

doi <https://doi.org/10.18265/2447-9187a2022id8011>
ARTIGO ORIGINAL

SUBMETIDO 01/10/2023

APROVADO 11/12/2023

PUBLICADO ON-LINE 21/12/2023

VERSÃO FINAL DIAGRAMADA 16/05/2025

EDITOR ASSOCIADO

Prof. Dr. Francisco Petrônio Alencar de Medeiros

Previsão de falhas em ambientes de nuvem: uma revisão sistemática de literatura

RESUMO: Atualmente, os serviços de computação em nuvem são fundamentais para a economia moderna. Em vista disso, diversos setores têm reorientado seus serviços para a nuvem. A complexidade e natureza distribuída desses sistemas, no entanto, contribuem para a ocorrência de diversos tipos de falhas que, quando não tratadas, podem resultar em danos catastróficos. Para mitigar seus efeitos, diversos métodos de previsão de falhas foram propostos. Dada a evolução desses métodos ao longo do tempo, este estudo tem como objetivo apresentar uma revisão sistemática da literatura a respeito das técnicas e métodos utilizados na previsão de falhas em ambientes de nuvem. Foram analisados quarenta artigos de quatro bases de dados cientificamente reconhecidas. Os resultados evidenciaram a predominância de técnicas de aprendizagem de máquina baseadas em árvore de decisão e destacaram a escassez de conjuntos de dados do mundo real sobre o avanço da pesquisa neste campo. Os esforços de estudos futuros podem explorar métodos para agregar amostras de dados de diferentes tarefas e conjunto de dados, além de técnicas para balanceamento de classes em modelos de previsão de falhas em ambiente de nuvem. Esta revisão sistemática fornece uma visão geral e atualizada das pesquisas relacionadas à previsão de falhas nos sistemas em nuvem. As observações e tendências identificadas neste trabalho podem ser úteis para futuros pesquisadores.

Palavras-chave: aprendizagem de máquina; computação em nuvem; previsão de falhas; revisão sistemática.

Failure prediction in cloud environments: a systematic literature review

ABSTRACT: *Cloud computing services are currently essential to the modern economy, with numerous sectors adopting cloud-based solutions. However, cloud systems' complexity and distributed nature increase their vulnerability to various types of failures, which, if not addressed, can lead to significant damage. Several failure prediction techniques have been proposed to mitigate the impact of these failures. Given the evolution of these methods over time, this study presents a systematic literature review of techniques and methods used for failure prediction in cloud environments. Forty articles from four scientifically*

 José Carlos dos Santos ^[1] *

 Tiago Buarque Assunção de Carvalho ^[2]

 Erica Teixeira Gomes de Sousa ^[3]

[1] carlos.santos2@ufrpe.br

[3] erica.sousa@ufrpe.br

Universidade Federal Rural de Pernambuco (UFRPE), Recife, Pernambuco, Brasil

[2] tiago.buarque@ufape.edu.br
Universidade Federal do Agreste de Pernambuco (UFAPE), Garanhuns, Pernambuco, Brasil

* Autor para correspondência.



recognized databases were analyzed. The results highlight the predominance of machine-learning techniques based on decision trees and the scarcity of real-world datasets that support the advancement of research in this field. Future research efforts could explore methods for aggregating data samples from different tasks and datasets, and class-balancing techniques in failure prediction models for cloud environments. This systematic review provides an updated overview of the current state of research in cloud system failure prediction, and the trends and observations identified here may serve as a guide for future studies.

Keywords: cloud computing; failure prediction; machine learning; systematic review.

1 Introdução

Atualmente, diversos setores, como negócios, redes sociais, computação científica e de alto desempenho fazem uso da computação em nuvem (Bui *et al.*, 2020), modelo que possibilita acesso onipresente, conveniente e sob demanda a um conjunto compartilhado de recursos de computação configuráveis, que podem ser provisionados e liberados rapidamente, com o mínimo de esforço de gerenciamento e com interação do provedor de serviços (Mell; Grance, 2011). Os serviços de computação em nuvem são fundamentais para a economia moderna, uma vez que podem facilitar a expansão de empresas em todo o mundo (Buyya *et al.*, 2018). Devido à escalabilidade e à flexibilidade proporcionadas por essa tecnologia, pequenas e grandes empresas têm reorientado suas infraestruturas para nuvem (Roumani; Nwankpa, 2019). Estimativas recentes apontam que, até 2025, mais de três quartos dos negócios globais estarão nesse ambiente (Herbst *et al.*, 2018).

O fato de esse modelo de computação possibilitar o uso de recursos sob demanda, em que os clientes pagam apenas por aqueles que forem utilizados, tem atraído cada vez mais interessados, aumentando a necessidade de se manter a disponibilidade dos serviços fornecidos por esses sistemas (Alahmad; Daradkeh; Agarwal, 2021). Cumpre atentar, entretanto, para a complexidade e a natureza distribuída dos sistemas em nuvem, aspectos que contribuem para a ocorrência de variados tipos de falhas, como falhas de software e de hardware, congestionamento de rede e falta de recursos (sobrecarga de servidor). Essas falhas são as principais responsáveis pela indisponibilidade dos serviços em nuvem (Gao; Wang; Shen, 2022; Rawat; Bhadoria, 2021).

Segundo alguns estudos, as falhas são inevitáveis em sistemas complexos (Gupta *et al.*, 2017; Mariani *et al.*, 2018; Monni; Pezzè; Prisco, 2019). Quando não tratadas, essas falhas podem resultar em perdas de bilhões de dólares, especialmente em dados e aplicações de negócios críticos (Roumani; Nwankpa, 2019). Estima-se que as falhas causem um prejuízo anual de 700 bilhões de dólares (Li *et al.*, 2020).

Nesse contexto, diversas pesquisas foram realizadas com o objetivo de mitigar os efeitos das falhas nos sistemas de computação em nuvem. As tecnologias de previsão de falhas têm, em particular, atraído bastante atenção nos últimos anos (Liu *et al.*, 2017). A previsão de falhas é definida como um conjunto de ações proativas para detectar condições anormais antes que elas realmente aconteçam (Gokhroo; Govil; Pilli, 2017). Os preditores modernos utilizam métricas coletadas dos nós de computação nos sistemas em nuvem para inferir a presença desse tipo de problema. Os dados coletados são utilizados no processo de treinamento de modelos que possam prever a ocorrência de falhas em um futuro próximo (Monni; Pezzè; Prisco, 2019).

Considerando a diversidade de métodos propostos para prever falhas nos sistemas de computação em nuvem (Gupta *et al.*, 2017) e a evolução desses métodos ao longo do tempo, questiona-se: *Quais são os métodos atualmente utilizados para previsão de falhas nos sistemas em nuvem?* Este trabalho visa apresentar uma revisão sistemática da literatura a respeito dos métodos e técnicas utilizadas no processo de previsão de falhas nos sistemas em nuvem.

O artigo está organizado da seguinte forma: na Seção 2 são apresentadas as técnicas de previsão de falhas; na Seção 3 constam a metodologia e o processo de pesquisa. Na Seção 4, é apresentada a análise dos resultados; na Seção 5, expõem-se e se discutem os resultados; e, por fim, a Seção 6 conclui o artigo.

2 Técnicas de previsão de falhas

Os métodos de previsão de falhas podem ser classificados em três categorias (Gao; Wang; Shen, 2022): i) abordagem estatística; ii) abordagem de aprendizado de máquina; iii) abordagem de aprendizagem profunda.

Entre os modelos da primeira abordagem, o Autorregressivo Integrado de Média Móvel (ARIMA) é um dos mais aplicados para a previsão de séries temporais (Hermias; Teknomo; Monje, 2017). Modelos baseados em ARIMA podem ser utilizados para prever o comportamento futuro do sistema (Shao; Zhang, 2018). A título de exemplo, citamos Rawat *et al.* (2021), que utilizaram ARIMA para prever falhas em máquinas virtuais em um *datacenter* de um sistema de computação em nuvem.

A segunda abordagem – o aprendizado de máquina – diz respeito a um ramo da inteligência artificial que permite que computadores aprendam de forma autônoma, ou seja, sem uma programação explícita (Su *et al.*, 2020). As técnicas de aprendizado de máquina têm sido utilizadas com maior ênfase desde 2016, o que se deve à sua capacidade de classificação e previsão (Bhanage; Pawar; Kotecha, 2021). A aprendizagem profunda, por sua vez, é uma subárea do aprendizado de máquina, entretanto seus algoritmos necessitam de grandes volumes de dados e de alto desempenho computacional (Fahim; Sillitti, 2019; Su *et al.*, 2020).

3 Metodologia e processo de pesquisa

A revisão sistemática da literatura (RSL) visa caracterizar, sintetizar e comparar os estudos mais recentes sobre determinado tema de pesquisa, possibilitando a transferência de conhecimento na comunidade de pesquisa (Kashani; Mahdipour, 2022). Este trabalho de revisão segue os procedimentos apresentados por Majid e Anuar (2022), que incluem planejamento, execução e relatório. O planejamento da RSL inicia-se com a definição de um protocolo de revisão, isto é, um conjunto claro de regras que orientam a condução da RSL, mitigando possíveis vieses decorrentes das expectativas dos pesquisadores (Dias; Correia; Malheiros, 2021). O protocolo de revisão especifica os seguintes aspectos: objetivo da pesquisa, questões de pesquisa, palavras-chave a serem utilizadas nos mecanismos de busca, critérios de seleção das bases de busca, idioma das publicações, métodos de busca, bases de busca, critérios de inclusão e exclusão, critérios de qualidade bem como os dados que serão extraídos dos estudos para responder às questões de pesquisa.

3.1 Identificação do tema e questão de pesquisa

O aumento dos serviços providos pelos sistemas de computação em nuvem tem como consequência o crescimento do tráfego de rede e o aumento do uso de recursos, elevando a probabilidade de ocorrência de falhas (Kabir *et al.*, 2021). Normalmente, a ocorrência de falhas nesses sistemas está associada a perdas financeiras, mas, em alguns casos, pode resultar em perda de vidas humanas (Irrera, 2016).

