UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JANETE PEREIRA DO AMARAL

UMA ARQUITETURA AUTONÔMICA PARA ALOCAÇÃO PREDITIVA DE RECURSOS NO GERENCIAMENTO DE CAPACIDADE EM NUVEM UTILIZANDO REDES NEURAIS

FORTALEZA
2023

JANETE PEREIRA DO AMARAL

UMA ARQUITETURA AUTONÔMICA PARA ALOCAÇÃO PREDITIVA DE RECURSOS NO GERENCIAMENTO DE CAPACIDADE EM NUVEM UTILIZANDO REDES NEURAIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Doutor em Ciência da Computação. Área de concentração: Redes de Computadores.

Orientador: Prof. Dr. José Neuman de Souza.
Coorientador: Prof. Dr. Alberto Sampaio Lima.

FORTALEZA

2023

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

A514a Amaral, Janete Pereira do.
Uma arquitetura autonômica para alocação preditiva de recursos no gerenciamento de capacidade em nuvem utilizando redes neurais / Janete Pereira do Amaral. – 2023.
181 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação , Fortaleza, 2023.
Orientação: Profa. Dra. Jose Neuman de Souza.
Coorientação: Profa. Dra. Alberto Sampaio Lima.

1. Computação em nuvem. 2. Computação autonômica em nuvem. 3. Stacked Long Short-Term Memory. 4. Gerenciamemto de capacidade em nuvem. 5. Bussines-Driven IT Management. I. Título.
CDD 005

JANETE PEREIRA DO AMARAL

UMA ARQUITETURA AUTONÔMICA PARA ALOCAÇÃO PREDITIVA DE RECURSOS NO GERENCIAMENTO DE CAPACIDADE EM NUVEM UTILIZANDO REDES NEURAIS

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Doutor em Ciência da Computação. Área de concentração: Redes de Computadores.

BANCA EXAMINADORA

Aprovada em: 31 / 01 / 2023.

Prof. José Neuman de Souza, DSc. (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Alberto Sampaio Lima, DSc. (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Lincoln Souza Rocha, DSc.
Universidade Federal do Ceará (UFC)

Prof. José Maria da Silva Monteiro Filho, DSc.
Universidade Federal do Ceará (UFC)

Prof. Joaquim Celestino Junior DSc.
Universidade Estadual do Ceará (UECE)

Prof. José Antão Beltrão Moura, DSc.
Universidade Federal de Campina Grande (UFCG)

À minha mãe, dedico o meu esforço, lutas e glórias. Ela sempre será meu exemplo de luta pela vida.

AGRADECIMENTOS

Agradeço a **Deus** por estar presente em minha vida e ao **Santo Expedito**, a quem sempre pedi intercessão nas horas difíceis.

Ao meu coorientador **Dr. Alberto Sampaio Lima** pela dedicação, colaboração, estímulo e, acima de tudo, por me apoiar nos momentos de dificuldades.

Ao **Dr. José Neuman de Souza**, pela oportunidade de tê-lo como meu orientador **e** pela oportunidade de fazer este doutorado.

Aos professores **Dr. José Maria Monteiro, Dr. Joaquim Celestino Jr**., **Dr. Lincoln Rocha e Dr. José Antão Beltrão Moura**, participantes das bancas examinadoras. Minha eterna gratidão pelo apoio e pelas valiosas colaborações e sugestões.

À **Joana**, meu amor incondicional e minha vida por ti. Que Deus te abençoe e ilumine.

À **minha família** que, embora não entendendo muito bem este meu sonho, numa hora tão crucial para todos nós, assumiram as minhas ausências.

Aos amigos de todas as horas, **Letícia Adriana**, **Aminadabe Sousa** e **Patrícia Vasconcelos** que sempre estiveram do meu lado quando eu precisava chorar.

Aos colegas do grupo de pesquisa, **Germano Fenner** e **Maristela Ribas** pelas reflexões, críticas e sugestões recebidas.

Ao MDCC-Mestrado e Doutorado em Ciência da Computação/Departamento de Computação da Universidade Federal do Ceará pelo apoio fornecido, especialmente pelo suporte recebido pelo Jonatas Martins.

Aos amigos, agregados a esta ciranda de apoio para a realização deste sonho.

RESUMO

A computação em nuvem tem sido apontada como uma solução para o uso racional de recursos da Tecnologia da Informação. Os provedores de serviços em nuvem oferecem ambientes compartilhados que podem ser dimensionados para atender aos requisitos flutuantes de seus clientes. O desafio imposto é a utilização de mecanismos capazes de otimizar o uso dos recursos e, simultaneamente, garantir que o desempenho desses serviços continue atendendo às métricas *Quality of Experience* (QoE), *Quality of Service* (QoS), bem como aos *Service Level Indicators* (SLI) e aos respectivos *Service Level Agreements* (SLA) estabelecidos. Os provedores necessitam oferecer mecanismos autonômicos para promover a escalabilidade dos recursos em tempo hábil, ao tempo em que os clientes precisam confiar no desempenho e nos custos envolvidos nas negociações. No processo de **Gerenciamento de Capacidade em Nuvem,** diversas abordagens **preditivas de escalonamento de recursos** já foram propostas para superar as limitações das abordagens reativas convencionais. Entretanto, tais abordagens ainda não demostraram resultados satisfatórios, em termos de custo, desempenho e autonomia. Esta pesquisa propõe uma **Arquitetura Autonômica para Alocação Preditiva de Recursos no Gerenciamento de Capacidade em Nuvem utilizando Redes Neurais** que combina características reativas e preditivas para o escalonamento de recursos. Para suportar o provisionamento preditivo de recursos, foi utilizado *Recurrent Neural Networks (RNNs)* na arquitetura *Stacked Long Short-Term Memory*, buscando suplantar os resultados já alcançados. Na abordagem autonômica, adotou-se o modelo MAPE-K, recorrendo aos princípios da *Autonomic Cloud Computing* (ACC). Visando demonstrar a viabilidade da proposta foi elaborado um estudo de caso utilizando *traces* experimentais. Na avaliação da acurácia da predição utilizou-se um comparativo entre a rede LSTM clássica e a rede *Stacked LSTM*. Para análise operacional foi proposta uma arquitetura autonômica utilizando o modelo preditivo. Os resultados obtidos demonstraram a viabilidade da proposta, trazendo como benefício a utilização das *Stacked LSTMs* na predição do **provisionamento de recursos em nuvem.** Como trabalhos futuros, pretende-se implementar a arquitetura em uma ferramenta operacional, em código livre, para apoiar provedores de serviços de pequeno e médio porte e permitir o planejamento de capacidade no processo de migração para nuvem.

Palavras-chave: computação em nuvem; stacked long short-term memory; computação autonômica em nuvem; gerenciamento de capacidade em nuvem; business-driven information technology management.

ABSTRACT

Cloud computing has been identified as a solution for the rational use of Information Technology resources. Cloud service providers offer shared environments that can scale to meet their customers' fluctuating requirements. The challenge imposed is the use of mechanisms capable of optimizing the use of resources and, simultaneously, ensuring that the performance of these services continues to meet the Quality of Experience (QoE), Quality of Service (QoS) metrics, as well as the Service Level Indicators (SLI) and the respective Service Level Agreements (SLA) established. Providers need to offer autonomic mechanisms to promote resource scalability promptly, while customers need to trust the performance and costs involved in negotiations. In the Cloud Resource Management process, several predictive resource scheduling approaches have already been proposed to overcome the limitations of conventional reactive techniques. However, such methods have not demonstrated satisfactory cost, performance, and autonomy results. This research proposes an Autonomic Cloud Resource Management Model that combines reactive and predictive features for resource scheduling. Recurrent Neural Networks (RNNs) were used in the Stacked Long Short-Term Memory architecture to support the predictive provisioning of resources, seeking to overcome the results already achieved. The MAPE-K model was adopted in the autonomic approach, using the principles of Autonomic Cloud Computing (ACC). A case study was elaborated using experimental traces to demonstrate the proposal's viability. In evaluating the model's accuracy, a comparison between the classic LSTM network and the Stacked LSTM network configurations was used. A prototype was implemented using components of cloud infrastructure simulators to analyze the operational viability. The obtained results demonstrated the feasibility of the proposal, bringing as a benefit the use of Stacked LSTMs in predicting the provisioning of cloud resources. As future work, it is intended to evolve the prototype into an operational tool, in open source, to support small and medium-sized service providers and allow capacity planning in the cloud migration process.

Keywords: cloud computing; stacked long short-term memory; cloud autonomic computing; capacity planning; business-driven information technology management.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AI	*Artificial Intelligence* - Inteligência Artificial
ANN	*Artificial neural network*
API	*Application Program Interface* - Interface de Programação de Aplicação
ARIMA	*Autoregressive integrated moving average*
ARMA	*Autoregressive moving average*
AWS	*Amazon Web Services*
BPNN	*Backpropagation neural network*
CI/CD	*Continuous Integration / Continuous Delivery* - Método para entrega frequente de aplicações aos clientes.
CEO	*Chief Executive Officer*. Diretor executivo. Principal cargo da empresa. Atua na comunicação entre o operacional e o conselho de administração.
CFO	*Chief Financial Officer*. Diretor Financeiro. Responsável pelo planejamento econômico e financeiro da empresa.
CIO	*Chief Information Officer*. Gerente, Superintendente, Diretor ou vice-presidente da Tecnologia da Informação de uma empresa.
CMMI	*Capability Maturity Model Integration* - Modelo Integrado de Maturidade e Capacidade.
CNN	*Convolutional neural network*
CPU	*Central processing unit*
CSCC	*Cloud Standard Customer Council*
CSMIC	*Cloud Services Measurement Initiative Consortium*
CSP	*Cloud Service Provider*
DC	*Datacenter*
DAG	*Directed acyclic graph*
DevOps	União de pessoas, processos e tecnologias para fornecer continuamente valor aos clientes.
GA	*Genetic Algorithm*
GCP	*Google Cloud Platform*
GPU	*Graphics Processing Unit* – Unidade de Processamento Gráfico
HMM	*Hidden Markov Modeling*
IaaS	*Infrastructure as a Service* - Infraestrutura como Serviço
IDE	*Integrated Development Environment* - Ambiente de Desenvolvimento

	Integrado
IDS	*Intrusion Detection System* - Sistemas de Detecção de Intrusão
IPS	*Intrusion Prevention System* - Sistemas de Prevenção de Intrusão
ITIL	*Information Technology Infrastructure Library*
KNN	*K-nearest neighbors*
KPIs	*Key Performance Indicators*
KQIs	*Key Quality Indicators*
LR	*Linear Regression*
LSTM	*Long short-term memory*
MAD	*Mean absolute deviation*
MAE	*Mean Absolute Error*
MAPE	*Mean absolute percentage error*
mLSTM	*Multiplicative Long short-term memory*
MSE	*Mean Squared Error*
NIST	*National Institute of Standards and Technology*
PaaS	*Platform as a Service* - Plataforma como Serviço
PM	*Physical machine*
PSO	*Particle swarm optimization*
QA	*Quality Assurance* - Garantia de qualidade
QoS	*Quality of Service* - Qualidade de serviço
RMSE	*Mean Squared Error*
RNN	*Recurrent neural network*
ROI	*Return of investment* - Retorno do investimento
SaaS	*Software as a Service* - Software como Serviço
SAN	*Storage Area Network* - Rede privativa de armazenamento conectada para manter servidores e *storages* via LAN ou WAN.
SLA	*Service-Level Agreement* - Acordo de Nível de Serviço (ANS).
SLAV	*Service-Level Agreement violation*
SOA	*Service-Oriented Architecture* - Arquitetura Orientada a Serviços
SoS	*System of Systems* - Sistema para criar sistemas com capacidade maior que a soma dos sistemas constituintes
SVM	*Support Vector Machine*
SVR	*Support Vector Regression*

TaaS	*Test as a Service* - Teste como Serviço.
TCO	*Total Cost of Ownership* – Custo Total de Propriedade.
VM	*Virtual machine.*
RTO	*Recovery Time Objective* (Objetivo do Tempo de Recuperação) - mensura o tempo máximo em que um sistema poderá ficar indisponível após uma falha.
RPO	*Recovery Point Objective* (Ponto Objetivo de Recuperação) - representa a quantidade de recursos mínimos a serem recuperados em caso de falhas ou perda de dados.

SUMÁRIO

1 INTRODUÇÃO

A **Computação em Nuvem,** por sua flexibilidade, elasticidade, baixo custo e uso ilimitado de recursos, tornou-se uma infraestrutura eficiente e valiosa para as operações de muitas organizações. Tem sido responsável por uma revolução no uso da Tecnologia da Informação (TI). Os impactos dessa transformação são percebidos, enquanto os serviços são ofertados, com maior segurança e a custos mais atraentes e competitivos. Mesmo sendo um paradigma em evolução, pode ser considerado uma das inovações mais disruptivas da tecnologia nos últimos anos. Esse modelo computacional ganhou popularidade por empregar uma infraestrutura dinamicamente escalável e pronta para uso, bem como por minimizar os Custos de Propriedade (TCO-*Total Cost of Ownership*) dos recursos de TI.

No eficiente manuseio dos recursos em nuvem, os *Cloud Service Providers* (CSPs) necessitam de métodos dinâmicos e inteligentes para o gerenciamento da infraestrutura. Uma provisão proativa necessita prevê as flutuações de demanda, de forma que os CSPs possam adequar o fornecimento de recursos, antes da intermitência da carga de trabalho. Da mesma forma, na redução desta demanda, os recursos alocados devem ser liberados e utilizados para apoiar outros serviços.

A Computação em Nuvem proporciona mudanças no modo como as empresas oferecem seus produtos e serviços, atingem novos clientes, acompanham os clientes existentes, praticam marketing e gerenciam os recursos físicos e financeiros. Essas mudanças modificam as relações entre as empresas, permitindo que organizações de pequeno e médio porte, tenham acesso a recursos que antes só estavam disponíveis para grandes empresas. Ao alterar as condições de competitividade no mercado, criam oportunidades, sem exigir a antecipação de investimentos.

Mesmo com atributos positivos, essa tecnologia apresenta pontos de preocupação, como o dimensionamento eficiente de recursos, consumo de energia, localização de máquinas virtuais, segurança, privacidade, utilização de recursos etc.

Segundo Fenner *et al*. (2021), em um ambiente de nuvem, a capacidade de processamento e armazenamento pode ser aumentada em pouco tempo para o atendimento de uma demanda que irá durar apenas algumas horas, sem a necessidade de se investir na aquisição de equipamentos. Ribas (2015) afirma que cabe ao provedor de recursos em nuvem disponibilizar a capacidade necessária para conseguir suportar a demanda de todos os seus clientes. Fenner *et al*. (2021) afirmam que no contexto do provedor de serviços em nuvem, diversas variáveis devem ser consideradas: alinhamento estratégico entre TI e negócio,

gerenciamento da demanda, gerenciamento da capacidade e lucro, entre outras. O dimensionamento da capacidade real e o entendimento do processo de demanda e utilização dos serviços da nuvem impactam diretamente nos resultados do negócio.

A computação em nuvem tem se tornado a palavra da moda na indústria de tecnologia da informação (TI) (Wang *et al*., 2010; Vouk, 2008) visando proporcionar serviços sob demanda com pagamento baseado no uso (*pay-per-use*). Nos provedores de serviços, cada máquina física possui as mesmas configurações de software, mas pode ter variação na capacidade de hardware em termos de CPU, memória e armazenamento em disco. Dentro de cada máquina física existe um número variável de máquinas virtuais ou nós virtuais em execução, conforme a capacidade do hardware disponível na máquina física. Os dados são persistidos, geralmente, em sistemas de armazenamento distribuídos. Conforme o tipo de segmento (nível de segurança, número de usuários, tipo da tecnologia, complexidade do segmento do negócio), pode existir a necessidade de se construir infraestruturas de TI complexas, onde os usuários necessitam realizar instalação, configuração e atualização de sistemas de hardware e software.

Dessa forma, os usuários passam a acessar um conjunto de serviços sob demanda e independentes de localização. O *hardware* e o *software* em uma nuvem são configurados e orquestrados automaticamente, além de suas modificações serem apresentadas de forma transparente para os usuários, que possuem perfis diferentes e podem personalizar os seus ambientes computacionais. Os recursos computacionais do provedor são organizados em um *pool* para servir a múltiplos usuários usando um modelo multi-inquilino *(Multitenant)*, com diferentes recursos físicos e virtuais, atribuídos dinamicamente e ajustados conforme a demanda dos usuários.

1.1 Motivação e descrição do problema

A Computação em Nuvem apresenta um novo cenário, onde a TI é fundamental. Seu impacto muda a qualificação profissional, traduzindo-se em serviços adequados, seguros e econômicos. Essa tecnologia não é apenas escalável, mas oferece também uma sistemática de pagamento pelo uso, o que a torna popular entre as pequenas e médias organizações.

1.1.1 Problema de negócio (Business Problem)

Um dimensionamento inadequado de recursos por parte dos provedores de serviços em nuvem pode ocasionar problemas que impactam nos resultados do negócio, entre

os quais pode-se citar:

a) alocação inadequada de um determinado recurso desnecessário para o cliente (a maior ou a menor);
b) cobrança inadequada pelo uso repassada aos clientes, por conta de recursos mal dimensionados ou desnecessários;
c) baixa qualidade dos serviços por conta de recursos insuficientes;
d) violações de acordos de nível de serviços (Service Level Agreements – SLAs).

Geralmente, os gestores de provedores de serviços podem planejar e executar em seus processos de gerenciamento de recursos/capacidade, três respostas de gestão em relação ao processo de gerenciamento de recursos a partir da demanda por seus serviços: **reativa, proativa ou sem ação**.

Na resposta **reativa** as ações ocorrem a partir da ocorrência de eventos provisionados ou previstos pelo provedor, sendo a decisão tomada depois que o evento acontece. Na resposta **proativa**, o provedor busca a tomada de decisão a partir de previsões realizadas, buscando evitar a ocorrência de eventos que possam impactar nos resultados do negócio. Já a resposta denominada **sem ação**, consiste no fato de simplesmente se aceitar os eventos da forma como acontecem, dependendo do contexto para a tomada de decisão (pós-evento). A resposta **proativa** para o gerenciamento de recursos, tem se tornado uma necessidade real no mercado, em função da alta competitividade e da busca da qualidade de serviços por parte dos provedores de serviços em nuvem. Foi identificado durante essa pesquisa que, apesar de grandes provedores de serviços em nuvem, tais como *Google* e *Amazon,* afirmarem em suas postagens que realizam essa proatividade, a literatura que versa sobre o gerenciamento de capacidade/recursos em nuvem ainda apresenta lacunas de pesquisa relacionadas à efetividade dessas ações proativas, e pelo fato desse processo ser de extrema relevância para o negócio de provedores de serviços em nuvem, os grandes provedores, que possuem a maior quantidade de ferramentas automatizadas para a gestão de seus recursos, mantém sigilo sobre a especificação técnica de suas soluções voltadas para essa área. Essa realidade de mercado e da área acadêmica foi um dos fatos motivadores para a realização da presente pesquisa.

1.1.2 Problema técnico (Technical problem)

Por conta da necessidade de atender às demandas por serviços com preços competitivos, os provedores de serviços em nuvem de todos os portes necessitam de uma

arquitetura de gerenciamento de recursos autonômica e proativa que possa otimizar o processo de alocação da forma mais efetiva e eficaz.

Um ponto crítico da entrega de serviços de recursos dimensionados com base na demanda de uso é se ter um escalonamento efetivo de recursos que viabilize a redução de tempo e custo na execução dos serviços. Uma questão a ser considerada é que nos serviços em nuvem existem, primariamente, dois intervenientes: **provedores e consumidores de serviços** (Liu *et al*., 2011). Geralmente essas partes possuem interesses conflitantes. De um lado, os CSPs mantêm recursos de computação em seus *Data Centers* (DC) e alugam estes recursos para organizações usuárias. No outro lado, existem organizações (gestores de serviço) que possuem aplicativos com cargas flutuantes e que alugam os recursos desses CSPs para executarem seus aplicativos. A incerteza na utilização de recursos traz desafios que não podem ser satisfeitos com políticas tradicionais de alocação de recursos.

Nesse contexto, os recursos alocados são rapidamente associados à demanda e a elasticidade é alcançada. Assim, o SLA (*Service Level Agreement)* é satisfeito, os desperdícios são evitados e o provisionamento sob demanda é atendido (Shivani; Singh, 2018). O desafio é fornecer serviços em nuvem que garantam dinamicamente os requisitos de *Quality of Service* (QoS) das organizações consumidoras e evitem a violação dos acordos de serviço estabelecidos.

Singh; Chana (2016) observam que a complexidade do **gerenciamento de recursos em nuvens** vem aumentando em função da heterogeneidade, da incerteza e da dispersão de recursos envolvidos nos DCs, o que faz com que a alocação de recursos, não possa ser resolvida com *frameworks* de gerenciamento triviais.

Em decorrência dos objetivos desta pesquisa, os conceitos de elasticidade e escalabilidade serão pontuados:

a) **elasticidade** - é uma das principais características dos sistemas em nuvem. Elasticidade tem sido confundida com escalabilidade. Elasticidade diz respeito à capacidade (proativa ou reativa) de aumentar ou diminuir recursos utilizados em um serviço, em tempo de execução (Moore et al., 2013). Embora a importância da elasticidade seja alocar a quantidade de recursos ao aplicativo em consonância com a respectiva necessidade, o tempo que os recursos levam para estarem prontos para uso é um problema em potencial (Al-Dhuraibi et al. 2018);

b) **escalabilidade** - define a habilidade de um sistema lidar com uma quantidade maior de carga de trabalho, enquanto novos recursos são adicionados, mantendo um nível de desempenho aproximado.

1.2 Questões de pesquisa

Em decorrência das fragilidades pontuadas na seção anterior, considera-se que a *Primary Research Question* (PRQ) a ser investigada neste trabalho é a seguinte:

Qual tecnologia apresenta uma melhor acurácia para o provisionamento autonômico da carga de trabalho dos serviços em nuvem, considerando os objetivos dos provedores e das empresas usuárias dos serviços.

Para auxiliar na investigação da solução da questão de pesquisa central, propõe-se quatro *Secondary Research Questions* (SRQ), as quais orientam o presente estudo:

a) **SRQ1**: quais as soluções adotadas por pesquisadores para o provisionamento dos recursos em nuvem que apresentaram uma melhor acurácia? Esta questão analisará os trabalhos de pesquisa elaborados e realizará um estudo comparativo das metodologias e técnicas utilizadas para identificar quais delas apresentam uma melhor capacidade preditiva no provisionamento de recursos;

b) **SRQ2**: como prover a autonomicidade no provisionamento da carga de trabalho dos serviços em nuvem? - Identificar arquiteturas mais efetivas para atender as características desejadas de autonomicidade;

c) **SRQ3**: como definir uma arquitetura preditiva de provisionamento de recursos, de alta acurácia, para ser utilizado por provedores de serviços em nuvem? - Com a análise obtida como resultado do SRQ1, definir uma arquitetura preditivo que atenda aos objetivos desta pesquisa.

1.3 Objetivos

O objetivo geral e os objetivos específicos que norteiam a realização desta Tese são declarados a seguir.

1.3.1 Objetivo geral

O objetivo geral desta Tese foi propor uma arquitetura autonômica que forneça suporte a um processo mais eficaz de alocação dinâmica de recursos em provedores de infraestrutura como serviço (IaaS).

1.3.2 Objetivos específicos

A partir do objetivo geral, os seguintes objetivos específicos foram estabelecidos, considerando-se como escopo as empresas provedoras de serviços em nuvem:

a) propor uma arquitetura preditiva que possibilite a identificação de quantitativos de recursos demandados por uma aplicação, ao longo de sua utilização, num ambiente de nuvem;
b) instanciar a arquitetura, permitindo analisar a efetividade da proposta e quantificar os resultados potencialmente obtidos, em termos do desempenho (acurácia na predição);
c) comparar a arquitetura proposta às demais abordagens analisadas, por métricas de desempenho e de sua capacidade de acompanhar o escalonamento preditivo.

1.4 Hipóteses de pesquisa

Responder às questões de pesquisa envolve validar o uso da arquitetura proposta neste trabalho por provedores de serviço a respeito do gerenciamento de recursos de serviços por provedores de serviços em nuvem. Para contribuir com este esforço, quatro hipóteses foram investigadas:

a) **Hipótese 1 (H1)** - A arquitetura proposta é preferível em relação às soluções atuais utilizadas pelo provedor;
b) **Hipótese 2 (H2)** - A arquitetura proposta é útil para suportar o processo decisório;
c) **Hipótese 3 (H3)** - A arquitetura proposta é acurada o suficiente para suportar o processo de gerenciamento de recursos de serviços em nuvem;
d) **Hipótese 4 (H4):** A arquitetura proposta é eficaz para suportar o processo de gerenciamento de recursos.

A seguir são detalhadas as hipóteses avaliadas.

1.4.1 Hipótese 1 – Preferência

a) **Hipótese nula** - não há diferença na preferência dos gestores quanto ao uso da arquitetura e do método atual de alocação de recursos;
b) **Hipótese alternativa** - preferências diferentes em relação ao uso da arquitetura e do método atual de alocação de recursos;

c) **Medição necessária** - preferência dos gestores em relação ao método de alocação de recursos.

1.4.2 Hipótese 2 – Utilidade

a) **Hipótese nula -** os gestores não consideram a arquitetura útil;
b) **Hipótese alternativa** - a arquitetura é útil;
c) **Medição necessária** - utilidade da arquitetura.

1.4.3 Hipótese 3 – Acurácia

a) **Hipótese nula** - os gestores não consideram que a arquitetura possui acurácia em sua estimativa;
b) **Hipótese alternativa** - a arquitetura possui acurácia;
c) **Medição necessária** - acurácia da arquitetura.

1.4.4 Hipótese 4 – Eficácia

a) **Hipótese nula:** os gestores não consideram que a arquitetura eficaz para fornecer suporte ao processo de gerenciamento de recursos;
b) **Hipótese alternativa:** a arquitetura é eficaz para fornecer suporte ao processo de gerenciamento de recursos;
c) **Medição necessária:** eficácia da arquitetura.

1.5 Síntese da proposta

A arquitetura proposta nesta Tese é específica para suporte ao gerenciamento de serviços, capacidade e recursos, considerando a estratégia de organizações provedoras de serviços em nuvem. A partir da problemática apresentada, surgiu a necessidade de se desenvolver uma solução que possa fornecer suporte à busca do equilíbrio entre necessidade dos usuários, estratégia do negócio e retorno sobre o investimento, que pudesse ser utilizada no **processo de gerenciamento de recursos** em provedores de serviços de nuvem.

Adotando uma abordagem mais formal, o objetivo desta tese de doutorado foi apresentar uma arquitetura autonômica para suporte ao processo de gerenciamento de recursos (no contexto do gerenciamento de capacidade) de infraestrutura em provedores de nuvem. Em termos específicos, a arquitetura proposta utiliza uma metodologia híbrida com técnicas preditivas e reativas. Para o segmento preditivo, objetivando-se alcançar a acurácia desejada

na predição, foram utilizadas Redes Neurais Recorrentes na arquitetura Memória de Curto Longo Prazo Empilhada *(Stacked Long Short-Term Memory*).

Como resultados produzidos, são apresentados a arquitetura e o módulo de alocação de recursos. Dentre as diversas contribuições que serão apresentadas a partir dos resultados deste trabalho de pesquisa que resultou neste documento de tese, destacam-se a definição, formalização e aplicação de uma arquitetura para alocação de recursos de serviços de infraestrutura em nuvem. Por meio da realização de um estudo de caso em um provedor de serviços real, no qual houve a avaliação de todo o arcabouço conceitual e de sua proposta preditiva (*engine*) em experimento desenvolvido em ambiente simulado, a arquitetura se apresentou efetiva e recebeu avaliação favorável entre os gestores de serviços e especialistas envolvidos.

1.6 Contribuições da pesquisa

Neste trabalho buscou-se identificar contribuições científicas que pudessem produzir resultados práticos para os provedores de serviços em nuvem, em especial para os provedores de pequeno e médio porte.

Em relação à pesquisa científica, são apontadas as seguintes contribuições:

a) desenvolver uma arquitetura para a solução de um problema de negócio de provedores de serviços em nuvem, com a aplicação efetiva de uma solução computacional, contribuindo para o estado da arte na área de pesquisa denominada *Business-driven IT Management (BDIM*), com foco na área de gerenciamento de serviços em nuvem (capacidade e recursos).
b) combinar e avaliar as séries temporais com os indicadores da análise técnica.
c) estudar a teoria na qual são baseadas as redes neurais recorrentes LSTM autorregressivas com entradas exógenas (variável vindas de fora do modelo) para séries temporais.
d) propor uma arquitetura preditivo para alocação de recursos em nuvem utilizando *Stacked Long short-term memory neural network*.

Considera-se que as contribuições desta pesquisa também são relevantes para o estado da arte em **Computação em Nuvem**. Considerando-se a complexidade, tornou-se importante estabelecer instrumentos eficientes para a análise, buscando uma melhor compreensão dos dados envolvidos. Assim, a abordagem proposta permite a criação de instrumentos mais eficientes e diretos para melhor explorar o aproveitamento dos recursos e redução do tempo de execução das aplicações.

1.7 Organização do documento

Este documento foi elaborado visando fornecer uma visão global sobre a pesquisa realizada. Para cumprir seu objetivo, o documento está dividido em seis capítulos, organizado como segue: na seção 2 apresenta-se a fundamentação teórica. Na seção 3 são abordadas soluções já fornecidas para alguns dos problemas levantados. Os aspectos metodológicos são relatados na seção 4. Na seção 5 é apresentada a arquitetura proposta, seguida de uma discussão sobre suas propriedades e resultados obtidos durante a pesquisa. Finalmente, na seção 6 é apresenta uma visão conclusiva do trabalho, destacando-se suas contribuições e limitações da pesquisa e trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

> *With the significant advances in Information and Communications Technology over the last half century, there is an increasingly perceived vision that computing will one day be the 5th utility (after water, electricity, gas, and telephony) (Buyya et al. 2009).*

Neste capítulo são apresentados conceitos e teorias que deram suporte a esta pesquisa e que se tornam indispensáveis para a compreensão das demais etapas da pesquisa.

2.1 Computação em nuvem

A **Computação em Nuvem** se beneficia das pesquisas em virtualização, computação distribuída, computação utilitária e, mais recentemente, em serviços de rede. Kumar *et al.* (2018) pontua que o aprimoramento das pesquisas em Bancos de Dados Distribuídos, Computação Paralela, Computação em Grade e Computação Distribuída resultou na Computação em Nuvem.

A Computação em Nuvem oferece uma arquitetura orientada a serviços, redução da sobrecarga de tecnologia para o usuário, flexibilidade, redução no custo total de propriedade, serviços sob demanda e muitas outras facilidades. Essa tecnologia apresenta um modelo que permite o acesso por rede ubíqua e sob demanda, a um conjunto compartilhado de recursos de computação (como redes, servidores, armazenamento, aplicações e serviços) que possam ser rapidamente provisionados e liberados com um mínimo de esforço de gerenciamento ou interação com o *Cloud Service Provider* (CSP) (Mell; Grance, 2011).

O NIST- *National Institute of Standards and Technology* (Mell; Grance, 2011) relaciona um conjunto de características essenciais do modelo da Computação em Nuvem, os quais são considerados os principais benefícios do modelo:

a) **amplo acesso por rede:** os recursos estão disponíveis através de redes padrões e acessados por mecanismos padronizados que promovem o uso em diversificadas plataformas;
b) **agrupamento de recursos:** os recursos de computação do CSP são agrupados para atender a múltiplos consumidores em modalidade *Multitenancy*, com recursos físicos e virtuais dinamicamente atribuídos, conforme a demanda. Há independência de localização geográfica, posto que o consumidor, em geral, não controla ou conhece a localização exata dos recursos fornecidos;
c) **elasticidade rápida:** os recursos podem ser provisionados e liberados, automaticamente, para aumentar ou diminuir conforme a demanda. Para o

consumidor do serviço, os recursos disponíveis parecem ser ilimitados e podem ser alocados em qualquer quantidade e a qualquer tempo;

d) **serviço mensurado:** os sistemas na nuvem controlam e otimizam o uso dos recursos através de medições em um nível de abstração apropriado para o tipo de serviço. A utilização de recursos pode ser monitorada, controlada e informada, gerando transparência tanto para o provedor como para o usuário;

e) **autosserviço sob demanda:** o consumidor pode provisionar recursos de computação, como tempo de servidor e armazenamento em rede, automaticamente e conforme necessário, sem intervenção humana dos CSPs.

Considerando os benefícios da computação em nuvem, cada vez mais as empresas transferem seus serviços para *Data Centers* (DC). É importante que os CSPs forneçam recursos em nuvem com alta elasticidade e custo-benefício e, em seguida, obtenham uma boa qualidade de serviço (*Quality of Service*-QoS) para seus clientes. Atender a QoS com recursos econômicos é um problema desafiador para os CSPs porque as cargas de trabalho das *Virtual Machines* (VMs) sofrem variação ao longo do tempo. É necessário fornecer um método preciso de previsão de carga de trabalho de VMs para provisionamento de recursos que gerencie com eficiência os recursos da nuvem (Aloufi *et al*., 2021).

Os serviços básicos oferecidos são definidos como **Infraestrutura como Serviço (IaaS), Software como Serviço (SaaS) e Plataforma como Serviço (PaaS).** Uma nuvem pública ou uma nuvem externa são serviços em que os clientes recebem máquinas virtuais sob demanda. As nuvens privadas são utilizadas por grandes organizações. Eles constroem seus próprios ambientes e os utilizam. As nuvens híbridas apresentam uma combinação de ambientes de nuvem pública e privada; esta é a escolha preferida para a maioria dos usuários de nuvem.

No **modelo IaaS**, o provedor de serviços gerencia a infraestrutura (servidores reais, rede, virtualização e armazenamento de dados) usando uma conexão com a Internet. A organização usuária pode acessá-la por meio de uma *Application Program Interface* (API) ou painel de controle e, essencialmente, contrata o uso da infraestrutura. A organização usuária gerencia componentes como sistema operacional, aplicações e middleware, enquanto o provedor fornece o hardware, a rede e os servidores, tendo também a responsabilidade pela correção de interrupções, reparos e resolução de problemas de hardware.

Os serviços ofertados no **modelo PaaS** incluem não apenas o ambiente de implementação, mas também repositórios, ambientes de desenvolvimento, de teste, de gerenciamento de desempenho; e serviços de correio, de log e de banco de dados. Nesse

modelo os provedores gerenciam os recursos de infraestrutura e o cliente pode implantar aplicativos na infraestrutura de nuvem utilizando linguagens de programação e ferramentas. Isso permite que os usuários se concentrem apenas na manutenção de seus negócios. Esse modelo se popularizou por eliminar custos e a complexidade no gerenciamento dos recursos de hardware e software para executar aplicativos de negócios. Registra-se dois benefícios significativos: a redução de custos e do tempo do ciclo de desenvolvimento.

Os serviços PaaS são utilizados para projetar, experimentar, construir, testar e entregar aplicativos personalizados. Nesse processo, os desenvolvedores de aplicativos e as demais equipes podem se concentrar no conhecimento de aplicativos e no domínio de seus negócios, em lugar de gerenciar recursos de hardware e de software. A plataforma garante que os consumidores não precisem continuar investindo em atualizações e manutenção de sistema operacional. É responsabilidade do provedor gerenciar recursos de infraestrutura e plataformas para que as organizações não precisem se preocupar com licenças, versões de software, gerenciamento de *patches* e assim por diante. Além disso, a flexibilidade e a disponibilidade de recursos melhoram a colaboração entre as equipes de desenvolvimento e teste. A maioria dos aplicativos é hospedada no ambiente de nuvem virtualizado baseado em *Multitenancy*. Este conceito foi introduzido para solucionar os problemas de escalabilidade, definido como “uma única instância de plataforma / *contêiner* capaz de manipular ou implantar diferentes tipos de aplicativos”. Nesse ambiente, a escalabilidade é uma tarefa desafiadora para a utilização efetiva dos recursos e para aumentar o lucro dos provedores.

No **modelo SaaS** certos pressupostos precisam ser analisados, em detrimento de softwares tradicionais. A confiança no fornecedor é um quesito fundamental em qualquer solução. Para software licenciado, a confiança resume-se na qualidade dos serviços de suporte, na atualização e na longevidade do fornecedor. As soluções **SaaS** oferecem uma interface mais nítida de terceirização, com responsabilidades definidas e vinculadas aos serviços entregues. Permite estabelecer relações de confiança em um novo nível, onde qualquer falha interna é imediatamente imputada ao provedor.

Essa analogia também pode ser utilizada em relação à segurança lógica e física, visto que, teoricamente, quanto mais encapsulado for o produto, mais seguro será o serviço, ficando os componentes encapsulados na infraestrutura do fornecedor. A propriedade de dados é um aspecto muito impactado com essa mudança de paradigma, onde a questão preponderante é onde estão armazenados os dados e quem tem acesso. Os serviços normalmente oferecem garantias de recuperação, cópia e redundância de dados em um nível mais abstrato.

Como aplicativos **SaaS** são dirigidos a uma arquitetura *Multitenancy*, normalmente não aceitam customizações. Porém, aplicações SaaS são projetadas para suportar configurações em parâmetros, que afetam as funcionalidade e aparência da aplicação.

Considerando-se que não há preocupações com instalação, configuração e manutenção do ambiente de desenvolvimento, os desenvolvedores podem se preocupar unicamente com os detalhes das regras de negócios, diminuindo o tempo para desenvolvimento do software e aumentando por consequência sua produtividade.

Aplicações SaaS não podem acessar os sistemas internos de uma empresa, elas precisam oferecer protocolos de integração bem como APIs que possibilitem conexões através da rede. A ubiquidade de aplicações SaaS e de outros serviços da internet, além da padronização de APIs, aumentou o desenvolvimento de *mashups* (aplicações que combinam dados, apresentação e funcionalidades de múltiplos serviços). Fato que aumenta a diferença para os softwares licenciados, visto que estes não são facilmente integrados fora do firewall da empresa. Certamente inspirado no sucesso das redes sociais, aplicações SaaS podem prover características que permitam os usuários colaborarem e compartilharem informações. Uma colaboração, implícita ou explícita, entre usuário de clientes distintos torna-se possível a partir de um software com hospedagem centralizada. Permite o acesso e utilização dos softwares disponíveis no mercado através de uma rede remota ou conexão à Internet. O **SaaS** ainda deve ser acessível por qualquer tipo de dispositivo computacional, garantindo que todos possam ver as informações simultaneamente, sendo um desafio melhorar o acesso aos dados, para facilitar o gerenciamento de privilégios e o controle da utilização dos dados.

O modelo de licenciamento dos **SaaS** é baseado em assinatura, utilizando uma abordagem baseada em tempo de uso, cobrado somente aquilo que foi utilizado pelo usuário. As aplicações **SaaS** são completamente gerenciadas pelo fornecedor, e acordos SLA regem a qualidade, disponibilidade e suporte que o provedor deve prover para o cliente. Algumas formas de gerenciamento essências para SaaS incluem o provisionamento, funções de configuração, controle de pagamento, monitoração e suporte.

Algumas limitações podem atrapalhar a aceitação de aplicações **SaaS** ou até inviabilizar sua utilização. Como os dados estão sendo armazenados na nuvem, nos servidores do provedor, a segurança pode ser vista como um problema. O **SaaS** está hospedado na nuvem, longe dos usuários da aplicação. Este fato introduz latência para o ambiente, tornando o modelo não apropriado para aplicações que exigem um pequeno tempo de resposta. Arquiteturas multiclientes, que impulsionam a eficiência de custos para os provedores de solução, não permitem uma customização verdadeira para clientes em larga escala, proibindo

sua utilização em cenários que a customização é imprescindível. Algumas aplicações de negócio requerem acesso ou integração com dados atuais do cliente. Quando o volume de dados é muito grande ou os dados contêm informações sigilosas, a integração com o software remoto é cara ou apresenta muitos riscos.

A utilização de **SaaS** traz vantagens largamente apontadas, pelo fato do aplicativo se manter disponível a partir de quaisquer computadores ou dispositivos computacionais. Por essa razão **SaaS** tende a ter altos índices de adoção, com pouco esforço para aprendizagem, dado que já existe uma familiarização geral com o uso da internet.

Não existem taxas de licença, cobrado apenas pela utilização, o que significa menor custo inicial, sendo o provedor o gerente de toda a infraestrutura de TI, bem como seus custos e ainda os gerentes humanos. As atualizações são transparentes e de única responsabilidade do provedor, não existem arquivos para download ou para serem instalados. Além disto, o provedor gerencia a disponibilidade e o cliente não tem que se preocupar em adicionar hardware ou largura de banda com o crescimento de usuários. Os provedores podem escalar sua infraestrutura de forma indefinida para atender a demanda de um cliente, além de poder oferecer funcionalidades para atender necessidades específicas, agregando integração a sistemas ERP (*Enterprise Resource Planning*) legados ou sistemas de produtividade.

2.1.1 Modelos de disponibilização dos serviços em nuvem

Os modelos de disponibilização em nuvem representam diferentes tipos de ambiente, distinguindo-os por propriedade, tamanho e acesso. Os modelos mais comuns são: Públicas, Comunitárias, Privadas e Híbridas. O modelo a ser utilizado é uma escolha particular. Nenhuma nuvem é igual a outra, nem mesmo quando elas são do mesmo tipo. Também não há dois serviços utilizados para resolver o mesmo problema. Toda nuvem extrai, agrupa e compartilha recursos de computação escaláveis em uma rede. Elas são criadas usando uma combinação exclusiva de tecnologias, que quase sempre inclui um sistema operacional, algum tipo de plataforma de gerenciamento e interfaces de programação de aplicações (APIs). Além disso, é possível adicionar aplicações de virtualização e automação a todos os tipos de nuvem para incluir mais recursos ou obter maior eficiência. A seguir serão apresentadas diferenças sobre os diferentes tipos de disponibilização em nuvem:

a) **nuvem pública** - são ambientes criados em uma infraestrutura de TI que não são de propriedade do usuário final. Eles têm sua infraestrutura localizada nas instalações de um provedor. São utilizadas por empresas para aplicações

secundárias, como e-mail ou a hospedagem de sites. Por esse motivo os recursos são compartilhados entre os clientes, cada um com seu nível separado de acesso aos próprios dados. Nesse tipo de nuvem, os dados são isolados. Portanto, os clientes do mesmo fornecedor não têm acesso aos dados uns dos outros. Suas vantagens são o rápido provisionamento, custo sob demanda e custos reduzidos em relação a uma nuvem privada. As nuvens públicas tradicionais eram executadas ***off-premises***. Atualmente os provedores de nuvem oferecem serviços nos *data centers* ***on-premise*** dos clientes. Com isso, as distinções baseadas em local e propriedade se tornaram ultrapassadas. Todas as nuvens se tornam públicas quando os ambientes são particionados e redistribuídos para vários locatários. A cobrança de taxas deixou de ser uma característica essencial das nuvens públicas. Alguns provedores de nuvem permitem que os locatários as usem gratuitamente. A infraestrutura *bare-metal* usada por provedores de nuvem pública também pode ser extraída e vendida como IaaS, ou desenvolvida e comercializada como PaaS;

b) **nuvem privada** - são ambientes de nuvem dedicados a um usuário final. A infraestrutura é utilizada somente por um cliente. Ou seja, não é compartilhada mesmo que esteja remotamente localizada. Existe, também, a opção de nuvem privada no local, que gera mais custo, porém proporciona mais controle sobre a infraestrutura. Geralmente, esse tipo de nuvem é escolhido para armazenar dados sigilosos e estratégicos da empresa. O tempo de resposta é mais rápido, já que os servidores estão dentro da corporação, garantindo uma baixa latência de rede, além de mais segurança. O ambiente é geralmente executado atrás do *firewall* do usuário. Todas as nuvens se tornam privadas quando a infraestrutura de TI subjacente é dedicada e o cliente tem acesso totalmente isolado a ela. As nuvens privadas não precisam mais ser baseadas em infraestruturas de TI *on-premise.* Atualmente, as organizações estão criando nuvens privadas em *data centers* alugados localizados *off-premise*. Dessa forma, todas as regras sobre local e propriedade estão obsoletas. Isso também gerou vários subtipos de nuvem privada, incluindo:

- **nuvem privada gerenciada -** os clientes criam e usam uma nuvem privada implantada, configurada e gerenciada por um fornecedor terceirizado. Trata-se de uma opção para empresas com poucos funcionários ou com equipes de TI sem a qualificação necessária para fornecer infraestrutura e serviços de

nuvem privada adequados;

- **nuvem dedicada –** uma nuvem dentro de outra. É possível ter uma nuvem dedicada em uma nuvem pública ou em uma nuvem privada. É possível implantar uma nuvem dedicada para o departamento de contabilidade dentro da nuvem privada da organização.

c) ***nuvem híbrida*** - é caracterizada quando a empresa opta por duas formas de armazenamento (pública/privada). O software permite transferir cargas de trabalho entre as duas. O desafio em usar esse tipo de nuvem são os custos de link e a segurança das informações. Geralmente, a nuvem privada executa softwares com informações confidenciais, e não convém transferir esses dados para terceiros, mesmo que por um curto período. É um ambiente de TI aparentemente único criado a partir de vários outros ambientes conectados por redes locais (LANs), redes de área ampla (WANs), redes privadas virtuais (VPNs) e/ou APIs. As características das nuvens híbridas são complexas e os requisitos podem variar dependendo da pessoa a quem você pergunta. Por exemplo, uma nuvem híbrida pode ter de incluir:

   - no mínimo, uma nuvem privada e uma nuvem pública;
   - duas ou mais nuvens privadas;
   - duas ou mais nuvens públicas;
   - um ambiente virtual ou *bare-metal* conectado a, no mínimo, uma nuvem pública ou privada

   Mas todo sistema de TI se torna uma nuvem híbrida quando aplicações podem se mover por vários ambientes diferentes, mas conectados entre si. Pelo menos alguns desses ambientes devem ser originados de recursos de TI consolidados que possam ser escalados sob demanda. Todos esses ambientes precisam ser gerenciados como um só por meio de uma plataforma integrada de gerenciamento e orquestração;

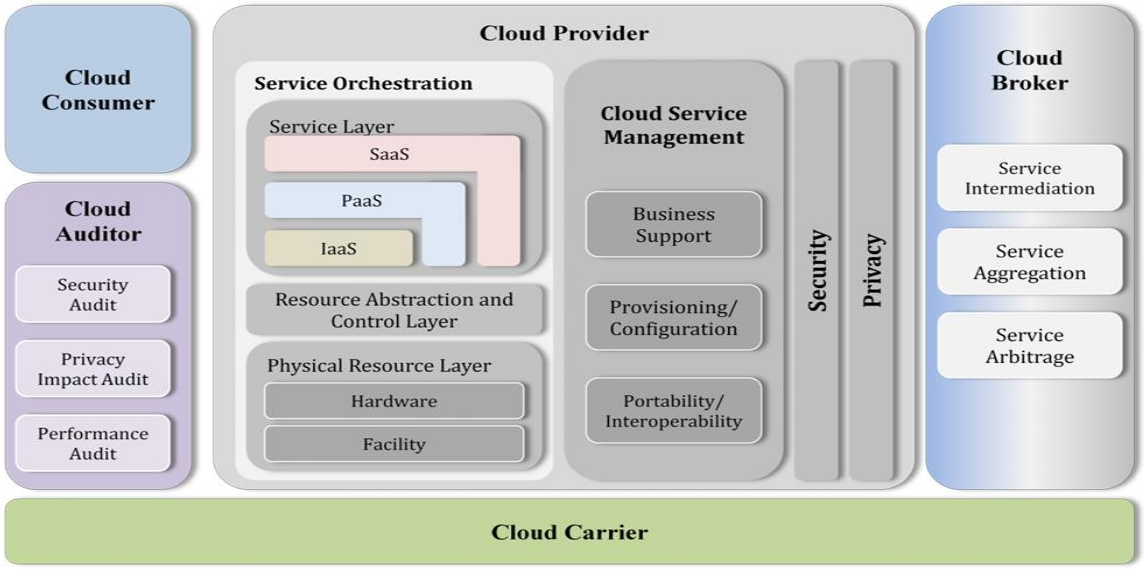
d) **multicloud** - É uma abordagem composta por mais de um serviço e de um fornecedor de nuvem, pública ou privada. Todas as nuvens híbridas são multiclouds, mas nem todas as multiclouds são nuvens híbridas. As multiclouds tornam-se nuvens híbridas quando várias nuvens estão conectadas por alguma forma de integração ou orquestração. Um ambiente multicloud pode existir propositalmente (para melhor controle de dados confidenciais ou como espaço de armazenamento redundante para recuperação de desastres

aprimorada) ou por acidente (normalmente, como resultado de TI invisível). Garantidamente, ter várias nuvens está se tornando mais comum entre empresas que buscam melhorar a segurança e o desempenho em um portfólio expandido de ambientes.

2.1.2 Provedores de serviço em nuvem

O Modelo de Referência Conceitual da computação em nuvem, apresentado pelo NIST (Liu *et al*., 2011) e exibido na Figura 1, descreve uma arquitetura genérica de alto nível para facilitar o entendimento dos requisitos, usos, características e padrões da computação em nuvem.

Figura 1 – Modelo de Referência Conceitual



Fonte: Liu *et al*. (2011).

Conforme a Figura 1, a arquitetura define cinco atores: consumidor, provedor, operador, auditor e corretor de nuvem. Esses atores representam uma entidade que participa de uma transação ou executa tarefas, compondo assim o ecossistema dos serviços em nuvem. São eles:

a) consumidor *(Cloud Consumer)* - mantém relacionamento comercial e usa serviços de provedores de nuvem;
b) provedor *(Cloud Provider)* - disponibiliza serviço às partes interessadas;
c) auditor *(Cloud Auditor)* – realiza avaliação independente de serviços em nuvem, operações de sistemas de informação, desempenho e segurança da

implementação da nuvem;

d) operador *(Cloud Broker)* - gerencia o uso, desempenho e entrega de serviços em nuvem e negocia relacionamentos entre consumidores de nuvem;

e) corretor *(Cloud Carrier)* - fornece conectividade e transporte de serviços de nuvem de provedores de nuvem para consumidores de nuvem.

Definir claramente as atribuições de um *Cloud Service Provider* (CSP) não é trivial, especialmente do ponto de vista de um empresário interessado em saber se seu provedor está efetivamente lhe atendendo a partir de uma nuvem. Segundo a definição apresentada em Axelos (2021, p.2):

> "O provedor de serviços em nuvem é uma organização que oferece infraestrutura, plataformas ou aplicativos baseados em nuvem privada, ou pública sob demanda (ou pagamento por uso)".

Um CSP é uma empresa que oferece componente/serviço de computação em nuvem. Um **serviço** em nuvem é qualquer sistema que fornece disponibilidade sob demanda de recursos de computador, e.g. armazenamento de dados e poder de computação, sem o gerenciamento ativo direto pelo usuário.

Alguns CSPs se especializam em apenas um tipo de serviço, enquanto outros fornecem uma combinação de modelos de serviço e de implantação. Alguns fornecem serviços altamente padronizados para uma gama diversificada de consumidores, enquanto outros fornecem serviços personalizados para organizações de consumidores individuais.

As empresas não necessitam manter sua própria infraestrutura de computação ou *Data Center* (DC). Elas podem alugar acesso de aplicativos a armazenamento de um provedor que trata e processa os dados da empresa. Podem armazenar e processar as informações na nuvem, seja por terceirização ou utilizando uma nuvem interna (*private cloud)* que eles próprios desenvolveram ao implementá-lo nos recursos dedicados da empresa e na infraestrutura usando serviços *on-premise*. Elas também podem usar uma abordagem diversificada ou **nuvem híbrida**, onde é utilizada uma abordagem privada e pública.

