# Atif Mashkoor

## List of Publications by Citations



| | | | |
|---|---|---|---|
| 54 papers | 375 citations | 10 h-index | 16 g-index |
| 63 ext. papers | 488 ext. citations | 1.1 avg, IF | 4.41 L-index |

| # | Paper | IF | Citations |
|---|---|---|---|
| 54 | Software Safety and Security Risk Mitigation in Cyber-physical Systems. *IEEE Software*, **2018**, 35, 24-29 | 1.5 | 63 |
| 53 | The Hemodialysis Machine Case Study. *Lecture Notes in Computer Science*, **2016**, 329-343 | 0.9 | 19 |
| 52 | Asm2C++: A Tool for Code Generation from Abstract State Machines to Arduino. *Lecture Notes in Computer Science*, **2017**, 295-301 | 0.9 | 17 |
| 51 | Utilizing Event-B for domain engineering: a critical analysis. *Requirements Engineering*, **2011**, 16, 191-207 | 2.7 | 16 |
| 50 | Integrating formal methods into medical software development: The ASM approach. *Science of Computer Programming*, **2018**, 158, 148-167 | 1.1 | 14 |
| 49 | Formal validation and verification of a medical software critical component **2015**, | | 14 |
| 48 | Evaluating the suitability of state-based formal methods for industrial deployment. *Software - Practice and Experience*, **2018**, 48, 2350-2379 | 2.5 | 14 |
| 47 | Towards the Trustworthy Development of Active Medical Devices: A Hemodialysis Case Study. *IEEE Embedded Systems Letters*, **2016**, 8, 14-17 | 1 | 13 |
| 46 | Discovery and classification of user interests on social media. *Information Discovery and Delivery*, **2017**, 45, 130-138 | 1.4 | 12 |
| 45 | Improving the Understandability of Formal Specifications: An Experience Report. *Lecture Notes in Computer Science*, **2014**, 184-199 | 0.9 | 10 |
| 44 | Design and validation of a C++ code generator from Abstract State Machines specifications. *Journal of Software: Evolution and Process*, **2020**, 32, e2205 | 1 | 10 |
| 43 | **2018**, | | 9 |
| 42 | Incremental Construction of Realizable Choreographies. *Lecture Notes in Computer Science*, **2018**, 1-19 | 0.9 | 8 |
| 41 | Model-driven development of high-assurance active medical devices. *Software Quality Journal*, **2016**, 24, 571-596 | 1.2 | 8 |
| 40 | Domain Engineering with Event-B: Some Lessons We Learned **2010**, | | 8 |
| 39 | Stepwise Validation of Formal Specifications **2011**, | | 8 |
| 38 | Towards Validation of Requirements Models. *Lecture Notes in Computer Science*, **2010**, 404-404 | 0.9 | 8 |

| | | | |
|---|---|---|---|
| 37 | Formal Probabilistic Analysis of Cyber-Physical Transportation Systems. *Lecture Notes in Computer Science*, **2012**, 419-434 | 0.9 | 8 |
| 36 | How to Select the Suitable Formal Method for an Industrial Application: A Survey. *Lecture Notes in Computer Science*, **2016**, 213-228 | 0.9 | 8 |
| 35 | Validation of formal specifications through transformation and animation. *Requirements Engineering*, **2017**, 22, 433-451 | 2.7 | 7 |
| 34 | Build Software or Buy: A Study on Developing Large Scale Software. *IEEE Access*, **2017**, 5, 24262-24274 | 3.5 | 7 |
| 33 | Refinement-based Validation of Event-B Specifications. *Software and Systems Modeling*, **2017**, 16, 789-808 | 1.9 | 6 |
| 32 | A systematic literature review of the use of formal methods in medical software systems. *Journal of Software: Evolution and Process*, **2018**, 30, e1943 | 1 | 5 |
| 31 | Guidelines for Formal Domain Modeling in Event-B **2011**, | | 5 |
| 30 | Refinement-Based Development of Software-Controlled Safety-Critical Active Medical Devices. *Lecture Notes in Business Information Processing*, **2015**, 120-132 | 0.6 | 5 |
| 29 | Observation-Level-Driven Formal Modeling **2015**, | | 4 |
| 28 | Validation Obligations: A Novel Approach to Check Compliance between Requirements and their Formal Specification **2021**, | | 4 |
| 27 | Validation of Transformation from Abstract State Machine Models to C++ Code. *Lecture Notes in Computer Science*, **2018**, 17-32 | 0.9 | 4 |
| 26 | Live and global consistency checking in a collaborative engineering environment **2019**, | | 3 |
| 25 | A Literature Review of Using Machine Learning in Software Development Life Cycle Stages. *IEEE Access*, **2021**, 9, 140896-140920 | 3.5 | 3 |
| 24 | Unified Syntax for Abstract State Machines. *Lecture Notes in Computer Science*, **2016**, 231-236 | 0.9 | 3 |
| 23 | Generation of Behavior-Driven Development C++ Tests from Abstract State Machine Scenarios. *Communications in Computer and Information Science*, **2018**, 146-152 | 0.3 | 3 |
| 22 | Deriving Software Architectures for CRUD Applications: The FPL Tower Interface Case Study **2007**, | | 2 |
| 21 | Security Risk Mitigation of Cyber Physical Systems: A Case Study of a Flight Simulator. *Communications in Computer and Information Science*, **2019**, 129-138 | 0.3 | 2 |
| 20 | Formal Verification and Safety Assessment of a Hemodialysis Machine. *Lecture Notes in Computer Science*, **2018**, 241-254 | 0.9 | 2 |