Assim, esta pesquisa busca revisar o estado da arte de forma sistemática, analisando os métodos mais recentes empregados na previsão de falhas em sistemas de computação em nuvem, com o objetivo de responder às seguintes questões de pesquisa:

- Quais são os métodos utilizados para previsão de falhas em sistemas em nuvem?
- Quais são as características dos conjuntos de dados utilizados na previsão de falhas nos sistemas em nuvem?
- Quais são as métricas utilizadas para avaliar modelos de previsão de falhas em sistema em nuvem?

Quadro 1 ▼

Critérios de inclusão e exclusão dos artigos na RSL.
Fonte: dados da pesquisa

3.2 Critérios de inclusão e exclusão

Os critérios utilizados para inclusão e exclusão dos artigos estão listados no Quadro 1.

Inclusão	Exclusão
Publicação decorrente de estudos primários	Publicação decorrente de estudos secundários
Publicação em língua inglesa	Publicação duplicada nas bases de busca
Publicação disponível na íntegra	Publicação que fuja ao escopo da pesquisa
–	Publicação relacionada ao atributo “segurança de dependabilidade”
–	Publicação em idioma diferente do inglês

3.3 Método de busca de trabalhos

Para assegurar a legitimidade deste estudo, foram selecionadas quatro bases de dados cientificamente reconhecidas. Além da relevância das bases, foram consideradas a acessibilidade com conta acadêmica, a possibilidade de busca por palavras-chave e a opção de exportação de resultados. As bases escolhidas foram: ACM Digital Library, IEEE Xplore, Scopus e SpringerLink.

Nesta etapa da RSL, definiram-se as palavras-chave e respectivos sinônimos utilizados nas bases de busca. Após uma pesquisa exploratória e análise de trabalhos relacionados, foram identificados os termos mais recorrentes na comunidade científica, possibilitando a extração das seguintes palavras-chave: *fault prediction*, *failure prediction*, *cloud computing*, *cloud service*, *cloud infrastructure*, *cloud platform* e *cloud datacenter*.

As *strings* adequadas para cada mecanismo de busca foram definidas após diversos testes, pois uma *string* muito geral geraria milhares de resultados, enquanto uma *string* demasiadamente específica poderia excluir resultados relevantes. Após uma pesquisa prévia manual em cada base de busca, identificaram-se as peculiaridades de cada

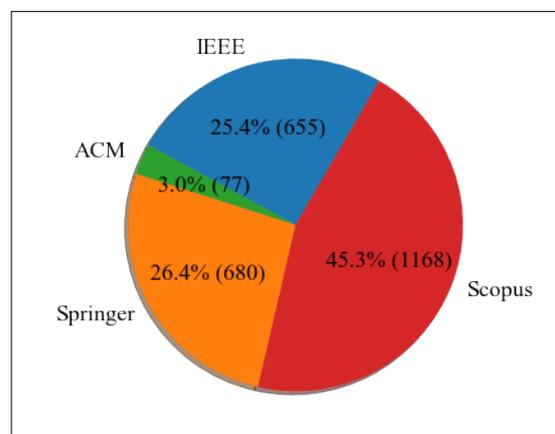
Quadro 2 ▼
Strings por base de busca.
Fonte: dados da pesquisa

mecanismo, e, com o uso dos operadores booleanos (*AND*, *OR*), foram definidas as strings de busca apropriadas para cada base. No Quadro 2, são apresentadas as bases de busca utilizadas e suas respectivas strings.

Bases de busca	Strings de busca
ACM Digital Library	<code>[[All: "fault prediction"] OR [All: "failure prediction"]] AND [[All: "cloud computing"] OR [All: "cloud service"] OR [All: "cloud infrastructure"] OR [All: "cloud platform"] OR [All: "cloud datacenter"]]</code>
IEEE Xplore	<code>((("Full Text & Metadata":"fault prediction") OR ("Full Text & Metadata":"failure prediction"))) AND ((("Full Text & Metadata":"cloud computing") OR ("Full Text & Metadata":"cloud service") OR ("Full Text & Metadata":"cloud infrastructure") OR ("Full Text & Metadata":"cloud platform") OR ("Full Text & Metadata":"cloud datacenter")))</code>
Scopus	<code>(ALL ("fault prediction") OR ALL ("failure prediction") AND ALL ("cloud computing") OR ALL ("cloud service") OR ALL ("cloud infrastructure") OR ALL ("cloud platform") OR ALL ("cloud datacenter"))</code>
SpringerLink	<code>((("fault prediction") OR ("failure prediction"))) AND ((("cloud computing") OR ("cloud service") OR ("cloud infrastructure") OR ("cloud platform") OR ("cloud datacenter")))</code>

O processo de busca adotou os seguintes critérios de exclusão: i) trabalhos publicados antes de 2017, pois poderiam apresentar soluções obsoletas; ii) artigos de revisão, que frequentemente apresentam poucos detalhes sobre os métodos empregados, abordando o estado da arte sob uma perspectiva mais conceitual (Santos *et al.*, 2020). A Figura 1 ilustra os resultados da busca primária, em que um total de 2.580 artigos foi retornado, sendo 1.168 da base Scopus, 680 da Springer, 655 da IEEE e 77 da ACM.

Figura 1 ►
Resultado da busca primária.
Fonte: dados da pesquisa



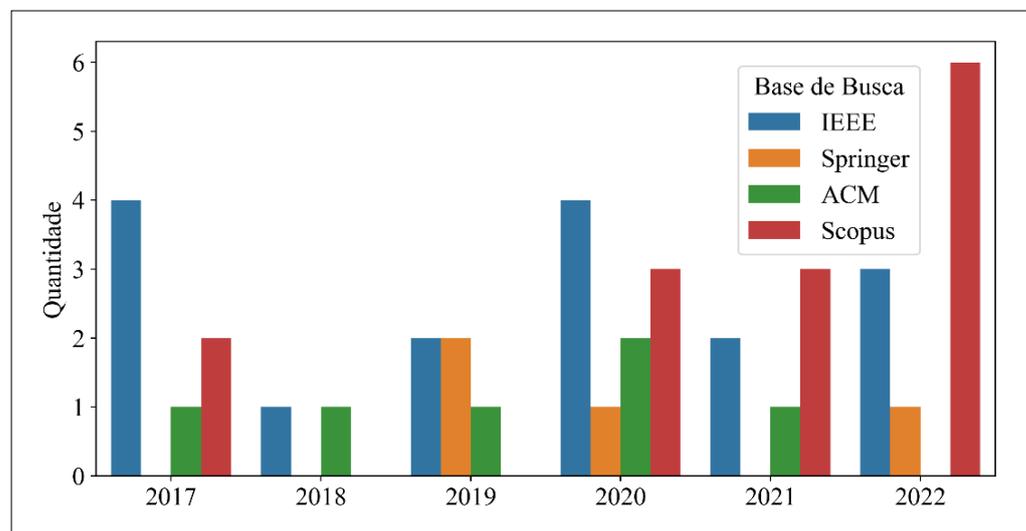
Após a busca primária, foram aplicados os critérios de inclusão e exclusão. Inicialmente, 119 artigos, correspondentes a 4,61% do total de artigos da busca primária, foram eliminados por estarem duplicados nas bases de busca. Em seguida, realizou-se a leitura dos títulos, resumos, introdução e conclusão, e, quando necessário, dos textos completos, a fim de se verificar a relevância e a relação de cada trabalho com o escopo da pesquisa, além de sua conformidade com os critérios de inclusão e exclusão. Destaca-se que os três autores desta pesquisa participaram ativamente da revisão dos artigos para garantir a qualidade e imparcialidade dos resultados obtidos. Após as leituras necessárias, 2.421 artigos, correspondendo a 93,8% do total da busca primária, foram excluídos, resultando na inclusão de 40 artigos na revisão, o que representa 1,55% do total inicial de artigos.

A Figura 2 apresenta a distribuição dos artigos incluídos na revisão sistemática por base de busca e ano de publicação. Embora a base Scopus tenha retornado 45,3% (1.168) do total de artigos na busca primária, a base IEEE foi a que mais contribuiu com artigos incluídos, totalizando 16, seguida das bases Scopus, ACM e Springer, com 14, 6 e 4 artigos, respectivamente. Nota-se também uma concentração maior de publicações nos anos de 2020 e 2022.

Figura 2 ▶

Artigos aceitos por base de busca.

Fonte: dados da pesquisa



4 Análise de resultados

Quadro 3 ▼

Síntese dos artigos da RSL desta pesquisa.

Fonte: dados da pesquisa

A última etapa da RSL tem por objetivo sintetizar e correlacionar os dados extraídos das publicações para formular respostas às questões de pesquisa (Dias; Correia; Malheiros, 2021). O Quadro 3 apresenta uma síntese dos artigos incluídos na RSL, contendo as referências, métodos, conjuntos de dados empregados e métricas de avaliação.

