



# A Systematic Mapping Study in AIOps

Paolo Notaro<sup>1,2</sup> , Jorge Cardoso<sup>2,3</sup>, and Michael Gerndt<sup>1</sup>

<sup>1</sup> Chair of Computer Architecture and Parallel Systems,  
Technical University of Munich, Munich, Germany  
[paolo.notaro@tum.de](mailto:paolo.notaro@tum.de), [gerndt@in.tum.de](mailto:gerndt@in.tum.de)

<sup>2</sup> Ultra-scale AIOps Lab, Huawei Munich Research Center, Munich, Germany  
{[paolo.notaro](mailto:paolo.notaro@huawei.com), [jorge.cardoso](mailto:jorge.cardoso@huawei.com)}@huawei.com

<sup>3</sup> Department of Informatics Engineering/CISUC,  
University of Coimbra, Coimbra, Portugal

**Abstract.** IT systems of today are becoming larger and more complex, rendering their human supervision more difficult. Artificial Intelligence for IT Operations (AIOps) has been proposed to tackle modern IT administration challenges thanks to AI and Big Data. However, past AIOps contributions are scattered, unorganized and missing a common terminology convention, which renders their discovery and comparison impractical. In this work, we conduct an in-depth mapping study to collect and organize the numerous scattered contributions to AIOps in a unique reference index. We create an AIOps taxonomy to build a foundation for future contributions and allow an efficient comparison of AIOps papers treating similar problems. We investigate temporal trends and classify AIOps contributions based on the choice of algorithms, data sources and the target components. Our results show a recent and growing interest towards AIOps, specifically to those contributions treating failure-related tasks (62%), such as anomaly detection and root cause analysis.

**Keywords:** AIOps · Operations and Maintenance · Artificial Intelligence

## 1 Introduction

Modern society is increasingly dependent on large-scale IT infrastructures. At the same time, the latest IT challenges impose higher levels of reliability and efficiency on computer systems. Because of the large increase in size and complexity of these systems, IT operators are increasingly challenged while performing tedious administration tasks manually. This has sparked in recent years much interest towards the study of self-managing and autonomic computing systems to improve efficiency and responsiveness of IT services. While many static algorithmic solutions have been proposed, these automated solutions often show limitations in terms of adaptiveness and scalability. The presence of large data volumes in different modalities motivates the investigation of intelligent learning systems, able to adapt their behavior to new observations and situations.

Artificial Intelligence for IT Operations (AIOps) investigates the use of Artificial Intelligence (AI) for the management and improvement of IT services. AIOps relies Machine Learning, Big Data, and analytic technologies to monitor computer infrastructures and provide proactive insights and recommendations to reduce failures, improve mean-time-to-recovery (MTTR) and allocate computing resources efficiently [3]. AIOps offers a wide, diverse set of tools for several applications, from efficient resource management and scheduling to complex failure management tasks such as failure prediction, anomaly detection and remediation [13,23]. However, being a recent and cross-disciplinary field, AIOps is still a largely unstructured research area. The existing contributions are scattered across different conferences and apply different terminology conventions. Moreover, the high number of application areas renders the search and collection of relevant papers difficult. Some previous systematic works only treat single tasks or subareas inside AIOps [20,31]. This motivates the need for a complete and updated study of AIOps contributions.

In this paper, we present in-depth analysis of AIOps to cover for these limitations. We have identified and extracted over 1000 AIOps contributions through a systematic mapping study, enabling us to delineate common trends, problems and tools. First, we provide an in-depth description of the methodology followed in our mapping study (Sect. 2), reporting and motivating our planning choices regarding problem definition, search, selection and mapping. Then, we present and discuss the results drawn from our study, including the identification of most common topics, data sources, and target components (Sect. 3). Finally, Sect. 4 summarizes the outcomes and conclusions treated in this work.

## 2 Methodology

### 2.1 Systematic Mapping Studies

A systematic mapping study (SMS) is a research methodology widely adopted in many research areas, including software engineering [34]. The ultimate goal of a SMS is to provide an overview of a specific research area, to obtain a set of related papers and to delineate trends present inside such area. Relevant papers are collected via predefined search and selection techniques and research trends are identified using categorization techniques across different aspects of the identified papers, e.g. topic or contribution type. We choose to perform a SMS because we are interested in gathering contributions and obtaining statistical insights about AIOps, such as the distribution of works in different subareas and the presence of temporal trends for particular topics. SMSs have also been shown to increase the effectiveness of follow-up systematic literature reviews [34]. To this end, we have also used our systematic mapping study to collect references for a survey on failure management in AIOps separately published.

