

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Davi Felipe Ramos de Oliveira Vilarinho

**Análise de Técnicas de Redução de  
Amostragem em Sistemas de Orquestraçāo  
Inteligentes**

**Uberlândia, Brasil**

**2025**

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Davi Felipe Ramos de Oliveira Vilarinho

**Análise de Técnicas de Redução de Amostragem em  
Sistemas de Orquestração Inteligentes**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, como parte dos requi-  
sitos exigidos para a obtenção título de Ba-  
charel em Ciência da Computação.

Orientador: Rafael Pasquini

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Davi Felipe Ramos de Oliveira Vilarinho

## **Análise de Técnicas de Redução de Amostragem em Sistemas de Orquestração Inteligentes**

Trabalho de conclusão de curso apresentado  
à Faculdade de Computação da Universidade  
Federal de Uberlândia, como parte dos requi-  
sitos exigidos para a obtenção título de Ba-  
charel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 14 de abril de 2025:

---

**Rafael Pasquini**  
Orientador

---

**Rodrigo Sanches Miani**

---

**Marcelo Zanchetta do Nascimento**

Uberlândia, Brasil  
2025

# Resumo

Este trabalho aborda o desafio da sobrecarga que a quantidade de métricas coletadas e em trânsito pelo monitoramento pode impor em um sistema de orquestração baseado em aprendizado de máquina. São propostas e avaliadas três técnicas capazes de reduzir a frequência de coleta das amostras: eliminação de amostras intermediárias, agregação em janelas com funções e a descrição da distribuição da janela com 8 estatísticas para cada métrica. Os resultados da eliminação de amostras representam a importância da manutenção do equilíbrio da quantidade de características com a de amostras, já que os resultados foram satisfatórios para 128 e 256 segundos apenas quando utilizada seleção de características, mas limitou-se a 16 segundos quando treinado com a integralidade das métricas. A agregação de janelas com funções, entretanto, alcançou boas janelas de até 64 segundos, e, acima deste valor, concluiu-se que o pré-processamento pode alterar negativamente a distribuição das amostras. Por fim, descrever cada métrica em 8 novas estatísticas descritivas pôde alcançar uma janela de até 128 segundos, porém exigiu a utilização de seleção de características dada a alta dimensionalidade, já que, sem a utilização da técnica, as janelas alcançaram 32 segundos satisfatoriamente. Conclui-se que a alta dimensionalidade deve ser acompanhada da adequada quantidade de amostras.

**Palavras-chave:** aprendizado de máquina, QoS, engenharia de características, monitoramento, vídeo.

# Listas de ilustrações

Figura 1 – Associação entre as amostras do conjunto de dados $X$ e $Y$ original com exemplos de métricas reais coletadas. As amostras de $X$ e $Y$ são associadas pelo <i>timestamp</i> de coleta, considerando dez dígitos do <i>Unix Timestamp</i> (escala de segundos). Fonte: Elaborado pelo autor. . . . .	20
Figura 2 – Comparação dos histogramas da métrica de Quadros por Segundo de cada conjunto de dados. A denominação “Uma Aplicação” refere-se ao conjunto de dados coletado em experimentos executando apenas a aplicação de vídeo sob demanda. A denominação “Ambas Aplicações” refere-se ao conjunto de dados coletado em experimentos executando ambas as aplicações em paralelo, vídeo sob demanda e banco de dados distribuído ( <i>Voldemort</i> ). Periódico e Repentino referem-se ao padrão do gerador de carga. Fonte: Elaborado pelo autor. . . . . . . . . . .	21
Figura 3 – Ilustração do procedimento de <i>Univariate Feature Selection</i> ao aplicar um teste F e encontrar as 12 características mais relevantes que formam o conjunto de dados $X_{reduzido}$ . Fonte: Elaborado pelo autor. . . . .	24
Figura 4 – Arquitetura proposta neste trabalho. Métricas brutas coletadas pelo sistema de monitoramento da infraestrutura ( $X$ ) são pré-processadas (eliminação de amostras, agregação em janelas ou descrição estatística, descritas neste capítulo) e opcionalmente submetidas à seleção de características, gerando $X'$ . Esses dados são utilizados para estimar métricas de QoS associadas à cada amostra ( $Y$ ) que alimentam decisões de orquestração. Fonte: Elaborado pelo autor. . . . . . . . . . .	28
Figura 5 – Ilustração de como o Experimento 1 seleciona as amostras a cada $p$ segundos e elimina as restantes, neste exemplo $p = 4$ . Fonte: Elaborado pelo autor. . . . . . . . . . .	29
Figura 6 – Ilustração do pré-processamento realizado pelo Experimento 2 com $j = 4$ . Fonte: Elaborado pelo autor. . . . . . . . . . .	30
Figura 7 – Ilustração do pré-processamento do Experimento 3 realizado com $j = 16$ . Cada métrica em uma janela de $j$ segundos é transformada em 8 diferentes características estatísticas. Fonte: Elaborado pelo autor. . . .	32
Figura 8 – Exemplificação do corte realizado nos conjuntos de dados em 7200 amostras, equivalente a aproximadamente 20% dos conjuntos de dados originais. Fonte: Elaborado pelo autor. . . . . . . . . . .	33

Figura 9 – <i>Boxplot</i> da NMAE do Experimento 1 comparando a utilização de Seleção de Características. As linhas representam os resultados médios da literatura (PASQUINI; STADLER, 2017), a contínua quando não utilizada seleção de características e a tracejada com uso da técnica. Fonte: Elaborado pelo autor. . . . .	36
Figura 10 – <i>Boxplot</i> da NMAE para diferentes tamanhos de janela $j$ Experimento 2. Cada caixa representa uma função de agregação utilizada no pré-processamento e se considera a utilização de seleção de características ou não para o valor de $j$ associado. As linhas representam os resultados médios da (PASQUINI; STADLER, 2017), onde a contínua é a NMAE média sem utilizar a seleção de característica e a tracejada com uso da técnica. Fonte: Elaborado pelo autor. . . . .	40
Figura 11 – <i>Boxplot</i> da NMAE do Experimento 3 comparando a utilização de Seleção de Características. As linhas representam a média dos resultados da literatura (PASQUINI; STADLER, 2017), a contínua sem utilização de características e a tracejada com utilização da técnica. Fonte: Elaborado pelo autor. . . . .	43
Figura 12 – <i>Boxplot</i> da NMAE para cada experimento com apenas as 7200 amostras disponíveis para o pré-processamento. A linha contínua representa o resultado médio da literatura sem utilização de seleção de características (PASQUINI; STADLER, 2017). Fonte: Elaborado pelo autor. . .	45
Figura 13 – Diferença da NMAE obtida por todos experimentos sem utilização de seleção de características utilizando o método de validação cruzada <i>K-Fold</i> e <i>train_test_split</i> . A linha contínua referencia a média da NMAE obtida pela literatura (PASQUINI; STADLER, 2017) também sem utilização de seleção de características. Fonte: Elaborado pelo autor.	46
Figura 14 – Comparação das densidades dos conjuntos $Y'$ em um mesmo grau de raleamento para as partições de teste e treinamento com a utilização do <i>train_test_split</i> e <i>K-Fold</i> ( <i>fold</i> arbitrário) para o conjunto sem pré-processamento e com $p = 128$ . Eixo Y em escala logarítmica para ressaltar as variações de densidade. Fonte: Elaborado pelo autor. . . . .	47