Referência	Método	Dataset	Métrica
Prathibha (2019)	K-means, Decision Tree, KNN	WorkFlowSim	Precisão 83,33%
Shetty; Sajjan; G. (2019)	XGBoost	Google 2011 V.2	Precisão 92%, Recall 94,8%
Liu, X. et al. (2020)	SVM, XGBoost, Random Forest	Logs Baidu	Precisão 93,33%
Ma et al. (2020)	RNN	Alibaba V.2018	Precisão 97,1%
Adamu et al. (2017)	SVM, Regressão Linear	The Computer Failure Data Repository (CFDR)	-
Gupta et al. (2017)	LSTM, BI-LSTM	Logs do Docker	Recall 94%, Precisão 12%, F1 Score 22%
Mariani et al. (2018)	IBM ITOA-PI	Logs de servidor SIP	Recall 92%, Precisão 93%, F1 Score 92%
Jassas; Mahmoud (2021)	ANN	Google 2009, Google 2011 V.2, Mustang, Trinity	Recall 99%, Precisão 99%, F1 Score 99%
Islam; Manivannan (2017)	LSTM	Google 2011 V.2	Precisão 87%
Liu et al. (2017)	OS-ELM, OS-SVM	Google 2011 V.2	Precisão 93%, Acurácia 93%
Das; Mueller; Rountree (2020)	LSTM	Logs Hadoop e Cassandra	Recall 86%, Acurácia 80%

continua

continuação

Gao; Wang; Shen (2022)	BI-LSTM	Google 2011 V.2	Precisão 90%
Alahmad, Daradkeh; Agarwal (2021)	ANN e CNN	Google 2011 V.2, Trinity, Alibaba V.2017	Precisão 94%
Su <i>et al.</i> (2020)	LSTM	Backblaze (Smart HD)	-
Padmakumari; Umamakeswari (2019)	<i>Naive Bayes, Random Forest, Rule-based, MF2N2</i>	WorkFlowSim	<i>Recall</i> 92,1%, <i>Especificidade</i> 95,1%, <i>F1 Score</i> , <i>AUC</i> 97,7%
Chen <i>et al.</i> (2019)	XGBoost	Microsoft Cloud	Precisão 82,75%, <i>Recall</i> 76,74%, <i>F1 Score</i> 79,63%
Bhattacharyya <i>et al.</i> (2017)	SVM	Google 2011 V.2 e BugBench	Precisão 100%
Lin <i>et al.</i> (2018)	<i>Random Forest, LSTM</i>	Sistema X Alibaba	<i>Recall</i> 63,5%, Precisão 92,4%, <i>F1 Score</i> 75,2%
Luo <i>et al.</i> (2021)	NTAM	Microsoft Azure (Smart HD) Backblaze (Smart HD)	<i>Recall</i> 57,58%, Precisão 81,7%, <i>F1 Score</i> 67,34%
Chakrabortii; Litz (2020)	<i>Isolation Forest, Autoencoder</i>	Google Cloud SSD	<i>AUC</i> 99%
Li <i>et al.</i> (2020)	MING, LSTM, <i>Random Forest</i>	Sistema X Alibaba	<i>AUC</i> 92%
Tehrani; Safi-Esfahani (2017)	SVM	Simulado no CloudSim	Precisão 99%, <i>Acurácia</i> 99,89%
Vu <i>et al.</i> (2021)	LSTM, BI-LSTM	Cloud Bitbrain	Precisão 97%
Khalil; Sheta; Elmaghraby (2016)	SVM	Google 2011 V.2	Precisão 99,04%, <i>Recall</i> 90,22%, <i>F1 Score</i> 93,26%
Liu, C. <i>et al.</i> (2020)	RMTL	Google 2011 V.2	Precisão 97%, <i>F1 Score</i> 98,46%
Nam <i>et al.</i> (2021)	CNN	Logs de VM OpenStack	Precisão 95%, <i>F1 Score</i> 67%
Liu, D. <i>et al.</i> (2020)	RNN	Smart Dataset W, M e S	Precisão 100%
Rawat <i>et al.</i> (2021)	ARIMA	Simulados em R	RMSE 0,0457443 MAE 0,0344786 MASE 0,6036391
Jassas; Mahmoud (2020)	<i>Decision Tree, Random Forest, Naive Bayes, QDA</i>	Google 2011 V.2, Mustang e Trinity	Precisão 99%, <i>Recall</i> 99%, <i>F1 Score</i> 99%
Chhetri <i>et al.</i> (2022)	<i>Random Forest, GB, LSTM, GRU</i>	Nuvem Universidade de Tartu	Precisão 99%, <i>Acurácia</i> 99%, <i>Recall</i> 100%, <i>F1 Score</i> 99%
Gollapalli <i>et al.</i> (2022)	RNA e SVM	Google 2019	Precisão 99,8%, <i>Acurácia</i> 99,8%
Abro <i>et al.</i> (2022)	<i>Naive Bayes, Random Forest, Regressão Linear</i>	Simulado no CloudSim	<i>Acurácia</i> 99%, <i>Sensibilidade</i> 98%, <i>Especificidade</i> 99%
Yang; Kim (2022)	LSTM	Logs OpenStack	Precisão 96,1%, <i>Recall</i> 95%, <i>F1 Score</i> 95,5%, <i>AUC</i> 98%
Jassas; Mahmoud (2022)	<i>Random Forest, Decision Tree, QDA, Naive Bayes, XGBoost</i>	Google 2011 V.2, Mustang e Trinity	Precisão 98%, <i>Recall</i> 95%, <i>F1 Score</i> 97%
Asmawi; Ismail; Shen (2022)	XGBoost, <i>Random Forest, Regressão Logística, Decision Tree, LSTM</i>	Google 2011 V.2	Precisão 94,31%, <i>Acurácia</i> 94,35%, <i>Sensibilidade</i> 91,92%, <i>Especificidade</i> 96,07%, <i>F1 Score</i> 93,1%
Mohammed <i>et al.</i> (2019)	ARIMA, KNN, SVM, <i>Random Forest</i>	CFDR NERSC 2019	<i>Acurácia</i> 90,76%, <i>Sensibilidade</i> 67,53%
Saxena; Singh (2022a)	SVM, FNN, Regressão Linear	Google 2011 V.2	<i>Acurácia</i> 97,8%
Liang <i>et al.</i> (2022)	<i>Random Forest</i>	Backblaze (Smart HD)	Precisão 98,44%, <i>Acurácia</i> 97,57%
Zhang <i>et al.</i> (2022)	XGBoost	Alibaba	Precisão 48,65%, <i>Recall</i> 79,34%, <i>F1 Score</i> 60,31%
Saxena; Singh (2022b)	SVM, <i>Random Forest, Regressão Linear, NN</i>	Google 2011 V.2	<i>Acurácia</i> 98,14%

4.1 Conjuntos de dados

O gerenciamento de falhas na computação em nuvem e o desenvolvimento de ambientes de nuvem confiáveis dependem de dados estatísticos obtidos em sistemas do mundo real. Poucos desses conjuntos de dados, no entanto, estão disponibilizados publicamente (Islam; Manivannan, 2017). O Quadro 4 apresenta os conjuntos de dados mais comuns identificados na RSL bem como os dados utilizados na abordagem experimental de cada trabalho analisado. Observa-se a predominância de dados referentes ao uso de CPU, memória e disco nas abordagens experimentais. As variáveis estáticas referem-se ao provisionamento de recursos no agendador de tarefas do Google, ou seja, aos recursos solicitados, mas não necessariamente utilizados pelas tarefas ao longo de seu ciclo de vida. Em alguns casos, como se observa, todos os dados disponíveis no conjunto foram utilizados.

Quadro 4 ▼

Conjuntos de dados identificados na RSL em relação aos dados utilizados na abordagem experimental.

Fonte: dados da pesquisa

Dataset	Referência	Dados utilizados
Google 2011 V.2	Shetty; Sajjan; G. (2019)	Utilização de CPU, memória, disco e duração do trabalho.
Google 2011 V.2	Jassas; Mahmoud (2021)	Variáveis estáticas.
Google 2011 V.2	Islam; Manivannan (2017)	Utilização de CPU, memória, cache, disco e variáveis estáticas.
Google 2011 V.2	Liu <i>et al.</i> (2017)	Utilização de CPU, memória e disco solicitado.
Google 2011 V.2	Gao; Wang; Shen (2022)	Utilização de CPU, memória, cache, disco e variáveis estáticas.
Google 2011 V.2	Alahmad; Daradkeh; Agarwal (2021)	Utilização de CPU, memória, disco e variáveis estáticas.
Google 2011 V.2	Bhattacharyya <i>et al.</i> (2017)	Utilização de CPU, memória e cache.
Google 2011 V.2	Khalil; Sheta; Elmaghraby (2016)	Utilização de CPU, memória e prioridade.
Google 2011 V.2	Liu, C. <i>et al.</i> (2020)	Utilização de CPU, memória, cache e disco.
Google 2011 V.2	Jassas; Mahmoud (2020)	Variáveis estáticas.
Google 2011 V.2	Jassas; Mahmoud (2022)	Variáveis estáticas.
Google 2011 V.2	Asmawi; Ismail; Shen (2022)	Variáveis estáticas.
Google 2011 V.2	Saxena; Singh (2022a)	Utilização de CPU e memória.
Google 2011 V.2	Saxena; Singh (2022b)	Utilização de CPU e memória.
Trinity	Jassas; Mahmoud (2021)	Todo conjunto de dados.
Trinity	Alahmad; Daradkeh; Agarwal (2021)	ID de trabalho, ID de grupo, ID de usuário, tempo inicial e de submissão, número de tarefas e nós requisitados.
Trinity	Jassas; Mahmoud (2020)	Todo conjunto de dados.
Trinity	Jassas; Mahmoud (2022)	Todo conjunto de dados.
Mustang	Jassas; Mahmoud (2020)	Todo conjunto de dados.
Mustang	Jassas; Mahmoud (2021)	Todo conjunto de dados.
Mustang	Jassas; Mahmoud (2022)	Todo conjunto de dados.
Alibaba V.2018	Ma <i>et al.</i> (2020)	Utilização de CPU e memória.
Alibaba V.2017	Alahmad; Daradkeh; Agarwal (2021)	Utilização de CPU, ID de trabalho, ID de tarefa, ID de máquina e solicitação de CPU e memória.
Google 2009	Jassas; Mahmoud (2021)	Todo o conjunto de dados.
Google 2019	Gollapalli <i>et al.</i> (2022)	Utilização de CPU, memória e disco.

As demais publicações utilizaram outros conjuntos de dados pouco difundidos ou gerados pelos próprios autores, os quais não estão disponibilizados publicamente.