2.1.3 Riscos da computação em nuvem

Embora os benefícios da computação em nuvem superem os **riscos,** as empresas usuárias devem tomar cuidado durante a migração. Questões relativas à segurança, disponibilidade, confiabilidade e tolerância a falhas são pontos que preocupam. Os servidores

em nuvem são tecnologias que requerem criptografia e constante monitoramento para que eles possam operar em sua totalidade, sem causar interrupções. Com a contratação de serviços em nuvem, essas desvantagens poderão ser substituídas por benefícios, como: alta disponibilidade da nuvem, melhor desempenho dos servidores e programação em nuvem integrada. Algumas das preocupações são:

a) **o tempo de inatividade** - Como os sistemas em nuvem são baseados na internet, interrupções de serviço podem ocorrer. Existem práticas para minimizar esse tempo em um ambiente de nuvem, projetando serviços com alta disponibilidade e recuperação de desastres. Se os serviços tiverem uma baixa tolerância de falhas, considerar implantações em várias regiões com *failover* automatizado para garantir a melhor continuidade dos negócios. Implementar um plano de recuperação de desastres alinhado visando negócios que forneça o menor tempo de recuperação (*Recovery Time Objective* - RTO) - indicador que mensura o tempo máximo em que um sistema ou uma informação pode ficar indisponível após uma falha;
b) **objetivos de ponto de recuperação (RPO)** - Para manter os servidores em nuvem sempre ativos, será necessário ter um ambiente de TI com alta disponibilidade, backup em nuvem e contar com um suporte on-line para que as soluções em nuvem possam ser implementadas no ambiente em homologação. Os servidores em nuvem podem estar sempre ativos por se tratar de um ambiente com alta disponibilidade e alta taxa de transferência: os erros de nuvem indisponível ou servidor sobrecarregado, poderão ser evitados com um suporte em TI totalmente dedicado às aplicações em nuvem;
c) **acesso compartilhado -** um dos princípios fundamentais da computação em nuvem é o modelo *Multitenancy*. Geralmente os clientes compartilharem os mesmos recursos de computação: CPU, armazenamento, espaço, memória etc. A *Multitenancy* é uma preocupação não só pelos riscos de os dados privados vazarem acidentalmente para outros inquilinos, mas pelos riscos adicionais do compartilhamento de recursos;
d) **vulnerabilidades virtuais -** cada CSP é um grande usuário de virtualização. E cada camada de virtualização representa uma importante plataforma na infraestrutura de TI, com vulnerabilidades que podem ser exploradas. Servidores virtuais estão sujeitos aos mesmos ataques que atingem os servidores físicos, assim como novas ameaças estão explorando falhas do

hypervisor;

e) **autenticação, autorização e controle de acesso -** os mecanismos de controle de autenticação, autorização e acesso à nuvem são fundamentais;

f) **disponibilidade -** quando se está no papel de cliente de um provedor de nuvem pública, redundância e tolerância a falhas não estão sob seu controle. Todo provedor alega implementar tolerância a falhas e disponibilidade. Por conta própria, a empresa usuária deveria fazer *backup* dos dados compartilhados na nuvem. Ou se resguardar, em contrato, estabelecendo as responsabilidades por perdas de dados. Deve-se considerar um modelo de governança em que um fornecedor detém a responsabilidade global para as interrupções e as falhas de segurança;

g) **posse -** o risco é quase sempre uma surpresa para os clientes de nuvem, mas muitas vezes eles não são os únicos proprietários dos dados. Muitos provedores de nuvem pública possuem cláusulas em contratos que explicitam que os dados armazenados pertencem ao provedor – e não ao cliente;

h) **visibilidade da nuvem -** Não é fácil determinar a probabilidade de falhas de segurança ou disponibilidade, especialmente para um determinado fornecedor, ou se esses riscos vão levar a danos substanciais para os clientes. É preciso analisar detalhadamente as opções para proteção de dados sensíveis oferecidas pelos CSPs, o quanto fluem através da rede, o quanto residem em um servidor, ou na infraestrutura de armazenamento. É conveniente ter-se uma estratégia de mitigação de riscos de modo que seja possível migrar o trabalho para um novo provedor (ou voltar a mantê-lo *on-premise*) com rapidez e facilidade em caso da ocorrência de uma eventualidade (*Vendor Lock-in*).

Especificamente no que tange à segurança dos serviços em nuvem, alguns pontos são ainda destacáveis:

a) **privacidade** - os dados dos usuários podem ser acessados pela empresa anfitriã sem permissão. O CSP pode acessar os dados que estão na nuvem a qualquer momento. Eles podem alterar acidental ou deliberadamente, ou até excluir informações;

b) **segurança** - os serviços baseados em nuvem envolvem terceiros para armazenamento e segurança. A segurança é considerada uma ameaça real para os serviços em nuvem;

c) **conformidade** - existem muitos regulamentos relacionados a dados e a hospedagem. Para cumprir estes regulamentos o usuário precisa adotar modelos de implantação, nem sempre de baixo custo;
d) **sustentabilidade** - como minimizar os efeitos da computação em nuvem no meio ambiente. Os países com condições favoráveis, buscam atrair data centers de serviços em nuvem. A questão aberta é, se além dos benefícios da natureza, esses países teriam infraestrutura suficiente para manter os avanços da tecnologia;
e) **abuso** - ao fornecer serviços em nuvem, deve-se verificar se o cliente está adquirindo esses serviços para fins legais;
f) **custo mais alto** - para usar serviços em nuvem é preciso ter uma rede com largura de banda maior que as redes comuns de internet. Observam-se problemas ao utilizar-se um serviço de nuvem comum, quando se trabalha com projetos complexos;
g) **recuperação de dados perdidos em contingênci**a - antes da assinatura com qualquer CSP deve-se analisar todas as normas e documentações do provedor e verificar se seus serviços atendem aos requisitos e com a devida manutenção;
h) **manutenção (gerenciamento) da nuvem -** os provedores devem possuir uma grande infraestrutura de recursos e, como consequência, enfrentam grandes desafios, tais como riscos, satisfação do usuário etc.;
i) **falta de recursos/experiência qualificada** - um dos problemas enfrentados é a falta de recursos humanos qualificados. A carga de trabalho em nuvem vem aumentando, de modo que as empresas de prestação de serviços em nuvem precisam de um avanço rápido e contínuo. À medida que novas tecnologias estão surgindo, recursos humanos mais qualificados precisam ser alocados;
j) **pagamento por uso** - os serviços em nuvem são cobrados sob demanda. O usuário pode estender ou compactar o volume dos recursos, conforme suas necessidades. Não é fácil para os gestores identificar a demanda, bem como as flutuações nas estações e para os diferentes eventos.

2.1.4 O Futuro da Computação em Nuvem

A ascensão do Aprendizado de Máquina (*Machine Learning* - ML) levou à

computação em nuvem a merecer ênfase com essa perspectiva. As novas plataformas baseadas em ML utilizam recursos dos CSPs e os combinam com os principais cenários de aprendizado de máquina. O resultado é uma plataforma de autoatendimento, fácil de usar, capaz de executar trabalhos de treinamento em ML e um ambiente de hospedagem escalável que oferecem gerenciamento eficiente do ciclo de vida de modelos de aprendizado de máquina.

No ML os desenvolvedores executam o trabalho de treinamento várias vezes com um conjunto diferente de hiper parâmetros até que estejam convencidos com a precisão do modelo. O modelo treinado é implantado para inferência na PaaS, aproveitando a infraestrutura baseada em CPUs (*Central Processing Unit*) e GPUs (*Graphics Processing Unit*) do provedor.

Especialistas em marketing e veteranos em tecnologia apresentam as seguintes considerações sobre o futuro da computação em nuvem (Dillon *et al.* (2010), (Buyya *et al.* (2018):

a) **nuvens privadas x públicas** - as economias de escala, especialização e benefícios de terceirização de nuvens públicas são tão vantajosas que não fará sentido para as empresas operarem seus próprios data centers. Entende-se que será necessária a existências de muitas medidas de segurança e isolamento;
b) **nuvens especializadas** - existem muitas dimensões para um aplicativo: o padrão de carga de trabalho, os regulamentos governamentais, o acesso geográfico, a linguagem de programação utilizada, o framework suportado, os níveis de segurança, desempenho e confiabilidade exigidos e vários outros requisitos mais especializados;
c) **regulamentação governamental** - os provedores de nuvem se tornarão infraestruturas estratégicas. Os provedores se tornarão uma infraestrutura crucial para a economia e os interesses de suas respectivas nações. Qualquer mudança em seus preços afetarão a economia, correndo o risco de um aumento repentino de demanda não prevista;
d) **o debate controle x liberdade - liberdade** é a palavra geral para a adoção da nuvem (sem custos iniciais, sob demanda, autoatendimento, capacitação da classificação). **Controle** (ou falta dele) é a palavra-chave para barrar à adoção por grandes empresas. Os países democráticos experimentaram essa estratégia: às vezes há uma contradição entre os chamados princípios sagrados do estado de direito e da liberdade pessoal;

e) **federações de nuvem** - embora alguns provedores desfrutem de sucesso internacional, observa-se que em qualquer negócio que dependa de confiança, nada supera a confiança em um provedor local. Os clientes migrarão para a nuvem de sua empresa de telecomunicações confiável ou grande provedor de TI. Mas, por outro lado, eles precisarão atingir um público global e desejarão servidores em todo o mundo. Como resultado, observa-se a formação de federações de nuvem, como vemos em alianças de companhias aéreas;
f) **eficiência financeira e sofisticação** - a computação é uma commodity e toda commodity, no futuro, poderá ser negociada, intermediada, arbitrada, especulada e manipulada com instrumentos derivativos;
g) **padrões de nuvem** - já ocorreu uma onda de interesse e discussão sobre a necessidade de padrões em nuvem. No entanto, ainda são encontrados padrões concorrentes. Pelo menos uma especificação padrão formal de um corpo de padrões e vários padrões de fato de grandes players comerciais é necessário;
h) **a guerra do ecossistema** - o sucesso na construção de um ecossistema será um fator determinante para quem ganha e quem perde na nuvem. Não se trata do tamanho e da amplitude do ecossistema, mas de como tudo funciona bem em conjunto;
i) **consolidação horizontal e vertical** - como geralmente acontece em qualquer setor, à medida que a computação em nuvem amadurece, ela se consolida. Isso acontecerá horizontal e verticalmente.

2.2 Gerenciamento de recursos em nuvem

O gerenciamento de recursos em nuvem é o processo de alocação de recursos de computação, armazenamento, rede e indiretamente de energia a um conjunto de aplicações, procurando atender aos objetivos de desempenho dos provedores, usuários e aplicações. Os usuários tendem a se concentrar no desempenho do aplicativo. A estrutura conceitual fornece uma visão de alto nível do componente funcional dos sistemas de gerenciamento de recursos em nuvem e todas as suas interações. Essa área é classificada em oito categorias, ou seja, as atividades de gestão de recursos são as seguintes (Kumar *et al.* 2018):

e) planejamento global de recursos virtualizados;
f) perfil de demanda de recursos;
g) exercício de estimativa de recursos;

h) precificação de recursos e maximização de lucros;
i) agendamento local de recursos;
j) dimensionamento e provisionamento de aplicativos;
k) gerenciamento de carga de trabalho;
l) sistemas de gerenciamento de nuvem.

2.2.1 Planejamento de capacidade

Entregar os projetos no tempo determinado e com a qualidade esperada é um dos principais anseios da gerência das empresas. Entretanto, para que este anseio se concretize é necessário que os gestores conheçam a capacidade de entrega de suas empresas. Para isso é fundamental que ocorra um eficiente **Planejamento de Capacidade,** definindo as reais possibilidades da empresa para a produção e atendimento das demandas, bem como a existência de uma realística análise de impacto. Por meio desse planejamento é possível saber quantos projetos a empresa consegue gerenciar em um período específico, garantindo o estabelecimento de prazos exequíveis diante de novas aplicações ou melhorias (Techtarget, 2021).

Para entender o **Planejamento de Capacidade** é fundamental saber quais são os recursos disponíveis para a realização dos projetos e como eles podem ser alocados para garantir uma maior produtividade em um determinado período. Essa simples ação evita que a empresa cometa erros na definição de prazos, prejudicando sua credibilidade junto às áreas de negócio e aponta quais necessidades precisam ser supridas em caso de expansão.

O Gerenciamento de Capacidade é um processo importante para uma eficiente Entrega de Serviços. Seu objetivo é assegurar que a capacidade da infraestrutura de TI, esteja alinhada com as necessidades do negócio, suportando assim todos os processos do negócio que necessitam da TI, em um custo aceitável. O Plano de Capacidade é o documento principal que descreve as necessidades previstas para o próximo período e pode-se dizer que é a saída deste processo.

O processo de **Gerenciamento de Capacidade** é dividido em três subprocessos:

a) **Gerenciamento de Capacidade de Negócio -** assegurar que as necessidades atuais e futuras do negócio serão consideradas nas operações de TI.
b) **Gerenciamento de Capacidade de Serviço -** garantir que o desempenho dos serviços de TI, corresponda com os Níveis de Serviço (SLAs) acordados.

c) **Gerenciamento de Capacidade de Recursos -** gerenciamento dos recursos individuais da TI: software, hardware e pessoas.

As quatro atividades, a seguir, fazem parte do Gerenciamento de Capacidade, sendo chamadas de atividades interativas, como um PDCA (*P-Plan, D-Do, C-Check, A-Act*) do Gerenciamento de Capacidade:

a) **Monitoramento:** verificar se todos os Níveis de Serviço (SLAs) previamente acordados estão sendo alcançados.
b) **Análise:** os dados coletados através do monitoramento precisam ser analisados para geração de predições futuras.
c) **Ajuste:** implementa o resultado do monitoramento e análise para assegurar o uso otimizado da infraestrutura atual e futura.
d) **Implementação:** implementa a nova capacidade.

Todas as informações coletadas no processo são armazenadas no Banco de Dados de Capacidade (BDC). Este banco é utilizado para formar a base dos relatórios para este processo e contém informações técnicas para o Gerenciamento de Capacidade.

Outros processos do Gerenciamento de Capacidade são:

a) **Gerenciamento da Demanda -** responsável pelo gerenciamento da carga de trabalho na infraestrutura visando utilizar melhor a capacidade atual ao invés de aumentá-la. O comportamento do usuário é influenciado para que se use uma carga de trabalho diferente, como usar recursos de TI em outro horário do dia para aliviar a falta de capacidade.
b) **Dimensionamento de Aplicação -** relacionado à avaliação dos requisitos de capacidade das aplicações durante seu planejamento e desenvolvimento.
c) **Modelagem:** por simulação ou com auxílio de modelos matemáticos é possível a predição dos requisitos futuros da capacidade.
d) **Plano de Capacidade -** desenhado a partir da base dos dados do BDC, dados financeiros, dados do negócio, dados técnicos etc. O plano é orientado para o futuro, tendo como base um período mínimo de 12 meses.
e) **Relatórios -** conferem a desempenho da capacidade durante um período determinado. Os relatórios, por exemplo, podem trazer números que sirvam para comparar os índices dos Acordos de Nível de Serviços.

O gerenciamento de capacidade é parte da Entrega de Serviços. Está diretamente relacionado com os requisitos do negócio e ligado com quase todos os processos do ITIL

(*Information Technology Infrastructure Library*), para monitorar os incidentes e problemas referentes a capacidade e suportando os SLAs acordados (Axelos, 2021).

O **gerenciamento de capacidade para nuvem** está diretamente relacionado a como criar um plano que garanta o volume suficiente de servidores, armazenamento, rede etc., sempre que necessário. Significa a criação de modelos complexos para verificar o quanto está sendo gasto com a infraestrutura de TI. O gerenciamento de capacidade considera ainda os seguintes aspectos:

a) não se pode assumir que a capacidade de computação é dedicada a um grupo de usuários ou um grupo de processos. Tudo em um ambiente de serviços em nuvem é compartilhado usando algum tipo de modelo *Multitenant*, ou seja, para múltiplos clientes. Isso complica a capacidade de modelagem e planejamento;
b) com auto provisionamento, alguns aspectos da capacidade de planejamento perdem importância porque a capacidade pode ser alocada conforme a necessidade. No entanto, como o custo é um motivo fundamental para o uso da computação em nuvem, usar a capacidade que não é necessária tira o valor da nuvem;
c) pode-se utilizar sistemas em nuvem a custos reduzidos, se necessário, para fornecer capacidade temporária.

Os prestadores de serviços e o uso da computação em nuvem tornam o processo de planejamento de capacidade mais complexo. As abordagens de modelagem e as tecnologias estão mudando para acomodar esta complexidade crescente, mas os profissionais de planejamento de capacidade necessitam atualizar as suas competências. Com o passar do tempo, os fornecedores de serviços em nuvem proporcionarão mais desempenho e capacidade de serviços de monitoramento e gestão.

2.2.2 Custos dos recursos em nuvem

O **gerenciamento de custos da nuvem** envolve também o gerenciamento dos custos e das necessidades associadas. Isso significa encontrar maneiras econômicas de maximizar o uso e a eficiência da nuvem. Esses custos incluem instâncias de máquinas virtuais, memória, armazenamento, tráfego de rede, suporte e licenças de software. Esses custos podem ser classificados em:

a) **custo dos serviços** - O gerenciamento de custos da nuvem torna-se complexo, já que esses custos são geralmente descentralizados e variáveis. A implementação de uma estratégia de gerenciamento de custos poderá ajudar as organizações a analisar os custos, o consumo, a comparação e o planejamento da nuvem. Com uma melhor compreensão dos custos e do uso, as empresas poderão realizar a contabilidade com mais eficiência e melhorar o desempenho e o rendimento dessa tecnologia;

b) **custos de migração** - Para definir os custos de migração para nuvem, os CIOs devem calcular o custo para executar uma determinada carga de trabalho e quanto esse custo diminuirá quando essa carga for movida. Estas informações também serão úteis para priorizar e identificar quais provedores serão os melhores para seus requisitos de desempenho e para garantir uma migração bem-sucedida.

É preciso também entender que os custos atuais não desaparecerão imediatamente. O custo de alimentação, manutenção, resfriamento e operação de um *storage array* ou *chassi de servidor* não desaparece até que esses estejam vazios, mesmo que tal custo reduza, conforme a utilização diminui. Se tais componentes, atendem a muitas cargas de trabalho, isso pode levar meses.

O custo de uma equipe dedicada ao gerenciamento de matrizes de armazenamento ou chassis de servidor serão reduzidas. Entretanto, se eles forem responsáveis por outros chassis ou *arrays*, esses custos permaneceram contínuos até que todas as cargas de trabalho sejam encerradas. Ou seja, o custo de operação do DC só desaparece quando o próprio DC é migrado. As operações na nuvem não são gratuitas. O pessoal qualificado ainda é essencial, pois poderá haver uma nova gama de ferramentas de provisionamento, gerenciamento e monitoramento.

Os CIOs que planejam uma migração para a nuvem, mesmo que não estejam fazendo por economia, precisam identificar os custos da migração. Não apenas pelo que vão gastar no ambiente de nuvem, mas pelo planejamento e fases de migração e todos os custos de operação do ambiente após estabelecido. Independente da motivação para a migração, um componente crucial do processo é calcular seus custos. Mais especificamente, os custos de preparação para a migração, da migração para a nuvem e da pós-migração.

Isso é crítico porque nenhuma das motivações para migrar uma infraestrutura de negócios para a nuvem é realmente sem custo. Na verdade, o benefício comercial desejado é sempre oneroso em relação ao custo. Mesmo quando a gestão decide não haver alternativa

aceitável para a mudança para a nuvem, eles precisam ter uma ideia dos custos de migração para a nuvem. Somente assim poderão tomar decisões sobre como e com que velocidade a mudança poderá prosseguir.

O consumo da **computação em nuvem** é geralmente identificado como um modelo *pay-per-use*, caracterizado pela eficiência nos custos. Entretanto, muitas empresas não fazem verificações de rotina para ver o quanto está sendo efetivamente utilizado. Como consequência, elas pagam por recursos que não são utilizados. Quando uma instância específica de um fornecedor de nuvem é adquirida, o cliente recebe uma variedade de opções de tamanhos de máquinas virtuais **(Virtual Machine VM)** para escolher, como apresentado na figura 2 (Saveincloud, 2022). As opções oferecidas, geralmente, dobram os quantitativos, ficando, de princípio, recursos ociosos. Algumas organizações cobram uma taxa única de instalação para provisionar e configurar uma VM. Essa abordagem é utilizada por diversos CSPs.

Figura 2 - Opções fornecidas para máquinas virtuais

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
Compute Optimized - Current Generation					
c4.large	2	8	3.75	EBS Only	$0.1 per Hour
c4.xlarge	4	16	7.5	EBS Only	$0.199 per Hour
c4.2xlarge	8	31	15	EBS Only	$0.398 per Hour
c4.4xlarge	16	62	30	EBS Only	$0.796 per Hour
c4.8xlarge	36	132	60	EBS Only	$1.591 per Hour

Fonte: Saveincloud (2022).

O primeiro desafio é identificar o tamanho suficiente para um bom desempenho, durante uma carga média, e um espaço extra para a expansão dentro da máquina a ser utilizada. O seguinte desafio ocorre quando a VM atual se torna muito pequena para as necessidades do projeto. Se faz necessário aumentar a capacidade para uma VM com maior poder, o que geralmente será duas vezes maior.

A questão é a alocação ser eficaz ou está ociosa, especialmente durante o tempo de baixa utilização. Como resultado, você ainda está pagando por esses recursos de computação reservados, mas não usados. Quando aumentando a infraestrutura, adicionando mais Máquinas Virtuais (VMs) para outros projetos, aplicações, o problema de capacidade não utilizada aumenta mais ainda. Recursos ociosos aumentam proporcionalmente e, como

resultado, a eficiência diminui ainda mais.

O modelo de faturamento *pay-per-use* não é tão flexível quanto a cobrança de eletricidade. Você simplesmente não pode solicitar uma VM que atenda precisamente aos requisitos do projeto agora e que seja dimensionada sem configurações extras e esforços de migração conforme a carga aumenta. Como resultado, tem-se a solicitação de VMs maiores e o pagamento por recursos não utilizados. Alguns provedores não exigem a compra de capacidade com antecedência, mas são faturados com base no consumo efetivo.

Obviamente, os gastos dependem muito do fornecedor de nuvem escolhido e qual unidade de recursos é utilizada para o escalonamento, a disponibilidade de escalonamento automático, entre outros. Para alcançar a máxima eficiência, o usuário precisa conhecer o modelo de precificação com pequenos preços de escala e redimensionamento suave com base na carga, a fim de não reservar recursos extras antecipadamente sem necessidade real.

2.2.3 Acordos de serviço

O SLA (*Service Level Agreement*) determina o que o provedor promete aos clientes sobre disponibilidade, desempenho etc. O SLO (*Service Level Object*) é uma meta que o provedor de serviços deseja atingir. Já o SLI (*Service Level Indicators*) é uma medida que o provedor utiliza para avaliar sua meta. O SLA, SLO e o SLI são relacionados, mas são conceitualmente diferentes. Na prática, as SLIs são as métricas do sistema de monitoramento; os SLOs são regras de alerta e os SLAs são os números das métricas de monitoramento aplicadas aos SLOs.

O SLA é um ponto fundamental no uso dos serviços em nuvem, posto estabelecer a garantia da prestação de serviços e definir os níveis de qualidade que devem ser atendidos pelo provedor. Trata-se de **um documento a ser criado para que os serviços possam ser mensurados**, funcionando como um contrato entre a gestão e o cliente, apresentando uma maior transparência na obtenção dos resultados pretendidos. Nele podem constar todos os serviços que o contratante espera do fornecedor ou as responsabilidades de cada envolvido, deixando claro as métricas, as expectativas e as responsabilidades em termos de disponibilidade, escalabilidade, confiabilidade e segurança do serviço. **O SLA se torna uma garantia para quem paga pelo serviço e para quem o presta (**Hein, 2022**).**

Normalmente, o SLO e o SLA são semelhantes, enquanto o SLO é mais rígido que o SLA. Os SLOs são geralmente visualizados apenas pelos colaboradores internos e os SLAs são para os externos. Se a disponibilidade de um serviço viola o SLO, as operações

precisam reagir rapidamente para evitar a quebra do SLA; caso contrário, a empresa precisará fazer algum tipo de ressarcimento aos clientes. O SLA, SLO e SLI baseiam-se na suposição de que o serviço não estará disponível, 100%. Em vez disso, garante que o sistema estará disponível maior que um determinado número, como 99,5%.

A QoS depende de dois tipos de SLA: o **SLA da Aplicação**, a qual é um contrato entre o cliente (proprietário da aplicação) e os usuários finais; e o **SLA do Recurso**, acordado entre o provedor e a organização usuária. **O SLA da aplicação** é um certo tempo de resposta e o **SLA do recurso** é o % de disponibilidade da infraestrutura. Na maioria das vezes os SLAs são misturados, pois para satisfazer o SLA da aplicação é necessário que o provedor cumpra o SLA do recurso.

Os aplicativos hospedados em ambientes de nuvem podem ser de natureza diversa: trabalhos em lote, tarefas de redução de mapa, jogos, aplicativos da web, serviços de streaming de vídeo e muito mais. A alocação de recursos para aplicativos em lote é geralmente indicada como agendamento e envolve o cumprimento de um determinado prazo de execução do trabalho. Entretanto, é fundamental ressaltar que não se poderá ter um acordo – SLA - engessado, pois inúmeras mudanças poderão acontecer. O acordo deverá ser dinâmico e aberto às novas rotinas que poderão acontecer.

2.2.4 A Migração para nuvem

Considera-se que a estrutura em nuvem será utilizada por diversas empresas, não importando seu porte. Alguns pontos deverão ser observados nesse processo:

a) **análise os benefícios** - analisar as vantagens da migração, e quais serão os benefícios concretos;
b) **redução de custos** - principalmente as relativas à infraestrutura. Não será necessário espaço físico para os servidores. Os custos da infraestrutura, manutenção e compra de equipamentos serão eliminados;
c) **utilização de ambiente colaborativo** - as estruturas em nuvem têm plataformas prontas para o ambiente colaborativo. Será uma grande facilidade compartilhar, editar e criar documentos, projetos e reuniões;
d) **recorrer à mobilidade** - as informações poderão ser acessadas a partir de qualquer dispositivo - desktop, tablet, smartphone, laptop. Essa mobilidade é um benefício para empresas que contarem com colaboradores em *home office* ou que trabalhem em campo;

e) **realização do planejamento do processo** - migrar para um ambiente de nuvem é uma tarefa complexa. Isso requer uma reflexão inicial e um planejamento de modo a garantir o uso eficaz de recursos, o gerenciamento de riscos, a implementação no prazo e dentro do orçamento e, em última análise, o sucesso operacional. O problema é agravado se a equipe encarregada da migração não tiver experiência na elaboração de projetos complexos de TI.

Antes de iniciar o processo de migração será necessário ter uma estratégia bem definida para que tudo ocorra rápido, tranquilo e seguro. Nesse processo deverão ser identificados e avaliados os aplicativos, a infraestrutura local, os dados e mapeadas as dependências entre os aplicados. Assim, será possível definir prioridades durante o processo.

Deve-se avaliar, ainda, se todos os recursos utilizados serão necessários, evitando-se a migração desnecessária. Deve-se estabelecer prioridade nos processos, estabelecendo-se a ordem de migração. Com um planejamento ideal, todo o processo ocorrerá da forma mais tranquila possível e sem sobressaltos. É importante migrar os softwares que são totalmente independentes e que não requerem dados ou aplicações de outros programas. Defina o que será migrado para a nuvem. É fundamental definir tudo o que será migrado para a nuvem para ter noção do espaço que será necessário no novo servidor. Faça um amplo levantamento de todos os dados e arquivos da empresa e verifique se eles ainda precisam ser acessados rapidamente. Caso contrário, os mantenha em um servidor local apenas para fazer uma consulta esporádica, quando for necessário.

Todo o processo de migração terá que ser analisado para evitar danos a aplicativos ou dados. Para isso será recomendado analisar a seguinte abordagem de migração:

a) **rehost** – também conhecida como *lift and shift* - consiste na migração dos programas de forma rápida e segura. A transferência de dados para a nuvem ocorre como eles estão no servidor da sua empresa —processo recomendado quando houver urgência na transferência dos softwares para a nuvem;
b) **refatorar** - é um processo em que os programas sofrem algum tipo de alteração, mas sem mudanças ao seu código. Ela é indicada quando for necessário usar uma base de código já existente;
c) **rearquitetar** - essa função acontece quando for necessário fazer mudanças no software a ser migrado, seja para melhorar, modificar ou estender seu código, buscando otimização e melhoria de seu funcionamento;
d) **reconstruir** - nesse sistema de migração, será feita a recompilação do software,

usando as tecnologias que passarão a existir na nuvem.

Devido à migração envolver transferência de dados do servidor local para a nuvem, será necessário estar atento aos riscos que o processo poderá resultar. A recomendação inicial é fazer um backup completo, evitando a perda de arquivos durante o processo. Outro ponto é definir **regras rígidas de acesso** aos arquivos na nuvem, determinando o nível de acesso e o tipo de interação para cada arquivo (apenas leitura, edição parcial ou total).

Para realizar a migração para nuvem é imprescindível que a solução encontrada contemple o processo de backup. Esse processo precisa ser recorrente e sua restauração deve ser ágil, em caso de falha. Verificar a segurança da estrutura - Questionar a segurança dos dados. Uma nuvem segura precisa ter: dados criptografados, autenticação dupla, atualização constante e automática para as novas ameaças, concordância com o *compliance*.

2.2.4.1 Preparação para a migração

Os CIOs devem ter uma ideia clara de quanto custará a preparação para a migração, posto que consumirá um tempo significativo da equipe. Essa equipe geralmente se concentra em gerentes de serviços ou aplicativos e em equipes de sistemas associados. A equipe irá necessitar de recursos de armazenamento e de engenharia de rede e segurança. Dependendo do aplicativo, a migração poderá exigir a entrada de especialistas em integração, equipes de gerenciamento de risco e times de desenvolvimento.

Além do tempo da equipe, os CIOs devem estar preparados para fazer um orçamento para os seguintes custos de migração:

a) contratação ou desenvolvimento de pessoal para preparar funcionários ou serviços profissionais. Eles irão ajudar a mover os sistemas, se os recursos de pessoal forem insuficientes (quantidades e habilidades);
b) ferramentas de avaliação para identificar interdependências de carga de trabalho;
c) instrumentos de avaliação e provisionamento para determinar como provisionar os componentes de computação, armazenamento, rede e segurança para a carga de trabalho no ambiente de nuvem;
d) ferramentas de gerenciamento mais capazes de apresentar informações de situação e solução de problemas para as operações em andamento.
e) mudanças nos custos do aplicativo e da plataforma com base na hospedagem na nuvem. Os sistemas operacionais, bancos de dados e outros *middleware* e

aplicativos empacotados podem ter licenciamento diferentes quando executados na nuvem ou local. Ou seja, poderão exigir mais ou menos licenças para funcionar na nuvem;

f) adicionar rede de acesso à nuvem do DC para o ambiente de destino. Por exemplo, conexão direta ou uma conexão por meio de uma troca na nuvem. Contudo, isso deve ser feito antes do início da migração, se o período de migração se estender por meses, ou se houver qualquer licença significativa.

2.2.4.2 Da migração para a nuvem

A migração em si também consumirá tempo da equipe, das mesmas pessoas que fazem o planejamento, seja para realizar a migração para a nuvem ou, após o fato, para verificar se a migração foi bem-sucedida. Contudo, a verificação pós-movimentação será mais demorada nas primeiras vezes.

Entretanto, enquanto a equipe ganha experiência e a otimização das configurações é concluída, torna-se mais rara a necessidade de usar engenheiros de rede para verificar se a carga de trabalho migrada está sendo executada em um ambiente configurado corretamente.

Contudo, além do tempo da equipe, os CIOs devem planejar um orçamento para os seguintes custos de migração:

a) gastar dinheiro em sistemas em dois lugares simultaneamente. Por exemplo: Pagar o custo de uma plataforma para um grande servidor de aplicativos, enquanto os aplicativos estão migram para fora dele;

b) manter os gastos com continuidade de negócios e recuperação de desastres (*Business Continuity and Disaster Recovery*-BCDR) para sistemas locais até bem após a produção mudar para a nuvem;

c) possivelmente precisando de serviços profissionais ou novo software para gerenciar migrações reais.

2.2.4.3 Pós-migração

Em média, as empresas gastam cerca de 12% a mais para executar uma carga de trabalho em IaaS do que para executá-la em seu próprio DC. Essa média engloba tanto as cargas de trabalho nas quais eles conseguem grande economia quanto aquelas nas quais os custos dobram ou triplicam quando comparados ao custo de prestação do próprio serviço.

Portanto, quanto mais trabalho é levantado e alterado sem modificação, provável é que o CIO tenha de fazer um orçamento para aumentar custos.

Contudo, há uma variedade de ferramentas para estimar os custos de computação, armazenamento e rede associados a um ambiente de nuvem planejado. Elas estão disponíveis nos principais provedores de IaaS, bem como em corretores de serviços de nuvem terceirizados e fornecedores de gerenciamento de custos de nuvem. Para saber se deve ser esperado gastar mais ou menos nos custos de migração para a nuvem, os CIOs devem levantar uma informação fundamental: quanto gastam agora para executar uma determinada carga de trabalho e quanto esse custo diminuirá quando a carga de trabalho for movida.

2.2.5 Indicadores de desempenho dos serviços em nuvem

Em decorrência da computação em nuvem se tornar um elemento comum para empresas competitivas, surgiu a preocupação em como avaliar o sucesso na migração para a nuvem. Indicadores de desempenho são essenciais para avaliar se esta atuação está sendo satisfatória.

Há diferentes tipos de *Key Performance Indicator* (KPI) que atendem às diversas áreas de atuação da empresa. Um KPI é uma forma de medir o sucesso de uma ação. Pode ser único ou composto de diversas iniciativas. Eles são utilizados para medir a efetividade de uma prática. Os principais tipos são:

a) **Econômicos**, como custo de processos, rentabilidade, despesas;
b) **Financeiros**, como Retorno sobre Investimento (ROI);
c) **Logísticos**, como quantidade de produtos ou serviços disponíveis;
d) **Produção**, como custo de produção, tempo gasto na produção;
e) **Clientes**, como quantidade de clientes, quantidade de clientes novos e perdidos.

Os KPIs mais utilizados para a análise da migração para a nuvem são:

a) Custos totais de migração - é possível listar gastos, como recursos utilizados; gastos do tipo de serviço contratado (SaaS, IaaS, PaaS, Daas…); integração e monitoramento.

b) Duração - lida com a duração de cada fase da migração, o tempo para teste da migração dos dados e da migração do aplicativo e os resultados da migração.

c) **Quantidade de interrupções** – avalia a gravidade de problemas: disponibilidade de serviços críticos; tempo de inatividade de serviços; degradação do serviço pela inatividade.
d) **Avaliabilidade de infraestrutura** - agregam indicadores de infraestrutura: porcentagem de uso da CPU e da memória; latência da internet; desempenho de processamento.
e) **Desempenho de aplicativos** - o uso de aplicativos na nuvem também é avaliado, sendo medido por: taxa de erros, como requisições, falhas de usuário e total de requisições; Avaliabilidade do aplicativo, latência do aplicativo, taxa de transferência.
f) **Visibilidade da nuvem** – este indicador é uma medição que abrange KPIs de custo, uso, desempenho, segurança, disponibilidade etc. Eles são: variação do orçamento em relação ao real orçamento por aplicativo ou equipe, precisão da previsão, incidente de segurança por mês por equipe, vulnerabilidades de segurança identificadas por mês, tempo médio para vulnerabilidade anunciada.
g) **Otimização da nuvem** - organização da nuvem. Para isso, há KPIs específicos para otimização, como: porcentagem de infraestrutura em execução on-demand, economia de redimensionamento (uso sob medida da nuvem), custo efetivo por recurso (por hora), tempo médio para reparação de falhas, tempo médio entre falhas, número de falhas de segurança, número de recursos fora dos padrões de configuração.
h) **Governança da TI** - processo de definir as melhores práticas e de notificar infraestrutura fora de conformidade. Ele pode ser financeiro, operacional ou de segurança e conformidade. Para cada um deles, há diferentes tipos de KPIs: custo otimizado ao longo do tempo, porcentagem de políticas em estado de compatibilidade com a nuvem, tempo economizado como resultado das políticas, tempo para corrigir violações de segurança, porcentagem de serviço disponível, tempo para implementação de políticas.
i) **Integração com os negócios** - após a migração para nuvem e definição das melhores práticas, deve-se otimizar a nuvem com foco nos negócios. Trata-se da maturação dos serviços de nuvem, de forma que ele funcione em prol da empresa. Alguns indicadores para essa etapa são: custo por cliente, gastos na nuvem como porcentagem de receita, redução no CPV (custo dos produtos vendidos) por tempo, custo da receita por tempo, tempo para levar novos

serviços ao mercado, satisfação do cliente, onde é utilizado o Net Promoter Score (NPS*).*

2.3 Provisionamento de recursos em nuvem

O **Gerenciamento de Recursos em Nuvem** é uma importante área de pesquisa em decorrência do tempo e custo dos recursos envolvidos (KUMAR, 2018). Tem-se aqui duas subáreas de atuação: **o Provisionamento e o Agendamento de Recursos.**

Diferentes critérios e parâmetros de agendamento de recursos são direcionados para as diferentes categorias de *Resource Scheduling Algorithm* (RSAs) (Singh; Chana, 2016). O agendamento eficaz reduz o custo e o tempo de execução, o consumo de energia e considera outros requisitos de *Quality of Service* (*QoS*), como confiabilidade, segurança, disponibilidade e escalabilidade. O agendamento é classificado em dois níveis, como agendamento ao nível de usuário e ao nível de sistema.

Carga de Trabalho (*Workload*) é definida como todas as solicitações de entrada, enviadas por meio de interações *on-line* dos usuários finais, para os serviços em nuvem ou para trabalhos processados em lote.

Neste capítulo serão enfatizadas questões sobre a **carga de trabalho** no **Gerenciamento de Recursos** em computação em nuvem, destacando-se características do processo, tais como parâmetros, padrões, arquiteturas e modelos de solução utilizados. O objetivo é identificar:

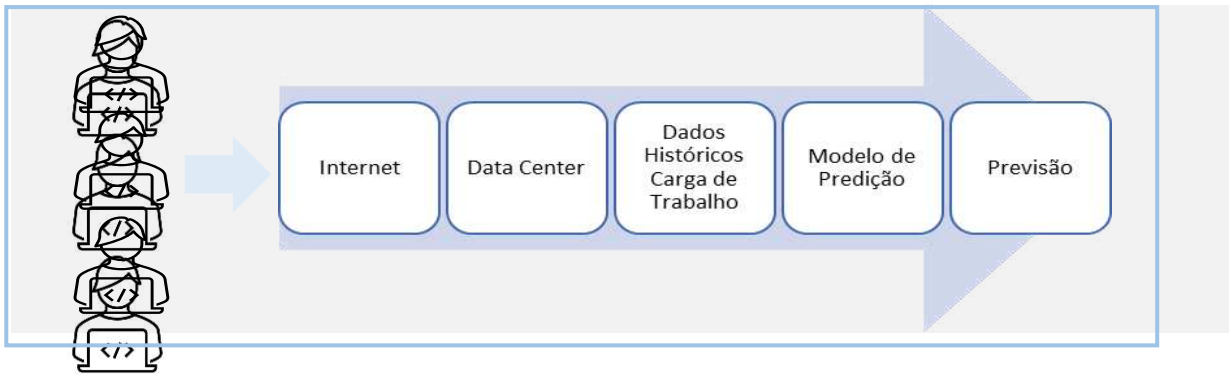
a) Quais as principais contribuições dos esquemas de previsão de carga de trabalho?
b) Quais algoritmos utilizados para prever, com precisão, a carga de trabalho?
c) Quais dados de carga de trabalho são aplicados em cada esquema?
d) Quais fatores e ambientes utilizados para avaliar a precisão e a eficácia de cada esquema?

2.3.1 O Processo de provisionamento

O provisionamento de recursos em nuvem, por meio da análise da carga de trabalho atual, é uma estratégia adotada para prover eficiência e redução do custo operacional dos serviços em nuvem. O provisionamento utiliza as informações disponíveis no presente para prever o futuro. Enquanto o gerenciamento de recursos é o requisito que motiva a previsão.

A Figura 3 ilustra um cenário geral de provisionamento de carga de trabalho em um *data center* em nuvem. Os servidores recebem milhões de solicitações enviadas pelos usuários. Essas requisições são processadas e registradas como dados históricos. Esses dados são preparados e utilizados por um sistema de predição, para serem posteriormente utilizados para prever a carga de trabalho futura.

Figura 3 - Modelo Geral de Provisionamento



Fonte: Yadav, (2022).

2.3.2 Padrões de carga de trabalho

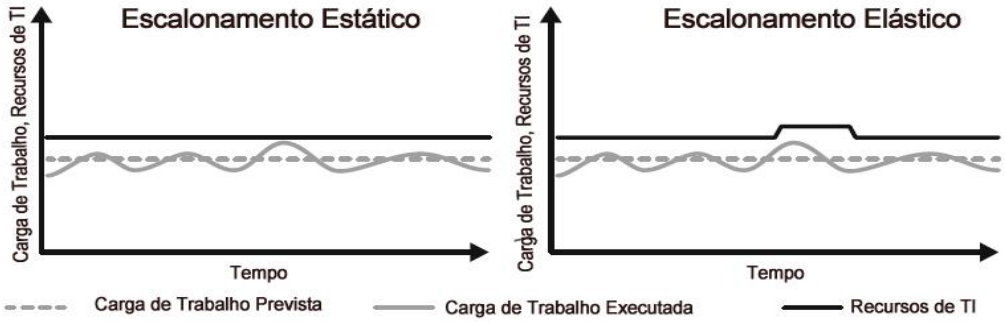
Como atributo de uma carga de trabalho, o **histórico** é essencial para sua previsão. A carga de trabalho possui ainda atributos que descrevem seu comportamento. Eles podem ser expressos em termos de tipo de recurso, dependências e quantidade de consumo. Alguns atributos comuns são localização geográfica, tipo de recurso, largura da banda de rede e segurança. A carga de trabalho é medida em termos do número de solicitações atribuídas ou executadas por máquina (servidor ou estação), em um determinado intervalo de tempo.

Tarefa e **Usuário** são atributos importantes para a previsão de carga de trabalho. A **Tarefa** define o tipo de computação e o volume a ser atribuído. O **Usuário** tem a responsabilidade de criar a configuração do sistema necessária para a computação. Ele também é responsável pela análise das várias características de configuração do sistema. Os aplicativos implantados em nuvem são *Multitenancy* por natureza e, inevitavelmente, operam sob mudanças dinâmicas na carga de trabalho. Com base na **carga de trabalho**, uma previsão deve ser estabelecida para o dimensionamento de recursos. O padrão de carga de trabalho altamente flutuante e imprevisível afeta negativamente os aplicativos em execução.

Segundo Fehling *et al.* (2014)**,** padrões de carga de trabalho em nuvem podem ser categorizados em: estática, periódica, “uma vez na vida”, imprevisível, e de “mudança contínua”. As características desses padrões serão sintetizadas a seguir:

a) **Carga de Trabalho Estática** - A carga de trabalho prevista é sempre a mesma da carga executada. Não ocorre mudança na demanda. Este tipo de carga não tira proveito de ambiente pago por uso proposto pela computação em nuvem. A Figura 4 apresenta este padrão.

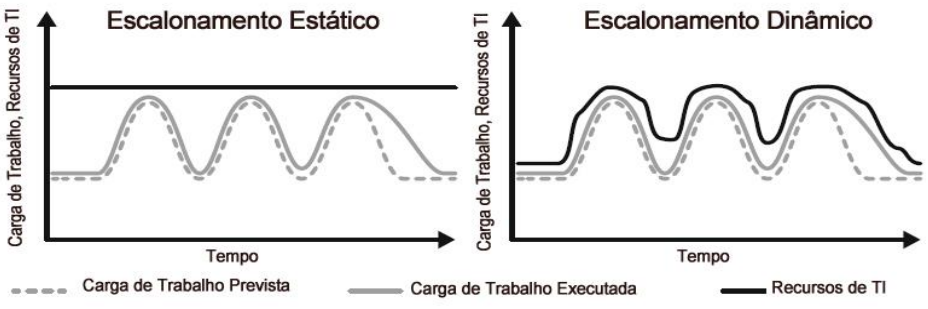
Figura 4 - Carga de Trabalho Estática



Fonte: Fehling *et al.* (2014).

b) **Carga de Trabalho Periódica -** Considera-se que este é o perfil de aplicação mais presente no dia a dia. Têm-se as horas de *rush* ou os processamentos mensais de determinadas aplicações. São sistemas que ficam adormecidos por grande parte do tempo e, quando necessário, usam processamento, otimizando custos de utilização. A Figura 5 apresenta o padrão de carga de trabalho periódica.

Figura 5 - Carga de Trabalho Periódica

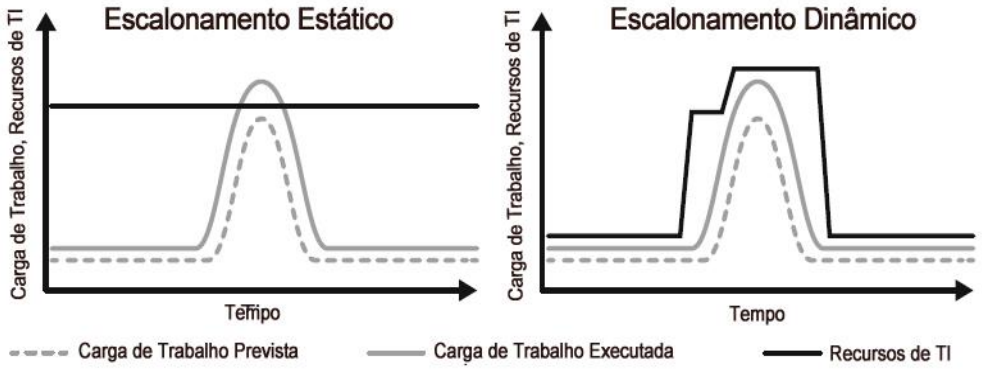


Fonte: Fehling *et al.* (2014).

c) **Carga de Trabalho "Uma vez na vida" -** Perfil derivado da carga de trabalho periódica. A repetição de picos não ocorre frequentemente, mas apenas em eventos conhecidos (e.g. Black Friday). A preparação de recursos é planejada e

muitas vezes com alocação de recursos da nuvem de forma manual e dirigida. A Figura 6 apresenta esse padrão de carga de trabalho.

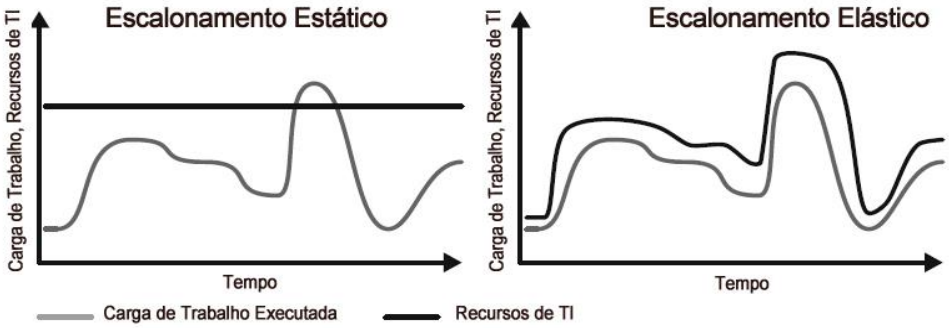
Figura 6 - Carga de Trabalho “Uma vez na vida



Fonte: Fehling *et al.* (2014).

d) **Carga de Trabalho Imprevisível** - É um tipo de pico demandado por situações não previstas de processamento (e.g.: uso de recurso computacional para prevenir ataques de “força bruta” a websites). Existe um comissionamento e descomissionamento de recursos de forma mais automatizada e com alerta para apontar anomalias. A Figura 7 apresenta esse padrão de carga de trabalho.

Figura 7 - Carga de Trabalho Imprevisível

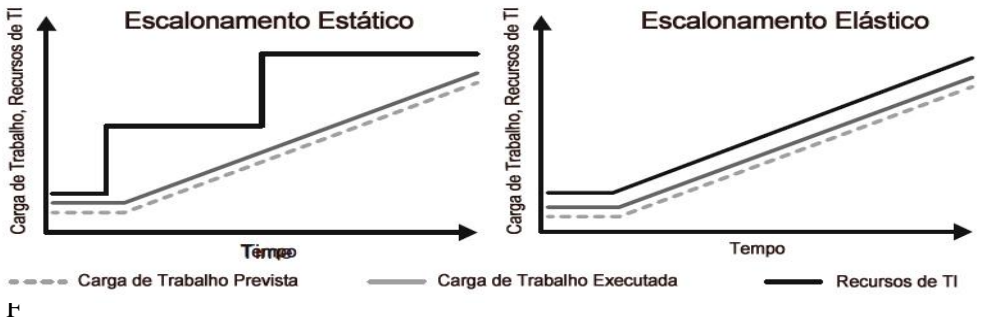


Fonte: Fehling *et al.* (2014).

e) **Carga de Trabalho de Mudança Contínua** - É o tipo que será funcionalmente transformado em outra aplicação ou crescerá conforme a demanda de utilização. O alinhamento do previsto em relação ao executado deve apontar melhorias de otimização ou até mesmo se a estratégia de descomissionamento de funcionalidades é viável (e.g.: “desmontar” uma

aplicação pode criar mais complexidade em novas cargas de trabalho que o previsto). A Figura 8 apresenta esse padrão de carga de trabalho.

Figura 8 - Carga de Trabalho de Mudança Contínua



Fonte: Fehling *et al.* (2014).

Para o entendimento sobre qual tipo de arquitetura em nuvem trará benefício, deve-se considerar aspectos das aplicações e de suas características lógicas de construção. Alguns aspectos podem direcionar essas questões, quais sejam:

a) **Capacidade e estilo de processamento** – identificar qual modelo de processamento será mais adequado para obter-se desempenho e otimização. A maioria dos CSPs oferece recursos com características distintas. Importante considerar que o conhecimento desta arquitetura previne situações de alto ou baixo uso, além do desempenho da aplicação. Destacam-se dois temas: centro **de gravidade** e **isolamento de processamento**.

O **centro de gravidade** aponta o melhor perfil de onde se deve processar a carga de trabalho. Os tipos de carga de trabalho ajudam nesta análise. Algumas perguntas são pertinentes para definir a melhor escolha:

- O processamento está ligado ao banco de dados ou dependente de armazenamento?
- O processamento intensivo é por CPU?
- Existem cargas paralelas de trabalho?
- Deve existir proximidade da camada de plataforma de aplicação com o banco de dados?

O **isolamento de processamento** determina quão *multitenant* ou *single-tenant* será a arquitetura para aplicação. Existem opções de isolamento que habilitam uma ida para a nuvem. Mesmo assim, não deveria ser uma questão muito crítica, pois os CSPs têm disponibilizado esta camada sem expor a

complexidade, mesmo no modelo em *IaaS* (*Infrastructure as a Service*).

b) **Dependência da carga de outras aplicações** - deve-se considerar como uma aplicação implica na carga da outra. É comum aplicações dependerem entre si. Independentemente do formato, ocorrerá influência no comportamento da carga de trabalho. O que muda num *deployment* em nuvem é um aspecto de latência e banda de dados. Em certas situações, nem toda parte da aplicação reside num modelo de nuvem pública. Pode-se ter uma parte num *tenant* em nuvem pública e outro ainda *on-premise* (ou outro CSP). Arquiteturas de *networking* considerando características dos *DCs* existentes e dos CSPs devem ser consideradas para análise da resiliência e do desempenho da aplicação.

c) **Exposição das aplicações** - outro fator importante é entender como as aplicações se expõem, conforme o nível de segurança requerido. Ao mover para outra arquitetura é importante estabelecer uma correspondência técnica e funcional de elementos existentes no *DC* da empresa e o existente no CSP. Este tipo de preocupação é essencial e o correto entendimento das equipes dessas capacidades do CSP ajuda a mitigar riscos de segurança e imprevistos de comportamento de carga de trabalho.

d) **Ambientes das aplicações** - um tema relacionado com automação é levar os ambientes distintos das aplicações para a nuvem. Os ambientes de desenvolvimento e de testes não necessitam estar ativos o tempo todo, diferentemente do ambiente de produção. Neste caso, o time de operações deve implementar práticas e arquitetar formatos para esse processo, conforme as necessidades das equipes.

2.3.3 Modalidades de provisionamento

O dimensionamento da aplicação deve ser implementado sem esforço, adaptando os recursos atribuídos à aplicação à demanda inicial, fornecida pelo usuário.

O principal problema é como alocar um certo quantitativo de recursos, com base no “pagamento conforme o uso”. A identificação da quantidade de recursos a serem alocados para atender ao SLA estabelecido, mantendo o custo geral baixo, não é uma tarefa fácil. Muitas técnicas foram propostas para automatizar o dimensionamento de aplicativos.

O provisionamento de recursos em nuvem, geralmente, acontece nas modalidades de operação: **reativa ou proativa**. Esta classificação nem sempre deixa claro se uma

determinada abordagem é puramente reativa ou proativa. A **abordagem reativa** consiste em uma reação programável às mudanças percebidas na aplicação/infraestrutura, mas não as antecipa. A **abordagem proativa ou preditiva** busca antecipar a carga da aplicação para tomar decisões sobre a capacidade adequada. Antecipam demandas futuras e tomam decisões levando-as em consideração (Lorido-Botran *et al.* 2014).

2.3.3.1 Abordagens reativas

Essa abordagem se tornou popular devido à sua aparente simplicidade. Utiliza informações sobre o estado atual da aplicação e do ambiente, para decidir o provisionamento. Considera um conjunto de regras para decidir quando e em qual quantidade a aplicação deverá ser provisionada. O sistema reagirá a mudanças na carga de trabalho somente quando essas mudanças forem detectadas, utilizando os últimos valores obtidos do conjunto de variáveis monitoradas; consequentemente, como o provisionamento de recursos leva tempo, o efeito desejado pode chegar tarde demais. Devido à complexidade do modelo em nuvem, as abordagens reativas foram forçadas a suporem limitantes nas condições / requisitos de operação ou nos padrões de carga de trabalho esperados (Lorido-Botran *et al.* 2014).

Os CSPs oferecem dimensionamento **automático** puramente reativo, usando **limites baseados em regras**. As decisões de dimensionamento são acionadas com base em algumas métricas de desempenho e limites predefinidos. Os **autos escaladores baseados em regras** são fáceis de fornecer como um serviço em nuvem e são fáceis de configurar pelos clientes. Entretanto, a eficácia das regras em cargas de trabalho é questionável.

2.3.3.2 Abordagens proativas ou preditivas

Muitas **abordagens preditivas** para o escalonamento de recursos foram propostas para superar as limitações das abordagens reativas convencionais. Como tal, é extremamente difícil para os usuários de serviços em nuvem identificar qual preditor funcionará melhor para sua atividade específica na nuvem, especialmente quando considerar padrões de carga de trabalho altamente variáveis, modelos de faturamento não triviais, variedade de recursos para adicionar / subtrair etc.

A ideia da predição fundamenta-se na análise do comportamento futuro de uma aplicação, baseada em aspectos capturados em momentos anteriores. Segundo Amiri; Mohammad-Khanli (2017), o provisionamento de recursos pode ser realizado sob diferentes dimensões, conforme apresentado na Figura 9.