### 2.2 Planning

According to the step outline followed in [34], a systematic mapping study is composed of:

- **Formulation**, i.e. express the goals intended for the study through research questions. Equally important is to clearly define the scope of investigation;
- **Search**, i.e. define strategies to obtain a sufficiently high number of papers within the scope of investigation. This comprises the selection of one or more search strategies (database search, manual search, reference search, etc.);
- **Selection** (or screening), i.e. define and apply a set of inclusion/exclusion criteria for identifying relevant papers inside the search result set;
- **Data Extraction and Mapping**, i.e. gather the information required to map the selected papers into predefined categorization scheme(s). Finally, results are presented in graphical form, such as histograms or bubble plots.

The next sections illustrate and motivate our choices regarding these four steps for our systematic mapping study in AIOps.

### 2.3 Formulation

The main goal of this mapping study is to identify the extent of past research in AIOps. In particular, we would like to identify a representative set of AIOps contributions which can be grouped based on the similarity of goals, employed data sources and target system components. We also wish to understand the relative distribution of publications within these categories and the temporal implications involved. Formally, we articulate the following research questions:

**RQ1. What categories can be observed while classifying AIOps contributions in scientific literature?**

**RQ2. What is the distribution of papers in such categories?**

**RQ3. Which temporal trends can be observed for the field of AIOps?**

In terms of scope, we express the boundaries of AIOps as the union of goals and problems in IT Operations when dealt with AI techniques. To circumvent ambiguity about the term AI, we adopt an inclusive convention where we consider AI both data-driven approaches, such as Machine Learning and data mining, as well as goal-based approaches, such as reasoning, search and optimization approaches. However, we mostly concentrate our efforts on the first category due to its stronger presence and connection to AIOps methodologies (e.g. data collection).

### 2.4 Search and Selection

**Selection Criteria.** We start illustrating the selection principles beforehand, so that the discussion will appear clearer when we describe our result collection strategy, composed of search and selection altogether. In terms of inclusion criteria, we define only one relevance criterion, based on the main topic of the document. Following from our discussion on scoping such inclusion criterion comprises two necessary conditions:

**Table 1.** The two keyword sets obtained via PICO used for database search.

AI Keywords	IT Operations Keywords
("AI" OR "artificial intelligence") "classification" "clustering" "logistic regression" "regression" ("DL" OR "deep learning") ("ML" OR "Machine Learning") ("inference" OR "logic" OR "reasoning") ("supervised" OR "unsupervised" OR "semi-supervised" OR "reinforcement") AND ("learning") ("support vector machine" OR "SVM") ("tree" OR "tree-based" OR "trees" OR "forest") (("bayesian" OR "neural" AND "network") (((("hidden" AND "markov") OR "gaussian" AND "mixture")) AND "model") (((("datacenter" OR "data center") AND "management")	("DevOps" OR "site reliability engineering" OR "SRE") ("IT operations") ("anomaly detection" OR "outlier detection") ("cloud computing") ("cloud") ("fault detection" OR "failure detection") ("fault localization" OR "failure localization") ("fault prediction" OR "failure prediction") ("fault prevention" OR "failure prevention") ("log" OR "logs" OR "log analysis") ("metrics" OR "KPI" OR "key performance indicator") ("remediation" OR "recovery") ("root-cause analysis" OR "root cause analysis") ("service desk automation") ("tracing" OR "trace" OR "traces")

- The document references one or more AI methods. These mentions can either be part of the implementation or as part of its discussion/analysis (e.g. in a survey). Any mention to AI algorithms employed by others (i.e. mentioned in the related work section or as baseline comparison) that is not strictly the focus of the document, is not considered valid;
- The document applies its concepts to some kind of IT system management. We therefore exclude papers with no specific target domain or with a target domain outside of IT Operations.

In terms of exclusion criteria, we define the following as exclusion rules:

- The language of the document is not English;
- The document is not accessible online;
- The document does not belong to the following categories: scientific article (conference paper, journal article), book, white paper;
- The main topic of the document is one of the following: cybersecurity, industrial process control, cyber-physical systems, and optical sensor networks.

For the special case of survey and review papers, we consider them relevant as long while carrying out our mapping study, but we then exclude them from our final result set, as these articles are useful to find other connected works through references, but they do not constitute novel contributions to the field.

**Database Search.** For the search process, database search represents the first and most important step, as it aims to provide the highest number of results and perform an initial screening of irrelevant papers. We perform database search in three steps: keywording, query construction and result polling. For keywording we use the PICO technique presented in [34] to derive a set of keywords for AI and a set of keywords for IT Operations. The keywords are listed in Table 1. Then, following our scoping considerations, we construct queries so that they return results where both AI and IT Operations are present. In particular, we apply logic conjunction of keywords across all combinations of the two keyword sets (e.g. "logistic regression" and "cloud computing"). This helps enforcing