# Lista de tabelas

Tabela 1 – Experimento 1: Tabela de NMAE para diferentes cargas, aplicações e métodos para cada período de raleamento. $p = 1$ é o valor referência da literatura (PASQUINI; STADLER, 2017). RT representa o algoritmo <i>Regression Tree</i> e RF o <i>Random Forest</i> . Fonte: elaborado pelo autor.	35
Tabela 2 – Experimento 1 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada período de raleamento. $p = 1$ é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: elaborado pelo autor.	36
Tabela 3 – Experimento 2 com função média: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas. $j = 1$ é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: elaborado pelo autor.	37
Tabela 4 – Experimento 2 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas com utilizando função média quando submetido à técnica de redução de dimensionalidade <i>Univariate Feature Selection</i> . $j = 1$ é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: elaborado pelo autor.	38
Tabela 5 – NMAE média e intervalo de confiança (teste $t$ de Student) em valores porcentuais para o Experimento 2, comparando a função utilizada no pré-processamento e quanto à utilização de seleção de características. $j = 1$ é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: Elaborado pelo Autor.	39
Tabela 6 – Experimento 3 sem Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas. A primeira coluna da NMAE é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: elaborado pelo autor.	41
Tabela 7 – Experimento 3 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas. A primeira coluna da NMAE é o valor referência da literatura (PASQUINI; STADLER, 2017). Fonte: elaborado pelo autor.	42
Tabela 8 – Resultados da NMAE (valores percentuais) com intervalos de confiança calculados via teste $t$ de Student para os experimentos 1 a 3, comparando a utilização de seleção de características. Fonte: Elaborado pelo Autor.	44



# **Lista de abreviaturas e siglas**

QoS	<i>Quality of Service</i> (Qualidade de Serviço)
QoE	<i>Quality of Experience</i> (Qualidade da Experiência)
VoD	<i>Video-over-Demand</i> (referência à conjunto de dados da literatura)
RT	<i>Regression Tree</i> (Árvore de Regressão)
RF	<i>Random Forest</i> (Floresta Aleatória)
ML	<i>Machine Learning</i> (Aprendizado de Máquina)
SLA	<i>Service Level Agreement</i> (Acordo de Nível de Serviço)
NMAE	<i>Normalized Mean Absolute Error</i> (Erro Médio Absoluto Normalizado)
FS	<i>Feature Selection</i> (Seleção de Características)
UFS	<i>Univariate Feature Selection</i>

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
<b>1.1</b>	<b>Objetivos</b>	<b>12</b>
<b>1.2</b>	<b>Estrutura do Trabalho</b>	<b>13</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>14</b>
<b>2.1</b>	<b>Fundamentação Teórica</b>	<b>14</b>
2.1.1	Aprendizado de Máquina	14
2.1.2	Monitoramento	17
2.1.3	Aplicações de <i>Streaming</i>	19
2.1.4	Engenharia de Características	21
2.1.4.1	Redução de Dimensionalidade	22
<b>2.2</b>	<b>Trabalhos Relacionados</b>	<b>25</b>
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>27</b>
<b>3.1</b>	<b>Experimento 1: Eliminação de Amostras</b>	<b>27</b>
<b>3.2</b>	<b>Experimento 2: Janelas</b>	<b>29</b>
<b>3.3</b>	<b>Experimento 3: Janelas da Distribuição das Métricas</b>	<b>31</b>
<b>3.4</b>	<b>Experimento 4: Diminuição do Tempo de Análise</b>	<b>31</b>
<b>4</b>	<b>RESULTADOS</b>	<b>34</b>
<b>4.1</b>	<b>Experimento 1: Eliminação de Amostras</b>	<b>34</b>
<b>4.2</b>	<b>Experimento 2: Janelas</b>	<b>37</b>
<b>4.3</b>	<b>Experimento 3: Janelas da Distribuição das Métricas</b>	<b>41</b>
<b>4.4</b>	<b>Experimento 4: o Impacto da Quantidade de Amostras</b>	<b>43</b>
<b>4.5</b>	<b>Avaliação das Técnicas de Particionamento</b>	<b>45</b>
<b>5</b>	<b>CONCLUSÃO</b>	<b>48</b>
	<b>REFERÊNCIAS</b>	<b>51</b>

# 1 Introdução

Com o surgimento de tecnologias de transmissão de vídeo avançadas, grandes empresas começaram a oferecer serviços de *streaming* para os usuários finais. No entanto, para garantir uma experiência de visualização de vídeo sob demanda de alta qualidade, é necessário ter uma conexão estável e de alta velocidade. Qualquer deterioração na qualidade de serviço (QoS, do inglês “*Quality of Service*”) pode resultar em interrupções, baixa qualidade de imagem e problemas de sincronização entre áudio e vídeo, o que é imediatamente perceptível pelos usuários finais. Diante disso, para atender às exigências crescentes dos clientes, é fundamental desenvolver um sistema robusto e capaz de fornecer um serviço de alta qualidade a todos os clientes finais, pois os clientes insatisfeitos tendem a buscar alternativas oferecidas por concorrentes no mercado.

Monitorar e orquestrar sistemas são duas áreas amplamente pesquisadas pela comunidade científica de redes. Um bom sistema de monitoramento compreende a utilização de ferramentas capazes de lidar com o volume de dados de monitoramento, ao passo que um bom orquestrador visa garantir a QoS acordada pelo Acordo em Nível de Serviço (SLA, do inglês “*Service-level Agreement*”) por meio, por exemplo, da atuação de pessoas especialistas ou da adoção de mecanismos autônomos ([SALEHIE; TAHVILDARI, 2009](#); [KEPHART; CHESS, 2003](#)) que consideram políticas pré-estabelecidas de operação ou limites (*thresholds*) de métricas de operação. Mais recentemente, diversas pesquisas consideram sistemas de monitoramento e orquestração atuando com técnicas de inteligência artificial ([SALEHIE; TAHVILDARI, 2009](#)).

Neste contexto, considerando as fortes exigências atuais em termos de QoS e, também, de qualidade de experiência (QoE, em inglês “*Quality of Experience*”), monitoramento e orquestração precisam operar em intervalos de tempo cada vez menores, com o intuito de sanar rapidamente possíveis degradações no funcionamento das aplicações.

A utilização de vídeo sob demanda cresceu significativamente nas últimas décadas e existem diversas plataformas que fornecem este serviço, como a *Netflix* e *YouTube*. Ainda em 2011, a *Netflix* respondia por cerca de 30% de todo o tráfego de rede norte-americano ([KUROSE, 2013](#)). Para auxiliar no atendimento dessa elevada demanda, técnicas que elevam a QoS e a QoE foram implementadas, dentre as quais se destaca o DASH (*Dynamic Adaptive Streaming over HTTP*), que permite que os parâmetros da transmissão sejam negociados com o servidor de forma à disponibilizar diferentes qualidades de vídeo, implementação de *buffers* que suportam a instabilidade da conexão, dentre outros mecanismos.

Este trabalho avalia a sensibilidade dos modelos de aprendizado de máquina em

relação ao volume, periodicidade e qualidade dos dados utilizados durante a fase de treinamento. Em linhas gerais, sistemas de monitoramento, também conhecidos como sistemas de telemetria, são construídos para observar o funcionamento das aplicações por meio da coleta de estatísticas, desde a infraestrutura à aplicação em execução, que são fornecidas aos modelos de aprendizado de máquina, os quais emitem estimativas da QoS oferecida.

Embora o monitoramento seja indispensável neste contexto, ele implica em custos, principalmente representados pela coleta e transporte das métricas através da infraestrutura, o que aumenta o consumo da capacidade de processamento, armazenamento, memória, rede, dentre outros. Dessa forma, monitorar um sistema tem, inclusive, potencial para influenciar negativamente o funcionamento das aplicações, na medida em que pode concorrer por recursos computacionais com o próprio serviço oferecido aos clientes.