Figura 9 - Diferentes dimensões do provisionamento de recursos

Fonte: Amiri; Mohammad-Khanli (2017).

O dimensionamento de recursos requer que o sistema em nuvem retenha o QoS, segundo o SLA estabelecido. Conforme a demanda futura do aplicativo, o provisionamento eficiente de recursos deve detectar a quantidade mínima de recursos para atender aos parâmetros de QoS, como utilização da CPU, tempo de resposta, disponibilidade, confiabilidade e segurança. No Quadro 1 tem-se o registro do QoS por tipo de aplicação, conforme apresentado em Singh; Chana (2016a).

Quadro 1 - Aplicativos em nuvem e seus requisitos de *Quality of Service.*

Aplicativos	**Requisitos de QoS**
Web sites	Armazenamento confiável, alta largura de banda de rede, alta disponibilidade.
Computação Tecnológica	Capacidade de computação, armazenamento confiável
Endeavour software	Segurança, alta disponibilidade, cliente, nível de confiança, correção.
Teste de desempenho	Tempo de execução, consumo de energia e custo de execução.
Processamento de transações on-line	Segurança, alta disponibilidade, acessibilidade à internet, usabilidade.
Serviços financeiros centrais	Segurança, alta disponibilidade, mutabilidade, integridade.
Serviços de armazenamento e	Confiabilidade, persistência.

backup	
Aplicativos de produtividade	Largura de banda de rede, latência, backup de dados, segurança.
Desenvolvimento e testes de Software / Projeto	Taxa de autoatendimento do usuário, flexibilidade, grupo de criativos de serviços de infraestrutura, tempo de teste.
Gráficos orientados	Largura de banda de rede, latência, backup de dados, visibilidade.
Aplicativos críticos da Internet	Alta disponibilidade, capacidade de manutenção, usabilidade.
Serviços de computação móvel	Alta disponibilidade, confiabilidade, portabilidade.

Fonte: Singh; Chana (2016a).

As vantagens da **escalabilidade automática** estão relacionadas ao esforço necessário para manter os aplicativos em funcionamento. Os aplicativos tradicionais geralmente exigem monitoramento constante e possível reconfiguração quando o padrão de carga de trabalho é alterado. Isso acontece quando o número de clientes aumenta ou quando ocorre o comprometimento da eficiência dos recursos utilizados, em decorrência da execução de outros aplicativos.

Ao utilizar dados de monitoramento, os aplicativos autoescaláveis podem ser mais eficientes do que os seus equivalentes operados manualmente, devido ao menor tempo de reação. Isso é especialmente importante em ambientes, mudando dinamicamente quando o padrão de carga de trabalho não pode ser determinado ou previsto antecipadamente.

Construir um aplicativo autoescalável é desafiador. Essa funcionalidade é frequentemente implementada em um módulo separado, geralmente denominado **módulo de gerenciamento.** Este módulo é responsável por analisar a carga de trabalho da aplicação, com base nos dados de monitoramento e executar o procedimento de dimensionamento do aplicativo, iniciando uma nova instância do aplicativo em um servidor diferente. **O módulo de gerenciamento** implementa funcionalidades, tais como:

a) **Monitoramento on-line** - Como acontece com a administração humana, a aplicação requer dados on-line sobre a carga de trabalho atual no sistema para reagir a qualquer alteração.
b) **Detecção de eventos** - O módulo de gerenciamento deve identificar automaticamente momentos em que o aplicativo precisa ser dimensionado. Esses eventos podem diferir para diferentes tipos de aplicação.
c) **Escalonamento de Recursos -** está relacionado à aquisição de recursos adicionais pelo aplicativo. Este passo é também dependente da aplicação, e pode envolver a configuração de uma nova conexão de banco de dados ou a adição de uma nova máquina a uma rede privada virtual.

d) **Identificação de recursos** - envolve a identificação de recursos que podem ser utilizados durante o procedimento de dimensionamento. Na maior parte das vezes, esses recursos precisam ser preparados.

Além dessas funcionalidades, aplicativos para o **gerenciamento da escalabilidade** exigem conhecimento sobre eventos que devem acionar o escalonamento. Esse conhecimento pode estar na forma de regras, que definem condições em que o módulo de gerenciamento executará algumas ações. Este conhecimento é geralmente obtido a partir da observação da aplicação em cenários reais.

2.3.4 Predição de carga de trabalho

Várias técnicas para predição de carga de trabalho em nuvem foram apresentadas na literatura. Nesta seção serão relacionadas algumas delas. Buscou-se descrever as técnicas, bem como suas principais contribuições para os estudos sobre predição de carga de trabalho em nuvem.

2.3.4.1 Proposta de MASDARI, KHOSHNEVIS

O trabalho realizado por Masdari; Khoshnevis (2020) buscou identificar os seguintes pontos sobre o provisionamento de carga de trabalho:

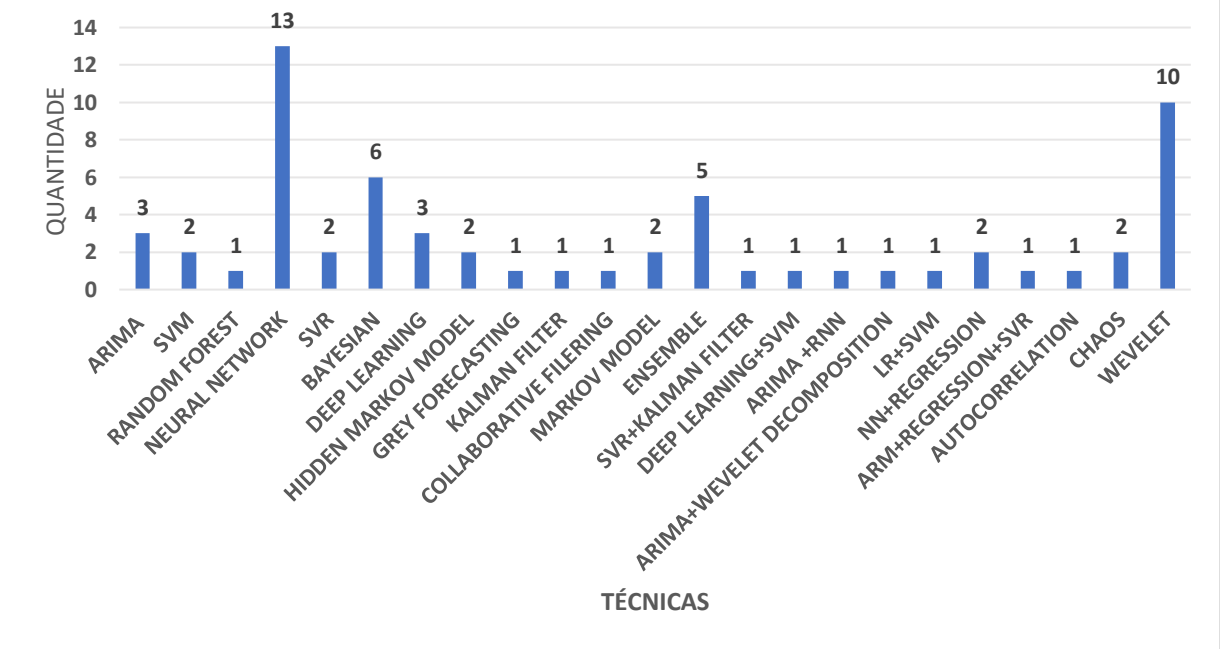
a) Quais as principais contribuições de cada esquema?
b) Quais algoritmos são utilizados para prever a carga de trabalho com precisão?
c) Quais dados de carga de trabalho são aplicados em cada esquema de previsão?
d) Quais ambientes são usados para avaliar cada esquema de previsão de carga de trabalho?
e) Quais fatores de avaliação são aplicados para identificar a precisão e a eficácia de cada esquema?
f) Quais recursos são previstos por cada esquema reconhecer a carga de trabalho incorrida?

Conforme apresentado no trabalho de Masdari; Khoshnevis (2020), apresentado na Figura 10, observou-se uma maior incidência de casos uso nas técnicas *Neural Network* e *Wavelet*.

Como resultante da pesquisa, as técnicas foram categorizadas em dez diferentes tipos. A técnica denominada **Híbrida** foi acrescentada quando constatada a existência de mais de uma técnica no experimento analisado. As categorias explicitadas foram: **baseadas em**

regressão, classificadores, processos estocásticos, *grey forecasting*, autocorrelação, caos, modelo de filtro de kalman, *wavelet*, filtros colaborativos, *ensemble* e híbridos. A Figura 10 apresenta uma síntese dessa categorização. No **Apêndice D – Trabalhos Anteriores** tem-se um maior detalhamento dos experimentos realizados em cada uma dessas categorias.

Figura 10 - Uso de Técnicas de Predição



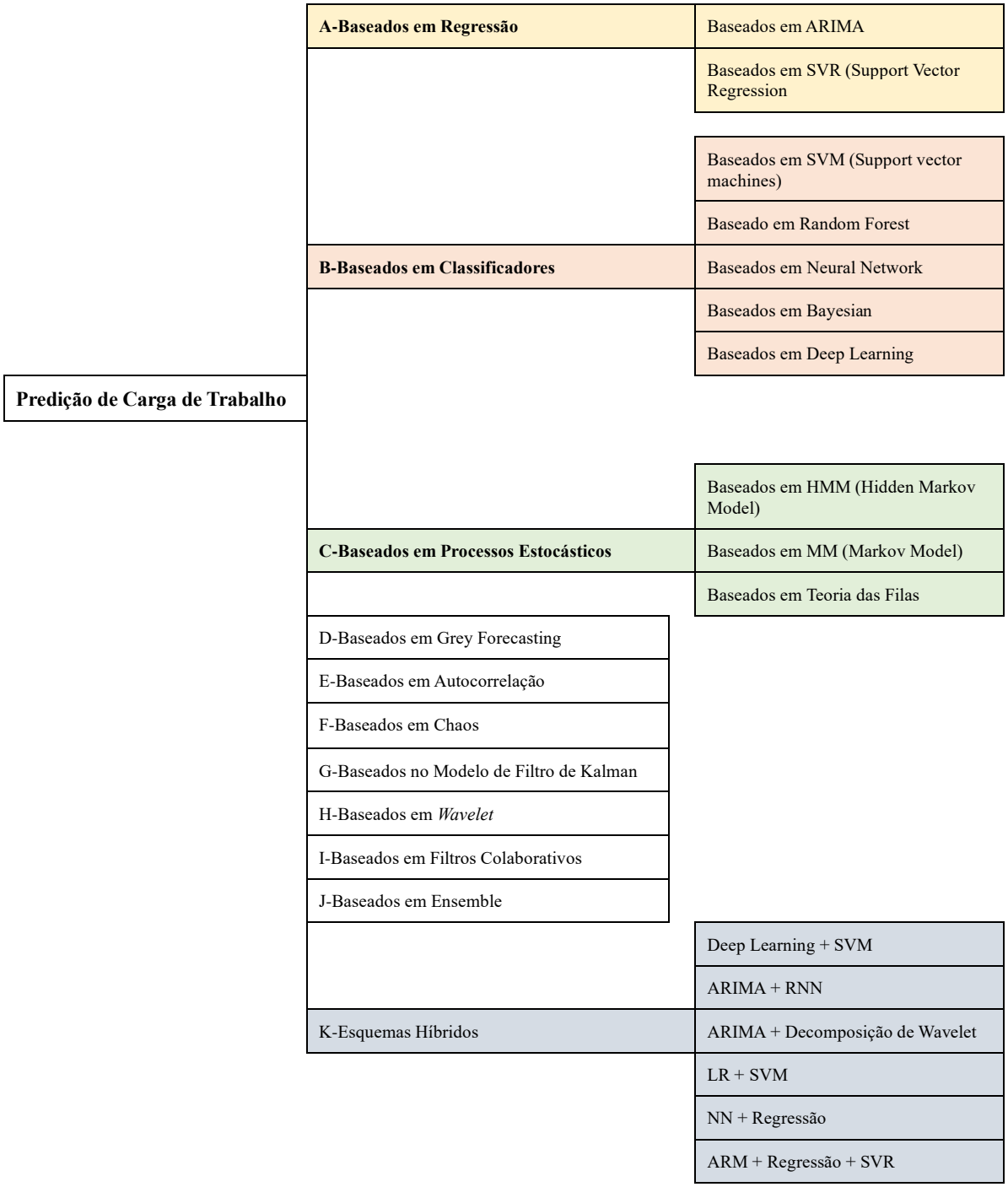
Fonte: Masdari; Khoshnevis (2020).

Masdari; Khoshnevis (2020**)** conclui seu trabalho alertando a necessidade da realização de maiores estudos. Dentre os pontos mencionadas destacam-se as seguintes questões que poderão ser mais investigadas:

a) Explorar outras técnicas de aprendizado de máquina para melhorar o desempenho da previsão de carga de trabalho.
b) Fornecer melhores esquemas de previsão de carga para reconhecer padrões de solicitação mais realistas e complexos.
c) Definir novas métricas de previsão de carga de trabalho, e.g.: atrasos nas previsões de intermitência. Além disso, como o custo dos erros de previsão no ambiente de nuvem não é simétrico, a definição de melhores métricas de avaliação deve ser considerada nessa questão.
d) Quanto à adequação dos modelos de previsão não lineares para prever séries temporais com variações sazonais, eles podem ser utilizados para otimizar processos com horizontes de tempo mais longos.

e) Investigar os algoritmos de gerenciamento de recursos para utilizar os resultados de previsão alcançados.

Figura 11 – Taxonomia de Predição de Carga em Computação em Nuvem



Fonte: Masdari; Khoshnevis (2020).

2.3.4.2 Proposta de Lorido-Botrán

A classificação proposta por Lorido-Botran *et al.* (2014) possui grande aceitação

entre pesquisadores. Os autores propõem uma classificação em 05 categorias, quais sejam: Limites baseados em regras - *Threshold-based rules* (rules), Aprendizado por reforço - *Reinforcement learning* (RL), Teoria das filas - *Queuing theory* (QT), Teoria de controle - *Control theory* (CT), Análise de séries temporais - *Time series analysis* (TS).

a) **Limites baseados em regras** – políticas de dimensionamento automático baseadas em limites são muito populares entre CSPs. A simplicidade e a natureza intuitiva dessas políticas as tornam atraentes para os clientes de nuvem. Definir os limites correspondentes é uma tarefa que requer uma profunda compreensão dos métodos orientados por tabela de tendências de carga de trabalho.

b) **Aprendizado por reforço** - abordagem automática de tomada de decisão utilizada para implementar auto escalonadores. Sem conhecimento anterior, as técnicas de RL podem determinar a melhor ação de dimensionamento a ser executada para cada estado do aplicativo, considerando a carga de trabalho de entrada. O RL se concentra no aprendizado por meio da interação direta entre um agente (o auto escalador) e seu ambiente. O auto escalador aprenderá com a experiência (método de tentativa e erro) a melhor ação de dimensionamento a ser tomada.

c) **Teoria das filas -** tem sido amplamente aplicada a sistemas de computação, para encontrar a relação entre as atividades que chegam e saem de um sistema. Uma abordagem simples consiste em modelar cada VM (ou conjunto de VMs) como uma fila de solicitações para estimar diferentes métricas de desempenho, como o tempo de resposta. Uma limitação principal dos modelos QT é que eles são muito rígidos e precisam ser recalculados quando há mudanças na aplicação ou na carga de trabalho.

d) **Teoria de controle -** objetiva controlar recursos compartilhados entre aplicativos em nuvem. Se o modelo controlar um recurso, por exemplo, CPU, um modelo SISO (Single Input Single Output) é usado. O modelo SISO correlaciona a saída à entrada. Em Liu et al. (2005), o modelo SISO mapeia o compartilhamento de CPU da aplicação ao inverso de seu tempo de resposta. Caso contrário, se o controlador operar em vários recursos, é utilizado um modelo Multi Input Multi Output (MIMO).

e) **Análise de séries temporais -** abrange uma ampla gama de métodos para detectar padrões e prever valores futuros em sequências de pontos de dados. A

precisão no valor da previsão (número futuro de solicitações ou utilização média da CPU) dependerá da correta seleção da técnica e da configuração dos parâmetros, especialmente a janela de histórico e o intervalo de previsão. A análise de séries temporais é o principal facilitador de técnicas proativas de autoescalonamento.

Nas categorias – *Queuing theory* (**QT**) e *Control theory* (**CT**) tem-se dois métodos de dimensionamento automático que dependem da modelagem do sistema para determinar as necessidades futuras de recursos. A categoria **QT** tem sido aplicada a sistemas de computação, a fim de encontrar a relação entre os empregos que chegam e saem de um sistema. Uma abordagem simples consiste em modelar cada **VM** (ou conjunto de VMs) como uma fila de requisições para estimar diferentes métricas de desempenho, como o tempo de resposta.

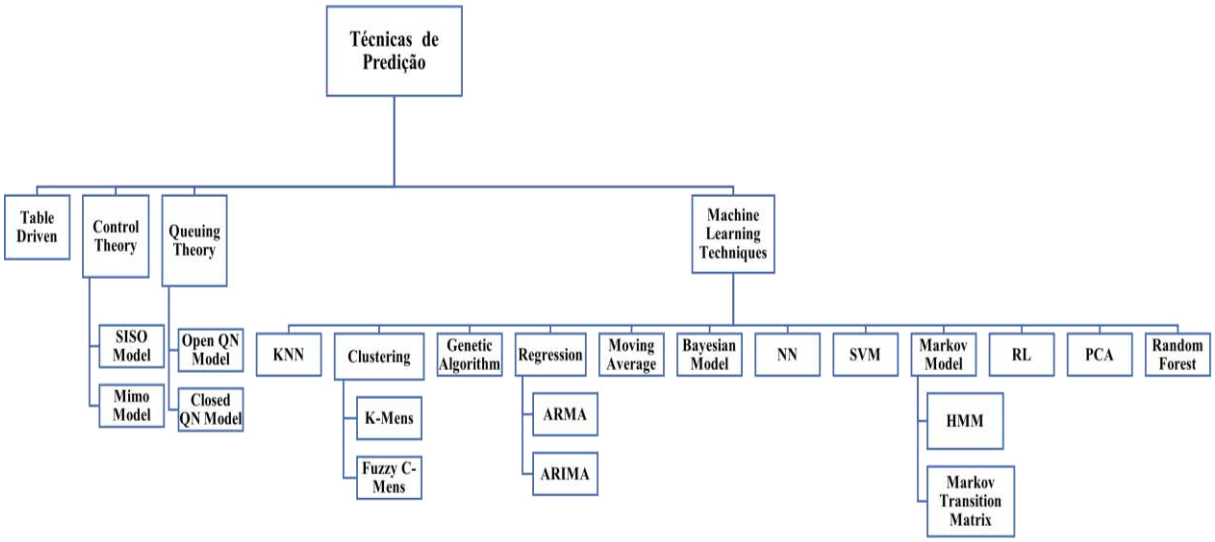
A principal limitação dos modelos **QT** é que eles são muito rígidos e precisam ser recalculados quando há mudanças na aplicação ou na carga de trabalho. A categoria **CT** também se baseia na criação de um modelo do aplicativo. O objetivo é definir um controlador (reativo ou proativo) para ajustar automaticamente os recursos necessários às demandas da aplicação. A natureza e o desempenho de um controlador dependem muito do modelo da aplicação e do próprio controlador. Muitos pesquisadores consideram que esse tipo de autoescalonamento tem um grande potencial, principalmente quando combinado com a previsão de recursos.

A categoria **RL** contém propostas baseadas no aprendizado por reforço. Da mesma forma que a teoria de controle, a **RL** tenta automatizar a tarefa de dimensionamento, mas sem usar nenhum conhecimento prévio ou modelo da aplicação. Em vez disso, a **RL** tenta aprender a ação mais adequada para cada estado específico, com uma abordagem de tentativa e erro. Embora a ausência de modelo e adaptabilidade da técnica possa parecer apelativa para o autoescalonamento, a **RL** sofre de longas fases de aprendizagem. O tempo necessário para o método convergir para uma política ótima pode ser inviável.

2.3.4.3 Proposta de Amiri e Mohammad-Khanli

Amiri; Mohammad-Khanli (2017) apresentaram o problema da predição de carga de trabalho em computação em nuvem também abordando diferentes estratégias. A Figura 12 apresenta o quadro de técnicas, onde são categorizados quatro grandes agrupamentos: **métodos orientados a tabelas**, **teoria de controle**, **teoria de filas** e **técnicas de aprendizado de máquina**, conforme os algoritmos aplicados no processo de previsão.

Figura 12 - Categorização de técnicas de predição



Fonte: Amiri; Mohammad-Khanli (2017).

2.3.4.4 Propostas comerciais

Dentre os módulos disponibilizados pelas plataformas de serviços em nuvem comerciais tem-se o **Elastic Beanstalk *(EB)*** da AWS. O *EB* utiliza os módulos **Amazon EC2,** o **Amazon Elastic Container Service (Amazon ECS),** o **Auto Scaling** e o **Elastic Load Balancing.** Esses módulos trabalham para oferecer suporte a aplicativos que precisam de escalabilidade para atender a um número elevado de usuários. O **Elastic Beanstalk *(EB)*** - suporta aplicativos escritos em linguagens e *frameworks* populares. O desenvolvedor faz o *upload* do código e o serviço processa automaticamente todos os detalhes, como provisionamento de recursos, balanceamento de carga, escalabilidade automática e monitoramento. O código é recebido por meio da console do gerenciamento ou da interface em linha de comando do próprio *EB*. As opções de política permitem a escolha entre a velocidade e a segurança da implantação dos aplicativos, enquanto reduz a carga administrativa. Para monitorar e gerenciar a integridade dos aplicativos é fornecida uma interface unificada para a qual são coletadas métricas e atributos essenciais para determinar esta integridade. O Elastic Beanstalk Health Dashboard oferece uma interface que permite a visualização da integridade geral do aplicativo e a personalização das verificações desejadas. O *EB* é integrado ao **Amazon CloudWatch** e ao **AWS X-Ray**. Pode-se aproveitar o painel de monitoramento para observar-se as principais métricas de desempenho. Também é possível configurar os alarmes do **CloudWatch** quando as métricas excederem os limites escolhidos. O *EB* utiliza o **Elastic Load Balancing** e o **Auto Scaling** para escalonar automaticamente o

aplicativo, com base em suas necessidades específicas. Com o *EB* é possível selecionar o tipo de instância do **Amazon EC2**, que seja ideal para o aplicativo. Permite ainda a obtenção de acesso e manutenção do controle total sobre os recursos da **AWS** que capacitam o aplicativo. O ***EC2 Auto Scaling*** - permite a inclusão de novas instâncias de forma contínua e automática quando a demanda aumenta, e encerra automaticamente as instâncias desnecessárias. Efetua o escalonamento dinamicamente com base em métricas do **Amazon CloudWatch** (AWS, 2021) ou de forma previsível de acordo com sua própria programação. Recebe notificações pelo **Amazon Simple Notification Service (Amazon SNS)** sobre o uso de alarmes do **Amazon CloudWatch** para iniciar ações do **Amazon EC2 Auto Scaling** ou quando o **EC2 Auto Scaling** completa uma ação. O **Amazon EC2 Auto Scaling** substitui automaticamente instâncias não íntegras ou inacessíveis para manter uma maior disponibilidade dos aplicativos. Para automatizar o gerenciamento de frota para instâncias, o **EC2 Auto Scaling** monitora a integridade das instâncias em execução, substituindo automaticamente as instâncias com falhas e equilibrando a capacidade nas zonas de disponibilidade.

A **escalabilidade preditiva** da **AWS** prevê o tráfego futuro, inclusive picos de ocorrência regular, e fornece o número adequado de instâncias do EC2, antes das ocorrências previstas acontecerem. Os algoritmos de *Machine Learning* da escalabilidade preditiva **detectam mudanças nos padrões diários e semanais**, ajustando automaticamente suas previsões. Isso elimina a necessidade de ajuste manual dos parâmetros de *Auto Scaling*, facilitando sua configuração. O **Amazon Auto Scaling** com escalabilidade preditiva proporciona um provisionamento de capacidade rápido, simples e preciso, gerando custos mais baixos e aplicativos com maior capacidade de resposta (Barr, 2021).

O *Amazon EC2 Auto Scaling* permite provisionar e escalonar automaticamente instâncias de diversas opções de compra, zonas de disponibilidade e famílias de instâncias em um único aplicativo para otimizar escala, desempenho e custo. Para economizar recursos computacionais podem ser incluídas instâncias *spot* com instâncias sob demanda e reservadas em um único grupo de *Auto Scaling* (ASG).

A **Google Cloud Platform (GCP)** oferece recursos de escalonamento automático que permite adicionar ou excluir automaticamente instâncias de um grupo, com base no aumento ou diminuição da carga. O escalonamento automático da GCP ajuda o aplicativo a lidar com o aumento do tráfego e reduzir o custo, quando a necessidade de recursos é menor. O **escalonamento automático** utiliza os seguintes conceitos:

a) **Modelos de instância** - utilizado para criar instâncias de VM e grupos de instâncias gerenciadas. Especifica o tipo de máquina, imagem do disco de

inicialização ou do contêiner, rótulos e outras propriedades da instância.

b) **Grupos de instâncias gerenciadas -** grupo de instâncias homogêneas, criadas a partir de um modelo de instância. Um autoescalador adiciona ou remove instâncias de um grupo de instâncias gerenciadas com base na política de escalonamento. Embora **o GCP Compute Engine** tenha grupos de instâncias gerenciadas e não gerenciadas, apenas grupos gerenciados podem ser utilizados para o escalonamento automático do **Google Cloud**.

c) Política de escalonamento automático e utilização de destino - Para criar um autoescalador é preciso especificar a **política de escalonamento automático**, bem como um nível de utilização de destino, que o autoescalador determine quando dimensionar o grupo. Pode-se optar por dimensionar usando as seguintes políticas:

d) Utilização média da CPU

e) Capacidade de serviço de balanceamento de carga HTTP - pode ser baseada em utilização ou solicitações por segundo.

f) Métricas do *Stack driver Monitoring* - O autoescalador coleta informações com base na política. Em seguida, compara com a utilização de destino desejada e determina se precisará executar o dimensionamento.

A GCP oferece os seguintes tipos de escalonamento**:**

a) baseado na utilização da CPU. Define a utilização de CPU de destino que o autoescalador deve manter e o autoescalador trabalhará para manter esse nível. O auto escalador calcula o nível de utilização de CPU desejado como uma fração do uso médio de todas as vCPUs ao longo do tempo no grupo de instâncias. Se o uso médio do total de vCPUs for maior que a utilização de destino, o autoescalador adicionará mais máquinas virtuais. Se for definida a utilização de destino como 0,75, o autoescalador tentará manter um uso médio de 75% entre todas as vCPUs no grupo de instâncias.

b) baseado na capacidade de serviço do balanceamento de carga - o *Compute Engine* oferece suporte para balanceamento de carga em grupos de instâncias. Pode-se utilizar o escalonamento automático com balanceamento de carga configurando um escalonador automático que escala com base na carga das instâncias. Um balanceador de carga distribui a carga pelos serviços de *back-end*, que distribui o tráfego entre os grupos de instâncias. No serviço de *back-*

end, pode-se definir a capacidade de balanceamento de carga dos grupos de instâncias como utilização máxima da CPU, máximo de solicitações por segundo (RPS) ou máximo de solicitações por segundo do grupo. Quando um grupo de instâncias atingir a capacidade de veiculação, o serviço de *back-end* começará a enviar tráfego para outro grupo de instâncias.

c) baseado nas métricas do *Stackdriver Monitoring* – Pode-se configurar o dimensionamento automático com base nas métricas. As métricas podem ser padrões fornecidos pelo serviço *Stackdriver Monitoring* ou métricas personalizadas do *Stackdriver Monitoring*

2.3.5 Síntese das propostas

Na realização de uma síntese sobre as técnicas analisadas, observa-se que elas podem ser classificadas em 03 categorias: séries temporais-clássicas, técnicas de aprendizado de máquina e técnicas de redes neurais, conforme detalhadas a seguir.

2.3.5.1 Modelos de Séries Temporais – Modelo Clássico

A modelagem de séries temporais vem sendo estudada há décadas pela comunidade de aprendizado de máquina e de estatística. Uma série temporal é um conjunto de observações ordenadas no tempo. Como exemplos de séries temporais, pode-se citar:

a) valores diários de poluição em uma cidade;
b) valores mensais de temperatura registrados em uma cidade;
c) índices diários da Bolsa de Valores;
d) acidentes ocorridos nas rodovias de uma específica cidade, durante um determinado período.

Os modelos utilizados para descrever séries temporais são processos controlados por leis probabilísticas (Kumar; Singh, 2019). Tem-se na literatura diferentes modelos para descrever o comportamento de uma série particular. Sua construção depende de vários fatores, tais como o comportamento do fenômeno ou o conhecimento *anterior* da natureza e dos objetivos da anál**ise.**

A primeira classe destes modelos, como: autorregressão (**AR**), média móvel (**MA**), suavização exponencial (**ES**), média móvel integrada regressiva automática (**ARIMA**) e outros, foram amplamente utilizados para previsão de carga de trabalho. Estes métodos não

confirmaram sua efetividade na previsão de um maior tempo de análise (Kumar; Singh, 2019).

O modelo **ARIMA** consiste em ajustar modelos autorregressivos integrados de médias móveis, **ARIMA** (p, d, q), a um conjunto de dados. Para a construção do modelo um algoritmo no qual a escolha da estrutura do modelo é baseada nos próprios dados. As previsões utilizando modelos **ARIMA** são eficazes para um período curto e as melhores previsões são aquelas que apresentam um erro quadrático médio mínimo (EQM) (Kumar *et al.*, 2022).

2.3.5.2 Modelos de técnicas de Machine Learning (ML) Convencionais

As abordagens de aprendizado de máquina convencionais, tais como: mapa de recursos de organização automática (*self-organizing feature map* - SOFM), máquinas de vetores de suporte (*Support Vector Machine* - SVM), e k-vizinhos mais próximos (*K-Nearest Neighbors - KNN*), são amplamente adotados para prever a próxima carga de trabalho no servidor (Amiri; Mohammad-Khanli, 2017).

2.3.5.3 Modelos de técnicas de Redes Neurais (Deep Learning)

As *Recurrent Neural Networks* (RNNs) utilizam algoritmos de redes neurais para o processamento de dados sequenciais, como som, dados de séries temporais ou linguagem natural. Esta arquitetura é utilizada para a construção de modelos dinâmicos.

As RNNs diferem das redes *feedforward* porque incluem um loop de *feedback*, pelo qual a saída do passo N-1 é alimentada de volta à rede para afetar o resultado do passo n, e assim por diante. O modelo recorrente inclui o estado oculto que determinou a classificação anterior em uma série. Em cada etapa subsequente, esse estado oculto é combinado com os dados de entrada do novo passo para produzir um novo estado oculto e, em seguida, uma nova classificação.

Kumar *et al.* (2017) propôs o uso das redes LSTM-RNN para o provisionamento automático de recursos em computação em nuvem.

2.3.6 Avaliação da predição de carga de trabalho

Alguns modelos de *Machine Learning* têm por objetivo a **Classificação** e outros a **Regressão.** Não é possível utilizar métricas de classificação em problemas de regressão e vice-

versa. Portanto, ao avaliar modelos de regressão, no caso modelos de previsão de carga de trabalho, é preciso utilizar abordagens específicas para essa avaliação.

Para avaliar a eficácia da previsão de carga de trabalho e analisar seu impacto nos recursos, diversas métricas podem ser utilizadas. Neste tópico serão apresentados os indicadores e métricas a serem utilizadas na avaliação de um modelo preditivo.

2.3.6.1 Análise da acurácia

Os termos precisão, exatidão e acurácia muitas vezes são tratados como sinônimos. Baseadas em normas internacionais, seguem as definições:

a) **precisão** - grau de variação de um conjunto de medições. Quanto maior a precisão, menor a variabilidade entre as medidas. É resultante de um conjunto de medições realizadas. Quanto mais preciso o processo, menor é a variabilidade entre os valores encontrados. Os modelos de previsão são avaliados pelos resultados previstos e cujas saídas são mais próximas dos valores reais. As métricas de desvio ou erro medem a diferença entre o comportamento real e o comportamento previsto. O resultado do erro de previsão, representam problemas como sub-provisionamento e excesso de provisionamento.

b) **exatidão** - Medida de proximidade entre uma determinada medição (ou média de medições) e um valor tido como verdadeiro (ou de referência). Abrange os erros sistemáticos de um conjunto de dados. Exatidão pode ser contrastada com precisão.

c) **acurácia -** segundo a norma ISO 5725-1 (ISO, 1994), acurácia pode ser tratada como a combinação entre **exatidão** e **precisão**. Ou seja, "acurácia = exatidão + precisão". Quando um conjunto de medições é realizado, este conjunto apresenta um componente de erros aleatórios (que não podem ser previstos) e um componente de erros sistemáticos (que seguem uma tendência). Neste caso, exatidão está relacionada com o erro entre a média destes valores e o valor tido como referência. **Precisão** é a dispersão entre os valores de medição**. Acurácia** é a combinação entre estes dois valores. Considera-se que a acurácia é a proximidade de um resultado com o seu valor de referência real. Tomando-se, por exemplo, um serviço de leitura automatizada de documentos que fornece 90% de acurácia, as chances de que

os dados extraídos sejam idênticos aos do documento real são de 90%. Pode-se considerar que 90% dos casos estão corretos e 10% estão errados. Portanto, a **acurácia** pode ser descrita como uma medida obtida de um conjunto de eventos, conforme a verificação a ser feita. A acurácia pode ir de 0% a 100%.

- **acurácia entre 0% e 30%** - considera-se que estes níveis são baixos, havendo pouca convicção de que os resultados encontrados são próximos da realidade. Deve-se evitar soluções com níveis tão baixos de acurácia.
- **acurácia entre 30% e 90%** - são considerados níveis médios, representando risco moderado de que os resultados não sejam condizentes com os valores reais. Soluções com esse nível ainda podem ser considerados. Tem-se um risco moderado de não apresentar resultados aceitáveis em relação aos valores de referência.
- **nível de acurácia entre 90% e 100%** - representa resultados de alta precisão e baixo risco, o que traz mais segurança para a tomada de decisão. A partir de 90% de acurácia, os valores encontrados podem ser considerados provados. A variação em relação ao valor real de referência é muito baixa.

A precisão, exatidão e acurácia são conceitos **qualitativos** e não podem ser medidos**.** As medições devem ser realizadas com variáveis **quantitativas** como: desvio-padrão; erro padrão; erro RMS (valor quadrático médio); dispersão. Portanto, não é correto afirmar: "a precisão da medição foi de 0.54 cm"**.** O certo seria afirmar: "a precisão da medição foi de 0.54 cm (desvio-padrão)".

2.3.6.2 Métricas para predição de erro

Todas as métricas apresentadas neste artigo utilizam essa mesma ideia de cálculo da diferença entre o **valor real e o previsto**, contudo com algumas diferenças. Estas diferenças são importantes para apresentar diferentes perspectiva sobre o desempenho do modelo.

Para a análise de um **modelo preventivo para séries temporais** deve-se medir e analisar os erros que ele apresenta, ou seja, vamos comparar Y e $\hat{Y}$ (Y real e Y previsto, respectivamente) e dar atenção a esses resíduos. A seguir são apresentadas diferentes técnicas para aferição de um modelo:

a) **Erro Médio Absoluto - MAE** (*mean absolute error*) - calcula os erros entre valores reais e valores de predições. Ele é calculado a partir da **média dos erros absolutos**, ou seja, utiliza-se o módulo de cada erro para evitar a

subestimação do valor ser menos afetado por pontos extremos (outliers). Cada erro, pode ser interpretado como a diferença entre Y e Ŷ e assim, tem-se:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | Yi - \hat{Y}i | \qquad MAE = \left(\frac{\sum_{i=1}^{n} |previsão_i - actual_i|}{n}\right)$$

Utiliza-se essa medida em séries temporais, pois há casos em que o erro negativo pode zerar o positivo ou dar uma ideia de que o modelo é preciso. Mede-se apenas a distância do valor real (acima ou abaixo). O **MAE**, mede a média da diferença entre o valor real e o predito. Mas, por haver valores positivos e negativos, é adicionado um módulo entre a diferença dos valores. Essa métrica não é afetada por valores discrepantes — os denominados *outliers*. Nesta equação há o cálculo da média da diferença entre o valor predito ŷ e o real y. **Quanto menor o valor de MAE, significa que melhor são os resultados preditos pelo modelo**. O valor de saída da equação tem a mesma escala dos dados utilizados para previsão, logo fica mais fácil a sua interpretação. Se o valor de **MAE** resultante for igual a 10,01 m, este resultado significa que o modelo pode estar errando em média 10,01 m para mais quanto para menos em relação ao valor correto. Por isso que para uma previsão futura, este resultado precisa ser considerado para a tomada de decisão. Contudo, o quanto este erro representa em relação ao valor real percentualmente?

b) **Erro Quadrático Médio – MSE** – (Mean Squared Error) - é utilizado para verificar a **acurácia de modelos** e dá um maior peso aos maiores erros, já que, ao ser calculado, cada erro é elevado ao quadrado individualmente e, somente após, a média desses erros é calculada. Usa-se o mesmo conceito de erro utilizado anteriormente. Tem-se a seguinte equação:

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Por conta do expoente ao quadrado que o erro assume, essa métrica é bastante sensível a outliers (valores discrepantes) e, caso tenha muitos erros significativos em sua análise, essa métrica poderá ser extrapolada.

O **MSE** é uma métrica que calcula a média de diferença entre o valor predito

com o real, como a métrica **MAE**. Entretanto, ao invés de usar o módulo do resultado entre o valor de *y* e *ŷ*, nesta métrica a diferença é elevada ao quadrado. Desta maneira penalizando valores que sejam muito diferentes entre o previsto e o real. **Portanto, quanto maior for o valor do MSE, significa que o modelo não performou bem em relação às previsões.** Nesta equação há o cálculo da diferença entre o valor real y e o valor predito ŷ, porém elevando o resultado ao quadrado. Desta forma, valores altos, ou seja, que a previsão esteja muito diferente da previsão, são mais penalizados que os demais.

Apesar de sua ideia poderosa, a métrica MSE apresenta um problema de interpretabilidade. Por haver a elevação ao quadrado, a unidade fica distorcida. Ou seja, se a unidade medida for metros (m), o resultado será em m². Por isso que uma adaptação da MSE é a **RMSE** que será apresentada a seguir.

c) **Raiz Quadrada do Erro Médio – RMSE** - (*Root Mean Squared Error*) - é o desvio padrão dos resíduos (erros de previsão). Resíduos são uma medida de quão longe os pontos de dados da linha de regressão estão. A RMSE é uma medida de quão espalhados estão esses resíduos. Informa o quão concentrados os dados estão em torno da linha de melhor ajuste. A **RMSE** é a medida que calcula **a raiz quadrática média** dos erros entre valores reais e valores de predição (Wesner, 2016).

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

Uma característica da **RMSE** é que os erros (reais - predições) são elevados ao quadrado antes de ter a média calculada. Portanto, pesos diferentes serão atribuídos à soma e, conforme os valores de erros das instâncias aumentem, o índice do **RMSE** aumenta consideravelmente. Ou seja, se houver um *outlier* no conjunto de dados, seu peso será maior para o cálculo do **RMSE** e, por conseguinte, prejudicará sua métrica deixando-a maior. O **RMSE** é, basicamente, o mesmo cálculo de **MSE**, contendo ainda a mesma ideia de penalização entre diferenças grandes do valor previsto e o real. Porém, para lidar com o problema da diferença entre unidades, é aplicada a raiz quadrática. Assim a unidade fica na mesma escala que o dado original, resultando em uma

melhor interpretabilidade do resultado da métrica. Nessa equação há o cálculo da diferença entre o valor y e ŷ, contudo com a elevação do resultado ao quadrático. Mas para deixar o resultado na mesma escala que os dados, é aplicado a raiz quadrada no resultado.

Apesar do valor ter a mesma unidade, ele não costuma se assemelhar ao resultado encontrado de **MAE**, demonstrando como os *outliers* podem estar impactando nas previsões do modelo. Mas a sua interpretabilidade pode seguir a mesma lógica, onde o resultado da métrica sendo igual a 80,0 m, significa que o modelo pode estar errando em 80,0 m para mais ou para menos. Por essa razão, essa métrica pode ser uma boa opção quando é preciso ter uma avaliação mais criteriosa sobre as previsões do modelo.

d) **Média Percentual Absoluta do Erro – MAPE** - outra métrica interessante para ser utilizada, devido ao erro ser medido como uma porcentagem, possibilitando fazer comparações entre erros percentuais do modelo entre produtos. A fórmula para cálculo é a seguinte:

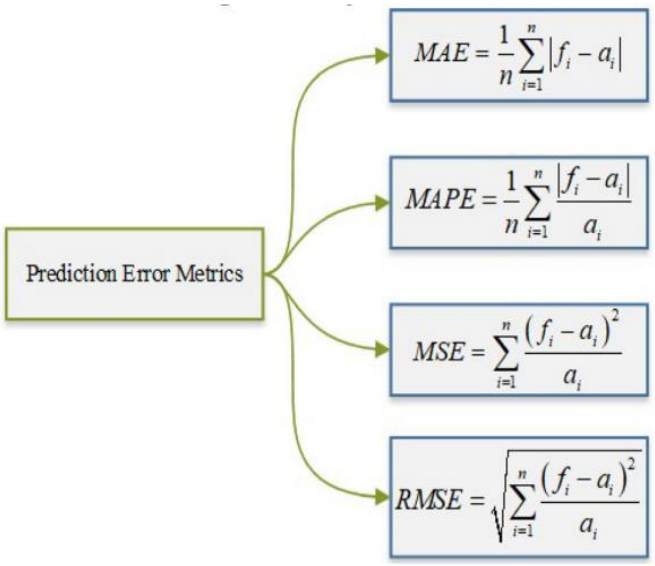
$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}.100\% \qquad MAPE = média\left(\frac{ABS(Valores\ Reais - Previsão)}{Valores\ Reais} * 100\right)$$

A métrica mostra a porcentagem de erro em relação aos valores reais. O cálculo de **MAPE** parece com o do **MAE**, mas com o acréscimo de uma divisão por $|y|$**.** Se o resultado de **MAPE** for igual a 40% significa que o modelo faz previsões nas quais, em média, a diferença entre o valor previsto e o real equivale a 40% do valor real tanto para mais quanto para menos. A métrica **MAPE** é uma das métricas mais usadas para reportar a desempenho do modelo, trazendo uma compreensão mais abrangente do resultado de **MAE**.

A Figura 13 apresenta um agrupamento de métrica de predição, sintetizando as fórmulas das métricas apresentada nesta seção.

Observa-se que a melhor métrica depende da questão em análise. Uma abordagem interessante é utilizar uma grande variedade e modelos para se ter diferentes perspectivas em relação ao desempenho do modelo.

Figura 13 - Métricas de Erro de Predição



Fonte: Masdari; Khoshnevis (2020).

Quadro 2 – Síntese dos Modelos de Avaliação

Métrica	Entendimento
MAE	**Quanto menor for o resultado**, melhor o resultado preditivo alcançado. Utiliza valores absolutos dos erros, o que não é desejável em muitos cálculos matemáticos.
MSE	**Quanto maior**, significa que o modelo não apresenta boa performance preditiva.
RMSE	Penaliza os erros de maior magnitude. Pode não descrever sozinho o erro médio, com outras implicações que são mais difíceis de descobrir.
MAE e **MAPE**	Apresenta uma boa interpretabilidade, principalmente para reportar resultados do modelo.
RMSE e MSE	São afetados por valores discrepantes, o que pode ser importante quando é preciso ter uma avaliação mais criteriosa do modelo.
RMSE e MAE	**Quanto menor, melhor.** São as mais utilizadas para medir a acurácia de modelos preditivos com variáveis contínuas. Expressam o erro médio do modelo preditivo, em relação aos dados originais (treino e/ou teste). Estão no intervalo 0-infinito e retornam a magnitude dos erros e não sua direção. A diferença das métricas pode auxiliar no diagnóstico de predições muito ruins ou *outliers*.
RMSE => MAE	O **RMSE** sempre resultará em valor maior ao **MAE** (no mundo real). Se o **RMSE** ficar muito maior que **MAE,** esta suposição deve ser verificada.
RMSE<= **MAE** * srqt(n) n é a composição da amostra de teste	A diferença entre as métricas é maior quando todo o erro da previsão está em uma única amostra. *Concentrando-se no limite superior, isso significa que o **RMSE** tem uma tendência a ser crescente que o **MAE** à medida que o tamanho da amostra de teste aumenta.* Isso pode ser problemático ao comparar os resultados do **RMSE** calculados em amostras de teste de tamanhos diferentes, o que é frequentemente o caso na modelagem de casos reais.

RMSE = MAE	Erros uniformes.
RMSE x **MAPE**	**RMSE** é uma medida de erro nas unidades originais da variável. O **MAPE** é uma medida de erro relativo, ou seja, a porcentagem de erro.

Fonte: Elaborada pelo autor.

2.4 Computação autonômica em nuvem

As pesquisas em **Computação Autonômica em Nuvem** (*Autonomic Cloud Computing* - ACC) aportaram nesta pesquisa em função do objetivo da construção de um modelo autonômico para o gerenciamento de recursos em nuvem.

O termo "computação autonômica**"** foi cunhado em 2001 por Paul Horn, pesquisador da IBM (Parashar; Hariri, 2007). Horn imaginava uma infraestrutura de TI autônoma que pudesse detectar, analisar e responder a situações automaticamente, aliviando a necessidade de profissionais de TI, realizarem tarefas tediosas de gerenciamento de sistemas. Os princípios da Computação Autonômica foram inspirados no sistema nervoso humano (IBM, 2005). Seu objetivo é a concepção de sistemas que possam se gerenciar, conforme a orientação de alto nível de humanos. Trata-se de uma adaptação inteligente ao ambiente e às solicitações dos usuários. Os sistemas autônomos são caracterizados por possuírem propriedades como autoconfiguração, autocura, auto otimização e autoproteção.

Segundo (Kephart; Chess, 2003), um sistema autonômico compreende um conjunto de elementos autonômicos. Um elemento autonômico é um componente responsável para o gerenciamento de seu próprio comportamento, conforme políticas definidas, e para interação com outros elementos autonômicos que provêm ou consomem serviços computacionais. O elemento básico é o loop de controle que atua como um gerenciador de recursos através de ações de monitoramento, análise e tomada de decisão, seguindo políticas específicas. Como exemplo citam que um loop de controle inteligente consegue prover capacidades de Computação Autonômica, tais como ciclos de processamento adicionais quando necessário, atualizações de software, reinicialização da execução de uma aplicação após falhas e criação de backups após o processamento diário.

2.4.1 Computação autonômica - Definições

Enfrentar os grandes desafios da **Computação Autonômica** requer avanços científicos e tecnológicos em uma variedade de áreas, bem como novos paradigmas de programação e arquiteturas de software que suportem a integração efetiva das tecnologias

constituintes. A Computação Autonômica é definida como um ambiente de computação com a capacidade de gerenciar a si mesmo e se adaptar dinamicamente às mudanças, de acordo com políticas e objetivos definidos. Os ambientes de autogerenciamento podem realizar essas atividades com base nas situações que observam no ambiente em que residem, em vez de exigir que humanos iniciem a tarefa (Ibrahim *et al*., 2017). A Computação Autonômica surgiu como uma solução altamente dinâmica para problemas que vão além da simples automação de sistemas adaptáveis.

A **Computação em Nuvem** associou-se à **Computação Autonômica** para enfrentar alguns de seus desafios. Essa combinação, conhecida como *Autonomic Cloud Computing* (ACC), surgiu como uma progressão natural para ambas as áreas. A ACC ajuda a enfrentar os desafios relacionados à *Quality-of-Service* (QoS), garantindo que o *Service Level Agreement* (SLA) seja atendido. O monitoramento autonômico é implementado em camadas específicas da arquitetura da computação em nuvem.

A **auto escalabilidade** é parte de um conjunto de recursos que denotam a **autonomicidade** de um aplicativo. O conjunto de tais recursos incluem as seguintes capacidades, dentre outras (Singh; Chana, 2016):

a) **auto recuperação** - capacidade de um sistema identificar, analisar e se recuperar automaticamente de uma falha.
b) **auto-organização** - capacidade de um sistema se ajustar dinamicamente sua organização lógica ou física, em tempo de execução, a novos requisitos.
c) **auto adaptação** - capacidade de um sistema se adaptar a um ambiente em mudança de maneira automática.
d) **autoproteção** – Capacidade de um sistema autonômico proteger-se contra ameaças e intrusões - Reflete a necessidade de identificação proativa e proteção contra-ataques arbitrários.

2.4.2 O Modelo MAPE-K

A IBM (IBM, 2005) propôs um modelo de referência para computação autonômica, denominado MAPE, que inclui quatro fases em *loop*, quais sejam: (M) - monitoramento, (A) - análise, (P) - planejamento e (E) - execução. Na fase de **monitoramento**, acontece a coleta das informações sobre os recursos e a carga de trabalho enviada pelos usuários. Na **análise** essas informações são utilizadas para estimar a utilização futura de recursos. Na fase de **planejamento** é determinada uma ação adequada (e.g.

aumentar ou diminuir) a quantidade de VMs a serem alocadas. Finalmente, a ação adequada é realizada na fase de **execução**. As quatro fases são executadas regularmente em intervalos de tempo específicos (Arcaini *et al.,* 2015).

Na definição de um **Sistema Autonômico**, a IBM (2005) estabeleceu oito condições para serem apresentadas em tais sistemas:

a) conhecer a si próprio em termos de quais recursos ele tem acesso, quais são suas capacidades e limitações e como e por que está conectado a outros sistemas;
b) conseguir configurar e reconfigurar-se automaticamente, dependendo da mudança do ambiente de computação;
c) conseguir otimizar seu desempenho para garantir o processo de computação mais eficiente;
d) conseguir solucionar os problemas encontrados reparando-se ou encaminhando as funções para longe do problema;
e) detectar, identificar e se proteger contra vários tipos de ataques para manter a segurança e a integridade gerais do sistema;
f) adaptar-se ao ambiente à medida que muda, interagindo com os sistemas vizinhos e estabelecendo protocolos de comunicação;
g) confiar em padrões abertos e não pode existir em um ambiente proprietário;
h) antecipar a demanda de seus recursos, mantendo a transparência para os usuários.

Embora o comportamento dos sistemas autonômicos apresente variações, todo sistema autonômico deverá exibir um conjunto mínimo de propriedades para atingir seu objetivo. Ele deve ser:

a) **automático**: conseguir controlar suas funções e operações internas. Como tal, um sistema autonômico deve ser independente e capaz de iniciar e operar sem qualquer intervenção manual ou ajuda externa. O conhecimento necessário para inicializar o sistema deve ser inerente ao próprio sistema.
b) **adaptativo**: poder alterar sua configuração, estado e funções. Isso permite que o sistema lide com mudanças temporais e espaciais em seu contexto operacional, seja de longo prazo (customização / otimização do ambiente) ou de curto prazo (condições excepcionais, como ataques maliciosos, falhas etc.).
c) **consciente**: monitorar (detectar) seu contexto operacional e seu estado interno

para poder avaliar se sua operação atual atende a seu objetivo. A conscientização controlará a adaptação de seu comportamento em resposta a mudanças de contexto ou estado.

A **Computação Autonômica** tem sido amplamente aceita (IBM, 2005). Isso se deve ao aumento da complexidade dos sistemas computacionais. A complexidade implica que mais funcionalidades e capacidades serão exigidas por usuários e empresas. Isso resultou na necessidade de configurar ou integrar novas soluções nos sistemas existentes. Com novos dispositivos e maior mobilidade, a interoperabilidade também é uma grande preocupação.

A **Computação Autonômica** mudará como os sistemas de software serão desenvolvidos. Por um lado, a mudança deve ser tal que garanta que a crescente dependência de sistemas de computação cada vez mais complexos permaneça segura e protegida. Por outro lado, a mudança deve preservar a tendência atual no desenvolvimento de sistemas computacionais cada vez mais complexos, oferecendo serviços melhores e mais inovadores para a sociedade. Espera-se que a Engenharia de Software evolua para oferecer os artefatos necessários para facilitar o desenvolvimento de sistemas autonômicos (Lalanda *et al.,* 2013).

A computação em nuvem fornece um tipo de solução que permite que os usuários dessa tecnologia tenham uma percepção do poder e da funcionalidade de computação ilimitados. Isso levou a indústria a fornecer uma solução que compreenda a auto otimização, autorrecuperarão, autoproteção e autoconfiguração. As soluções de **computação em nuvem autonômica** foram amplamente implantadas para resolver problemas que surgem do gerenciamento de serviços de nuvem existentes.

2.5 Redes neurais de memória de curto longo prazo

Nesta seção será apresentada a rede neural ***Long Short-Term Memory*** **(LSTM)** e suas variantes, enfatizando-se a rede ***Stacked Long Short-Term Memory***, que se torna alvo desta pesquisa.

Uma LSTM é uma ***Recurrent Neural Network*** (**RNN**) caracterizada por ser uma rede com memória de longo prazo e adequada para modelar sequências temporais. A LSTM aborda o problema do *vanishing gradient* das RNNs tradicionais por meio de portas que controlam o fluxo de informações na célula LSTM, permitindo que a rede LSTM controle o fluxo de gradientes ao longo do tempo, resolvendo assim o problema *vanishing gradient.*

Diversas arquiteturas de **RNN** foram propostas para tarefas que exigem o aprendizado de dependências temporais de longo alcance, incluindo tradução automática,

geração de legenda de imagem, reconhecimento de fala etc. que têm apresentado bom desempenho na modelagem de relações temporais complexas. Seu desempenho superior se deve a uma combinação de células de memória e mecanismos de portas que mantêm e misturam informações não linearmente ao longo do tempo.

