolving functionalities?
3. Given the importance of regulatory compliance in safety-critical domains, what steps should be taken to ensure that SAMOTA aligns with industry standards and regulatory requirements, and how can industry collaboration facilitate this alignment?