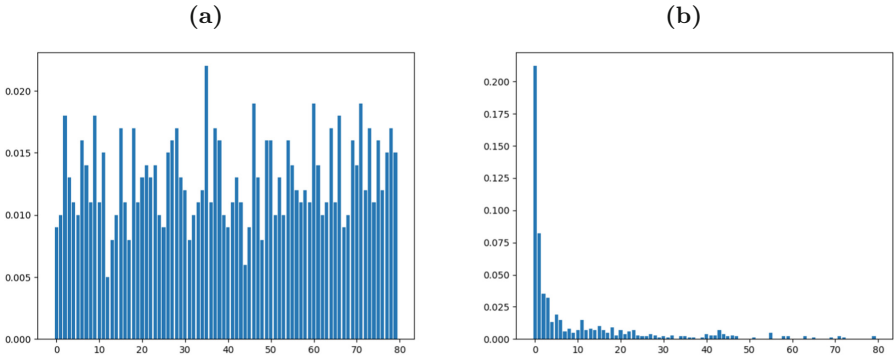
precision in our search results. For keywords with synonyms and abbreviations, we allow all equivalent expressions via OR disjunction. We also perform general search queries, related to the topic as a whole (e.g. “AIOps”). Finally, we group some queries with common terms to reduce the number of queries.

We select three online search databases that are appropriate for the scope of investigation: IEEE Xplore, ACM Digital Library and arXiv. For each query we restrict our analysis to the top 2000 results returned. We aggregate results from all searches in one large set of papers, removing duplicates and annotating for each item corresponding search metadata (e.g. number of hits, index position in corresponding searches, etc.). The result from this step consists of 83817 unique articles. For each item we collect the title, authors, year, publication venue, contribution type and citation count (from Google Scholar).

**Preliminary Filtering and Ranking-Based Selection.** In the filtering step we start improving the quality of our selection of papers. First, papers are automatically excluded based on publication venue, for those venues that are clearly irrelevant for topic reasons (e.g. meteorology). We also exclude based on the year of publication (year < 1990) as it precedes the advent of large-scale IT services. By doing so, we can exclude approximately 8000 elements.

Usually at this point, a full-text analysis would be performed on all the available papers to screen relevant contributions using the above cited selection rules. Although we partly filtered results, it is still not feasible to perform an exhaustive selection analysis, even as simple as filtering by title. It is also impractical to attempt an automated selection by content, as it is not clear how to perform an efficient, high-recall, high-precision text classification without supervision. Therefore, before proceeding with the rest of the search and selection steps, we apply a ranking procedure on these intermediate results, so that we can prioritize investigation of more relevant papers. We apply the exclusion and inclusion rules of Sect. 2.4 to the papers examined in ranking order.

This approximate procedure however raises the question of when it is convenient to stop our selection and discard the remaining items. To solve this, we develop a new approach from our observations of ranked items. We base the method on the following assumption: a considerable ratio of relevant papers can be identified by ranking and selecting top results using different relevance criteria (conference, position index in the query result set, number of hits in all queries, etc.), but in this sorting scenario we also observe a long-tail distribution of relevant documents, i.e. some relevant papers appear in the last positions even after sorting with our relevance heuristics (see Fig. 1). This is coherent with the known impossibility of performing exhaustive systematic literature reviews and mapping studies, as completing the long tail provides less results at the expense of a larger research effort. We assume the ratio of relevant papers in the long tail to be constant and comparable in magnitude to the number of relevant papers when sampled randomly from the result set. Based on this assumption, we proceed as follows:



**Fig. 1.** Estimated relevance probability for collected papers (y-axis), as a function of the index in the result set (x-axis, in thousands), with paper arranged: (a) in random order (b) using a relevance heuristic based on search hits. We can observe how, thanks to the heuristic (b), the majority of relevant papers can be identified by examining only a small fraction of the set (the top results on the left side).

- We start screening all papers in the result set, ranked according to different relevance heuristics (e.g. number of hits in queries), and we observe the ratio of relevant papers identified over time;
- We examine the same papers in random order, and measure the same ratio;
- When the two ratios are comparable, we assert we reached the tail of the distribution of relevant papers and stop examining and selecting new papers.

As sorting criteria, we use the number of hits in the search performed in the previous step, as well as other more complex heuristics, taking into account the index position in result sets and the number of citations. When examining a paper, we look into the full content to identify concepts related to our selection criteria previously illustrated. As done previously with search results, we gather relevant papers in one unique group. Using this stopping criterion, we conclude this selection step when we have identified 430 relevant papers.

## 2.5 Additional Search Techniques

The “early stopping” criterion previously described, while allowing a feasible and comprehensive selection strategy across thousands of contributions, has a natural tendency towards discarding relevant papers. We also expect to miss other relevant papers, not present in the initial set of 83817, because they were not identified by our database search. To cover for these limitations, we apply other search techniques in addition to database search. Differing from before, we here apply our selection criteria exhaustively for each document retrieved.

