IVVES (Industrial-Grade Verification and Validation of Evolving Systems)

Pekka Aho[2], Tanja E. J. Vos[1,2], Otto Sybrandi[3], Sorin Patrasoiu[4], Joona Oikarinen[4], Olivia Rodriguez Valdes[2] and Lianne V. Hufkens[2]

[1]*Universitat Politècnica de València (UPV), Spain*
[2]*Open Universiteit (OU), The Netherlands*
[3]*Marviq B.V., The Netherlands*
[4]*F-Secure, Finland*

Abstract

An increasing number of information systems are based on machine learning (ML) or artificial intelligence (AI). In some cases, the systems are adapting their behaviour during operation based on the data being gathered. This introduces new challenges for verification, validation and software testing. The traditional way of testing the systems during the development and before the deployment does not suffice anymore. IVVES (Industrial-Grade Verification and Validation of Evolving Systems) project aims to address these challenges by researching and developing methods to test ML and AI solutions and evolving systems, and using AI and ML to improve and automate development and testing. We summarise the results of the project at a high level, and provide more details on the research and collaboration related to scriptless end-to-end testing through graphical user interface.

Keywords

software testing, evolving systems, artificial intelligence, machine learning

1. Introduction

Artificial Intelligence (AI) and Machine Learning (ML) are enabling technologies disrupting innovation in all industrial domains by redefining approaches for information processing, decision making, automation and system engineering. The footprint of ML-based applications will dramatically increase in the coming years. IVVES project unites companies from the most relevant industrial domains in Europe to boost mutual learning in applying AI in their businesses and products in these competition-critical areas, and covers the industrial sectors of Transportation, Banking and Finance, Healthcare, Industrial Automation, and Cybersecurity.

Use of complex, evolving systems (ES), i.e., systems that rapidly change, either due to fast iteration cycles in development or caused by their capability to self-adapt and learn, will grow significantly in automation, computation and novel digital services. This includes mission- and safety-critical functions for transportation, financial markets, medicine, and energy. While the criticality of the ES demands rigorous, comprehensive and trustworthy quality assurance, both

Joint Proceedings of RCIS 2022 Workshops and Research Projects Track, May 17-20, 2022, Barcelona, Spain
pah@ou.nl (P. Aho); tvos@dsic.upv.es (T. E. J. Vos); otto.sybrandi@marviq.com (O. Sybrandi)
0000-0002-9252-1799 (P. Aho); 0000-0002-6003-9113 (T. E. J. Vos); 0000-0002-7824-2320 (O. Sybrandi); 0000-0002-7562-8199 (O. R. Valdes)


before and after deployment, the sheer size and complexity of these systems, high innovation dynamics and run-time learning and adaptation require completely new development and quality assurance approaches. Targeting the challenges in verification and validation of ES, IVVES will systematically develop AI-based approaches for robust and comprehensive, industrial-grade verification and validation of "embedded AI", i.e., ML for control of complex, mission-critical evolving systems and services covering the major industrial domains in Europe.

IVVES project will develop the approaches in three directions to cover the three main aspects in quality development for ES:

- Verification and validation approaches dedicated to ML-enabled applications in ES, including innovation in ML-based testing, transparent and robust assessment of ML models and trained networks, and metrics for quality and trustworthiness of data used for ML training.
- AI and data-driven verification and validation techniques dedicated to ES to cover areas that currently cannot be covered by state-of-the-art specification-based testing. This includes ML-based test case development and test automation, and continuous and run-time testing to detect quality degradation of ES during operation.
- Smart engineering approaches that use data analytics to establish efficient and high-quality engineering processes for ES to improve coverage of testing in simulated environments prior to initial end-user exposure.

All IVVES methods, techniques and tools will be driven and evaluated by the case studies representing relevant industrial domains: Transportation, Banking and Finance, Healthcare, Industrial Automation, Cybersecurity. To conclude, IVVES will develop cross-domain solutions with broad applicability that will be the foundation for standardisation and certification. Thus, IVVES will shape a breakthrough in innovation power for European industry in AI-based systems and applications.

