Software Quality Journal
https://doi.org/10.1007/s11219-019-09448-3

A survey on software aging and rejuvenation in the cloud

Roberto Pietrantuono[1] · **Stefano Russo**[1]

Published online: 07 June 2019

Abstract
The adoption of cloud computing for providing resource and delivering services is an irreversible trend. For most IT companies, the quality of provided services is strongly dependent on reliability and performance of the underlying cloud technologies. A widely studied problem that can greatly affect the user experience is runtime *software aging*, whose main countermeasure is a proactive maintenance action known as *rejuvenation*. This article reviews the effort conducted so far by the *software aging and rejuvenation* (SAR) community in the cloud domain. A set of 105 papers has been examined from three source digital libraries in order to have a clear view of the state of the art. The paper characterizes the cloud-related SAR literature according to four dimensions: the publication trends, the aging analysis methods and metrics, the rejuvenation solutions, the validation approach. Results witness an increasing interest in this area (with 58% of the studies published in the last 5 years), an equivalent role of measurement-based and model-based solutions for *aging analysis* (42 and 40 studies, respectively) and a prevalent interest for rejuvenation (76/105 of the studies deals with rejuvenation).

Keywords Software aging · Rejuvenation · Cloud · Performance · Virtualization · Virtual machine · Survey · Literature review

1 Introduction

Cloud computing has become the predominant paradigm for many companies to deliver their services; store, manage, and process data; develop, test, and deploy applications. Cloud applications and services have spread over all domains, in healthcare, transportation, mobile computing and many more. Last year, Forrester's estimated a global public cloud market

✉ Roberto Pietrantuono
roberto.pietrantuono@unina.it

Stefano Russo
stefano.russo@unina.it

[1] Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Naples, Italy

for 2018 of $178 billion, up from $146 billion in 2017, with more than 50% of global enterprises will relying on at least one public cloud platform (Columbus 2018). The 2019 predictions report about a global market, including cloud platforms, business services and SaaS, that will exceed $200 billion, with nearly 60% of enterprises relying on a public cloud platform (Bartoletti 2019). The huge impact of the cloud poses stringent requirements on availability, reliability and performance of software services and of underlying platforms. Failures on user requests or service delays have severe costs for providers; services need to be delivered without interruption while keeping acceptable performance over time.

Software aging is a well-known phenomenon that causes performance of a software system to gradually degrade and, eventually, leads to failure. Since about 20 years, it has been observed in many systems in operation, in business- and safety-critical applications. It is caused by the so-called aging-related bugs, a type of bugs that, once activated, leads to the accumulation of erroneous conditions and to a progressive consumption of resources, like physical memory. Software aging is difficult to detect before release, as aging-related bugs are hard to reproduce (Cotroneo et al. 2012; Cavezza et al. 2014). *Software rejuvenation* is a cost-effective preventive maintenance action to counteract aging, which cleans and restores the state of the application's environment. Many studies are available dealing with software aging and rejuvenation (SAR)—an overview is found in Cotroneo et al. (2014).

Since the spread of virtualization and of cloud computing, researchers of the SAR community started investigating the aging problem for this type of systems. Many studies have been conducted on aging in the cloud and, more broadly, on virtualized systems meant as the key technological enabler of the cloud. Researchers have proposed modeling- and measurement-based approaches tailored for common cloud configurations at different layers of the stack—physical hosts, virtual machine monitor (VMM), and virtual machines—with the goal of *detecting* and *assessing* aging problems, and then deciding the appropriate time for *counteracting* actions. Moreover, several interesting techniques have been developed for a "lightweight" rejuvenation, able of exploiting the advantages given by VMs as application containers to considerably reduce the downtime of applications at relatively low implementation cost. Because of the increasing complexity of cloud-related systems along with the importance of guaranteeing high levels of continuity of services, SAR research in this area is likely to keep on growing in the near future.

This paper surveys the research work so far about SAR in cloud-based systems. Recently, we performed a preliminary literature review on a set of 72 papers taken from the Scopus digital library (Pietrantuono and Russo 2018). Here, we extend that work both in terms of examined papers, considering two more digital libraries, and in terms of analyzed dimensions, with the goal of giving a comprehensive characterization of the literature. We look at *publication trends*, at *methods and metrics for aging analysis*, at *solutions for rejuvenation*, and at *how such methods and solutions are validated*. The results yield a clear view of research trends and give a basis to plan for future SAR research in the cloud. The next Section gives a short background. Section 3 describes the design of the conducted study, defining the analysis dimensions. Section 4 reports results and Section 5 discusses the main insights. Section 6 warns against threats to validity and Section 7 concludes the article.

2 Aging and rejuvenation in the cloud

The analysis of the software aging phenomenon deals with detection and prediction of the most likely Time To Aging Failure (TTAF), within which a preventive action, such as

rejuvenation, should be taken. The main strategies for aging analysis are categorized as: *model-based*, when analytic models are used to describe the phenomenon and to estimate the TTAF based on estimated parameters; *measurement-based*, when observed field data are used to infer the real trend of aging that is occurring, and predict the TTAF; *hybrid*, when field data are used to feed analytical models.

Model-based techniques can be applied to a wide range of systems, and they may provide more general findings than measurement-based ones. However, they can be less effective, since they have some simplifying assumptions, such as the one that the probability distributions characterizing the system behavior (e.g., the time-to-failure distribution) are known. Measurement-based methods forecast aging based on direct measurements (e.g., on time series analysis and machine learning), and provide empirical data about aging phenomena. Their advantage is that prediction can adapt to the current condition of the system (e.g., the current operational profile, which may not have been foreseen before operation), and can accurately determine the TTAF. On the other hand, they may be not easily generalizable to other systems, since they exploit aspects related to the nature of the considered system. Hybrid techniques try to combine both, feeding models online by field data. In our analysis, we keep this classification to be aligned with the SAR literature.

The effects of software aging can be assessed by means of proper metrics. These can be (i) direct indicators of the aging phenomenon, such as system resources usage like free physical memory, used swap space, file and process tables size, response time or throughput—usually referred to as *aging indicators* (Grottke et al. 2008), or (ii) indirect indicators, usually dependability attributes like availability, performability, and survivability, used as proxy to assess the effect of aging and rejuvenation on a system, regardless of which specific resource is being depleted. In our analysis, we refer to both as *aging metrics*.

In virtualized environments, the set of potential aging metrics is larger, as the execution environment includes not only the physical machine with its operating system (OS) and low-level software but also the virtual environment, such as the virtual machine (VM) on which the software is running and/or the whole virtualization technology; this includes the virtual machine monitor (VMM), the most important layer responsible for creating a VM environment for an OS and its applications.

Virtualization is managed (i) via *native* VMMs (also known as *bare-metal hypervisors*, such as *Xen, VmWare ESX Oracle Vm Server*), in which there is no host OS but the VMM runs on and controls directly the hardware, or (ii) via *hosted* solutions (also known as *hosted hypervisors*, like *VMware Workstation, VirtualBox, Parallels*), where the hardware is accessed via a host OS. On top of the VMM, there are VMs; depending on the proactive/reactive fault tolerance strategy (including rejuvenation), there can be *standby* VMs, which can take over active VMs in case of failures or upon rejuvenation actions. In more general schemes (e.g., in virtualized data centers), there may be other physical nodes, and a VM can be migrated to a different node if, for instance, its hosting node (or part thereof, like the VMM) fails or needs to be rejuvenated.

In this scenario, the *execution environment* is quite different from a traditional non-virtualized environment. In the SAR perspective, this affects (i) the choice of the most appropriate strategy for aging analysis (i.e., the way to predict the most likely time of aging failure occurrence, and ultimately *when* to schedule rejuvenation); (ii) the aging metrics to look at, and (iii) the consequent rejuvenation action to enact (i.e., *how* to perform rejuvenation). In the following analysis, all these aspects are taken into account.

3 Study design

For a systematic analysis, we followed well-established guidelines for literature review (Kitchenham and Brereton 2013; Petersen et al. 2015).

3.1 Research questions

The study targets the following research questions:

RQ1 – *What are the publication trends of research studies about SAR in the cloud?* The goal of this RQ is to characterize the intensity of scientific interest for this research area, the base of researchers working on it, and what are the relevant venues where they publish their results.

RQ2 – *How do researchers target the aging analysis of cloud-based systems?* This RQ aims at characterizing how researchers detect and assess the phenomenon of software aging in cloud systems, what models, statistical techniques and aging indicators they use.

RQ3 – *How do researchers target rejuvenation of cloud-based systems?* This RQ concerns the approaches chosen for rejuvenating cloud-based systems.

RQ4 – *How do researchers validate their proposals?* This RQ looks at how SAR models and solutions are validated, so as to provide figures on the extent to which they can be applied in practice.

3.2 Selection process

The following selection process was followed:

1. **Initial search and filtering**. The initial selection was performed by a keywords-based search on three major digital libraries—*SciVerse Scopus*, *IEEE Xplore Digital Library (DL)*, *ACM Digital Library (DL)*. *Scopus* is one of the largest general purpose DBs for peer-reviewed literature and indexes journals/proceedings of the most common publishers (Elsevier, Springer, Wiley, IEEE, ACM, etc.). As general purpose DB, we preferred *Scopus* to *Google Scholar*, as the latter includes non-published literature (e.g., preprint), papers that are not scientific articles or not peer-reviewed (e.g., technical reports, theses and other grey literature). In order to complement the search with specific (non-general purpose) DBs, we opted for a major library to cover the *computer engineering* area (*IEEE Xplore DL*) and for a major library to cover the *computer science* area (*ACM DL*)—the fields where we expect the studies about software aging to appear.

   The search required the words "software" and ("aging" or "rejuvenation") to appear together with at least "cloud" or "virtual" in the metadata (title, abstract, and keywords) of publications.

   This provided 298, 367, and 202 studies from *Scopus*, *IEEE Xplore DL*, and *ACM DL*, respectively (including duplicates). Due to the conservative search, many studies were found to be unrelated to the computer engineering/science—mostly belonging to medicine fields, wherein the words "aging" and "rejuvenation" are common. Further not relevant results were editorials, standards, and conference proceedings material (e.g., welcome messages, ToC). These papers were removed based on the title and publication venue, getting to an initial set of 219 studies from the three libraries, without duplicates.

2. **Application of selection criteria**. On the set of 219 studies, the following inclusion/exclusion criteria are applied:

   – *Inclusion Criterion 1*. Studies targeting the software aging and/or software rejuvenation problem with reference to cloud-based systems meant as described in Section 2.
   – *Inclusion Criterion 2*. Studies subject to peer review.
   – *Inclusion Criterion 3*. Studies written in English.
   – *Exclusion Criterion 1*. Studies wherein the word "virtual" refers to the Java virtual machine and not related to the use of virtualization as cloud-enabling technology.
   – *Exclusion Criterion 2*. Studies focusing on software aging or rejuvenation but not referring to the cloud (i.e., in which the occurrence of the word "cloud" or "virtual" in the metadata or abstract is coincidental) and studies focusing on cloud computing but not related to SAR (i.e., where the words aging and rejuvenation in the metadata or abstract occur coincidentally).
   – *Exclusion Criterion 3*. Secondary or tertiary studies (e.g., systematic literature reviews, surveys).
   – *Exclusion Criterion 4*. Studies not available as full-text.

   No filtering based on the publication year and/or venue is applied. With such criteria, the initial set of 219 papers is reduced to 105 papers.

The complete list is available at https://github.com/rpietrantuono/SQJ2018. The final list of 105 papers is completely covered by the *SciVerse Scopus* DB. Out of the 219 studies before applying inclusion/exclusion criteria, 63 studies appear only in *Scopus*, 59 only in *IEEE Xplore DL*, and 33 only in the *ACM DL*; the remaining 64 studies appear in at least two libraries. Hence, in the first phase, all the three libraries contributed to select the studies. After applying inclusion/exclusion criteria, there is a higher overlap: 36 out of the final 105 studies are present only in the *Scopus* DB; the remaining 69 studies are in *Scopus* DB and in at least one further library (69 out of 105), while there is no study in *IEEE Xplore DL* or *ACM DL* that is not indexed by *Scopus*.

3.3 Data extraction and classification

The analysis refers to four dimensions related to the research questions:

Publication trends The first dimension refers to the publication trends and includes these attributes: number of publications by year, publication type (i.e., journal, conference, or workshop) and venue, publications' authors and research groups, and number of citations.