**Reference Search.** For each of the 430 relevant papers identified in the previous step, we search inside their cited references. In particular, we adopt backward

**Table 2.** Sample  $k$ -shingles with relevance probability (and total occurrences).

<b>k=1</b>	<b>k=2</b>	<b>k=3</b>
tpc-w, 1.00 (13)	defect prediction, 1.00 (34)	software defect prediction, 1.00 (22)
log-based, 0.92 (11)	(work)load prediction, 1.00 (32)	disk failure prediction, 1.00 (8)
sla, 0.84 (48)	software aging, 1.00 (13)	failure prediction model, 1.00 (7)
stragglers, 0.83 (10)	resource allocation, 1.00 (6)	cloud resource provisioning, 1.00 (5)
vm, 0.83 (59)	hardware failures, 0.89 (8)	automatic anomaly detection, 0.88 (7)

snowball sampling [18]: we include in our relevant set all papers previously cited by a relevant paper whenever they fulfill the selection requirements mentioned above. By doing so, we obtain 631 relevant elements, for a total of 1061.

**Conference Search.** Reference search allows to identify prominent contributions frequently mentioned by other authors. A drawback is the introduction of bias towards specific research groups and authors. We also observed how reference search rewards specific tasks and research fields as they are typically more cited. We therefore apply other search techniques to compensate for these facts. We perform a manual search by inspecting papers published in relevant conferences. These relevant conferences are identified via correlation with other relevant papers and have also been confirmed by experts in the field. We look at the latest 3 editions of each conference, in an effort to compensate the sampling of dated papers performed by reference search. We obtain 5 more papers with this method.

**Iterative Search Improvement.** To conclude our search, we attempt at improving our initial guess on IT Operations keywords via analysis of the available text content (text and abstract). Using our relevant paper set as positive samples, we perform a statistical analysis to identify  $k$ -shingles (sets of  $k$  consecutive tokens) that appear often in relevant documents (Table 2). In particular, we measure the document relevance probability given the set of shingles observed in the available text content. We choose  $k = 1, \dots, 5$ . We use these shingles as keywords to construct new queries along with previously used AI keywords. We here limit the collection to 20 results per query. Thanks to this step, we identify 20 new relevant papers. As a by-product, we get in contact with frequently cited concepts and keywords in AIOps, later useful for taxonomy and classification.

## 2.6 Data Extraction and Categorization

After obtaining the result set of relevant papers (counting 1086 contributions), we analyze the available information to draw quantitative results and answer our research questions. We describe here the data extraction process and the analysis techniques employed to gather insights and trends for the AIOps field.

First, we classify the relevant papers according to target components and data sources. Target components indicate a high-level piece of software or hardware in an IT system that the document tries to enhance (e.g. hard drive for hard

**Table 3.** Selection of result papers grouped by data sources, targets and (sub)categories.

Ref.	Data Sources							Targets				Cat.		
	Source Code	Testing Resources	System Metrics	KPIs/SLO data	Network Traffic	Topology	Incident Reports	Event Logs	Execution Traces	Source Code	Application		Hardware	Network
[27]	•								•					1.1
[32]	•	•							•					1.2
[16]			•						•				•	1.3
[41]			•	•					•				•	1.3
[29]									•				•	1.4
[47]			•							•				2.1
[14]			•						•				•	2.1
[12]								•					•	2.1
[46]								•			•	•		2.1
[8]	•								•				•	2.2
[11]	•	•							•				•	2.2
[17]	•	•							•				•	2.2
[35]	•			•					•				•	2.2
[24]								•	•				•	2.2
[37]								•	•	•	•		•	2.2
[45]								•	•				•	2.2
[43]	•							•					•	3.1
[42]				•					•					3.1
[40]			•	•					•				•	3.1
[21]				•	•						•		•	3.1
[22]				•	•				•	•			•	3.1
[15]									•					3.1
[10]									•	•			•	3.1
[6]									•				•	3.1
[28]									•				•	3.1
[30]													•	3.2
[49]	•								•				•	3.3
[1]	•	•								•	•			4.1
[33]			•									•	•	4.1
[5]				•								•	•	4.1
[44]	•									•				4.2
[4]	•		•	•							•			4.2
[19]									•				•	4.2
[9]										•			•	4.2
[36]		•									•			4.3
[7]			•	•							•		•	4.3
[26]									•				•	4.3
[2]										•			•	4.3
[39]											•		•	5.1
[48]									•				•	5.2
[25]									•	•			•	5.2
[38]			•	•							•		•	5.3

