

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

José Carlos dos Santos^{[1]*}, Tiago Buarque Assunção de Carvalho^[2], Erica Teixeira Gomes de Souza^[3]

^[1] carlos.santos2@ufrpe.br, ^[3] erica.sousa@ufrpe.br. Universidade Federal Rural de Pernambuco (UFRPE), Brasil

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

* autor correspondente

Resumo

Atualmente, os serviços de computação em nuvem são fundamentais para a economia moderna. Diversos setores têm reorientado seus serviços para a nuvem. No entanto, a complexidade e natureza distribuída desses sistemas contribuem para a ocorrência de diversos tipos de falhas que, quando não tratadas, podem resultar em danos catastróficos. Para mitigar os efeitos das falhas, diversos métodos de previsão de falhas foram propostos. Como esses métodos mudam com o passar do tempo, esse trabalho tem como objetivo apresentar uma revisão sistemática da literatura a respeito das técnicas e métodos utilizados na previsão de falhas em nuvem. Nesse sentido, 40 artigos de quatro bases de dados cientificamente reconhecidas foram analisados. Ficou evidenciada a predominância de técnicas de aprendizagem de máquina baseadas em árvore de decisão, assim como a escassez de conjuntos de dados do mundo real para o avanço das pesquisas de previsão de falhas em nuvem. Esforços de pesquisa futuros podem avaliar métodos para agregar as amostras de dados de trabalhos e tarefas no conjunto de dados, bem como técnicas para balanceamento de classes em modelos de previsão de falhas em nuvem. Essa revisão sistemática da literatura apresenta o estado atual 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

Currently, cloud computing services are fundamental to the modern economy. Various sectors have reoriented their services to the cloud. However, the complexity and distributed nature of these systems contribute to the occurrence of various types of failures, which when not addressed, can result in catastrophic damage. To mitigate the effects of failures, various failure prediction methods have been proposed. As these methods change over time, this work aims to present a systematic review of the literature on the techniques and methods used in cloud failure prediction. In this sense, 40 articles from four scientifically recognized databases were analyzed. The predominance of decision tree-based machine learning techniques was evidenced, as well as the scarcity of real-world data sets for the advancement of cloud failure prediction research. Future research efforts can evaluate methods for aggregating data samples from works and tasks in the dataset, as well as techniques for class balancing in cloud failure prediction models. This systematic literature review presents the current state of research related to failure prediction in cloud systems. The observations and trends identified in this work may be useful for future researchers.

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). A computação em nuvem é definida como um 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 interação do provedor de serviços (Mell; Grance, 2011). Os serviços de computação em nuvem são fundamentais para a economia

moderna, além de facilitar a expansão de empresas em todo o mundo (Buyya *et al.*, 2018). Devido à escalabilidade e 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 na nuvem (Herbst *et al.*, 2018).

O fato desse modelo de computação possibilitar o uso de recursos sob demanda, onde os clientes são cobrados apenas pelos recursos utilizados, tem atraído cada vez mais clientes, aumentando a necessidade de manter a disponibilidade dos serviços fornecidos por esses sistemas (Alahmad; Daradkeh; Agarwal, 2021). A complexidade e natureza distribuída dos sistemas em nuvem contribuem para a ocorrência dos mais variados tipos de falhas, tais como, falha de software, falha de hardware, congestionamento de rede e falta de recursos (sobrecarga de servidor). As falhas são o principal motivo de indisponibilidade na computação em nuvem (Gao; Wang; Shen, 2020; Rawat; Bhadoria, 2021).

Muitos concordam que as falhas são inevitáveis em sistemas complexos (Gupta *et al.*, 2017; Mariani *et al.*, 2018; Monni; Pezzè; Prisco, 2019;). Quando não tratadas, as falhas podem resultar em perdas de bilhões de dólares, principalmente quando se tratar de 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 sentido, diversas pesquisas foram conduzidas com o objetivo de mitigar os efeitos das falhas nos sistemas de computação em nuvem. Em particular, as tecnologias de previsão de falhas têm 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; Pili, 2017). Os preditores de falhas modernos utilizam métricas coletadas dos nós de computação nos sistemas em nuvem para inferir a presença de falhas. 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).

Como diversos métodos foram propostos para prever a ocorrência de falhas nos sistemas de computação em nuvem (Gupta *et al.*, 2017), e esses métodos mudaram com o passar do tempo, o seguinte questionamento é levantado: quais os métodos atualmente utilizados para previsão de falhas nos sistemas em nuvem? Este trabalho tem como objetivo 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. A metodologia e processo de pesquisa são apresentados na Seção 3. Na Seção 4 é apresentada a análise dos resultados. Na Seção 5 são discutidos os resultados. 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, 2020): i) abordagem estatística; ii) abordagem de aprendizado de máquina; iii) abordagem de aprendizagem profunda.