O *dataset* Google 2011 V.2 é o conjunto de dados mais popular e tem sido o mais utilizado nos últimos anos. Seis anos após seu lançamento, o *dataset* já havia sido utilizado em mais de 450 publicações (Amvrosiadis *et al.*, 2018). O *dataset* Google 2011 V.2 consiste em um registro de eventos de um *cluster* de computação em nuvem com aproximadamente 12.500 nós de computação, realizado em 2011, com duração de 29 dias. Esse conjunto de dados inclui informações sobre especificações das máquinas e o ciclo de vida das tarefas e dos trabalhos solicitados pelos usuários (Alahmad; Daradkeh; Agarwal, 2021). As especificações de hardware, no entanto, não foram divulgadas pela Google (Amvrosiadis *et al.*, 2018). Embora o *dataset* Google 2009 tenha sido utilizado com propósitos comparativos pelos autores Jassas e Mahmoud (2021), atualmente esse conjunto de dados é considerado obsoleto e sua utilização não é recomendada. A Google também disponibilizou uma terceira versão do *dataset*, lançado no ano de 2020 (Wilkes, 2020); contudo, apenas um trabalho identificado nesta RSL utilizou essa versão (Gollapalli *et al.*, 2022).

Para reduzir a dependência excessiva do *dataset* da Google, os autores Amvrosiadis *et al.* (2018) lançaram o repositório Atlas, que inclui, entre outros conjuntos de dados, os *datasets* Trinity e Mustang, coletados no *Los Alamos National Laboratory*. O *dataset* Trinity, segundo mais utilizado nesta RSL, contém o registro de eventos de um supercomputador com 9.408 nós de computação idênticos, totalizando 301.056 núcleos Intel Xeon E5-2698v3 de 2,3 GHz e 1,2 PB de RAM, tornando-se o maior *cluster* rastreado publicamente por número de núcleos de CPU. Esses dados abrangem um período de 3 meses, entre fevereiro e abril de 2017, com 25.237 trabalhos de 88 usuários. O *dataset* Mustang, por sua vez, cobre um período de 61 meses – de outubro de 2011 a novembro de 2016 –, o que o torna o *dataset* público com o período de registro mais extenso publicamente disponível. Esse *cluster* conta com 1.600 nós de computação idênticos, com um total de 38.400 núcleos AMD Opteron 6176 de 2,3 GHz e 102 TB de RAM, e abriga 2,1 milhões de trabalhos de 565 usuários (Amvrosiadis *et al.*, 2018; Jassas; Mahmoud, 2021).

Outro conjunto de dados identificado neste estudo foi o *dataset* Alibaba V.2018, que registra eventos de um *cluster* com 4.000 máquinas, 9.000 serviços e 4 milhões de trabalhos, abrangendo um período de 8 dias. Esse *dataset* fornece detalhes sobre o ciclo de vida dos trabalhos e das tarefas, mas, por motivos de confidencialidade, as informações de hardware, como memória e disco, foram normalizadas (Guo *et al.*, 2019). O Quadro 5 sumariza os dados disponíveis nos conjuntos de dados identificados.

Quadro 5 ▼

Dados disponíveis nos *datasets*.

Fonte: dados da pesquisa

<i>Dataset</i>	Dados no <i>dataset</i>
Google 2011 V.2	ID de trabalho, tarefa, usuário, máquina e plataforma; capacidade de CPU e memória; <i>status</i> de trabalho e tarefa; solicitação e utilização de CPU, memória e disco; prioridade da tarefa
Trinity	Horário dos eventos, ID de trabalho, usuário e grupo; número de nós e tarefas solicitadas; tarefas por nó, tarefas por trabalho, processadores por tarefa e <i>status</i> do trabalho
Mustang	ID de grupo e usuário; horário de início e fim do trabalho, número de nós por trabalho, tarefas solicitadas e <i>status</i> do trabalho
Alibaba V.2018	ID de máquina, trabalho e tarefa; <i>timestamp</i> e <i>status</i> de máquina, trabalho e tarefa; número de CPU, memória e disco; solicitação e utilização de CPU, memória e disco; utilização média, mínima e máxima de CPU

continua

continuação

Alibaba V.2017	ID de máquina, trabalho e tarefa; <i>timestamp</i> e <i>status</i> de máquina, trabalho e tarefa; número de CPU, memória e disco; solicitação e utilização de CPU, memória e disco
Google 2009	<i>Timestamp</i> , ID de trabalho e tarefa, <i>status</i> de trabalho e utilização de CPU e memória
Google 2019	ID de trabalho, tarefa, usuário, máquina e plataforma; capacidade de CPU e memória; <i>status</i> de trabalho e tarefa; solicitação e utilização de CPU, memória e disco; prioridade da tarefa; taxa média e máxima de utilização de CPU e memória

Alguns autores utilizaram conjuntos de dados próprios, obtidos por meio de rastreios não disponibilizados publicamente, ou *datasets* sintéticos gerados em ambientes de simulação. Chen *et al.* (2019), por exemplo, utilizaram um conjunto de dados da nuvem Microsoft, gerado por eles mesmos. Por motivos de confidencialidade, não foram apresentadas informações relevantes a respeito do *dataset*, tampouco este foi disponibilizado publicamente. Luo *et al.* (2021) utilizaram dois conjuntos de dados de unidades de disco do Microsoft Azure, também não disponibilizados publicamente. Zhang *et al.* (2022) empregaram um conjunto de dados resultante de um registro de eventos da nuvem Alibaba por um período de um ano, não disponibilizado publicamente. Os autores Prathibha (2019), Padmakumari e Umamakeswari (2019), Tehrani e Safi-Esfahani (2017), Rawat *et al.* (2021) e Abro *et al.* (2022) utilizaram *datasets* obtidos em simulações, impossibilitando a reprodução dos experimentos.

4.2 Métodos

Em relação às técnicas de previsão de falhas, a Tabela 1 exibe os métodos predominantemente utilizados. Como se pode observar nessa tabela, os métodos mais utilizados foram *Random Forest*, LSTM e SVM.

Tabela 1 ►

Métodos predominantemente utilizados.
Fonte: dados da pesquisa

Método	Nº de artigos
<i>Random Forest</i>	12
<i>Long Short Term Memory</i> (LSTM)	10
<i>Support Vector Machine</i> (SVM)	9
XGBoost	6
Naive Bayes	4
Regressão Linear	4
<i>Decision Tree</i>	4
<i>Bidirectional Long Short Term Memory</i> (BI-LSTM)	3

Outros algoritmos apresentados no Quadro 3 foram utilizados por um ou dois autores. Islam e Manivannan (2017), a partir do algoritmo LSTM, propuseram um modelo para prever falhas de trabalhos e tarefas de aplicações em nuvem. Nos experimentos, os autores constataram alta correlação entre tempo e consumo de recursos das tarefas que falharam e propuseram um método para estimar uma redução no consumo de recursos, como CPU e memória, em função da previsão de falhas.

Embora os autores acima tenham considerado o SVM inadequado para a previsão de falhas com o *dataset* do Google 2011 V.2, Khalil, Sheta e Elmaghraby (2016) propuseram um método que utiliza SVM para prever falhas de máquinas por sobrecarga de trabalhos em um sistema em nuvem, treinado exclusivamente com dados de CPU e memória. O modelo categoriza as falhas, seleciona as tarefas de maior prioridade e calcula o uso máximo de CPU e memória desse conjunto de tarefas, possibilitando a classificação das próximas tarefas de acordo com o consumo de CPU e memória. Desse modo, quando o consumo atual de CPU ou memória excede o consumo máximo previamente calculado, considera-se que há sobrecarga de uso de recursos e iminente falha.

Contrastando com o trabalho anterior, X. Liu *et al.* (2020) obtiveram melhores resultados na previsão de falhas em nuvem utilizando o algoritmo *Random Forest* em detrimento do algoritmo SVM. Os autores modelaram e implantaram a ferramenta COMPASS com o objetivo de prever falhas de hardware nos servidores da nuvem Baidu. A solução foi avaliada durante um período de três meses e alcançou 93,33% de precisão para previsões de falhas com até 30 minutos de antecedência. Com base na análise dos dados da nuvem Baidu, os autores concluíram que 63% das interrupções do sistema são provocadas por falhas de hardware.

Ainda no contexto de previsão de falhas em hardware na nuvem, Chakrabortii e Litz (2020) propuseram um método com base nos algoritmos *Isolation Forest* e *Autoencoder* para prever falhas em SSDs em servidores na nuvem. Utilizando dados de mais de 30.000 SSDs coletados ao longo de seis anos na nuvem do Google, a abordagem emprega apenas a classe majoritária (sem falhas) no processo de treinamento. O método proposto oferece as vantagens de não sofrer com *overfitting* (sobreajuste aos dados de treinamento), prever falhas inéditas não vistas no conjunto de teste, além de não ser afetado pelo desbalanceamento de classes, efeito comum em modelos de classificação.

4.3 Métricas

Diversos aspectos são considerados na avaliação de modelos preditivos, como desempenho, interpretabilidade e recursos computacionais empregados. As métricas de desempenho avaliam quão bem o modelo produziu o resultado esperado. As métricas comumente utilizadas em modelos de classificação incluem as proporções de previsões corretas e incorretas, como acurácia, *recall*, precisão e *F1 score*. Por sua vez, os modelos de regressão são avaliados pela medição dos erros estimados, como o *Root Mean Square Error* (RMSE) – ou erro quadrático médio – ou o *Mean Absolute Percentage Error* (MAPE) – ou erro percentual absoluto médio (Janiesch; Zschech; Heinrich, 2021).