2.5.1 LSTM - uma visão geral

Em 1991, Sepp Hochreiter descreveu o problema do *vanishing gradient* fazendo surgir a rede **LSTM**, posteriormente detalhadas por Sepp Hochreiter e Jürgen Schmidhuber em 1997 (Hochreiter, Schmidhuber, 1997). As LSTMs podem "aprender" dependências de longo prazo que as **RNNs** tradicionais não conseguem. O principal *insight* dessa capacidade é um módulo persistente chamado de *estado da célula* que compreende um segmento comum ao longo do tempo. A LSTM possui dependências de longo alcance que tornam o LSTM mais precisa do que as RNNs convencionais. Registram-se ainda muitas aplicações práticas para as LSTMs, incluindo processamento de linguagem natural, geração automática de texto, análise de séries temporais etc.

Ao contrário das RNN tradicionais, a LSTM contém unidades especiais chamadas blocos de memória na camada oculta. Os blocos de memória contêm células de memória com auto conexões que armazenam o estado temporal da rede, além de unidades multiplicativas especiais chamadas portas para controlar o fluxo de informações. Cada bloco de memória na arquitetura original contém três tipos de portas que são:

a) porta de entrada: O portão de entrada controla o fluxo de ativações de entrada na célula de memória.
b) porta de saída: A porta de saída controla o fluxo de saída das ativações de células para o resto da rede.
c) porta de esquecimento: dimensiona o estado interno da célula antes de adicioná-la como entrada à célula por meio da conexão auto recorrente da célula, portanto esquecendo ou redefinindo a memória da célula de forma adaptativa.

Diversas modificações na LSTM original foram propostas, sugerindo-se a existência de novas arquiteturas para atender às demandas específicas. Dentre tais mudanças na arquitetura básica da célula da LSTM, denominada Clássica ou *Vanilla* são evidenciadas a *Peephole Connections, a Gated Recurrent Unit* (GRU), a *Stacked* LSTM e muitas outras.

Brownlee (2020) apresenta uma classificação de variantes da LSTM, segundo a

quantidade de variáveis que manipulam: *Univariate*, *Multivariate*; e *Multi-step* e *Multivariate*.

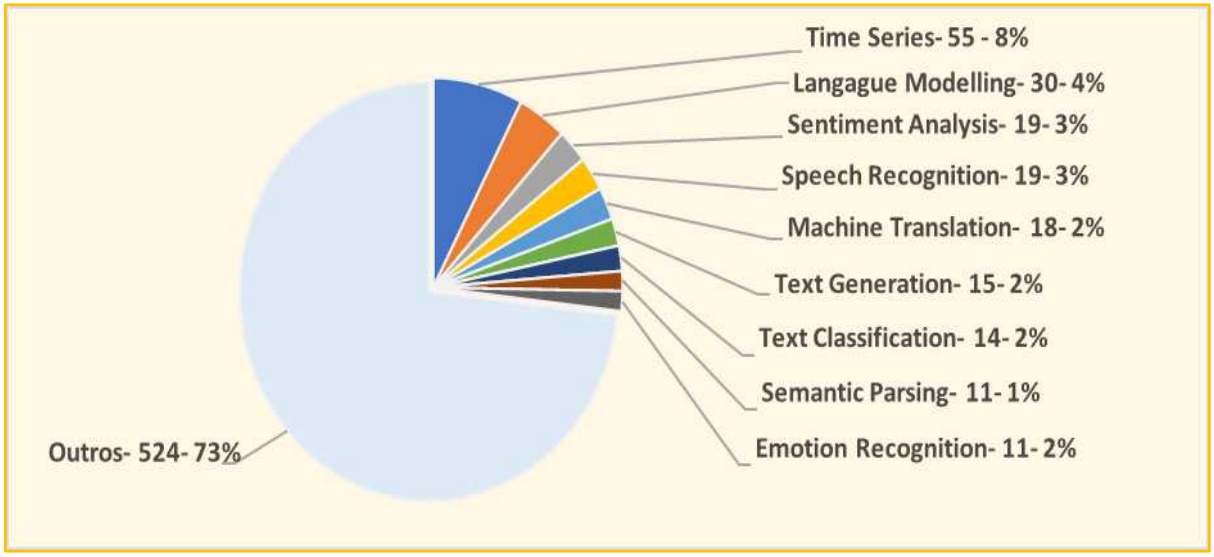
Dentre as *Univariates* o pesquisador distingue: *Vanilla LSTM, Stacked LSTM, Bidirecional LSTM, CNN LSTM e a CONVLSTM*. No agrupamento das *Multivariate*, apresenta a *Multiple Input Series e a Multiple Parallel Series*. Nas *Multi-Step*, são apresentados dois modelos: *Vector Output Model e a Encoder-Decoder Model*. Finalmente, no grupo das *Multivariate Multi-Step LSTM* apresenta a *Multiple Input Multi-Step Output e a Multiple Parallel Input and Multi-Step Output*.

Motivados pelas críticas às redes LSTMs**,** Josefowicz *et al.* (2015) buscaram por uma arquitetura que apresentasse um maior desempenho que as LSTM. A conclusão dos pesquisadores foi que, muito embora tenham identificado arquiteturas que superavam as **LSTMs** em alguns problemas, eles não conseguiram encontrar uma arquitetura que a superasse, consistentemente, em todas as condições experimentais.

Segundo a comunidade *Papers with Code* (Papers With Code, 2022), que mantém um repositório aberto de recursos em *Machine Learning,* a LSTM tem sido abordada em uma quantidade expressiva de artigos, e com uma maior proeminência nas pesquisas sobre a aplicação - *Time Series,* conforme demonstrado na Figura 14. Esta observação apoia a ideia da utilização da LSTM para os objetivos deste trabalho.

A seguir serão explicitadas as características fundamentais da arquitetura da LSTM Clássica, bem como as diferentes apresentações desta arquitetura, as quais são denominadas **Variantes da LSTM**.

Figura 14 - Artigos sobre a Rede Neural LSTM



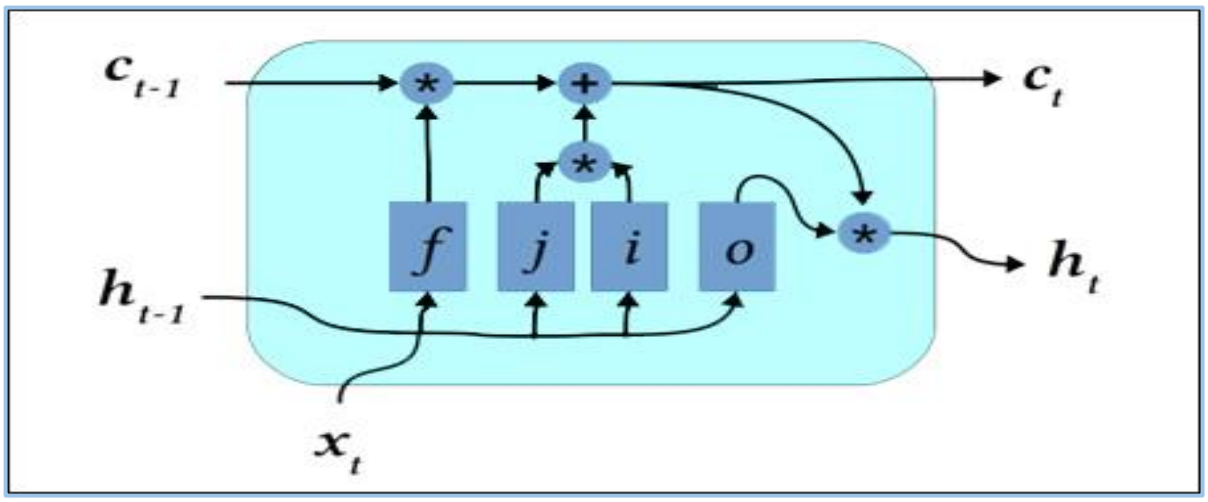
Fonte: Papers With Code (2022).

2.5.2 LSTM - clássica

A arquitetura **LSTM** clássica, conhecida como *Vanilla **LSTM***, é caracterizada por um estado da célula linear persistente, cercado por camadas não lineares que alimentam a entrada e analisam sua saída.

A Figura 15 apresenta um esquema básico de sua arquitetura tradicional. Observa-se que o estado da célula funciona em conjunto com 4 camadas de porta, muitas vezes chamadas de **portas de esquecimento**, 2 camadas - entrada e saída

Figura 15 - Arquitetura básica da LSTM Clássica



Fonte: Vu (2019).

A **porta de esquecimento** escolhe quais valores antigos do estado da célula devem ser eliminados, considerando os dados atuais de entrada. As duas portas de entrada (frequentemente denotadas por **i** e **j**) trabalham para decidir o que adicionar ao estado da célula, dependendo da entrada. As portas **i** e **j** normalmente têm diferentes funções de ativação, que intuitivamente espera-se que sejam utilizadas para sugerir um vetor de escala e valores candidatos para adicionar ao estado da célula. Finalmente, a porta de saída determina quais partes do estado da célula devem ser passadas para a saída. Observa-se que no caso das *LSTMs* clássicas, a **saída h** consiste em ativações da camada oculta (e.g.: elas podem ser submetidas a outras camadas para classificação), e a **entrada c** consiste na saída do estado oculto anterior e quaisquer novos dados **X** fornecidos no atual passo de tempo.

Observa-se mais detalhes na Figura 16, onde pode ser observado que a rede recebe três entradas. Observar os elementos enfatizados que são fundamentais para o

entendimento do funcionamento das redes LSTM, quais sejam:

a) **Xt** é a entrada do passo de tempo atual.
b) $\mathbf{h_t}$ é a saída da rede atual.
c) $\mathbf{h_{t-1}}$ é a saída da unidade LSTM anterior
d) $\mathbf{C_{t-1}}$ é a memória da unidade anterior (a entrada mais importante).
e) $\mathbf{C_t}$ é a memória atual da unidade.

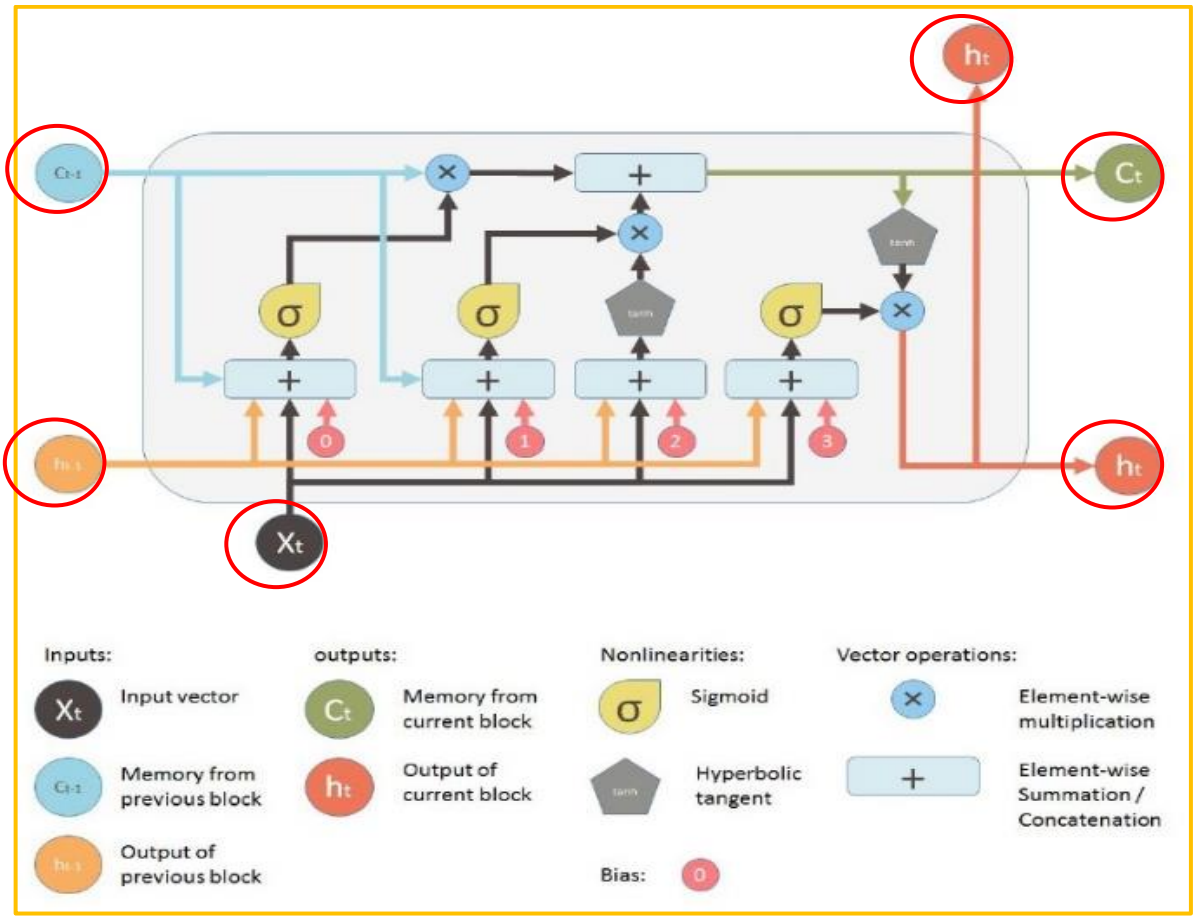
A rede LSTM original melhorou o estado da arte de um conjunto de experimentos com longos intervalos de tempo e com dados relevantes. Atualmente ainda se observa a rede LSTM Clássica formando um elemento central de avanços de aprendizado por reforço de última geração.

2.5.3 LSTM - Variantes

Compreender a arquitetura da rede LSTM é fundamental para a escolha da arquitetura mais adequada para um projeto. A escolha da LSTM clássica ou de uma de suas variantes deve estar alinhada com os requisitos do projeto, características dos dados e restrições computacionais. Compreender os pontos fortes e os recursos exclusivos de cada variante LSTM permite a tomada de decisão fundamentada, garantindo que a arquitetura selecionada seja adequada para as complexidades de uma tarefa específica. Com os avanços contínuos nas pesquisas em aprendizagem profunda, novas arquiteturas LSTM poderão ser introduzidas, expandindo ainda mais o conjunto de ferramentas disponíveis.

Nas seções seguintes serão abordadas as principais variantes da arquitetura LSTM que foram analisadas dentro deste estudo.

Figura 16 - Arquitetura do Modelo LSTM



Fonte: Yan (2016).

2.5.3.1 LSTM Encoder–Decoder

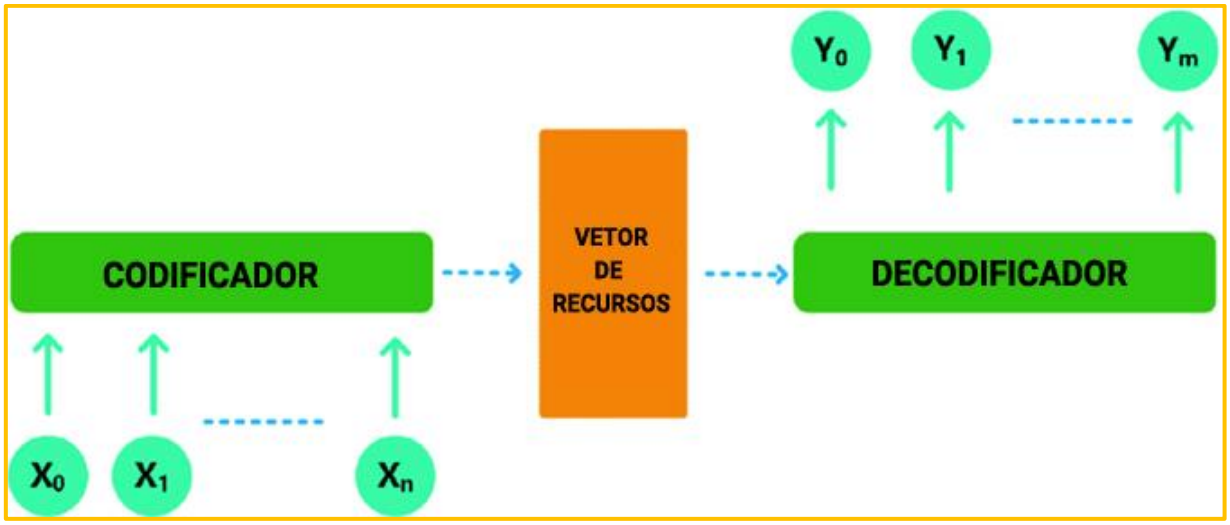
A LSTM *Encoder-Decoder* agrega duas LSTMs. A primeira LSTM (Codificador), processa a sequência de entrada e gera um estado codificado. O estado codificado resume as informações na sequência de entrada. A segunda LSTM (Decodificador), usa o estado codificado para produzir uma sequência de saída. As sequências de entrada e saída podem ter comprimentos diferentes. Várias aplicações têm sido indicadas para o modelo LSTM *Encoder-Decoder*. Algumas das aplicações são: máquina de tradução, resumo de texto, processamento de imagem, assistente virtual para comunicação com usuários e previsão de séries temporais.

A Figura 17 mostra uma representação da arquitetura do modelo LSTM *Encoder-Decoder*, onde os elementos **X** são as entradas para o modelo e **y** as saídas. Na arquitetura apresentada tem-se três componentes: Codificador, Vetor de recursos e o Decodificador

a) **codificador**: recebe os elementos da sequência de entrada a cada passo de tempo, aprende a informação da entrada e propaga para outros processos.
b) **vetor de recursos**: estado interno e intermediário mantém as informações sequenciais da entrada que são úteis para o decodificador fazer previsões.
c) **decodificador**: realiza as previsões decodificando o resultado pelo codificador novamente em um formato sequencial.

Registra-se que a abordagem *Encoder-Decoder* apresenta problema ao codificar uma sequência de entrada em um vetor de saída de tamanho fixo para grandes sequências.

Figura 17 - Arquitetura LSTM *Encoder-Decoder*



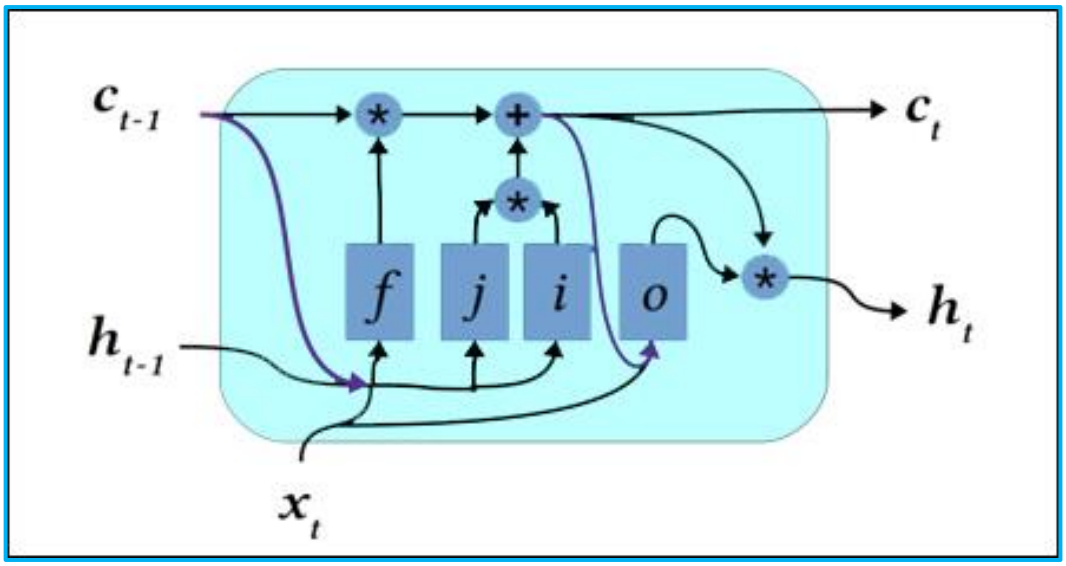
Fonte: Verma (2021).

2.5.3.2 Peephole Connections

A rede LSTM clássica superou o problema de gradientes que desaparecem em uma rede neural recorrente. Isto acontece conectando todos os pontos de tempo através de um estado persistente da célula (chamado de carrossel de erro constante). No entanto, as camadas de restrição que determinam o que esquecer, o que adicionar e até o que levar do estado da célula, como saída, não consideram o conteúdo da própria célula.

Faz sentido que um modelo deseje conhecer as memórias que já possui antes de substituí-las por novas, inserindo as conexões do olho mágico LSTM. Essa modificação (mais escuro na Figura 18) concatena o conteúdo do estado da célula para as entradas da camada de restrição. O fornecimento de algumas conexões do estado da célula para as camadas em uma rede *LSTM* continua sendo uma prática comum, embora variantes específicas apresentem divergências em quais camadas serão fornecidas acesso.

Figura 18 - LSTM *Peephole Connection*



Fonte: Vu (2019).

2.5.3.4 LSTM com Attention

Attention no aprendizado de máquina diz respeito à capacidade de um modelo se concentrar em elementos específicos nos dados, como as saídas de estado oculto de uma LSTM. Zhu *et al*. (2019) utilizou uma arquitetura que consiste em uma rede de **atenção** intercalada entre codificação e decodificação de camadas LSTM para alcançar a tradução automática de última geração. A demonstração do uso de ferramentas da OpenAI em um ambiente de aprendizado por reforço é um exemplo da capacidade de LSTMs com *Attention* em uma tarefa complexa e não estruturada.

2.5.3.5 Gated Recurrent Unit (GRU)

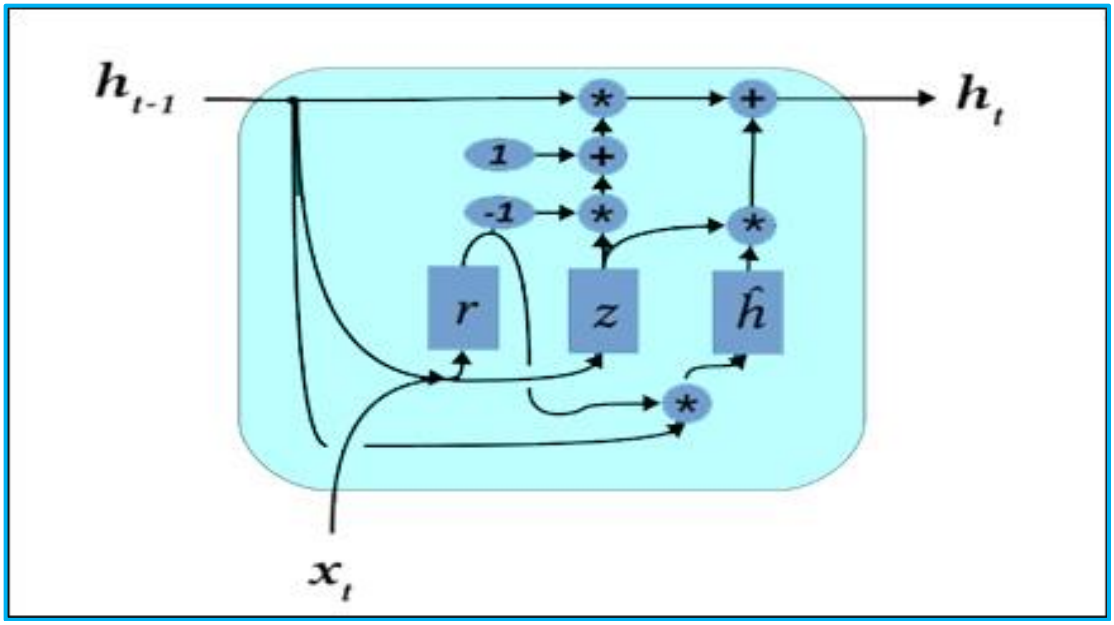
A *Gated Recurrent Unit* (GRU) foi proposta por Cho *et al*. 2014. A GRU é considerada uma variação da LSTM porque ambas são projetadas de maneira semelhante e, em alguns casos, produzem excelentes resultados. A GRU visa resolver o problema da dissipação do gradiente que é comum em uma rede neural recorrente padrão. Uma GRU é mais simples que uma LSTM clássica. Seu processo de treinamento é um pouco mais rápido que o LSTM tradicional.

Unidades recorrentes fechadas (GRUs) têm sido usadas como base para demonstrar conceitos, como GPUs neurais, bem como um modelo mais simples para o

aprendizado sequência a sequência em geral, como tradução automática. Embora ela possa aprender rapidamente em tarefas como música ou geração de texto, ela é considerada menos poderosas que as LSTMs clássicas.

Conforme apresentado na Figura 19, a GRU combina as funções de porta da porta de entrada **j** e da porta de esquecimento **f** em uma única porta de atualização **z.** Isso significa que as posições do estado da célula marcadas para esquecimento serão correspondidas por pontos de entrada para novos dados. Outra diferença importante da **GRU** é que o estado da célula e a saída oculta **h** foram combinados em uma única camada de estado oculto, enquanto a unidade também contém um estado oculto interno intermediário.

Figura 19 - LSTM *Gated Recurrent Unit*



Fonte: Vu (2019).

2.5.3.6 Multiplicative LSTM

Krause *et al*. (2016) propuseram uma arquitetura híbrida denominada *Multiplicative Long Short-Term Memory* (mLSTM). A rede mLSTM é uma arquitetura de RNN projetada inicialmente para modelagem de sequência, combinando a memória de longo prazo (LSTM) e a arquitetura *multiplicative Recurrent Neural Network* (mRNN). A mLSTM é caracterizada por sua capacidade de ter diferentes funções de transição recorrente para cada entrada. A mLSTM é composta pela transição fatorada de oculto para oculto das mRNNs e a estrutura de **portões** das LSTMs. Os autores colocam o estado intermediário **mt** da mRNN em cada unidade de porta da LSTM para combinar as arquiteturas mRNN e LSTM.

Essa variante tornou-se peça central nas pesquisas em Processamento de Linguagem Natural (PLN). Sua utilização mais conhecida foi o neurônio de **sentimento não supervisionado** do OpenAI (Krause *et al.*, 2017).

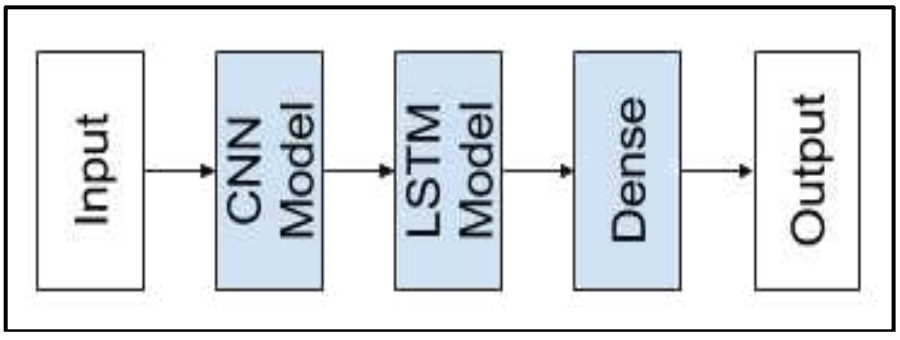
Os pesquisadores demonstraram que a mLSTM supera o LSTM padrão e suas variantes para tarefas de modelagem de linguagem ao nível de character. Ao treinar um grande modelo mLSTM na previsão não supervisionada de texto, observou que ele se tornou mais capaz e poderia executar em alto nível em uma bateria de tarefas de Programação Neurolinguística (PNL) com ajuste fino mínimo. Vários recursos interessantes de texto (como sentimento) foram mapeados de forma emergente para neurônios específicos. Notavelmente, o mesmo fenômeno de neurônios de classificação interpretáveis emergentes do aprendizado não supervisionado foi relatado no aprendizado de sequências de proteínas.

2.5.3.7 CNN LSTM

A *Convolutional Neural Network* (CNN) é uma rede desenvolvida para trabalhar com dados de imagem bidimensionais. Ela poderá ser eficaz em extrair e aprender recursos de dados de sequência unidimensionais, como dados de séries temporais com uma única variável.

Uma CNN poderá ser utilizada em um modelo híbrido com um *back-end* LSTM. Nesta arquitetura a CNN é utilizada para interpretar subsequências de entrada que são fornecidas como uma sequência para interpretação por um modelo LSTM. Este modelo híbrido é denominado CNN-LSTM. A Figura. 20 apresenta uma visão dessa arquitetura.

Figura 20 - Arquitetura CNN-LSTM



Fonte: Brownlee (2017).

O primeiro passo é dividir as sequências de entrada em subsequências que podem ser processadas pelo modelo. Pode-se primeiro dividir os dados de séries temporais de variáveis únicas em amostras. Cada amostra pode ser dividida em duas subamostras, cada uma com duas etapas de tempo. A CNN pode interpretar, cada subsequência e fornecer uma série temporal de interpretações das subsequências ao modelo LSTM para processar como entrada.

Pode-se parametrizar e definir o número de subsequências utiliza-se a opção (**n_seq)** e o número de intervalos de tempo com a opção (**n_steps)**. O modelo CNN tem uma camada *convolucional* para leitura nas subsequências que requerem um número de filtros e um tamanho de kernel a ser especificado. A quantidade de filtros é o número de leituras da sequência de entrada. O tamanho do *kernel* é o número de etapas de tempo incluídas em cada operação de leitura da sequência de entrada.

A camada de convolução é seguida por uma camada de agrupamento máximo que destila os mapas de filtro até metade do seu tamanho. Essas estruturas são então achatadas em um único vetor unidimensional para ser usado como uma única etapa de tempo de entrada para a camada LSTM.

Os dados de entrada que representam períodos mais extensos podem ser filtrados e reduzidos com base em operações de convolução incorporadas em redes LSTM ou diretamente na estrutura de células LSTM. Os portões recebem os recursos gerados como novas entradas. Eles são uma representação reduzida que captura apenas as informações mais relevantes para que a eficiência do mecanismo de atualização do estado da célula seja aprimorada.

2.5.3.8 Nested LSTM

Moniz *et. al.* (2017) propuseram a *Nested* LSTMs (NLSTM). Uma arquitetura com múltiplos níveis de memória adicionando profundidades às LSTMs fazendo oposição às redes *Stacked* LSTM (SLSTM). As NLSTMs superam as SLSTMs e as de camada única com números semelhantes de parâmetros, conforme observado em alguns experimentos em aplicações de modelagem de linguagem ao nível de caracter.

2.5.3.9 Stacked Long Short-Memory - (SLSTM)

O empilhamento de camadas no LSTM faz com que o modelo ganhe mais precisão que é o apelo das técnicas de *deep learning*. A profundidade das redes neurais é atribuída ao sucesso da abordagem em uma ampla gama de problemas de previsão. O sucesso das redes neurais profundas é comumente atribuído à hierarquia introduzida devido às várias camadas.

Cada camada processa parte da tarefa que deseja resolver e passa para a próxima. Nesse sentido, a *Deep Neural Networks* **(**DNN) pode ser vista como um *pipeline* de processamento, no qual cada camada resolve uma parte da tarefa antes de passá-la para a próxima até que finalmente a última camada forneça a saída.

As camadas ocultas adicionais objetivam recombinar a representação aprendida das camadas anteriores e criar representações com mais alto nível de abstração.

O aprendizado profundo é construído em torno da hipótese de que um modelo hierárquico profundo pode ser exponencialmente mais eficiente na representação de algumas funções. Os mesmos benefícios poderiam ser aproveitados para as redes LSTMs. Dado que as LSTMs operam em dados de sequência, isso significa que a adição de camadas incluiria níveis de abstração de entrada ao longo do tempo.

A rede neural *Stacked* LSTM, também denominada LSTM Empilhada, LSTM profunda ou *Stacked Long Short-Memory* (SLSTM) foi projetada, inicialmente, para reconhecimento de fala. Este modelo é composto por várias camadas LSTM ocultas empilhadas. Uma camada LSTM requer uma entrada tridimensional, e as células LSTMs, por padrão, produzem uma saída bidimensional como uma interpretação do final da sequência. Neste caso, é necessário realizar uma adaptação para se ter uma saída 3D da camada LSTM oculta e transformá-la assim numa entrada para a próxima camada (Wu *et al*., 2018).

A questão que inspirou a criação desta variante foi pensar que as RNNs poderiam se beneficiar da profundidade no espaço; ou seja, empilhar várias camadas ocultas recorrentes, umas sobre as outras, assim como as camadas de *feedforward* são empilhadas em redes profundas convencionais. Foi identificado que a profundidade da rede era mais importante que o número de células de memória em uma determinada camada.

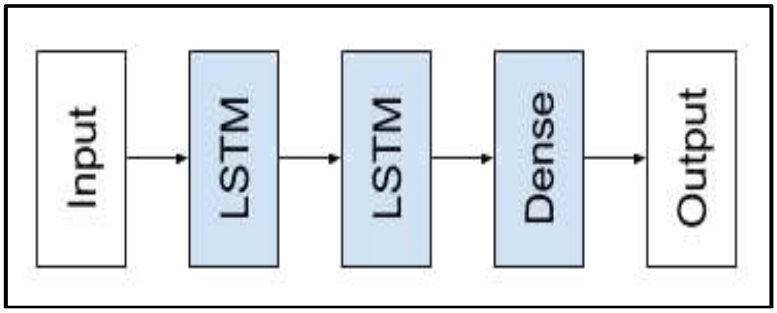
Uma camada LSTM acima fornece uma saída de sequência em vez de uma saída de valor único para a camada LSTM abaixo. Pode-se fazer com que a LSTM produza um valor para cada intervalo de tempo nos dados de entrada, definindo o argumento **return_sequences=True** em sua camada. Isso permite ter uma saída 3D da camada LSTM oculta como entrada para a próxima.

O empilhamento de camadas ocultas do LSTM torna o modelo mais profundo, ganhando mais precisão como uma técnica de aprendizado profundo. É a profundidade das redes neurais, geralmente atribuída ao sucesso da abordagem em uma ampla gama de problemas. Cada camada processa alguma parte da tarefa que deseja resolver e passa para a próxima. Nesse sentido, a *Stacked* LSTM pode ser vista como um *pipeline* de processamento (Brownlee, 2019). A Figura 21 apresenta uma visão esquemática da *Stacked* LSTM. A Figura 22 apresenta um maior detalhamento da arquitetura da *Stacked* LSTM, apresentado a sequência das camadas.

Aumentar a profundidade da rede fornece uma solução alternativa que requer menor quantidade de neurônios e apresenta um processo de treinamento mais rápido.

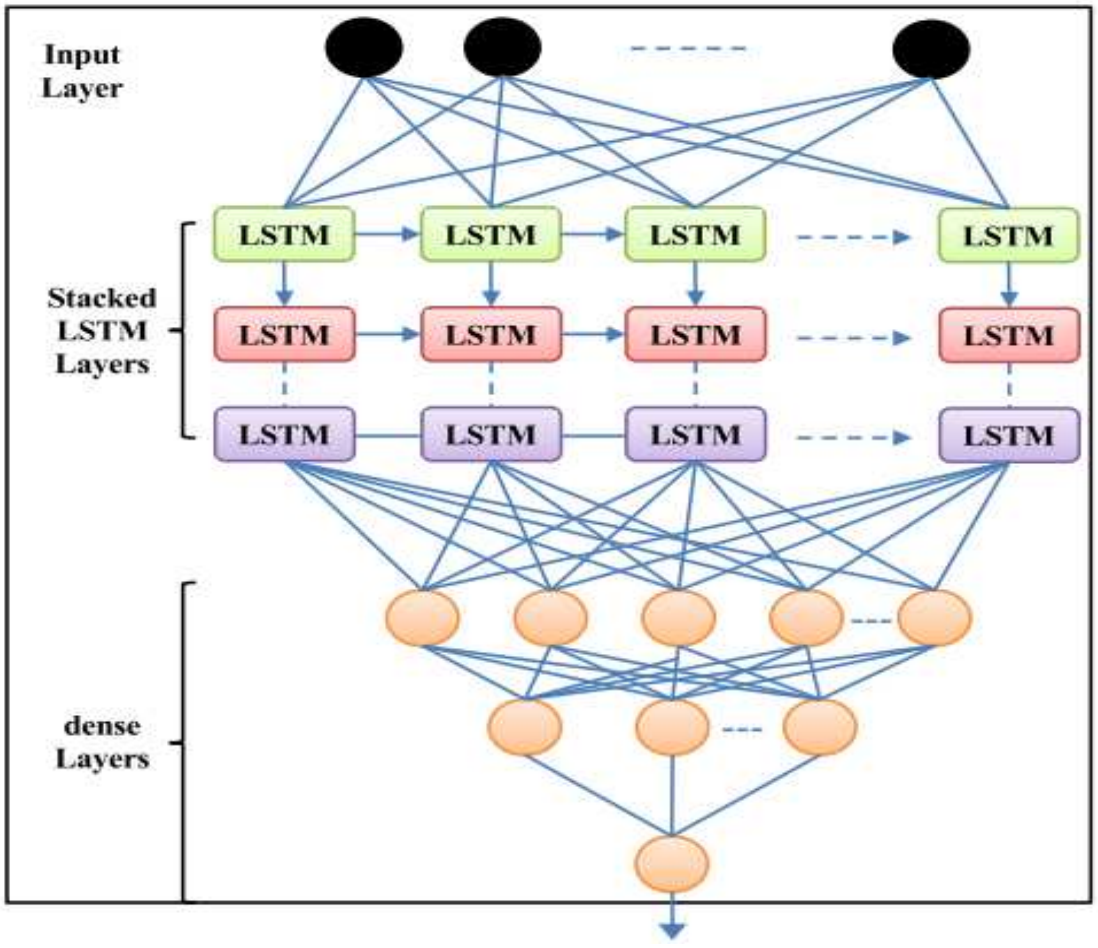
Adicionar profundidade é um tipo de otimização representacional.

Figura 21 - Visão Geral - Arquitetura da *Stacked* LSTM



Fonte: Brownlee (2019).

Figura 22 - Arquitetura Interna – SLSTM



Fonte: Farrag (2021).

2.5.4 Síntese das variantes

Nesta seção foram apresentadas variantes da rede **LSTM,** as quais foram experimentadas por diferentes pesquisadores, na busca de um melhor desempenho para determinados tipos de aplicações. Convém observar a existência de diversos outros modelos idealizados para problemas específicos. Obviamente, estes modelos apresentam pontos positivos e negativos.

Vários artigos compararam variantes da **LSTM** e seu desempenho em uma variedade de tarefas. De uma maneira geral, a LSTM original de 1997 funciona tão bem quanto às variantes mais recentes. Josefowicz *et al*. (2015) analisaram o desempenho de mais de 10.000 permutações de LSTM geradas como "mutantes" e descobriram que algumas das mutações tiveram um desempenho melhor do que as variantes LSTM, mas não em todas as tarefas estudadas.

A melhor LSTM será aquela que estiver otimizada para o problema. Portanto, é importante entender como ela funciona e comparar com as necessidades observadas da aplicação.

3 TRABALHOS RELACIONADOS

> *No princípio era o Verbo, e o Verbo estava com Deus, e o Verbo era Deus. Ele estava no princípio com Deus. Tudo foi feito por Ele; e nada do que tem sido feito, foi feito sem Ele. Nele estava a vida e a vida era luz dos homens. (João 1:1-4)*

Na literatura são encontradas diversas propostas de solução para o **Provisionamento Preditivo**, mantendo a respectiva da manutenção do SLA estabelecido. Algumas delas utilizam recursos de Computação Autonômica no processo do provisionamento preditivo.

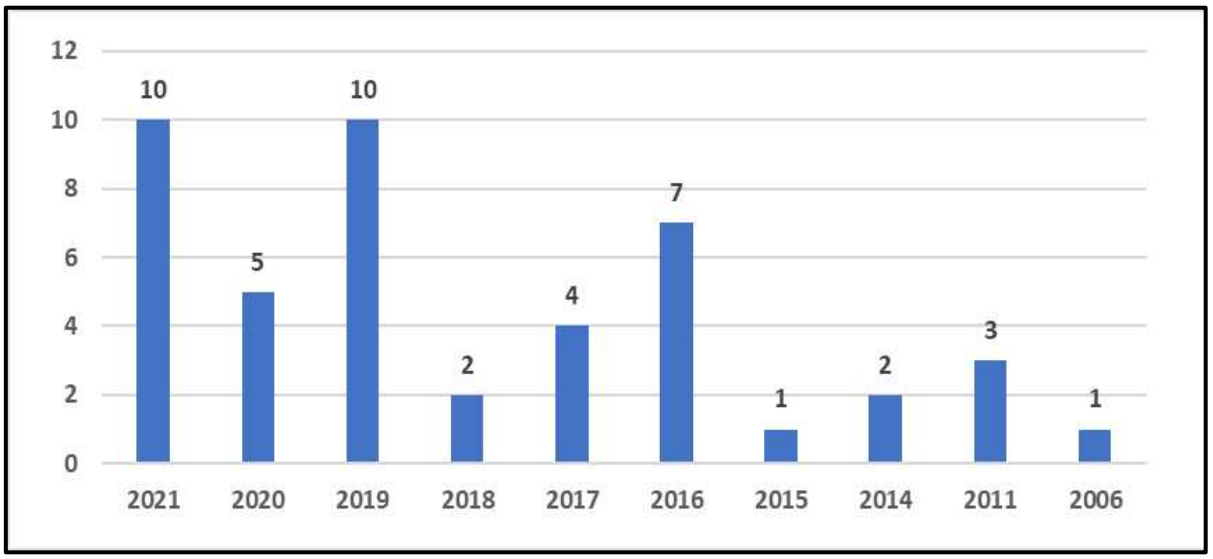
Devido à diversidade de tecnologias e a pouca disponibilidade de informações acerca dos requisitos e da configuração utilizada em tais propostas, nem sempre é viável repetir tais experimentos ou até mesmo tecer uma avaliação comparativa entre elas.

Neste capítulo serão apontadas arquiteturas que apresentam características relacionadas com os interesses estabelecidos nesse trabalho, quais sejam, *uma arquitetura preditiva para o provisionamento autonômico de um ambiente em nuvem*. De uma maneira bem específica, este capítulo destaca questões sobre os esquemas preditivos investigados, buscando-se capturar contribuições, algoritmos, conjunto de dados, simuladores, métricas de avaliação e pontos de falhas, para, a partir daí, identificar melhorias dentro dos objetivos desta proposta de trabalho.

Essa atividade beneficiou-se de *surveys* realizados por diversos pesquisadores e de demais pesquisas realizadas em bases de dados *on-line*, como IEEE Xplore (https://ieeexplore.ieee.org), ACM DL (https://dl.acm.org), Science Direct (https://www.sciencedirect.com) e Google Scholar (https://scholar.google.com/intl/en-US/scholar/about.htm). As pesquisas que apresentam características desejáveis para o objetivo deste estudo foram analisadas e registradas. Os dados foram consolidados por ano da pesquisa, título da proposta, autores, mecanismos, avaliação e base de dados/trace utilizados. Uma visão aberta dessa revisão, com respectivos links, é apresentada no **Apêndice D – Trabalhos Anteriores**. Considerando o conjunto das informações coletadas, a Figura 23 exibe uma tabulação das pesquisas realizadas nos últimos anos sobre **Provisionamento Preditivo em Nuvem,** demonstrando a atualidade no interesse nesse tema.

Dentre os trabalhos pesquisados foram distinguidas 04 abordagens

Figura 23 - Provisionamento Preditivo em Nuvem - Interesse na pesquisa



Fonte: Elaborada pelo autor.

3.1 Arquiteturas preditivas – usando machine learning

Observa-se uma necessidade constante do provimento de escalabilidade automática nos ambientes ofertados pelos CSPs. A maioria dos provedores de grande porte, manifesta a utilização de recursos de **escalabilidade preditiva** com o uso de ***Machine Learning*****,** muito embora não externalizem os detalhes técnicos de sua utilização**.** Existe grande expectativa para o uso de **redes neurais** na implementação de modelos de estimativa de carga de trabalho.

Muitas propostas analisadas utilizam técnicas diversificadas e convencionais de *Machine Learning,* como as encontradas nos trabalhos a seguir:

a) Cao et al. (2003) - apresentaram uma estrutura preditiva baseada em mapa auto-organizáveis (*Self-Organizing Map*-SOM) e Máquinas de Vetores de Suporte (Support Vector Machine-SVM) para dados de séries temporais. Esta abordagem empregou duas fases. Inicialmente os dados são agrupados usando SOM e posteriormente a previsão é realizada usando o SVM.

b) Ban et al. (2013) - experimentaram uma arquitetura em dois níveis usando kNN (k-Nearest Neighbors) para previsão de séries temporais. Em seguida, uma rede neural foi utilizada para modelar a variação da carga de trabalho em projetos multimídia.

c) Jheng et al. (2014) - utilizaram o Grey Forecasting Model para alocar VMs. Usaram a carga dependente de tempo, no mesmo período em cada dia, e

previram se a tendência da carga de trabalho era de aumento ou diminuição. Compararam o valor previsto com a carga do período anterior e decidiram qual VM deveria ser migrada.

3.2 PRMF - framework preditiva de gerenciamento de recursos

Balaji *et al.* (2018) propuseram uma **Framework Preditiva de Gerenciamento de Recursos** (*Predictive Resource Management Framework* - PRMF). As principais métricas dos padrões de carga de trabalho foram monitoradas e analisadas *offline,* usando o módulo da PRMF para determinar a métrica de avaliação principal. O modelo de melhor ajuste seria reavaliado, caso o nível de confiança de 95% do valor previsto sobre a métrica real fosse violado. Para os experimentos, foram considerados o *Request Arrival,* como a principal métrica de avaliação, e ARIMA, como o modelo de melhor ajuste. A abordagem preditiva proposta apresentou melhores resultados que a abordagem reativa durante o provisionamento / desprovisionamento de instâncias nos experimentos em tempo real. Para a construção do PRMF, diferentes abordagens e métricas foram analisadas, conforme observado no Quadro 3, construído pelos autores do framework.

Quadro 3 - Métricas de *Workload* analisadas em PRMF

ABORDAGEM	MÉTRICAS DE WORKLOAD
Análise de logs do Balanceador de carga em tempo real (Iqbal *et al.*, 2009).	Tempo de resposta
Técnicas de Aprendizagem de Máquina (Xiong *et al.*, 2011).	Memória e CPU
Técnicas de Filas e Análise Combinatória (Bennani; Menasce, 2005).	Tempo de resposta e de vazão média
Abordagem baseada em fila (Bhulai *et al.*, 2007).	Tempo de resposta
Modelo linear e de filas (Shi *et al.*, 2011).	Tempo de resposta
Aprendizado de Máquina e Regressão Linear (Bankole; Ajila, 2013).	Tempo de resposta, índice de vazão média e tempo de utilização de CPU
Teoria das filas e suavização exponencial (Wang *et al.*, 2014).	Taxa de Chegada
Algoritmos de Mineração de Dados (Tammaro *et al.*, 2011).	Taxa de Chegada / Tempo de Desmontagem
Modelo ARIMA (Han *et al.*, 2013).	Taxa de Chegada / Taxa de Partida
Modelo ARIMA (Deng *et al.*, 2013); (Calheiros *et al.*, 2011); (Calheiros *et al.*, 2014).	Requisição Processada / Vazão Média
Provisionamento baseado em Políticas (Elprinc, 2013)	CPU, Memória e Tempo de resposta
Técnicas de Aprendizagem de Máquina (Kim *et al.*, 2011)	Tempo de Execução / Vazão Média
Provisionamento e desprovisionamento baseado	Requisição Processada / Vazão Média

em Filas (Ali-Eldin *et al.*, 2012)	
Provisionamento e desprovisionamento baseado em políticas (Kouki *et al.*, 2014); (Kouki *et al.*, 2015).	Carga de Trabalho / Vazão Média

Fonte: Balaji *et al.* (2018).

3.3 Arquitetura autonômica para elasticidade

A arquitetura proposta por Coutinho *et al.* (2014) objetivou avaliar o comportamento do ambiente, mediante cargas de trabalho, e como se comportam de maneira autonômica para adaptação às variações de demanda (elasticidade) e manutenção do SLA.

Foi utilizada a elasticidade horizontal, através da qual à medida que os recursos são necessários, novas instâncias de máquinas virtuais são adicionadas, por meio de um balanceador de carga, e retiradas, caso não sejam mais necessárias. O intervalo de coleta de dados foi definido a cada **x** segundo. A métrica utilizada para disparar ações de elasticidade foi a **média do percentual de utilização de CPU das máquinas virtuais**. Os limites utilizados para a execução das atividades foram: acima de 70% (aloca uma nova máquina virtual), abaixo de 60% (desaloca uma máquina virtual), e entre 60% e 70% (mantêm alocação). Esse valor foi calculado como a média das 10 últimas coletas de utilização de CPU nas máquinas virtuais. Como mecanismo de predição foi utilizado **regressão multilinear** sobre valores de utilização de CPU, memória, disco e rede, coletados em experimentos prévios, com cargas de trabalho semelhantes. A arquitetura proposta possui os seguintes componentes, conforme representados na Figura 24:

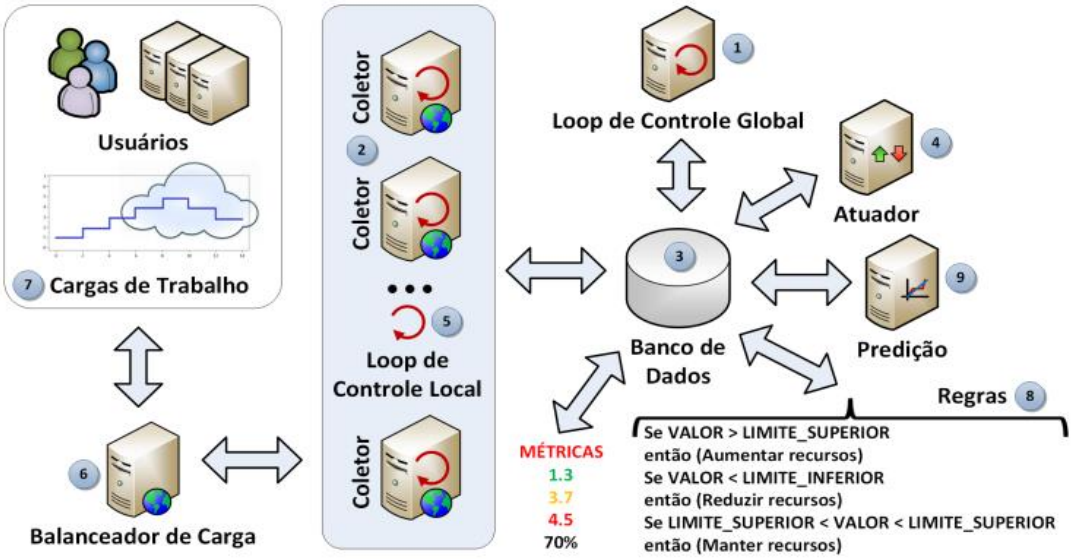
a) **loop de Controle Global**: organiza as atividades relacionadas à elasticidade. É o gerente do ambiente. Dispara eventos de coleta, predição, análise e ações;
b) **coletor**: mecanismo para coleta de dados do ambiente. Sua ação é recuperar dados sobre recursos e armazená-los no banco de dados;
c) **banco de dados**: repositório de informações do ambiente. Armazena os dados, métricas e configurações do ambiente;
d) **atuador**: executa ações de adição e remoção de máquinas virtuais no balanceador de carga;
e) **loop de controle local**: gerencia as operações locais de coleta de dados e consolidação;
f) **balanceador de carga**: provê a distribuição das requisições entre as máquinas virtuais do ambiente;

g) **cargas de trabalho**: carga de trabalho gerada por usuários e suas aplicações, por traços computacionais ou por benchmarks sintéticos;

h) **regras**: definem limites de recursos do ambiente a serem monitorados. Limiares de qualidade possuem um limite superior, inferior e intermediário, onde dependendo do valor coletado, ações serão executadas;

i) **predição**: mecanismo baseado em técnicas, como média móvel e regressão multilinear, que procura prever eventos de adição ou remoção de recursos do ambiente, com a intenção de evitar quebras no SLA e ociosidade.

Figura 24 - Provisionamento Preditivo em Nuvem - Interesse na pesquisa



Fonte: Coutinho *et al. (2014).*

3.4 Arquiteturas preditivas – usando LSTM-RNN

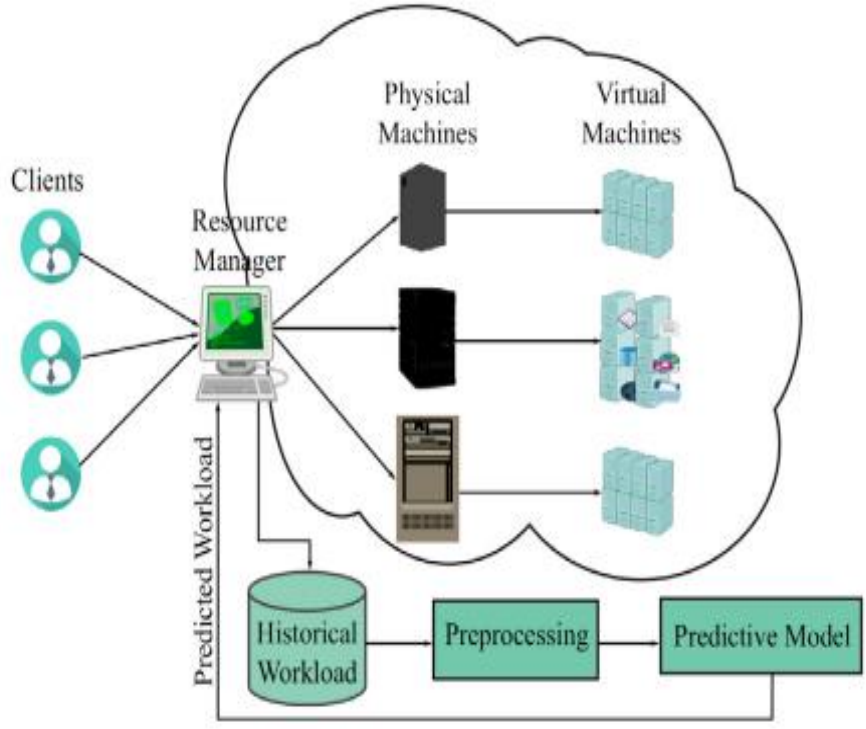
As informações sobre a carga de trabalho de um sistema apresentam parâmetros essenciais para o provimento de um escalonamento dinâmico de recursos. A eficiência no escalonamento de recursos torna o sistema rentável, reduzindo o consumo de energia ao desativar recursos que não foram utilizados. Dessa forma, o sistema também se tornará ecologicamente correto.