(Sub)Category Legend			
1.1 Software Defect Prediction	2.2 System Failure Prediction	4.2 Root Cause Diagnosis	
1.2 Fault Injection	3.1 Anomaly Detection	4.3 RCA - Others	
1.3 Software Rejuvenation	3.2 Internet Traffic Classification	5.1 Incident Triage	
1.4 Checkpointing	3.3 Log Enhancement	5.2 Solution Recommendation	
2.1 Hardware Failure Prediction	4.1 Fault Localization	5.3 Recovery	

disk failure prediction). We group components in five high-level categories: code, application, hardware, network and datacenter. Data sources provide an indication of the input information of the algorithm (such as logs, metrics, or execution traces). Data sources are categorized in source code, testing resources, system metrics, key performance indicators (KPIs), network traffic, topology, incident reports, logs and traces. the “AI Method” axis denotes the actual algorithm employed, with similar methods aggregated in bigger classes to avoid excessive fragmentation (e.g. ‘clustering’ may contain both k-means and agglomerative hierarchical clustering approaches). Table 3 presents a selection of papers from the result set with the corresponding target, source and category annotation.

Then, we use the result set to infer a taxonomy based on tasks and target goals. The taxonomy is depicted in Fig. 2. We divide in AIOps contributions in failure management (FM), the study on how to deal with undesired behavior



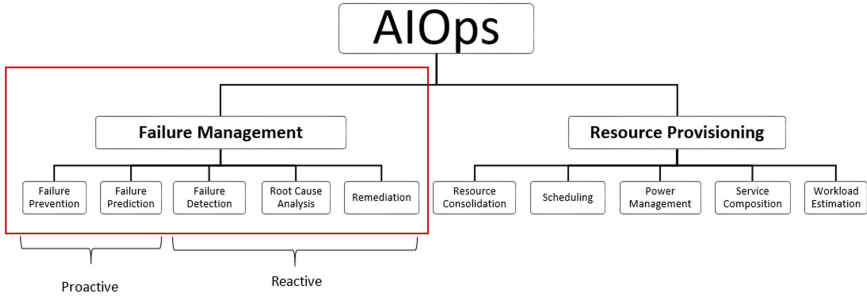


Fig. 2. Taxonomy of AIOps as observed in the identified contributions

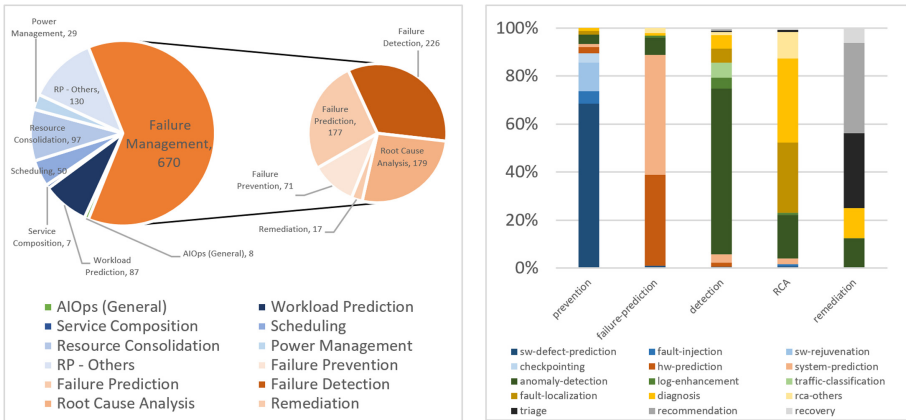


Fig. 3. Left: distribution of AIOps papers in macro-areas and categories. Right: percent distribution of failure management papers by category in corresponding sub-categories.

in the delivery of IT services; and resource provisioning, the study of allocation of energetic, computational, storage and time resources for the optimal delivery of IT services. Within each of these macro-areas, we further distinguish approaches in categories based on the similarity of goals. In failure management, these categories are failure prevention, online failure prediction, failure detection, root cause analysis (RCA) and remediation. In resource provisioning, we divide contributions in resource consolidation, scheduling, power management, service composition, and workload estimation. We further choose to expand our analysis of FM (red box of Fig. 2) by applying for this macro-area an additional subcategorization based on specific problems. Examples of subcategories are checkpointing for failure prevention, or fault localization for root cause analysis (see also Table 3).

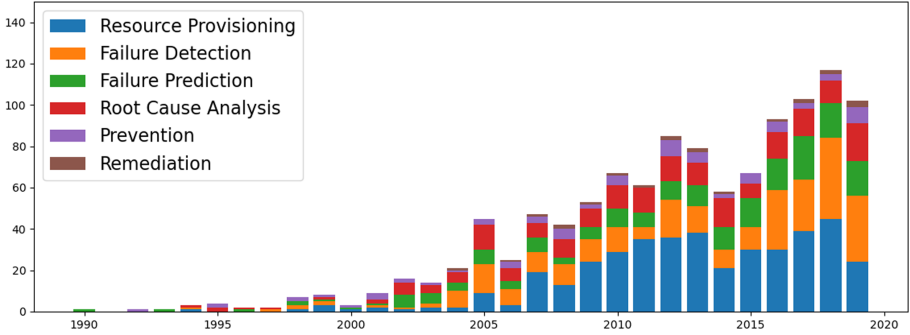


Fig. 4. Published papers in AIOps by year and categories from the described taxonomy.