Utilizada para avaliar o desempenho geral do modelo, a acurácia é definida como a proporção de previsões corretas em relação ao total de previsões. A precisão é a proporção de previsões positivas corretas em relação ao total de previsões positivas. O *recall* refere-se à proporção de exemplos positivos corretamente identificados pelo modelo em relação ao total de exemplos positivos (Jauk; Yang; Schulz, 2019). O *F1 score*, por sua vez, é a média harmônica entre precisão e *recall* (Notaro; Cardoso; Gerndt, 2021).

Além das métricas comumente utilizadas para avaliar modelos de previsão de falhas, alguns autores apresentaram outras métricas significativas para avaliar a abordagem proposta. Em função da particularidade do método utilizado por Chakrabortii e Litz (2020), no qual apenas a classe majoritária foi utilizada no treinamento do modelo, os autores empregaram a *Area Under Curve* (AUC) – ou área sob a curva – para avaliar o modelo, uma vez que as métricas tradicionais poderiam apresentar resultados distorcidos, considerando o risco de *overfitting* para classe majoritária (prevendo-se todas

as amostras para essa classe). Os autores também apresentaram métricas referentes ao tempo de treinamento para cada algoritmo avaliado. De forma semelhante, mas com o objetivo de manter o acordo de confidencialidade com o Alibaba, Li *et al.* (2020) utilizaram exclusivamente a AUC na avaliação do método proposto. Também avaliaram o desempenho do método com diferentes janelas de previsão, obtendo melhores resultados com o algoritmo *Random Forest*, que alcançou uma AUC de 92% para previsões entre 2 e 48 horas de antecedência.

Além da precisão do modelo proposto, Gao, Wang e Shen (2022) consideraram a sobrecarga de tempo de treinamento e teste dos modelos avaliados, buscando um equilíbrio entre ambas as métricas com o objetivo de manter a qualidade do serviço na nuvem. Nessa abordagem, os autores alcançaram 90% de precisão na previsão de falhas de trabalhos e tarefas com 15 minutos de antecedência. Entre os trabalhos que apresentaram o tempo para previsão de falhas, o de Das, Mueller e Rountree (2020) foi o que apresentou métricas de tempo de previsão mais curtas. Os autores apresentaram um *framework* para prever falhas em nós de computação em nuvem com tempo de previsão de três minutos de antecedência. O *framework* obteve *recall*, precisão e acurácia de 86%, 88% e 80%, respectivamente. Por outro lado, Chakrabortii e Litz (2020) avaliaram o desempenho de dois diferentes algoritmos (*Isolation Forest* e *Autoencoder*) com tempos de previsão entre um e quatro dias, caracterizando a abordagem com maior tempo de previsão identificada nesta RSL.

Embora tenham apresentado o fluxograma de atividades do método proposto, Su *et al.* (2020) não apresentaram nenhuma métrica para avaliação da abordagem proposta, tampouco os resultados obtidos na avaliação experimental. De forma semelhante, Adamu *et al.* (2017) não apresentaram a métrica utilizada na avaliação do método proposto, mas informaram a quantidade de falhas previstas no conjunto de dados.

5 Discussão

Nesta seção, são discutidos os principais resultados obtidos, com o objetivo de responder às questões de pesquisa, além de serem apresentadas observações com base nos trabalhos analisados.

5.1 Desbalanceamento de classes

Em aprendizagem de máquina e estatística, uma característica recorrente nos conjuntos de dados é o desbalanceamento de classes. Nesse processo, as classes que se deseja prever não apresentam a mesma proporção no conjunto de dados e, normalmente, a classe alvo é a minoritária. Essa característica pode resultar em modelos com baixo desempenho (Mohammed; Rawashdeh; Abdullah, 2020). O desequilíbrio de classes é geralmente tratado com estratégias de amostragem, como *oversampling* ou *undersampling*. O *oversampling* adiciona amostras da classe minoritária, enquanto o *undersampling* remove, aleatoriamente, amostras da classe majoritária (Li *et al.*, 2020). Entre os trabalhos analisados, poucos autores utilizaram técnicas para lidar com o desequilíbrio de classes. Shetty, Sajjan e G. (2019) e Chen *et al.* (2019) utilizaram a Técnica de Sobreamostragem Minoritária Sintética (*Synthetic Minority Oversampling Technique – SMOTE*), que consiste em gerar amostras sintéticas da classe minoritária, reduzindo ou eliminando o desequilíbrio entre as classes. Embora Lin *et al.* (2018)

também tenham utilizado a SMOTE, os autores relataram resultados pouco satisfatórios, com altas taxas de falsos positivos.

Li *et al.* (2020) implementaram e aprimoraram uma técnica de *oversampling* para selecionar mais amostras positivas, capturando um número maior de sintomas de falhas. Nos experimentos, os autores relataram melhores resultados com o uso da técnica proposta em comparação com outras técnicas de amostragem. Em virtude do desequilíbrio excessivo dos dados utilizados, Nam *et al.* (2021) aplicaram *oversampling* 2 vezes nos dados de falhas e *undersampling* 60 vezes nos dados normais.

Em contrapartida, Luo *et al.* (2021) propuseram um método denominado *Temporal Progressive Sampling* (TPS) para tratar o desequilíbrio de classes, o qual gerava novas amostras da classe minoritária em função do tempo, sendo definido, pelos autores, como um método para aprimoramento de dados. Os demais trabalhos incluídos nesta revisão sistemática não utilizaram técnicas para tratar o desequilíbrio de classes.

5.2 Sobre os métodos

Em conformidade com a pesquisa recente realizada por Bhanage, Pawar e Kotecha (2021), evidenciou-se, neste estudo, a predominância de técnicas de aprendizagem de máquina nas abordagens de previsão de falhas – em especial, os algoritmos baseados em árvores, como *Random Forest* e *Decision Tree*. Essa tendência possivelmente se explica pelos excelentes resultados obtidos a partir desses modelos, que superaram técnicas como SVM e redes neurais (Jauk; Yang; Schulz, 2019). Em concordância com Bhanage, Pawar e Kotecha (2021), foi constatado, nesta revisão de literatura, um aumento no uso de técnicas de aprendizagem profunda nos últimos anos, destacando-se o uso amplo do algoritmo LSTM.

Em relação à natureza das falhas a serem previstas em sistemas em nuvem, observou-se a predominância de métodos voltados para a previsão de falhas em trabalhos e tarefas (14 artigos), seguidos por falhas de servidor (10 artigos), serviços ou software de forma geral (6 artigos), falhas de máquina virtual (5 artigos) e falhas de disco (5 artigos).

Destaca-se que, em relação aos métodos que utilizaram o conjunto de dados Google 2011 V.2, alguns autores consideraram mais de um estado final de trabalhos e tarefas como falhas. Bhattacharyya *et al.* (2017) e Shetty, Sajjan e G. (2019) classificaram as tarefas encerradas como falhas, enquanto Islam e Manivannan (2017), Liu *et al.* (2017) e C. Liu *et al.* (2020) classificaram as tarefas removidas e encerradas como falhas. Além dessas, Asmawi, Ismail e Shen (2022) classificaram as tarefas e os trabalhos perdidos como falhas. A classificação desses estados de trabalhos e tarefas como falhas deve ser cuidadosamente avaliada, pois pode impactar os resultados dos métodos propostos ao se adicionarem dados aos modelos.

Outro aspecto relevante em relação aos métodos que utilizaram os dados do Google diz respeito à quantidade de dados empregada nos modelos de previsão de falhas. Embora o conjunto de dados Google 2011 V.2 consista em um registro de 29 dias, alguns autores utilizaram uma fração desses dados, possivelmente devido a limitações de recursos computacionais para a realização dos experimentos. Liu *et al.* (2017) utilizaram as 12 primeiras horas do conjunto de dados, enquanto C. Liu *et al.* (2020) utilizaram os 3 primeiros dias. Jassas e Mahmoud (2020) limitaram-se aos primeiros 7 dias, enquanto Asmawi, Ismail e Shen (2022) utilizaram os primeiros 14 dias do

conjunto de dados. Como uma fração dos dados pode não representar o comportamento do conjunto completo, os resultados apresentados podem ser distorcidos.

Ainda sobre os métodos que utilizaram o *dataset* Google 2011 V.2, alguns autores relataram alta incidência de falhas nos dias iniciais do conjunto de dados. Asmawi, Ismail e Shen (2022), por exemplo, identificaram uma proporção anormal de tarefas com falhas nos primeiros 14 dias. Os autores ainda relataram a existência de três falhas para cada tarefa finalizada. De forma semelhante, Jassas e Mahmoud (2020) observaram alta incidência de falhas nos dias iniciais do *dataset*. Como esses autores não trataram essa alta incidência de falhas nos dias iniciais, os resultados podem ter sido superestimados em função da predominância de tarefas com falhas nos dados utilizados.

5.3 Sobre os conjuntos de dados

Embora já tenha transcorrido mais de uma década desde o lançamento do *dataset* Google 2011 V.2, evidenciou-se neste trabalho a preferência dos pesquisadores por esse conjunto de dados em detrimento de outros disponibilizados publicamente. Essa preferência reforça a dependência excessiva apontada por Amvrosiadis *et al.* (2018). Como alternativa ao *dataset* de 2011 V.2, o *dataset* Google 2019 V.3, lançado no ano de 2020, pode ser utilizado nas pesquisas sobre previsão de falhas em nuvem, possivelmente refletindo de forma mais precisa a realidade dos sistemas atualmente em uso.

Quanto à disposição dos dados, ou seja, quanto ao formato em que são apresentados, observou-se que alguns conjuntos de dados disponibilizados publicamente, como Google 2011 V.2 e Google 2019 V.3, possuem dados ofuscados ou normalizados. A ofuscação visa preservar a confidencialidade de dados considerados sigilosos pela organização, enquanto a normalização é uma técnica comumente empregada nas abordagens de modelagem preditiva, com o objetivo de redimensionar os dados para um intervalo comum, contribuindo para o aprendizado dos algoritmos de aprendizagem de máquina (Singh; Singh, 2020). Tanto o conjunto de dados de 2011 quanto o de 2019 têm os dados referentes ao uso de memória RAM e CPU normalizados para o valor máximo de 1, o que impossibilita a identificação dos valores reais disponíveis e utilizados nas operações do sistema, como a quantidade de memória e capacidade de processamento disponíveis nos servidores.