O modelo Autorregressivo Integrado de Média Móvel (ARIMA) é uma das abordagens estatísticas mais utilizadas para a previsão de séries temporais (Hermias; Teknomo; Monje, 2017). Os modelos baseados em ARIMA podem ser utilizados para prever o comportamento futuro do sistema (Shao; Zhang, 2018). Por exemplo, Rawat *et al.* (2021) utilizaram ARIMA para prever falhas em máquinas virtuais em um *datacenter* de um sistema de computação em nuvem. O aprendizado de máquina é um ramo da inteligência artificial que permite que os 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 é uma subárea do aprendizado de máquina, entretanto seus algoritmos necessitam de maiores quantidades de dados e exigem maior desempenho computacional (Fahim; Sillitti, 2019; Su *et al.*, 2020).

3 Metodologia e processo de pesquisa

A revisão sistemática da literatura (RSL) busca caracterizar, sintetizar e comparar os últimos estudos referentes a um determinado tema de pesquisa, possibilitando a transferência de conhecimento na comunidade de pesquisa (Kashani; Mahdipour, 2022). Nesse sentido, este trabalho de revisão segue os procedimentos apresentados em Majid e Anuar (2022), definidos como: planejamento, execução e relatório. O planejamento da RSL inicia-se com a definição de um protocolo de revisão, ou seja, a definição clara das regras que serão utilizadas na condução da RSL, mitigando um possível viés de pesquisa em razão das expectativas dos pesquisadores (Dias; Correia; Malheiros, 2021). No protocolo de revisão são definidos os seguintes aspectos: objetivo da pesquisa, questões de pesquisa, palavras-chave que serão utilizadas nos mecanismos de busca, critérios de seleção das bases de busca, idioma das publicações, métodos de busca, bases de buscas, critérios de inclusão e exclusão, critérios de qualidade, bem como os dados que serão extraídos dos estudos com objetivo de responder as 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, a elevação do tráfego de rede e a maior utilização dos recursos, aumentando 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 perdas de vidas humanas (Irrera, 2015).

Assim, este trabalho de pesquisa busca revisar o estado da arte de forma sistemática, analisando os métodos mais recentes utilizados no processo de previsão de falhas em sistemas de computação em nuvem e, como resultado, responder as seguintes questões de pesquisa:

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

3.2 Critérios de inclusão e exclusão

Os critérios utilizados para inclusão e exclusão dos artigos são listados na Tabela 1.

Tabela 1 – Critérios de inclusão e exclusão dos artigos na RSL

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

Fonte: dados da pesquisa

3.3 Método de busca de trabalhos

Para manter a legitimidade deste trabalho, foram selecionadas quatro bases de busca cientificamente reconhecidas. Além da relevância das bases, também foram levadas em consideração a possibilidade de utilização com acesso acadêmico, possibilidade de busca por palavras-chave e exportação dos resultados. As bases escolhidas foram: ACM Digital Library, IEEE Xplore, Scopus e SpringerLink.

Nesta fase da RSL foram definidas as palavras-chave e respectivos sinônimos utilizados nas bases de busca. Após uma pesquisa exploratória e estudo dos trabalhos, verificou-se os termos recorrentes utilizados pela comunidade científica, possibilitando a extração das seguintes palavras-chave: *fault prediction*, *failure prediction*, *cloud computing*, *cloud service*, *cloud infrastructure*, *cloud platform* e *cloud datacenter*.

A *string* de busca adequada a cada mecanismo de busca foi definida após vários testes, pois uma *string* geral geraria milhares de resultados, enquanto uma *string* muito específica eliminaria diversos

resultados relevantes. Após realizar uma pesquisa prévia manual em cada base de busca, as peculiaridades de cada mecanismo de busca puderam ser verificadas, e com o uso dos operadores booleanos (*AND*, *OR*), foram definidas as *strings* de busca adequadas a cada mecanismo. Na Tabela 2, é possível visualizar as bases de busca utilizadas, bem como as respectivas *strings* de busca.