2. IVVES project

IVVES (Industrial-Grade Verification and Validation of Evolving Systems) is an ITEA[1] project, consisting of 26 partners from 5 countries, running 3 years during 2019-2022. The technical work of the project focuses on the following three topics (work packages):

- Validation techniques for ML, including model quality, training data quality, and testing techniques for ML.
- Validation techniques for complex evolving systems, including ML-driven testing, testing with uncertainties, and online testing and monitoring.
- Data-driven engineering, including data collection techniques, instrumentation, and smart probes, pattern recognition for predictive maintenance and fault analysis, and data analytics in engineering and operation.

More details can be found from the project website (https://ivves.eu/).

[1] https://itea4.org/

2.1. Objectives

To leverage the quality assurance of ES and to kick-start a European market for respective verification and validation tools and services, IVVES will pursue the following technical objectives:

- Enable rigorous verification and validation means to assess ES over the complete system life cycle. Efficient and effective test and release strategies for ML-based applications and ES include AI-based and other data-driven or search-based verification and validation approaches. IVVES will develop methodologies based on industrial use cases in the domains: Automotive and Transportation, Banking and Finance, Healthcare, Telecom, Industrial Automation, Agriculture and Forestry, Cybersecurity.
- Develop strategies for rigorous data quality assurance through the definition of quality attributes, quality metrics and quality assurance procedures for data and data sets to extend the notion of system quality in ES by covering the dependency between data quality and trustworthiness.
- Providing acceptance procedures dedicated to ES by addressing uncertainties that arise from the autonomy of systems and their ability to learn in order to support risk-based acceptance and certification approaches.
- Providing tools, techniques and methods for the verification and validationof ES at runtime to enable continuous verification and validation of ES for different domains, including critical ones.
- Providing tools, techniques and methods to address the engineering challenges of ES with respect to the correlations among development and data artefacts throughout the entire ES lifecycle including both development and operation.
- Providing a platform for experimentation, tool and knowledge transfer, reaching the entire European industry. This platform facilitates building a community for the definition of new approaches and services based on IVVES results aiming for speeding the integration and exploitation of the IVVES technology in different industrial domains.

2.2. Expected outputs

Four main expected outcomes of IVVES project have been identified:

Outcome 1: Methods, techniques and tools for verification and validation of evolving systems (ES). These methods, techniques and tools will cover different kinds of testing and test automation approaches including model-based testing, search-based testing, fault-based testing and will cover all test relevant aspects for ES considering ML algorithms, data quality and different types of ES (e.g. highly iterative, adaptive, runtime evolving, etc.). The methods, techniques and tools are designed to cover the whole lifecycle of an ES. They can be used in different environments and in different domains, thus increasing their impact in the AI and systems engineering community. The IVVES methods, techniques and tools address: 1) general and fundamental testability of AI and the associated artefacts such as models, data, and features, and 2) the increase of automation, efficiency of testing and trustworthiness of test results in the context of the entire life cycle of an ES. The latter is achieved through the use of AI to extend and improve model-based and search-based testing techniques.

Outcome 2: Methods, techniques, patterns and tools for data analytics in engineering and operation. IVVES will provide techniques for data collection in engineering processes using non-intrusive probe agents, processing and analysis of data during engineering, monitoring and log analysis, extraction of diagnostics data, and identification of operational and behavioral patterns to support failure and anomaly detection throughout the entire product life cycle. Additionally, these techniques will be used to assess availability, reliability, and maintainability of a system, and provide improvement recommendations based on the identified patterns, e.g. as part of a decision support system.