### 3 Results

We now discuss the results of our mapping study. We first analyze the distribution of papers in our taxonomy. The left side of Fig. 3 visualizes the distribution of identified papers by macro-area and category. Excluding papers treating AIOps in general (8), we observe that more the majority of items (670, 62.1%) are associated with failure management (FM), with most contributions concentrated in online failure prediction (26.4%), failure detection (33.7%), and root cause analysis (26.7%); the remaining resource provisioning papers support in large part resource consolidation, scheduling and workload prediction. On the right side, we can observe that the most common problems in FM are software defect prediction, system failure prediction, anomaly detection, fault localization and root cause diagnosis. To analyze temporal trends present inside the AIOps field, we measured the number of publications in each category by year of publication. The corresponding bar plot is depicted in Fig. 4. Overall, we observe a large, on-growing number of publications in AIOps. We can observe how failure detection has gained particular traction in recent years (71 publications for the 2018–2019 period) with a contribution size larger than the entire resource provisioning macro-area (69 publications in the same time frame). Failure detection is followed by root cause analysis (39) and online failure prediction (34), while failure prevention and remediation are the areas with the smallest number of attested contributions (11 and 5, respectively).

### 4 Conclusion

In this paper, we presented our contribution towards better structuring the AIOps field. We planned and conducted a systematic mapping study by means of pre-established formulation, search, selection, and categorization techniques, thanks to which we collected more than 1000 contributions and grouped into several categories thanks to our proposed taxonomy, and differing substantially in terms of goals, data sources and target components. In our result section,

we have shown how the majority of papers address failures in different forms. From a time perspective, we observed a generalized on-growing research interest, especially for tasks such as anomaly detection and root cause analysis.

## References

1. Abreu, R., Zoetewij, P., Gemund, A.J.V.: Spectrum-based multiple fault localization. In: IEEE/ACM International Conference on Automated Software Engineering, November 2009. <https://doi.org/10.1109/ase.2009.25>
2. Aguilera, M.K., Mogul, J.C., Wiener, J.L., Reynolds, P., Muthitacharoen, A.: Performance debugging for distributed systems of black boxes. *ACM SIGOPS Oper. Syst. Rev.* **37**(5), 74–89 (2003). <https://doi.org/10.1145/1165389.945454>
3. Lerner, A.: AIOps Platforms, August 2017. <https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/>
4. Attariyan, M., Chow, M., Flinn, J.: X-ray: automating root-cause diagnosis of performance anomalies in production software. In: Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation, OSDI 2012, Hollywood, CA, USA, pp. 307–320, October 2012. <https://doi.org/10.5555/2387880.2387910>
5. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D.A., Zhang, M.: Towards highly reliable enterprise network services via inference of multi-level dependencies. In: Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM (2007). <https://doi.org/10.1145/1282380.1282383>
6. Barham, P., Isaacs, R., Mortier, R., Narayanan, D.: Magpie: online modelling and performance-aware systems. In: Proceedings of the 9th Conference on Hot Topics in Operating Systems, HOTOS 2003, Lihue, Hawaii, vol. 9, p. 15, May 2003. <https://doi.org/10.5555/1251054.1251069>
7. Bodik, P., Goldszmidt, M., Fox, A., Woodard, D.B., Andersen, H.: Fingerprinting the datacenter: automated classification of performance crises. In: Proceedings of the 5th European Conference on Computer Systems - EuroSys 2010 (2010). <https://doi.org/10.1145/1755913.1755926>
8. Chalermarwong, T., Achalakul, T., See, S.C.W.: Failure prediction of data centers using time series and fault tree analysis. In: IEEE 18th International Conference on Parallel and Distributed Systems, December 2012. <https://doi.org/10.1109/icpads.2012.129>
9. Chen, M., Kiciman, E., Fratkin, E., Fox, A., Brewer, E.: Pinpoint: problem determination in large, dynamic Internet services. In: Proceedings of IEEE International Conference on Dependable Systems and Networks (2002). <https://doi.org/10.1109/dsn.2002.1029005>
10. Chow, M., Meisner, D., Flinn, J., Peek, D., Wenisch, T.F.: The mystery machine: end-to-end performance analysis of large-scale internet services. In: OSDI 2014: Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation, pp. 217–231 (2014). <https://doi.org/10.5555/2685048.2685066>
11. Cohen, I., Goldszmidt, M., Kelly, T., Symons, J., Chase, J.S.: Correlating instrumentation data to system states: a building block for automated diagnosis and control. In: Proceedings of the 6th USENIX Conference on Symposium on Operating Systems Design & Implementation, OSDI 2004 (2004). <https://doi.org/10.5555/1251254.1251270>