| | | | |
|---|---|---|---|
| 19 | AsmetaA: Animator for Abstract State Machines. *Lecture Notes in Computer Science*, **2018**, 369-373 | 0.9 | 2 |
| 18 | Using Probabilistic Analysis for the Certification of Machine Control Systems. *Lecture Notes in Computer Science*, **2013**, 305-320 | 0.9 | 2 |
| 17 | Conceptual Modelling of Hybrid Systems. *Lecture Notes in Computer Science*, **2017**, 277-290 | 0.9 | 2 |
| 16 | Formal design of scalable conversation protocols using Event-B: Validation, experiments, and benchmarks. *Journal of Software: Evolution and Process*, **2020**, 32, e2209 | 1 | 2 |
| 15 | Towards Optimal Assembly Line Order Sequencing with Reinforcement Learning: A Case Study **2020**, | | 2 |
| 14 | Safe and secure cyber-physical systems. *Journal of Software: Evolution and Process*, **2021**, 33, e2340 | 1 | 2 |
| 13 | Abstract State Machines, Alloy, B, TLA, VDM, and Z. *Lecture Notes in Computer Science*, **2016**, | 0.9 | 2 |
| 12 | Multifaceted Consistency Checking of Collaborative Engineering Artifacts **2019**, | | 2 |
| 11 | Validation of Formal Models by Timed Probabilistic Simulation. *Lecture Notes in Computer Science*, **2021**, 81-96 | 0.9 | 2 |
| 10 | An Event-B-based approach to hybrid systems engineering and its application to a hemodialysis machine case study. *Computer Languages, Systems and Structures*, **2018**, 54, 297-315 | | 1 |
| 9 | Collaboratively enhanced consistency checking in a cloud-based engineering environment **2019**, | | 1 |
| 8 | Handling Reparation in Incremental Construction of Realizable Conversation Protocols. *Communications in Computer and Information Science*, **2018**, 159-166 | 0.3 | 1 |
| 7 | Ensuring safe and consistent coengineering of cyber-physical production systems: A case study. *Journal of Software: Evolution and Process*, **2020**, 33, e2308 | 1 | 1 |
| 6 | Evaluating the alignment of sequence diagrams with system behavior. *Procedia Computer Science*, **2021**, 180, 502-506 | 1.6 | 1 |
| 5 | **2018**, | | 1 |
| 4 | Model-driven engineering of safety and security software systems: A systematic mapping study and future research directions. *Journal of Software: Evolution and Process*, | 1 | 0 |
| 3 | Model-Driven Re-engineering of a Pressure Sensing System: An Experience Report. *Lecture Notes in Computer Science*, **2018**, 264-278 | 0.9 | |
| 2 | Intelligent Autonomous Systems. *Computer*, **2020**, 53, 20-23 | 1.6 | |

# List of Publications

| | | |
|---|---|---|
| 1 | A Conceptual Model for Mitigation of Root Causes of Uncertainty in Cyber-Physical Systems. *Communications in Computer and Information Science*, **2021**, 9-17 | 0.3 |