Serão apresentados a seguir 03 experimentos selecionados dentre as pesquisas que utilizam LSTM. Esses experimentos apresentam objetivos assemelhados aos propostos nesta pesquisa.

3.4.1 Kumar et al.

No modelo proposto por Kumar *et al*. (2018), mostrado na Figura 25, a saída da unidade preditiva é alimentada por um dispositivo chamado *Resource Manager* que também considera o estado atual do DC antes de tomar decisões de dimensionamento de recursos. O modelo foi implementado em Python, com a biblioteca Keras. Foram realizados experimentos em três conjuntos de dados **D1-NASA** (trace das requisições HTTP dos servidores WWW da NASA - Kennedy Space Center na Florida), **D2-Calgary** (contendo aproximadamente um ano das requisições HTTP dos servidores WWW da *University of Calgary's Department of Computer Science* - Alberta-Canada) e **D3 – Saskatchewan** (sete meses de logs HTTP dos servidores WWW da University of Saskatchewan (Canadá)). As requisições HTTP são armazenadas em arquivos ASCII com uma linha para cada requisição. Os atributos dos dados são: *hosts, timestamp, request, HTTP reply code and bytes no reply.*

Figura 25 - Modelo de previsão de carga de trabalho



Fonte: Kumar *et al*. (2018).

A Tabela 1 apresenta dados da métrica MSE (*Mean Squared Error*) para LSTM-RNN, *Blackhole e Back Propagation*, obtida sobre os dados de teste para todos os experimentos. O erro mínimo alcançado para D1-NASA, D2-Calgary e D3-Saskatchewan são 4,79; 3,42 e 3,17, respectivamente. A precisão do modelo proposto é comparando com métodos de

previsão baseados em algoritmos *Blackhole* e *Back Propagation*. Os resultados dos experimentos mostram claramente que o modelo de previsão baseado em LSTM-RNN supera as demais abordagens.

Tabela 1 - *Mean Squared Error* do Modelo

	Mean Squared Error								
PWS[a]	**LSTM**			**Blackhole**			**Back Propagation**		
	D1	**D2**	**D3**	**D1**	**D2**	**D3**	**D1**	**D2**	**D3**
1	13.06	3.42	5.00	21.45	6.03	1.61	243.17	297.31	40.23
5	479	4.10	3.17	8.45	5.99	2.51	302.25	290.01	336.95
10	6.66	6.11	5.26	12.03	14.78	5.55	281.80	286.55	290.40
20	7.01	5.99	5.56	7.78	12.50	8.82	338.06	278.27	318.29
30	6.43	7.12	4.79	9.06	17.77	8.03	278.42	500.08	507.30
60	5.59	8.03	4.50	23.13	19.70	9.30	333.85	297,02	286.33

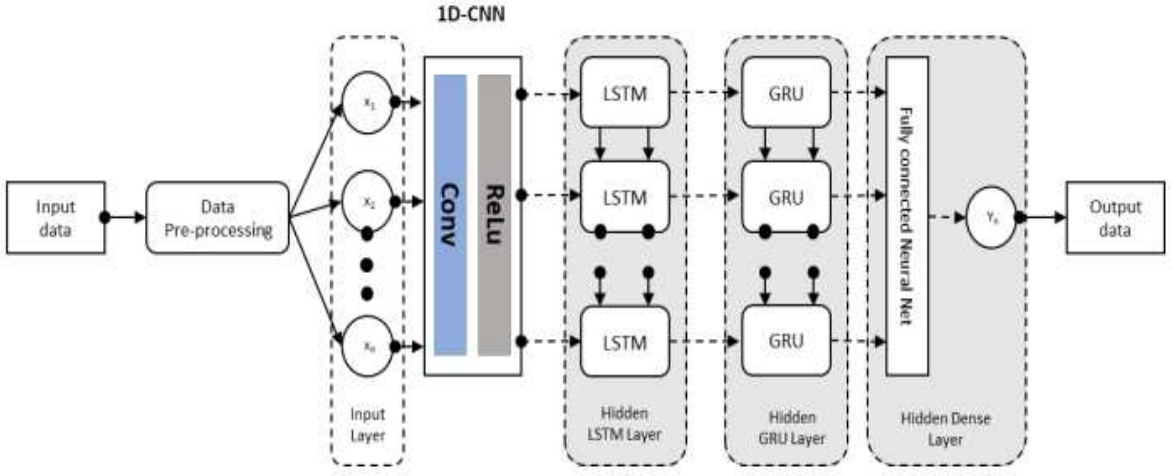
[a] tamanho da janela de predição (prediction window size)

Fonte: Kumar *et al.* (2018).

3.4.2 LSRU - Modelo híbrido entre LSTM e GRU

Shuvo *et al*. (2020) propôs um modelo híbrido denominado LSRU: *Novel Deep Learning based Hybrid Method to Predict the Workload of Virtual Machines in Cloud Data Center,* enfocando a predição em recursos, como uso de CPU, largura de banda, disco e memória. A Figura 25 apresenta a arquitetura proposta. O modelo LSRU é uma combinação dos modelos *Long Short Term Memory* (LSTM) e *Gated Recurrent Unit* (GRU) com a camada de convolução 1D-CNN (Kiranyaz *et al*., 2021) no topo.

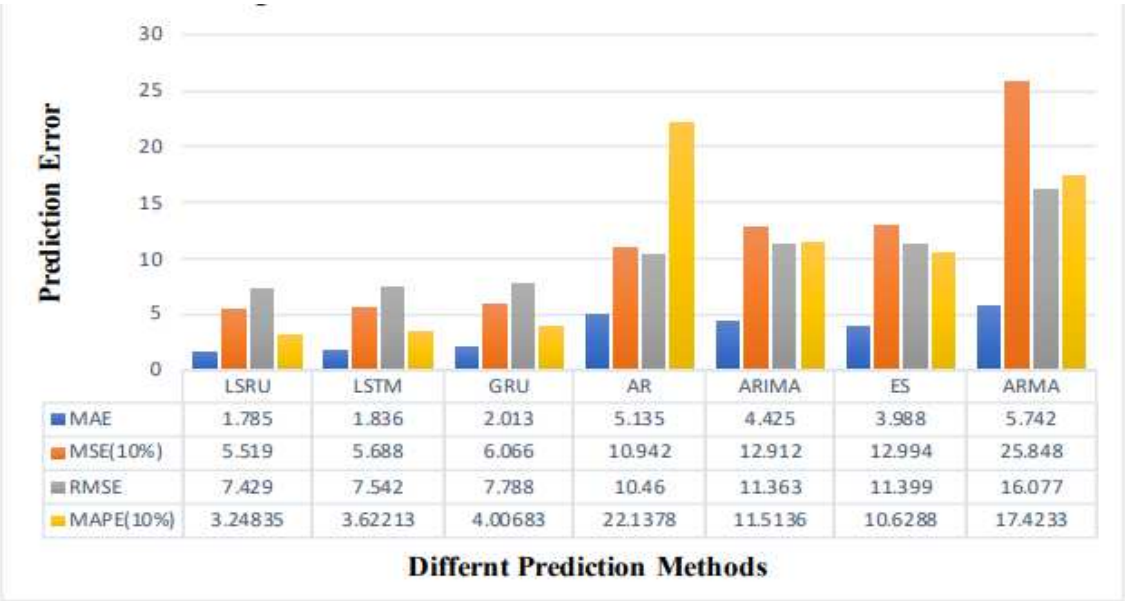
Figura 26- Arquitetura do Modelo LSRU



Fonte: Shuvo *et al*. (2020).

Os autores realizaram comparativo do desempenho do modelo com métodos estatísticos aplicados à análise de séries temporais como AR (Autorregressão), ES (Suavização Exponencial), ARMA (Média Móvel Autoregressiva) e ARIMA (Média Móvel Integrada Autoregressiva), bem como os modelos LSTM e GRU individualmente. A Figura 26 apresenta a comparação realizada. Os autores consideram que o método proposto pode realizar a predição para CPU, disco, memória e largura de banda para curto e longo prazo.

Figura 27- Comparação dos Modelos de Predição



	LSRU	LSTM	GRU	AR	ARIMA	ES	ARMA
MAE	1.785	1.836	2.013	5.135	4.425	3.988	5.742
MSE(10%)	5.519	5.688	6.066	10.942	12.912	12.994	25.848
RMSE	7.429	7.542	7.788	10.46	11.363	11.399	16.077
MAPE(10%)	3.24835	3.62213	4.00683	22.1378	11.5136	10.6288	17.4233

Fonte: Shuvo *et al*. (2020).

3.4.3 Stacked LSTM para predição de ocupação de estacionamento em Birmingham

Jose; V, (2021) propuseram um modelo utilizando a rede neural **Stacked LSTM** para prever a taxa de ocupação de estacionamento na cidade de Birmingham. Os autores consideram que a Stacked LSTM, com suas camadas ocultas aumentam a precisão da previsão. O modelo foi avaliado comparativamente com outros modelos tradicionais de previsão de séries temporais, tais como ARIMA (*Autoregressive integrated moving average*) e SARIMA (*Seasonal ARIMA*). O modelo proposto superou em precisão por uma margem considerada significativa, conforme pode ser visto na Tabela 2 - Comparação de Performance, que enfatiza os resultados obtidos por meio das métricas MAE (*Mean Absolute Error*) e RMSE (*Root Mean Square Error)*. Os autores consideram que as informações meteorológicas e a velocidade do tráfego podem influenciar o comportamento de estacionamento. No futuro, pretendem incorporar esses parâmetros como entrada para o sistema para uma maior precisão de previsão.

Tabela 2 - Comparação de Performance

Modelo	Mean Absolute Error (MAE)	Root Mean Square Error (RMSE)
ARIMA	5,53	12,2
SARIMA	4,98	9,81
Stacked LSTM	4,21	9,02

Fonte: Jose; V, (2021).

3.5 Síntese das arquiteturas preditivas usando LSTM

Na seção 3.4 anterior, foram descritas características de 03 arquiteturas preditivas que utilizam a rede LSTM, apresentadas por terem objetivos assemelhados aos desta pesquisa. Para uma melhor explicitação, o Quadro 4 apresenta uma síntese dessas características.

Quadro 4 - Características essenciais das arquiteturas preditivas avaliadas

CARACTERÍSTICAS	1 - Kumar *et al.* (2018)	2 – LSRU Shuvo *et al.* *(2020)*	3 - Jose; V (2021)
Objetivo	Previsão de carga de trabalho em *Data Center*	Previsão de carga de trabalho de máquinas virtuais em *Data Center*	Taxa de ocupação de estacionamento.
Autonomicidade	Não informado	Não informado	Não informado
Predição	LSTM	LSTM - GRU	Stacked LSTM
Bases Utilizadas	D1- NASA D2- Calgary D3-Saskatchewan	GWA-T-12 Bitbrains	Parking Birmingham dataset
Comparação de Performance	Algoritmos Blackhole Back Propagation	LSTM – GRU AR – ARIMA - ES - ARMA	ARIMA - SARIMA

Fonte: Elaborado pelo autor.

4 ASPECTOS METODOLÓGICOS

A pesquisa realizada envolveu a triangulação entre revisão de literatura, observação, proposta de uma arquitetura (validada por meio de um experimento e um estudo de caso em empresa real), envolvendo ainda os atores da gestão de um provedor *IaaS*. Informações sobre os processos de gerenciamento de recursos adotados no provedor avaliado foram obtidas por meio de pesquisa documental,

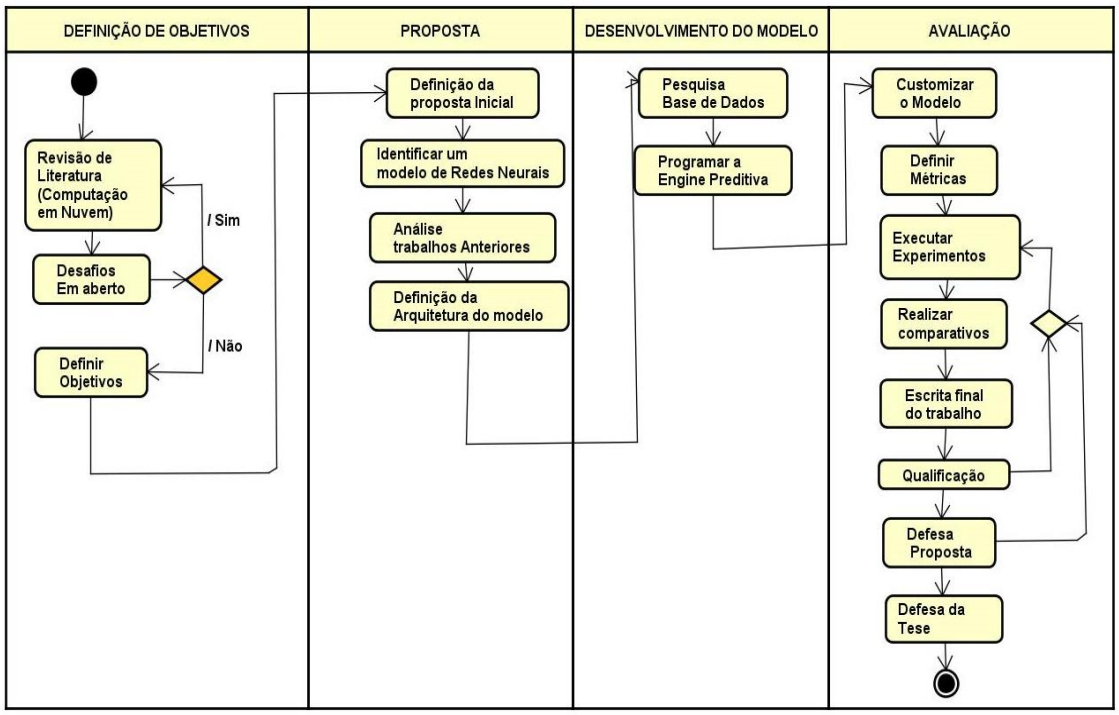
A realização desta pesquisa objetivou trabalhar as questões de pesquisa e os objetivos geral e específicos estabelecidos. Numa visão geral, a metodologia definida para a consecução das atividades apresenta quatro segmentos. Cada um deles apresenta um conjunto de atividades que estabelecem o sequenciamento das etapas metodológicas, conforme pode ser observado na Figura 28.

No primeiro segmento, descreve-se o processo de **definição do problema alvo da pesquisa** – compreendendo a revisão de literatura para identificação das fragilidades da computação em nuvem, foco inicial do trabalho e a definição dos objetivos deste trabalho. O segundo segmento trata da **Proposta** e, consequentemente, no terceiro segmento tem-se o Desenvolvimento da Proposta (detalhamento das soluções para alcançar os objetivos delimitados). Por fim, o quarto segmento apresenta o processo de **Avaliação do modelo** proposto.

Detalhando-se os segmentos, tem-se que na revisão da literatura apresenta as tarefas relacionadas à investigação dos desafios da computação em nuvem ao tempo em que buscar melhor usufruir os benefícios e, simultaneamente, para ajudar na definição dos objetivos desta tese. Além disso, uma revisão de literatura secundária foi realizada para investigar temas relacionados ao objetivo principal desta pesquisa, ou seja, que ajudam a propor soluções para o processo. Esta atividade teve como base *surveys* e demais pesquisas realizadas pelo autor em bases de dados on-line, como IEEE Xplore (https://ieeexplore.ieee.org), ACM DL (https://dl.acm.org), Science Direct (https://www.sciencedirect.com) e Google Scholar (https://scholar.google.com/intl/en-US/scholar/about.htm). As técnicas que apresentaram características desejáveis para o objetivo deste estudo foram registradas e detalhadas em separado. A consolidação desta revisão é apresentada nos Apêndices C e D.

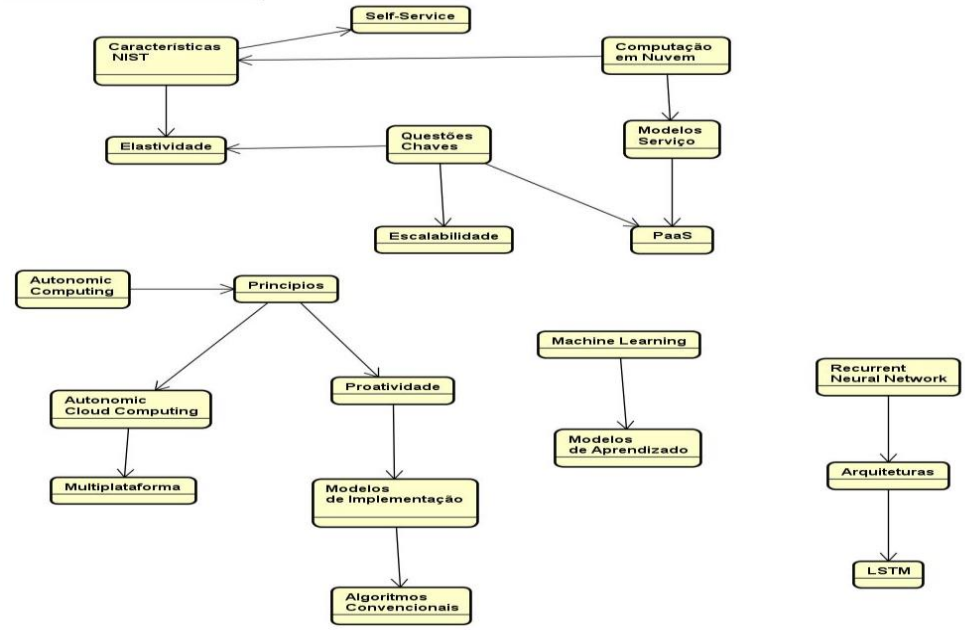
A última fase, nomeada de **Avaliação**, mostra as atividades referentes a avaliação das propostas especificadas no terceiro segmento. Por fim, uma comparação dessas soluções é realizada conforme os resultados obtidos nos experimentos.

Figura 28- Metodologia de Pesquisa do Projeto



Fonte: Elaborada pelo autor.

Figura 29 - Visão Conceitual – Síntese



Fonte: Elaborada pelo Autor

5 ARQUITETURA PROPOSTA

Não existe nada de audacioso sem a desobediência às regras. ***Jean Cocteau***.

Neste capítulo será apresentada a arquitetura proposta. De princípio, buscou-se a cristalização das pesquisas em autonomicidade e no escalonamento preditivo e reativo de recursos, em plataforma em nuvem. Desta forma, a proposta utiliza os avanços das pesquisas em *Machine Learning*, especificamente, em *Deep Learning,* para apresentar sua efetividade no provisionamento autonômico de recursos em nuvem.

Foram elencadas as seguintes características subjacentes aos objetivos da proposta:

a) integrar a computação em nuvem com aprendizado de máquina. Duas áreas em grande crescimento na computação.
b) mostrar a viabilidade no uso de **variantes** das redes **LSTM** para previsão de séries temporais, apresentando a acurácia do modelo (desempenho x custos).
c) viabilizar a autonomicidade no *Workload* da infraestrutura em nuvem, disponibilizando uma arquitetura para o **Gerenciamento de Capacidade** em provedores de serviços em nuvem. A arquitetura apoia provedores de pequeno e médio porte de serviços em nuvem no âmbito regional.
d) integrar o *engine* do escalonamento preditivo, implementado em *Deep Learning*, com a arquitetura para o **Gerenciamento de Capacidade** proposta.

5.1 Bases teóricas

5.1.1 Aspectos formais relacionados ao negócio

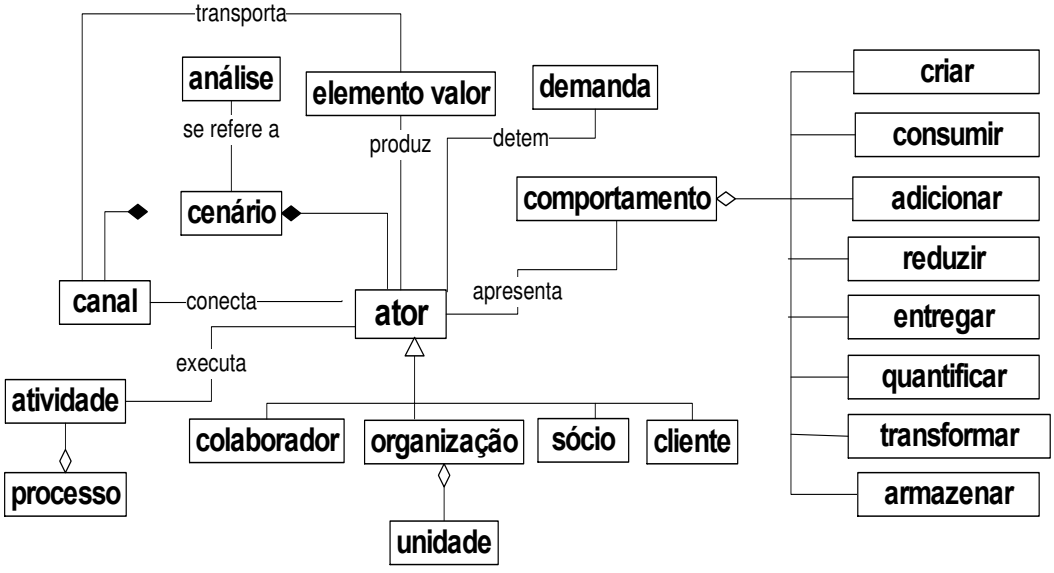
Com base na revisão bibliográfica e na observação de cenários reais de negócio, os processos produtivos se encontram fortemente apoiados em serviços de infraestrutura (*IaaS)*. Foram identificadas características desejáveis para desenvolvimento de uma arquitetura para dar suporte ao negócio, considerando-se o foco estabelecido para o gerenciamento de recursos que apresente contribuição relevante para a área em pesquisa.

Essas características são enumeradas na sequência, como requisitos definidos para a arquitetura proposta. Os requisitos foram consolidados, na medida em que o trabalho foi desenvolvido, de acordo o processo cíclico que compõe a metodologia adotada na pesquisa.

a) **a arquitetura deve mostrar aspectos formais** - a descrição da arquitetura deve ser expressa por meio de uma linguagem sem ambiguidades, visando um melhor entendimento de sua concepção.
b) **a arquitetura deve fornecer um método de predição** - a arquitetura deve fornecer um método consistente para predição de alocação de recursos em um contexto particular. Com base neste método, a arquitetura permite a comparação quantitativa e qualitativa de um conjunto de contextos analisados.
c) **a arquitetura deve capturar os recursos entregues -** a arquitetura deve capturar as particularidades de diferentes contextos de negócio que envolvem a diversidade de clientes de serviços.
d) **a arquitetura deve ser simples** - a aplicação da arquitetura deve ser simples, na execução da predição de recursos de forma autonômica, para dar suporte às decisões requeridas pelos gestores.

A visão de entrega de serviços *IaaS* utilizada na base conceitual da arquitetura foi baseada na abordagem de entrega de serviços proposta por Oliveira (2010) *apud* Fenner (2019), apresentada na Figura 30.

Figura 30 - Entidades do modelo que envolvem a entrega de serviços



Fonte: Oliveira (2010).

As seguintes entidades propostas por Oliveira (2010), foram consideradas para o desenvolvimento da arquitetura apresentada nesta Tese:

a) **cenário** - o contexto em que uma análise é realizada, composto de um conjunto de atores e os canais que os conectam.
b) **análise** - a identificação e quantificação das transferências de valor que ocorrem dentro de um cenário.
c) **ator** - entidade capaz de criar, transformar, armazenar, adicionar (agregar), consumir (fazer desaparecer), transformar e entregar valor aos negócios. Um ator pode ser um cliente, um colaborador ou uma organização (ou uma parte dela – unidade).
d) **elemento valor** - qualquer contribuição entregue **a um** ator e que seja capaz de satisfazer uma necessidade ou atender a uma expectativa deste ator.
e) **canal** - uma relação de conectividade entre atores, através da qual a entrega de valor se faz possível. Só quando há um canal entre dois atores é que o valor pode ser transferido de um para outro.
f) **demanda** - um elemento do qual um ator tem necessidade para cumprir um ou um conjunto de seus objetivos.
g) **comportamento** - é o conjunto de operações possíveis de serem executadas sobre o valor por um ator:
   - **criar** (cria um elemento),
   - **consumir** (destrói um elemento),
   - **adicionar (**aumentar o valor de um elemento),
   - **reduzir** (reduz o valor de um elemento),
   - **entregar** (oferece um contra elemento para outro ator),
   - **quantificar** (expressar em valores numéricos),
   - **transformar** (modifica o valor),
   - **armazenar** (adiciona um elemento valor para o conjunto de elementos detido por um ator).

5.1.2 Redes Neurais

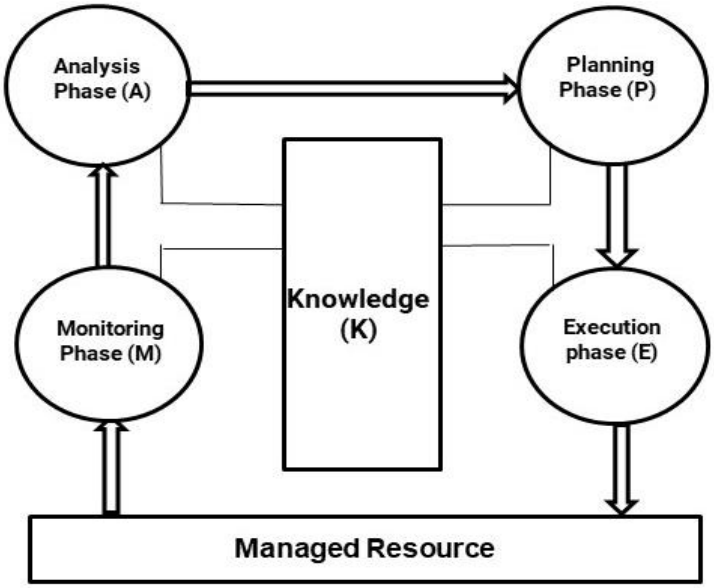
As redes *Long Short-Term Memory* (LSTM) vêm apresentando resultados notáveis em problemas na área de *Machine Learning.* Notadamente vêm apresentando resultados positivos na previsão de cargas de trabalho em infraestrutura em nuvem, utilizando a abordagem de séries temporais. Muitas variantes foram propostas às redes LSTM, incluindo mecanismos e abordagens específicas para determinadas aplicações no alvo deste estudo.

> *[the success of deep neural networks] is commonly attributed to the hierarchy that is introduced due to the several layers. Each layer processes some part of the task we wish to solve and passes it on to the next. In this sense, the DNN can be seen as a processing pipeline, in which each layer solves a part of the task before passing it on to the next until finally, the last layer provides the output.* Hermans; Schrauwen, 2013.

5.1.3 Autonomicidade

A filosofia adotada para fornecer autonomicidade na arquitetura proposta foi a MAPE-K - *Monitoring phase (M), Analysis phase (A), Planning phase (P), Execution phase (E), Knowledge (K).* As quatro fases são executadas regularmente em intervalos de tempo específicos. Por assumir uma clara separação entre as fases do modelo MAPE-K, será considerada a visão multiagentes (Arcaine *et al.* 2015). A Figura 31 apresenta o esquema diagramático do loop MAPE-K mostrando o sequenciamento das ações.

Figura 31- Modelo Loop MAPE-K



Fonte: Elaborada pelo autor

5.2 Modelo arquitetônico proposto

A Figura 32 apresenta uma síntese das principais atividades da arquitetura proposta no contexto do loop MAPE-K. O detalhamento dessas etapas será apresentado decorrer dessa seção.

Na **Fase de Monitoramento (M – Monitoring Phase)** serão coletadas as informações sobre os recursos e a carga de trabalho utilizada. Nesta fase o auto escalonador deverá monitorar os indicadores de desempenho (QoS, SLA etc.) especificados para determinar a necessidade de operações de dimensionamento.

Na **Fase de Análise (A – *Analysis Phase*)** as informações coletadas serão utilizadas para estimar a utilização futura de recursos. Nesta fase, o auto escalonador determinará se é necessário realizar ações de dimensionamento, conforme as informações monitoradas. As seguintes questões deverão ser consideradas:

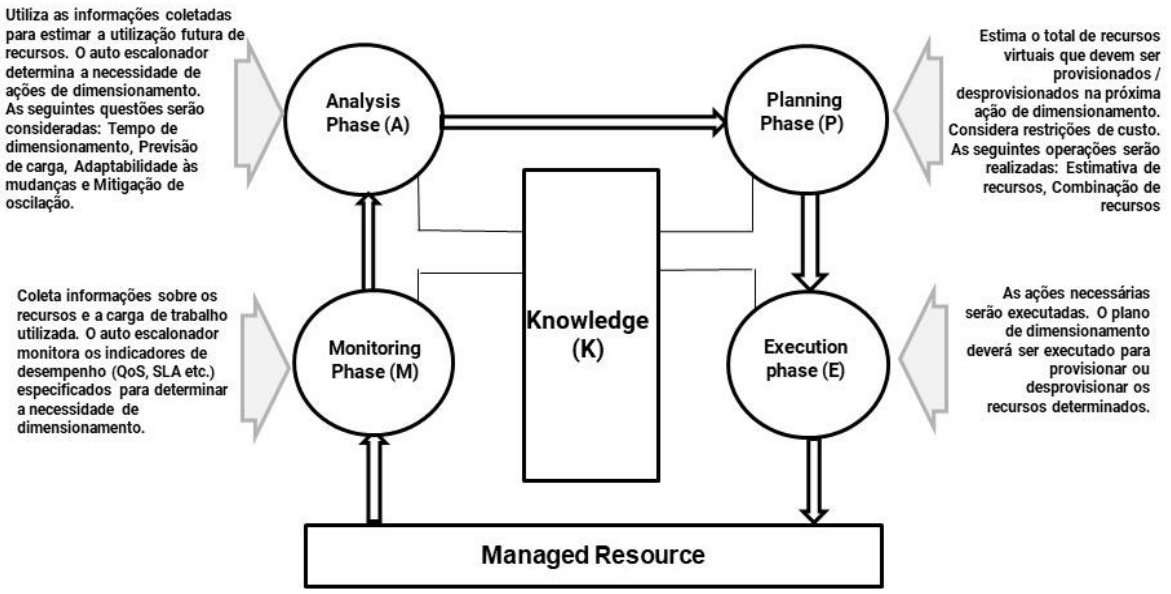
a) **tempo de dimensionamento** - o auto escalonador decidirá sobre a ação de dimensionamento. Ele poderá provisionar ou desprovisionar os recursos de forma reativa/proativa;
b) **previsão de carga** - se o auto escalonador for proativo, a carga deverá ser prevista com precisão;
c) **adaptabilidade às mudanças** - o auto escalonador deverá lidar com as mudanças e adaptar o modelo, de forma oportuna, à nova situação;
d) **mitigação de oscilação** - a oscilação de escala acontece quando o auto escalonador executa ações opostas em um curto período. Esse problema causa alto desperdício de recursos.

Na **Fase de Planejamento (P - *Planning Phase*)** é determinada a ação adequada de recursos que deverão ser alocados (ampliar ou reduzir). Nesta fase é estimado o total de recursos virtuais que devem ser provisionados / desprovisionados na próxima ação de dimensionamento, considerando restrições como custo monetário. As seguintes operações serão realizadas:

a) **estimativa de recursos**: o planejamento deverá estimar quantos recursos serão suficientes para lidar com a carga atual ou de entrada. O auto escalonador precisa determinar os recursos necessários sem poder realmente executar o plano de dimensionamento para observar o desempenho real do aplicativo e deve considerar o modelo de implantação do aplicativo específico nesse processo.
b) **combinação de recursos**: para provisionar os recursos poderá ser utilizado o dimensionamento vertical ou dimensionamento horizontal. Se o dimensionamento horizontal for empregado, pois os CSPs oferecem vários tipos recursos, o auto escalonador poderá escolher um deles.

Na **Fase de Execução (E – *Execution Phase*)** as ações necessárias serão executadas. Esta fase (última etapa) o plano de dimensionamento deverá ser executado para provisionar ou desprovisionar os recursos determinados.

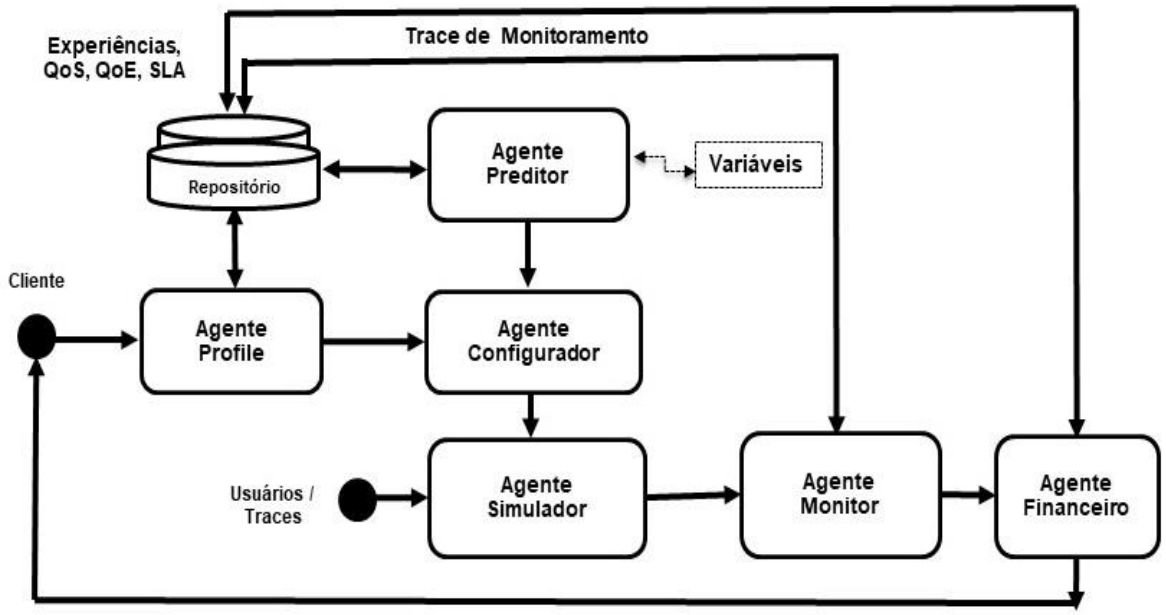
Figura 32- Principais Atividades - Loop MAPE-K



Fonte: Elaborada pelo autor

Utilizando a abordagem *Multi-Agent System* (MAS), a arquitetura será composta por 6 Agentes e 2 Repositórios. A Figura 33 apresenta a arquitetura de alto nível da proposta.

Figura 33- Arquitetura de alto nível da proposta



Fonte: Elaborada pelo autor.

a) Agentes do modelo

a.1 Agente de Planejamento de *Workload* Inicial (Agente Profile) - identifica os recursos necessários para a aplicação, a partir de métricas específicas iniciais, necessárias e desejadas da aplicação. Observa-se que para este dimensionamento são utilizadas as métricas QoE, QoS e o SLA pré-estabelecidos. A chegada de um novo trabalho (pela primeira vez) ativa este agente. Os recursos selecionados serão então alocados inicialmente para execução. As informações de chegada do trabalho (definidas) serão também armazenadas no **repositório de Monitoramento**. As informações de carga de trabalho desse repositório (profile) serão utilizadas tanto para dimensionamento real quanto para o dimensionamento preditivo. As seguintes caraterísticas deverão ser observadas neste módulo:

- As ações deste agente serão ativadas na chegada de um novo trabalho (pela primeira vez);
- Por perguntas deverá ser identificado os recursos necessários para a aplicação, a partir de métricas específicas iniciais, necessárias e desejadas da aplicação. Ou seja, a aplicação define algumas características de recursos necessários para seu processamento.

- Serão utilizadas as métricas QoE, QoS e o SLA pré-estabelecidos.
- Os recursos selecionados serão alocados inicialmente para execução.
- As informações definidas serão armazenadas inicialmente no **repositório de Monitoramento**.
- As informações de carga de trabalho desse repositório (profile) serão utilizadas tanto para dimensionamento real quanto para disparar o dimensionamento preditivo.

a.2 Agentes Preditivo de *Workload* (Agente Preditor) - o agente entra em ação preventivamente por periodicidade ou quando acionado pelo Agente de monitoração na estratégia de alarme. A predição utiliza rede neurais recorrentes (RNN) na arquitetura memória de longo curto prazo empilhada (*STACKED*-LSTM), também denominada *Stacked* Long Short-Memory Network (SLSTMN). A Figura 34 apresenta a estrutura do Preditor na *STACKED LSTM.* A predição utiliza rede neurais recorrentes (RNN) na arquitetura memória de longo curto prazo empilhada (*STACKED*-LSTM), também denominada *Stacked* Long Short-Memory Network (SLSTMN).

a.3 Agente de Configuração (Configurador) – mediante dados da predição de *workload*, este agente define a carga de infraestrutura necessária junto ao *Data center.*

a.4 Agente de Simulação (Simulador) – realiza a simulação de uso dos recursos do *Data Center*. Conforme as necessidades definidas pelo módulo de configuração, o módulo de simulação aloca os recursos necessários em termos de VM, área em disco.

a.5 Agente de Monitoramento (Monitor) - atua com um sensor, capturando informações sobre o desempenho atual da aplicação e registrando ocorrências (variáveis, séries temporais) para predição futura. Alertam quando observam uma previsão de estado crítico. Eles são considerados reativos. Consideram as medidas estabelecidas no SLA e no Gerenciamento de Capacidade (ITIL), bem como as definições recebidas do módulo de predição.

a.6 Agente Financeiro (Financeiro) - efetua o registro financeiro (serviço, data-hora, recurso, quantidade e valor) pela alocação efetiva de recursos. A informação da modalidade deverá ser definida pelo cliente. Deverá ser estabelecido o limite

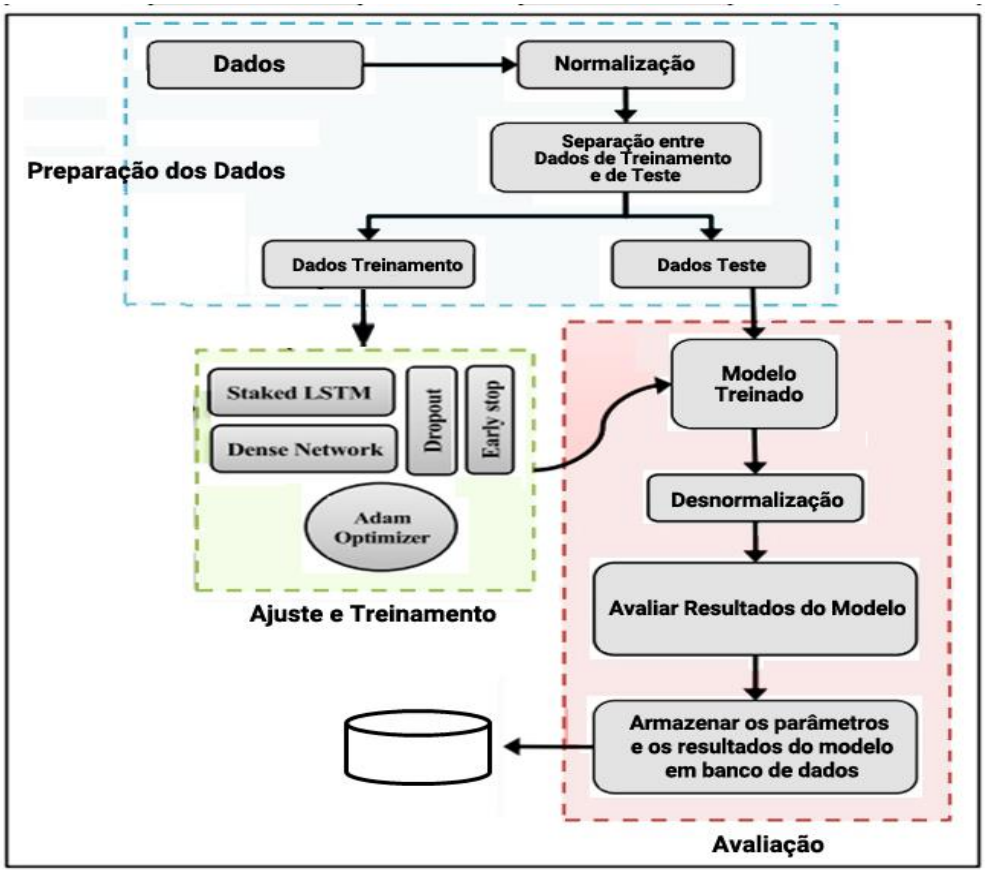
máximo financeiro que o cliente pretende desembolsar, independentemente dos requisitos que o módulo estipulou para a aplicação. Esta situação de *backtrack*, permitirá a remoção dos requisitos até chegar no valor limitante.

b) Repositórios

b.1 Repositório do Profile dos Serviços – mantém informações sobre as condições do contrato de serviço, em termos de SLA, QoS e QoE, bem como limitantes financeiros estabelecidos.

b.2 Repositório de Monitoramento - mantém o registro das informações atuais e fornece entradas para a realização de predições.

Figura 34 - Arquitetura do Preditor em SLSTMN



Fonte: Elaborada pelo autor.

5.3 Avaliação da engine preditiva

Trabalhos relacionados à análise de desempenho utilizaram benchmarks e cargas de trabalho para seus experimentos. Diversos benchmarks e cargas de trabalho foram identificados na revisão bibliográfica e citados no Capítulo 3 – Trabalhos Relacionados.

5.3.1 Descrição do experimento

Para efeito de avaliação da *Engine Preditiva* do modelo, foi utilizado um experimento com as seguintes características:

a) Ambiente:

Software: Anaconda 2.3.2, Jupyter 6.4.12, Python 3.9.13, bibliotecas (Keras 2.6.0, Scikit-learn 1.1.2, Pytorch 1.10.2).

Hardware: Processador 11th Gen Intel Core i7-1165G7@2.80GHz 1.69 GHz - RAM 16,0 GB – SO 64 bits, processador x64.

b) Dataset:

Foi utilizado o *dataset* mantido pela *Delft University of Technology* - GWA-T-12 (http://gwa.ewi.tudelft.nl/datasets/gwa-t-12-bitbrains). O *dataset* contém informações sobre o desempenho de 1.750 VMs do DC *Bitbrains* (provedor especializado em hospedagem gerenciada e computação empresarial). Os clientes da *BitBrains* são bancos internacionais, operadoras de cartão de crédito, seguradoras etc. O DC hospeda aplicativos que geram relatórios financeiros, utilizados predominantemente no final dos trimestres financeiros. A escolha desse *dataset* se deu pela afinidade de conteúdo para esta pesquisa.

A Tabela 3 contém um esboço do conteúdo do conjunto de dados. Cada arquivo contém métricas de desempenho de uma VM específica. O *dataset* possui 02 agrupamentos de traces: *fastStorage e Rnd*.

Tabela 3 - Composição do Dataset

Trace	**#VMs**	**Período**	**#Memória**	**#Núcleos**
Fast Storage	1.250	1 Mês	17.729 Gb	4.057
Rnd	500	3 Mês	5.485 Gb	1.444
Total	1.750	5.446811 Horas de CPU	23.214 Gb	5.501

Fonte: Elaborada pelo Autor

Neste projeto foi utilizado o trace *fastStorage* que consiste em 1.250 VMs conectadas a dispositivos de armazenamento SAN (rede de área de armazenamento rápido). Considera-se suficiente o quantitativo desse trace, bem como a quantidade de variáveis disponibilizadas para simulação. O Quadro 5 apresenta as variáveis do *dataset*, explicitando a unidade das respectivas variáveis.

Quadro 5 - Variáveis do GWA-T-12 da *Bitbrains*

N	VARIÁVEIS	UNIDADE
1	**Timestamp**	número de milissegundos desde 1970-01-01
2	**Núcleos de CPU**	número de núcleos de CPU virtuais provisionadas
3	**Capacidade da CPU provisionada** (CPU solicitada)	A capacidade das CPUs em termos de MHZ é igual ao número de núcleos x velocidade por núcleo
4	**Uso da CPU**	em MHZ
5	**Uso da CPU**	em porcentagem
6	**Memória provisionada** (memória solicitada)	a capacidade da memória da VM em KB
7.	**Uso de memória**	a memória utilizada ativamente em KB.
8.	**Taxa de transferência de leitura de disco**	em KB/s
9	**Taxa de transferência de gravação de disco**	em KB/s
10	**Taxa de transferência recebida pela rede**	em KB/s
11	**Rendimento transmitido pela rede**	em KB/s

Fonte: Elaborada pelo Autor

A Figura 35 apresenta um recorte da base utilizada neste experimento com o intuito de permitir a observação das varáveis disponíveis na base GWA-T-12 da *Bitbrains.* O formato de cada arquivo é baseado em linha, e cada linha representa uma observação das métricas de desempenho. Como acontece na simulação dos modelos de redes neurais, das observações contidas na série temporal disponibilizada no *dataset*, 2/3 são utilizadas para treinamento do modelo e as demais para a predição.

A arquitetura proposta permite que diferentes variáveis sejam utilizadas como entrada, como CPU (foco da presente pesquisa), memória, largura de banda, entre outras. A arquitetura deverá receber previamente a identificação da variável desejada para monitoramento e suas variações.

Figura 35- Recorte da Base Experimental – BitBrains

Timestamp [ms]	CPU cores	CPU capacity provisioned [MHZ]	CPU usage [MHZ]	CPU usage [%]	Memory capacity provisioned [KB]	Memory usage [KB]	Disk read throughput [KB/s]	Disk write throughput [KB/s]	Network received throughput [KB/s]	Network transmitted throughput [KB/s]
1376314846	4	11700.99824	10912.027692428667	93.23333333333333	6.710886E7	6129274.4	0.13333333333333333	15961.6	0.0	2.1333333333333333
1376315146	4	11700.99824	10890.57096202	93.05	6.710886E7	6756624.0	1.3333333333333333	19137.333333333332	0.0	2.6
1376315446	4	11700.99824	10434.11443096	89.15	6.710886E7	8947846.4	2.5333333333333333	19974.933333333334	935.6666666666666	23.933333333333334
1376315746	4	11700.99824	10539.450415120002	90.05	6.710886E7	1.8790479466666665E7	5.466666666666667	8791.8	349.6666666666667	5.466666666666667
1376316046	4	11700.99824	10951.04101983333	93.56666666666666	6.710886E7	9305760.533333333	5.4	15679.533333333333	0.0	2.0666666666666667
1376316346	4	11700.99824	10913.97838800001	93.25	6.710886E7	6352970.4	4.466666666666667	16593.733333333334	0.0	2.0666666666666667
1376316646	4	11700.99824	10855.4383676	92.75	6.710886E7	6129274.4	1.3333333333333333	19182.666666666665	0.0	2.6
1376316946	4	11700.99824	10157.11980594667	86.78333333333335	6.710886E7	9753152.266666668	2.7333333333333334	16234.266666666665	932.6666666666667	24.466666666666665
1376317246	4	11700.99824	10477.029091173334	89.51666666666667	6.710886E7	2.0837962933333334E7	0.0	1.4	295.53333333333333	3.8666666666666667
1376317546	4	11700.99824	11128.551659666669	95.08333333333334	6.710886E7	1.0111066933333334E7	0.0	1.7333333333333334	0.0	1.3333333333333333
1376317846	4	11700.99824	11031.0183412	94.25	6.710886E7	6306231.733333333	0.0	1.4	0.0	1.6
1376318146	4	11700.99824	6485.965691333334	55.41666666666667	6.710886E7	1.1059562533333335E7	2.5333333333333333	10.733333333333333	719.9333333333333	21.466666666666665
1376318446	4	11700.99824	85.82932042666667	0.7333333333333333	6.710886E7	1.0334782E7	0.0	1.3333333333333333	0.0	1.0
1376318746	4	11700.99824	70.22398944000001	0.6	6.710886E7	1476393.6	0.0	1.3333333333333333	0.0	1.0
1376319046	4	11700.99824	76.07596856	0.65	6.710886E7	0.0	0.0	1.3333333333333333	0.0	1.0

Fonte: Elaborada pelo autor.

c) Métricas alvo do experimento:

Várias métricas de avaliação poderiam ser utilizadas para estimar a precisão da previsão dos métodos. Por se tratar de um modelo de regressão, as métricas enfatizadas nesta avaliação foram:

- *Mean Absolute Percentage error (MAPE)*
- *Root Mean Squared Error (RMSE)*
- *Mean Squared Error (MSE)*
- *Mean Absolut Error (MAE)*

Os detalhes sobre a composição de tais métricas, bem com a análise obtida resultante de sua avaliação estão apresentadas na seção *2.3.6 Avaliação da Previsão de Carga de Trabalho.* As métricas foram calculadas pela biblioteca do *Scikit-learn,* por meio das chamadas*:*

- *from sklearn.metrics import mean_squared_error as mse*

sklearn.metrics mean_squared_error (y_true, y_pred)
y_true - Valores-alvo de verdade absoluta (corretos)
y_pred – Valores estimados

- *from sklearn.metrics import mean_absolute_error as mae*

mean_squared_error
sklearn.metrics mean_absolute_error (y_true, y_pred)
y_true - Valores-alvo de verdade absoluta (corretos)
y_pred – Valores estimados

- *from sklearn.metrics import mean_absolute_percentage_error as mape*

sklearn.metrics mean_absolute_percentage_error (y_true, y_pred)
y_true - Valores-alvo de verdade absoluta (corretos)
y_pred – Valores estimados

5.3.2 Etapas do processo de predição

Conforme apresentado na Figura 33 da seção 5.2 - Modelo Arquitetônico Proposto, as etapas definidas para o processo de predição compreendem 03 momentos, a saber: a) preparação dos dados; b) ajuste e treinamento; e c) avaliação.

a) **Preparação dos Dados** - Compreendeu a conversão inicial do dataset para formato de entrada no algoritmo (.csv). Após a leitura dos dados o algoritmo prepara a normalização do dataset. O dataset normalizado é segmentado em dados para teste e dados para treinamento, conforme algoritmo a seguir

```
# split into train and test sets
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]
```

b) **Ajuste e Treinamento -** A partir das experimentações realizadas foi possível confirmar a necessidade da sintonia de todos os parâmetros do modelo utilizados com as características do problema alvo, como o conjunto de dados, a quantidade de camadas e o número de épocas.

```
# criar a ajustar a rede Stacked LSTM
model = Sequential()
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True,
return_sequences=True))
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True))
model.add(Dense(1)),
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])
```

..... o número de camadas é estabelecido o parâmetro **return_sequences=True**
.....foi utilizado o otimizador Adam - método de otimização que calcula as taxas de aprendizagem adaptativa

```
# define as épocas
for i in range(100):
   model.fit(trainX, trainY, epochs=50, verbose=2, shuffle=False)
   model.reset_states()
```

..... os dados de entrada passarão pela rede neural 100 x 50 vezes

```
# realiza a predição
trainPredict = model.predict(trainX, batch_size=batch_size)
model.reset_states()
testPredict = model.predict(testX, batch_size=batch_size)
```

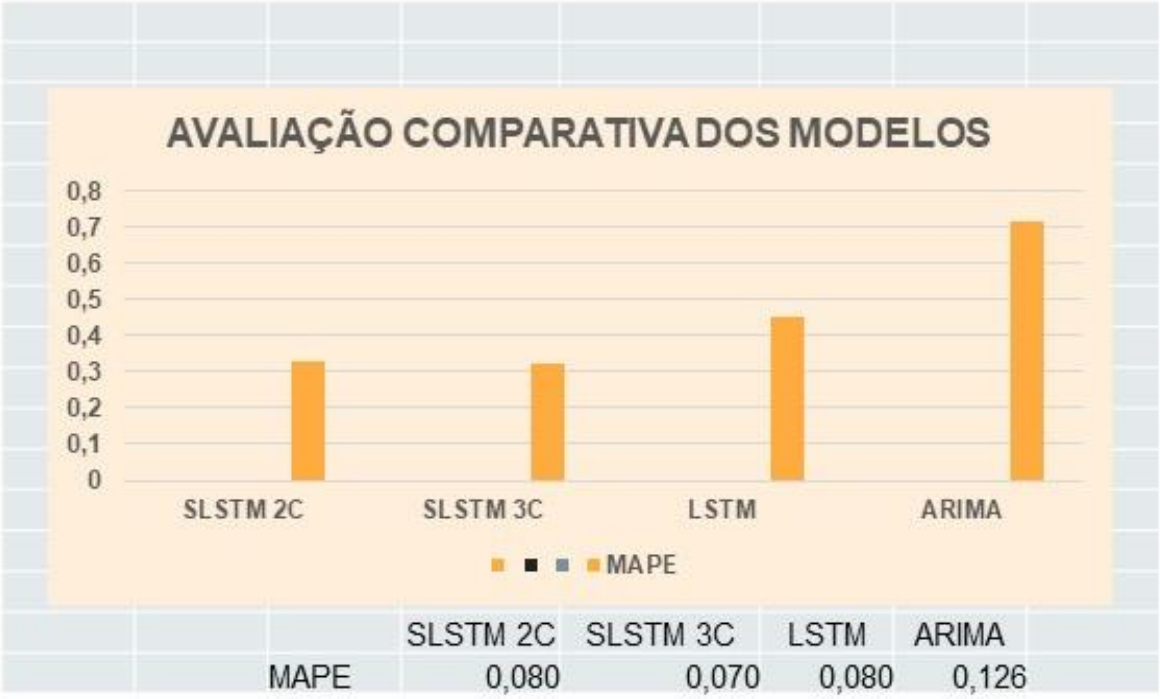
c. Avaliação – Prepara-se os valores da predição para a plotagem.

```
# inverte a predição
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])
```

Neste experimento foram realizadas 03 execuções com o intuito de observar os resultados no uso da rede neural **LSTM x *Stacked* LSTM**, com variações relativas à profundidade das camadas, em 2 e 3 camadas (quantidade do empilhamento da LSTM), bem como as épocas do modelo. Na Figura 36 são apresentadas as métricas obtidas para as diferentes execuções dos modelos ARIMA, LSTM e *Stacked* LSTM, realizadas para demonstrar o desempenho do modelo.