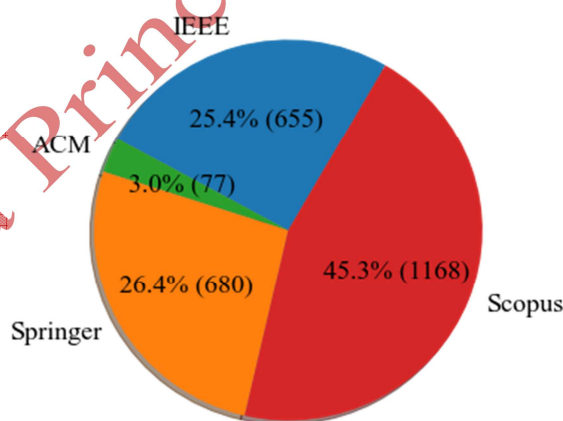
Tabela 2 – *Strings* por base de busca

Bases de busca	Strings de busca
ACM Digital Library	[[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"]]
IEEE Xplore	((("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")))
Scopus	(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"))
SpringerLink	((("fault prediction") OR ("failure prediction")) AND ("cloud computing") OR ("cloud service") OR ("cloud infrastructure") OR ("cloud platform") OR ("cloud datacenter"))

Fonte: dados da pesquisa

O processo de busca apresentou os seguintes critérios: i) não foram considerados trabalhos com ano de publicação anterior a 2017, pois poderiam apresentar soluções obsoletas; ii) também foram desconsiderados artigos de revisão, pois muitas vezes apresentam poucos detalhes a respeito dos métodos utilizados, descrevendo o estado da arte a partir de aspectos conceituais (Santos *et al.*, 2020). A Figura 1 ilustra os resultados da busca primária, onde um total de 2.580 artigos foram retornados, sendo 1.168 da base Scopus, 680 da Springer, 655 da IEEE e 77 da ACM.

Figura 1 – Resultado da busca primária

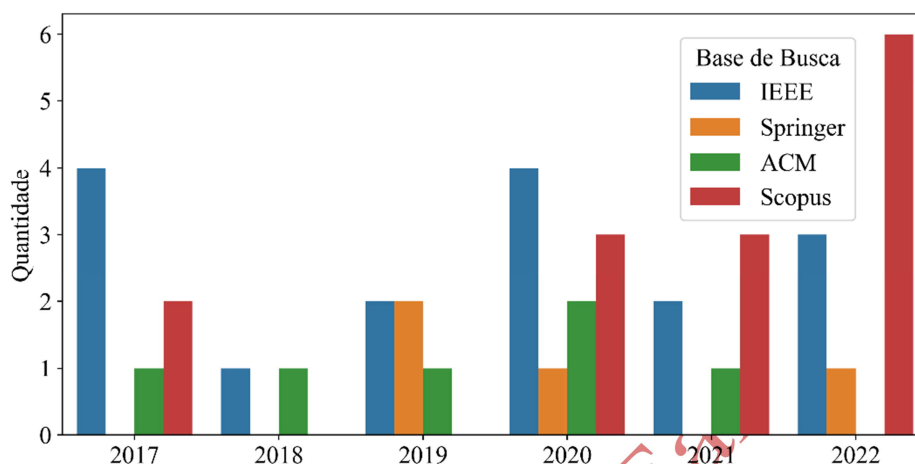


Fonte: dados da pesquisa

Após a realização da busca primária, foram aplicados os critérios de inclusão e exclusão. Inicialmente, 119 artigos que correspondem a 4,61% do número total de artigos selecionados na busca primária foram eliminados por estarem duplicados nas bases de busca. Em seguida, foi realizada a leitura dos títulos e resumos, introdução e conclusão e finalmente, quando necessária, a leitura das obras na íntegra para verificar a relevância e relação de cada trabalho com o escopo da pesquisa bem como a adequação com os critérios de inclusão e exclusão. É importante salientar que os três autores desta pesquisa participaram ativamente da revisão dos artigos com o objetivo de garantir a qualidade e imparcialidade dos resultados obtidos. Após as etapas de leitura necessárias, 2.421 artigos que correspondem a 93,8% do total de artigos selecionados na busca primária foram eliminados, resultando na inclusão de 40 artigos na revisão, o que corresponde a 1,55% do total de artigos.

Na Figura 2 é possível visualizar a distribuição dos trabalhos incluídos na revisão sistemática por base e ano de publicação. Embora a base de busca Scopus tenha retornado 45,3% (1.168) do total de artigos na busca primária, a base de busca IEEE foi a que contou com a maior quantidade de artigos incluídos, 16 artigos, seguida das bases Scopus, ACM e Springer com 14, 6 e 4 artigos, respectivamente. Também é possível observar uma maior concentração de publicações nos anos de 2020 e 2022.