Para o desenvolvimento deste trabalho, utilizou-se um conjunto de dados publicamente disponível ([PASQUINI, 2017](#)), coletado em uma estrutura com 9 servidores, uma rede *OpenFlow* e um serviço de vídeo sob demanda. Os dados foram coletados com periodicidade de monitoramento em intervalos de um segundo. Os experimentos utilizaram um gerador de carga que fornecia dois padrões de carga, um periódico, que emula uma senoide com comportamento consistente, e outro que emula demandas repentinhas (denominado *Flash-Crowd*). O conjunto de dados possui aproximadamente 10.500 métricas relativas ao funcionamento da infraestrutura e 17 métricas relativas ao comportamento da aplicação de vídeo sob demanda, com destaque especial ao número de quadros por segundo recebido pelos clientes, que representa a qualidade do vídeo, tanto em momentos de baixa, quanto em momentos de alta carga na aplicação.

Partiu-se do conjunto de dados ([PASQUINI, 2017](#)) e pelos resultados de modelos baseados em Árvore de Regressão ([JAMES et al., 2023](#)) e Floresta Aleatória ([BREIMAN, 2001](#)) treinados por esses dados em [Pasquini e Stadler \(2017\)](#), e, então, três técnicas são investigadas neste trabalho:

1. Eliminação de amostras do conjunto de dados em intervalos de  $p$  segundos, com  $p$  variando exponencialmente entre 2 e 256 segundos;
2. Análise de janelas de dados em intervalos de  $j$  segundos, com  $j$  variando exponencialmente entre 2 e 256 segundos, aplicando funções tradicionais de mínimo, máximo e média;
3. Engenharia de características com as informações da distribuição da métrica em intervalos de  $d$  segundos, variando exponencialmente de 8 a 256 segundos, considerando as seguintes estatísticas: média, desvio padrão, mediana, assimetria, curtose, percentil 25, percentil 75 e amplitude.

Como forma de complementar as análises e enriquecer as investigações realizadas, este trabalho aplica o conceito de Seleção de Características (JAMES et al., 2023) existente no contexto de Aprendizado de Máquina. Simplificadamente, essa técnica analisa o conjunto de dados e identifica um subconjunto de características mais representativas para o treinamento de modelos. Segundo Shahabsamani (2024), essa abordagem melhora a qualidade das estimativas ao aprimorar a capacidade de generalização dos modelos.

Dentre os principais resultados deste trabalho, identifica-se que a escolha correta do intervalo de amostragem e a forma como as características são apresentadas pelo sistema de monitoramento são relevantes para uma boa estimativa da QoS nos clientes para os sistemas de orquestração baseados em Inteligência Artificial. A eliminação de amostras, que simula a redução da frequência de coleta, alcançou resultados em que períodos de até 16 segundos eram viáveis sem seleção de características ou acima de 128 com a utilização da técnica, que equilibra a quantidade de características para o tamanho da amostragem. Com a agregação em janelas, porém, o tamanho total do conjunto de dados pôde ser mais reduzido, embora ainda alcançasse uma barreira aos 64 segundos de intervalos de dados. Finalmente, utilizar várias métricas descritivas manteve uniforme as previsões para uma janela de até 128 segundos quando utilizada a seleção de características, porém o aumento da dimensionalidade impactou os resultados sem a utilização da técnica já aos 32 segundos.

## 1.1 Objetivos

Este trabalho tem como objetivo geral a análise de técnicas de redução de amostras em sistemas de monitoramento, de forma que as decisões realizadas por sistemas de orquestração dependam de menos recursos computacionais da infraestrutura e mantenham o mesmo nível de precisão já encontrado pela literatura (PASQUINI; STADLER, 2017). E tem como objetivos específicos:

- Replicar parâmetros e resultados já obtidos pela literatura (PASQUINI; STADLER, 2017) para os algoritmos de Árvore de Regressão e Floresta Aleatória;
- Avaliar as três técnicas de pré-processamento (remoção de amostras, agregação em janelas com funções ou descrição estatística) que poderiam ser adotadas por um sistema de monitoramento, as quais são capazes de reduzir a quantidade de amostras ou características necessárias para fornecer boas estimativas para sistemas de orquestração (SHAHABSAMANI, 2024; BAIG et al., 2020);
- Identificar parâmetros de coleta adequados e apresentar expectativas do comportamento dos modelos em diferentes cenários;

- Avaliar como a relação da dimensionalidade de métricas e a quantidade de amostras impactam as estimativas dos modelos;
- Disponibilizar código-fonte aberto ([VILARINHO, 2025](#)) para trabalhos futuros, replicabilidade dos experimentos e adaptação em variados contextos de monitoramento.

## 1.2 Estrutura do Trabalho

O restante deste trabalho está organizado da seguinte forma: o Capítulo 2 introduz os principais conceitos envolvidos nesta pesquisa, como aprendizado de máquina, monitoramento, QoS em transmissão de vídeo e engenharia de características, assim como apresenta trabalhos relacionados; o Capítulo 3 detalha as três técnicas propostas para reduzir o período de monitoramento: a eliminação das amostras, agregação em janelas e a utilização de métricas descritivas para cada métrica; o Capítulo 4 apresenta os resultados e discute cada um dos experimentos propostos; e, finalmente, o Capítulo 5 conclui este trabalho e propõe alguns trabalhos futuros.

## 2 Revisão Bibliográfica

Este capítulo apresenta a fundamentação teórica e os trabalhos relacionados que embasam este estudo. A Seção 2.1 introduz os conceitos e como eles foram adotados por este trabalho, tais como Aprendizado de Máquina, monitoramento, *Streaming* e seleção de características. A Seção 2.2 apresenta os trabalhos relacionados que foram utilizados como ponto de partida para este trabalho.

### 2.1 Fundamentação Teórica

Nesta seção, o tema deste trabalho é dividido nos principais conceitos relacionados ao aprendizado de máquina, QoS em aplicações de vídeo, monitoramento e engenharia de características, com o objetivo de fundamentar a metodologia em elementos matemáticos e estatísticos, mas também em elementos pragmáticos do consumo e desempenho computacional de sistemas de transmissão de vídeo.

Inicia-se com a conceitualização de Aprendizado de Máquina e como a literatura embasa os parâmetros e técnica utilizados por este trabalho. O papel do monitoramento é abordado como parte do processo de orquestração e também descreve o conjunto de dados utilizado neste trabalho. Adiante, é contextualizado como as técnicas descritas se aplicam ao cenário de *streaming* e as melhores formas de abordar o problema com monitoramento e técnicas de Aprendizado de Máquina, bem como a natureza do problema quanto aos dados encontrados. Finaliza-se a primeira seção com técnicas de engenharia de características disponíveis e utilizadas neste trabalho, bem como o embasamento teórico e justificativa das escolhas.

#### 2.1.1 Aprendizado de Máquina

De maneira geral, em [Mello e Ponti \(2018\)](#) o Aprendizado de Máquina é apresentado como uma área que estuda como um computador pode aprender a dar boas respostas, sejam elas classificações ou estimativas numéricas. Uma técnica capaz de aprender é aquela capaz de decidir valores ou classificações mais precisas com base em experiências passadas.

Por outro lado, [James et al. \(2023\)](#) comprehende o Aprendizado de Máquina em uma área mais ampla e associada à estatística: o *Statistical Learning*, traduzido do inglês como Aprendizado Estatístico. Esta área consiste no estudo das técnicas que visam fornecer uma estimativa  $\hat{Y}$  a partir de um conjunto de entrada  $X$  de forma a diminuir o erro aos valores reais  $Y$ , à medida que mais amostras de  $X$  são apresentadas ao modelo.