12. Costa, C.H., Park, Y., Rosenburg, B.S., Cher, C.Y., Ryu, K.D.: A system software approach to proactive memory-error avoidance. In: SC 2014: International Conference for High Performance Computing, Networking, Storage and Analysis, November 2014. <https://doi.org/10.1109/sc.2014.63>
13. Dang, Y., Lin, Q., Huang, P.: AIOps: real-world challenges and research innovations. In: IEEE/ACM 41st International Conference on Software Engineering: Companion, May 2019. <https://doi.org/10.1109/icse-companion.2019.00023>
14. Davis, N.A., Rezgui, A., Soliman, H., Manzanares, S., Coates, M.: FailureSim: a system for predicting hardware failures in cloud data centers using neural networks. In: IEEE 10th International Conference on Cloud Computing (CLOUD), Jun 2017. <https://doi.org/10.1109/cloud.2017.75>
15. Du, M., Li, F., Zheng, G., Srikumar, V.: DeepLog: anomaly detection and diagnosis from system logs through deep learning. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security (2017). <https://doi.org/10.1145/3133956.3134015>
16. Garg, S., van Moorsel, A., Vaidyanathan, K., Trivedi, K.: A methodology for detection and estimation of software aging. In: Proceedings Ninth International Symposium on Software Reliability Engineering (Cat. No.98TB100257). IEEE Computer Society (1998). <https://doi.org/10.1109/issre.1998.730892>
17. Islam, T., Manivannan, D.: Predicting application failure in cloud: a machine learning approach. In: 2017 IEEE International Conference on Cognitive Computing (ICCC), Jun 2017. <https://doi.org/10.1109/iecc.2017.11>
18. Jalali, S., Wohlin, C.: Systematic literature studies: database searches vs. backward snowballing. In: Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, pp. 29–38, September 2012. <https://doi.org/10.1145/2372251.2372257>
19. Kandula, S., Katabi, D., Vasseur, J.P.: Shrink: a tool for failure diagnosis in IP networks. In: Proceedings of the 2005 ACM SIGCOMM Workshop on Mining Network Data - MineNet 2005 (2005). <https://doi.org/10.1145/1080173.1080178>
20. Kobbacy, K.A.H., Vadera, S., Rasmy, M.H.: AI and OR in management of operations: history and trends. *J. Oper. Res. Soc.* **58**(1), 10–28 (2007). <https://doi.org/10.1057/palgrave.jors.2602132>
21. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM 2004. ACM Press (2004). <https://doi.org/10.1145/1015467.1015492>
22. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: Proceedings of the 2005 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications - SIGCOMM 2005. ACM Press (2005). <https://doi.org/10.1145/1080091.1080118>
23. Li, Y., et al.: Predicting node failures in an ultra-large-scale cloud computing platform: an AIOps solution. *ACM Trans. Software Eng. Methodol.* **29**(2), 1–24 (2020). <https://doi.org/10.1145/3385187>
24. Liang, Y., Zhang, Y., Xiong, H., Sahoo, R.: Failure prediction in IBM BlueGene/L event logs. In: Seventh IEEE International Conference on Data Mining (ICDM) (2007). <https://doi.org/10.1109/icdm.2007.46>
25. Lin, F., Beadon, M., Dixit, H.D., Vunnam, G., Desai, A., Sankar, S.: Hardware remediation at scale. In: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), June 2018. <https://doi.org/10.1109/dsn-w.2018.00015>