Figura 2 – Artigos aceitos por base de busca



Fonte: dados da pesquisa

4 Análise de resultados

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

Tabela 3– Síntese dos artigos da RSL desta pesquisa

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%
Gao; Wang; Shen (2020)	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 e Elmaghaby (2017)	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%, Recall 79,34%, F1 Score 60,31%
Saxena; Singh (2022b)	SVM, Random Forest, Regressão Linear, NN	Google 2011 V.2	Acurácia 98,14%

Fonte: dados da pesquisa

4.1 Conjuntos de dados

O gerenciamento de falhas na computação em nuvem e o desenvolvimento de ambientes de nuvens confiáveis dependem de dados estatísticos de sistemas do mundo real. No entanto, um número escasso desses conjuntos de dados foi disponibilizado publicamente (Islam; Manivannan, 2017). A Tabela 4 apresenta os conjuntos de dados mais difundidos identificados na RSL, assim como os dados utilizados na abordagem experimental proposta em cada trabalho analisado. É possível observar a predominância de dados referentes à utilização de CPU, memória e disco nas abordagens experimentais. As variáveis estáticas correspondem ao provisionamento de recursos no agendador de tarefas do Google, ou seja, os recursos solicitados, mas não necessariamente utilizados pelas tarefas durante seu ciclo de vida. Como pode ser observado, alguns autores utilizaram todos os dados disponíveis no *dataset*.

Tabela 4 – Conjuntos de dados identificados na RSL em relação aos dados utilizados na abordagem experimental

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 (2020)	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 conjunto de dados.
Google 2019	Gollapalli <i>et al.</i> (2022)	Utilização de CPU, memória e disco.

Fonte: dados da pesquisa

As demais publicações utilizaram outros *datasets* pouco difundidos ou gerados pelos próprios autores e não disponibilizados publicamente.

O *dataset* Google 2011 V.2 é o conjunto de dados mais popular e conseqüentemente foi 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 no ano de 2011 com duração de 29 dias. O *dataset* inclui informações sobre as especificações de máquinas e o ciclo de vida das tarefas e trabalhos que foram solicitados pelos usuários (Alahmad; Daradkeh; Agarwal, 2021). As especificações de *hardware* não foram divulgadas pelo Google (Amvrosiadis *et al.*, 2018). Embora o *dataset* Google 2009 tenha sido utilizado com propósitos de comparação pelos autores Jassas e Mahmoud (2021), atualmente esse conjunto de dados é considerado obsoleto e sua utilização não é recomendada. O Google ainda disponibilizou uma terceira versão do *dataset* lançado no ano de 2020 (Wilkes, 2020), entretanto um único trabalho identificado nesta RSL utilizou esse conjunto de dados (Gollapalli *et al.*, 2022).

Com o objetivo de reduzir a dependência excessiva do *dataset* disponibilizado pelo Google, os autores Amvrosiadis *et al.* (2018) lançaram o repositório Atlas que, entre outros conjuntos de dados, é composto pelos *datasets* Trinity e Mustang, dois conjuntos de dados coletados do *Los Alamos National Laboratory*. O *dataset* Trinity, segundo mais predominante nesta RSL, consiste em um registro de eventos de supercomputador composto por 9.408 nós de computação idênticos, com um total de 301.056 núcleos Intel Xeon E5-2698v3 de 2.3 GHz e 1.2 PB de RAM. Essas especificações de *hardware* tornam este o maior cluster com rastreo disponibilizado publicamente por número de núcleos de CPU. Os dados abrangem um período de três meses, entre fevereiro e abril de 2017, e consistem em 25.237 trabalhos de 88 usuários. Por sua vez, o *dataset* Mustang abrange um período de 61 meses, de outubro de 2011 a novembro de 2016, o que o torna o *dataset* mais longo disponível publicamente. O 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. O *dataset* consiste em 2.1 milhões de trabalhos de 565 usuários (Amvrosiadis *et al.*, 2018; Jassas; Mahmoud, 2021).

Outro conjunto de dados identificado neste trabalho foi o *dataset* Alibaba 2018, que consiste em um registro de eventos de um cluster com 4.000 máquinas, 9.000 serviços e 4 milhões de trabalhos no período de 8 dias. O *dataset* apresenta detalhes do ciclo de vida dos trabalhos e tarefas. Por motivo de confidencialidade, as informações de *hardware* como memória e disco foram normalizadas nos *datasets* (Guo *et al.*, 2019). A Tabela 5 sumariza os dados disponíveis nos *datasets* identificados.