5.4 Sobre as métricas

No que se refere às métricas para avaliação dos métodos propostos, observou-se que os autores Saxena e Singh (2022a, 2022b) utilizaram exclusivamente a acurácia. Entretanto, o uso exclusivo dessa métrica não é adequado para a avaliação de modelos com conjuntos de dados desbalanceados. No contexto de previsão de falhas em tarefas, um modelo que sempre classifica todas as tarefas como sem falhas terá alta acurácia, mesmo que ignore as tarefas com falhas, pois, nos conjuntos de dados, estas são exceções (Jauk; Yang; Schulz, 2019). Notaro, Cardoso e Gerndt (2021) consideraram a acurácia uma métrica enganosa para avaliação da qualidade das previsões, pois, na maioria das vezes, essa métrica explora a assimetria dos dados. Embora a AUC seja recomendada para avaliar modelos com dados desbalanceados (Fahim; Sillitti, 2019), somente Padmakumari e Umamakeswari (2019), Chakrabortii e Litz (2020), Li *et al.* (2020) e Yang e Kim (2022) utilizaram essa métrica.

6 Conclusão

A previsão de falhas nos sistemas de computação em nuvem é um tema que tem atraído considerável atenção nos últimos anos. Dado o número de métodos propostos para essa finalidade, esta revisão sistemática da literatura examina as características desses métodos, incluindo as técnicas utilizadas, os conjuntos de dados e as métricas empregadas para sua avaliação. Para tanto, foram analisados 40 artigos de um total de 2.580 retornados na busca inicial das bases científicas ACM, IEEE, Scopus e SpringerLink.

As lacunas de pesquisa identificadas incluem a ausência de estudos com foco na previsão do tempo de ocorrência das falhas, assim como a escassez de abordagens com o *dataset* Google 2019 V.3. Evidencia-se a necessidade de disponibilização de conjuntos de dados do mundo real, uma vez que atualmente há uma oferta limitada de *datasets* públicos.

Esforços de pesquisa futuros podem explorar métodos para agregar amostras de dados de trabalhos e tarefas no *dataset*, além de técnicas para balanceamento de classes em modelos de previsão de falhas em nuvem.

Esta revisão sistemática da literatura apresenta o estado atual das pesquisas sobre a previsão de falhas nos sistemas em nuvem. Acredita-se que as observações e tendências aqui identificadas podem ser úteis para pesquisas futuras. A metodologia empregada permite a reprodução deste estudo pela comunidade acadêmica.

Financiamento

Esta pesquisa foi financiada pela Universidade Federal Rural de Pernambuco (UFRPE).

Conflito de interesses

Os autores declaram não haver conflito de interesses.

Nota

Este artigo foi feito com base na dissertação de José Carlos dos Santos para o curso de Mestrado em Informática Aplicada da Universidade Federal Rural de Pernambuco (UFRPE), disponível em: <https://www.ppgia.ufrpe.br/sites/default/files/testes-dissertacoes/APRENDIZADO%20DE%20M%C3%81QUINA%20APLICADO%20%C3%80%20PREDI%C3%87%C3%83O%20DE%20FALHAS%20DE%20SOFTWARE%20NA%20COMPUTA%C3%87%C3%83O%20EM%20NUVEM.pdf>.

Contribuições ao artigo

SANTOS, J. C.: concepção ou desenho do estudo/pesquisa; análise e/ou interpretação dos dados; revisão final com participação crítica e intelectual no manuscrito.
CARVALHO, T B. A.; SOUSA, E. T. G.: revisão final com participação crítica e

intelectual no manuscrito. Todos os autores participaram da escrita, discussão, leitura e aprovação da versão final do artigo.

Referências

ABRO, J. H.; LI, C.; SHAFIQ, M.; VISHNUKUMAR, A.; MEWADA, S.; MALPANI, K.; OSEI-OWUSU, J. Artificial intelligence enabled effective fault prediction techniques in cloud computing environment for improving resource optimization. **Scientific Programming**, v. 2022, 7432949, 2022. DOI: <https://doi.org/10.1155/2022/7432949>.

ADAMU, H.; MOHAMMED, B.; MAINA, A. B.; CULLEN, A.; UGAIL, H.; AWAN, I. An approach to failure prediction in a cloud based environment. *In: IEEE INTERNATIONAL CONFERENCE ON FUTURE INTERNET OF THINGS AND CLOUD (FICLOUD)*, 5., 2017, Prague. **Proceedings** [...]. Prague: IEEE, 2017. DOI: <https://doi.org/10.1109/FiCloud.2017.56>.

ALAHMAD, Y.; DARADKEH, T.; AGARWAL, A. Proactive failure-aware task scheduling framework for cloud computing. **IEEE Access**, v. 9, p. 106152-106168, 2021. DOI: <https://doi.org/10.1109/ACCESS.2021.3101147>.

AMVROSIADIS, G.; PARK, J. W.; GANGER, G. R.; GIBSON, G. A.; BASEMAN, E.; DEBARDELEBEN, N. On the diversity of cluster workloads and its impact on research results. *In: 2018 USENIX ANNUAL TECHNICAL CONFERENCE (USENIX ATC'18)*, 2018, Boston. **Proceedings** [...]. Boston: USENIX, 2018. Disponível em: <https://www.usenix.org/conference/atc18/presentation/amvrosiadis>. Acesso em: 10 jul. 2023.

ASMAWI, T. N. T.; ISMAIL, A.; SHEN, J. Cloud failure prediction based on traditional machine learning and deep learning. **Journal of Cloud Computing**, v. 11, 47, 2022. DOI: <https://doi.org/10.1186/s13677-022-00327-0>.

BHANAGE, D. A.; PAWAR, A. V.; KOTECHA, K. IT infrastructure anomaly detection and failure handling: a systematic literature review focusing on datasets, log preprocessing, machine & deep learning approaches and automated tool. **IEEE Access**, v. 9, p. 156392-156421, 2021. DOI: <https://doi.org/10.1109/ACCESS.2021.3128283>.

BHATTACHARYYA, A.; SINGH, H.; JANDEGHI, S. A. J.; AMZA, C. Online detection of anomalous applications on the cloud. *In: ANNUAL INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND SOFTWARE ENGINEERING*, 27., 2017, Markhan. **Proceedings** [...]. Markhan: ACM, 2017. p. 161-169. Disponível em: <https://dl.acm.org/doi/10.5555/3172795.3172814>. Acesso em: 8 maio 2023.

BUI, K. T.; VO, L. V.; NGUYEN, C. M.; PHAM, T. V.; TRAN, H. C. A fault detection and diagnosis approach for multi-tier application in cloud computing. **Journal of Communications and Networks**, v. 22, n. 5, p. 399-414, 2020. DOI: <https://doi.org/10.1109/JCN.2020.000023>.

BUYYA, R.; SRIRAMA, S. N.; CASALE, G.; CALHEIROS, R.; SIMMHAN, Y.; VARGHESE, B.; GELENBE, E.; JAVADI, B.; VAQUERO, L. M.; STELMAR NETTO, M. A.; TOOSI, A. N.; RODRIGUEZ, M. A.; LLORENTE, I. M.; DI VIMERCATI, S. D. C.; SAMARATI, P.; MILOJICIC, D.; VARELA, C.; BAHSOON, R.; ASSUNCAO, M. D.;

RANA, O.; ZHOU, W.; JIN, H.; GENTZSCH, W.; ZOMAYA, A. Y.; SHEN, H. A manifesto for future generation cloud computing: research directions for the next decade. **ACM Computing Surveys**, v. 51, n. 5, 105, 2018. DOI: <https://doi.org/10.1145/3241737>.

CHAKRABORTTI, C.; LITZ, H. Improving the accuracy, adaptability, and interpretability of SSD failure prediction models. *In: ACM SYMPOSIUM ON CLOUD COMPUTING (SoCC'20)*, 11., 2020, Virtual Event. **Proceedings** [...]. Virtual Event: ACM, 2020. p. 120-133. DOI: <https://doi.org/10.1145/3419111.3421300>.

CHEN, Y.; YANG, X.; LIN, Q.; ZHANG, H.; GAO, F.; XU, Z.; DANG, Y.; ZHANG, D.; DONG, H.; XU, Y.; LI, H.; KANG, Y. Outage prediction and diagnosis for cloud service systems. *In: THE WORLD WIDE WEB CONFERENCE (WWW'19)*, 2019, San Francisco. **Proceedings** [...]. San Francisco: ACM, 2019. p. 2659-2665. DOI: <https://doi.org/10.1145/3308558.3313501>.

CHHETRI, T. R.; DEHURY, C. K.; LIND, A.; SRIRAMA, S. N.; FENSEL, A. A combined system metrics approach to cloud service reliability using artificial intelligence. **Big Data and Cognitive Computing**, v. 6, n. 1, 26, 2022. DOI: <https://doi.org/10.3390/bdcc6010026>.

DAS, A.; MUELLER, F.; ROUNTREE, B. Aarohi: making real-time node failure prediction feasible. *In: 2020 IEEE INTERNATIONAL PARALLEL AND DISTRIBUTED PROCESSING SYMPOSIUM (IPDPS)*, 2020, New Orleans. **Proceedings** [...]. New Orleans: IEEE, 2020. DOI: <https://doi.org/10.1109/IPDPS47924.2020.00115>.

DIAS, A. H. T.; CORREIA, L. H. A.; MALHEIROS, N. A systematic literature review on virtual machine consolidation. **ACM Computing Surveys**, v. 54, n. 8, 176, 2021. DOI: <https://doi.org/10.1145/3470972>.