Aging analysis The main goal of aging analysis—encompassing aging detection and estimation—is to determine the most likely time of aging failure occurrence, so as to figure out *when* to schedule a rejuvenation action. As described in Section 2, this is done by approaches known as model-based, measurement-based, or hybrid. This dimension distinguishes between them, looking at the *modeling formalisms* and *aging metrics* for the model-based approaches; the *statistical techniques* and *aging metrics* for the measurement-based ones; and *modeling formalisms*, *statistical techniques*, and *aging metrics* for hybrid strategies.

Rejuvenation techniques The goal of rejuvenation is to figure out *how* a rejuvenation action should be performed to minimize the negative impact on the user experience

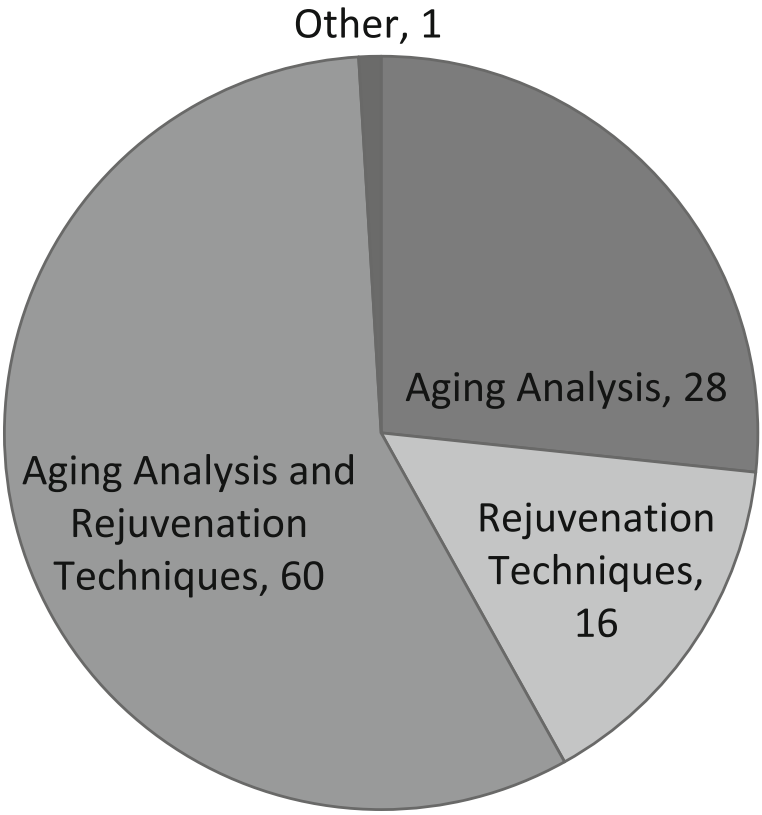


Fig. 1 Overview of analyzed papers

(e.g., downtime). This dimension distinguishes the techniques adopted for rejuvenating applications running in the cloud and/or parts of the cloud computing platform itself.

With reference to RQ2 and RQ3 (which deal with the research proposal of the study), we classify the studies as belonging to the following categories:

- *Aging analysis*, with studies dealing only with the aging analysis problem: aging detection and/or aging estimation, which is the input to decide *when* to perform rejuvenation, but not *how* to perform it;
- *Rejuvenation techniques*, with studies focusing on the proposed rejuvenation techniques assuming that the output of an aging analysis is already available;
- *Aging analysis and rejuvenation techniques*, with studies concerning with both aging analysis and the proposed rejuvenation technique.

Figure 1 shows that the majority of papers cope with both aging analysis and rejuvenation. Rejuvenation in the cloud plays an important role: a significant slice of papers is concerned exclusively with the rejuvenation problem and overall 76 out of 105 studies (i.e., the sum of categories 2 and 3) are concerned with how rejuvenation should be performed, unlike the general SAR literature (Cotroneo et al. 2014). "Other" refers to 1 study dealing with a "static" analysis of aging-related bugs in cloud computing software.

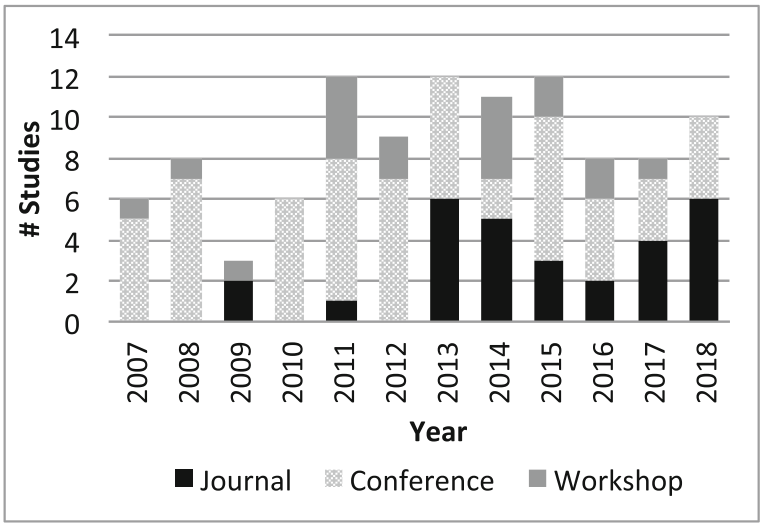


Fig. 2 Number of publications per year

Validation The fourth dimension is about how cloud SAR solutions are validated—namely, by numerical analysis, by simulation, or by experiments on case studies. In the following, results about each dimension are provided.

4 Results

4.1 RQ1: publication trends

Publications by year Figure 2 plots the selected studies by year and publication type (*Journal*, *Conference*, or *Workshops* proceedings). Publications in this area start from 2007, when the theme of cloud computing and virtualization started gaining popularity, with a peak from 2011 to 2015. Many papers appeared also in 2017 and 2018, especially on journals. The average number of cloud-related SAR publications was 7.5 papers per year. Most of the considered studies are published in conference proceedings (58/105), followed by journals (29/105) and workshops (18/105). However, if we consider the last 5 years (2013–2018), journal papers are 26, conference papers are 26, and workshop papers are 9. Hence, the research area has a considerable interest, likely due to challenges and opportunities offered by cloud-based applications, and is mature enough to produce mostly journal-level contributions.

Publications venues Papers appeared in 74 different venues,[1] denoting a large diversity of potentially interested research communities. Fifteen venues host almost half of the papers (46/105); this is shown in Fig. 3. The highest number of papers appeared in WoSAR, because of its specific focus on SAR, and the journal special issues promoted by WoSAR – Performance Evaluation (PE) and Journal on Emerging Technologies in Computing Systems

[1]The full list of venues is available at https://github.com/rpietrantuono/SQJ2018

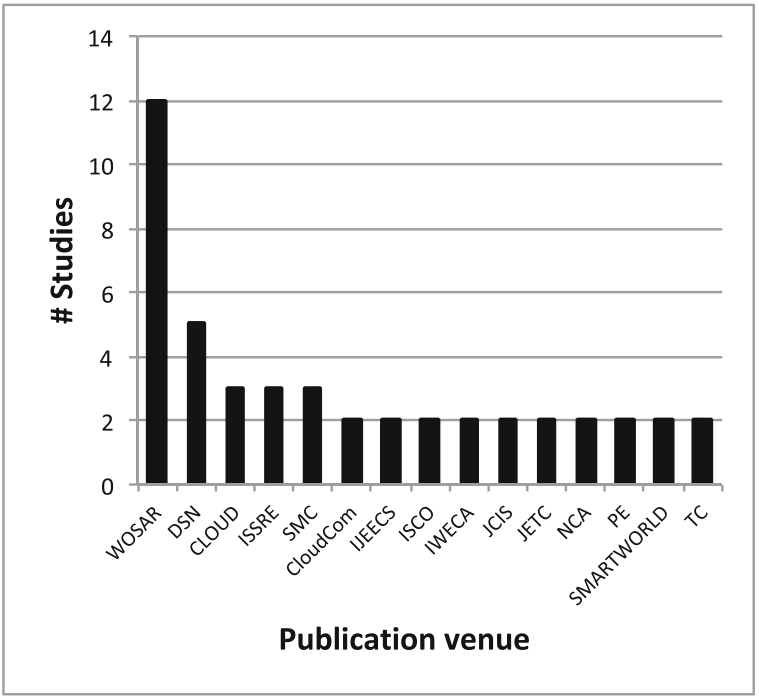


Fig. 3 Number of publications in major journals and conference proceedings

(JETC) in Fig. 3 are among these. The top venues include the best dependability/reliability conferences, such as DSN and ISSRE (WoSAR proceedings are part of ISSRE). It is worth noting the interest of cloud-related conferences, with 5 papers published in international conferences on Cloud Computing Technology and Science (CloudCom) and on Cloud Computing (CLOUD).

Research groups The total number of *different* authors for the 105 studies is 186. The total number of authors is 314; hence, in the average, it is about 3 authors per paper. Figure 4 reports the number of authors who co-authored more than one paper, denoting that this research area involves quite a high number of people.

Note that about 1/4 of the total (48/186) co-authored more than one paper, while 28/186 co-authored more than two. A large share (138/186) published exactly one study on the topic. Some authors (with their research group) are very active in the field, with more than 5 (9/186), 6 (7/186), and up to 7 (4/186) published studies. It is worth to mention the name of researchers who published more, to figure out the most active research groups: *K.S. Trivedi* published 9 papers; *F. Machida*, *P. Maciel*, and *R. Matias* published 8 papers; and *J. Alonso*, *J. Araujo*, and *R. Matos* published 7 papers.

Similarly, we consider the number of different institutions (taken from the affiliation) with an authors involved in the publication. The total number of different institutions for the 105 studies is 87. Figure 5 reports the number of institutions whose members published more than one study. Almost the half of institutions (39/87) appear more than one study, and 16/87 appear in more than two. The most active research groups come from the following institutions: *Duke University* and *Federal University of Pernambuco*, with 10 occurrences;

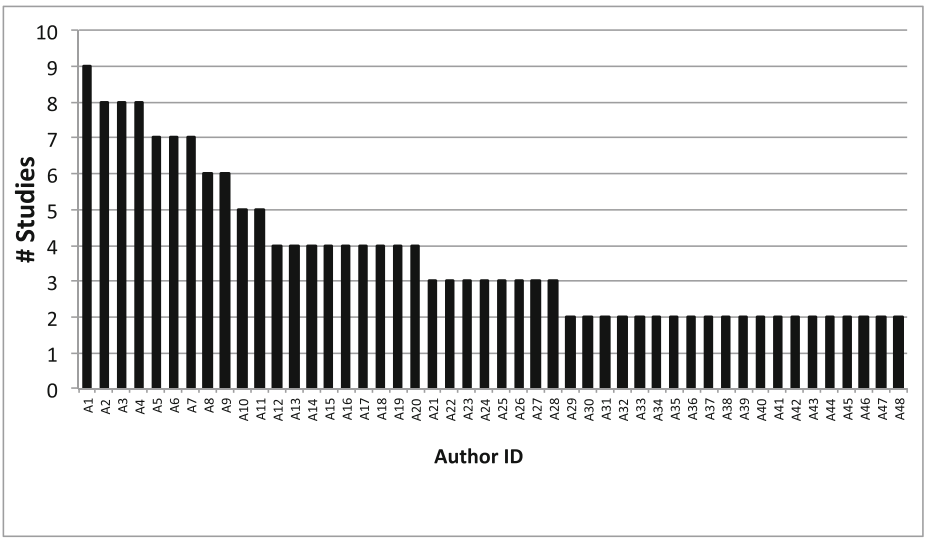


Fig. 4 Number of authors publishing more than one study

NEC Corporation, 8 studies; *Korea Aerospace University*, 7 studies; and *Federal University of Uberlandia*, 6 studies.

Finally, we consider the countries involved Fig. 6 . Twenty-one different countries have researchers that published at least one paper about cloud-related SAR, with China (21 occurrences) and the USA (12 occurrences) being the most represented countries. These numbers denote a certain diversity of contributors to this research area.

Number of citations The total number of citations[2] for the selected 105 studies amounts to 869, with an average of 8 citations per paper. Figure 7 a and b plot the absolute number of citations by year and the number of citations per paper by year. The trend of the two figures is very similar. The average citations per paper has a peak of 27 citations per paper in 2009 and is between 5 and 15 until 2015. More recent papers are of course less cited; if we normalize also with respect to the age of the paper (i.e., number of citations per paper divided by number of years since their publication), the trend is still very similar (Fig. 7c). Table 1 shows the top-5 cited papers in terms of number of citations normalized over their age and the top-5 cited papers in terms of absolute number of citations.