Tabela 5 – Dados disponíveis nos *datasets*

<i>Dataset</i>	Dados no <i>dataset</i>
Google 2011 V2	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
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, disco; prioridade da tarefa; taxa média e máxima de utilização de CPU e memória

Fonte: dados da pesquisa

Em oposição aos trabalhos anteriores, alguns autores utilizaram seus próprios conjuntos de dados obtidos por meio de rastreios não disponibilizados publicamente ou *datasets* sintéticos obtidos em ambientes de simulação. Por exemplo, Chen *et al.* (2019) utilizaram um *dataset* da nuvem Microsoft gerado pelos próprios autores. Devido a confidencialidade, não foram apresentadas informações relevantes a respeito do *dataset*, tampouco foi disponibilizado publicamente. Luo *et al.* (2021) utilizaram dois conjuntos de dados estatísticos de unidades de disco gerados do Microsoft Azure, não disponibilizados publicamente. Zhang *et al.* (2022) utilizaram 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 a partir de estratégias de simulação, impossibilitando a reprodução dos experimentos.

4.2 Métodos

Em relação às técnicas de previsão de falhas, a Tabela 6 exibe os métodos predominantemente utilizados. Como observado, os métodos mais utilizados foram *Random Forest*, LSTM e SVM.

Tabela 6 – Métodos predominantemente utilizados

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

Fonte: dados da pesquisa

Os demais algoritmos apresentados na Tabela 3 foram utilizados por um ou dois autores. Islam e Manivannan (2017) propuseram um modelo para prever falhas de trabalhos e tarefas de aplicações em nuvem com o algoritmo LSTM. Nos experimentos, os autores constataram alta correlação entre tempo e consumo de recursos com tarefas que falharam, além de apresentarem um método para estimar uma redução no consumo de recursos como CPU e memória em decorrência da previsão de falhas.

Embora os autores Islam e Manivannan (2017) tenham apontado o algoritmo SVM como inadequado para previsão de falhas com o *dataset* do Google 2011 V.2, Khalil, Sheta e Elmaghraby (2016) propuseram um método para prever falhas de máquinas por sobrecarga de trabalhos em um sistema em nuvem com uso do SVM. Para isso, o modelo foi treinado somente com as informações de CPU e memória. O uso do SVM foi possível graças ao método que os autores utilizaram para categorizar as falhas. Selecionaram as tarefas de maior prioridade e calcularam 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, ou seja, caso o consumo atual de CPU ou memória de uma tarefa seja maior que o consumo máximo calculado, pressupõe-se uma sobrecarga de uso de recursos e iminente falha.

Em contraste com o trabalho anterior, Liu, X. *et al.* (2020) alcançaram resultados mais promissores em previsão de falhas em nuvem com o algoritmo *Random Forest* em detrimento do algoritmo SVM. Os autores modelaram e implantaram a ferramenta COMPASS que tem o objetivo de prever falhas de *hardware* nos servidores da nuvem Baidu. A solução foi avaliada durante um período de 3 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, constatou-se 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 para prever falhas em SSDs em servidores na nuvem com uso dos algoritmos *Isolation Forest* e *Autoencoder*. Na abordagem, foi utilizado um *dataset* com dados de mais de 30.000 SSDs coletados por um período de seis anos da nuvem do Google, e utilizou somente a classe majoritária (sem falhas) no processo de treinamento. O método proposto possui as vantagens de não

sofrer com *overfitting* (sobreajuste aos dados de treinamento), consegue prever falhas inéditas não vistas no conjunto de teste, além de não ser afetado pelo desbalanceamento de classes, comum em modelos de classificação.

4.3 Métricas

Vários aspectos são levados em consideração na avaliação de modelos preditivos, como desempenho, interpretabilidade e recursos computacionais utilizados. As métricas relacionadas a desempenho avaliam quão bem o modelo produziu o resultado esperado. As métricas comumente utilizadas em modelos de classificação são as proporções de instâncias previstas corretas e incorretamente, como acurácia, *recall*, precisão, e *F1 score*. Por sua vez, os modelos de regressão são avaliados medindo os erros estimados, como *root mean square error* (RMSE) ou o *mean absolute percentage error* (MAPE) (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* é a proporção de exemplos positivos corretamente identificados pelo modelo em relação ao total de exemplos positivos (Jauk; Yang; Schulz, 2019). O *F1 score* consiste na média harmônica da precisão e *recall* (Notaro; Cardoso; Gerndt, 2021).