Ao assumir que  $Y = f(X)$  para alguma função  $f$ , o modelo cria uma estimativa  $\hat{Y}$  para uma função estimada  $\hat{f}$  e dados de entrada  $X$  definida por  $\hat{Y} = \hat{f}(X)$ . Existe também um erro  $\epsilon$  que representa a diferença entre o valor estimado e o valor real  $Y = \hat{f}(X) + \epsilon$ . O erro  $\epsilon$  pode reduzir na medida em que a função estimada  $\hat{f}$  é mais adequada, mas é irredutível a partir do momento em que mais variáveis podem ser necessárias para explicar a variabilidade de  $Y$  de forma generalizada. Portanto, o objetivo do Aprendizado Estatístico é encontrar a função  $\hat{f}$  que minimize o erro  $\epsilon$  ao valor irredutível.

[James et al. \(2023\)](#) ainda descreve que, para estimar a função  $\hat{f}$ , utiliza-se um subconjunto de amostras  $\{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$  disponíveis para treinamento (o  $X_t$ ). Dessa forma, é possível utilizar uma divisão do conjunto de dados de tal forma que parte dele seja destinada para melhorar a estimativa de  $\hat{f}$ , o conjunto de treinamento, e outra parte para teste e validação da função encontrada.

Encontrar e utilizar as partições destinadas para teste e treinamento dependem da metodologia escolhida. [James et al. \(2023\)](#) descreve duas técnicas de particionamento denominadas “Conjunto de Validação” e a “Validação Cruzada K-Fold”. Para a primeira, ocorre a divisão de todo o conjunto de dados em conjuntos disjuntos  $(X, Y)_{\text{treinamento}}$  e  $(X, Y)_{\text{validação}}$  de tamanhos complementares, ou seja, juntos formam todo o conjunto  $X$ . De forma similar, a segunda divide em  $k$  partições, entretanto, a  $k$ -ésima partição é destinada para validação e o restante é utilizada para treinamento, e o erro é computado pela média dos  $k$  erros encontrados. Na biblioteca *scikit learn*, a primeira técnica é implementada pela função `train_test_split` e a segunda com a classe `KFold` ([PEDREGOSA et al., 2011](#)).

Em séries temporais, entretanto, ambos mecanismos não consideram a dependência temporal no particionamento. Existem técnicas propostas por [James et al. \(2023\)](#) como considerar a ordem temporal nas partições, Janelas Deslizantes ou *Backtesting* (que considera dados históricos em diferentes pontos). Entretanto, [Pasquini e Stadler \(2017\)](#) não utilizam estas técnicas, uma vez que os conjuntos de dados utilizados ressaltam a diferença de carga geradas ao longo do tempo de execução, o que justifica uma divisão não com um viés temporal, e sim operacional.

Em [James et al. \(2023\)](#) também é definido que o aprendizado pode ser:

- Supervisionado, Não-Supervisionado ou Semi-Supervisionado: quando para qualquer dado de entrada disponível existe também a respectiva saída (rótulo) conhecida, enquanto não-supervisionado não há saída (rótulo) correspondente; finalmente o semi-supervisionado combina ambos, onde parte dos dados são rotulados e uma quantidade não é;
- Regressão ou Classificação: saída numérica e agrupamento de diferentes amostras em categorias, respectivamente. O aprendizado supervisionado é adequado para ambos

casos, mas não é possível aplicar aprendizado Não-Supervisionado para problemas de regressão.

Por fim, a avaliação dos resultados obtidos depende do erro  $\epsilon$  e pode ser mensurado de diversas maneiras diferentes a depender da natureza do problema. Como o objetivo é entender a diferença média de  $\hat{Y}$  com o próprio  $Y$ , as propostas mais simples são o somatório das diferenças absolutas ou quadráticas, erros conhecidos respectivamente como Absoluto Médio (*MAE* abreviação do inglês para *Mean Absolute Error*) e Quadrático Absoluto (*MSE* abreviação do inglês para *Mean Squared Error*) (JAMES et al., 2023). O primeiro é mais adequado para reduzir o impacto da diferença de *outliers*, enquanto o segundo é otimizado para métodos que precisem de cálculos de derivadas pela possibilidade de diferenciar a função (quadrática), como Redes Neurais.

Além do MAE e MSE, outras medidas também são comumente utilizadas para avaliar modelos de Aprendizado de Máquina em problemas de regressão. Um deles é Raiz do Erro Quadrático Médio (*RMSE*, do inglês *Root Mean Squared Error*) que deriva da MSE, mas mantém as mesmas unidades da variável, o que facilita interpretação de resultados (MITCHELL, 1997; MELLO; PONTI, 2018). Há também o Erro Percentual Absoluto Médio (*MAPE*, do inglês *Mean Absolute Percentage Error*), que retorna o erro em termos percentuais aos originais, embora seja limitado para valores pequenos (LACSON et al., 2022). Já o coeficiente de determinação, o  $R^2$ , mede a variância explicada pelo modelo (BREIMAN, 2001). Finalmente, Pasquini e Stadler (2017) utilizam o Erro Absoluto Médio Normalizado (*NMAE*, do inglês *Normalized Absolute Error*), que normaliza o MAE e permite uma comparação mais justa em conjuntos de dados distintos ou modelos com escalas distintas (BREIMAN, 2001), dada pela equação:  $NMAE = \frac{1}{\bar{y}} \left( \frac{1}{m} \sum_{i=1}^m |y_i - \hat{y}_i| \right)$ , onde  $\hat{y}$  é o valor estimado,  $y_i$  o valor original e  $\bar{y}$  a média das  $m$  amostras de teste.

Dessa forma, é possível observar que o Aprendizado de Máquina evolui da utilização de métodos estatísticos tradicionais baseados na regressão e ajustes de parâmetros, e abre margem para uma gama de métodos diferentes e, especificamente neste trabalho, destaca-se aqueles baseados em árvores. Uma Árvore de Regressão possui uma série de nós, em que cada um deles possui uma regra de avaliação do valor de entrada (MITCHELL, 1997). Uma árvore sozinha é estatisticamente capaz de oferecer boas estimativas, mas Breiman (2001) demonstra a possibilidade de utilizar não só uma árvore, mas uma coleção delas, para obter uma estimativa por meio de votação após a criação de uma série de árvores, este é o método denominado *Random Forest* (traduzido como Floresta Aleatória).

Árvores de Regressão recursivamente particionam as amostras de treinamento em subconjuntos mais homogêneos e com menor variância dentro de cada partição (JAMES et al., 2023). A escolha da partição de uma amostra é resultado de uma condição binária

(BREIMAN, 2001) que decide se uma amostra está associada à subárvore esquerda ou direita. Enquanto isso, o ponto de corte tipicamente baseia-se na escolha de um ponto em que a subárvore minimize o MSE de ambas partições definida por  $MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$  (JAMES et al., 2023). Cada etapa encontra diferentes limiares até que atinjam critérios de parada, normalmente baseados na mínima quantidade de amostras por nó ou profundidade máxima (PEDREGOSA et al., 2011). Finalmente, os pontos escolhidos e as condições formam os conjunto de regras que são os parâmetros dos modelos baseados no método.

O Floresta Aleatória combina diferentes árvores de regressão treinadas com um subconjunto aleatório de amostras e também das características, enquanto a previsão é dada pela média dos resultados de cada árvore, o que melhora a robustez do modelo e diminui o *overfitting* (BREIMAN, 2001; JAMES et al., 2023). Esses dois processos de aleatorização são respectivamente conhecidos como Amostragem com Reposição (em inglês denominado *Bootstrap*) e Subamostragem de Atributos (em inglês denominado *Feature Bagging*) e permitem a sobreposição de amostras ou características escolhidas em outros subconjuntos. James et al. (2023) destaca que este método apresenta menor variância em comparação ao uso de uma única árvore, principalmente quando lidam com conjuntos de dados altamente dimensionais.