4.2 RQ2: aging analysis

This section deals with the studies concerning the aging analysis. With reference to the classification in Fig. 1, there are the studies falling in category 1 (*Aging Analysis*) and category 3 (*Aging Analysis and Rejuvenation techniques*)—or a total of 88 papers corresponding, respectively, to the papers with the following identifiers: from P-1 to P-28 and from P-45 to P-104 in the mentioned list of papers. Aging analysis is coped with either a *model-based*, *measurement-based*, or *hybrid* approach. Figure 8 sketches the breakdown of the papers.[3]

[2]The count of citations is taken from the three digital libraries we used, Sciverse Scopus, IEEEXplore, and ACM DL

[3]The number of papers is 90 because 2 papers adopt both a model-based and a measurement-based approach.

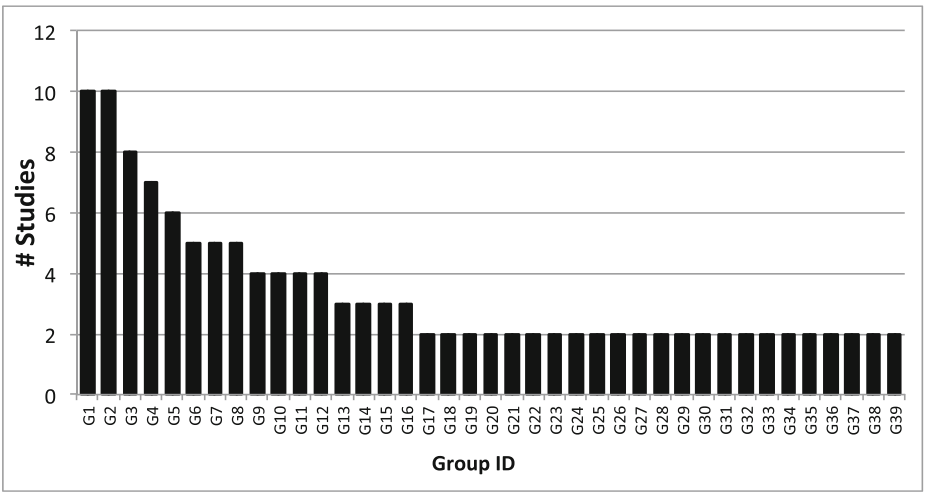


Fig. 5 Number of institutions whose members published more than one study

It highlights a balance between model-based and measurement-based studies. Rejuvenation is faced in most of the model-based, measurement-based, and hybrid papers—specifically, in 35 out of 40 model-based papers, 20 out of 42 measurement-based papers, and 6 out of 8 hybrid strategy papers.

As can be noted, the cloud aging phenomena have been analyzed almost the same number of times by model-based as by measurement-based techniques, with a non-negligible number of studies addressing the analysis by hybrid approaches. Figure 9 plots the trend over years.

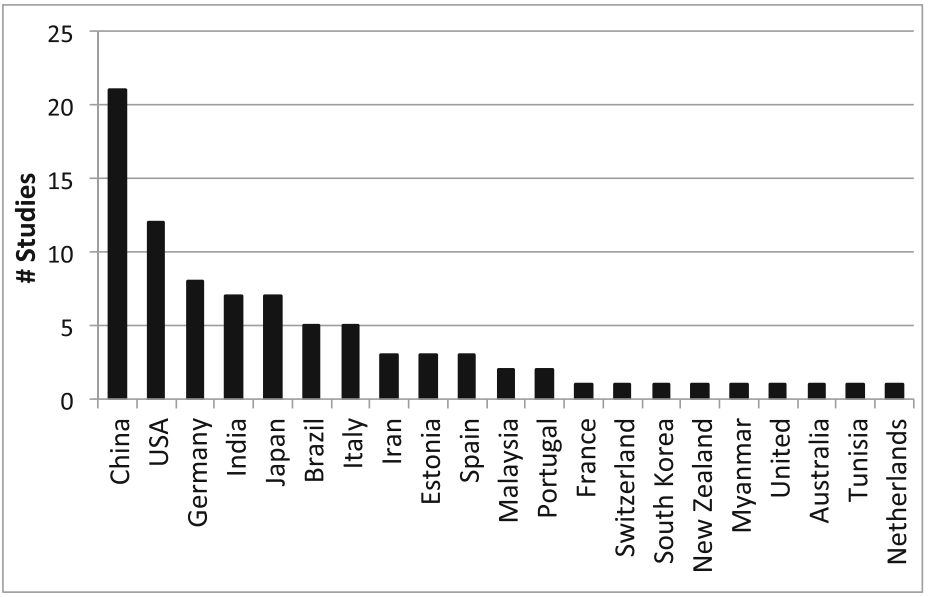
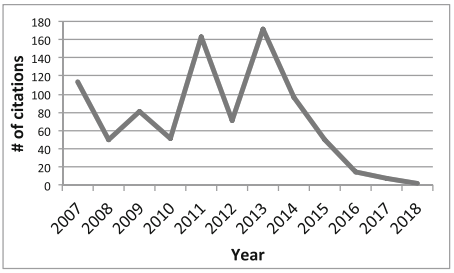
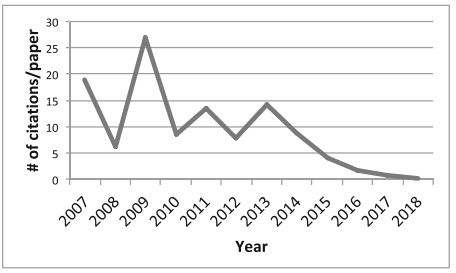


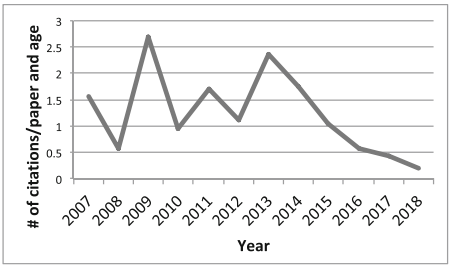
Fig. 6 Countries with authors contributing to the cloud-related SAR literature



(a) Citations by year



(b) Citations per paper by year



(c) Citations per paper and age by year

Fig. 7 Trends of citations

Since 2009, there is a substantial balance between the two approaches in every year, while in 2017 and 2018, the occurrences are highly unbalanced: in 2017, 7 measurement-based studies and just 1 model-based appeared, while in 2018, 7 model-based and just 1 measurement-based. This is likely to be attributed just to delays for publishing an article (considering that 10 out of 18 papers in 2017–2018 are on journals, which have a highly variable publication schedule) and less likely to systematic causes—2 years are not enough to claim a trend shift. In the following, the three categories of studies are examined.

4.2.1 Model-based aging analysis

Model-based studies are distinguished by (i) the adopted modeling formalism and (ii) the chosen metric of the effect of aging. Both aspects are explored hereafter.

Modeling formalism The aging process in more or less complex virtualized systems (with one or more VMs, physical hosts, and various rejuvenation strategies) is modeled by several types of stochastic formalisms, including state-space models—such as stochastic Petri nets (SPN) and stochastic reward nets (SRN)—continuous-time Markov chains (CTMC), semi-Markov processes (SMP), and combinatorial models—such as reliability block diagrams (RBD) and dynamic fault trees (DFT).

Figure 10 shows how many times each formalism has been used for SAR in the cloud domain.[4] Examples of works adopting the various models follow.

[4]The counting excludes the 8 hybrid studies, discussed later. Also, the sum is 46 instead of 40 because 6 studies adopt 2 formalisms each.

Table 1 Top-5 cited papers

Title	Year	Normalized no. of cit.	Absolute no. of cit.
Rated by normalized citations			
Workload-based software rejuvenation in cloud systems	2013	8.5	51
Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration	2013	5.6	34
Fast software rejuvenation of virtual machine monitors	2011	4.5	36
A fast rejuvenation technique for server consolidation with virtual machines	2007	4.08	49
Software rejuvenation based fault tolerance scheme for cloud applications	2015	4	16
Rated by absolute number of citations			
Workload-based software rejuvenation in cloud systems	2013	8.5	51
A fast rejuvenation technique for server consolidation with virtual machines	2007	4.08	49
Hypervisor-based efficient proactive recovery Server consolidation with virtual machines	2007	3.41	41
Fast software rejuvenation of virtual machine monitors	2011	4.5	36
Availability analysis of application servers using software rejuvenation and virtualization	2009	3.4	34

– *SPN-based models*. The most common formalisms are SPN-based, accounting together (SRN + SPN + MRSPN) for 15 occurrences. Examples are the work by Xu et al. (2014a) and by Rezaei and Sharifi (2010), who use SRNs to model a *single-server* virtualized system with time-based rejuvenation applied at VMM level and a measurement-based rejuvenation at VM level. SRNs are also used by Nguyen et al. (2014), who considers various failure and recovery modes of multiple VMs and VMMs; by Han and guo Xu (2013), who consider three different rejuvenation policies (no rejuvenation, time-based rejuvenation, and time and load-based delay rejuvenation) for single-server virtualization systems with multiple VMs on a single VMM; by Machida et al. (2010, 2013), who model various rejuvenation policies for VMs and VMM in a virtualized environment; recently, by Escheikh (2016, 2017) who analyze power management *performability*, evaluating the impact on performance and energy consumption in virtualized systems.

  Markov Regenerative Stochastic Petri Nets (MRSPNs) are adopted by Okamura et al. to perform the transient analysis of the two main rejuvenation policies, *Cold-* and *Warm-VM* rejuvenation (Okamura et al. 2014).
– CTMCs are used in 12 cases. For instance, Myint and Thein present a *multiple-host multiple-VMs availability analysis* where a primary standby CTMC model is defined,

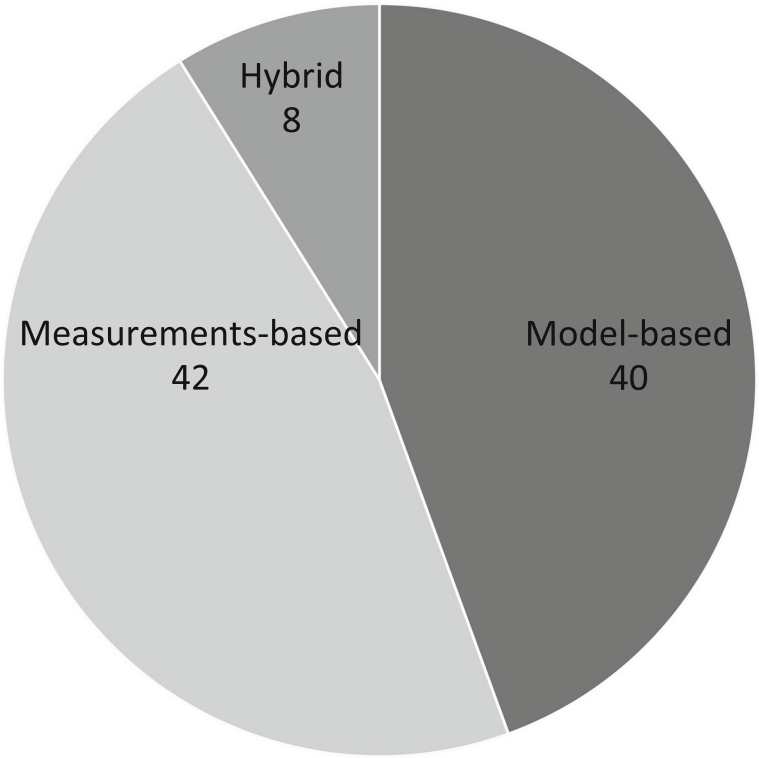


Fig. 8 Overview of papers coping with aging analysis

which includes a load balancer VM in each node for monitoring resources and a rejuvenation agent on each VM (Myint and Thein 2010).

The work by Thein et al. (2008) analyzes system *availability* with the time-based rejuvenation policy under different cluster configurations, 2 VMs hosted on a single physical server and 2 VMs per a physical server in dual physical servers. The same authors further present a software rejuvenation framework named VMSR to offer high *availability* for application server systems (Thein and Park 2009), proposing again a *CTMC* to model a *single host, multiple VMs* in the scheme with hot standby replicas.

– *Combinatorial models*: RBD and DFT occur in 8 cases: examples are the work by Melo et al., who formulate an availability model considering live migration for VMM rejuvenation based on extended RBD coupled with Deterministic Stochastic Petri Nets (DSPNs) (Melo et al. 2013b), and the work by Rahme et al., who exploit DFT to model cloud-based software rejuvenation (Rahme and Xu 2015).
– *Semi-Markov processes* are used 3 times. Examples include the works by Machida et al., analyzing the job completion time under aging and rejuvenation of the VMM (Machida et al. 2011; Machida 2014).
– *Optimization model*: 2 model-based studies, as well as 1 hybrid study, formulate optimization models to analyze and mitigate aging effects. An example is the work by Wu et al., which aims at reducing power consumption and transmission delay for computation offloading in mobile devices affected by aging (Wu and Wolter 2015).