Outcome 3: Pre-standardization methodology for data-driven engineering and the verification and validation of ES. The IVVES methodology will summarize all aspects of engineering ES as well as verification and validation of ES in a way that simplifies the adoption of the IVVES results by different industrial domains. It will summarize the methods, techniques, tools and processes being developed for engineering and verification and validation of ES in a publicly available documentation including test models, test patterns, risk patterns, test generation models, test coverage metrics and experience reports from the various case studies. This includes: 1) Risk and test patterns catalogue for ES: Risk patterns cover common faults and potential technical consequences while test patterns will provide guidance on how to test ML-based systems and ES in general. The catalogue will correlate risk and test pattern and will be instantiated for. 2) Validation models and techniques to support transparent ML: This comprises knowledge representation and reasoning, knowledge-based interpretability, validation and explanations of ML models (explainable AI). 3) (ML-based) generation models and techniques for testing ES: This includes test models and fault models for different kinds of testing (e.g. functional testing, testing extra-functional properties, run-time testing) as well as their general and domain-specific layout.

Outcome 4: Experimentation platform and training. The IVVES experimentation platform will demonstrate the overall applicability of the IVVES techniques and methods and show how the tools can improve the development and quality assurance process for ES. The provision of the platform will include: 1) An experience package to share experiences of adequate tools, testing techniques and methods, as well as the applicability of processes. It answers the question, why and where to apply the IVVES technologies. 2) A training package that helps exploitation partners and other industry stakeholders to apply IVVES outcomes into their development and quality assurance processes effectively. In the final phase of the project, IVVES will integrate the tools in the experimentation platform. Demonstrations will show how the methods and techniques are applied to the different case studies.

2.3. Relevance to RCIS

An increasing number of information systems are based on machine learning (ML) or artificial intelligence (AI). In some cases, the systems are adapting their behaviour during operation based on the data being gathered. To develop trustworthy information systems, new methods and tools are required for verification, validation and software testing. The traditional way of testing the systems during the development and before the deployment does not suffice anymore.

IVVES (Industrial-Grade Verification and Validation of Evolving Systems) project aims to

address these challenges by researching and developing methods to test ML and AI solutions and evolving systems, and using AI and ML to improve and automate development and testing.

3. Summary of the current results

To summarize the current technical results, we follow the three focus areas of the IVVES project. For the topic of validation techniques for ML [1], we have developed methods and tools for the quality assurance of the incoming data[2], for example, data fault injection to test machine learning systems [2], generating synthetic data for healthcare and cybersecurity, automating explorative data analysis and selection, audio data of industrial hardware, and using natural language processing (NLP) methods on semi-natural languages [3], and metamorphic testing for text-driven environmental, social, and governance investment systems.

We have developed validation methods and techniques for evolving systems, for example, test generation and test prioritization[3] for fault detection [4], scriptless end-to-end test generation for ES with coverage analysis [5], machine learning-assisted automated performance testing [6], anomaly detection for industrial environments, flaky test detection[4], automated test failure root cause analysis, test oracle mining, automated behavioral change detection, conformal prediction for edge applications, and code defect risk prediction.

In the area of data-driven engineering, we have developed methods and tools for data collection, for example, for customer program resource utilization in production, simulating operational technology networks, and collecting data during automated exploration of graphical user interfaces. The collected data is being used, e.g., for using AI to enhance the verification flow for healthcare devices, detecting abnormal network behavior, testing automated investment systems, testing automated driving[5] [7], predictive analysis in industrial environments, fault analysis and anomaly detection, improve automated exploration and testing through graphical user interface, test duration optimization, and performance testing[6] [8] and analysis.

4. IVVES research on scriptless GUI testing

This chapter gives more details on the research results on scriptless end-to-end testing through graphical user interface (GUI), using model inference and ML to improve automated exploration of GUI, and using inferred models for automated change analysis between consequent system versions.

4.1. TESTAR research and development

TESTAR[7] [5] is an open source tool for scriptless test automation through GUI. In scriptless GUI testing, the tests are generated at run-time during the test execution, based on the observed

[2] https://github.com/soft-nougat/dqw-ivves
[3] https://github.com/F-Secure/pytest-rts
[4] https://github.com/F-Secure/flaky-tests-detection
[5] https://github.com/mahshidhelali/Deeper_ADAS_Test_Generator
[6] https://github.com/mahshidhelali/RL-Assisted-Performance-Testing
[7] https://testar.org/