Além das métricas geralmente utilizadas na avaliação de modelos de previsão de falhas, alguns autores apresentaram outras métricas significativas na avaliação da abordagem proposta. Devido à particularidade do método utilizado por Chakrabortii e Litz (2020), onde somente a classe majoritária foi utilizada no processo de treinamento do modelo, os autores utilizaram a *Area under curve* (AUC) para avaliar o modelo, pois as métricas tradicionais poderiam apresentar resultados distorcidos, uma vez que havia a possibilidade de *overfitting* para classe majoritária (prever todas as amostras para classe majoritária). Os autores também apresentaram as métricas referentes ao tempo de treinamento para cada algoritmo avaliado. De forma semelhante, mas com 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 obteve AUC de 92% para previsões entre 2 e 48 horas de antecedência.

Além da precisão do modelo proposto, Gao, Wang e Shen (2020) levaram em consideração a sobrecarga de tempo de treinamento e teste dos modelos avaliados, buscando equilíbrio entre ambas as métricas com objetivo de manter a qualidade do serviço na nuvem. Na 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, Das, Mueller e Rountree (2020) foram os que apresentaram 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 3 minutos. 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 identificado 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 apresentaram os resultados obtidos na avaliação experimental. De forma semelhante, Adamu *et al.* (2017) não apresentaram a métrica utilizada na avaliação do método proposto, entretanto apresentaram a quantidade de falhas previstas no *dataset*.

5 Discussão

Nesta seção, são discutidos os principais resultados obtidos com o objetivo de responder às questões de pesquisa. Também são 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 das classes. Ou seja, as classes que se deseja prever não têm 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 normalmente tratado com estratégias de amostragem, por exemplo, *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 *Synthetic Minority Oversampling Technique* (SMOTE) que consiste em uma técnica de amostragem da classe minoritária com o objetivo de gerar amostras sintéticas, reduzindo ou eliminando o desequilíbrio entre as classes. Embora Lin *et al.* (2018) também tenham utilizado SMOTE, os autores relataram resultados pouco satisfatórios, com altas taxas de falsos positivos.

Li *et al.* (2020) utilizaram uma técnica de *oversampling* aprimorada pelos autores para selecionar mais amostras positivas, capturando mais 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), utilizaram *oversampling* em 2 vezes nos dados de falhas e *undersampling* em 60 vezes nos dados normais.

Por outro lado, Luo *et al.* (2021) propuseram um método para lidar com o desequilíbrio de classes denominado *Temporal Progressive Sampling* (TPS). O método proposto gera novas amostras da classe minoritária em função do tempo e é definido pelos autores como um método para aprimoramento de dados. Os demais trabalhos incluídos nesta revisão sistemática não utilizaram nenhuma técnica para tratar o desequilíbrio de classes.

5.2 Sobre os métodos

Em conformidade com a pesquisa recente realizada por Bhanage, Pawar e Kotecha (2021), ficou evidenciada neste trabalho a predominância de técnicas de aprendizagem de máquina nas abordagens de previsão de falhas. Especificamente, algoritmos baseados em árvore como *Random Forest* e *Decision Tree*. Possivelmente, essa tendência pode ser explicada pelo fato dos modelos baseados em *Random Forest* apresentarem excelentes resultados, superando técnicas como SVM e redes neurais (Jauk; Yang; Schulz, 2019). Em concordância com Bhanage, Pawar e Kotecha (2021), constatou-se nesta revisão da literatura um aumento no uso de técnicas de aprendizagem profunda nos últimos anos. Em particular, o algoritmo LSTM tem sido amplamente utilizado.

Em relação à natureza das falhas as quais se pretende prever nos sistemas em nuvem, observou-se a predominância de métodos voltados à previsão de falhas de trabalhos e tarefas (14 artigos), seguido 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).

É importante notar que, em relação aos métodos que utilizaram o conjunto de dados Google 2011 V2, alguns autores consideraram mais de um estado final dos trabalhos e tarefas como falhas. Por exemplo, Bhattacharyya *et al.* (2017) e Shetty, Sajjan e G. (2019) classificaram as tarefas encerradas como falhas. Por outro lado, Islam e Manivannan (2017), Liu *et al.* (2017) e Liu C. *et al.* (2020) classificaram as tarefas removidas e encerradas como falhas. Além das tarefas removidas e encerradas, Asmawi, Ismail e Shen (2022) classificaram as tarefas e trabalhos perdidos como falhas. A classificação desses estados de trabalhos e tarefas como falhas deve ser cuidadosamente avaliada, pois pode impactar nos resultados dos métodos propostos, uma vez que adicionam dados aos modelos.