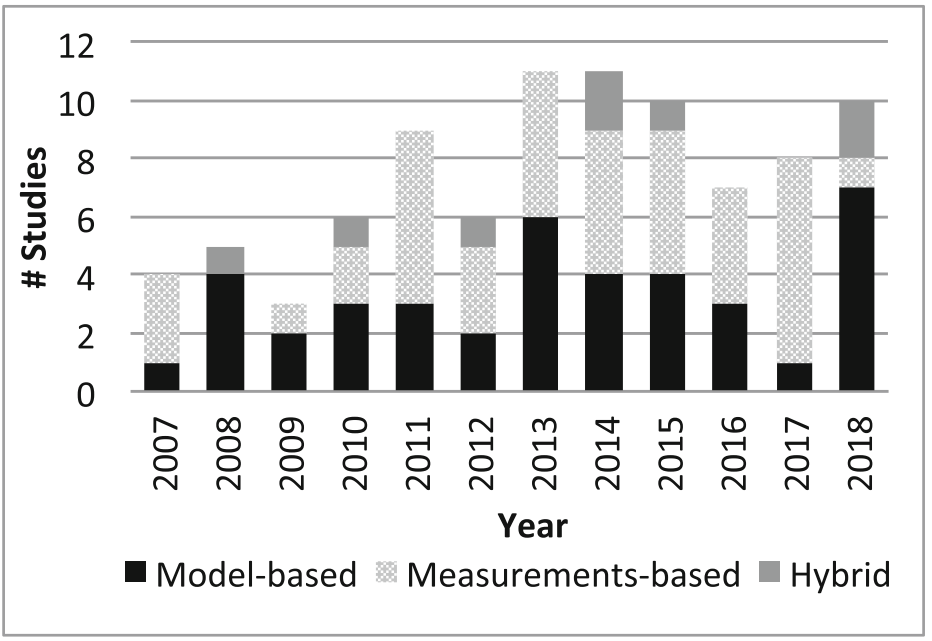


Fig. 9 Number of publications per year

Metrics Usually, model-based approaches do not refer to system-level resource depletion metrics, since the model is assumed to work regardless of which resource is being depleted. They mostly base their analysis on dependability attributes—the most common one being *availability*—and/or refer to user-perceived performance degradation metrics. Figure 11 reports the occurrences in model-based papers. The main metrics are discussed hereafter.

– *Availability* is by far the most common measure in model-based studies, with 25 occurrences. Availability is of particular relevance for cloud environments. In 14 cases, it is

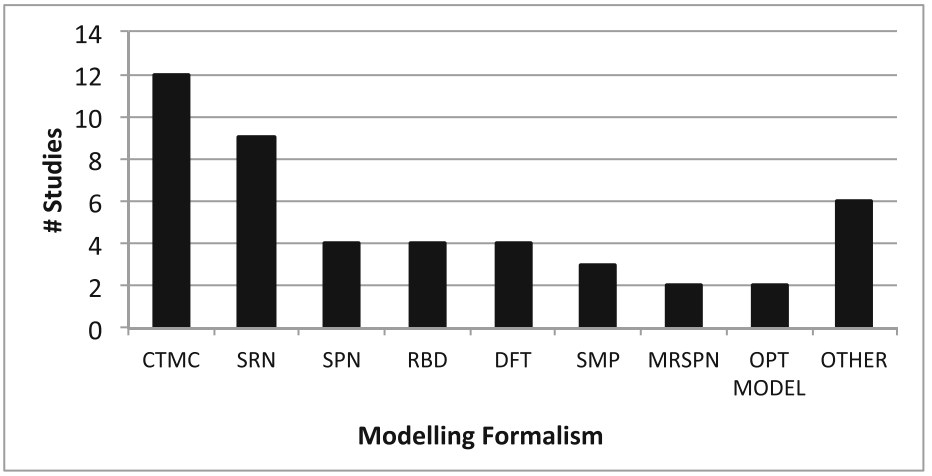


Fig. 10 Formalisms in the model-based approaches

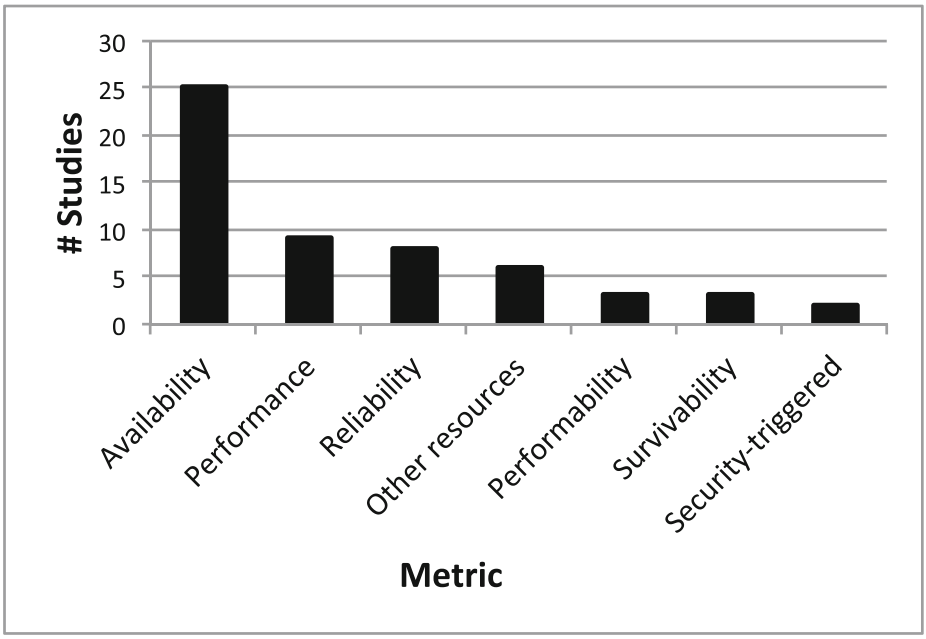


Fig. 11 Metrics used in model-based studies

analyzed with reference to no specific user-perceived metric, while in the remaining 11 cases a performance degradation metric (e.g., response time, throughput, latency) is considered. Examples include the mentioned works by Xu et al. (2014a), Rezaei and Sharifi (2010), and Nguyen et al. (2014). Melo et al. also model *cloud availability* by a SPN under two migration-based rejuvenation strategies (with and without a test before migration) (Melo et al. 2013a). Thein et al. analyze *availability* with the time-based rejuvenation policy under different cluster configurations (2 VMs on a single server and 2 VMs per server in dual physical servers) and later propose a rejuvenation framework named VMSR for application servers (Thein and Park 2009).

– Other dependability attributes are as follows: *reliability* in 8 cases (in all the cases in which a RBD/DFT is used), *performability* and *survivability* with 3 occurrences. The mentioned work by Rahme et al. models reliability of the cloud-based software rejuvenation (Rahme and Xu 2015). Escheikh analyzes power-management *performability*, evaluating the impact on performance and energy consumption in virtualized systems (Escheikh et al. 2017). The work by Chang et al. focuses on *survivability* of virtualized systems with VM/VMM rejuvenation (Chang et al. 2016).
– *Performance degradation metrics*, such as response time, job completion time, latency or throughput, are used in 9 model-based studies. In 3 cases, the model focuses on a performance degradation metrics with no explicit reference to a dependability attribute, for instance, in Machida et al. (2011), the model is conceived to analyze the job completion time, while in Wu and Wolter (2015) and Xia et al. (2014), the models refer to performance and energy consumption. In the other cases, the analyses look always at the impact on one among availability, reliability, survivability, and performability. Examples are the work by Nguyen, where transaction loss, beside availability, is evaluated

(Nguyen et al. 2014) and the work by Machida, where job time analysis is extended with steady-state availability analysis (Machida 2014).

– Six studies consider resource consumption in the model such as storage, CPU, network, and power consumption—e.g., the works by Escheikh where performance and energy consumption are modeled together (Escheikh et al. 2017)—while we found no model-based study considering memory consumption. Two studies consider in their models possible aging effects caused by security attacks.

4.2.2 Measurement-based aging assessment

Measurement-based approaches use observations gathered by monitoring relevant aging indicators to infer or predict the aging state of a system. Typical metrics are system resources consumption—e.g., memory and storage—and user-perceived metrics—e.g., response time, latency, and throughput. Data are collected during the execution and are used to forecast future trends based on past observations, making decision accordingly about rejuvenation, as well as about load balancing and static or dynamic resources allocation. The cloud-related approaches are hereafter distinguished by (i) the technique to analyze the data and (ii) the monitored aging metric.

Technique Techniques are grouped as as follows: *time series analysis*, *machine learning*, *threshold-based approaches*, and *other*s. Figure 12 shows their occurrence.

– *Time series analysis* is based on trend *detection* and *estimation* of a set of aging indicators. Tests for trend detection aim at accepting/rejecting the hypothesis of no trend in data (e.g., Mann-Kendall test). Trend estimation can exploit many models, e.g., (multiple) linear regression, regression smoothing, Sen's slope estimation, autoregressive models, and non-linear models. In the common case of presence of correlation

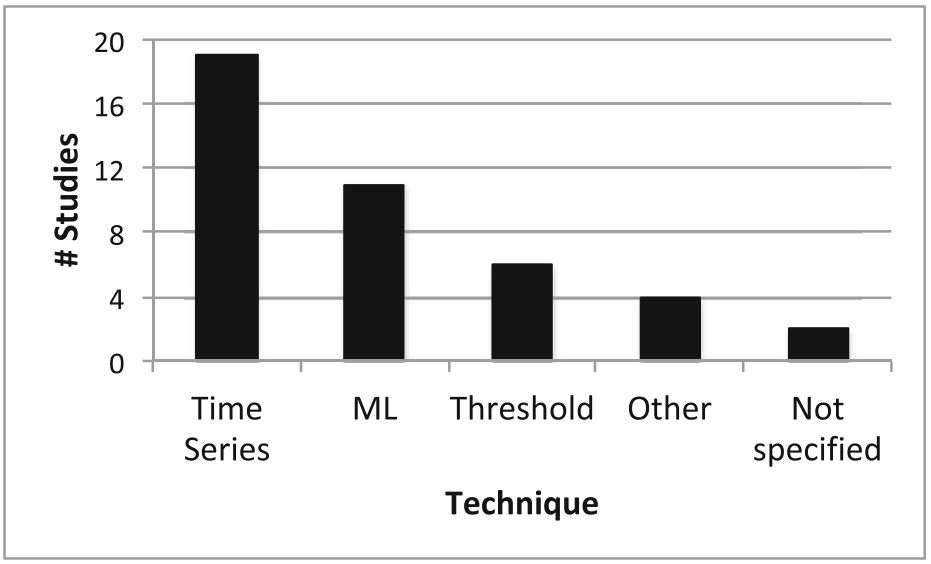


Fig. 12 Techniques used in the measurement-based approaches

among multiple aging indicators, data transformation, feature selection, or dimensionality reduction techniques are used. An example is principal component analysis followed by regression (Cotroneo et al. 2013, 2016; DeCelles et al. 2016).

Time series analysis was applied in 19 out of 42 measurement-based studies. Relevant examples include the works by Araujo et al. on the Eucalyptus cloud computing framework (Araujo et al. 2014). They adopt several regression models including *linear*, *quadratic*, *exponential growth*, and *Pearl-Reed logistic* models, to predict memory consumption trends and schedule software rejuvenation properly. Umesh et al. also exploit time series models to forecast software aging patterns of Windows active directory service (Umesh and Srinivasan 2017). DeCelles et al. (2016) apply anomaly detection technique based on *principal component analysis* (PCA) aimed at incipient faults such as software aging. Using case studies involving long-running enterprise benchmark applications, Trade6 and RuBBoS, with injected memory leaks, performance of the PCA-based detector when using just the compressed data is almost equivalent to the case in which the raw data is completely available, but with fewer samples with a compression rate exceeding 75%. Mohan and Reddy (2015) study the effect of aging on power usage by using *linear regression* to estimate the trend. Energy consumption is also considered in Villalobos et al. (2014) where an IDS-based self-protection mechanism at the virtual machine level inspired by software rejuvenation concepts is presented. A correlation between IDS accuracy, attack rate, cloud system workload, energy consumption, and response time is identified. Ficco et al. perform a workload-dependent analysis of performance degradation and memory indicators in Apache Storm, an event stream processing (ESP) application deploying tasks over a cloud architecture, by means of workload-dependent time series analysis. The Mann-Kendall test and Sen's procedure, often used for aging trend detection and estimation, are used on sliced windows where the workload and the performance/memory trends are judged to be in contrast (e.g., their trends increase despite the workload decreases) (Ficco et al. 2018). Sukhwani et al. apply several regression models to IBM cloud controller systems, including the following: Sen's slope, linear model, quadratic model, growth curve model, piecewise linear model, and, for datasets exhibiting seasonal behavior, Holt, Holt-Winters, and ARIMA (Autoregressive Integrated Moving Average) (Sukhwani et al. 2017).