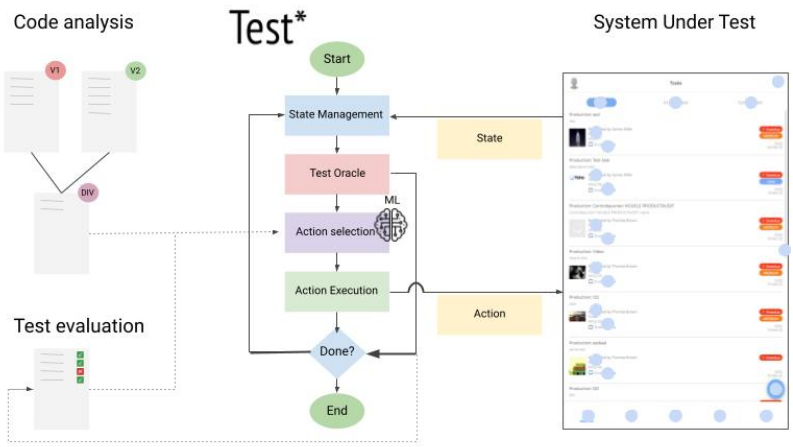

Figure 1: Architecture of TESTAR with static code analysis

state of the GUI and available actions. This means that the tool automatically explores the GUI, trying to find failures. To match the requirements of IVVES project partners and use cases, TESTAR has been extended and improved in many ways.

One of the main research topics of IVVES is applying AI and ML to improve testing. In TESTAR, we are using reinforcement learning to optimise action selection for improved GUI exploration [9, 10]. As ML requires a model to learn, an important research direction has been state model inference during the GUI exploration [5]. To infer a model faster, we are researching a distributed approach for model inference, using multiple independent TESTAR instances and a shared state model database, so that each instance reserves different unvisited action from the model and shares the results of the GUI exploration in the state model. Another approach for TESTAR ML research is to use evolutionary algorithms to evolve action selection rules.

In collaboration with Marviq (NL) and Innspire (NL), TESTAR has been improved in various ways, for example to generate better test reports. One research collaboration is aiming to map code coverage footprints of GUI actions, so that TESTAR action selection algorithm could target specific parts of the source code of the system under testing. TESTAR could target the changed parts of the code or code smells given by static analysis tools, for example SonarCube. By running a version comparison in version control (e.g., Git), TESTAR can prevent targeting code smells in parts that have already been evaluated. By gathering input from the user, for example in form of evaluating the test results, ML techniques can be trained with labelled data. Through this approach, the monkey testing that TESTAR performs, using a random action selection method on the SUT, is extended to a smarter monkey testing that increases the quality of the findings (test oracle verdicts) and increases the code coverage in addition to human designed tests. The approach has been illustrated in the Figure 1.

Code coverage mapping could be used also to try to cover more branches of the source code,

analysing the branch conditions and trying to generate inputs to match the conditions.

4.2. Scriptless testing in IVVES use cases

TESTAR is being evaluated in the use cases of ING (NL) and F-Secure (FI). At ING[8], TESTAR has been extended to support mobile apps through Appium[9] and new features have been added to improve testing of web apps using Selenium WebDriver[10]. The idea was not to (entirely) replace scripted testing, but to complement it by reducing the effort to create test scripts and covering paths outside the happy user scenarios, concentrating on robustness testing instead of functional testing.

Research collaboration with F-Secure aims to automated change analysis between system versions. The goal is to detect unintended changes that might be regression bugs. With TESTAR, we compare the inferred models of consequent system versions to analyse what has changed between the versions, and there is a new open source tool for the change detection[11] that also visualizes the detected changes for the user.

Change-Analyzer[12] is an open-source framework, developed by F-Secure, utilising Reinforcement Learning techniques, and leveraging OpenAI Gym library, Ludwig-AI framework and TensorFlow. Change-Analyzer framework aims to allow product teams to get feedback regarding the quality of their software products, without having prior knowledge about the software. In short, the main features of the framework are: 1) Exploration feature - this is achieved by randomly using the available actions within the software, or more efficient by using Reinforcement Learning to explore the software by following certain policies (for instance keep exploring only a specific domain or application). 2) Replay feature - previously generated tests are executed against new versions of the same software. 3) Analysis feature - two test results from the same test sequence are compared and differences are highlighted for the Software engineer to validate if it is a defect or an intended behaviour due to new changes. Currently, Change-Analyzer has support for Web-based applications and Windows applications.