Um outro aspecto importante relacionado aos métodos que utilizaram os dados do Google é a quantidade de dados utilizada nos modelos de previsão de falhas. Embora o conjunto de dados Google 2011 V2 consista em um registro de 29 dias, alguns autores utilizaram uma fração desses dados, possivelmente devido às limitações de recursos de computação para realização dos experimentos. Liu *et al.* (2017) utilizaram as doze primeiras horas do conjunto de dados, enquanto Liu C. *et al.* (2020) utilizaram os três primeiros dias. Por sua vez, Jassas e Mahmoud (2020) se limitaram à utilização dos primeiros sete dias. Por outro lado, Asmawi, Ismail e Shen (2022) utilizaram os primeiros catorze dias do conjunto de dados. Como uma fração dos dados pode não representar o comportamento da totalidade dos dados, os resultados apresentados podem ser distorcidos.

Ainda em relação aos métodos que utilizaram o *dataset* Google 2011 V2, alguns autores relataram alta incidência de falhas nos dias iniciais do conjunto de dados. Por exemplo, Asmawi,

Ismail e Shen (2022) relataram uma proporção anormal de tarefas que falharam nos catorze dias iniciais do *dataset*. Os autores ainda relataram a existência de três tarefas com falha a cada tarefa finalizada. De forma semelhante, Jassas e Mahmoud (2020) indicaram a alta incidência de falhas nos dias iniciais do *dataset* Google 2011 V2. Como os autores anteriormente citados não trataram a alta incidência de falhas nos dias iniciais do *dataset*, os resultados apresentados podem ter sido superestimados em função da predominância de tarefas com falhas nos dados utilizados.

5.3 Sobre os conjuntos de dados

Embora tenha passado mais de uma década desde o lançamento do *dataset* Google 2011 V2, neste artigo ficou evidenciada a preferência dos pesquisadores por esse conjunto de dados em detrimento dos demais identificados neste trabalho e disponibilizados publicamente. Essa preferência reforça a dependência excessiva apontada por Amvrosiadis *et al.* (2018). Como alternativa ao *dataset* de 2011 V2, o *dataset* Google 2019 V3, lançado no ano de 2020, pode ser utilizado nas pesquisas relacionadas à previsão de falhas em nuvem. Por se tratar de um registro de eventos mais recente, pode refletir melhor a realidade dos sistemas atualmente utilizados.

Em relação a disposição dos dados, ou seja, como os dados estão formatados, foi observado que alguns conjuntos de dados disponibilizados publicamente, como Google 2011 V2 e Google 2019 V3 possuem dados ofuscados ou normalizados. A ofuscação tem o objetivo de manter a confidencialidade dos dados considerados sigilosos pela organização. Por sua vez, a normalização é uma técnica comumente utilizada 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 possuem os dados referentes à utilização de memória RAM e CPU normalizados para o valor máximo de 1. Essa característica impossibilita a identificação dos valores reais disponíveis e utilizados nas operações do sistema. Por exemplo, não é possível identificar a quantidade de memória e capacidade de processamento disponíveis nos servidores.

5.4 Sobre as métricas

Em relação às métricas utilizadas na avaliação dos métodos propostos, observou-se que os autores Saxena e Singh (2022a) e Saxena e Singh (2022b) utilizaram exclusivamente a acurácia. Entretanto, o uso exclusivo dessa métrica não é adequado para avaliação de modelos com conjuntos de dados desbalanceados. Por exemplo, 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. Isso ocorre porque, nos conjuntos de dados, as falhas são exceções (Jauk; Yang; Schulz, 2019). Notaro, Cardoso e Gerndt (2021) rotulam a acurácia como enganosa para avaliação da qualidade de previsões, pois na maioria das vezes essa métrica explora a assimetria dos dados. Embora a AUC seja apontada como apropriada para avaliação de 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 bastante atenção nos últimos anos. Tendo em vista que diversos métodos foram propostos para a previsão de falhas nesses sistemas, este trabalho de revisão sistemática da literatura examina as características desses métodos, incluindo as técnicas utilizadas, os conjuntos de dados e as métricas empregadas para avaliar os métodos propostos. Nesse sentido, 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 o objetivo de previsão do tempo para a ocorrência das falhas, assim como a escassez de abordagens com o *dataset* Google 2019 V3. Ficou evidenciada a necessidade de disponibilização de conjuntos de dados do mundo real, pois atualmente um número limitado de *datasets* foi disponibilizado publicamente.

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

Esta revisão sistemática da literatura apresenta o estado atual das pesquisas relacionadas à previsão de falhas nos sistemas em nuvem. As observações e tendências identificadas neste trabalho podem ser úteis para futuros pesquisadores. A metodologia utilizada neste trabalho permite sua reprodução pela comunidade acadêmica.