FAHIM, M.; SILLITTI, A. Anomaly detection, analysis and prediction techniques in IoT environment: a systematic literature review. **IEEE Access**, v. 7, p. 81664-81681, 2019. DOI: <https://doi.org/10.1109/ACCESS.2019.2921912>.

GAO, J.; WANG, H.; SHEN, H. Task failure prediction in cloud data centers using deep learning. **IEEE Transactions on Services Computing**, v. 15, n. 3, p. 1411-1422, 2022. DOI: <https://doi.org/10.1109/TSC.2020.2993728>.

GOKHROO, M. K.; GOVIL, M. C.; PILLI, E. S. Detecting and mitigating faults in cloud computing environment. *In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE & COMMUNICATION TECHNOLOGY (CICT)*, 3., 2017, Ghaziabad. **Proceedings** [...]. Ghaziabad: IEEE, 2017. DOI: <https://doi.org/10.1109/CICT.2017.7977362>.

GOLLAPALLI, M.; ALMETRIK, M. A.; ALNAJRANI, B. S.; ALOMARI, A. A.; ALDAWOU, S. H.; ALMUNSOUR, Y. Z.; ABDULQADER, M. M.; ALOUP, K. M. Task failure prediction using machine learning techniques in the Google cluster trace cloud computing environment. **Mathematical Modelling of Engineering Problems**, v. 9, n. 2, p. 545-553, 2022. DOI: <https://doi.org/10.18280/mmep.090234>.

GUO, J.; CHANG, Z.; WANG, S.; DING, H.; FENG, Y.; MAO, L.; BAO, Y. Who limits the resource efficiency of my datacenter: an analysis of Alibaba datacenter traces. *In: INTERNATIONAL SYMPOSIUM ON QUALITY OF*

SERVICE (IWQoS'19), 2019, Phoenix. **Proceedings** [...]. Phoenix: ACM, 2019. 39. DOI: <https://doi.org/10.1145/3326285.3329074>.

GUPTA, S.; MUTHIYAN, N.; KUMAR, S.; NIGAM, A.; DINESH, D. A. A supervised deep learning framework for proactive anomaly detection in cloud workloads. *In: IEEE INDIA COUNCIL INTERNATIONAL CONFERENCE (INDICON), 14., 2017, Roorkee. Proceedings* [...]. Roorkee: IEEE, 2017. DOI: <https://doi.org/10.1109/INDICON.2017.8488109>.

HERBST, N.; BAUER, A.; KOUNEV, S.; OIKONOMOU, G.; VAN EYK, E.; KOUSIOURIS, G.; EVANGELINO, A.; KREBS, R.; BRECHT, T.; ABAD, C. L.; IOSUP, A. Quantifying cloud performance and dependability: Taxonomy, metric design, and emerging challenges. **ACM Transactions on Modeling and Performance Evaluation of Computing Systems**, v. 3, n. 4, 19, 2018. DOI: <https://doi.org/10.1145/3236332>.

HERMIAS, J. P.; TEKNOMO, K.; MONJE, J. C. N. Short-term stochastic load forecasting using autoregressive integrated moving average models and Hidden Markov Model. *In: 2017 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES (ICICT), 2017, Karachi. Proceedings* [...]. Karachi: IEEE, 2017. DOI: <https://doi.org/10.1109/ICICT.2017.8320177>.

IRRERA, I. **Fault injection for online failure prediction assessment and improvement**. 2015. Tese (Doutorado em Ciências e Tecnologias da Informação) – Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, 2016. Disponível em: <https://hdl.handle.net/10316/29182>. Acesso em: 14 dez. 2023.

ISLAM, T.; MANIVANNAN, D. Predicting application failure in cloud: a machine learning approach. *In: 2017 IEEE INTERNATIONAL CONFERENCE ON COGNITIVE COMPUTING (ICCC), 2017, Honolulu. Proceedings* [...]. Honolulu: IEEE, 2017. DOI: <https://doi.org/10.1109/IEEE.ICCC.2017.11>.

JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. **Electronic Markets**, v. 31, p. 685-695, 2021. DOI: <https://doi.org/10.1007/s12525-021-00475-2>.

JASSAS, M. S.; MAHMOUD, Q. H. A failure prediction model for large scale cloud applications using deep learning. *In: 2021 IEEE INTERNATIONAL SYSTEMS CONFERENCE (SysCon), 2021, Vancouver. Proceedings* [...]. Vancouver: IEEE, 2021. DOI: <https://doi.org/10.1109/SysCon48628.2021.9447141>.

JASSAS, M. S.; MAHMOUD, Q. H. Analysis of job failure and prediction model for cloud computing using machine learning. **Sensors**, v. 22, n. 5, 2035, 2022. DOI: <https://doi.org/10.3390/s22052035>.

JASSAS, M. S.; MAHMOUD, Q. H. Evaluation of a failure prediction model for large scale cloud applications. *In: GOUTTE, C.; ZHU, X. (ed.). Advances in Artificial Intelligence: Canadian AI 2020*. Cham: Springer, 2020. (Lecture Notes in Computer Science, v. 12109). DOI: https://doi.org/10.1007/978-3-030-47358-7_32.

JAUK, D.; YANG, D.; SCHULZ, M. Predicting faults in high performance computing systems: an in-depth survey of the state-of-the-practice. *In: INTERNATIONAL*

CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS (SC'19), 2019, Denver. **Proceedings** [...]. Denver: ACM, 2019. 30. DOI: <https://doi.org/10.1145/3295500.3356185>.

KABIR, H. M. D.; KHOSRAVI, A.; MONDAL, S. K.; RAHMAN, M.; NAHAVANDI, S.; BUYYA, R. Uncertainty-aware decisions in cloud computing: foundations and future directions. **ACM Computing Surveys**, v. 54, n. 4, 74, 2021. DOI: <https://doi.org/10.1145/3447583>.

KASHANI, M. H.; MAHDIPOUR, E. Load balancing algorithms in fog computing: A systematic review. **IEEE Transactions on Services Computing**, v. 16, n. 2, p. 1505-1521, 2022. DOI: <https://doi.org/10.1109/TSC.2022.3174475>.

KHALIL, M. H.; SHETA, W. M.; ELMAGHRABY, A. S. Categorizing hardware failure in large scale cloud computing environment. *In*: 2016 IEEE INTERNATIONAL SYMPOSIUM ON SIGNAL PROCESSING AND INFORMATION TECHNOLOGY (ISSPIT), 2016, Limassol. **Proceedings** [...]. Limassol: IEEE, 2016. DOI: <https://doi.org/10.1109/ISSPIT.2016.7886058>.

LI, Y.; JIANG, Z. M. J.; LI, H.; HASSAN, A. E.; HE, C.; HUANG, R.; ZENG, Z.; WANG, M.; CHEN, P. Predicting node failures in an ultra-large-scale cloud computing platform: an AIOps solution. **ACM Transactions on Software Engineering Methodology**, v. 29, n. 2, 13, 2020. DOI: <https://doi.org/10.1145/3385187>.

LIANG, C.; DENG, L.; ZHU, J.; CAO, Z.; LI, C. Disk failure prediction based on SW-disk feature engineering. *In*: IEEE INTERNATIONAL CONFERENCE ON BIG DATA SECURITY ON CLOUD (BIGDATASECURITY), IEEE INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE AND SMART COMPUTING (HPSC) AND IEEE INTERNATIONAL CONFERENCE ON INTELLIGENT DATA AND SECURITY (IDS), 8., 2022, Jinan. **Proceedings** [...]. Jinan: IEEE, 2022. DOI: <https://doi.org/10.1109/BigDataSecurityHPSCIDS54978.2022.00027>.

LIN, Q.; HSIEH, K.; DANG, Y.; ZHANG, H.; SUI, K.; XU, Y.; LOU, J.-G.; LI, C.; WU, Y.; YAO, R.; CHINTALAPATI, M.; ZHANG, D. Predicting node failure in cloud service systems. *In*: ACM JOINT MEETING ON EUROPEAN SOFTWARE ENGINEERING CONFERENCE AND SYMPOSIUM ON THE FOUNDATIONS OF SOFTWARE ENGINEERING, 26., 2018, Lake Buena Vista. **Proceedings** [...]. Lake Buena Vista: ACM, 2018. p. 480-490. DOI: <https://doi.org/10.1145/3236024.3236060>.

LIU, C.; DAI, L.; LAI, Y.; LAI, G.; MAO, W. Failure prediction of tasks in the cloud at an earlier stage: a solution based on domain information mining. **Computing**, v. 102, p. 2001-2003, 2020. DOI: <https://doi.org/10.1007/s00607-020-00800-1>.

LIU, C.; HAN, J.; SHANG, Y.; LIU, C.; CHENG, B.; CHEN, J. Predicting of job failure in compute cloud based on online extreme learning machine: a comparative study. **IEEE Access**, v. 5, p. 9359-9368, 2017. DOI: <https://doi.org/10.1109/ACCESS.2017.2706740>.

LIU, D.; WANG, B.; LI, P.; STONES, R. J.; MARBACH, T. G.; WANG, G.; LIU, X.; LI, Z. Predicting hard drive failures for cloud storage systems. *In*: WEN, S.; ZOMAYA, A.; YANG, L. T. (ed.). **Algorithms and Architectures for Parallel Processing**: ICA3PP 2019. Cham: Springer, 2020. (Lecture Notes in Computer Science, v. 11944). DOI: https://doi.org/10.1007/978-3-030-38991-8_25.

LIU, X.; HE, Y.; LIU, H.; ZHANG, J.; LIU, B.; PENG, X.; XU, J.; ZHANG, J.; ZHOU, A.; SUN, P.; ZHU, K.; NISHI, A.; ZHU, D.; ZHANG, K. Smart server crash prediction in cloud service data center. *In: IEEE INTERSOCIETY CONFERENCE ON THERMAL AND THERMOMECHANICAL PHENOMENA IN ELECTRONIC SYSTEMS (ITherm), 19., 2020, Orlando. Proceedings [...].* Orlando: IEEE, 2020. DOI: <https://doi.org/10.1109/ITherm45881.2020.9190321>.

