Tutorial Title:

The Datamorphic Testing Methodology: Principles, Tools and Applications to Machine Learning

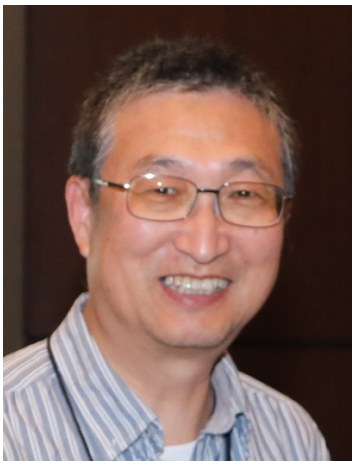
Speaker:

Prof. Hong Zhu, School of Engineering, Computing and Mathematics, Oxford Brookes University, Oxford OX33 1HX, UK, Email: hzhu@brookes.ac.uk

Short Bio:

Dr. Hong Zhu is a professor of computer science at the Oxford Brookes University, Oxford, UK, where he chairs the Cloud Computing and Cybersecurity Research Group. He obtained his BSc, MSc and PhD degrees in Computer Science from Nanjing University, China, in 1982, 1984 and 1987, respectively. He was a faculty member of Nanjing University from 1987 to 1998. He joined Oxford Brookes University in November 1998 as a senior lecturer in computing and became a professor in Oct. 2004. His research interests are in the area of software development methodologies, including software engineering for cloud computing and software engineering of intelligent systems, formal methods, software design, programming languages and automated tools, software modelling and testing. He has published 2 books and more than 190 research papers in journals and international conferences. He is a senior member of IEEE, a member of British Computer Society, and ACM.

Photo:



Abstract:

Datamorphic testing methodology regards software testing as an engineering process in which a test system is developed, maintained, evolved and operated to achieve software testing purposes. It defines software test systems as consists of a set of test entities and test morphisms, where the former are the objects, data, documents etc created, used and managed during testing process while the latter are the operators and transformers on the test entities. Typical examples of test morphisms include test case generators, data augmentations (which are called datamorphisms in datamorphic testing and play a significant role), test oracles (which are called metamorphisms), test adequacy metrics, etc. One of the most important principles of datamorphic testing methodology is that a test system should be explicitly defined and implemented (especially when testing is complicated and expensive) so that testing can be ensured of high quality and conducted effectively and efficiently, and also testing resources represented and embodied in test systems can be reused and efficiently evolved as valuable resources. Research demonstrated that when such a test system is implemented and maintained effectively, test automation can be achieved at three different abstract levels. At activity level, testing actions can be performed by invoking test morphisms. At the strategy level, test strategies can be formally defined as algorithms with test entities and test morphisms as parameters and applied via invoking the corresponding algorithms. At process level, the activities and the application of strategies can be recorded to form test scripts, which can be edited and replayed. Since such test scripts are at a higher level of abstraction than traditional test scripts, they are more reusable and less fragile to modification to the software under test. The principles of the datamorphic testing methodology has been applied to a number of testing problems for machine learning applications, including confirmatory testing of ML models, exploratory testing of ML classifiers, and to scenario-based functional testing for improve ML model performances. The tutorial will consist of three parts: (1) the principles and basic concepts of datamorphic testing methodology, (2) the automated testing tool and test environment Morphy that supports datamorphic testing methodology, and (3) the applications to machine learning models.