Pasquini e Stadler (2017) utilizam das Árvores de Regressão e da Floresta Aleatória com 120 árvores para prever a quantidade de quadros por segundo em aplicações de *streaming*, utilizando o método do Conjunto de Validação com 70% do conjunto de dados destinado para treinamento e 30% para teste. Este trabalho, investiga as três técnicas propostas de monitoramento utilizando as mesmas parametrizações e técnicas de Aprendizado de Máquina, porém com a utilização da Validação Cruzada com  $k = 3$ , com o objetivo de diminuir o viés que o particionamento mais simples poderia causar com menos amostras de  $X$ . A opção por  $k = 3$  visa manter um conjunto de testes de tamanho similar ao original (um terço do conjunto de dados). Assim é possível comparar a NMAE obtida em cada experimento com aqueles obtidos por Pasquini e Stadler (2017).

### 2.1.2 Monitoramento

O monitoramento é a ferramenta a partir da qual o orquestrador pode tomar decisões importantes durante a execução de um sistema. Kephart e Chess (2003) descreve um ciclo que comprehende monitorar o sistema, analisar seu estado atual, planejar ações corretivas e executar estas ações, para garantir estabilidade e qualidade adequadas ao sistema, denominado *MAPE loop*. Dessa forma, é necessário que um sistema tenha planos sólidos de resposta às mais variadas cargas e condições de operação (SALEHIE; TAH-VILDARI, 2009). Esses planos e as ações adotadas só são descobertas na medida em que a infraestrutura é analisada pelas métricas coletadas.

Um sistema que implementa este ciclo *MAPE* deveria ser capaz de coletar os

dados necessários na fase de monitoramento, de forma que técnicas utilizadas na fase de análise possam identificar as causas que dificultam o funcionamento do sistema, para que planos/ações a fossem escolhidos(as) e executados(as) adequadamente.

O objetivo de um sistema de monitoramento é atingir o que Kephart e Chess (2003) conceitua como *computação autônoma*, em que sistemas são independentes e capazes de lidar com a própria configuração, otimização, recuperação de erros e proteção contra ataques (SALEHIE; TAHVILDARI, 2009).

Ahmed et al. (2020) descreve que apenas coletar as métricas e apresentar em painéis, mas manter o processo de decisão humano e manual é um processo entediante e com alta chance de erro, e que utilizar métodos automáticos de detecção de problemas na QoE é fundamental para diminuir o tempo inativo dos sistemas. Uma alternativa automatizada é justamente a implementação de sistemas de orquestração baseados em Aprendizado de Máquina.

Dentre os programas para coletar as métricas da infraestrutura, o SAR permite analisar estruturas de dados do *kernel* do sistema operacional (GODARD, 2007), onde estão disponíveis milhares de métricas como a utilização de CPU, temperatura, pacotes de rede trafegados, entre outras. Pasquini e Stadler (2017) também utilizou de métricas de rede provenientes e próprias do protocolo *OpenFlow*. O SAR consegue ser parametrizável quanto à métricas, intervalo e contagem das amostras escolhidas, mas não inferior à um segundo.

O pacote que contém o comando SAR é o Sysstat... (2023) e contém também outras ferramentas que permitem monitorar uma ampla gama de métricas para análise da performance de sistemas, como o iostat (IOSTAT(1)..., 2023), o qual fornece estatísticas do uso de CPU e operações de entrada e saída realizadas pelo sistema nos dispositivos montados.

Há diferentes tipos de métricas em monitoramento de sistemas. Algumas são absolutas, como `tcpsock`, que representa a quantidade absoluta de *sockets* TCP em uso; outras são percentuais, como `%memused`, que indica a porcentagem de memória utilizada. Existem também métricas de fluxo, como `packet/s`, que mede o número de pacotes recebidos por segundo, ou métricas baseadas em funções, como `ldavg-1`, que descreve a carga do sistema no último minuto. Um ponto importante é que o uso de métricas de fluxo ou funções em diferentes intervalos de tempo não altera sua relação com a unidade de tempo original — por exemplo, `packet/s` continuará se referindo a pacotes por segundo. O SAR não abrange sistemas externos e é indicada como uma das primeiras ferramentas para a análise da atividade de um sistema.

Pasquini e Stadler (2017) avalia a utilização de métricas da infraestrutura para prever o QoS em serviços de banco de dados e *Streaming*. Rezende (2020) propõe um

orquestrador que implementa o *MAPE loop* ao utilizar o programa *Grafana* e *Prometheus* para coleta e apresentação de métricas dos recursos alocados, o *Influx DB* para armazenar as séries temporais com granularidade por segundo e o *Kafka* para manter a comunicação entre as unidades de origem das estatísticas e as consumidoras. Os dados são posteriormente analisados com os modelos e Seleção de Características de forma a reduzir a carga do sistema de monitoramento. Os planos são executados de acordo com a demanda dos recursos, a fim de atender as necessidades de elasticidade do sistema.

Os experimentos de [Pasquini \(2017\)](#) coletaram dados com granularidade de 1 segundo com períodos de aproximadamente 10 horas por experimento, o que resultou em 8 conjuntos de dados baseados nas seguintes condições: variações na geração de carga adotando-se os padrões periódico ou repentino (em inglês “*Periodic*” e “*Flashcrowd*”), variações nas aplicações em execução, a aplicação observada (*streaming VLC* e banco de dados *Voldemort*) e se haviam ambas aplicações ou apenas a aplicação observada em execução. No contexto deste trabalho, apenas os quatro conjuntos de dados cuja aplicação observada é o *streaming* serão analisados.

O conjunto *X* é formado por mais de 10.500 métricas da infraestrutura, indexado pelo dispositivo em que foi coletado, sendo as métricas coletadas subdivididas em 3 categorias: as coletadas pelo SAR ([GODARD, 2007](#)) do sistema operacional; as estatísticas relativas aos fluxos *OpenFlow* de dados; e as estatísticas *OpenFlow* coletadas por porta de cada *switch* presente na rede.

O conjunto *Y* é composto por 17 métricas de QoS do reprodutor de vídeo VLC de um cliente durante a transmissão. Dentre elas destaca-se a métrica denominada “*DispFrames*”, que representa o número de quadros por segundo entregue aos clientes, sendo esta a principal estatística *Y* estimada pelos experimentos deste trabalho. A Figura 1 apresenta a associação das amostras de *X* com as amostras de *Y*, realizada a partir do *timestamp* de coleta das amostras de dados, em intervalos de um segundo.

Cada um dos quatro conjuntos de dados possui, em média, 700MB, e mais de 37.000 amostras, coletadas em intervalos de um segundo. No contexto deste trabalho, são avaliadas técnicas que podem aumentar da granularidade dos dados, ou seja, a elevação dos períodos de monitoramento ou reportagem e, consequentemente, resultando na diminuição do tamanho do conjunto de dados ou na diminuição do impacto do monitoramento no sistema.