– *Machine learning (ML)*. Machine learning algorithms in SAR are used to detect/predict the possible aging system state. We found them applied 11/42 measurement-based studies in the cloud domain. Examples include the works by Sudhakar et al., who use artificial neural networks to capture non-linear relationships between resource usage and time to failure in cloud systems (Sudhakar et al. 2014); Avresky et al. present a framework using machine learning for predicting failures caused by accumulation of anomalies and a proactive scale-up/scale-down technique in the cloud (Avresky et al. 2015). Simeonov proposes a framework with three VMs, one master and two identical slaves: the slaves send health data to the master, which predicts aging based on machine learning algorithms (Simeonov and Avresky 2010). Umesh et al. combine machine learning and adaptive genetic algorithms to predict software aging and schedule rejuvenation in virtualized environments (Umesh and Srinivasan 2016). Self-organizing maps (SOM) are also adotped for aging detection in virtual machine monitors (VMM) (Xu et al. 2014b).
– *Threshold-based approaches* define thresholds for some aging indicators, so as to trigger rejuvenation when they are exceeded. For instance, Silva et al. propose a rejuvenation framework called *VM-Rejuv*, exploiting virtualization to optimize recovery,

that adopts thresholds on mean response time and on quality of service indicators for aging detection (Silva et al. 2009). Another example is the work in Araujo et al. which combines threshold-based approach (with a threshold on memory utilization) with time series analysis (Araujo et al. 2011). It implements a rejuvenation policy in the Eucalyptus cloud computing infrastructure, by using multiple thresholds and forecasting by time series analysis models.

– *Other* techniques include *grey correlation*, used to compute a metric of aging based on fuzzy evaluation (Liu et al. 2013), and a technique for signal reconstruction via adaptive sampling, exploited to treat the performance time series gathered in data centers and detect the trend from the reconstructed signal (Huang et al. 2016).

Metrics measurement-based techniques gather aging data through system-level probes of resources exhaustion (e.g., memory, storage), as well as through probes at user level (e.g., response time, throughput). Figure 13 summarizes the used metrics.

– Memory indicators are very common in cloud-related studies, appearing in 28/42 measurement-based studies, in contrast to model-based ones where they never appear. Examples include free RAM, caching, buffers size, and the resident set size (RSS) of processes of interest.
– *Other resources* include CPU, power consumption, number of threads/processes, storage space, and network metrics. In cloud systems, indicators can be measured at any layer of the virtualization stack, e.g., at application/OS layer within the VM, at VM layer to probe the state and resource consumption of the VMs (e.g., for load balancing, scaling, VM migration ,and rejuvenation decisions) and at VMM layer. Besides system resources, indicators of interest include the number of VM allocations/releases, the VM start/stop time, and the time to migration. An example is the number of orphan VMs

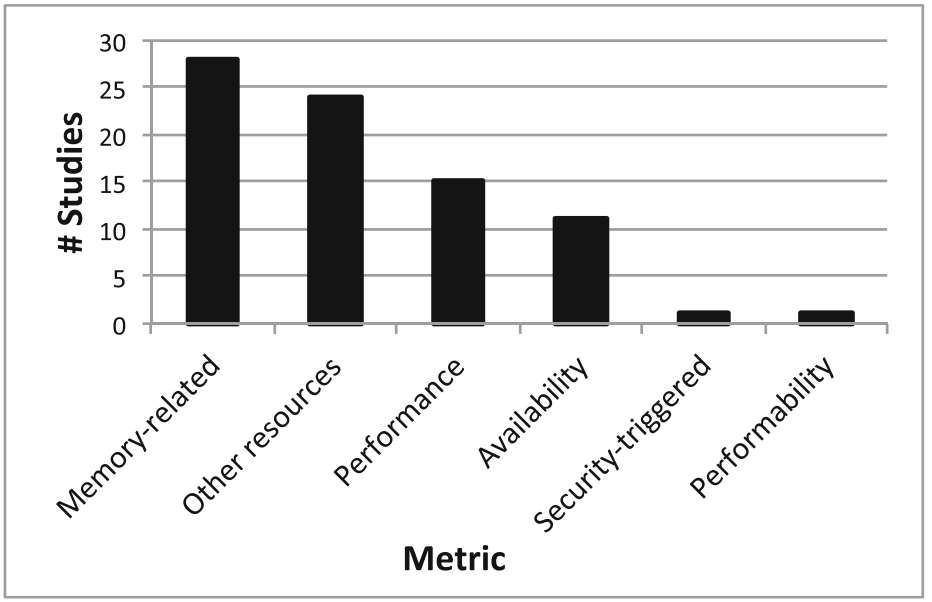


Fig. 13 Metrics used in measurement-based studies

measured by Dabrowsk and Mills (2011) in order to quantifying the leakage of VMs caused by attacks.
- User-perceived performance degradation is the third most adopted indicator (15/42), the most common indicators being the throughput, response time/job completion time, latency, and also number of SLA violations.
- Dependability attributes also appear in these studies, but never alone (as they are not the direct subject of "measurement"). Availability is the most common case with 11 occurrences.

In other cases, the study did not mention a specific resource to measure, but generically mentions a resource monitoring task from which aging can be assessed (such as elastic increase and decrease of resource capacity, virtual machine migrations, dynamic resource configuration changes).

4.2.3 Hybrid analysis techniques

An important generalization of the previous two methods for aging analysis is what are called hybrid approaches, proposed by some researchers as a combination of model-based and measurement-based solutions. Hybrid solutions adopt a stochastic model to describe the phenomenon, and determine the model parameters through measurement, that is, via observed data. Solutions of this type, although not much common, are able to put together the advantages of both approaches. Eight out of 105 cloud-related studies adopted this strategy. As for modeling, they adopt 8 different formalisms (SPN, SRN, SMP, RBD, CTMC, a non-markovian stochastic process, an optimization model, and a *Markov Renewal Processes*, MRP). As for measurements, four of them use time series analysis for prediction-based rejuvenation, and four use a threshold-based approach. As aging indicators, six use availability coupled with either memory-related or performance degradation indicators; one adopts reliability; one adopts only memory indicators. An example is the hybrid solution by Liu et al. (2015): they measure the trends of various resources (CPU, storage, network) at several layers in a cloud-based streaming system with ATM endpoints, and use them to parametrize a model to schedule rejuvenation.

In the above classes of studies, a further analysis factor is workload dependency. Since aging has been shown to be correlated with workload (Bovenzi et al. 2011, 2012), some studies accounted for its impact. An example is by Bruneo et al., who present a workload-based analysis of VMM aging and rejuvenation under different policies for availability maximization (Bruneo et al. 2013). They exploit dynamic reliability theory and symbolic algebraic techniques, representing the CDFs associated with the VMM events by continuous phase-type (CPH) distributions, and using the Kronecker algebra to implement the conservation of reliability principle and the variable timer policy.

The solution proposed by Kadirvel and Fortes (2010) is applied to a system manager for a batch-based job submission system on a virtualized platform, aimed at managing and controlling virtualized resources to support remediation approaches (such as elastic increase and decrease of resource capacity, VM migrations, dynamic resource configuration changes). The technique combines a Petri net–based approach to model a system manager module, suffering from health deterioration due to resource exhaustion, with the usage of feedback control theory to control resource consumption and delay/prevent resource. Three different rejuvenation strategies are implemented and tested—process rejuvenation, VM migration, and dynamic increase of resource allocation, chosen based on the planning module.

The study by Zhao et al. formulates the problem by a game method, representing the different goals of a service provider (who wants to maximize availability) and a maintainer (who wants to minimize cost), and uses MRP to determine the optimal rejuvenation schedule and compute the steady-state availability and maintenance cost (Zhao et al. 2014).

4.3 Rejuvenation

This section deals with the studies concerning the rejuvenation techniques proposed for cloud systems. With reference to the classification in Fig. 1, there are studies falling in category 2 (*Rejuvenation Techniques*) and category 3—for a total of 76 papers corresponding, respectively, to the papers with the following identifiers: from P-29 to P-44 and from P-45 to P-104 in the mentioned list of papers.

Software rejuvenation can act on the virtualization infrastructure (VI) and/or on the VMs. About the VI rejuvenation, the key component subject to rejuvenation is most of the times the VMM. Alternative strategies depend on whether rejuvenation affects only the VMM, or also the VMs running on it (Machida 2013).

Figure 14 presents a taxonomy of rejuvenation techniques at VMM level and at VM level. Figures 15 and 16 show the breakdown of papers by rejuvenation technique for VI/VMM and VMs, respectively.

VMM can be rejuvenated by the following techniques (Machida 2013):

- **Cold-VM rejuvenation**, the simplest and most used approach, which simply shuts down the hosted VMs before triggering the VMM rejuvenation (e.g., via restart) and then restarts the VMs after the completion of VMM rejuvenation. This indirectly rejuvenates also the VMs.
- **Warm-VM rejuvenation**, in which the execution state of VMs is stored to persistent memory and resumed after the restart of the VMM, so as to reduce the downtime of VMs compared with cold-VM. Though, in this case, VMs are not rejuvenated. This operation can be quickly performed by an on-memory suspend/resume mechanism (the memory image of VMs is preserved in RAM during the VMM restart rather than on persistent storage Kourai and Chiba 2011). Kourai et al. propose such a variant and show