Para análise do resultado das avaliações foi adotada a métrica MAPE por reduz ruídos, sendo sua leitura apresentada por "quanto menor, melhor o desempenho do modelo preditivo". Na Figura 36 observa-se que o menor MAPE é apresentado para o modelo ***Stacked* LSTM** com empilhamento de 3 camadas **(SLSTM 3C - 0,070)**. Portanto esse modelo apresenta uma melhor avaliação quanto à predição dos resultados.

Figura 36 - Comparativos do Desempenho dos Modelos

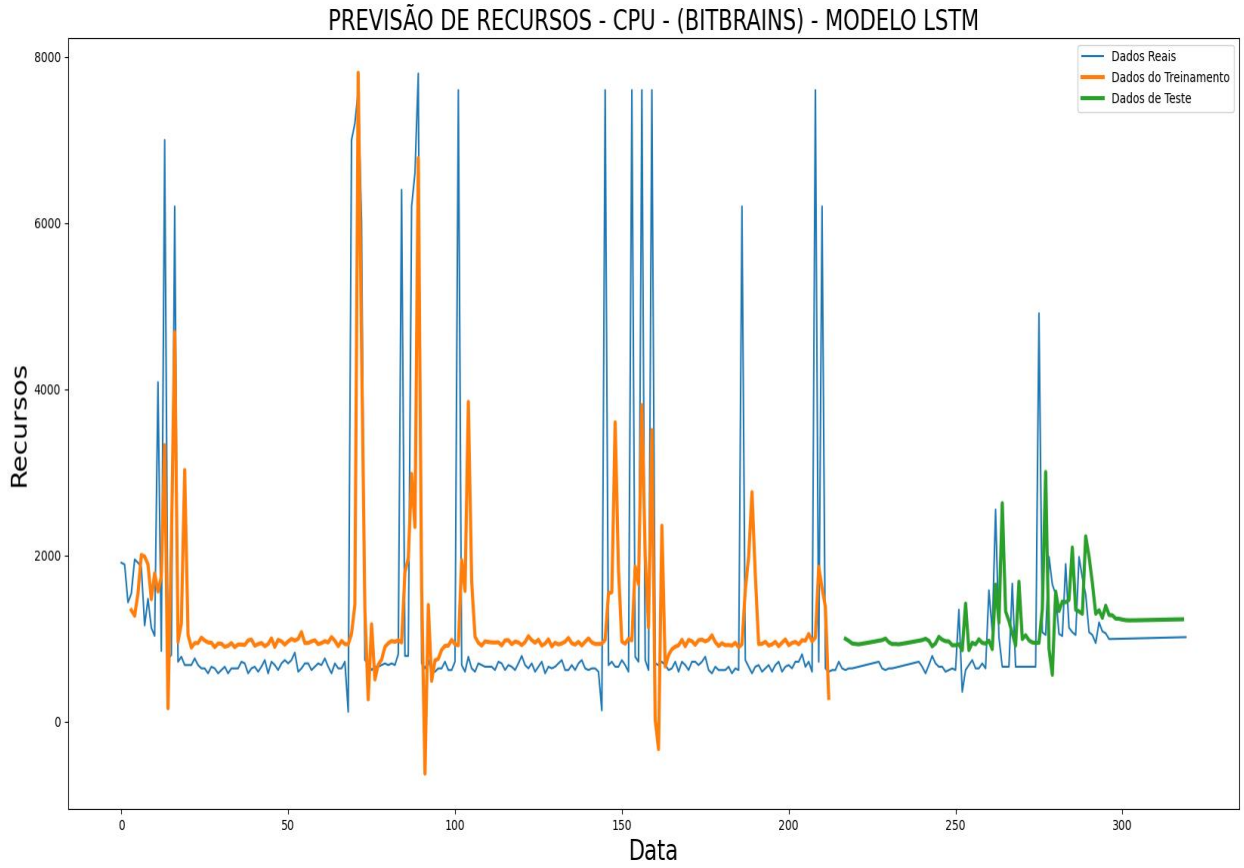


	SLSTM 2C	SLSTM 3C	LSTM	ARIMA
MAPE	0,080	0,070	0,080	0,126

Fonte: Elaborada pelo autor.

A seguir, nas Figuras 37 e 38, tem-se em cada uma delas, 03 gráficos (em três distintas cores) resultante da predição realizada. Em cada uma das figuras são apresentados os valores resultantes da experimentação da LSTM-*Vanilla (01 camada, da SLSTM (2 camadas) e da SLSTM (03 camadas)*. Para cada figura é apresentado o gráfico da cor *azul*, representando os dados reais (obtido diretamente do *dataset*). O gráfico apresentado em *laranja* exibe os dados gerados a partir do treinamento realizado pela rede LSTM (nas respectivas camadas). Em *verde* tem-se a plotagem dos valores disponibilizados pela rede LSTM após o treinamento que corresponde aos valores de previsão alcançado pelo modelo.

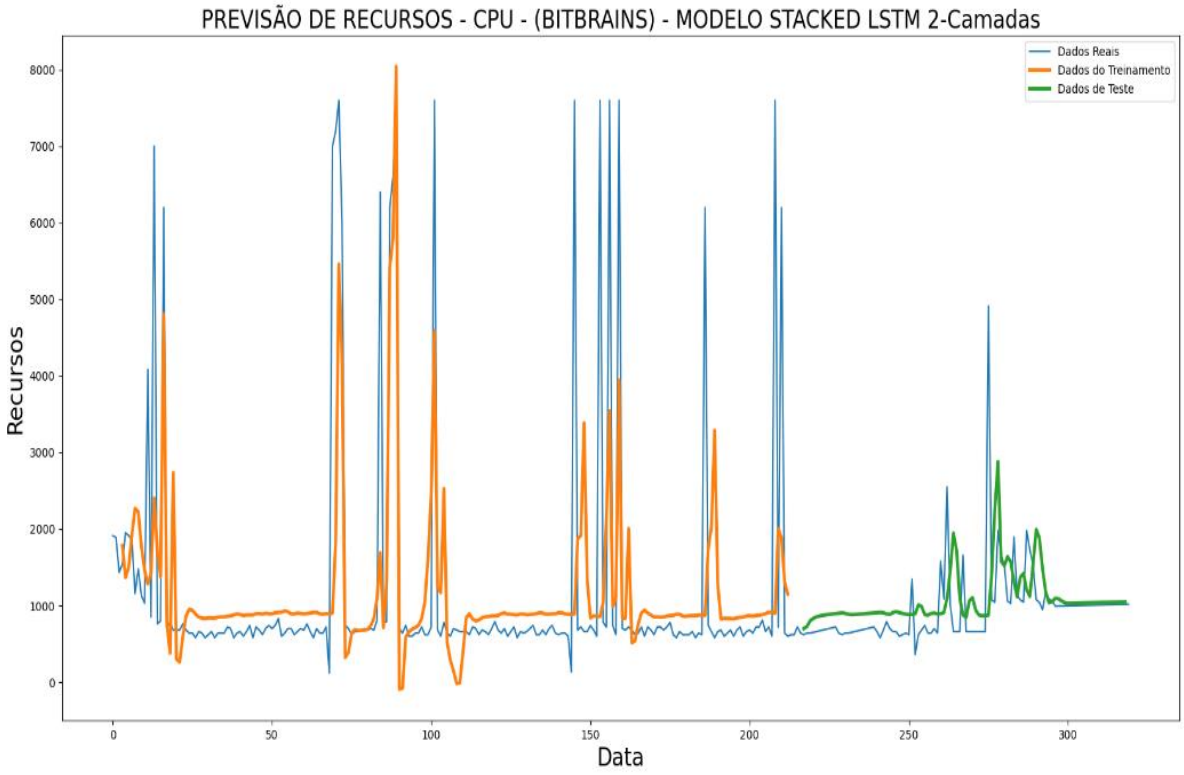
Figura 37- Gráfico plotado experimento – LSTM



Fonte: Elaborada pelo autor.

Na Figura 38 tem-se os gráficos resultantes da predição realizada com a rede **Stacked LSTM** na qual foram utilizadas **02 camadas**. Para esse experimento foi utilizada a variável CPU obtida do dataset (Bitbrains). Inicialmente tem-se o gráfico em *azul* que representa os dados reais (obtidos diretamente do *dataset*). O gráfico apresentado em *laranja* exibe os dados gerados a partir do treinamento realizado pela rede **Stacked LSTM**. Em *verde* tem-se a plotagem dos valores disponibilizados pela rede **Stacked LSTM após o treinamento** que corresponde aos valores de previsão alcançados pelo modelo.

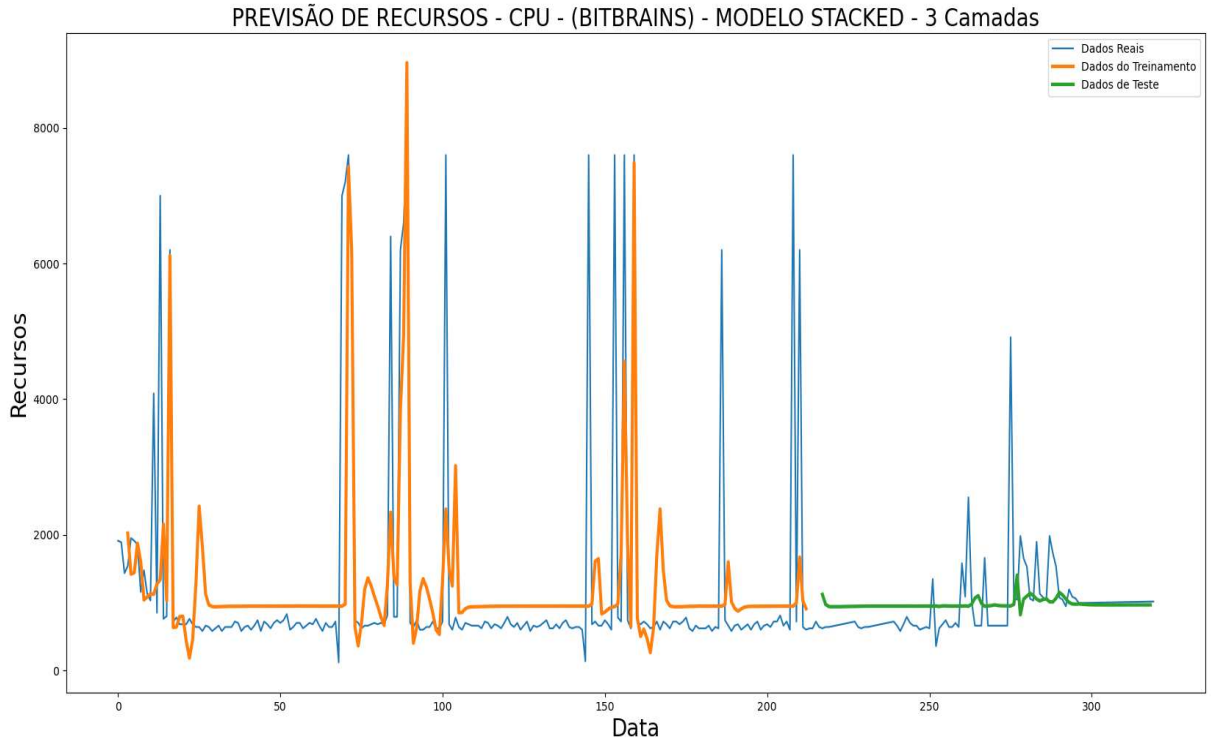
Figura 38- Gráfico plotado pelo experimento Stacked – LSTM – 2 Camadas



Fonte: Elaborado pelo autor.

Na Figura 29 tem-se os gráficos resultante da predição realizada com a rede ***Stacked*** **LSTM** na qual foram utilizadas **03 camadas** de empilhamento com a variável CPU obtida do dataset (Bitbrains). Inicialmente tem-se a plotagem do gráfico em *azul* que representa os dados reais (obtidos diretamente do *dataset*). O gráfico apresentado em *laranja* representa os dados gerados a partir do treinamento realizado pela rede Stacked LSTM. Em verde tem-se a plotagem dos valores disponibilizados pela rede *Stacked* LSTM após o treinamento que corresponde aos valores de previsão alcançados pelo modelo.

Figura 39- Gráfico plotado pelo experimento Stacked – LSTM – 3 Camadas



Fonte: Elaborada pelo autor.

5.4 Validação da proposta

A Questão Primária de Pesquisa foi identificar a tecnologia que apresenta uma melhor acurácia para o provisionamento autonômico de recursos no Gerenciamento de Capacidade em nuvem, considerando os objetivos dos provedores.

5.4.1 Validação da engine preditiva

Como trabalho anteriores foram analisadas diversas pesquisas que demonstram as vantagens das técnicas de redes neurais frente aos tradicionais.

Na avaliação realizada (seção 5.3) comparou-se o modelo **ARIMA (autorregressivo integrado de médias móveis**), considerado tradicional, com relação às técnicas de Redes Neurais, modelo LSTM clássica (*Vanilla*) e modelo *Stacked* LSTM.

Como resultado do experimento, observou-se que na métrica MAPE, a rede LSTM obtêm um melhor resultado preditivo que o modelo ARIMA. A métrica MAPE reduz os ruídos, considerando-se quanto menor o resultado, melhor o desempenho do modelo preditivo. Ressalte-se ainda uma vantagem do modelo LSTM é sua capacidade de realizar análise com um pequeno volume de dados e obter-se um resultado preciso e com um alcance

de longo prazo, devido sua capacidade de memorização.

O estudo avaliou os modelos de Redes Neurais (LSTM x Stacked LSTM) na busca de melhores resultados, considerando-se o desempenho das Redes Neurais Profundas.

Destaca-se a importância de prever a carga futura na computação em nuvem para agendamento eficiente de recursos, alocação e balanceamento de carga em VMs. Para solucionar este problema, esta arquitetura foi proposta. O método proposto tem um desempenho melhor do que métodos estatísticos tradicionais e de IA, conforme apresentado no capítulo 5.3 Avaliação da Engine Preditiva. O método proposto pode trabalhar com varáveis como uso de CPU, disco, memória e largura de banda para previsão de longo prazo.

No futuro, planejamos desenvolver uma estratégia eficiente de alocação de recursos para minimizar o custo de consumo de energia, custo de largura de banda e violação de SLA.

5.4.2 Validação da arquitetura

Entre as estratégias adotadas na validação da arquitetura proposta, foi utilizado o método da **validade de aparência** (Runerson; Host, 2009), ou seja, verificar se a arquitetura aparenta ser útil e preferível em relação à forma de gerenciamento atual utilizada na empresa, acurada para suportar a tomada de decisão no gerenciamento de recursos/capacidade e ainda se a arquitetura proposta aparenta ser eficaz em função dos seus objetivos. Utilizando essa metodologia, a arquitetura proposta e os resultados do experimento foram apresentados a 12 gestores de serviços e especialistas do **Provedor *Alpha***, os quais responderam a um questionário, indicando, em sua visão, se o modelo aparenta ser adequado. O questionário foi construído de modo similar ao apresentado em Lima (2010), para avaliar as hipóteses de pesquisa:

a) **Hipótese 1 (H1) -** a arquitetura proposta é preferível em relação às soluções atuais utilizadas pelo provedor;
b) **Hipótese 2 (H2) -** a arquitetura proposta é útil para suportar o processo decisório;
c) **Hipótese 3 (H3) -** a arquitetura proposta é acurada o suficiente para suportar o processo de gerenciamento de recursos de serviços em nuvem;
d) **Hipótese 4 (H4) -** a arquitetura proposta é eficaz para suportar o processo de gerenciamento de recursos.

Em relação à hipótese de preferência (**H-1**), os resultados obtidos indicaram que 100% dos gestores preferem a arquitetura proposta e apresentada em relação ao método atual utilizado na empresa avaliada. Quanto à hipótese referente à utilidade da arquitetura (**H-2**) para os gestores e especialistas, os resultados obtidos sinalizaram igualmente que 100% dos respondentes consideraram que a arquitetura é útil para subsidiar o processo de alocação de serviços de nuvem. Quanto à terceira hipótese, os resultados obtidos indicaram que 11 respondentes consideraram a arquitetura suficientemente *acurada* (**H-3**) para dar suporte ao processo autonômico no gerenciamento de recursos. Avaliar a eficácia da arquitetura significa estimar o grau com que o seu propósito foi atingido. Para avaliação da eficácia da arquitetura (**H-4**), buscou-se identificar aspectos relacionados à eficácia do módulo de predição no experimento realizado, bem como a disposição lógica de todos os seus módulos.

Para testar as hipóteses nulas, foi utilizado um teste binomial com nível de significância 5% e com 12 tentativas, correspondente ao número de respondentes. Resultado: a hipótese nula (H0) pode ser rejeitada nos testes de hipóteses, quando pelo menos “11 pessoas entre 12 respondem com sucesso", o que corresponde ao percentual aproximado de 90%.

a) Dificuldades enfrentadas e necessidades adicionais

Além das respostas ao questionário, foi realizada uma entrevista não estruturada, onde os gestores avaliaram as vantagens e dificuldades identificadas para uma possível utilização da arquitetura em produção. Apesar da técnica de predição ser uma novidade em termos técnicos, muitos entrevistados enfatizaram a inovação que vislumbraram em trabalhar com previsão autonômica em seus processos de alocação de recursos. Os gestores citaram que os provedores regionais de serviços em nuvem, como é o caso do provedor Alpha, geralmente não têm acesso ao uso de ferramentas mais especializadas, como a proposta que lhe foi apresentada durante o estudo de caso. Todos os gestores e especialistas entrevistados consideraram que seria uma consequência natural a implementação de uma ferramenta de software automatizada para operacionalização da arquitetura.

b) Sugestões e comentários

Os gestores forneceram sugestões e comentários sobre a arquitetura apresentada, sendo citadas, na sequência, as considerações mais relevantes no processo de **validação de**

aparência:

"Com relação a previsão do uso de recursos realizada da arquitetura, entendo que ela é muito útil, possibilitando a implantação de uma cobrança por uso de recursos que possa estar alinhada com a demanda, de uma forma mais precisa" (Gestor Respondente 1).

"Talvez seja interessante ter algum mecanismo/procedimento para garantir que o conjunto de dados de entrada esteja completo. Para garantir a precisão, as bases de dados poderiam estar sempre sendo atualizadas" (Gestor Respondente 2).

"A arquitetura e a pesquisa apresentada chama a atenção por dar suporte aos provedores regionais. Com seu uso, vislumbramos novas oportunidades para concorrer com provedores de serviços de maior porte" (Gestor Respondente 3).

"É uma novidade, um trabalho de pesquisa, que aborda um tema tão importante como a alocação de recursos para dar suporte automático no dimensionamento da capacidade" (Gestor Respondente 4).

"A proposta avaliada permite que tomemos decisões mais sensatas em relação ao uso dos recursos" (Gestor Respondente 5).

6 CONCLUSÕES E TRABALHOS FUTUROS

> *Um dia me disseram que as nuvens não eram de algodão. Sem querer eles me deram as chaves que abrem essa prisão* - **Somos quem podemos ser - Engenheiros do Hawaii.**

O objetivo geral desta pesquisa foi propor uma arquitetura autonômica para alocação preditiva de recursos, que forneça suporte ao processo de gerenciamento de recursos nos provedores de serviços em nuvem (no contexto do gerenciamento de capacidade). Dessa forma, uma arquitetura baseada em redes neurais foi proposta como solução para lidar com a complexidade inerente ao processo de gerenciamento de recursos do tipo *IaaS*, visando aperfeiçoar o entendimento dos problemas envolvidos e gerar *pontos de reflexão* que possam aumentar a eficácia da gestão autonômica de recursos em provedores.

O foco desta tese de doutorado envolveu o gerenciamento autonômico de recursos na área de gerenciamento de serviços, baseado em simulação por meio de redes neurais e tomada de decisão. A arquitetura proposta foi planejada, tendo obtido resultados promissores a partir da validação do seu módulo de predição (arcabouço principal), realizada por meio de experimento científico. O estudo de caso realizado em uma empresa real foi planejado, desenvolvido e validado, em consonância com as recomendações de Runerson e Host (2009). Após a execução do estudo de caso, os dados obtidos foram confrontados com a revisão de literatura e observações realizadas.

A virtualização fornece uma solução eficiente para os objetivos da computação em nuvem, facilitando a criação de Máquinas Virtuais (VMs) sobre os servidores físicos, levando a uma melhor utilização de recursos. Essa tecnologia poderá solucionar um conjunto complexo de tarefas em um menor tempo com a utilização adequada de recursos.

O crescimento da computação em nuvem e, consequentemente, a variedade de serviços ofertados trazem muitas facilidades ao desenvolvimento de software. Garantir a *Quality-of-Service* (QoS), aumentar a taxa de transferência e o retorno dos investimentos são obstáculos que podem ser alcançados pelo gerenciamento eficaz de recursos em nuvem.

O uso eficiente dos recursos é essencial para obter o máximo da tecnologia da nuvem e isso pode ser alcançado mantendo, em um máximo de tempo, os recursos ocupados. Para um servidor em nuvem, nem sempre é possível ter a quantidade adequada de demanda para manter todos os recursos em uso. Para que a nuvem funcione com eficiência, as melhores estratégias de alocação de recursos devem ser empregadas. A utilização de recursos é uma das tarefas de extrema importância no ambiente de nuvem, no qual os trabalhos dos usuários são

agendados para diferentes máquinas. Nesse cenário, a **previsão de carga de trabalho** torna-se útil nas decisões de dimensionamento de recursos. Os resultados alcançáveis com os métodos de previsão de carga de trabalho encorajam as pesquisas na exploração desse domínio e vários esquemas de previsão foram apresentados na literatura na busca de um modelo eficaz, baseado em dados históricos e no tratamento de questões, como flutuação de carga de trabalho e efeitos *Slashdot*.

Com base nos resultados alcançados nesta pesquisa, pode-se concluir que um **modelo de previsão de carga de trabalho** eficaz, não apenas ajuda nas decisões de dimensionamento de recursos, mas também poderá ajudar na promoção da **computação verde**, reduzindo o número de máquinas ativas.

A previsão de carga de trabalho nos serviços nuvem é uma etapa essencial no gerenciamento adequado de recursos e nas abordagens de dimensionamento autonômico que auxiliam os CSPs no provisionamento / desprovisionamento de recursos. Erros de previsão podem causar problemas, reduzindo o desempenho da nuvem e leva a violações de SLAs, gerando desperdício nos recursos. O agendamento eficaz de recursos reduz o custo de execução, o tempo de execução, o consumo de energia e aprimoram outros requisitos de QoS, como confiabilidade, segurança, disponibilidade e escalabilidade.

O **provisionamento reativo** é a técnica usual de provisionamento adotada na computação na nuvem e está majoritariamente presente nas soluções comerciais de pequeno e médio porte. As técnicas reativas também são amplamente abordadas na literatura através do provisionamento automático baseado em informações específicas da aplicação (como tempo de resposta, taxa de chegada de requisições, tipos de requisições etc.) que possuem uma relação direta com a QoS da aplicação.

Um serviço de provisionamento autonômico precisa lidar com métricas não-intrusivas, obtidas no nível da infraestrutura virtual, inviabilizando o uso dessas soluções para este fim. As soluções reativas não-intrusivas assumem que a técnica simplesmente baseada em limiares definidos (*threshold-based*) não é suficiente para assegurar os objetivos de provisionamento. Por esse motivo, essas soluções combinam o uso de regras de provisionamento com outras técnicas de tomada de decisão e atualização de limiares.

Para auxiliar na investigação da solução da questão central desta pesquisa, buscou-se identificar tecnologias que apresentem uma **melhor acurácia para o provisionamento autonômico da carga de trabalho dos serviços em nuvem**, considerando os objetivos dos provedores e das empresas usuárias dos serviços. Foram identificadas propostas com diferentes técnicas, apresentadas no **Apêndice C - Experimentos de Predição**

da Carga de Trabalho. Dentre as soluções adotadas, as que apresentaram uma melhor acurácia, predominantemente utilizam Redes Neurais, com ênfase para as redes LSTM e suas variantes.

Para prover a autonomicidade no provisionamento da carga de trabalho dos serviços em nuvem tem sido preferencialmente utilizado o modelo MAPE-K, na filosofia *Multi-Agent System (*MAS*)*. Utilizou-se esta abordagem na modelagem da arquitetura para o provisionamento de recursos proposta.

Para definir a arquitetura de provisionamento de recursos, de alta acurácia, que pudesse ser utilizado por provedores de serviços em nuvem, foram analisadas diversas pesquisas destacadas no **Apêndice C.** Essas ideias serviram de base para a modelagem da arquitetura proposta, a qual se pretende que seja utilizado por CSPs para gerenciar o orçamento dos serviços de TI.

6.1 Contribuições

As contribuições geradas por esta pesquisa são de nível prático (desenho de uma arquitetura autonômica para alocação de recursos e implementação efetiva de seu módulo de predição) e empírico (validação da arquitetura). A contribuição prática dada por este trabalho para os estudos em *BDIM* foi explorar o potencial que as técnicas de predição baseadas em redes neurais possuem para colaborar com esta área. O processo de validação da arquitetura foi baseado na percepção de sua utilidade por gestores de TI, de provedor *IaaS*. Pode-se elencar ainda as seguintes contribuições específicas a partir desta pesquisa:

a) Proposta de uma arquitetura para gerenciamento de recursos orientada ao negócio possibilita a realização de predição de recursos para suporte ao processo de gerenciamento de capacidade em cenários IaaS;
b) Propor um arcabouço preditivo para alocação de recursos em nuvem utilizando *Stacked Long Short-term Memory Neural Network*.

6.2 Limitações da pesquisa e trabalhos futuros

Como primeira limitação à realização desta pesquisa, aponta-se para restrições do ponto de vista *orçamentário*, de *logística* e, principalmente, de *acesso a provedores IaaS e pessoas dispostas a viabilizar um processo mais rigoroso de validação*. A metodologia aqui adotada para a proposta de uma arquitetura autonômica para alocação de recursos em nuvem é um processo iterativo, que demanda tempo e dedicação de todos os envolvidos. O esforço de

desenvolvimento, verificação e validação também demandou o acesso a dados históricos sobre gerenciamento de recursos. Este acesso foi restrito, tanto pelo caráter estratégico das informações quanto pela falta da cultura do registro por parte das pesquisas e das empresas.

A impossibilidade de generalização dos resultados também é uma limitação desta pesquisa. Em relação à validade de construção (questionários), sempre há dúvidas sobre se as variáveis são bem compreendidas pelos gestores e essa subjetividade leva a uma ameaça: pode-se estar obtendo resultados desassociados da realidade.

Foi possível ainda identificar algumas questões que poderão ser melhor investigadas em estudos futuros:

a) adequar os modelos de previsão não lineares para prever séries temporais com variações sazonais, eles podem ser utilizados para otimizar processos com horizontes de tempo mais longos.
b) explorar os constantes avanços nas pesquisas em técnicas de *Machine Learning* para melhorar o desempenho da previsão de carga de trabalho. Investigar algoritmos de gerenciamento de recursos para utilizar os resultados de previsão alcançados.
c) fornecer melhores esquemas de previsão de carga para reconhecer padrões de solicitação mais realistas e complexos que podem ocorrer.
d) definir novas métricas de previsão de carga de trabalho sobre os atrasos nas previsões de intermitência. Como o custo dos erros de previsão no ambiente de nuvem não é simétrico, a definição de melhores métricas de avaliação deve considerar essa questão.
e) criação de esquemas de previsão de carga de trabalho leves para serem aplicados nas tecnologias emergentes recentemente, como IoT, cloudlets, fog computing e computação de borda móvel, com recursos mais limitados que os DCs em nuvem.
f) integrar os esquemas de previsão de carga com os esquemas de detecção de intrusão para reconhecer os ataques DDoS dos efeitos Slashdot.
g) integrar os esquemas de autoscaling com os sistemas IDS (Sistemas de Detecção de Intrusão) e IPS (Sistemas de Prevenção de Intrusão) para melhor lidar com os ataques DDoS e Yo-Yo. Os sistemas de escalonamento automático convertem os ataques DDoS em ataques EDoS para lidar com comportamentos maliciosos. Reconhecer a carga de trabalho DDoS da carga de trabalho dos usuários é uma questão aberta.

REFERÊNCIAS

AL-DHURAIBI, Yahya; PARAISO, Fawaz; DJARALLAH, Nabil; MERLE, Philippe. **Elasticity in Cloud Computing: State of the Art and Research Challenges**. IEEE Transactions on Services Computing, v. 11, n. 2, mar./abr. 2018.

ALI-ELDIN, Ahmed; TORDSSON, Johan; ELMROT, Erik. **An Adaptive Hybrid Elasticity Controller for Cloud Infrastructures.** IEEE Network Operations and Management Symposium, p.16-20, abr. 2012.

ALOUFI, Omar F.; DJEMAME, Karim; GHABAN, Fahad; SAEED, Faisal**. A survey on predicting workloads and optimizing QoS in cloud computing**. In: Proceedings of the 2021 International Congress of Advanced Technology and Engineering, ICOTEN 2021, Taiz, Yemen, p. 1-7, jul. 2021.

AMIRI, Maryam, MOHAMMAD-KHANLI, Leyli. **Survey on prediction models of applications for resources provisioning in the Cloud.** Journal of Network and Computer Applications, v. 82, p. 93-113, 2017.

ANH, Tuan Le. **Workload prediction for resource management in data centers**. Disponível em: https://umu.diva-portal.org/smash/get/diva2:957163/ FULLTEXT01.pdf. Acesso em: 30 Jan. 2020.

ANTONESCU, A.-F.; BRAUN, T. **Simulation of SLA-based VM scaling algorithms for cloud-distributed applications**. Future Gener. Comput. Syst. v.54, p. 260–273, 2016.

ARCAINI, Paolo; RICCOBENE, Elvinia; SCANDURRA, Patrizia. **Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation,** IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, p. 13-23, 2015.

AWS. **Amazon Cloudwatch - Capacidade de Observação dos seus recursos da AWS e Aplicativos na AWS e no local**. Disponível em: https://aws.amazon.com/pt/cloudwatch/. Acesso em: 21 maio 2021.

AWS. **Pricing Calculator-User Guide**. Disponível em: https://calculator.aws/#/. Acesso em: 30 set. 2022.

AXELOS**. ITIL 4: Acquiring and Managing Cloud Services**. 1st Edition. UK: The Stationery Office, 2021.

BABU, S. Kishore; VASAVI, S.; NAGARJUNA, K**. Framework for Predictive Analytics as a Service Using Ensemble Model.** IEEE 7th International Advance Computing Conference (IACC), p. 121-128, 2017.

BALAJI, Mahesh, KUMAR, Ch. Aswani, RAO, G. Subrahmanya V.R.K. RAO. **Predictive Cloud resource management framework for enterprise workloads**. Journal of King Saud University – Computer and Information Sciences v. 30, p. 404-415, 2018.

BALDAN, F. J., Ramirez-Gallego, S., Bergmeir, C., Benitez-Sanchez, J.M., Herrera, F.: **A**

forecasting methodology for workload forecasting in cloud systems. IEEE Trans. Cloud Comput. v. 6, n. 4, p. 929–941, 2016.

BAN, Tao; ZHANG, Ruibin; PANG, Shaoning; SARRAFZADEH, Abdolhossein; INOUE, Daisuke. **Referential kNN Regression for Financial Time Series Forecasting.** ICONIP 2013, Part I, LNCS 8226, Berlin Heidelberg: Springer-Verlag, p. 601–608, 2013.

BARR, Jeff. **New – Predictive scaling for EC2, powered by machine learning.** Disponível em: https://aws.amazon.com/blogs/aws/new-predictive-scaling-for-ec2-powered-by-machine-learning/ Acesso em: 21 maio 2021.

BENNANI, M. N., MENASCE, D. A. **Resource allocation for autonomic data centers using analytic performance models,** Second International Conference on Autonomic Computing (ICAC'05), p. 229–240, 2005.

BROWNLEE, Jason. **CNN Long Short-Term Memory Networks**. Disponível em: https://machinelearningmastery.com/cnn-long-short-term-memory-networks/ Acesso em 21 ago. 2017.

BROWNLEE, Jason. **How to develop LSTM Models for Time Series Forecasting**. Disponível em: https://machinelearningmastery.com/how-to-develop-lstm-models-for-time- series-forecasting. Acesso em: 28 ago. 2020.

BROWNLEE, Jason. **Stacked Long Short-Term Memory Networks**. Disponível em: https://machinelearningmastery.com/stacked-long-short-term-memory-networks/ Acesso em 14 ago. 2019.

BUYYA, Rajkumar and Srirama, Satish Narayana and Casale, Giuliano and Calheiros, Rodrigo and Simmhan *et al.***, A Manifesto for Future Generation Cloud Computing: Research Directions for the Next Decade**, In: Association for Computing Machinery, v. 51, 2018.

BUYYA, Rajkumar; YEO, Chee Shin; VENUGOPAL, Srikumar; BROBERG, James; BRANDIC, Ivona. **Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility**. Fut. Gen. Comput. Syst. v. 25, p. 599–616, 2009.

CALHEIROS, R. N.; MASOUMI, Enayat; RANJAN, Rajiv; BUYYA, Rajkumar. **Workload prediction using ARIMA model and its impact**. IEEE Transactions on Cloud Computing, v. 3, n. 4, out./dez. 2015.

CALHEIROS, Rodrigo N.; Rajiv Ranjan; Rajkumar Buyya Calheiros et al. **Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments**. International Conference on Parallel Processing, p. 295-304, 2011.

CAO, Jian; FU, Jiwen; LI, Minglu; CHEN, Jinjun. **CPU load prediction for cloud environment based on a dynamic ensemble model.** Special Issue: Software Tools and Techniques for Monitoring and Prediction of Cloud Services. v. 44 n.7, p.793–804, 2014.

CAO, Lijuan. **Support vector machine experts for time series forecasting**. Neurocomputing v. 51, p. 321 – 339, 2003.

CHEN, Shi.; PAUL, Rajib.; JANIES, Daniel A.; MURPHY, Keith.; FENG, Tinghao; THILL, Jean-Claude. **Exploring Feasibility of Multivariate Deep Learning Models in Predicting COVID-19 Epidemic**. In: Frontiers in Public Health, v. 9, 2021.

CHO, Kyunghyun; VAN MERRIENBOER, Bart; BAHDANAU, DZmitry; BENGIO, Yoshua. **On the properties of neural machine translation: Encoder-decoder approaches**. arXiv preprint arXiv:1409.1259, 2014.

CHUNG, J.; GULCEHRE, C.; CHO, K.; BENGIO, Y. **Empirical evaluation of gated recurrent neural networks on sequence modeling,** NIPS 2014 - Workshop on Deep Learning, dez. 2014.

COUTINHO, Emanuel F., GOMES, Danielo G., SOUZA, José Neuman. **Uma Proposta de Arquitetura Autonômica para Elasticidade em Computação em Nuvem.** Anais do 4º Workshop de Sistemas Distribuídos Autonômicos - WoSiDA 2014.

COUTINHO, Emanuel; SOUSA, Flávio; GOMES, Danielo; SOUZA, Jose. **Elasticidade em Computação na Nuvem: Uma Abordagem Sistemática.** Livro de Minicursos do XXXI Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2013), Chapter: Elasticidade em Computação na Nuvem: Uma Abordagem Sistemática, Publisher: Sociedade Brasileira de Computação (SBC), Editores: Joni da Silva Fraga; Jacir Luiz Bordim; Rafael Timóteo de Sousa Júnior; William Ferreira Giozza, p. 215-258, 2013.

DANG-QUANG, Nhat-Minh; YOO, Myungsik. **Multivariate Deep Learning Model for Workload Prediction in Cloud Computing**, International Conference on Information and Communication Technology Convergence (ICTC), p. 858-862, 2021.

DARAJE, Megersa; SHAIKH, Javed. **Hybrid Resource Scaling for Dynamic Workload in Cloud Computing.** IEEE International Conference on Mobile Networks and Wireless Communications (ICMNWC), 2021.

DENG, Da; LU, Zhihui; FANG, Wei; WU, Jie. **CloudStreamMedia: A Cloud Assistant Global Video on Demand Leasing Scheme**. IEEE International Conference on Services Computing (SCC), 2013.

DILLON, Tharam; WU, Chen; CHANG, Elizabeth. **Cloud Computing: Issues and Challenges,** 24th IEEE International Conference on Advanced Information Networking and Applications, p. 27-33, 2010.

ELPRINCE, Noha. **Autonomous resource provision in virtual data centers**. 2013 IFIP / IEEE International Symposium on Integrated Network Management (IM 2013). p. 27-31, maio 2013.

FARGO, Farah. **Autonomic Workload and Resource Management of Cloud Computing Services**, IEEE International Conference on Cloud and Autonomic Computing, 2014. DOI 10.1109/ICCAC.2014.36.

FARRAG, Tamer Ahmed; ELATTAR, EHAB E. **Optimized Deep Stacked Long Short-Term Memory Network for Long-Term Load Forecasting**, IEEE Access 9:68511-68522. jan. 2021

FEHLING, Christoph; LEYMANN, Frank; RETTER, Ralph; SCHUPECK, Walter; ARBITTER, Peter. **Cloud Computing Patterns: fundamentals to design, build, and manage cloud applications.** USA: Springer, p. 367, 2014.

FEI, Bowen; ZHU, Xiaomin; LIU, Daqian; CHEN, Junjie; BAO, Weidong; LIU, Ling. **Elastic Resource Provisioning using Data Clustering in Cloud Service Platform**, IEEE Transactions on Services Computing., v. 15, n. 3, maio/jun. 2022.

FENNER, Germano; SAMPAIO, Albert, Lima; De SOUZA J. N.; MOURA, J. A. B.; BEZERRA, T. R. **Business-Driven Support for Infrastructure as a Service Capacity Management Through System Dynamics Simulations**, IEEE Transactions on Network and Service Management, v. 18, n. 2, p. 2063-2076, jun. 2021, DOI: 10.1109/TNSM.2020.3044892.

GALANTE, Guilherme; BONA, Luis Carlos E. de. **A survey on cloud computing elasticity.** Utility and Cloud Computing (UCC), IEEE Fifth International Conference, p. 263–270, 2012.

GANDHI, A., DUBE, P., KARVE, A., KOCHUT, A., ZHANG, L.: **Model driven optimal resource scaling in cloud**. Software System Model. v. 17, n. 2, p. 509–526, 2017.

GAO, Jiechao; WANG, Haoyu; SHEN, Haiying**. Machine Learning Based Workload Prediction in Cloud Computing,** 29th International Conference on Computer Communications and Networks (ICCCN), p. 1-9, 2020.

GERS, Felix A.; SCHMIDHUBER, Jurgen; CUMMINS, Fred. **Learning to Forget: Continual Prediction with LSTM**. Neural Computation, MIT Press. v. 12 n. 10, p. 2451– 2471, 2000.

GREFF, Klaus; SRIVASTAVA, Rupesh Kumar; KOUTNÍK, Jan; STEUNEBRINK, Bas R.; SCHMIDHUBER, Jürgen**. LSTM: A search space odyssey.** Disponível em: URL http://arxiv.org/ abs/1503.04069. Acesso em: 10 jan. 2022.

HAGAN, Martin T.; BEHR, Suzane M. **The time series approach to short-term load forecasting**. IEEE Trans. Power Syst. v. 2, n. 3, p. 785–791, 1987.

HAMZEH, Hamed; MEACHAM, Sofia; VIRGINAS, Botond; PHALP, Keith. **Taxonomy of Autonomic Cloud Computing.** International Journal of Computer and Communication Engineering. v. 7, n. 3, jul. 2018.

HAN, Yi; Jeffrey Chan Christopher Leckie. **Analysing Virtual Machine Usage in Cloud Computing**. IEEE Ninth World Congress on Services, 2013.

HEIN, Daniel. **5 things to look for in a Cloud Service Level Agreement**. Solutions Review. Best Practices. Disponível em: https://solutionsreview.com/cloud-platforms/5-things-to-look- for-in-a-cloud-service-level-agreement/ Acesso em: 10 jan. 2022.

HERMANS, Michiel; SCHRAUWEN, Benjamin. **Training and Analyzing Deep**

Recurrent Neural Networks. Advances in Neural Information Processing Systems (NIPS 2013). v. 26, 2013.

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. **Long Short-Term Memory.** Neural Computation, v. 9, n. 8, p. 1735–1780, 1997.

HYNDMAN, Rob J.; ATHANASOPOULOS, George. **Forecasting: Principles and Practice**. Australia: Monash University, 2017. Disponível em: http://otexts.org/fpp2/. Acesso em: 10 jan. 2022.

IBM. **An architectural blueprint for autonomic computing**. White Paper, 3a ed. Jun. 2005.

IBRAHIM, Yazid Ado; ADAMU, Alhassan; ABDULRAHMAN, Salisu Mamman, MUHAMMAD, Akilu Rilwan. **Autonomic Cloud Computing: A Review**. International Journal of Computer (IJC), v. 26, n. 1, p. 99-104, 2017.

IQBAL, Waheed; DAILEY, Matthew; CARRERA, David. **SLA-Driven Adaptive Resource Management for Web Applications on a Heterogeneous Compute Cloud**. CloudCom 2009, LNCS 5931, Springer-Verlag Berlin Heidelberg v. 20, p. 243–253, 2009.

IQBAL, Waheed; ERRADI, Abdelkarim; ABDULLAH, Muhammad, Arif Mahmood. **Predictive Auto-scaling of Multi-tier Applications Using Performance Varying Cloud Resources**. IEEE Transactions on Cloud Computing, v. 10, n.1, Jan-Mar, 2022.

ISO. ISO 5725-1 Accuracy (Trueness and Precision) of Measurement Methods and Results — Part 1: General Principles and Definitions, 1994.

JANARDHANAN, Deepak; BARRETT, Enda. **CPU Workload Forecasting of Machines in Data Centers using LSTM Recurrent Neural Networks and ARIMA Models.** The 12th International Conference for Internet Technology and Secured Transactions (ICITST-2017).

JIANG, J.; LU, J.; ZHANG, G.; LONG, G. **Optimal cloud resource auto-scaling for web applications**. 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, p. 58–65, 2013. DOI: 10.1109/CCGrid.2013.73.

JIRSIK, Tomas; TRCKA, Stepan; CELEDA, Pavel. **Quality of Service Forecasting with LSTM Neural Network.** IFIP/IEEE International Symposium on Integrated Network Management, 2019.

JOSE, Anu; V, Vidya. **A Stacked Long Short-Term Memory Neural Networks for Parking Occupancy Rate Prediction**. 10th IEEE International Conference on Communication Systems and Network Technologies, 2021.

JOZEFOWICZ, Rafal; ZAREMBA, Wojciech; SUTSKEVER, Ilya. **An Empirical Exploration of Recurrent Network Architectures**. Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:2342-2350, 2015.

KAVYASRI M. N., Dr. B. Ramesh. **Comparative Study of Scheduling Algorithms to Enhance the Performance of Virtual Machines in Cloud Computing**. International Conference on Emerging Trends in Engineering, Technology and Science, p. 1-5, 2016.

DOI: 10.1109/ICETETS.2016.7602980.

KEPHART, J. O.; CHESS, D. M. **The vision of autonomic computing**. Computer, v. 36, n. 1, p. 41–50, 2003.

KIM, In Kee; WANG, Wei; QI, Yanjun; HUMPHREY, Marty. **Empirical Evaluation of Workload Forecasting Techniques for Predictive Cloud Resource Scaling**. IEEE 9th. International Conference on Cloud Computing. San Francisco, p. 1-10, 2016.

KIRANYAZ, Serkan; AVCI, Onur; ABDELJABER, Osama; INCE, Turker; GABBOUJ, Moncef; INMAN, Daniel J. **1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing**. v. 151, abr. 2021.

KIRCHOFF, Dionatrã F.; XAVIER, Miguel; MASTELLA, Juliana; DE ROSE, César A. F. **A preliminary study of machine learning workload prediction techniques for cloud applications.** 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, 2019.

KOUKI, Yousri; OLIVEIRA, Frederico Alvares De; DUPONT, Simon; LEDOUX, Thomas. **A Language Support for Cloud Elasticity Management**. 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, p. 206-215, 2014.

KOUKI, Yousri; HASAN, Md Sabbir; LEDOUX, Thomas. **Delta Scaling: How Resources Scalability/Termination Can Be Taken Place Economically?** IEEE World Congress on Services, p. 55-62, 2015.

KRAUSE, Ben. **Flexible Neural Architectures for Sequence Modeling**. PhD Thesis. University of Edinburgh, 2019.

KRAUSE, Ben; LU, Liang; MURRAY, Iain; RENALS, Steve. **Multiplicative LSTM for Sequence Modelling**, arXiv:1609.07959v3, Neural and Evolutionary Computing, 2017.

KUMAR, ER. Manoj. **Cloud Computing in Resource Management**, International Journal of Engineering and Management Research (IJEMR). v. 8, n. 6, p. 93-98, 2018.

KUMAR, Jitendra, GOOMER, Rimsha, SINGH, Ashutosh Kumar. **Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters.** 6th International Conference on Smart Computing and Communications, ICSCC 2017, dez. 2017.

KUMAR, Jitendra. SINGH, Ashutosh Kumar. **Dynamic Resource Scaling in Cloud Using Neural Network and Black Hole Algorithm,** Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS). 2016.

KUMAR, Jitendra; SINGH, Ashutosh Kumar. **Cloud data center workload estimation using error preventive time series forecasting models**. Cluster Computing, v. 23 p. 1363–1379, 2020.

KUMAR, Jitendra; SINGH, Ashutosh Kumar. **Workload prediction in cloud using artificial neural network and adaptive differential evolution**, Future Generation Computer Systems**.**, v. 81, p. 41-52, 2018. Disponível em: https://www.sciencedirect.com/science/article/pii/S0167739X173004 44. Acesso em: 30 set.

2022.

KUMAR, Jitendra; SINGH, Ashutosh Kumar; MOHAN, Anand; BUYYA, Rajkumar. **Machine learning for cloud management.** First edition. Boca Raton: CRC Press, 2022.

KUMAR, Krishan; RAO, K. Gangadhara; BULLA, Suneetha; VENKATESWARULU**,** D. **Forecasting of Cloud Computing Services Workload using Machine Learning**, Turkish Journal of Computer and Mathematics Education. v. 12, n. 11, 2021.

LALANDA, Philippe; MCCANN, Julie A.; DIACONESCU, Ada. **Autonomic Computing Principles, Design, and Implementation.** London: Springer-Verlag, 2013.

LEE, Chun-Hsiang; HE, Zhengda; LI, Zhaofeng; LU, Xu; WANG, Jian; WU, Chao. **A Comparison of Machine Learning Algorithms for Automatic Cloud Resource Scaling on a Multi-Tenant Platform.** Journal of Physics: Conference Series 1828 (2021) 012039 IOP Publishing.

LEKA, Habte Lejebo; FENGLI, Zhang; KENEA, Ayantu Tesfaye, TEGENE, Abebe Tamrat, ATANDOH, Peter; HUNDERA, Negalign Wake. **A hybrid CNN-LSTM model for virtual machine workload forecasting in the cloud data center.** 18th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), p. 474–478. dez. 2021

LIMA, Alberto Sampaio; DE SOUZA, J. N. **O estado da arte da pesquisa em BDIM.** (Business-driven IT Management). UFC: Trabalho Técnico, 2011.

LIU, F.; TONG, J.; MAO, J.; BOHN, R.; MESSINA, J.; BADGER, M.; LEAF, D. NIST. **Cloud Computing Reference Architecture**, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, (2011). Disponível em: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id.909505. Acesso em: 12 Maio 2022.

LIU, Shihui. **A Computer Dynamic Index Forecast Model Based on Indicator of Long Short-Term Memory and Cloud Computing,** IEEE International Conference on Data Science and Computer Application (ICDSCA), p. 495-498, 2021.

LORIDO-BOTRAN, Tania; MIGUEL-ALONSO, Jose; LOZANO, José A. **A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments**. J Grid Computing, v. 12, p. 559–592. 2014.

LORIDO-BOTRAN, Tania; MIGUEL-ALONSO, Jose; LOZANO, José A**. Autoscaling Techniques for Elastic Applications in Cloud Environments**. Technical Report EHU-KAT- IK-09-12, 2012.

LORIDO-BOTRAN, Tania; MIGUEL-ALONSO, Jose; LOZANO, José A**. Comparison of Auto-scaling Techniques for Cloud Environments**. Actas de las XXIV Jornadas de Paralelismo, Servicio de Publicaciones, 2013.

MANSOURI, N.; GHAFARI, R.; ZADE, B. M. H. **Cloud Computing Simulators: A Comprehensive Review**. Simulation Modelling Practice and Theory. v. 104, nov. 2020,

MAO, Hongzi; ALIZADEH, Mohammad; MENACHE, Ishai; KANDULA, Srikanth.

Resource Management with Deep Reinforcement Learning. HotNets-XV, nov. 2016.

MARTIN, P.; BROWN, A.; POWLEY, W.; VAZQUEZ-POLETTI, J. L. **Autonomic management of elastic services in the cloud**. IEEE Computer Society ISCC, p. 135–140, 2011

MASDARI, M., KHOSHNEVIS, A. **A survey and classification of the workload forecasting methods in cloud computing**. Cluster Comput *v.* 23, p. 2399–2424, 2020.

MELL, P. and GRANCE, T. **The NIST Definition of Cloud Computing**, Special Publication, National Institute of Standards and Technology, 2011.

MONIZ, J. R. Antony; KRUEGER, David**. Nested LSTMs**. Proceedings of Machine Learning Research v. 77, p. 530–544, 2017.

MOORE, Laura R., BEAN, Kathryn; ELLAHI, Tariq. **A Coordinated Reactive and Predictive Approach to Cloud Elasticity**. The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, 2013. ISBN: 978-1-61208-271-4 92.

MORAIS, Fábio; LOPES, Raquel; BRASILEIRO, Francisco. **Provisionamento Automático de Recursos em Nuvem IaaS: eficiência e limitações de abordagens reativas.** Campina Grande: Universidade Federal de Campina Grande (UFCG) Laboratório de Sistemas Distribuídos.

MORENO-VOZMEDIANO, Rafael; MONTERO Rubén S.; HUEDO, Eduardo; LORENTE, Ignacio M. **Efficient resource provisioning for elastic Cloud services based on machine learning techniques**. Journal of Cloud Computing: Advances, Systems and Applications, v. 8. n. 5, p.1-8, dez. 2019.

MUNÕZ-ESCOÍ; Francesc D., BERNABEU-AUBAN, Jose M. **A Survey on Elasticity Management in the PaaS Service Model**. Technical Report ITI-SIDI-2015/002.

OLIVEIRA, J. A. **Um Modelo Formal para Avaliar o Valor de Negócio e sua Aplicação no Contexto de Gestão e Governança de TI**. PhD Thesis. Universidade Federal de Campina Grande, PB, 2010.

OMG Cloud Working Group. **Practical Guide to Cloud Governance**. Version 1.0, Document mars/2019-06-xx.

OMG Cloud Working Group. **Practical Guide to Cloud Service Agreements**. Version 3.0, Document mars/2019-02-01.

PAPER WITH CODE. **LSTM Explained-Papers with code**. Disponível em: https://paperswithcode.com/method/lstm. Acesso em: 30 set. 2022.

PARASHAR, Manish; HARIRI, Salim. **Autonomic computing: concepts, infrastructure, and applications**. 1st Edition, **CRC Press,** 2007**.**

PATEL, Pankesh, Ajith H. Ranabahu, and Amit P. Sheth, **Service level agreement in cloud computing,** 2009, Disponível em: https://corescholar.libraries.wright.edu/knoesis/78/ Acesso em: 30 set. 2022.

QU, C.; Calheiros, R.N., BUYYA, R. **Auto-scaling web applications in clouds: a taxonomy and survey**. ACM Computing Surveys, v. 51, n. 4, p. 73, jul. 2018. https://doi.org/10.1145/3148149.

RIBAS, Maristela. **Um Modelo de Decisão para Adoção de Serviços em Nuvem Usando Redes de Petri**. Tese de Doutorado, Universidade Federal do Ceará, CE, 2015.

ROY, Nilabja; DUBEY Abhishek; GOKHALE, Aniruddha. **Efficient Autoscaling in the Cloud using Predictive Models for Workload Forecasting.** IEEE 4th International Conference on Cloud Computing, 2011.

RUNERSON, P.; HOST, M. **Guidelines for conducting and reporting case study research in software engineering**, Springer: Empiric Software Engineering, v. 14, p. 31-164, DOI 10.1007, 2009.