26. Lin, Q., Zhang, H., Lou, J.G., Zhang, Y., Chen, X.: Log clustering based problem identification for online service systems. In: Proceedings of the 38th ACM International Conference on Software Engineering Companion (ICSE) (2016). <https://doi.org/10.1145/2889160.2889232>
27. Menzies, T., Greenwald, J., Frank, A.: Data mining static code attributes to learn defect predictors. *IEEE Trans. Software Eng.* **33**(1), 2–13 (2007). <https://doi.org/10.1109/TSE.2007.256941>
28. Chen, M.Y., Accardi, A., Kiciman, E., Lloyd, J., Patterson, D., Fox, A., Brewer, E.: Path-based failure and evolution management. In: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation, NSDI 2004, San Francisco, California, vol. 1, p. 23, March 2004. <https://doi.org/10.5555/1251175.1251198>
29. Moody, A., Bronevetsky, G., Mohror, K., Supinski, B.R.D.: Design, modeling, and evaluation of a scalable multi-level checkpointing system. In: 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, November 2010. <https://doi.org/10.1109/sc.2010.18>
30. Moore, A.W., Zuev, D.: Internet traffic classification using Bayesian analysis techniques. In: Proceedings of the 2005 ACM International Conference on Measurement and Modeling of Computer Systems - SIGMETRICS 2005 (2005). <https://doi.org/10.1145/1064212.1064220>
31. Mukwevho, M.A., Celik, T.: Toward a smart cloud: a review of fault-tolerance methods in cloud systems. *IEEE Trans. Serv. Comput.* **1** (2018). <https://doi.org/10.1109/tsc.2018.2816644>
32. Natella, R., Cotroneo, D., Duraes, J.A., Madeira, H.S.: On fault representativeness of software fault injection. *IEEE Trans. Software Eng.* **39**(1), 80–96 (2013). <https://doi.org/10.1109/tse.2011.124>
33. Nguyen, H., Shen, Z., Tan, Y., Gu, X.: FChain: toward black-box online fault localization for cloud systems. In: IEEE 33rd International Conference on Distributed Computing Systems, July 2013. <https://doi.org/10.1109/icdcs.2013.26>
34. Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf. Softw. Technol.* **64**, 1–18 (2015). <https://doi.org/10.1016/j.infsof.2015.03.007>
35. Pitakrat, T., Okanović, D., van Hoorn, A., Grunske, L.: Hora: architecture-aware online failure prediction. *J. Syst. Softw.* **137**, 669–685 (2018). <https://doi.org/10.1016/j.jss.2017.02.041>
36. Podgurski, A., et al.: Automated support for classifying software failure reports. In: Proceedings of IEEE 25th International Conference on Software Engineering (2003). <https://doi.org/10.1109/icse.2003.1201224>
37. Salfner, F., Malek, M.: Using hidden semi-Markov models for effective online failure prediction. In: 26th IEEE International Symposium on Reliable Distributed Systems (SRDS), October 2007. <https://doi.org/10.1109/srds.2007.35>
38. Samir, A., Pahl, C.: A controller architecture for anomaly detection, root cause analysis and self-adaptation for cluster architectures. In: International Conference on Adaptive and Self-Adaptive Systems and Applications (2019). 10993/42062
39. Shao, Q., Chen, Y., Tao, S., Yan, X., Anerousis, N.: Efficient ticket routing by resolution sequence mining. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD (2008). <https://doi.org/10.1145/1401890.1401964>

40. Sharma, A.B., Chen, H., Ding, M., Yoshihira, K., Jiang, G.: Fault detection and localization in distributed systems using invariant relationships. In: 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), June 2013. <https://doi.org/10.1109/dsn.2013.6575304>
41. Vaidyanathan, K., Trivedi, K.: A comprehensive model for software rejuvenation. *IEEE Trans. Dependable Secure Comput.* **2**(2), 124–137 (2005). <https://doi.org/10.1109/tdsc.2005.15>
42. Xu, H., et al.: Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In: Proceedings of the 2018 World Wide Web Conference on World Wide Web (2018). <https://doi.org/10.1145/3178876.3185996>
43. Xu, W., Huang, L., Fox, A., Patterson, D., Jordan, M.I.: Detecting large-scale system problems by mining console logs. In: Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles - SOSP 2009 (2009). <https://doi.org/10.1145/1629575.1629587>
44. Yuan, D., Mai, H., Xiong, W., Tan, L., Zhou, Y., Pasupathy, S.: SherLog: error diagnosis by connecting clues from run-time logs. In: ACM SIGARCH Computer Architecture News, vol. 38, no. 1, pp. 143–154 (2010). <https://doi.org/10.1145/1735970.1736038>
45. Zhang, K., Xu, J., Min, M.R., Jiang, G., Pelechrinis, K., Zhang, H.: Automated IT system failure prediction: a deep learning approach. In: IEEE International Conference on Big Data (2016). <https://doi.org/10.1109/bigdata.2016.7840733>
46. Zhang, S., et al.: Syslog processing for switch failure diagnosis and prediction in datacenter networks. In: IEEE/ACM 25th International Symposium on Quality of Service (IWQoS), June 2017. <https://doi.org/10.1109/iwqos.2017.7969130>
47. Zheng, S., Ristovski, K., Farahat, A., Gupta, C.: Long short-term memory network for remaining useful life estimation. In: IEEE International Conference on Prognostics and Health Management (ICPHM) (2017). <https://doi.org/10.1109/icphm.2017.7998311>
48. Zhou, W., Tang, L., Li, T., Shwartz, L., Grabarnik, G.Y.: Resolution recommendation for event tickets in service management. In: IFIP/IEEE International Symposium on Integrated Network Management (IM) (2015). <https://doi.org/10.1109/inm.2015.7140303>
49. Zhu, J., He, P., Fu, Q., Zhang, H., Lyu, M.R., Zhang, D.: Learning to log: helping developers make informed logging decisions. In: IEEE/ACM 37th IEEE International Conference on Software Engineering, May 2015. <https://doi.org/10.1109/icse.2015.60>