Financiamento

Esta pesquisa recebeu recursos financeiros da Universidade Federal Rural de Pernambuco (UFRPE).

Conflito de interesses

Os autores declaram não haver conflito de interesses.

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**, 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*: 2017 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>.

ADEGBOYEGA, A. Time-series models for cloud workload prediction: a comparison. *In*: 2017 IFIP/IEEE SYMPOSIUM ON INTEGRATED NETWORK AND SERVICE MANAGEMENT (IM), 2017, Lisbon. **Proceedings [...]**. Lisbon: IEEE, 2017. DOI: <https://doi.org/10.23919/INM.2017.7987292>.

ALAHMAD, Y.; DARADKEH, T.; AGARWAL, A. Proactive failure-aware task scheduling framework for cloud computing. **IEEE Access**, v. 9, 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>. Acessado em: 10 de julho de 2023.

ASMAWI, T. N. T.; ISMAIL, A.; SHEN, J. Cloud failure prediction based on traditional machine learning and deep learning. **Journal of Cloud Computing**, 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, 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, p. 161-169, 2017. Disponível em: <https://dl.acm.org/doi/10.5555/3172795.3172814>. Acessado em: 08 de maio de 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>.

BUYA, R.; SRIRAMA, S. N.; CASALE, G.; CALHEIROS, R.; SIMMHAN, Y.; VARGHESE, B.; GELENBE, E.; JAVADI, B.; VAQUERO, L. M.; NETTO, M. A. S.; TOOSI, A. N.; RODRIGUEZ, M. A.; LLORENTE, I. M.; VIMERCATI, S. D. C. D.; SAMARATI, P.; MILOJICIC, D.; VARELA, C.; BAHSOON, R.; ASSUNCAO, M. D. 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, p. 1-38, 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, p. 120-133, 2020. 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, p. 2659-2665, 2019. 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: 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, p. 1-38, 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, 2020. 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.; ALDAWOUD, 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. Disponível em: <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: 2017 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.; EYK, E. V.; 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. 4, n. 4, p. 1-36, 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) – Universidade de Coimbra, Coimbra, 2015. Disponível em: <https://estudogeral.sib.uc.pt/handle/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. Evaluation of a failure prediction model for large scale cloud applications. In: GOUTTE, C.; ZHU, X. (eds.). **Advances in Artificial Intelligence**. Canadian AI 2020. Lecture Notes in Computer Science, v. 12109. Cham: Springer, 2020. DOI: https://doi.org/10.1007/978-3-030-47358-7_32.

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, 2025, 2022. DOI: <https://doi.org/10.3390/s22052035>.

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. DOI: <https://doi.org/10.1145/3295500.3356185>.

KABIR, H. 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, p. 1-30, 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, p. 1-24, 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: 2022 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: 2018 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, p. 480-490, 2018. 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, n. 2001-2003, 2020a. 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. (eds.). Algorithms and Architectures for Parallel Processing. ICA3PP 2019. Lecture Notes in Computer Science (), v. 11944. Cham: Springer, 2020b. 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: 2020 IEEE INTERSOCIETY CONFERENCE ON THERMAL AND THERMOMECHANICAL PHENOMENA IN ELECTRONIC SYSTEMS (ITherm), 19., 2020c, 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: WEB CONFERENCE 2021 (WWW' 21)*, 2021, Ljubljana. **Proceedings [...]**. Ljubljana: ACM, p. 1181-1191, 2021. 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: 2020 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: 2018 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.; 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: 2020 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION SYSTEMS (ICICS)*, 11., 2020, Irbid. **Proceedings [...]**. Irbid: IEEE, 2020. DOI: <https://doi.org/10.1109/ICICS49469.2020.239556>.

MONNI, C.; PEZZÈ, M.; PRISCO, G. An RBM anomaly detector for the cloud. *In: 2019 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: 2021 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, p. 1-45, 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: 2021 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: 2020 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/s11227-021-04235-4>.

SAXENA, D.; SINGH, A. K. VM failure prediction based intelligent resource management model for cloud environments. *In: 2022 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: 2018 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.; G., S. Task resource usage analysis and failure prediction in cloud. *In: 2019 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. (eds.). Advances in Artificial Intelligence, Software and Systems Engineering.* AHFE 2019. Advances in Intelligent Systems and Computing, v. 965. Cham: Springer, 2020. 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**. 2020. Google research blog. 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*: 2022 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>.

Revista Principia - Early View