SAHU, Pradip Kumar; PAL, Santi Ranjan; DAS, Ajit Kumar. **Estimation and Inferential Statistics**. India: Springer, 2015. 317p.

SAUVE, J.; MOURA, A.; SAMPAIO, M.; JORNADA, J; RADZIUK, E. **An Introductory Overview and Survey of Business-Driven IT Management**, IEEE/IFIP Business Driven IT Management, p. 1-10, 2006.

SAVEINCLOUD. **Pague somente pelo que você usa**. Disponível em: https://saveincloud.com/pt/blog/virtuozzo-cloud/a-ilusoria-eficiencia-da-nuvem-voce-realmente-paga-o-quanto-usa. Acesso em: 27 jul. 2022.

SAXENA, Deepika. **Workload forecasting and resource management models based on machine learning for cloud computing environments**, 2021.

SCHMITT, Michael. **On the Complexity of Computing and Learning with Multiplicative Neural Networks**. Neural Computation. v. 14, n. 2, p. 241–301, fev. 2002.

SCHULZE, Bruno; SOUZA José Neuman; MURY, Antonio Roberto, BORGES, Hélder Pereira. **Computação em nuvem**. Disponível em: https://livroaberto.ibict.br/bitstream/1/861/1/computa%c3%87%c3%83o%20em%20nuvem.pdf Acesso em: 27 jul. 2022.

SHAHIN, Ashraf A. **Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network**. International Journal of Advanced Computer Science and Applications, v. 7, n. 12, 2016.

SHARIFFDEEN, R. S. **Adaptive Workload Prediction for Proactive Auto Scaling in PaaS Systems.** 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), p. 22-29, 2016. DOI: 10.1109/CloudTech.2016.7847713.

SHEN, Zhiming, SUBBIAH, Sethuraman Subbiah; GU, Xiaohui; WILKES, John. **Cloudscale: elastic resource scaling for multi-tenant cloud systems**. ACM Symposium on Cloud Computing. p. 5, 2011.

SHI, Y., Jiang, X., Ye, K., 2011. **An energy-efficient scheme for Cloud resource provisioning based on Cloudsim**, IEEE International Conference on Cluster Computing. p. 595–599, 2011.

SHIVANI; SINGH, Ajmer. **Taxonomy of SLA violation minimization techniques in cloud computing.** 2a. International Conference on Inventive Communication and Computational Technologies (ICICCT 2018). ISBN: 978-1-5386-1974-2.

SHUVO, Md. Nahid Hasan; MASWOOD, Mirza Mohd Shahriar; ALHARBI, Abdullah G. **LSRU: A Novel Deep Learning based Hybrid Method to Predict the Workload of Virtual Machines in Cloud Data Center.** 2020 IEEE Region 10 Symposium (TENSYMP), jun. 2020.

SIAMI-NAMINI, Sima; TAVAKOLI, Neda; SIAMI NAMIN, Akbar. **A Comparison of ARIMA and LSTM in Forecasting Time Series**, 17th IEEE International Conference on Machine Learning and Applications (ICMLA), p. 1394-1401, 2018.

SINGH, S.; CHANA, I. **Cloud resource provisioning: survey, status, and future research directions**. Knowledge and Information Systems, v. 49, n. 3, p. 1005-1069, 2016.

SINGH, Sukhpal; CHANA, Inderveer **A survey on resource scheduling in cloud Computing: issues and challenges**. Journal of Grid Computing, v. 14, n. 2, p. 217–264, 2016.

SINGH, Sukhpal; CHANA, Inderveer. **QoS-aware autonomic resource management in cloud computing: a systematic review**. ACM Computing Surveys, v. 48, n. 3, p. 42, dez. 2015. DOI: http://dx.doi.org/10.1145/2843889.

SINGH, Sukhpal; CHANA, Inderveer. **Resource provisioning and scheduling in clouds: QoS perspective.** The Journal of Supercomputing., v. 72, n. 3, p. 926– 960, mar. 2016.

SMAGULOVA, Kamilya; JAMES, Alex Pappachen. **Overview of Long Short-Term Memory Neural Networks.** In: James, A. P. (eds) Deep Learning Classifiers with Memristive Networks, Modeling and Optimization in Science and Technologies, v. 14. 2020. DOI: https://doi.org/10.1007/978-3-030-14524-8_11.

SONG, B., Yu, Y., Zhou, Y., Wang, Z., Du, S.: **Host load prediction with long short-term memory in cloud computing**. J. Supercomputers. v. 74, n. 12, p. 6554–6568, 2018.

SONKAR, S.K.; KHARAT, M.U. **A Survey on Resource Management in Cloud Computing Environment**. International Journal of Advanced Trends in Computer Science and Engineering. v. 4, n. 4, jul./ago. 2015. Disponível em: http://www.warse.org/IJATCSE/static/pdf/file/ijatcse01442015.pdf. Acesso em: 23 jun. 2021.

SUTSKEVER, Ilya; VINYALS, Oriol; LE, Quoc V**. Sequence to Sequence Learning with Neural Networks.** Advances in Neural Information Processing Systems (NIPS 2014). v. 27, 2014.

TAMMARO, Davide; Doumith Elias A.; Sawsan Al Zahr; Jean-Paul Smets; Maurice Gagnaire. **Dynamic resource allocation in cloud environment under time-variant job requests**, IEEE Third International Conference on Cloud Computing Technology and Science, 2011.

TAYLAN, Kamil. **T-Test - O que é um T-Test?** Disponível em: kamiltaylan.blog -

enciclopédia financeira https://pt.kamiltaylan.blog/t-test/ Acesso em: 23 jun. 2021.

TECHTARGET. **What is Cloud Management? Everything You Need to Know**. E-GUIDE. Disponível em: http://www.searchcloudcomputing.com. Acesso em: 10 out. 2021.

ULLAH, Arif; ABBASI, Irshad Ahmed; REHMAN, Muhammad Zubair; ALAM, Tanweer; AZNAOUI, Hanane. **Adapted Convolutional Neural Networks and Long Short-Term Memory for Host Utilization Prediction in Cloud Data Center**. Research Square, jul.2021. DOI: https://doi.org/10.21203/rs.3.rs-597475/v1.

VASHISTHA, Avneesh; VERMA, Pushpneel. **A Literature Review and Taxonomy on Workload Prediction in Cloud Data Center**, 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), p. 415-420, 2020.

VOUK, M. **Cloud Computing - Issues, Research and Implementations**. Journal of Computing and Information Technology, v.16, n. 4, p. 235–246. 2008. DOI:10.2498/cit.1001391

VU, Kevin**. 5 types of lstm recurrent neural networks and what to do with them.** Nov. 12, 2019. Disponível em: https://www.exxactcorp.com/blog/Deep-Learning/5-types-of-lstm-recurrent-neural-networks-and-what-to-do-with-them. Acesso em: 12 nov. 2019.

WANG, C.-F., Hung, W.-Y., Yang, C.-S. **A prediction-based energy conserving resources allocation scheme for Cloud computing**, IEEE International Conference on Granular Computing (GrC), p. 320–324, 2014.

WANG, L. et al. Cloud Computing: A Perspective Study. New Generation Computing, Ohmsha, Ltd., v. 28, p. 137–146, 2010. ISSN 0288-3635. 10.1007/s00354-008-0081-5. Disponível em: http://dx.doi.org/10.1007/s00354-008-0081-5. Acesso em: 04 out. 2017.

WESNER, Janet - MAE e RMSE — **Qual métrica é melhor.** Disponível em: https://medium.com/human-in-a-machine-world/mae-and-rmse-which-metric-is-better- 60ac3bde13d). Acesso em: 23 mar. 2016.

WU, Yuhuai; ZHANG, Saizheng; ZHANG, Ying; BENGIO, Yoshua; SALAKHUTDINOV, Russ R**. On Multiplicative Integration with Recurrent Neural Networks.** NIPS'16: Proceedings of the 30th International Conference on Neural Information Processing Systems, p. 2864–2872. dez. 2016.

XIAO, Frank. **Time Series Forecasting with Stacked Long Short-Term Memory Networks**. arXiv:2011.00697v1. nov. 2020.

XIONG, P., Chi, Y., Zhu, S., Moon, H.J., Pu, C., Hacigumus, H., 2011**. Intelligent management of virtualized resources for database systems in Cloud environment**, IEEE 27th International Conference on Data Engineering, p. 87–98.

YADAV, Archana; KUSHWAHA, Shivam; GUPTA, Jyoti; SAXENA, Deepika; SINGH, Ashutosh Kumar. **A survey of the workload forecasting methods in cloud computing.** Proceedings of 3rd International Conference on Machine Learning, Advances in Computing, Renewable Energy and Communication. p. 539–547, 2022.

YADAV, Mahendra Pratap; PAL, Nisha; YADAV, Dharmendar Kumar. **Workload Prediction over Cloud Server using Time Series Data**, 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), p. 267-272, 2021.

YAN, Shi. **Understanding LSTM and its Diagrams**. Disponível em: https://blog.mlreview.com/understanding-lstm-and-its-diagrams-37e2f46f1714. Acesso em: 13 mar. 2016.

YANG, Jingq; LIU, Chuanchang; SHANG, Yanlei; MAO, Zexiang; CHEN, Junliang. **Workload predicting-based automatic scaling in service clouds**. IEEE Sixth International Conference on IEEE. p. 810–815, 2013.

ZHANG, Li; ZHANG, Yichuan; JAMSHIDI, Pooyan; XU, Lei; PAHL, Claus. **Workload Patterns for Quality-Driven Dynamic Cloud Service Configuration and Auto-Scaling**, IEEE/ACM 7th International Conference on Utility and Cloud Computing, p. 156-165, 2014.

ZHONG, C., Yuan, X.: **Intelligent elastic scheduling algorithms for PaaS Cloud platform based on load prediction.** IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), p. 1500–1503, 2019.

ZHONG, Jiang; DUAN, Saisai; LI, Qing. **Auto-Scaling Cloud Resources using LSTM and Reinforcement Learning to Guarantee Service-Level Agreements and Reduce Resource Costs.** Journal of Physics: Conf. Series 1237, IOP Publishing, 2019. doi:10.1088/1742-6596/1237/2/022033.

ZHU, Yonghua; ZHANG, Weilin; CHEN, Yihai, GAO, Honghao. **A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud environment**. EURASIP Journal on Wireless Communications and Networking, 2019. https://doi.org/10.1186/s13638-019-1605-z.

APÊNDICE A – SIMULAÇÃO DO MODELO PREDITIVO

```
# Stacked LSTM para previsão de recursos em nuvem
# Usando LSTM para prever séries temporais
# Nesse experimento iremos demonstrar como uma LSTM pode ser usada para previsão em séries temporais.
#  Utilizaremos uma rede neural Stacked LSTM bem simples para tratar um caso de uma série temporal
# de recursos utilizados em vários anos

# Importando bibliotecas

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from pandas import read_csv
import tensorflow as tf
import math
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import LSTM
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import classification_report
from sklearn.metrics import mean_squared_error as mse
from sklearn.metrics import mean_absolute_error as mae
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_absolute_percentage_error as mape

# convert an array of values into a dataset matrix
def create_dataset(dataset, look_back=1):
                dataX, dataY = [], []
                for i in range(len(dataset)-look_back-1):
                                a = dataset[i:(i+look_back), 0]
                                dataX.append(a)
                                dataY.append(dataset[i + look_back, 0])
                return np.array(dataX), np.array(dataY)

# fix random seed for reproducibility
tf.random.set_seed(7)

# load the dataset
dataframe = read_csv(r'c:\\recursos-nuvem.csv', engine='python')

dataset = dataframe.values
dataset = dataset.astype('float32')

# normalize the dataset
scaler = MinMaxScaler(feature_range=(0, 1))
dataset = scaler.fit_transform(dataset)

# split into train and test sets
train_size = int(len(dataset) * 0.67)
test_size = len(dataset) - train_size
train, test = dataset[0:train_size,:], dataset[train_size:len(dataset),:]

# reshape into X=t and Y=t+1
look_back = 3
trainX, trainY = create_dataset(train, look_back)
testX, testY = create_dataset(test, look_back)

# reshape input to be [samples, time steps, features]
```

```
trainX = np.reshape(trainX, (trainX.shape[0], trainX.shape[1], 1))
testX = np.reshape(testX, (testX.shape[0], testX.shape[1], 1))

# create and fit the LSTM network
batch_size = 1
model = Sequential()
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True, return_sequences=True))
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True, return_sequences=True))
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True, return_sequences=True))
model.add(LSTM(4, batch_input_shape=(batch_size, look_back, 1), stateful=True))
model.add(Dense(1)),
model.compile(loss='mean_squared_error', optimizer='adam', metrics=['accuracy'])

# Define as épocas

for i in range(100):
   model.fit(trainX, trainY, epochs=50, batch_size=batch_size, verbose=2, shuffle=False)
   model.reset_states()

# make predictions
trainPredict = model.predict(trainX, batch_size=batch_size)
model.reset_states()
testPredict = model.predict(testX, batch_size=batch_size)

# invert predictions
trainPredict = scaler.inverse_transform(trainPredict)
trainY = scaler.inverse_transform([trainY])
testPredict = scaler.inverse_transform(testPredict)
testY = scaler.inverse_transform([testY])

# Calcula os erros de previsão
print ('Apresentando as Métricas')

# mean_absolute_percentage_error MAPE
print ('************************************')
print ('Mean Absolute Percentage error (MAPE)')
trainScore = mape(trainY[0], trainPredict[:,0])
print('Train Score: %.2f MAPE' % (trainScore))
testScore = mape(testY[0], testPredict[:,0])
print('Test Score: %.2f MAPE' % (testScore))

# Raiz Quadrada do Erro Médio – RMSE - (Root Mean Squared Error)
print ('************************************')
print ('Root Mean Squared Error (RMSE)')
trainScore = np.sqrt(mse(trainY[0], trainPredict[:,0]))
print('Train Score: %.2f RMSE' % (trainScore))
testScore = np.sqrt(mse(testY[0], testPredict[:,0]))
print('Test Score: %.2f RMSE' % (testScore))

# Erro Quadrático Médio – MSE – (Mean Squared Error
print ('************************************')
print ('Mean Squared Error (MSE)')
trainScore = mse(trainY[0], trainPredict[:,0])
print('Train Score: %.2f MSE' % (trainScore))
testScore = mse(testY[0], testPredict[:,0])
print('Test Score: %.2f MSE' % (testScore))

# Erro Médio Absoluto - MAE (mean absolut error)
print ('************************************')
print ('Mean Absolut Error (MAE)')
```

```
trainScore = mae(trainY[0], trainPredict[:,0])
print('Train Score: %.2f MAE' % (trainScore))
testScore = mae(testY[0], testPredict[:,0])
print('Test Score: %.2f MAE' % (testScore))

# Acuracy Score
# print ('************************************')
# _, accuracy = model.evaluate(trainX, trainY)
# print('Acurácia do Modelo: %.2f' % (accuracy*100))
# print ('************************************')
# Acurácia
# print(accuracy_score(testScore, trainScore))
# print(classification_report(trainScore, testScore))

# shift train predictions for plotting
trainPredictPlot = np.empty_like(dataset)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(trainPredict)+look_back, :] = trainPredict
# shift test predictions for plotting
testPredictPlot = np.empty_like(dataset)
testPredictPlot[:, :] = np.nan
testPredictPlot[len(trainPredict)+(look_back*2)+1:len(dataset)-1, :] = testPredict

# plot baseline and predictions
plt.figure(figsize=(20, 10))
plt.title('PREVISÃO DE RECURSOS - MODELO LSTM STACKED', fontsize = 30)
plt.plot((trainPredictPlot), label='Dados do Treinamento')
plt.plot((testPredictPlot), label='Dados de Teste')
plt.xlabel('Data', fontsize = 20)
plt.ylabel('Recursos', fontsize = 20)
plt.legend()
plt.show()

# plot baseline and predictions
plt.figure(figsize=(20, 10))
plt.title('PREVISÃO DE RECURSOS - MODELO LSTM STACKED', fontsize = 30)
plt.plot(scaler.inverse_transform(dataset), color='Red', label='Dados de Previsão')
plt.xlabel('Data', fontsize = 20)
plt.ylabel('Recursos', fontsize = 20)
plt.legend()
plt.show()
```

Algoritmo 1: Provisionamento Reativo

==========

Etapa 1. Atribua solicitação do usuário às instâncias da VM
Etapa 2. Monitore continuamente instâncias individuais usando métricas como CPU e memória
Etapa 3. Se as métricas monitoradas implicarem capacidade insuficiente de infraestrutura para processar a solicitação atribuída conforme a política de governança
Etapa 3a. Reter ou encerrar a solicitação do usuário
Etapa 3b. Provisionar instâncias de VM adicionais
Etapa 3c. Reenvie a solicitação para a nova instância da VM se a solicitação estiver em espera
Etapa 4. Outra VM atribuída continuaria processando as solicitações do usuário
Etapa 5. Se os recursos forem considerados excedentes, conforme definido pela política de governança
Etapa 5a. Desprovisionar as instâncias em excesso
Etapa 7. Senão, retenha todas as instâncias ativas
Etapa 8: Repita as etapas 1 a 8 para todas as solicitações do usuário

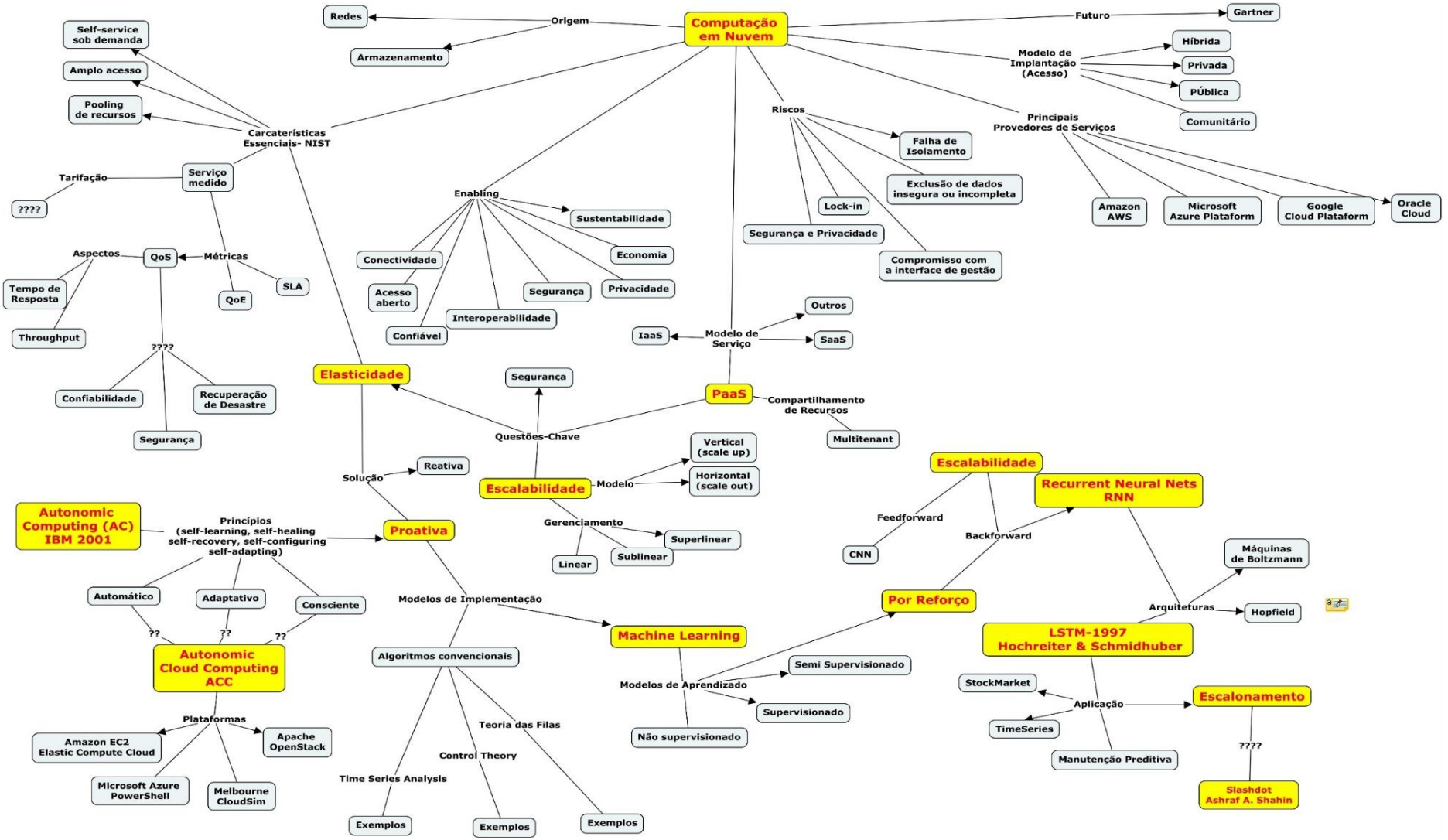
==========

Algoritmo 2: Provisionamento e desprovisionamento na abordagem proposta

==========

Etapa 1. Aplicação / carga de trabalho de execução seca usando a abordagem orientada a políticas e colete todas as métricas disponíveis, juntamente com a análise offline do estado da instância (provisionado / não provisionado) correspondente
Etapa 2. Selecione as métricas de avaliação relevantes para os negócios a partir das métricas coletadas
Etapa 3. Identifique a métrica mais impactante para instâncias de provisionamento / desprovisionamento usando a técnica de seleção estatística de recursos
Etapa 4. Crie o mapeamento entre a carga de trabalho e o estado da instância com base na métrica significativa identificada em tempo real
Etapa 5. Preveja a métrica significativa identificada usando um modelo apropriado
Etapa 6. Monitore a métrica significativa em tempo real e compare-a continuamente com a métrica prevista
Etapa 7. Se a métrica em tempo real não estiver dentro do nível de confiança de ± 95% da métrica prevista
Etapa 7a. Escolha um modelo apropriado usando o valor em tempo real
Etapa 8. Senão continue
Etapa 9. Se a capacidade atual da infraestrutura for considerada insuficiente para processar a carga de trabalho prevista com base no mapeamento criado na etapa 4
Etapa 9a. Provisionar instâncias de VM adicionais
Etapa 10. Atribua solicitação do usuário às instâncias da VM
Etapa 11. Se os recursos forem considerados excedentes para processar a carga de trabalho prevista
Etapa 11a. Desprovisionar as instâncias em excesso da VM
Etapa 12. Caso contrário, mantenha todas as instâncias ativas
Etapa 13. Repita as etapas 5 a 12 para todas as solicitações do usuário

APÊNDICE B – MAPA CONCEITUAL



APÊNDICE C – EXPERIMENTOS DE PREDIÇÃO DA CARGA DE TRABALHO EM SERVIÇOS NA NUVEM

A-Esquemas Baseados em Regressão
B-Baseados em Classificadores
C-Baseados em Estocástica
D-Grey predicting-based schemes
K-Hybrid

A-Esquemas Baseados em Regressão	
	Antonescu *et al. (2016)* apresentaram dois **algoritmos de dimensionamento de VM** com reconhecimento de **SLA preditivo para *Distributed Enterprise Information Systems* (dEISs)** para encontrar melhores condições de dimensionamento usando aplicativos distribuídos derivados de *benchmarks* de carga constante, com restrições de SLA. Foi utilizado dimensionamento com reconhecimento de SLA preditivo autorregressivo para garantir o desempenho nos aplicativos de nuvem distribuídos. Forneceram uma avaliação de seu trabalho em relação a métricas, como RMSD, tempo de execução, número de VMs etc.
Modelo de regressão linear	**Yang *et al.* (2013)** apresentaram um modelo de **regressão linear** para estimar a carga e aplicou-o em um mecanismo de auto escalonamento para dimensionar recursos virtuais em dimensionamento e pré-escalonamento em tempo real. Consideraram o pré-escalonamento usando programação inteira e introduziram um método guloso para previsão precisa que incorre em menor custo e SLAV.
A1-Baseados em ARIMA	**Li *et al*** - apresentaram uma técnica de provisionamento de VM baseada em previsão de carga (ARIMA-DEC). Apresentou um preditor de carga baseado em ARIMA com compensação dinâmica de erros e o aplica no TBAMP (um algoritmo de provisionamento) baseado em tempo e com reconhecimento de custo. ARIMA-DEC busca reduzir a taxa de inadimplência do SLA e o algoritmo TBAMP pode economizar custos de aluguel. O algoritmo TBAMP considera o custo das VMs ajustadas e leva em consideração o custo das VMs liberadas.
	Kumar *et al* realizaram uma previsão da carga para reduzir o custo de energia. Comparou o desempenho previsto do ARIMA, SARIMA (ARMA sazonal integrado) e ARFIMA (ARMA integrado fracionado) com o método de análise de espectro singular usando CPU, RAM e rastreamento de rede coletados do Wikimedia Grid. Mostraram que aumentar o tamanho da entrada não fornece melhores resultados de previsão, mas o modelo ARFIMA sofre com o alto tempo de computação quando o tamanho da entrada aumenta.
	Calheiros *et al.* apresentaram uma abordagem proativa para o recurso de provisionamento dinâmico em relação à previsão realizada com o modelo ARIMA. Aplica um componente analisador de carga que fornece estimativas aos outros componentes para permitir que eles dimensionem os recursos. Devido às limitações do ARIMA, este modelo não é capaz de prever o consumo de recursos de pico.
	Messias *et al.* buscaram prever os pedidos que chegam no próximo período para evitar sobrecarga. Este problema torna-se complexo quando os dados históricos não estão disponíveis para serem avaliados. Foi proposta uma abordagem de previsão usando AG para agregar modelos de previsão baseados em séries temporais. Foi realizada a previsão de carga de trabalho usando os métodos ARMA e ARIMA. Também usado o Holt-Winters.
A2-Support vector regression	**Barati *et al.*** apresentaram o TSVR, como uma abordagem baseada em SVM ajustada que treina três fatores baseados em SVR usando os algoritmos GA e PSO. Utiliza uma sequência caótica para melhorar a precisão da previsão e evitar a convergência prematura, aumentando a exploração e a diversidade no espaço de busca. Reduz a carga computacional de gerar números aleatórios em comparação com GA. Para a previsão de cargas de memória e de CPU são aplicados métodos baseados em kernel.

B-Baseados em Classificadores	
B1- Baseado em SVM	**Tong *et al.*** propuseram um coeficiente de periodicidade de características e implementaram métodos de classificação existentes. Experimentos no conjunto de dados do mundo real invalidam a eficiência do novo recurso proposto, que está nas combinações mais eficazes de recursos, aumenta a taxa de sucesso e diminui o MSE. O SVM pode atingir quase o mesmo desempenho que os métodos Bayes e seu desempenho é superior.
	Os autores apresentaram o WWSVM, um modelo de previsão de carga usando SVM *wavelet* ponderada para estimar a carga dos PMs na nuvem DC. Foi utilizada a forma wavelet trans como uma função kernel no SVM para atribuir um peso às amostras de acordo com sua importância e aumentar a precisão da previsão. Eles aplicaram o algoritmo PSO para otimização de parâmetros e usaram o conjunto de dados do Google para verificar sua abordagem. Esse esquema consiste em fases de pré-processamento de dados e previsão de carga, nas quais a primeira fase realiza a normalização da carga de trabalho e a análise de autocorrelação.
	Nikravesh *et al.* buscaram melhorar a precisão da previsão do escalonamento automático usando a classificação SVM e ANN. Eles indicaram que a precisão de previsão de SVM e ANN depende de seu padrão de carga, mas o SVM fornece melhor precisão de previsão com padrões de carga periódicos e crescentes, enquanto ANN apresenta melhores resultados na previsão de padrões de carga imprevistos.
B2- Baseado em Random forest	**Cetinski *et al*.** (**2015**) apresentaram o "*Advanced model for efficient workload prediction in the cloud"* (AME-WPC), um modelo para previsão de carga de trabalho em DCs que melhora a precisão da previsão. Lidaram com a previsão de carga usando métodos de classificação e regressão e testaram com o classificador *Random Forest*. O evento que influenciam nas flutuações de carga de trabalho não são considerados neste esquema.
B3-Artificial neural network-	**Imam *et al.* (2011)** empregaram um atraso de tempo ANN e um método de regressão para prever *jitter* na carga. Este modelo de regressão aplica-se moderadamente ao traço, como evidenciado pela interpolação *spline*. A análise mostra mais aprimoramento nas técnicas de modelagem de regressão ao lidar com tais traços.
	Yang *et al*. (2015) introduziram o POSITING, um modelo de previsão que conduz a mineração sequencial de padrões, aplica a correlação entre vários recursos e encontra o padrão comportamental das aplicações. Eles investigaram as capacidades do aprendizado online do POSITING para fornecer resultados confiáveis, mas não é adaptável às variações de carga. Como vantagem, este esquema considera a correlação entre diferentes recursos e extrai padrões comportamentais das aplicações de forma independente.
	Kumar *et al.* (2018) propuseram um modelo de previsão de carga usando algoritmo ANN e DE que é capaz de aprender o método de mutação e taxa de cruzamento adequados. As simulações realizadas na NASA forneceram rastreamentos HTTP. Como vantagem, esse esquema evita o risco de ficar preso no ótimo local.
	Lu *et al.* (2016) introduziram o RVLBPNN, um modelo de previsão de carga que usa o algoritmo BPNN para explorar as relações entre as cargas que chegam. O RVLBPNN melhora a precisão da previsão em comparação com os modelos baseados em classificador HMM e Naive Bayes por uma margem considerável. Questões como a periodicidade da carga de trabalho não são consideradas neste esquema.
	Zhou *et al.* (2016) apresentaram uma solução para carga dinâmica baseada em AHPGD e HHGA-RBF ANN que foca no balanceamento de carga da alocação de tarefas de requisição do usuário em uma nuvem. O modelo utiliza um AG hierárquico híbrido e o método recursivo de mínimos quadrados para treinar parâmetros de RNAs RBF. Ele é agregado com o algoritmo *round-robin* ponderado e atualiza os pesos de cada nó dentro do período. Propuseram três módulos em seu algoritmo: - módulo de monitoramento de informações de carga do nó, - módulo de previsão de carga e - módulo de agendamento de solicitações.
	Imam *et al.* apresentaram um esquema de alocação de recursos para suportar a crescente necessidade de VMs. Usaram RNA de atraso de tempo e técnicas de regressão para predição de carga. Eles utilizaram rastreamentos de carga real para avaliação de desempenho para mostrar que a RNA de atraso de tempo pode prever a carga em um ambiente de nuvem.
B4-Bayesian	**Di *et al.*** propuseram um método de previsão para estimar a carga em intervalos de longo prazo e a carga média em intervalos de tempo futuros, com base no modelo de Bayes. Detectaram recursos preditivos da carga para capturar a previsibilidade e o padrão de carga do host. Determinaram as combinações efetivas desses recursos para previsão. Como vantagem, pode-se detectar a carga média para horas futuras com alta precisão e baixo MSE, independentemente de flutuações.
	Dietrich *et al.* forneceram um preditor linear para Mínimos Quadrados Médios, uma identificação de parâmetro do sistema de modelo de regressão. A flutuação

	de carga é estimada por um modelo de parâmetros lineares. Essa observação reduz a complexidade da estimativa de parâmetros à medida que o LMS aprende os parâmetros do modelo de forma iterativa à medida que avança. O LMS nem sempre pode superar um controlador PID ajustado manualmente.
	Tian *et al.* minimizando a reorganização de conteúdo e tolerando a previsão de carga de trabalho imperfeita para serviços de vídeo sob demanda em nuvem **Nguyen *et al***. tentam reduzir a reorganização do conteúdo e tolerar a previsão de carga imperfeita. Eles apresentaram um sistema de manutenção de vídeo sob demanda de acordo com uma nuvem de pagamento conforme o uso. Eles propuseram um absorvedor de carga e projetaram um algoritmo de provisionamento chamado *Absorb Window*. Os absorvedores de carga eliminam o desperdício de largura de banda e reduzem a reorganização do conteúdo. Esquemas baseados em aprendizado profundo. As abordagens de aprendizado profundo são adequadas para previsão de cargas de trabalho a longo prazo e seu desempenho pode ser melhorado aumentando o tamanho dos dados de treinamento e a profundidade de o modelo. Várias abordagens baseadas em aprendizado profundo são fornecidas para prever a carga de trabalho nos DCs de nuvem.
B5 - Deep learning	**Patel *et al.*** buscaram uma correlação entre a carga de trabalho das VMs em relação e previu a carga de trabalho das próximas VMs com precisão. Otimizaram a granularidade dos dados de treinamento, funções de ativação e o número de camadas. Usaram informações de carga de trabalho previstas para gerenciamento de VM e a escolha do plano de migração será transferida para o CSP de aplicativos que receberá a solicitação de usuário aceita e aplicará a estratégia de posicionamento de VM adequada para mapear a VM para PMs. Eles avaliaram a eficácia de seu modelo de aprendizado profundo usando rastreamentos do PlanetLab e mostraram que o LSTM pode melhorar o desempenho da previsão de carga de trabalho, enquanto a ANN convolucional oferece um baixo desempenho. A arquitetura dessa abordagem é mostrada na Fig. 14. Seu modelo recebe a utilização de CPU das VMs como entrada e prevê a utilização de CPU no futuro.
	Gupta *et al.* propuseram modelos LSTM multivariados aplicados para prever o uso de recursos nos DCs de nuvem. Eles usaram os rastreamentos de cluster do Google e avaliaram o modelo LSTM e o modelo LSTM bidirecional com métodos baseados em diferenças fracionárias. Eles indicaram as dependências de longo alcance do modelo LSTM em dados de consumo de recursos baseados em séries temporais e produziram estimativas melhores a partir da amostra. Como vantagem, essas extensões multivariadas dos modelos LSTM e BLSTM geram melhores estimativas do que as univariadas.
	Zhang *et al. (2014)* introduziu um modelo de aprendizado profundo usando a decomposição poliádica canônica para prever a carga da nuvem. Eles usaram o modelo de aprendizado profundo para aprender recursos importantes dos dados de carga complexos em VMs e aplicaram a decomposição poliádica canônica para compactar parâmetros para melhorar a eficiência do treinamento.

C-Baseados em Estocástica	
C1-Baseado em Modelo de Markov	**Pacheco-Sanchez *et al.* (2011)** estudaram as flutuações de carga da web para descobrir como obter recursos virtuais em tráfego flutuante. Eles investigaram os processos de chegada markovianos ou MAP e o modelo de filas M/M/1 relacionado para previsão de desempenho dos servidores implantados. Os MAPs são um tipo especial de modelos de Markov aplicados como uma descrição compacta das características das cargas que variam no tempo. Os MAPs podem ser usados para distribuições de cauda pesada no tráfego HTTP e podem ser aplicados em modelos de filas analíticos para estimar o desempenho do sistema.
	Shen *et al.* (2011) apresentaram o *CloudScale*, para automatizar o dimensionamento de recursos usando previsão de solicitações e tratamento de erros de previsão. Ele lida com conflitos de escala usando a migração. Eles usaram o *CloudScale* em cima do *hypervisor Xen* e realizaram simulações usando o benchmark RUBiS orientado por rastreamentos reais do servidor Web. Como vantagem, este esquema emprega DVFS para mitigar o uso de energia em relação ao SLA.
C2-Hidden markov model	**Khan *et al.*** forneceram uma solução de co-clustering para encontrar grupos de VMs que tenham padrões de carga correlacionados e seus períodos de ativação. Eles introduziram um método baseado em HMM para detectar as correlações temporais nos clusters de VM e prever flutuações em seu padrão.
	Li et al. Prever e categorizar a carga de nuvem de prazo usando uma abordagem de clustering baseada em HMM. O critério de informação Bayesiano e o critério de informação de Akaike são usados para encontrar o tamanho ideal do modelo HMM e os números de cluster. Os HMMs são aplicados para detectar o cluster que pode possuir uma carga atual e com dados, uma rede Oman otimizada para o GA é preparado para que seus GA sejam preparados para uma carga

	futura. Não foi considerado um disco de memória entre cargas de trabalho de CPU e memória.
C3-Modelo de Fila	**Sahni *et al*.** apresentaram uma solução com reconhecimento de heterogeneidade para lidar com as cargas dinâmicas e manter o nível de QoS necessário. Ele realiza estimativas usando perfis de recursos online e histórico de carga de trabalho. Ele também fornece as configurações de recursos necessárias para alcançar QoS a custo reduzido e melhor utilização de recursos. Ele captura a variação de desempenho nas VMs e usa o padrão de chegada de solicitação e a taxa de serviço para configurar recursos. No entanto, esse modelo considera apenas aplicativos independentes e não suporta dependências entre as solicitações recebidas.
	Jian *et al 2013*. forneceu um esquema de dimensionamento automático de recursos em nível de VM para um aplicativo da Web que pode prever suas solicitações para determinar a demanda ideal de recursos usando teoria de filas e otimização multiobjetivo. Esse esquema leva em consideração fatores como custo, latência e fatores SLAV em cada reatribuição de unidade de tempo.

D-Grey predicting-based schemes
Jheng et al. (2014) apresentou uma abordagem de previsão de carga usando modelo de previsão cinza para alocar VMs. Os autores usaram a carga dependente do tempo no mesmo período em cada dia e previram se a tendência da carga do MV é de aumento ou diminuição? Eles compararam o valor previsto com a carga de trabalho do período anterior e decidiram qual VM no PM deve ser migrada para ter uma carga de trabalho equilibrada e menos uso de energia. Seus experimentos indicaram que esse esquema usa menos dados no processo de previsão e pode alocar os recursos das VMs com economia de energia.

E-Autocorrelation clustering-based schemes
Kluge *et al*. empregou agrupamento de autocorrelação para prever a carga de um aplicativo periódico *soft realtime*. Usando esse método de previsão, eles ajustam o desempenho do processador para cumprir todos os prazos. No entanto, eles não lidaram com as instabilidades numéricas induzidas pelo arredondamento implícito durante a execução do algoritmo de agrupamento de autocorrelação

F- Chaos
Ardagna *et al.* utilizaram técnicas aplicadas de alocação de capacidade para coordenar vários controladores de recursos distribuídos trabalhando em sites de nuvem geograficamente distribuídos. As soluções de alocação de capacidade são integradas a um mecanismo de redirecionamento de carga que encaminha as solicitações recebidas entre vários domínios. As vantagens incluem reduzir os custos das VMs alocadas e atender às restrições de QoS, como o tempo médio de resposta.
Qazi *et al*. apresentaram o PoWER que tenta prever o comportamento do cluster e distribui VMs no cluster e desliga PMs não utilizados para reduzir o consumo de energia. Eles usaram a teoria do caos para tornar a previsão indiferente ao tipo de carga e aos ciclos inerentes a ela e, ao conduzir experimentos, indicaram que sua abordagem supera o método de séries temporais baseado em FFT na previsão de carga.

G-Kalman filter
Hu *et al.* apresentaram três modelos para estimar a carga usando um modelo de filtro de Kalman e apresentaram um modelo de correspondência de padrões para prever a carga. Eles aplicaram seus resultados para fornecer uma nova estratégia de gatilho para o mecanismo de dimensionamento automático da elasticidade da nuvem. Esse modelo melhora a precisão da previsão e reduz o atraso de dimensionamento automático, mas deve ser estendido para dar suporte a outros cenários de previsão de carga de trabalho e melhorar sua precisão de previsão.

H- Wavelet-based
Liu *et al.* propuseram uma solução de migração de VM que aplica um algoritmo de previsão de carga baseado em séries temporais. Eles ajustaram os limites superior e inferior de carga para hosts e previram a tendência de suas cargas subsequentes criando uma série temporal de carregamento usando o modelo de nuvem. Posteriormente, eles estipularam um WAM de migração com reconhecimento de carga de VM que escolhe um PM de origem, um PM de destino e uma VM no PM de origem a ser migrado. Além disso, neste esquema, os autores consideraram o

consumo de CPU como carga de trabalho e aplicaram o conjunto de dados PlanetLab e o software CloudSim para avaliação.
Lyu *et al.* introduziram um método de previsão que consiste em um módulo de previsão, um módulo de ajuste e um módulo de coleta. O primeiro módulo aplica métodos de aprendizado de máquina para melhorar a precisão da previsão. Como vantagem, eles introduziram uma maneira eficaz de reconhecer o mecanismo de previsão de taxa de carga de limite duplo para equilibrar disponibilidade e lucro.
Qazi *et al*. apresentaram um método eficiente para prever o comportamento do cluster com base em seu histórico e redistribuir VMs para liberar PMs subutilizados e desligá-los para economizar energia. Eles avaliaram cargas reais e usaram uma série temporal caótica. A teoria do caos com otimizações torna esse framework indiferente ao tipo de carga e aos ciclos inerentes a ela.

I - Collaborative filtering-based
Duggan *et al.* (2013) apresentaram uma solução baseada em aprendizado para previsão de carga para bancos de dados analíticos aplicados por diferentes CSPs. A habilitação de estimativas de desempenho de carga que podem ser portadas em configurações de hardware pode ajudar os usuários da nuvem com suas decisões de compra de serviços e CSPs em suas decisões de provisionamento. Essa abordagem aplica filtragem colaborativa para prever impressões digitais de carga leve que modelam o comportamento de cargas de consulta simultâneas para a escolha de configurações de hardware.
Zhang *et al.* forneceram uma solução de dimensionamento baseada em previsão que usa filtragem colaborativa com uma técnica de correspondência de padrões. Ele aprimora as técnicas de escalabilidade baseadas em regras reativas e fornece um método para vincular o SLA de acordo com as métricas de nível inferior da infraestrutura. No entanto, para ajustar essa abordagem, mais métricas de infraestrutura devem ser consideradas. A Tabela 6 determina o conjunto de dados, o ambiente do software de simulação e os fatores previstos e avaliados na wavelet, esquemas de predição de carga de trabalho baseados em filtros colaborativos baseados em esquemas.

J-Ensemble-based schemes
Embora alguns dos esquemas de previsão de carga de trabalho discutidos anteriormente tenham aplicado um único método de previsão, sua precisão pode não ser a necessária e o comprimento da previsão pode não ser aumentado. Para mitigar esses problemas, vários frameworks de previsão de carga baseados em ensemble foram propostos na literatura e está subseção visa revisá-los.
Cao *et al.* (2014) propuseram um método *ensemble* que utiliza vários modelos para aumentar o desempenho e a previsão de carga da CPU. Eles aplicam um modelo de conjunto de duas camadas que consiste em camadas de previsão e de conjunto. A camada de otimização do preditor aplica novas instâncias do preditor e remove aquelas com baixo desempenho. A camada de conjunto produz a previsão final com base nos resultados de várias instâncias do preditor e pode fornecer feedback para a camada de otimização do preditor, o que a ajuda a adotar estratégias de otimização apropriadas. Nesse esquema, a substituição de preditores é usada em relação à avaliação de desempenho para manter o desempenho de um conjunto de preditores. Então, o preditor mais pobre deve ser removido e outro preditor deve ser adicionado.
Shariffdeen et al. (2016) forneceram um método de previsão para aumentar a precisão nos auto-escaladores usando uma abordagem de previsão de carga baseada em conjunto. Eles avaliaram vários modelos de previsão para prever vários padrões de carga. Essa técnica de conjunto é implementada usando três cargas do mundo real. Eles treinaram cada modelo em tempo real e agregaram os resultados previstos com base nos pesos calculados usando erros inversos dos valores ajustados para os dados de treinamento. É necessário mais trabalho para identificar o tamanho ideal da janela de entrada para maximizar a precisão e atender às restrições temporais no cálculo das previsões em tempo real.
Singh *et al.* (2014) tentaram reduzir o uso de energia, resfriamento e emissões de CO 2 dos PMs para melhorar a sustentabilidade da infraestrutura em nuvem. Eles usaram técnicas de previsão de carga que orientam na identificação de servidores, intervalos de tempo e outros parâmetros críticos necessários nos DCs de nuvem. Esse esquema pode lidar com cargas de trabalho não estacionárias e, ao atualizar seus parâmetros de aprendizado, evita o retreinamento de seus modelos de previsão. Além disso, eles aplicaram a Maioria Ponderada e os Especialistas Simuláveis para lidar com a extensa não estacionariedade e os enormes dados de streaming online.
Somer *et al.* (2016) propuseram o PRUF, um módulo de previsão baseado em conjunto para prever a utilização futura de VMs. Foi proposta uma política de migração de VM proativa usando detecção preditiva de sobrecarga.

K-Hybrid	
SVR (Support Vector Regression) + Kalman filter	**Hu *et al.*** apresentaram o KSwSVR, um método de previsão de carga multi-step-ahead, que integra SVR e Kalman smoother. O rastreamento público é aplicado para verificar sua precisão, estabilidade e adaptabilidade de previsão. O experimento de alocação de CPU indicou que o KSwSVR pode reduzir o uso de recursos enquanto atende aos requisitos de SLA. Neste esquema, o suavizador de Kalman é empregado para reduzir o ruído de dados de uso de recursos, causado por erros de medição.
Deep learning + SVM	**Tarsa *et al.*** utilizaram codificação esparsa hierárquica, que é uma forma de aprendizado profundo para modelar cargas orientadas pelo usuário usando contadores de desempenho de hardware no chip. Eles previram períodos de baixa taxa de transferência de instruções, cuja frequência e tensão podem ser dimensionadas para recuperar energia. Usando uma estrutura de codificação de várias camadas, esse método codifica recursos de valores de contador aprendidos com os dados e os passa para um classificador SVM, onde eles atuam como assinaturas para prever futuros estados de carga.
ARIMA + RNN	**Janardhanan et al.** focaram na previsão de séries temporais no uso de CPU em DCs usando a rede LSTM e a avaliou em relação ao modelo ARIMA.
ARIMA + wavelet decomposition	**Bi et al.** introduziram um método híbrido que usa decomposição wavelet e ARIMA para prever o futuro. A proposta tenta suavizar a série temporal da tarefa usando a filtragem SavitzkyGolay e a decompõe em vários componentes por meio da decomposição wavelet. Os resultados de previsão são reconstruídos via redução de wavelets para estimar o número de tarefas que chegam. Melhores algoritmos de suavização de dados poderão ser usados para melhorar ainda mais a precisão da previsão desse esquema.
LR + SVM	**Liu et al.** propuseram uma abordagem adaptativa para previsão de carga, que classifica a carga em várias classes atribuídas a vários modelos de previsão em relação às características de carga e atribui vários modelos de previsão em relação às características de carga de trabalho. Foi transformado o problema de classificação de carga em um problema de atribuição de tarefas, utilizando um modelo misto de programação inteira 0-1, fornecendo uma solução on-line para isso. Para previsão, foi utilizada regressão linear e SVM, que é bom na previsão de dados não lineares. Como vantagem, esta abordagem melhora os erros de previsão relativos cumulativos da plataforma.
ANN + regression	**Tang et al**. introduziram o MLWNN que aplica regressão linear e ANN *wavelet* para prever a carga de curto prazo. Forneceram um agendamento heurístico de tarefas com reconhecimento de energia com um método de previsão de carga e empregaram o algoritmo de retro propagação de erros para treinar um modelo WNN de *feedforward* de três camadas e obter um erro mínimo. A abordagem de agendamento de tarefas inclui um método de gerenciamento de recursos baseado na previsão de carga de trabalho MLWNN. Apontaram que a abordagem pode reduzir o uso de energia e aumentar a utilização de recursos. Foi utilizado o simulador CloudSim.
	Gandhi *et al.* (2017) buscou melhorar a alocação de recursos em DCs de nuvem para reduzir o uso de SLAV e energia. Foi utilizado um método de provisionamento de recursos preditivos, que lida com estimativa de carga em escalas de tempo grosseiras e provisionamento reativo para lidar com qualquer excesso de carga em escalas de tempo mais finas. A combinação de provisionamento preditivo e reativo alcança uma melhoria no atendimento ao SLA, na economia de energia e na redução dos custos de provisionamento.
ARM + Regressão + SVR	**Guo *et al.*** propuseram o NUP, um método de previsão híbrido, que usa o tipo de carga para alternar algoritmos de previsão. Foi usado coeficientes de autocorrelação e expoentes de Hurst de cargas para determinar se as cargas pertencem ao período ou à tendência. O NUP aplica regressão linear e semelhanças entre períodos para substituir dados ausentes de cargas de tendência e período. Utiliza regressão linear e ARMA para prever a tendência e SVR para prever o período.