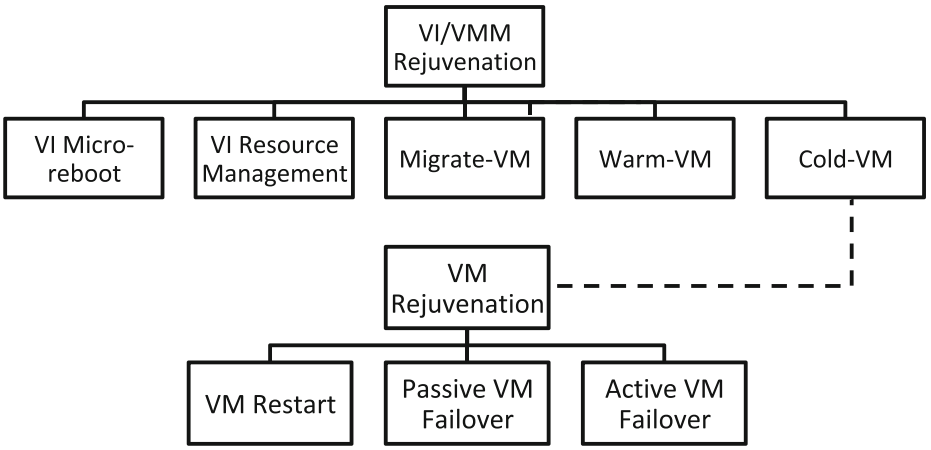


Fig. 14 A taxonomy of rejuvenation techniques in virtualized environments

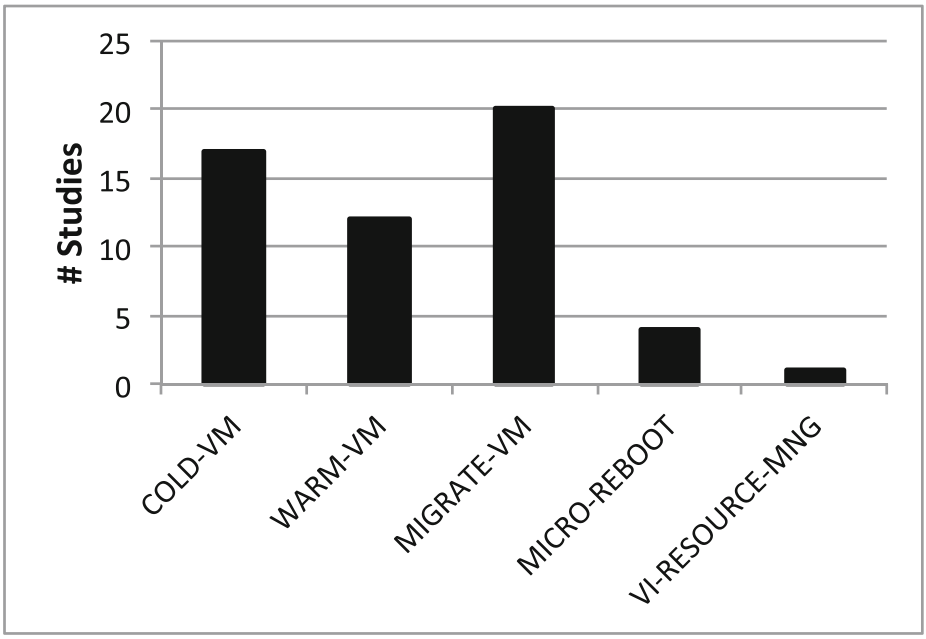


Fig. 15 VI/VMM rejuvenation techniques

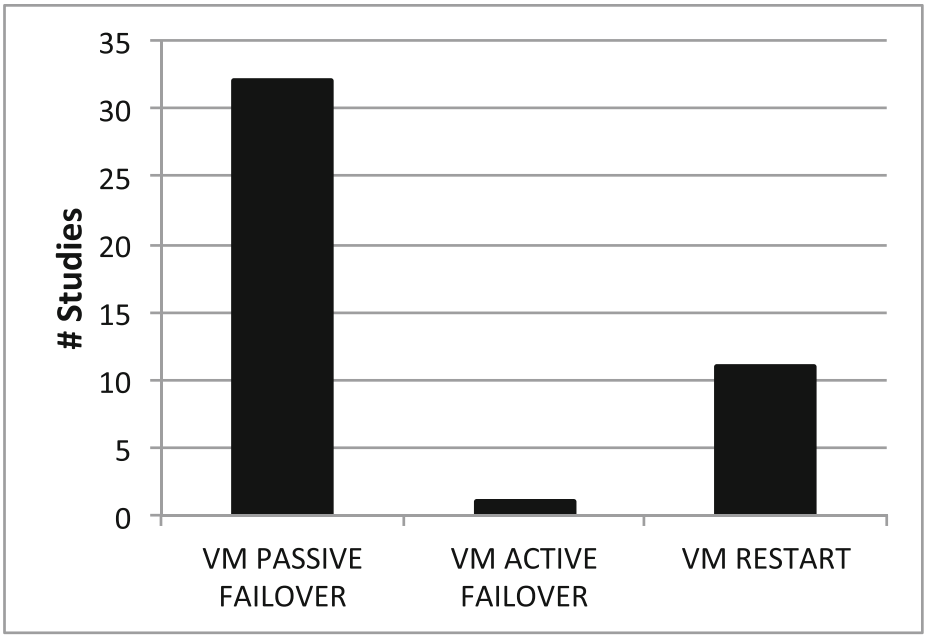


Fig. 16 VM rejuvenation techniques

that compared with cold-VM rejuvenation, warm-VM reboot improves the availability of the application hosted on VMs (Kourai 2007; Kourai and Chiba 2011).

– **Migrate-VM rejuvenation**, in which the downtime is further reduced compared with previous strategies, by migrating a VM to another host during the VMM rejuvenation, making VMs available. This does not rejuvenate VMs and is limited by the capacity of other hosts to accept migrated VMs. The technique can be divided based on the type of live VM migration (*stop-and-copy* or *pre-copy*), and the policies to return back to the original host (*return-back* or *stay-on*). The *stop-and-copy* migration stops the VM operations and copies the memory contents to the destination server. In the *pre-copy* variant, the VM memory is copied to the destination server without stopping its operation, hence causing dirty pages in the copied memory that are then updated by a stop-and-copy approach, so considerably reducing the downtime overhead compared with a complete stop-and-copy of the entire VM's memory. At VM restore, a *return-back* policy migrates the VMs back to the original host soon after the VMM rejuvenation; a *stay-on* policy allows the migrated VM to keep on running on the new server.

  The work by Machida (2013) uses SRNs for cold-VM rejuvenation, warm-VM rejuvenation, and migration. They studied the steady-state availability of VMs and the expected number of transactions lost, finding that migrate-VM rejuvenation generally achieves higher steady-state availability compared with cold- and warm-VM rejuvenation, because of the ability to preserve the VM execution even during VMM rejuvenation. Moreover, they found that using pre-copy migration is generally better than the pure stop-and-copy migration, and using the return-back policy for migrating back is generally more effective than the stay-on policy.

  A combined strategy makes sense, with different policies on different VMs depending on their aging state and criticality. In Machida et al. (2012a), the authors propose the contemporary rejuvenation of aged VMs and the VMM in virtualized data centers, forcing the shutdown only of aged VMs, while the robust VMs are moved out from the host server by live VM migration before VMM rejuvenation. Kourai et al. present *VMBeam*, a technique using *zero-copy* migration. It starts a new virtualized system at the same host by using nested virtualization, and then migrates all the VMs from the aged virtualized system to the clean one by relocating the VMs memory, without any copy (Kourai and Ooba 2015). The approach proposed by Torquato et al. foresees a rejuvenation phase performed by means of VM Live Migration, applied on Cloud Computing testbed which uses OpenNebula and KVM as VMM (Torquato et al. 2017).

  A further distinction is between *cold* and *warm migration*: the VM can be migrated to a physical standby node that is shut down (cold standby), or to a node up but not running virtual machines and with the VMM not accumulating aging (warm standby) (Torquato et al. 2018).

Rejuvenation at VI level is done also beyond the VMM, by:

– **VI Micro-reboot**. This is a sophisticated technique to perform a fine-grained reboot of software modules (micro reboot), tailored for the VI software. Le and Tamir (2012) applied micro-reboot to all modules of the Xen virtualization software, consisting of three main components: the privileged virtual machine, the device driver virtual machine, and the VMM.
– **VI Resource Management**. Dabrowsk et al. aim to detect and eliminate the number of orphan VMs, to quantify and reduce the leakage of VMs caused by attacks (Dabrowsk and Mills 2011). Although the cause of aging is a security attack in this case,

the approach can be generalized to improve the availability of the VI by periodically cleaning the resources it manages.

The best technique (or the best combination of them) depends on the speed of storing/migrating the state of VMs and on the capacity of hosts, as well as on the aging rate of VMs and VMMs, therefore the rejuvenation policy should be determined according to these factors. Migration is the most used approach with 20 occurrences, many of them (11/20) in the last two years, followed by cold-VM with 17 occurrences and warm-VM with 12 occurrences. Micro-reboot appears 4 times, which is a non-negligible percentage.

As for rejuvenation of the VMs, conventional replication approaches (besides the mentioned Cold-VM) can be applied:

– **VM Failover** is a replication-based approach. The idea is to redirect, upon detection of aging in a VM, all the incoming requests to a another VM, and then rejuvenate the aged VM, e.g., by restart. *Passive* (also known as primary-backup or passive/active) strategies are distinguished based on the replication policy as *cold*, *warm*, and *hot* standby. In *cold* standby, the standby node is completely powered off and it is initiated only when the primary node fails; in *warm* standby, the standby node is up and running with data mirrored regularly but with the replicated software being off; and in *hot* standby, the standby node is up and running with data mirrored in near real time and the replicated software is also up and running but without processing data or requests. *Active* failover (also known as active/active) foresees that both replicas are up and running and process data or requests in parallel.

  The most common strategy we encountered is by far the *passive* failover. An example is the mentioned framework, *VM-Rejuv*: a *load balancer* VM redirects requests to an *active* VM while it is correctly working and monitors it for aging; when rejuvenation is triggered, new requests are redirected to a *standby* VM (Silva et al. 2009). Chang et al. model rejuvenation at both VMM- and VM-level, using, for the latter, a passive VM failover with replicas on a same host, kept synchronized by a heartbeat mechanism (Chang et al. 2016); the active server is rejuvenated only after all pending requests have been processed and session data have been migrated to the standby server, in order to assure a clean restart (i.e., rejuvenation does not cause the loss of session data and the failure of user requests). In this and other cases (e.g., Xu et al. 2014a), researchers refer to *cold* and *hot* standby as, respectively, *passive* and *active* strategy, to distinguish the cases in which the standby is off or is running with the state transferred at the end of each operation or periodically. A "pure" *VM Active Failover* is rarely encountered. We found it only in the work by Tan et al., where active replicas (i.e., contemporarily processing requests) are used for intrusion tolerance, thus controlling the resource consumption by restarts, preventing the VMs from getting aged (Tan et al. 2010).

  In contrast, passive failover is used in 32 cases: 16/32 studies do not specify if hot or cold standby is implemented; in 4/32 studies, both approaches are explicitly mentioned; in 8/32 studies, the hot standby mechanisms is implemented, while in the remaining 4/32 cases, only cold standby is implemented. Most of times, failover is meant between VMs in a same host; in some cases, failover is also explicitly foreseen from a VM in one physical host to a VM in another physical host (e.g., Chen et al. 2018).
– **VM Restart**. Earlier studies applied raw restart of VMs, without any failover and no other means to reduce the downtime. This differs from *cold-VM* just in the objective: *Cold-VM* aims at rejuvenating the VMM; in *VM Restart*, the goal is just to rejuvenate the VM. However, the action at VM level is exactly the same, i.e., a restart.

Besides rejuvenation, a further remediation action applied to VMs is *life extension*, in which a VM is allocated additional memory upon aging detection in order to temporarily prolong its life (Machida et al. 2012b). An approach specific for mobile devices is *offloading*: aging can be mitigated by offloading part of the computation to a remote server in the cloud (Wu and Wolter 2015). These however are not classified as a conventional rejuvenation action, but as a (valuable) alternative remediation. Finally, note that other typical strategies, such as OS/node reboot, application/component restart, and micro-reboot, can be applied to hosts and/or to applications running in the VMs, but they are not specific to VIs or VMs. VMs can also be used to support rejuvenation at other layers: Kourai et al. propose *CacheMind* to handle the problem of inconsistency of in-memory file cache with disk after an OS reboot by running the OS in a virtual machine, and by using the VMM to keep track of the status of file cache pages, so as to guarantee consistency after reboot (Kourai 2010). In this case, in which an OS runs in a VM and is rejuvenated by an OS reboot, the virtual machine infrastructure (i.e., the VMM and the VM) is not affected by rejuvenation.

4.4 Validation

Figure 17 reports the methods chosen by researchers for validating their proposals. In most cases, researchers use experimentation (51/105), followed by numerical analysis (44/105). In very few cases (6/105) they prefer simulation, while in fewer cases (4/105), no form of validation is performed (e.g., short papers or fast abstracts). Despite its cost compared with numerical analysis, experimentation is going to be the preferred way to validate aging and rejuvenation studies. This is particularly true for measurement-based approaches and by "rejuvenation only" studies as well: 36/51 experimentation cases are measurement-based studies and 4/51 are hybrid studies, which still involve measurement-based analyses; 12/51

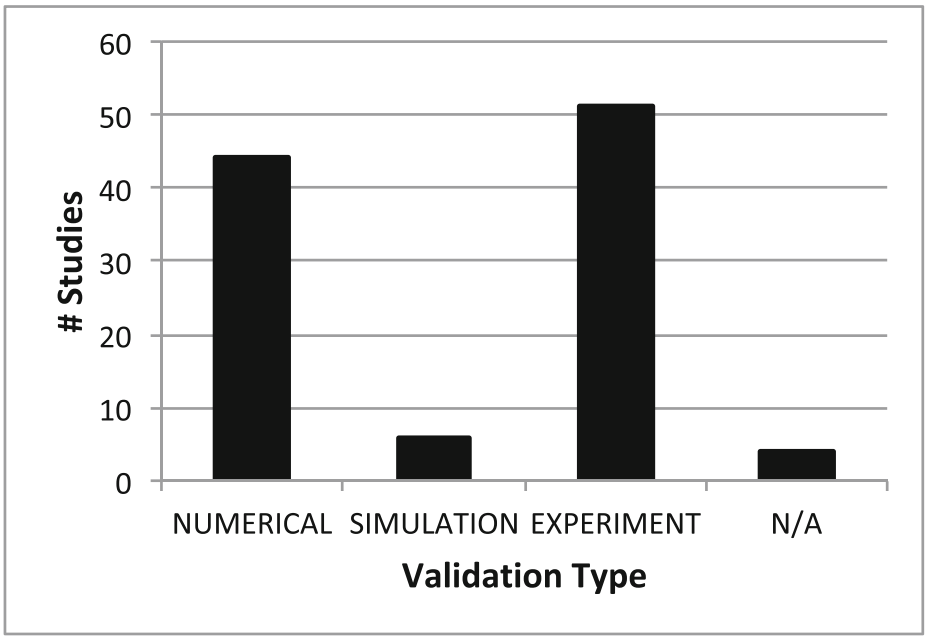


Fig. 17 Validation methods used

cases are "rejuvenation only" studies, and just 1 cases is a model-based study. In contrast, model-based studies prefer the numerical analysis as form of validation, and this is justified by the higher flexibility given by numerical analysis, allowing an easier evaluation of modeling alternatives, parameters tuning and sensitivity analysis. Thirty-seven out forty-four studies using numerical analysis are model-based ones, while 3/44 use measurement-based analysis, 3/44 are hybrid studies, and 1/44 is a "rejuvenation only" study.