LUO, C.; ZHAO, P.; QIAO, B.; WU, Y.; ZHANG, H.; WU, W.; LU, W.; DANG, Y.; RAJMOHAN, S.; LIN, Q.; ZHANG, D. NTAM: Neighborhood-temporal attention model for disk failure prediction in cloud platforms. *In: THE WEB CONFERENCE 2021 (WWW'21), 2021, Ljubljana. Proceedings [...].* Ljubljana: ACM, 2021. p. 1181-1191. DOI: <https://doi.org/10.1145/3442381.3449867>.

MA, Y.; WU, S.; GONG, S.; XU, C. Artificial intelligence-based cloud data center fault detection method. *In: IEEE JOINT INTERNATIONAL INFORMATION TECHNOLOGY AND ARTIFICIAL INTELLIGENCE CONFERENCE (ITAIC), 9., 2020, Chongqing. Proceedings [...].* Chongqing: IEEE, 2020. DOI: <https://doi.org/10.1109/ITAIC49862.2020.9338789>.

MAJID, H.; ANUAR, S. A systematic literature review of failure prediction in production environment using machine learning technique. **International Journal of Innovative Computing**, v. 12, n. 1, p. 81-90, 2022. DOI: <https://doi.org/10.11113/ijic.v12n1.348>.

MARIANI, L.; MONNI, C.; PEZZÉ, M.; RIGANELLI, O.; XIN, R. Localizing faults in cloud systems. *In: IEEE INTERNATIONAL CONFERENCE ON SOFTWARE TESTING, VERIFICATION AND VALIDATION (ICST), 11., 2018, [S.l.]. Proceedings [...].* [S.l.]: IEEE, 2018. DOI: <https://doi.org/10.1109/ICST.2018.00034>.

MELL, P. M.; GRANCE, T. **The NIST definition of cloud computing**: Recommendations of the National Institute of Standards and Technology. Gaithersburg: NIST, 2011. Disponível em: <https://dl.acm.org/doi/book/10.5555/2206223>. Acesso em: 15 dez. 2023.

MOHAMMED, B.; AWAN, I.; UGAIL, H.; YOUNAS, M. Failure prediction using machine learning in a virtualised HPC system and application. **Cluster Computing**, v. 22, p. 471-485, 2019. DOI: <https://doi.org/10.1007/s10586-019-02917-1>.

MOHAMMED, R.; RAWASHDEH, J.; ABDULLAH, M. Machine learning with oversampling and undersampling techniques: overview study and experimental results. *In: INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION SYSTEMS (ICICS), 11., 2020, Irbid. Proceedings [...].* Irbid: IEEE, 2020. DOI: <https://doi.org/10.1109/ICICS49469.2020.2395556>.

MONNI, C.; PEZZÈ, M.; PRISCO, G. An RBM anomaly detector for the cloud. *In: IEEE CONFERENCE ON SOFTWARE TESTING, VALIDATION AND VERIFICATION (ICST), 12., 2019, Xi'an. Proceedings [...].* Xi'an: IEEE, 2019. DOI: <https://doi.org/10.1109/ICST.2019.00024>.

NAM, S.; HONG, J.; YOO, J.-H.; HONG, J. W.-K. Virtual machine failure prediction using log analysis. *In: ASIA-PACIFIC NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM (APNOMS), 22., 2021, Tainan. Proceedings [...].* Tainan: IEEE, 2021. DOI: <https://doi.org/10.23919/APNOMS52696.2021.9562588>.

NOTARO, P.; CARDOSO, J.; GERNDT, M. A survey of AIOps methods for failure management. **ACM Transactions on Intelligent Systems and Technology**, v. 12, n. 6, 81, 2021. DOI: <https://doi.org/10.1145/3483424>.

PADMAKUMARI, P.; UMAMAKESWARI, A. Task failure prediction using combine bagging ensemble (CBE) classification in cloud workflow. **Wireless Personal Communications**, v. 107, p. 23-40, 2019. DOI: <https://doi.org/10.1007/s11277-019-06238-9>.

PRATHIBHA, S. Investigating the performance of machine learning algorithms for improving fault tolerance for large scale workflow applications in cloud computing. *In*: 2019 INTERNATIONAL CONFERENCE ON COMPUTATIONAL INTELLIGENCE AND KNOWLEDGE ECONOMY (ICCIKE), 2019, Dubai. **Proceedings** [...]. Dubai: IEEE, 2019. DOI: <https://doi.org/10.1109/ICCIKE47802.2019.9004379>.

RAWAT, A.; BHADORIA, R. S. Accuracy estimation for fault classification in virtual machine using deep learning. *In*: INTERNATIONAL CONFERENCE ON SECURE CYBER COMPUTING AND COMMUNICATIONS (ICSCCC), 2., 2021, Jalandhar. **Proceedings** [...]. Jalandhar: IEEE, 2021. DOI: <https://doi.org/10.1109/ICSCCC51823.2021.9478157>.

RAWAT, A.; SUSHIL, R.; AGARWAL, A.; SIKANDER, A. A new approach for VM failure prediction using stochastic model in cloud. **IETE Journal of Research**, v. 67, n. 2, p. 165-172, 2021. DOI: <https://doi.org/10.1080/03772063.2018.1537814>.

ROUMANI, Y.; NWANKPA, J. K. An empirical study on predicting cloud incidents. **International Journal of Information Management**, v. 47, p. 131-139, 2019. DOI: <https://doi.org/10.1016/j.ijinfomgt.2019.01.014>.

SANTOS, V. A.; MANACERO, A.; LOBATO, R. S.; SPOLON, R.; CAVENAGHI, M. A. A systematic review of fault tolerance solutions for communication errors in open source cloud computing. *In*: IBERIAN CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGIES (CISTI), 15., 2020, Seville. **Proceedings** [...]. Seville: IEEE, 2020. DOI: <https://doi.org/10.23919/CISTI49556.2020.9140933>.

SAXENA, D.; SINGH, A. K. OFP-TM: an online VM failure prediction and tolerance model towards high availability of cloud computing environments. **The Journal of Supercomputing**, v. 78, p. 8003-8024, 2022a. DOI: <https://doi.org/10.1007/s11227-021-04235-z>.

SAXENA, D.; SINGH, A. K. VM failure prediction based intelligent resource management model for cloud environments. *In*: INTERNATIONAL CONFERENCE ON POWER, CONTROL AND COMPUTING TECHNOLOGIES (ICPC2T), 2., 2022, Raipur. **Proceedings** [...]. Raipur: IEEE, 2022b. DOI: <https://doi.org/10.1109/ICPC2T53885.2022.9777020>.

SHAO, Y.; ZHANG, Y. A failure prediction method for spacecraft loads based on time series model. *In*: INTERNATIONAL CONFERENCE ON RELIABILITY, MAINTAINABILITY, AND SAFETY (ICRMS), 12., 2018, Shanghai. **Proceedings** [...]. Shanghai: IEEE, 2018. DOI: <https://doi.org/10.1109/ICRMS.2018.00059>.

SHETTY, J.; SAJJAN, R.; SHOBHA G. Task resource usage analysis and failure prediction in cloud. *In*: INTERNATIONAL CONFERENCE ON CLOUD COMPUTING, DATA SCIENCE & ENGINEERING (Confluence), 9., 2019, Noida. **Proceedings** [...]. Noida: IEEE, 2019. DOI: <https://doi.org/10.1109/CONFLUENCE.2019.8776612>.

SINGH, D.; SINGH, B. Investigating the impact of data normalization on classification performance. **Applied Soft Computing**, v. 97, Part B, 105524, 2020. DOI: <https://doi.org/10.1016/j.asoc.2019.105524>.

SU, C.-J.; TSAI, L.-C.; HUANG, S.-F.; LI, Y. Deep learning-based real-time failure detection of storage devices. *In*: AHRAM, T. (ed.). **Advances in Artificial Intelligence, Software and Systems Engineering: AHFE 2019**. Cham: Springer, 2020. p. 160-168. (Advances in Intelligent Systems and Computing, v. 965). DOI: https://doi.org/10.1007/978-3-030-20454-9_16.

TEHRANI, A. F.; SAFI-ESFAHANI, F. A threshold sensitive failure prediction method using support vector machine. **Multiagent and Grid Systems**, v. 13, n. 2, p. 97-111, 2017. DOI: <https://dx.doi.org/10.3233/MGS-170263>.

VU, D. D.; VU, X. T.; KIM, Y. Deep learning-based fault prediction in cloud system. *In*: 2021 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGY CONVERGENCE (ICTC), 2021, Jeju Island. **Proceedings** [...]. Jeju Island: IEEE, 2021. DOI: <https://doi.org/10.1109/ICTC52510.2021.9620802>.

WILKES, J. Yet more Google compute cluster trace data. **Google Research Blog**, 2020. Disponível em: <https://blog.research.google/2020/04/yet-more-google-compute-cluster-trace.html>. Acesso em: 15 dez. 2023.

YANG, H.; KIM, Y. Design and implementation of machine learning-based fault prediction system in cloud infrastructure. **Electronics**, v. 11, n. 22, 3765, 2022. DOI: <https://doi.org/10.3390/electronics11223765>.

ZHANG, P.; WANG, Y.; MA, X.; XU, Y.; YAO, B.; ZHENG, X.; JIANG, L. Predicting DRAM-caused node unavailability in hyper-scale clouds. *In*: ANNUAL IEEE/IFIP INTERNATIONAL CONFERENCE ON DEPENDABLE SYSTEMS AND NETWORKS (DSN), 52., 2022, Baltimore. **Proceedings** [...]. Baltimore: IEEE, 2022. DOI: <https://doi.org/10.1109/DSN53405.2022.00037>.