Fonte: Masdari, Khoshnevis (2020)

APÊNDICE D - TRABALHOS ANTERIORES - PREDIÇÃO DA CARGA DE TRABALHO EM SERVIÇOS NA NUVEM -

	TÍTULO	RESUMO	MECANISMOS	ACURÁCIA	DATA BASE TRACES
	Workload Prediction over Cloud Server using Time Series Data (Yadav et al., 2021)	A análise preditiva de dados identifica as tendências e permite que as organizações atuem de acordo com sua demanda. Pode ser aplicado em diferentes áreas, como previsão de preços de ações, previsão do tempo e carga de tráfego no servidor (computação em nuvem). Os CSPs utilizam a análise preditiva para evitar perdas, como indisponibilidade de serviços, consumo máximo de energia e perda do cliente.	Análise Preditiva / LSTM / Séries temporais Apresenta análise preditiva da previsão de séries temporais usando o método de aprendizado profundo (LSTM) para prever a carga futura nos servidores.	A precisão de previsão do LSTM foi medida usando três métricas: RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) MSE (Mean Squared Error) Os resultados mostram que a precisão do modelo é a melhor (erro mínimo de previsão).	Complutense University of Madrid. O dataset contém 02 tributos: timestamp e número de hits, respectivamente.
2	Multivariate Deep Learning Model for Workload Prediction in Cloud Computing Dang-Quang et al. 2021	Propõe um modelo de previsão de aprendizado profundo multivariado para prever a carga de trabalho de recursos futura para o ambiente de computação em nuvem. O modelo de previsão usa a rede neural recorrente (RNN) chamada memória bidirecional de longo prazo (Bi-LSTM). Mostra a vantagem de usar dados multivariados em comparação com dados univariados na previsão de séries temporais. Os experimentos, usando o conjunto de dados de carga de trabalho do mundo real, mostram que o modelo Bi-LSTM multivariado proposto supera o modelo Bi-LSTM univariado em precisão de previsão	*Prediction multivariate Bidirectional long short-term memory* (Bi-LSTM). As entradas dos modelos de predição multivariada Bi-LSTM propostos são o throughput de recebimento da rede e o uso da CPU. O Bi LSTM univariado receberá a taxa de transferência da rede como entrada. Ambos os modelos de previsão produzirão a carga de trabalho de taxa de transferência de recebimento de rede futura.	O Bi-LSTM multivariado proposto supera o modelo Bi-LSTM univariado em precisão de previsão Usado a métrica - uso de CPU e taxa de transferência de rede. Dados em 80% para treinamento e 20% para teste.	O GWA-T-12 Bitbrains - desempenho de 1.750 VMs de um datacenter distribuído da Bitbrains. O primeiro rastreamento, *fastStorage*, que consiste em 1.250 VMs conectadas a dispositivos de armazenamento SAN. Cada arquivo no conjunto de dados é composto por linhas, representando uma observação de 300 ms dos dados de desempenho de uma máquina virtual desde 01-01-2018.
3	A functional paradigm for Capacity Planning of Cloud Computing Workloads	Dadas as restrições técnicas e financeiras dos projetos de computação em nuvem, os processos de planejamento de capacidade ajudam a identificar a necessidade de recursos em cada carga de trabalho para que ela funcione adequadamente diante de	Planejamento de capacidade de nuvem não estabelecem a precisão desses planos. Visa estabelecer um paradigma funcional para planejamento de capacidade de cargas de trabalho de computação em		

	Diego Cavalcanti Pereira	diferentes contextos. O paradigma industrial atual considera procedentes históricos de uso para o planejamento de capacidade de cargas de trabalho de computação em nuvem. Além disso, as práticas atuais de planejamento de capacidade de nuvem não estabelecem a precisão desses planos.	nuvem, que inclui: um meta-modelo arquitetural; um método para classificação funcional e arquitetural de cargas de trabalho; uma equação de dimensionamento e índice de confiança.		
4	A hybrid CNN-LSTM model for virtual machine workload forecasting in cloud data center Leka et al. 2021	A proposta apresenta uma estratégia híbrida baseada em aprendizado profundo para previsão de carga de trabalho de VM. Para criar uma previsão precisa, o modelo de previsão sugerido integrou uma arquitetura de rede neural convolucional (CNN) e uma rede neural de memória de longo prazo (LSTM). O componente CNN é usado para obter atributos distintivos complexos dos dados de carga de trabalho da VM, enquanto o componente LSTM modela informações temporais e prevê a carga de trabalho futura da VM. Resultados experimentais no conjunto de dados do mundo real mostraram que o modelo CNN-LSTM é eficaz na previsão de carga de trabalho de VM e, quando comparado a modelos de previsão de carga de trabalho usados com frequência, a abordagem proposta melhora o desempenho de previsão de carga de trabalho	O componente CNN é utilizado para obter atributos distintivos complexos dos dados de carga de trabalho da VM. O componente LSTM modela informações temporais e prevê a futura carga de trabalho da VM. O modelo de previsão usa a utilização da CPU de todas as máquinas virtuais por categoria de carga de trabalho como informações de entrada. Em uma proporção de 70:30, os dados são separados em dados de treinamento e teste. Antes do treinamento e teste, todos os dados de utilização da unidade de processamento central foram normalizados em [0-1]. O vetor de entrada do modelo de aprendizado profundo é a janela do histórico de dados de consumo de CPU de todas as VMs em cada categoria de carga de trabalho. Foram selecionadas todas as 4 VMs por categoria como as que precisamos prever. A tarefa é prever o consumo de CPU de todas as quatro VMs em intervalos de tempo de 1 hora.	RMSE e MAE são usados para medir a precisão da previsão do modelo proposto. Previsão de carga de trabalho da VM O RMSE Médio (A-RMSE) e o MAE Médio (A-MAE) como índice de desempenho do múltiplo	GWA-T-12 Bitbrains - Rastreamento de carga de trabalho de larga escala e de longo prazo de um datacenter em nuvem distribuído. Dados para 1250 VMs em SAN (rede de área de armazenamento rápido). Os recursos como memória, unidade central de processamento, E/S de rede e utilização de E/S de disco foram coletados a cada 5 minutos. A partir do rastreamento de armazenamento rápido do conjunto de dados, foram selecionadas 12 amostras de VMs que exibem os padrões de carga de trabalho mais comuns, ou seja, periódico, crescente e imprevisível.
5	Workload forecasting and resource management	A estimativa proativa da carga de trabalho futura seguida pela decisão de alocação de recursos tornou-se uma solução prévia para lidar com outros desafios embutidos,	Estimativa proativa da carga de trabalho		

	models based on machine learning for cloud computing environments Saxena, 2021	como a *under/overloading* de máquinas físicas, desperdício de recursos, violações de QoS, balanceamento de carga, Migração de VM e muito mais. Apresenta um levantamento de modelos de previsão de carga de trabalho e gestão preditiva de recursos em ambiente de nuvem. Uma estrutura conceitual para previsão de carga de trabalho e gerenciamento de recursos, categorização de técnicas de alocação de recursos baseadas em aprendizado de máquina existentes e os principais desafios da distribuição ineficiente de distribuição de recursos físicos são discutidos em relação à computação em nuvem. É apresentada uma pesquisa das contribuições de última geração que capacitam abordagens baseadas em aprendizado de máquina no campo de previsão de carga de trabalho em nuvem e gerenciamento de recursos. Explora e conclui desafios emergentes e direções de pesquisa futuras sobre gerenciamento de recursos elásticos em ambiente de nuvem.			
6	*A survey on predicting workloads and optimising QoS in the cloud computing Aloufi et al., 2021.*	Apresenta o conceito e as características da computação em nuvem e aborda como a computação em nuvem oferece QoS ao usuário final. Em seguida, discute como agendar a carga de trabalho na infraestrutura usando tecnologias que surgiram recentemente, como Machine Learning (ML). Isso é seguido por uma visão geral de como o ML pode ser usado para gerenciamento de recursos. O objetivo foi delinear os benefícios do uso de ML para agendar demandas futuras para alcançar QoS e economizar energia. Além disso, revisamos a pesquisa relacionada aos métodos de ML para prever cargas de trabalho na computação em nuvem.	Machine Learning		

		Fornece informações sobre as abordagens de elasticidade, enquanto outra seção discute os métodos de previsão usados em estudos anteriores.			
7	*Adapted Convolutional Neural Networks and Long Short-Term Memory for Host Utilization prediction in Cloud data center* *Arif Ullah*	O modelo de serviço de infraestrutura fornece diferentes tipos de recursos de computação virtual, como rede, serviço de armazenamento e hardware, conforme as demandas do usuário. A previsão de carga do host é um elemento importante na computação em nuvem para melhoria nos sistemas de alocação de recursos. Problemas de inicialização de hospedagem ainda existem devido a esse problema, a alocação de recursos de hardware leva minutos de atraso no processo de resposta. Para resolver esse problema, técnicas de previsão são usadas para previsão adequada no data center em nuvem para dimensionar dinamicamente a nuvem para manter uma alta qualidade de serviços. No modelo híbrido proposto, o método de autorregressão vetorial é primeiramente utilizado para inserir os dados para análise que filtra as interdependências lineares entre os dados multivariados. Em seguida, os dados duradouros são computados e inseridos na camada de rede neural Convolucional que extrai recursos complexos para cada unidade de processamento central e componentes de uso da máquina virtual, depois que a memória de curto prazo longa é usada, adequada para modelar informações temporais de tendências irregulares em componentes de séries temporais.	Propomos uma rede neural Convolucional híbrida longa com modelo de memória de curto prazo para previsão de host. Em todo o processo, a principal contribuição é que usamos a função de ativação unitária constante polinomial escalonada que é mais adequada para este tipo de modelo. Devido à maior inconsistência no data center, a previsão precisa é importante em sistemas em nuvem	Os resultados demonstram que o método alcança desempenho de última geração com maior precisão em ambos os conjuntos de dados em comparação com os modelos ARIMA-LSTM, VAR-GRU, VAR-MLP e CNN Dois rastreamentos de carga do mundo real foram usados para avaliar o desempenho. Um é o rastreamento de carga no data center do Google tradicional, enquanto o outro está no sistema distribuído.	Dois rastreamentos de carga do mundo real foram usados para avaliar o desempenho: O rastreamento de carga no data center do Google, enquanto o outro está no sistema distribuído tradicional.
8	*Hybrid Resource Scaling for Dynamic Workload*	Propõe uma abordagem híbrida, hibridizando ambas as abordagens de dimensionamento para aumentar a	CloudSim		CloudSim foi utilizado para a implementação da abordagem desenvolvida.

	in Cloud Computing Daraje Megersa	utilização de recursos e fornece recursos flexíveis que possam satisfazer as solicitações do usuário executando nessa ordem abordagens verticais e horizontais. A proposta é mais eficiente em comparação com as existentes. O dimensionamento é executado seguindo a capacidade da máquina e o valor limite dos recursos.			
9	A Computer Dynamic Index Forecast Model Based on Indicator of Long Short-Term Memory Shihui Liu	Propõe um método para previsão de ações. Em termos de características das ações, considera as características dos indicadores macroeconômicos sociais que podem influenciar os preços das ações. Primeiro, rastreamos indicadores macroeconômicos sociais online. Foi proposto um método (SMILSTM, indicador macroeconômico social de *Long Short-Term Memory*) para prever preços de ações que considera características de indicadores macroeconômicos sociais. Comparado à regressão LSTM com recurso de preço, foi descoberto que o método teve uma grande melhoria na precisão da previsão de ações em experimentos.	SMILSTM, indicador macroeconômico social de LSTM		
10	Forecasting of Cloud Computing Services Workload using Machine Learning Krishan Kumar	Analisa e compara a precisão da previsão de diferentes algoritmos de aprendizado de máquina destinados a prever as cargas de trabalho dos logs do servidor. Realiza estudo comparativo aplicado Regressão Linear (LR), K-Vizinhos Mais Próximos (KNN), Máquina de Vetor de Suporte (SVM), ARMA, ARIMA e Regressão de Vetor de Suporte (SVR) para aplicações web para selecionar o algoritmo de acordo com os recursos de carga de trabalho.	Regressão Linear (LR), K-Vizinhos Mais Próximos (KNN), Máquina de Vetor de Suporte (SVM), ARMA, ARIMA e Regressão de Vetor de Suporte (SVR) para aplicações web para selecionar o algoritmo de acordo com os recursos de carga de trabalho.	Os resultados descrevem que o modelo ARIMA apresenta uma melhoria significativa nas métricas de QoS e melhora na disponibilidade do datacenter em nuvem em um ambiente de nuvem e previsão. Finalmente os resultados apresentados e as conclusões são tiradas.	Os experimentos usaram arquivos de rastreamento reais para avaliar o método mais adequado para prever as cargas de trabalho.
11	A survey and classification of the workload forecasting methods	A previsão precisa da carga de trabalho na computação em nuvem é de grande importância para melhorar o desempenho da nuvem, mitigar os consumos de energia,	Proative Apresenta uma extensa revisão da literatura dos esquemas de previsão de carga de trabalho propostos na	Oportunidades de pesquisa em aberto no campo da previsão de carga de trabalho são focadas e as considerações finais são	

	in cloud computing Masdari; Khoshnevis, 2020	atender ao nível de qualidade de serviço (QoS) exigido, prever o consumo de energia dos data centers (DCs) e melhorar a escalabilidade dos CSPs. No contexto de computação em nuvem, a previsão de carga de trabalho é um problema desafiador e vários esquemas usando aprendizado de máquina, mineração de dados e métodos matemáticos para lidar com esse problema.	literatura para melhorar o gerenciamento de recursos nos DCs em nuvem. Primeiro, fornece o conhecimento necessário sobre o contexto de previsão de carga de trabalho e apresenta uma taxonomia dos esquemas de previsão de carga de trabalho de acordo com o algoritmo de previsão aplicado. Além disso, são ilustradas as principais contribuições desses esquemas e especificadas suas principais vantagens e limitações.	apresentadas.	
12	Amazon EC2 Spot Price Prediction Using Regression Random Forests Veena Khandelwal	As instâncias spot foram introduzidas pelo *Amazon EC2* em 12/2009 para vender sua capacidade ociosa por meio de mecanismo de mercado baseado em leilão. Apesar de seus preços extremamente baixos, o mercado spot de nuvem tem baixa utilização. Como os preços spot são dinâmicos, as instâncias spot são propensas a falhas fora do lance. A complexidade dos lances é outro motivo pelo qual os usuários hoje ainda temem o uso de instâncias pontuais. Este trabalho tem como objetivo apresentar o modelo *Regression Random Forests* (RRFs) para prever preços spot de uma semana e um dia à frente. A previsão ajudaria os usuários da nuvem a planejar com antecedência quando adquirir instâncias pontuais, estimar os custos de execução e ajudá-los na tomada de decisões de licitação para minimizar os custos de execução e a probabilidade de falha fora da licitação. A comparação de previsões de preços à vista com base em RRFs com modelos de aprendizado de máquina não paramétricos existentes revela que a precisão da previsão baseada em RRFs	Modelo *Regression Random Forests* (RRFs) para prever preços spot de uma semana e um dia à frente. A previsão ajudaria os usuários da nuvem a planejar com antecedência quando adquirir instâncias pontuais, estimar os custos de execução e ajudá-los na tomada de decisões de licitação para minimizar os custos de execução e a probabilidade de falha fora da licitação.	Foi medida a precisão preditiva usando MAPE, MCPE, erro OOB e velocidade. Os resultados da avaliação mostram que MAPE < ¼ 10% para 66 a 92 por cento e MCPE < ¼ 15% para 35 a 81 por cento de um dia	Simulações com 12 meses reais do histórico spot do Amazon EC2 para prever preços spot futuros mostram a eficácia da técnica proposta.

		supera outros modelos.			
13	Machine Learning Based Workload Prediction in Cloud Computing Jiechao Gao,	Atender a QoS com recursos econômicos é um desafio para os CSPs porque as cargas de trabalho das VMs sofrem variação ao longo do tempo. É necessário fornecer um método preciso de previsão de carga de trabalho de VMs para provisionamento de recursos para gerenciar com eficiência os recursos da nuvem.	Foi comparado o desempenho de métodos representativos de previsão de carga de trabalho de última geração. Foi sugerido um método para realizar a previsão um certo tempo antes do ponto de tempo previsto, a fim de permitir tempo suficiente para o agendamento de tarefas com base na carga de trabalho prevista. Para melhorar a precisão da previsão, foi introduzido um método de previsão de carga de trabalho baseado em cluster, que primeiro agrupa todas as tarefas em várias categorias e depois treina um modelo de previsão para cada categoria, respectivamente.		Os experimentos orientados a rastreamento com base no rastreamento de cluster do Google demonstram que os métodos de previsão de carga de trabalho baseados em cluster superam outros métodos de comparação e melhoram a precisão da previsão em cerca de 90% tanto na CPU quanto na memória.
14	*A Hybrid Automatic Elasticity Solution for the IaaS Layer based on Dynamic Thresholds and Time Series* *Italo Gervásio*	As soluções auto elásticas mais populares são soluções baseadas em regras com limites fixos. Essas soluções lutam com dois: natureza reativa e problemas de configuração difíceis, Foi proposta uma abordagem híbrida de auto escalonamento que supera as deficiências das soluções baseadas em regras com limites fixos. O auto escalonamento proposto consiste em uma abordagem híbrida, consistindo em um conjunto de predições auto adaptativas e limites dinâmicos baseados em componentes reativos.			
15	A Literature Review and Taxonomy on Workload Prediction in Cloud Data Center (Vashistha, 2020)	A alocação de recursos ao longo do tempo pode levar ao ambiente de execução abaixo do ideal devido ao aumento e queda significativos na carga de trabalho que possuem alguns padrões dependentes do tempo. Portanto, requer algumas técnicas sensíveis ao tempo para otimizar a utilização de recursos no data center em nuvem.	Taxonomia de carga de trabalho que é classificada em (i) preditor de carga de trabalho e (ii) ajuste de modelo. Fornece uma extensa discussão sobre os preditores de carga de trabalho e classificados em temporais e não temporais.		

		Discute as técnicas de previsão de carga de trabalho que preveem a carga de trabalho no ambiente de nuvem e o valor dos guias de carga de trabalho previstos para otimizar os recursos. Além disso, apresentamos a taxonomia de carga de trabalho que é classificada em (i) preditor de carga de trabalho e (ii) ajuste de modelo. Além disso, fornecemos uma extensa discussão sobre os preditores de carga de trabalho e ainda classificados em temporais e não temporais.			
16	*Predictive Auto-scaling of Multi-tier Applications Using Performance Varying Cloud Resources (Iqbal et al., 2022)*	O desempenho do mesmo tipo de recursos de nuvem, como máquinas virtuais (VMs), varia ao longo do tempo, devido à heterogeneidade de hardware, contenção de recursos entre VMs colocalizadas e sobrecarga de virtualização. A variação de desempenho pode ser significativa, apresentando desafios para aprender políticas de provisionamento de recursos específicos de carga de trabalho para dimensionar automaticamente os aplicativos hospedados na nuvem para manter o tempo de resposta desejado. O escalonamento automático de aplicativos de várias camadas usando recursos mínimos é ainda mais desafiador, pois gargalos podem ocorrer em várias camadas simultaneamente. Aborda o problema de usar VMs de desempenho variável para dimensionar automaticamente um aplicativo de várias camadas usando recursos mínimos para lidar com cargas de trabalho que aumentam dinamicamente e atender aos requisitos de tempo de resposta.	Utiliza um método de aprendizado supervisionado para identificar o provisionamento de recursos apropriado para aplicações multicamadas com base na previsão do tempo de resposta da aplicação e na taxa de chegada de requisições. O método aprende um mapa de configuração de transição de estado que codifica estados de alocação de recursos invariáveis às variações de desempenho de VMs subjacentes. O mapa ajuda a usar recursos de variação de desempenho no método de dimensionamento automático preditivo.	A avaliação experimental usando um aplicativo da Web de várias camadas do mundo real hospedado em uma nuvem pública mostra um desempenho aprimorado do aplicativo com recursos mínimos em comparação com os métodos convencionais de dimensionamento automático preditivo.	
17	*Auto-Scaling Cloud Resources using LSTM and*	Os recursos de nuvem de dimensionamento automático visam responder às demandas de aplicativos	O modelo combina Long Short-Term Memory (LSTM) and Reinforcement Learning (RL) baseado no conceito	Foram realizados experimentos em dois rastreamentos de carga de trabalho do mundo real, e os	O NASA-HTTP contendo dois meses de todas as solicitações HTTP para o servidor WWW do

	Reinforcement Learning to Guarantee Service-Level Agreements and Reduce Resource Costs Jiang Zhong	dimensionando automaticamente os recursos de computação em tempo de execução para garantir acordos de (SLAs) e reduzir os custos de recursos. As abordagens existentes recorrem a conjuntos predefinidos de regras para adicionar / remover recursos, dependendo do uso do aplicativo. No entanto, as regras de adaptação ótima são difíceis de conceber e generalizar. Uma abordagem proativa é proposta para executar recursos de nuvem de dimensionamento automático em resposta a mudanças dinâmicas de tráfego.	de Computação Autonômica. O LSTM foi customizado para predizer a carga de trabalho de entrada usando a carga de trabalho histórica de forma eficaz. Aplicar o RL como um tomador de decisão que utiliza o histórico de utilização de recursos e os resultados previstos para obter a ação ideal para escalar ou escalar VMs. Aplica a LSTM para prever o número preciso de solicitações na próxima vez e aplica o *Reinforcement Learning* (RL) para obter a ação ideal para aumentar ou diminuir as máquinas virtuais.	resultados mostram que a abordagem pode garantir que as máquinas virtuais funcionem de forma constante e pode reduzir as violações de SLA em até 10% a 30% em comparação com outras abordagens. Esses rastreamentos de carga de trabalho representam variações de carga realistas ao longo do tempo, o que torna os resultados e as conclusões mais realistas e confiáveis para serem usados em ambientes reais.	NASA Kennedy Space Center na Flórida. O ClarkNet-HTTP contendo o valor de duas semanas de todas as solicitações HTTP para o servidor ClarkNet WWW.
18	*Dynamic provisioning of Cloud resource based on workload prediction Sivasankari Bhagavathiperumal*	Utilizaram várias abordagens para realizar o escalonamento automático. Uma estrutura baseada em provisionamento dinâmico de recursos de nuvem usando previsão de carga de trabalho é discutida. Os CSPs devem garantir mecanismos eficientes de manuseio de recursos para diferentes intervalos de tempo para evitar o desperdício de recursos. Os mecanismos de dimensionamento automático cuidariam do uso desses recursos adequadamente, além de fornecer uma excelente qualidade de serviço. Os pesquisadores usaram várias abordagens para realizar o escalonamento automático. Neste artigo, é discutida uma estrutura baseada no provisionamento dinâmico de recursos de nuvem usando a previsão de carga de trabalho.			
19	*A novel approach to workload prediction using attention-based LSTM encoder-decoder network in cloud*	A carga de trabalho do servidor na forma de clusters na nuvem é um fator chave na manutenção do servidor e no agendamento de tarefas. Como balancear e otimizar recursos de hardware e recursos de computação deve, portanto, receber mais	Previsão de carga de trabalho LSTM, Rede Codificador-Decodificador, Mecanismo de atenção usa Rede codificador-decodificador LSTM baseada em atenção para melhorar a previsão para cargas de trabalho em		Dois conjuntos de dados de nuvens reais foram utilizados - Alibaba cluster-trace-v2018 e Dinda2. O Alibaba cluster-trace-v2018 é fornecido pelo Alibaba Open Cluster Trace Program –

	environment (Zhu et al, 2019).	atenção. Observa-se que a execução desordenada do aplicativo em execução e do lote reduz seriamente a eficiência do servidor. Para melhorar a precisão da previsão de carga de trabalho, é proposto uma abordagem usando a rede codificador-decodificador de memória de longo prazo (LSTM) com mecanismo de atenção. Primeiro, a abordagem extrai os recursos sequenciais e contextuais dos dados históricos da carga de trabalho por meio da rede do codificador. Em segundo, o modelo integra o mecanismo de atenção à rede do decodificador, por meio do qual a previsão de cargas de trabalho em lote pode ser realizada. Terceiro, experimentos realizados no conjunto de dados de rastreamento de carga de trabalho Alibaba e Dinda demonstram que o método alcança desempenho de última geração na previsão de carga de trabalho mista em ambiente de computação em nuvem. Foi também proposto um método de previsão de rolagem, que divide uma longa sequência de previsão em várias sequências pequenas para monitorar e controlar a precisão da previsão. Esse trabalho ajuda a orientar dinamicamente a configuração para balanceamento de carga de trabalho.	lote o mecanismo de atenção é como o comportamento humano ao ler uma frase em que tende a não prestar a mesma atenção a cada palavra na frase, mas, em vez disso, concentre-se em palavras importantes.		versão que contém os rastreamentos de aproximadamente 4.000 máquinas em um período de 8 dias. Cada máquina no cluster fornece aplicativos de longa duração e cargas de trabalho em lote. No conjunto de dados de carga de trabalho Dinda – (Carnegie Mellon University) os traces foram coletados de agosto de 1997 a março de 1998. O *dataset* Dinda contém quatro tipos de carga de trabalho, que se referem a quatro cenários de tempo de execução diferentes.
20	*Cluster trace analysis for performance enhancement in cloud computing environments Hayder H. Maala*	Examina as características, o processo e as ferramentas de download e a análise *Google Cluster Trace* para obter informações sobre um tipo de data de rastreamento semelhante para dados que está no ambiente de nuvem. Foi utilizado o algoritmo de cluster K-means executado no SQL Server para usar a metodologia implementada para melhorar o desempenho do ambiente de nuvem, alocando os dados em clusters. Essa			Um cluster de rastreamento de aproximadamente 12.500 máquinas, conhecido como *Google cluster trace* foi iniciado pelo Google.

		alocação foi destinada a ser usada ao distribuir as próximas tarefas para o cluster mais adequado, depois para a máquina mais adequada que cobre sua necessidade de recursos. O processo de clustering gera alguns clusters dependendo da taxa de CPU para cada tarefa, esses clusters representam as máquinas adequadas para cada intervalo (média) da taxa de CPU que é necessária da próxima tarefa a ser alocada. Dependendo da relação entre tarefas e dados de máquina, m máquinas podem ser selecionadas de cada cluster produzido para calcular a disponibilidade de uso da CPU. Esse cálculo será a pedra de toque da alocação de tarefas futuras em máquinas de cluster de nuvem, dependendo de sua disponibilidade de recursos e sua adequação aos requisitos de recursos de tarefas futuras.			
21	*Elastic Resource Provisioning using Data Clustering in Cloud Service Platform* *Bowen Fei,*	Os tipos de tarefas apresentam uma tendência ascendente conforme o crescimento das demandas de serviço, e os diferentes tipos chegam ao sistema sem regularidade. Além disso, os recursos implantados na nuvem são insuficientes para serem provisionados de forma flexível diante de flutuações óbvias de carga de trabalho. Foi apresentado um método de provisionamento elástico de recursos usando cluster de data em plataforma de serviço em nuvem. A estrutura do método proposto consiste em três componentes: agrupamento de tarefas, previsão da quantidade de tarefas no agrupamento, provisionamento e agendamento de recursos dinâmicos. Na classificação de carga de trabalho, foi proposto um método de agrupamento de conjuntos, que utiliza um novo método de	Método de provisionamento elástico de recursos usando cluster de data em PaaS. A estrutura do método proposto consiste em três componentes principais: agrupamento de tarefas, previsão da quantidade de tarefas no agrupamento, provisionamento e agendamento de recursos dinâmicos. Na classificação de carga de trabalho, foi proposto um método de agrupamento de conjuntos, que utiliza um novo método de tomada de decisão de distância para obter os resultados. O método pode efetivamente particionar as tarefas que chegam em vários clusters com base na similaridade entre as tarefas. Para		Implementamos os experimentos no conjunto de dados de rastreamentos de nuvem do Google e os resultados mostram que nosso método atinge 92,3%, 91,2% e 3679,2 kW/h, respectivamente, em termos de taxa de garantia, utilização de recursos e consumo total de energia, o que demonstra a eficácia do método proposto para provisionamento dinâmico de recursos

		tomada de decisão de distância para obter os resultados. O método pode efetivamente particionar as tarefas que chegam em vários clusters com base na similaridade entre as tarefas. Para cada cluster, prevemos a quantidade de tarefas que chegam no próximo momento por modelo de previsão com base em séries temporais para fornecer referência para o provisionamento de recursos de acompanhamento. Depois, um método de provisionamento de recursos de economia de energia é projetado para fornecer recursos dinamicamente para tarefas em cada cluster para atender aos seus requisitos de desempenho. Implementamos os experimentos no conjunto de dados de rastreamentos de nuvem do Google e os resultados mostram que nosso método atinge 92,3%, 91,2% e 3679,2 kW·h, respectivamente, em termos de taxa de garantia, utilização de recursos e consumo total de energia, o que demonstra a eficácia do método proposto para provisionamento dinâmico de recursos	cada cluster, foi prevista a quantidade de tarefas que chegam no próximo momento por modelo de previsão com base em séries temporais para fornecer referência para o provisionamento de recursos de acompanhamento. Depois, um método de provisionamento de recursos de economia de energia é projetado para fornecer recursos dinamicamente para tarefas em cada cluster para atender aos seus requisitos de desempenho. Implementamos os experimentos no conjunto de dados de rastreamentos de nuvem do Google e os resultados mostram que nosso método atinge 92,3%, 91,2% e 3679,2 kW/h, respectivamente, em termos de taxa de garantia, utilização de recursos e consumo total de energia, o que demonstra a eficácia do método proposto para provisionamento dinâmico de recursos		
22	*A preliminary study of machine learning workload prediction techniques for cloud applications (Kirchoff et al., 2019)*	Ainda existem algumas oportunidades, especialmente na área de provisionamento e dimensionamento de recursos. Como a carga de trabalho pode flutuar muito em determinados ambientes, o provisionamento excessivo é uma prática comum para evitar quedas abruptas de Qualidade de Serviço (QoS) que podem resultar em violações do Acordo de Nível de Serviço (SLA), mas ao preço de um aumento nos custos de provisionamento e consumo de energia. A previsão de carga de trabalho é uma das estratégias pelas quais a eficiência e o custo operacional de uma nuvem podem ser melhorados.			

		Conhecer a demanda antecipadamente permite a alocação prévia de recursos suficientes para manter a QoS e evitar violações de SLA. Apresenta as vantagens e desvantagens de três técnicas de previsão de carga de trabalho quando aplicadas no contexto da computação em nuvem. Nossos resultados preliminares comparam ARIMA, MLP e GRU em diferentes configurações de nuvem para ajudar os administradores a escolher o modelo preditivo mais adequado e eficiente para seu problema específico.			
23	*Cloud datacenter workload estimation using error preventive time series forecasting models Kumar et. al, 2019*	A estimativa de carga de trabalho desempenha um papel vital no gerenciamento eficiente dos recursos da nuvem. Este artigo apresenta a pontuação preventiva de erros (EPS) em modelos de previsão de séries temporais para melhorar a precisão da previsão. O EPS analisa as estimativas mais recentes para capturar a tendência de erro de previsão e a utiliza para obter melhores previsões. Além disso, também propusemos duas métricas para avaliação de acurácia, a saber, previsões na faixa de erro (PER) e magnitude das previsões (MoP). Essas matrizes favorecem um modelo que possui previsões máximas próximas aos valores reais, avaliando o erro e a magnitude da previsão individual. O impacto do EPS na precisão é avaliado usando três modelos de estimativa de carga de trabalho.	Observa-se o modelo de suavização exponencial ponderada preventiva de erros produziu as melhores previsões.	O desempenho foi medido usando as métricas de coeficiente de correlação (CoC), soma do índice de elasticidade (SEI), erro de previsão quadrático médio (MPE), PER e MoP. Os modelos preventivos de erro alcançaram uma melhoria máxima de até 183,9%, 95,4% e 100,0% em relação aos modelos não preventivos de erro em CoC, SEI e MPE, respectivamente. Os modelos preventivos de erro reduziram significativamente o erro de previsão individual abaixo de 25% e as subestimações são reduzidas em um fator máximo de 55,2%. A superioridade do esquema proposto é validada por uma avaliação estatística baseada no teste de classificação sinalizada de Wilcoxon e teste de Friedman com análise post-hoc de Finner.	A análise experimental é realizada em cinco trace de dados e

24	*Dynamic Provisioning of Cloud Resources Based on Workload Prediction Sivasankari Bhagavathiperumal*	Uma plataforma de nuvem é um recurso compartilhado que fornece vários serviços, como software como serviço (SAAS), infraestrutura como serviço (IAAS) ou qualquer coisa como serviço (XAAS) necessário para desenvolver e implantar qualquer aplicativo de negócios. Esses serviços em nuvem são fornecidos como máquinas virtuais (VM) que podem lidar com os requisitos do usuário final. Os provedores de nuvem devem garantir mecanismos eficientes de manuseio de recursos para diferentes intervalos de tempo para evitar o desperdício de recursos. Os mecanismos de dimensionamento automático cuidariam do uso desses recursos adequadamente, além de fornecer uma excelente qualidade de serviço. Os pesquisadores usaram várias abordagens para realizar o escalonamento automático. Neste artigo, uma estrutura baseada no provisionamento dinâmico de recursos de nuvem usando a previsão de carga de trabalho é discutida			
25	*Host load prediction in cloud computing using Long Short-Term Memory Encoder–Decoder Hoang Minh Nguyen*	Como as estratégias reativas causam atrasos na alocação de recursos, são necessárias abordagens proativas que usam previsões. No entanto, devido à alta variação da carga do host da nuvem em comparação com a computação em grade, fornecer previsões precisas ainda é um desafio.	Foi proposto um método de previsão baseado em LSTM-*Encoder–Decoder* (LSTM-ED) para prever tanto a carga média em intervalos consecutivos quanto a carga real em vários passos à frente. Nossa abordagem baseada em LSTM-ED melhora a capacidade de memória do LSTM, que é usado no trabalho anterior recente, construindo uma representação interna de dados de séries temporais. Para avaliar nossa abordagem, realizamos experimentos usando um rastreamento de 1 mês de um data center do Google com mais de doze mil máquinas. Nossos		

			resultados experimentais mostram que, enquanto o LSTM multicamada causa *overfitting* e diminuição da precisão em comparação com o LSTM de camada única, que foi usado no trabalho anterior, nossa abordagem baseada em LSTM-ED atinge com sucesso maior precisão do que outros modelos anteriores, incluindo o recente LSTM.		
26	*Cloud Workload Prediction Using ConvNet and Stacked LSTM Peyman Yazdanian*	Investigar o rastreamento de nuvem para ter uma previsão de tráfego em tempos futuros para tarefas computacionais é de grande popularidade em trabalhos anteriores. Foi combinado 1D ConvNets e pilha de LSTM para processar uma longa sequência de dados de rastreamento do Google para ter uma previsão de computação precisa e leve de solicitações de RAM e CPU em *time stamps* futuros.	1D ConvNets e stack of Long-short term memory (LSTM)	Os resultados confirmam que a abordagem, embora tenha alta precisão, também não envolve cálculos pesados e funciona eficientemente com sequências longas.	*Google Trace*
27	*Workload prediction in cloud using artificial neural network and adaptive differential evolution Kumar et al. 2018*	Os principais desafios da computação em nuvem incluem dimensionamento dinâmico de recursos e consumo de energia. Esses fatores levam um sistema em nuvem a se tornar ineficiente e dispendioso. A previsão da carga de trabalho é uma das variáveis pelas quais a eficiência e o custo operacional de uma nuvem pode ser melhorado. A precisão é o componente chave na previsão de carga de trabalho e a abordagens demoram a produzir resultados 100% precisos. Os pesquisadores também estão colocando seus esforços consistentes para o seu aperfeiçoamento. Neste artigo, apresentamos um modelo de previsão de carga de trabalho usando rede neural e algoritmo de evolução diferencial auto adaptativo. O modelo é capaz de aprender a mutação mais adequada estratégia	Rede neural e algoritmo de evolução diferencial auto adaptativo.		Os experimentos foram realizados nos dados de referência conjuntos de rastreamentos HTTP dos servidores da NASA e Saskatchewan para diferentes intervalos de previsão. Os dados usados para experimentos são rastreamentos HTTP da NASA e rastreamentos HTTP de Saskatchewan. Esses conjuntos de dados de D1 e D2, respectivamente. NASA HTTP contém dois rastreamentos com valor de dois meses de todas as solicitações HTTP para o servidor WWW do Centro Espacial Kennedy da NASA na Flórida. Os dados HTTP de

		juntamente com a taxa de cruzamento ótima. Os experimentos foram realizados nos dados de referência conjuntos de rastreamentos HTTP dos servidores da NASA e Saskatchewan para diferentes intervalos de previsão. Nós comparamos os resultados com modelo de previsão baseado no conhecido algoritmo de aprendizado de retro propagação e recebido melhora significativa. O modelo proposto atingiu um deslocamento de até 168 vezes na redução de erros e o erro de previsão é reduzido até 0,001			Saskatchewan são sete meses de logs HTTP de um servidor WWW da universidade. Os rastreamentos são armazenados em arquivos ASCII com uma linha por solicitação
28	*Long Short-Term Memory Recurrent Neural Network (LSTM-RNN) Based Workload Forecasting Model for Cloud Datacenters* Kumar *et al., 2017*	São abordados dois problemas no *datacenter* em nuvem por meio da previsão de carga de trabalho. O modelo de previsão de carga de trabalho é desenvolvido usando redes LSTM.	LSTM	Os resultados empíricos mostram que o método proposto alcançou alta precisão nas previsões ao reduzir o erro quadrático médio em até 3,17 x 10-3.	Utilizado três conjuntos de dados de referência de logs de servidores web. NASA server / Calgary server / Saskatchewan server
29	*Auto-scaling web applications in clouds: A cost-aware approach Mohammad Sadegh Aslanpoura*	O processo de Auto-scaling é frequentemente implementado com base nas quatro fases do loop MAPE. Foi buscado melhorar o desempenho desse mecanismo com soluções diferentes para cada fase. As soluções são focadas na melhoria do desempenho nas três fases de monitoramento, análise e planejamento, enquanto a fase de execução é considerada com menor frequência. Apresenta um executor profissional de economia de custos que mostra a importância e eficácia desta fase do ciclo de controle. Ao contrário dos executores comuns, a solução proposta executa comandos de redução por meio da seleção consciente de máquinas virtuais excedentes; além disso, com seus novos recursos, as máquinas	MAPE: Monitoramento (M), Análise (A), Planejamento (P) e Execução (E).	Os resultados da simulação mostram que o executor proposto reduz o custo de aluguel de máquinas virtuais em 7% enquanto melhora o contrato de nível de serviço final do provedor da aplicação e controla a oscilação do mecanismo na tomada de decisão	

		virtuais excedentes são mantidas em quarentena pelo restante do período de cobrança para maximizar a eficiência de custos.			
30	*Framework for Predictive Analytics as a Service using ensemble model Babu et al,, 2017*	Na era do big data, as organizações exigem valor da big data. Para isso, eles não precisam implantar infraestrutura complexa, mas podem usar serviços que agreguem valor. Como tal, existe a necessidade de um serviço flexível e escalável chamado *Predictive Analytics as a Service (PAaaS*). A análise preditiva pode prever tendências, determinar probabilidades estatísticas e agir sobre fraudes e ameaças de segurança para aplicativos de big data, como trading de negócios, detecção de fraudes, investigação de crimes, bancos, seguros, segurança corporativa, governo, saúde, comércio eletrônico e telecomunicações. Algoritmos de previsão podem ser supervisionados ou não supervisionado com configurações diferentes, e o ideal pode ser diferente para cada tipo de dados.	Resume estruturas de serviço existentes para big data e propõe uma estrutura PaaS que pode ser usada pelas empresas para lidar com a previsão em big data.	Serviço flexível e escalável chamado *Predictive Analytics as a Service (PAaaS*). Este framework proposto é baseado no modelo ensemble que utiliza o melhor dos algoritmos de previsão como Redes Neurais Artificiais (RNA), algoritmo de Autorregressão (ARX) e processo Gaussiano (GP).	
31	*Survey on prediction models of applications for resources provisioning in cloud. Maryam Amiri, 2017.*	A velocidade de resposta às mudanças na carga de trabalho para atingir o nível de desempenho desejado é uma questão crítica para a elasticidade da nuvem. Para tanto, deve-se determinar a demanda futura de aplicações. Assim, a previsão da aplicação em diferentes aspectos (carga de trabalho, desempenho) é uma etapa essencial antes do provisionamento de recursos. De acordo com os resultados da previsão, os recursos suficientes são alocados às aplicações no tempo adequado de forma que a QoS seja garantida e a violação do SLA seja evitada. Revisa os métodos de previsão de aplicação em diferentes aspectos.	Apresentada uma taxonomia para os modelos de previsão de aplicação que investiga as principais características e desafios dos diferentes modelos. Questões de pesquisa em aberto e tendências futuras da previsão de aplicação são discutidas.		

32	*Empirical Evaluation of Workload Forecasting Techniques for Predictive Cloud Resource Scaling In Kee kim*	Muitas abordagens de dimensionamento de recursos preditivos foram propostas para superar as limitações das abordagens reativas convencionais. Em geral, devido à complexidade das nuvens, essas abordagens reativas eram frequentemente forçadas a fazer suposições limitantes significativas nas condições/requisitos operacionais ou nos padrões de carga de trabalho esperados. Como tal, é extremamente difícil para os usuários de nuvem saber qual – se houver – preditor de carga de trabalho existente funcionará melhor para sua atividade de nuvem específica, especialmente ao considerar padrões de carga de trabalho altamente variáveis, modelos de cobrança não triviais, variedade de recursos para adicionar/ subtrair etc. Para resolver esse problema, foram realizadas avaliações abrangentes para uma variedade de preditores de carga de trabalho em configurações de nuvem do mundo real. Os preditores de carga de trabalho abrangem quatro classes de 21 preditores: métodos ingênuos, de regressão, temporais e não temporais.		A avaliação confirma que nenhum preditor de carga de trabalho é melhor para todos os padrões de carga de trabalho e mostra que o *Predictive Scaling-out + Predictive Scaling-in* tem a melhor relação custo-benefício e a menor taxa de perda de prazo de trabalho no gerenciamento de recursos de nuvem, proporcionando em média 30% melhor eficiência de custo e 80% menos taxa de perda de prazo de trabalho em comparação com outros estilos Foi simulado um aplicativo em nuvem sob quatro padrões de carga de trabalho realistas, dois modelos de cobrança em nuvem diferentes e três estilos diferentes de dimensionamento preditivo.	
33	*Automatic Cloud Resource Scaling Algorithm based on Long Short-Term Memory Recurrent Neural Network Ashraf A. Shahin*	A escalabilidade é uma característica importante da computação em nuvem. Com escalabilidade, o custo é minimizado pelo provisionamento e liberação de recursos de acordo com a demanda. A maioria dos CSPs atuais de infraestrutura como serviço (IaaS) oferece técnicas de dimensionamento automático baseadas em limites. Configurar limites com valores corretos que minimizem custos e alcancem o SLA não é uma tarefa fácil, especialmente com mudanças de carga de trabalho variantes e repentinas. Foi	Rede Neural Recorrente de Memória de Longo Prazo e recursos virtuais de dimensionamento automático com base em valores previstos.	Os algoritmos foram avaliados e comparados com algoritmos existentes. Resultados mostram que os algoritmos propostos superam outros algoritmos.	NASA-Real trace called NASA Log Nasa-http, two months of http logs from NASA Kennedy Space Center WWW *server in* Florida, USA. [online] http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html (Accessed on October 1, 2016)

		proposto algoritmos de dimensionamento automático baseados em limites dinâmicos que preveem recursos necessários usando Rede Neural Recorrente de Memória de Longo Prazo e recursos virtuais de dimensionamento automático com base em valores previstos.			
34	*Adaptive Workload Prediction for Proactive Auto Scaling in PaaS Systems* *R.S. Shariffdeen*	O escalonamento automático reativo, no qual as ações de escalonamento ocorrem logo após atingir os limites de acionamento, sofre de vários problemas, como o risco de provisionamento insuficiente em cargas de pico e provisionamento excessivo em outros momentos. Soluções de dimensionamento proativo, em que a demanda futura de recursos pode ser prevista e ações de dimensionamento necessárias decretadas antecipadamente, podem superar esses problemas. A eficácia de tais soluções de escalonamento proativo depende da precisão dos métodos de previsão adotados. - Foi proposta uma técnica de previsão para melhorar a precisão da previsão de carga de trabalho em autos escaladores cloud. - Um mecanismo de previsão de carga de trabalho baseado em séries temporais e técnicas de aprendizado de máquina foi proposto para fazer previsões mais precisas em padrões de carga de trabalho diferentes.	Foram avaliados modelos de previsão quanto à sua aplicabilidade na previsão de diferentes padrões de carga de trabalho. A técnica de *ensemble* proposta foi implementada usando três modelos de reformulação bem conhecidos e testado para três cargas de trabalho do mundo real.		Os resultados mostraram que o método *ensemble* produz erros de previsão menores em comparação com o uso de modelos individuais e a técnica de previsão empregada no Apache Stratos - plataforma PaaS de código aberto
35	*Predictive Cloud resource management framework for enterprise workloads* *Balaji et al. 2018*	Propõe *Predictive Resource Management Framework* (PRMF) para superar as desvantagens das abordagens reativas. O desempenho foi comparado com a abordagem reativa, implantando um aplicativo de quadro de horários na nuvem. As principais métricas dos padrões de		A abordagem preditiva proposta teve um desempenho melhor do que a abordagem reativa ao provisionar / desprovisionar instâncias durante os experimentos em tempo real.	

		carga de trabalho simulados foram monitoradas e analisadas *offline* usando o módulo de ganho de informações presente no PRMF para determinar a principal métrica de avaliação. Posteriormente, o modelo de melhor ajuste para a métrica de avaliação chave entre ARIMA, suavização exponencial (*Single, Double & Triple*) e o modelo oculto de Markov, presente na biblioteca PRMF foi determinado. O modelo de melhor ajuste foi usado para prever a métrica de avaliação principal. Durante o tempo real, o módulo de validação do PRMF compararia continuamente a métrica de avaliação de chave real e prevista. O modelo de melhor ajuste seria reavaliado se o nível de confiança de 95% do valor previsto violasse a métrica real. Para os experimentos realizados no estudo atual, *Request Arrival* e ARIMA (2, 1, 3) foram considerados a principal métrica de avaliação e o modelo de melhor ajuste, respectivamente.			
36	*Dynamic Resource Scaling in Cloud Using Neural Network and Black hole Algorithm Kumar et al. 2016.*	O dimensionamento dinâmico de recursos e o consumo de energia levam um sistema em nuvem a ser ineficiente e caro. A previsão de carga de trabalho é um dos fatores pelos quais a eficiência de uma nuvem pode ser melhorada e o custo operacional reduzido.	Foi apresentado um modelo de previsão de carga de trabalho usando rede neural e algoritmo de buraco negro.	Foi conseguida uma melhoria no erro quadrático médio de até 134 vezes em relação à propagação reversa.	Utilizados os conjuntos de dados de referência de rastreamentos HTTP dos servidores web da NASA, Calgary e Saskatchewan.
37	*Resource provisioning and scheduling in clouds: QoS perspective Sukhpal Singh*	O provisionamento de recursos apropriados para cargas de trabalho na nuvem depende dos requisitos de QoS dos aplicativos. No ambiente de nuvem, heterogeneidade, incerteza e dispersão de recursos encontram um problema de alocação de recursos, que não pode ser resolvido com as estruturas de gerenciamento de recursos existentes. O		As cargas de trabalho na nuvem foram reagrupadas usando o algoritmo de *clustering* baseado em *k-means* após primeiro agrupá-las por meio de padrões de carga de trabalho para identificar os requisitos de QoS de uma carga de trabalho e, em seguida, com base nos requisitos	

		agendamento de recursos, se feito após o provisionamento eficiente de recursos, será mais eficaz e os recursos da nuvem serão agendados de acordo com os requisitos do usuário (QoS). A execução de cargas de trabalho na nuvem deve ser de acordo com os parâmetros de QoS para satisfazer totalmente o consumidor da nuvem. Portanto, com base em parâmetros de QoS, é obrigatório prever e verificar o provisionamento de recursos antes do agendamento real de recursos. Foi apresentada uma estrutura de provisionamento e agendamento de recursos que atende à distribuição de recursos provisionados e ao agendamento de recursos. As cargas de trabalho na nuvem foram reagrupadas usando o algoritmo de *clustering* baseado em *k-means* após primeiro agrupá-las por meio de padrões de carga de trabalho para identificar os requisitos de QoS de uma carga de trabalho e, em seguida, com base nos requisitos de QoS identificados, os recursos são provisionados antes do agendamento real. Além disso, o agendamento foi feito com base em diferentes políticas. O desempenho da proposta foi avaliado em ambiente de nuvem real e simulado e os resultados experimentais mostram que o framework provisiona e programa recursos de forma eficiente considerando consumo de energia, custo de execução e tempo de execução como parâmetros de QoS		de QoS identificados, os recursos são provisionados antes do agendamento real.	
38	*Comparative Study of Scheduling Algorithms to Enhance the Performance of*	Permitir o acesso a recursos remotos e distribuídos geograficamente com a ajuda da virtualização. O agendamento é necessário para gerenciar muitas solicitações de VM. O princípio do	Foi pesquisado diferentes tipos de algoritmos de escalonamento e tabulado seus parâmetros e fatores de escalonamento.		

	virtual machines in Cloud Computing. Kavyasri M. N.	algoritmo de agendamento foi fornecer proficiência no agendamento de tarefas e recursos. O objetivo foi fazer análise comparativa do algoritmo de escalonamento existente na plataforma onde os recursos possuem custo e eficiência computacional variados.			
39	*Cloud resource management: A survey on forecasting and profiling models. Rafael Weingärtner*	Para gerir aplicações e recursos de forma eficaz é fundamental a utilização de modelos e ferramentas que criem um perfil de aplicação que é utilizado para aplicar modelos de previsão para determinar a quantidade de recursos mais adequada para cada carga de trabalho. Existem modelos e ferramentas que tratam da criação de um perfil de aplicação para posteriormente aplicar alguma técnica de previsão e estimar a quantidade de recurso necessária para uma carga de trabalho.	A criação de perfil de aplicativo é uma técnica usada para descrever o uso de recursos de computação por um aplicativo e seus comportamentos esperados. Ele deve ser usado por CSPs para melhor entender e gerenciar aplicativos e recursos.	Apresenta uma taxonomia para modelos e ferramentas de perfis de aplicativos, apresentando suas principais características, desafios, descrevendo e comparando tais modelos e ferramentas. Apresenta ainda uma discussão sobre o uso de perfis de aplicativos e suas tendências futuras de pesquisa	
40	*A Review of Auto-scaling Techniques for Elastic Applications in Cloud Environments. Lorido-Botran et al. 2014*	O principal problema é como alugar o número certo de recursos, com base no pagamento conforme o uso. O redimensionamento da aplicação pode ser implementado sem esforço, adaptando os recursos atribuídos à aplicação à demanda de entrada do usuário. A identificação da quantidade certa de recursos a serem alugados para atender ao SLA exigido, mantendo o custo geral baixo, não é uma tarefa fácil.	Muitas técnicas foram propostas para automatizar o dimensionamento de aplicativos. Foi proposta uma classificação dessas técnicas em cinco categorias principais: regras baseadas em limites estáticos, teoria de controle, aprendizado por reforço, teoria de filas e análise de séries temporais. Em seguida, foi utilizada essa classificação para realizar uma revisão de literatura de propostas de auto escalonamento na nuvem.		
41	*Autonomic* Workload and Resource Management of Cloud Computing Services. (Fargo *et al*. 2014)	Método autônomo de gerenciamento de energia e desempenho para sistemas em nuvem, a fim de combinar dinamicamente os requisitos do aplicativo com recursos de sistema *suficientes* em tempo de execução que levam a uma redução de energia, ao mesmo tempo em que atendem aos requisitos de qualidade de serviço dos		Os resultados mostram que a abordagem pode reduzir o consumo de energia em até 87% quando comparada à estratégia de alocação de recursos estática, 72% em comparação com a estratégia de escala de frequência adaptativa e 66% em	Foi utilizado o benchmark RUBiS, um modelo de leilão emulando transações do eBay que gera uma ampla gama de cargas de trabalho.

		aplicativos em nuvem. A solução oferece: 1) monitoramento em tempo real dos recursos da nuvem e do comportamento da carga de trabalho executada em VMs. 2) determinação do ponto operacional atual das cargas de trabalho e das VMs que executam essas cargas de trabalho; 3) caracterização do comportamento da carga de trabalho e previsão do próximo ponto de operação para as VMs; 4) gerenciamento dinâmico dos recursos da VM (aumentando e reduzindo o número de núcleos, frequência de CPU e quantidade de memória) em tempo de execução; 5) atribuição dos recursos de nuvem disponíveis que possam garantir o consumo de energia ideal sem sacrificar os requisitos de QoS das cargas de trabalho de nuvem.		comparação com uma estratégia de gerenciamento multi-recurso.	
42	*A Science Cloud Resource Provisioning Model using Statistical Analysis of Job History* Seoyoung Kim	Modelo de provisionamento de recursos em nuvem usando análise estatística do histórico de tarefas. O modelo pode fornecer gerenciamento eficiente de recursos de nuvem para um CSP e reduzir a sobrecarga de gerenciamento na nuvem	Foi usado o histórico de tarefas gerado a partir de muitas execuções de aplicativos e identifica as características de um aplicativo por meio da aplicação de análise estatística. Os fatores efetivos são usados para selecionar o perfil de trabalho de referência e, em seguida, a VM é implementada no nó selecionado com base no perfil de referência.	Foi utilizada a técnica PCA (*Principal Component Analysis*) para analisar o histórico de execução das aplicações e extrair os fatores que contribuem para o tempo de execução	Uma aplicação foi executada e seu desempenho foi incorporado ao histórico de trabalho com a finalidade de avaliar o crédito do perfil
43	*Efficient Autoscaling in the Cloud using Predictive Models for Workload Forecasting* *Roy et al. 2011*	Aplicativos corporativos baseados em componentes de grande escala que aproveitam os recursos da Nuvem esperam garantias de Qualidade de Serviço (QoS) de acordo com acordos de nível de serviço entre o cliente e os CSPs. Na computação em nuvem, os mecanismos de escalonamento automático prometem	Apresenta três contribuições para superar a falta geral de técnicas eficazes para previsão de carga de trabalho e alocação ideal de recursos. Primeiro, discute os desafios envolvidos no escalonamento automático na nuvem. Em segundo, desenvolve um algoritmo de previsão	Foram fornecidos resultados empíricos que demonstram que os recursos podem ser alocados e desalocados pelo algoritmo de forma que a satisfaça, tanto a QoS da aplicação, mantendo os custos operacionais baixos.	

		garantir propriedades de QoS para os aplicativos, ao mesmo tempo em que fazem uso eficiente dos recursos e mantêm os custos operacionais baixos para os CSPs. Apesar das vantagens percebidas do escalonamento automático, é difícil realizar todo o potencial do escalonamento automático devido a vários desafios decorrentes da necessidade de estimar com precisão o uso de recursos diante da variabilidade significativa nos padrões de carga de trabalho do cliente.	de modelo para previsão de carga de trabalho que é usado para dimensionamento automático de recursos.		
44	*Empirical prediction models for adaptive resource provisioning in the cloud* *Sadeka Islam*	A inicialização de uma nova instância virtual em uma nuvem não é instantânea; plataformas de hospedagem em nuvem introduzem vários minutos de atraso na alocação de recursos de hardware. Neste artigo, desenvolvemos estratégias de provisionamento e medição de recursos baseadas em previsão usando Rede Neural e Regressão Linear para atender às próximas demandas de recursos.	Rede Neural e Regressão Linear	Os resultados demonstram que a técnica proposta oferece gerenciamento de recursos mais adaptativo para aplicativos hospedados no ambiente de nuvem, um mecanismo importante para alcançar a alocação de recursos sob demanda na nuvem.	TPC-W é uma especificação para benchmarking de *e-commerce*, para provisionamento de recursos, escalabilidade e planejamento de capacidade. O gerador imita várias sessões de usuário simultâneas para um site de *e-commerce*, onde os usuários podem navegar, solicitar e realizar transações de produtos em uma loja de varejo online por meio de navegadores emulados. Foi usada uma versão Java do TPC-W que emula uma livraria online e implantado o aplicativo na nuvem AWS EC2 visando fazer previsões para um site de *e-commerce* onde qualquer solicitação do usuário precisa ser atendida instantaneamente. O foco foi construir modelos de previsão que possam prever antecipadamente um aumento repentino na necessidade de recursos; por esta razão,

					customizamos a implementação do cliente TPC-W para gerar requisições da web de forma linear, uma vez que se assemelha à fase de aceleração do tráfego da web flash crowd. Em trabalhos futuros serão envolvidos outros padrões de carga de trabalho, como aumento exponencial etc. Foi utilizado a API de dimensionamento automático da AWS para definir regras de dimensionamento estático para provisionamento de recursos da livraria online no EC2. Foi coletada a porcentagem agregada de uso da CPU de todas as instâncias do EC2 por minuto; por esse motivo, a utilização da CPU às vezes vai além de 100% para os gráficos de utilização real e prevista
45	*A Hybrid Reinforcement Learning Approach to Autonomic Resource Allocation Gerald Tesauro**	A Aprendizagem por Reforço (RL) fornece uma abordagem promissora para o gerenciamento de desempenho de sistemas que difere radicalmente das abordagens teóricas de filas padrão que fazem uso de modelos explícitos de desempenho do sistema. A RL aprende automaticamente políticas de gerenciamento de alta qualidade sem um modelo de desempenho ou modelo de tráfego explícito e com pouco ou nenhum conhecimento específico do sistema integrado. Anteriormente foi mostrada a viabilidade do uso de RL online para aprender estimativas de valoração de recursos (na forma de tabela de pesquisa) que podem ser usadas para tomar decisões de alocação de servidor de	Foi identificado que Aprendizagem por Reforço (RL) pode lidar com transitórios e atrasos de comutação, que estão fora do escopo da teoria tradicional de filas em estado estacionário. Foi usado RL para treinar aproximadores de funções não lineares (*perceptrons* multicamadas) em vez de tabelas de pesquisa; isso permite o dimensionamento para espaços de estado substancialmente maiores.	Os resultados mostram que, tanto no tráfego de malha aberta ou fechada, o treinamento de RL híbrido pode alcançar melhorias significativas de desempenho em uma variedade de políticas iniciais baseadas em modelo.	

		alta qualidade em um multi-protótipo de aplicativo cenário de Data Center. Agora está sendo mostrado como combinar os pontos fortes dos modelos RL e de filas em uma abordagem híbrida, na qual RL treina *offline* nos dados coletados enquanto uma política de modelo de filas controla o sistema. Ao treinar offline, evita-se um desempenho ruim no treinamento online ao vivo.			

ACM Digital Library - IEEE Xplore - ScienceDirect – Elsevier.