5 Discussion

Reported results outline the main trends of SAR research devoted to virtualization and cloud computing. We summarize the main findings and provide further insights to better interpret the results and find room for future research.

About **RQ1**:

- SAR in the cloud is a relatively young research area—all the studies appeared after 2007, since the spread of virtualization and cloud computing—with a considerable number of studies, 105 in 11 years. In the last few years, the publications trend is increasing, as most of the studies (61/105, i.e., 58%) appeared after 2013.
- Journal publications exhibit an increasing trend: 26/61 studies in 2013–2018 are in journals (26/61 in conferences and 9/61 in workshops), while only 3 journal papers appeared before 2013. Assuming that journal papers require, in the average, a higher quality than conference and workshop papers, the trend in the last years highlights that also the quality of cloud-related SAR research is increasing.
- The SAR community focusing on cloud systems is represented by 186 different authors that published in 74 different venues, 87 different institutions, and 21 different countries with researchers involved in at least one cloud-related SAR publication. Moreover, assuming that authoring more papers indicates a stronger commitment on the topic (while authoring just one paper may be contingent), the number of researchers and research groups engaged more actively is encouraging—authors publishing at least two studies are about 25% of the total (48/186), and those co-authoring more than two studies are 15% (28/186); 44% of the institutions (39/87) appear in at least two studies and 18% in more than two studies (16/87).
- The impact of papers in terms of citations is not as high as expected, considering the spread of ever-running cloud systems. The average is about 4 citations each, with the top-cited paper achieving 51 citations in 5 years. The trend, even normalized over the years, is quiet stable.

  These results together highlight that the research field is well-rooted in practice and there is a stable community underpinning this research. Looking at the increasing trends along with the continuous expansion of cloud-related technologies and applications, it is easy to foresee a further increase of cloud SAR in the next few years. On the other hand, although a closer inspection of citations flow would be needed, the observed citation trends suggest that more work is needed to adequately inform researchers of other communities close to cloud computing about the aging phenomenon and its impact—publishing in cloud conferences and journals is a good point toward this direction.

About **RQ2**:

- Aging analysis by model-based techniques has considerable interest. These techniques naturally support the evaluation of alternatives: in the cloud domain, most of times

(35 out of 40, i.e., 87.5%), they are associated with the evaluation of several *rejuvenation* techniques, for which proper time/cost parameters are considered on different rejuvenation alternatives. The high number of studies about rejuvenation and the variety of explored techniques push researchers to develop new and more comprehensive models to account for cost/benefits of rejuvenation actions at several layers.

– There is an increasing trend in measurement-based approaches in the cloud domain, with even more papers than model-based ones (while results of a survey on all the SAR area in 2014 reported more model-based works Cotroneo et al. 2014). The analyzed systems range from cloud applications (often web/application servers running on the cloud) to entire platforms, the most studied one being Eucalyptus. Likely, the wide availability of cloud computing software favors the experimentation of such strategies without excessive cost—a factor that could foster an increase of measurement-based analyses in the cloud domain in the future. Adopted techniques are basically the same as in the conventional SAR literature. Metrics differ slightly because of the need of capturing behaviors related to the VM management and/or of differentiating resources of virtual and physical host, but much more can be done to identify aging indicators at VI/VMM layer or proper combination of indicators at different layers.
– Hybrid studies are the minority (8/105) and in a still relatively small number. Even though they are indeed more difficult and expensive to perform, we claim that they deserve greater attention because of the ability to combine benefits of both the previous two approaches. They describe the aging dynamics and the effects of rejuvenation strategies in a more accurate way than pure model- or measurement-based techniques—a need particularly felt for cloud systems, whose complexity requires a better prediction ability.

About **RQ3**:

– Rejuvenation in the cloud domain is addressed by many studies. Overall, 76/105 studies (i.e., 72.3%) include rejuvenation in their proposal, while 16/105 deal exclusively with a rejuvenation technique without caring for aging assessment. Indeed, the additional complexity and, at the same time, the possibilities offered by virtualization push researchers for devising several new options for rejuvenation at several layers able to drastically reduced the cost. In such studies, researchers often analyze several rejuvenation policies based on virtual machine and/or virtual machine monitor reboot/rejuvenation. The increasing economical impact of performance of cloud-based systems will also likely push toward this direction.

  Research opportunities include investigating carefully costs and benefits of replication strategies (including the scarcely addressed active replication), not only in relation to downtime reduction, but also considering the cost of setting up and managing replicas, their energy consumption, the side effects on security and on scalability, and so on. The overhead brought by rejuvenation is worth to be investigated too, and possibly reduced, as the advantages of using VMs have been shown to be counterbalanced by a higher memory fragmentation (Alonso et al. 2013). This stresses the importance of choosing the right set of techniques for applying rejuvenation in a virtualized environment, thus calling for new and more comprehensive models to account for cost/benefits of rejuvenation actions and their combination at several layers. Alternative remediation approaches, like life extension or offloading, are worth to be investigated too.
– As for rejuvenation techniques, migration is gaining high popularity (11/20 migration proposals appeared in 2016–2018) for its ability to reduce downtime, despite its greater complexity with respect to the simpler warm-VM and, especially, cold-VM.

Micro-reboot is also worth investigation as a very promising approach, e.g., for components in the virtualization technologies. Containerization is likely to further change this scenario and introduce more opportunities for rejuvenation. About rejuvenation at VM layer, the cost/benefit ratio of replication-based (passive/active) strategies is worth to be further investigated, with respect to downtime reduction as well as to power consumption, security, and scalability.

About **RQ4**:

– We looked at the type of validation too. Usually, model-based techniques are validated by means of numerical analyses, since collection of runtime data for all or part of the cases being modeled is expensive and not enough representative of what is modeled. In contrast, measurement-based works prefer experiments. This is in line with the general SAR literature.

  Despite the number of experimental studies, an aspect to improve is the involvement of industry. For SAR solutions to increase their popularity and usefulness, more real-world experiences are needed. A further direction is to develop tools for aging assessment and rejuvenation, which would greatly help non-experts to know about and deal with this phenomenon.

Finally, further opportunities can be caught by looking at the following aspects:

– Emerging attributes related with aging causes and/or effects are being considered by some studies. Security is one of this: overall, 7/105 studies (6.7%) are related to security, wherein the cause of aging are attacks and intrusions. Energy consumption is a further example, very relevant for cloud contexts. Seven out of one hundred five papers (6.7%) dealt with the effect of aging in terms of energy consumption. These are worth to be investigated further.
– As more attributes become important, such as security and energy consumption besides availability and performance, the adoption of multi-objective optimization models for aging mitigation and/or rejuvenation is worth to be explored, for a better allocation of resources. For instance, depending on the relative weight of the attributes of interest (e.g., energy, security, performance), some rejuvenation actions may be preferred to others (e.g., prefer the cheaper cold-VM approach to migrate-VM for some VMs, or decide to migrate one VM rather than another, depending on how these actions affect the attribute of interest and at what cost). This would require preliminarily characterizing rejuvenation actions according to their relation with these attributes, in order to figure out what action is more likely to optimize a given attribute and at what cost.
– SAR in emerging contexts related to the cloud/virtualization is worth to be explored, such as: edge and fog computing, internet of things, network function virtualization and software-defined networks are some fields which could benefit from research on aging and rejuvenation counteractions.

6 Threats to validity

In the following, the main threats to validity are reported. In the process of selecting the primary studies, some studies could have been missed (as pointed out by Wohlin et al. 2013, two mapping studies of the same topic end up with different sets of articles). To mitigate this threat and have a representative sample, a well-defined search strategy has been followed, consisting of both automatic search and manual inspection on the selected studies. The

search was conducted by querying multiple data sources, so as to cover a high number of publishers. Additionally, the terms used in the search string are very general. This allowed to be very conservative in the first search (which included 298, 367, and 202 from the three libraries) so as to not miss relevant studies. The resulting set was then manually filtered to discard irrelevant results up to 219 studies. Moreover, a set of documented inclusion and exclusion criteria was used to refine the search unambiguously.

Another threat can regard the quality of selected studies. To mitigate this threat, (i) only peer-reviewed studies have been considered (excluding the so-called grey literature, e.g., white papers, editorials); (ii) studies are searched among those indexed by the most used digital libraries in computer engineering and computer science, which filter out numerous low-quality conferences and journals; and (iii) inclusion/exclusion criteria applied on each of the 219 studies after full-text reading assured to keep only studies pertinent to SAR in the cloud. The analysis of the characteristics of selected studies to answer research questions could be biased by a subjective interpretation. To reduce such a risk, we adopted a well-known and accepted scheme for classifying studies unambiguously as model-based, measurement-based, and hybrid studies. For rejuvenation studies, a classification has been derived by extending an existing one (Machida 2013), by iteratively refining an initial scheme as more and more studies were analyzed. More generally, we have been followed in each phase the best practices for this kind of studies (Kitchenham and Brereton 2013; Petersen et al. 2015).

7 Conclusions

This paper zoomed in on software aging and rejuvenation research in the cloud domain. We searched for research studies proposing methods and solutions to assess software aging, predict the best time for rejuvenation, perform rejuvenation actions by finding the best tradeoff between cost and benefit.

Looking at the increasing trends of cloud computing, it is easy to envision a proportional increase in the studies dealing with SAR in the cloud. The rapid evolution of cloud technologies is likely to bring further new challenges, with wide room for future research in this area. Results of this work help to outline an overview of current trends which researchers are following, what strategies are deemed more appropriate for aging analysis, and what countermeasures can better suite to the cloud domain. This can serve as a basis to plan for future SAR research in the cloud.

Funding information This work has been partially supported by the GAUSS Italian research project, funded by MIUR under the PRIN 2015 program.

References

Alonso, J., Matias, R., Vicente, E., Maria, A., Trivedi, K. (2013). A comparative experimental study of software rejuvenation overhead. *Performance Evaluation*, *70*(3), 231–250.

Araujo, J., Matos, R., Maciel, P., Vieira, F., Matias, R., Trivedi, K. (2011). Software rejuvenation in Eucalyptus cloud computing infrastructure: a method based on time series forecasting and multiple thresholds. In *Third international workshop on software aging and rejuvenation (WoSAR)* (pp. 38–43). IEEE.

Araujo, J., Matos, R., Alves, V., Maciel, P., Vieira de Souza, F., Matias, R.Jr., Trivedi, K.S. (2014). Software aging in the Eucalyptus cloud computing infrastructure: characterization and rejuvenation. *ACM Journal on Emerging Technologies in Computing Systems*, *10*(1), 11:1–11:22.

Avresky, D.R., Sanzo, P.D., Pellegrini, A., Ciciani, B., Forte, L. (2015). Proactive scalability and management of resources in hybrid clouds via machine learning. In *14th International Symposium on Network Computing and Applications (NCA)* (pp. 114–119). IEEE.

Bartoletti, D. (2019). Predictions 2019: cloud computing comes of age as the foundation for enterprise digital transformation. [Online]. Available: https://go.forrester.com/blogs/predictions-2019-cloud-computing/.

Bovenzi, A., Cotroneo, D., Pietrantuono, R., Russo, S. (2011). Workload characterization for software aging analysis. In *22nd International Symposium on Software Reliability Engineering (ISSRE)* (pp. 240–249). IEEE.

Bovenzi, A., Cotroneo, D., Pietrantuono, R., Russo, S. (2012). On the aging effects due to concurrency bugs: a case study on MySQL. In *2012 IEEE 23rd International Symposium on Software Reliability Engineering* (pp. 211–220).

Bruneo, D., Distefano, S., Longo, F., Puliafito, A., Scarpa, M. (2013). Workload-based software rejuvenation in cloud systems. *IEEE Transactions on Computers*, *62*(6), 1072–1085.

Cavezza, D.G., Pietrantuono, R., Alonso, J., Russo, S., Trivedi, K.S. (2014). Reproducibility of environment-dependent software failures: an experience report. In *2014 IEEE 25th International Symposium on Software Reliability Engineering* (pp. 267–276).

Chang, X., Zhang, Z., Li, X., Trivedi, K.S. (2016). Model-based survivability analysis of a virtualized system. In *IEEE 41st Conference on Local Computer Networks (LCN)* (pp. 611–614).