### 2.1.3 Aplicações de *Streaming*

Por aplicações de *streaming* comprehende-se a transmissão de dados em tempo real para um cliente de forma contínua. O VLC é uma aplicação de reprodução de vídeo ([VIDEOLAN, 2024](#)) e suporta DASH, uma técnica de transmissão de vídeo que permite

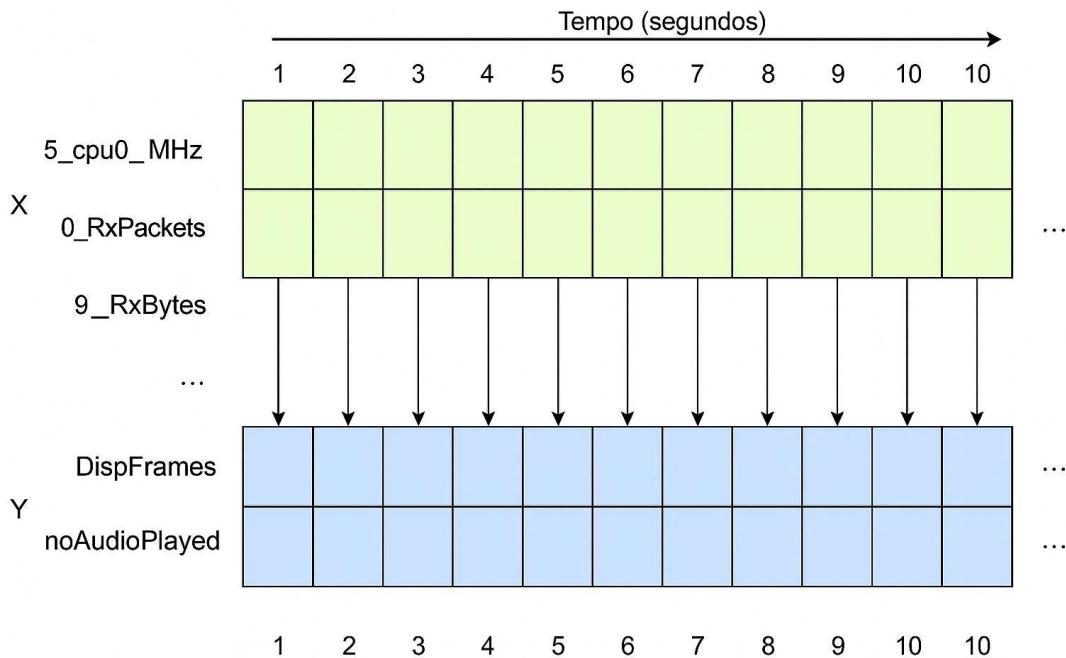


Figura 1 – Associação entre as amostras do conjunto de dados  $X$  e  $Y$  original com exemplos de métricas reais coletadas. As amostras de  $X$  e  $Y$  são associadas pelo *timestamp* de coleta, considerando dez dígitos do *Unix Timestamp* (escala de segundos). Fonte: Elaborado pelo autor.

a adaptação da qualidade da mídia de acordo com a largura de banda disponível ([LA-CERDA, 2020](#)).

O DASH é uma coleção de soluções que provê transmissão de vídeo sobre HTTP ([MPEG, 2022](#)). [Sodagar \(2011\)](#) descreve que o protocolo surgiu para contornar bloqueios de *firewalls* de pacotes UDP, que são associados com *streaming*, bem como dá o controle ao cliente do estado da reprodução da mídia, ao contrário do RTP (do inglês, *Real-time Transport Protocol*), em que o servidor é responsável por isso ([KUROSE, 2013](#)). O DASH opera na requisição de um manifesto da mídia pelo cliente, nele possui informações da estrutura do vídeo e dos parâmetros de transmissão. Em seguida, o cliente pode solicitar os quadros correspondentes aos segmentos via requests GET HTTP, na medida em que consome ou cria *buffers* de forma a controlar a variação da latência na rede.

Dessa forma é possível assumir que a QoS em aplicações de *streaming* é muito atrelada à métrica de quadros por segundo FPS (do inglês, “frames per second”) e bons resultados foram encontrados pela estimativa partindo de dados da infraestrutura como [Pasquini e Stadler \(2017\)](#), [Yanggratoke et al. \(2015\)](#) e [Matos \(2018\)](#) alcançaram. Outra métrica que foi estimada foi a quantidade de *buffers* de áudio, mas que não obtiveram resultados tão precisos quanto à métrica de FPS.

A métrica coletada de quadros por segundo apresenta comportamento bimodal

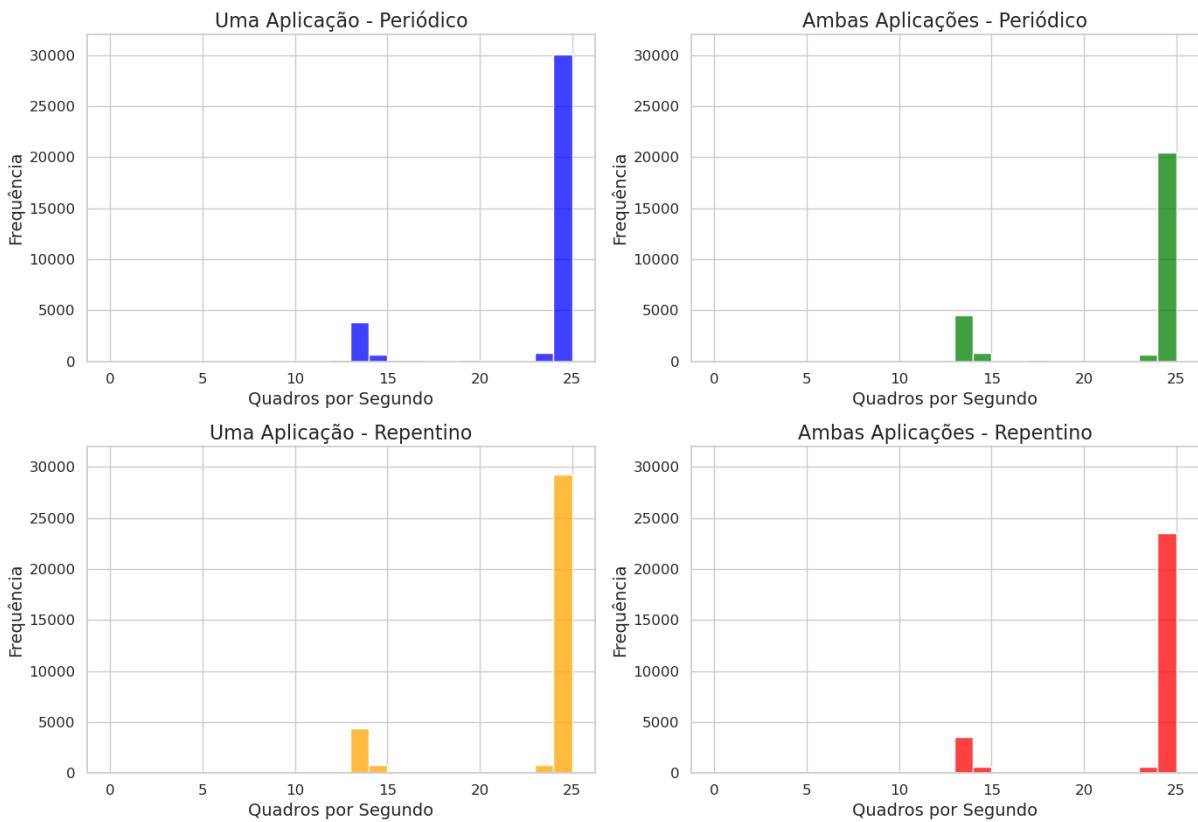


Figura 2 – Comparação dos histogramas da métrica de Quadros por Segundo de cada conjunto de dados. A denominação “Uma Aplicação” refere-se ao conjunto de dados coletado em experimentos executando apenas a aplicação de vídeo sob demanda. A denominação “Ambas Aplicações” refere-se ao conjunto de dados coletado em experimentos executando ambas as aplicações em paralelo, vídeo sob demanda e banco de dados distribuído (*Voldemort*). Periódico e Repentino referem-se ao padrão do gerador de carga. Fonte: Elaborado pelo autor.