Acknowledgments

This project has been labelled by ITEA3 and funded in the Netherlands by the Netherlands Enterprise Agency (RVO).

References

[1] L. Myllyaho, M. Raatikainen, T. Männistö, T. Mikkonen, J. K. Nurminen, Systematic literature review of validation methods for ai systems, Journal of Systems and Software 181 (2021) 111050. URL: https://www.sciencedirect.com/science/article/pii/S0164121221001473. doi:https://doi.org/10.1016/j.jss.2021.111050.

[8] https://medium.com/ing-blog/scriptless-gui-test-automation-at-ing-54c003649aa6
[9] https://appium.io/
[10] https://www.selenium.dev/documentation/webdriver/
[11] https://github.com/TESTARtool/ChangeDetection.NET
[12] https://github.com/F-Secure/change-analyzer/

[2] J. K. Nurminen, T. Halvari, J. Harviainen, J. Mylläri, A. Röyskö, J. Silvennoinen, T. Mikkonen, Software framework for data fault injection to test machine learning systems, in: 2019 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), 2019, pp. 294–299. doi:10.1109/ISSREW.2019.00087.

[3] Z. Hussain, J. K. Nurminen, T. Mikkonen, M. Kowiel, Command Similarity Measurement Using NLP, in: R. Queirós, M. Pinto, A. Simões, F. Portela, M. J. a. Pereira (Eds.), 10th Symposium on Languages, Applications and Technologies (SLATE 2021), volume 94 of *Open Access Series in Informatics (OASIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 2021, pp. 13:1–13:14. URL: https://drops.dagstuhl.de/opus/volltexte/2021/14430. doi:10.4230/OASIcs.SLATE.2021.13.

[4] E. Kauhanen, J. K. Nurminen, T. Mikkonen, M. Pashkovskiy, Regression test selection tool for python in continuous integration process, in: 2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER), 2021, pp. 618–621. doi:10.1109/SANER50967.2021.00077.

[5] T. E. J. Vos, P. Aho, F. Pastor Ricos, O. Rodriguez-Valdes, A. Mulders, TESTAR – scriptless testing through graphical user interface, Software Testing, Verification and Reliability 31 (2021) e1771. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/stvr.1771. doi:https://doi.org/10.1002/stvr.1771. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/stvr.1771, e1771 stvr.1771.

[6] A. Sedaghatbaf, M. Helali Moghadam, M. Saadatmand, Automated performance testing based on active deep learning, in: 2021 IEEE/ACM International Conference on Automation of Software Test (AST), 2021, pp. 11–19. doi:10.1109/AST52587.2021.00010.

[7] M. H. Moghadam, M. Borg, S. J. Mousavirad, Deeper at the sbst 2021 tool competition: Adas testing using multi-objective search, in: 2021 IEEE/ACM 14th International Workshop on Search-Based Software Testing (SBST), 2021, pp. 40–41. doi:10.1109/SBST52555.2021.00018.

[8] M. H. Moghadam, G. Hamidi, M. Borg, M. Saadatmand, M. Bohlin, B. Lisper, P. Potena, Performance testing using a smart reinforcement learning-driven test agent, in: 2021 IEEE Congress on Evolutionary Computation (CEC), 2021, pp. 2385–2394. doi:10.1109/CEC45853.2021.9504763.

[9] O. Rodriguez-Valdes, Towards a testing tool that learns to test, in: 2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2021, pp. 278–280. doi:10.1109/ICSE-Companion52605.2021.00127.

[10] A. van der Brugge, F. P. Ricos, P. Aho, B. Marín, T. E. Vos, Evaluating TESTAR's effectiveness through code coverage, in: S. Abrahão Gonzales (Ed.), JISBD2021, SISTEDES, 2021. URL: http://hdl.handle.net/11705/JISBD/2021/042.