Chen, Z., Chang, X., Han, Z., Li, L. (2018). Survivability modeling and analysis of cloud service in distributed data centers. *The Computer Journal*, *61*(9), 1296–1305.

Columbus, L. (2018). Forrester's 10 cloud computing predictions for 2018. [Online]. Available: https://www.forbes.com/sites/louiscolumbus/2017/11/07/forresters-10-cloud-computing-predictions-for-2018/#58f8da6d4ae1.

Cotroneo, D., Natella, R., Pietrantuono, R. (2012). Predicting aging-related bugs using software complexity metrics. *Performance Evaluation*, *70*(3), 163–178.

Cotroneo, D., Orlando, S., Pietrantuono, R., Russo, S. (2013). A measurement-based ageing analysis of the JVM. *Software Testing, Verification and Reliability*, *23*(3), 199–239.

Cotroneo, D., Natella, R., Pietrantuono, R., Russo, S. (2014). A survey of software aging and rejuvenation studies. *ACM Journal on Emerging Technologies in Computing Systems*, *10*(1), 8:1–8:34.

Cotroneo, D., Fucci, F., Iannillo, A.K., Natella, R., Pietrantuono, R. (2016). Software aging analysis of the Android mobile OS. In *2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 478–489).

Dabrowsk, C., & Mills, K. (2011). VM leakage and orphan control in open-source clouds. In *IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 554–559). IEEE.

DeCelles, S., Huang, T., Stamm, M.C., Kandasamy, N. (2016). Detecting incipient faults in software systems: a compressed sampling-based approach. In *9th IEEE International Conference on Cloud Computing (CLOUD)* (pp. 303–310). IEEE.

Escheikh, M., Tayachi, Z., Barkaoui, K. (2016). Workload-dependent software aging impact on performance and energy consumption in server virtualized systems. In *27th International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 111–118). IEEE.

Escheikh, M., Barkaoui, K., Jouini, H. (2017). Versatile workload-aware power management performability analysis of server virtualized systems. *Journal of Systems and Software*, *125*, 365–379.

Ficco, M., Pietrantuono, R., Russo, S. (2018). Aging-related performance anomalies in the Apache storm stream processing system. *Future Generation Computer Systems*, *86*, 975–994.

Grottke, M., Matias, R., Trivedi, K. (2008). The fundamentals of software aging. In *IEEE International Conference on Software Reliability Engineering Workshops*.

Han, L., & guo Xu, J. (2013). Availability models for virtualized systems with rejuvenation. *Journal of Computational Information Systems*, *9*(20), 8389–8396.

Huang, T., Kandasamy, N., Sethu, H., Stamm, M. (2016). An efficient strategy for online performance monitoring of datacenters via adaptive sampling. IEEE Transactions on Cloud Computing (Early Access).

Kadirvel, S., & Fortes, J.A.B. (2010). Self-caring IT systems: a proof-of-concept implementation in virtualized environments. In *IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 433–440). IEEE.

Kitchenham, B., & Brereton, P. (2013). A systematic review of systematic review process research in software engineering. *Information and Software Technology*, *55*(12), 2049–2075.

Kourai, K. (2007). A fast rejuvenation technique for server consolidation with virtual machines. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 245–255). IEEE.

Kourai, K. (2010). CacheMind: fast performance recovery using a virtual machine monitor. In *2010 International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 86–92). IEEE.

Kourai, K., & Chiba, S. (2011). Fast software rejuvenation of virtual machine monitors. *IEEE Transactions on Dependable and Secure Computing*, *8*(6), 839–851.

Kourai, K., & Ooba, H. (2015). Zero-copy migration for lightweight software rejuvenation of virtualized systems. In *Proceedings of the 6th Asia-Pacific Workshop on Systems (APSys)* (pp. 7:1–7:8). ACM.

Le, M., & Tamir, Y. (2012). Applying microreboot to system software. In *Sixth International Conference on Software Security and Reliability (SERE)* (pp. 11–20). IEEE.

Liu, F., Cao, J., Guo, J., Zhang, B. (2013). Research the measurement method of software aging in cloud. *Applied Mechanics and Materials*, *392*, 779–782.

Liu, Y., Liu, W., Song, J., He, H. (2015). An empirical study on implementing highly reliable stream computing systems with private cloud. *Ad Hoc Networks*, *35*(C), 37–50.

Machida, F. (2013). Modeling and analysis of software rejuvenation in a server virtualized system with live VM migration. *Performance Evaluation*, *70*(3), 212–230.

Machida, F. (2014). Job completion time on a virtualized server with software rejuvenation. *ACM Journal on Emerging Technologies in Computing Systems*, *10*(1), 10:1–10:26.

Machida, F., Kim, D.S., Trivedi, K.S. (2010). Modeling and analysis of software rejuvenation in a server virtualized system. In *Second International Workshop on Software Aging and Rejuvenation (WoSAR)*. IEEE.

Machida, F., Nicola, V.F., Trivedi, K.S. (2011). Job completion time on a virtualized server subject to software aging and rejuvenation. In *Third international Workshop on Software Aging and Rejuvenation (WoSAR)* (pp. 44–49). IEEE.

Machida, F., Xiang, J., Tadano, K., Maeno, Y. (2012a). Combined server rejuvenation in a virtualized data center. In 9th International Conference on Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC) (pp. 486–493). IEEE.

Machida, F., Xiang, J., Tadano, K., Maeno, Y. (2012b). Software life-extension: a new countermeasure to software aging. In *23rd International Symposium on Software Reliability Engineering (ISSRE)* (pp. 131–140). IEEE.

Melo, M., Araujo, J., Matos, R., Menezes, J., Maciel, P. (2013a). Comparative analysis of migration-based rejuvenation schedules on cloud availability. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 4110–4115). IEEE.

Melo, M., Maciel, P., Araujo, J., Matos, R., Araujo, C. (2013b). Availability study on cloud computing environments: live migration as a rejuvenation mechanism. In *43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE.

Mohan, B.R., & Reddy, G.R.M. (2015). The effect of software aging on power usage. In *9th International Conference on Intelligent Systems and Control (ISCO)*. IEEE.

Myint, M., & Thein, T. (2010). Availability improvement in virtualized multiple servers with software rejuvenation and virtualization. In *Fourth International Conference on Secure Software Integration and Reliability Improvement (SSIRI)* (pp. 156–162). IEEE.

Nguyen, T.A., Kim, D.S., Park, J.S. (2014). A comprehensive availability modeling and analysis of a virtualized servers system using stochastic reward nets. The Scientific World Journal.

Okamura, H., Yamamoto, K., Dohi, T. (2014). Transient analysis of software rejuvenation policies in virtualized system: phase-type expansion approach. *Quality Technology & Quantitative Management*, *11*(3), 335–351.

Petersen, K., Vakkalanka, S., Kuzniarz, L. (2015). Guidelines for conducting systematic mapping studies in software engineering: an update. *Information and Software Technology*, *64*, 1–18.

Pietrantuono, R., & Russo, S. (2018). Software aging and rejuvenation in the cloud: a literature review. In *29th International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 257–263). IEEE.

Rahme, J., & Xu, H. (2015). A software reliability model for cloud-based software rejuvenation using dynamic fault trees. *International Journal of Software Engineering and Knowledge Engineering*, *25*(09n10), 1491–1513.

Rezaei, A., & Sharifi, M. (2010). Rejuvenating high available virtualized systems. In *5th International Conference on Availability, Reliability, and Security (ARES)* (pp. 289–294). IEEE.

Silva, L., Alonso, J., Torres, J. (2009). Using virtualization to improve software rejuvenation. *IEEE Transactions on Computers*, *58*(11), 1525–1538.

Simeonov, D., & Avresky, D.R. (2010). Proactive software rejuvenation based on machine learning techniques. In Avresky, D.R., Diaz, M., Bode, A., Ciciani, B., Dekel, E. (Eds.) *Cloud computing. CloudComp 2009, ser. Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering*, (Vol. 34 pp. 186–200): Springer.

Sudhakar, C., Shah, I., Ramesh, T. (2014). Software rejuvenation in cloud systems using neural networks. In *International Conference on Parallel, Distributed and Grid Computing (PDGC)* (pp. 230–233). IEEE.

Sukhwani, H., Matias, R., Trivedi, K.S., Rindos, A. (2017). Monitoring and mitigating software aging on IBM cloud controller system. In *28th International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 266–272). IEEE.

Tan, Y., Luo, D., Wang, J. (2010). CC-VIT: virtualization intrusion tolerance based on cloud computing. In *2nd International Conference on Information Engineering and Computer Science (ICIECS)*. IEEE.

Thein, T., & Park, J.S. (2009). Availability analysis of application servers using software rejuvenation and virtualization. *Journal of Computer Science and Technology*, *24*(2), 339–346.

Thein, T., Chi, S., Park, J.S. (2008). Availability modeling and analysis on virtualized clustering with rejuvenation. *International Journal of Computer Science and Network Security*, *8*(9), 72–80.

Torquato, M., Maciel, P., Araujo, J., Umesh, I.M. (2017). An approach to investigate aging symptoms and rejuvenation effectiveness on software systems. In *12th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE.

Torquato, M., Umesh, I.M., Maciel, P. (2018). Models for availability and power consumption evaluation of a private cloud with VMM rejuvenation enabled by VM live migration. *The Journal of Supercomputing*, *74*(9), 4817–4841.

Umesh, I.M., & Srinivasan, G.N. (2016). Optimum software aging prediction and rejuvenation model for virtualized environment. *Indonesian Journal of Electrical Engineering and Computer Science*, *3*(3), 572–578.

Umesh, I.M., & Srinivasan, G.N. (2017). *Dynamic software aging detection-based fault tolerant software rejuvenation model for virtualized environment, ser. Advances in intelligent systems and computing* (Vol. 469, pp. 779–787). Singapore: Springer.

Villalobos, J.J., Rodero, I., Parashar, M. (2014). Energy-aware autonomic framework for cloud protection and self-healing. In *International Conference on Cloud and Autonomic Computing (ICCAC)* (pp. 3–4). IEEE.

Wohlin, C., Runeson, P., da Mota Silveira Neto, P.A., Engstrom, E., do Carmo Machado, I., de Almeida, E.S. (2013). On the reliability of mapping studies in software engineering. *Journal of Systems and Software*, *86*(10), 2594–2610.

Wu, H., & Wolter, K. (2015). Software aging in mobile devices: partial computation offloading as a solution. In *26th International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 125–131). IEEE.

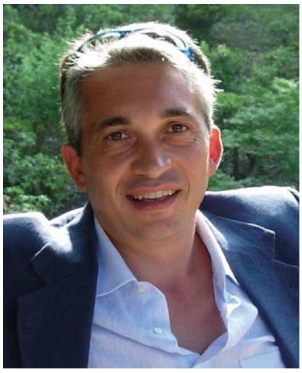
Xia, Y., Han, Y., Zhou, M., Li, J. (2014). A stochastic model for performance and energy consumption analysis of rejuvenation and migration-enabled cloud. In *Proc. of the 2014 International Conference on Advanced Mechatronic Systems* (pp. 139–144). IEEE.

Xu, J., Li, X., Zhong, Y., Zhang, H. (2014a). Availability modeling and analysis of a single-server virtualized system with rejuvenation. *Journal of Software*, *9*(1), 129–139.

Xu, J., wen Wu, W., yi Ma, C. (2014b). SOM-based aging detection for virtual machine monitor. In *IEEE Workshop on Electronics, Computer and Applications* (pp. 782–785). IEEE.

Zhao, J., Wang, Y.-B., Ning, G.-R., Wang, C.-H., Trivedi, K.S., Cai, K.-Y., Zhang, Z.-Y. (2014). Software maintenance optimization based on Stackelberg game methods. In *IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)* (pp. 426–430). IEEE.

Roberto Pietrantuono (SM'16) is Assistant Professor at Federico II University of Naples, where he teaches Software Engineering. His research interests are in the area of software reliability engineering, software testing, and verification of critical software systems. He co-authored more than 60 papers in these areas. He is cofounder of Critiware s.r.l., a company working in critical systems engineering. Since 2008, he has been involved in several EU and national projects on software engineering and software dependability.



Stefano Russo (SM'15) is Professor of Computer Engineering (since 2002) at the Federico II University of Naples, where he teaches Software Engineering and Distributed Systems, and leads the Dependable Systems and Software Engineering Research Team (DESSERT). He is Associate Editor of the IEEE Transactions on Services Computing. He co-authored over 150 papers in the areas of distributed software engineering, middleware technologies, software dependability, and mobile computing.