(PASQUINI; STADLER, 2017; YANGGRATOKE et al., 2015), onde a maior parte das amostras são próximas à 24 e 13 quadros por segundo. A Figura 2 apresenta a distribuição da frequência desta métrica nos quatro conjuntos de dados de Pasquini (2017) por meio de histogramas. A maioria das amostras é maior ou igual à 24 quadros por segundo, as quais compõem uma boa QoE. As amostras abaixo de 15 quadros compõe uma baixa QoE e representam em média aproximadamente 21% dos conjuntos.

#### 2.1.4 Engenharia de Características

A Engenharia de Características comprehende um passo na fase de pré-processamento capaz de alterar os valores de características específicas. Dessa forma, o objetivo deste passo é que os modelos sejam robustos e utilizem métricas que mantenham a relevância mesmo com mudanças na infraestrutura e contexto.

Existem casos onde a engenharia de características é essencial para evitar *overfitting*, e o principal exemplo são conjuntos de dados com alta-dimensionalidade, ou seja,

possuem mais características que amostras ([JAMES et al., 2023](#)). O problema, nesse cenário, é que pode existir uma grande quantidade de amostras esparsas dessas características adicionais e isso pode diminuir o erro na função de custo e dificultar o ajuste de parâmetros.

De acordo com [Hastie, Friedman e Tibshirani \(2001\)](#), o pré-processamento é uma etapa crucial na análise de qualquer conjunto de dados, seja por uma questão de desempenho, seja pela natureza matemática das amostras. Apesar das alternativas *Deep Learning*, que podem ter funções com resultados semelhantes em seus nós, ainda hoje são usadas transformadas conhecidas como entrada de redes neurais, como a *Wavelet* em imagens.

Dentre as técnicas de pré-processamento, as janelas deslizantes usadas em séries temporais condensam subintervalos ([BAIG et al., 2020](#)), com objetivo de ressaltar padrões em intervalos das séries ao invés de métrica à métrica. A ideia é realizar uma operação sobre um valor constante de  $j$  amostras passadas e saltar  $s$  amostras para o cálculo da próxima janela, dessa forma é possível que duas janelas sejam completamente disjuntas ou tenham ainda elementos de outras. As janelas não reduzem a quantidade de dimensões, mas permite agregar amostras e podem diminuir a quantidade total de amostras à depender da parametrização de  $s$ .

No contexto deste trabalho, as janelas são utilizadas como mecanismo de agregar amostras em dois experimentos propostos com objetivo de reduzir a quantidade de amostras do conjunto de dados quando o tamanho da janela equivale ao salto ( $j = s$ ). Em um dos experimentos, as janelas são utilizadas de forma à transformar cada métrica quanto ao resultado da aplicação de uma função  $f$  nas últimas  $j$  amostras, sendo  $f$  média, máximo e mínimo. Em outro, cada métrica retorna um conjunto de 8 estatísticas que descrevem o comportamento da métrica nos últimos  $j$  segundos. Ambos estão detalhados no Capítulo 3, para experimentos denominados 2 e 3, respectivamente.

Finalmente, este trabalho parte de 4 conjuntos de dados existentes ([PASQUINI, 2017](#)) no contexto de Aplicações de *Streaming* e propõe técnicas de pré-processamento com objetivo reduzir o tempo de coleta ou o de reportagem das métricas, minimizando o consumo de recursos pelo sistema de monitoramento.

#### 2.1.4.1 Redução de Dimensionalidade

Dentre os desafios da engenharia de características, destaca-se oferecer métricas cujas amostras sejam significativas e ressaltem padrões sem sub-representar comportamentos essenciais durante o monitoramento. Dessa forma além da qualidade, a quantidade de métricas precisa ser balanceada de forma que não suprima padrões incomuns, mas mantenham os relevantes.

A dimensionalidade é um dos desafios na redução do tamanho total de um conjunto

de dados. Ela se refere ao número de características  $X$  de um conjunto de dados. De acordo com [Domingos \(2012\)](#), conjuntos de dados com alta dimensionalidade acarretam, além do aumento no tempo de processamento, na dificuldade da visualização dos dados e degradação dos modelos de aprendizado de máquina, efeito conhecido como “maldição da dimensionalidade”. Nesse quesito, destacam-se as técnicas de redução de dimensionalidade a análise de componentes principais (PCA - do inglês, *Principal Component Analysis*), *autoencoders* e com destaque neste trabalho a seleção de atributos.

A Análise de Componentes Principais (PCA) é uma técnica estatística que encontra como a variabilidade de cada variável relaciona com a variável explicada. Ela encontra um subconjunto de novas variáveis, denominadas componentes principais, que são combinações lineares das variáveis originais ([JAMES et al., 2023](#)).

Os *autoencoders* formam camadas de redes neurais artificiais que reduzem variáveis à representações intermediárias, condensando a informação de várias características e utilizando o próprio treinamento da rede para representá-las. A ideia é que na rede existem duas partes na utilização dele: o codificador, que reduz a dimensão de entrada; e o decodificador, que aumenta-a com base na representação intermediária. A utilização por técnicas de aprendizado de máquina mostrou-se promissora na redução da dimensionalidade dos dados ([SHAHABSAMANI, 2024](#)), bem como integrada à janelas deslizantes ([BAIG et al., 2020](#)).

Pela seleção de atributos, porém, é possível reduzir a dimensionalidade escolhendo as características mais relevantes do conjunto de dados. Isso leva à situações onde modelos de Aprendizado de Máquina conseguem generalizar melhor ([DOMINGOS, 2012](#)), além de reduzir o tempo de treinamento e manter a precisão ([PASQUINI; STADLER, 2017](#)) ([SHAHABSAMANI, 2024](#)).

Dentre as técnicas de seleção de atributos é importante descrever a *Forward Stepwise Feature Selection* (traduzida do inglês para “Seleção de Características por Passos Incrementais”) e *Univariate Feature Selection* (traduzida como “Seleção Univariada de Características”). A primeira demanda mais recursos computacionais e treina um modelo adicionando variáveis uma a uma até alcançar um conjunto  $X_{reduzido}$ , cuja adição de uma nova variável piora a precisão anteriormente encontrada ([HASTIE; FRIEDMAN; TIBSHIRANI, 2001](#)). A segunda técnica cria uma lista ordenada pela relevância ([KUHN; JOHNSON, 2016](#)) de cada característica por meio de testes estatísticos para estimar a variável alvo, comumente o teste F ([HASTIE; FRIEDMAN; TIBSHIRANI, 2001](#)).

Dada a tendência de *overfitting* em conjuntos de dados de alta dimensionalidade ([SHAHABSAMANI, 2024](#)), comparar os resultados com e sem a utilização de seleção de características é um importante passo para avaliar qualitativamente as técnicas empregadas para previsão, uma vez que assim é possível avaliar se o modelo realmente é capaz de generalizar ou se adaptou à alta dimensionalidade e ao ruído.

Na literatura, Pasquini e Stadler (2017) utiliza o *Forward Stepwise Feature Selection* e encontra conjuntos de dados  $X'$  de até 12 características com precisão similar à utilização do conjunto  $X$  completo. Este trabalho, porém, emprega a técnica *Univariate Feature Selection* parametrizada também para 12 características. A escolha se justifica na limitação dos recursos computacionais destinados aos experimentos por este trabalho.

A Figura 3 ilustra o processo de redução do conjunto de dados  $X$  à um  $X_{reduzido}$  depois de escolher as 12 melhores características a partir do teste F. Observe que não há transformação nas amostras, apenas são selecionadas as 12 características mais relevantes como reportadas.

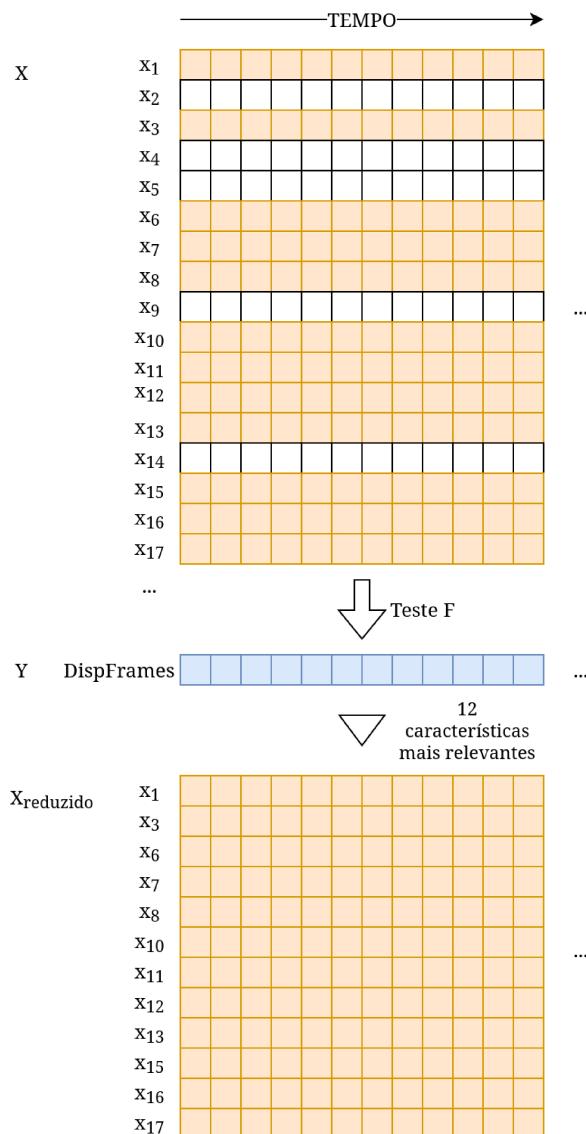


Figura 3 – Ilustração do procedimento de *Univariate Feature Selection* ao aplicar um teste F e encontrar as 12 características mais relevantes que formam o conjunto de dados  $X_{reduzido}$ . Fonte: Elaborado pelo autor.

A ordem de execução utilizada neste trabalho foi: 1) realizar o pré-processamento esperado de  $X$  pelo experimento (detalhados no Capítulo 3) para obter  $X'$  e, em seguida,

2) executar a técnica de seleção de características descrita sobre o  $X'$ . A justificativa está no objetivo do trabalho de simular a limitação na reportagem das amostras por parte do sistema de monitoramento, dessa forma, é parte dos experimentos avaliar se a capacidade de encontrar as 12 características mais relevantes ainda é mantida.

Portanto, o propósito dessa técnica neste trabalho é avaliativa: ela permite comparar se as operações realizadas nos conjuntos de dados realmente permitiram que os modelos generalizassem adequadamente ou apenas se adaptou com a dimensionalidade dos dados escolhidos. Além disso, outro propósito também mencionado é diminuir o custo computacional dos experimentos e por isso a escolha do *Univariate Feature Selection* e não do *Forward Stepwise Feature Selection* utilizado pela literatura ([PASQUINI; STADLER, 2017](#)).

## 2.2 Trabalhos Relacionados

Esta seção apresenta alguns trabalhos relacionados e os posicionam em relação à esta pesquisa. Os parâmetros para treinamento dos modelos referenciados são apresentados, bem como os conjuntos de dados utilizados pela pesquisa, as técnicas capazes de condensar os intervalos e como avaliá-las.

[Pasquini e Stadler \(2017\)](#) estimam a qualidade de serviço de um sistema de transmissão de vídeo ou banco de dados a partir de métricas da infraestrutura (servidores e rede *OpenFlow*) do sistema. Para alcançar esse objetivo, uma série de experimentos variavam a carga (periódica ou repentina) e a quantidade de aplicações servidas (vídeo e banco de dados). Utilizou-se uma infraestrutura de rede *OpenFlow*, cuja coleta das métricas era dada pelo programa SAR ([GODARD, 2007](#)) com granularidade por segundo ou pelo próprio *OpenFlow*, seja por porta ou por fluxo. Dessa forma, com os dados coletados, [Pasquini e Stadler \(2017\)](#) utilizaram dos métodos de aprendizado de máquina Árvore de Regressão e Floresta Aleatória para estimar diferentes métricas de QoS com os dados dispostos de maneira completa, minimizada ou granular. Enfim, mostraram ser possível estimar a qualidade de serviço em aplicações de transmissão de vídeo com dados da infraestrutura com erros inferiores a 10% em média para quadros por segundo, mas não para áudio, bem como avaliaram a possibilidade da utilização de seleção de características ou métricas de partes específicas do *cluster* de forma que conjuntos reduzidos de dados também desempenhavam bem.

Assim como [Pasquini e Stadler \(2017\)](#), este trabalho baseia-se nos parâmetros encontrados e parte do mesmo conjunto de dados com enfoque na estimativa de quadros por segundo e submete os modelos encontrados à técnicas que reduzem a amostragem por métodos alternativos e complementares à seleção de características.

[Shahabsamani \(2024\)](#) descreve técnicas de redução de dimensionalidade concen-

tradas em três métodos principais: seleção de características, PCA e *autoencoders*. Os dois primeiros métodos têm o objetivo de encontrar as variáveis mais relevantes ou combiná-las para a estimativa de métricas, enquanto o último utiliza redes neurais que comprimem dados em uma representação com menor dimensionalidade. O autor apresenta resultados robustos e conclui que conjuntos de dados densos podem ser representados por subconjuntos reduzidos que preservam características relevantes, conhecida como *Manifold Hypothesis*. Assim como os métodos descritos, este trabalho reduz a dimensionalidade de conjuntos densos de dados com uma metodologia alternativa onde a redução ocorre nas amostras disponíveis para o modelo. Assim, [Shahabsamani \(2024\)](#) fundamenta os experimentos deste trabalho como técnica de avaliação de modelos sob diferentes dimensionalidades e amostragens ao utilizar validação cruzada.

[Baig et al. \(2020\)](#) aborda a utilização de janelas deslizantes em sistemas orquestradores de *datacenters* de tamanho fixo e adaptivo. O autor conseguiu reduzir o *overfitting* e a baixa representatividade de padrões incomuns, bem como encontrou um método de janela adaptativa que funciona bem, mas as janelas de tamanho fixo funcionaram melhor que as outras duas janelas adaptativas. Este trabalho se fundamenta nestes resultados com a utilização de janelas de tamanho fixo para reduzir a amostragem do conjunto de dados de forma que tenha uma alternativa segura de *overfit* e também com um volume menor de amostras.

Com os trabalhos apresentados nesta seção, é possível unificar os esforços e utilizar as técnicas propostas para avaliar o comportamento dos modelos com a utilização de técnicas para reduzir a amostragem. [Pasquini e Stadler \(2017\)](#) exploram a estimativa de QoS com dados granulares e redução por seleção de características. [Shahabsamani \(2024\)](#) destaca a preocupação com a dimensionalidade e embasa a Validação Cruzada como método de comparação aos resultados com os conjuntos de dados reduzidos, enquanto [Baig et al. \(2020\)](#) avalia a utilização de janelas como meio de reduzir a amostragem e diminuir o *overfitting*. Com esses estudos, embasa-se a fundamentação teórica dessa pesquisa e a proposta dos experimentos com janelas como alternativa à amostragem para estimar a QoS em aplicações de *streaming*.

### 3 Desenvolvimento

Este capítulo descreve o desenvolvimento do trabalho, sendo que o foco está na descrição detalhada de cada técnica de pré-processamento adotada para redução do período de reportagem pelo sistema de monitoramento e está subdividido da seguinte maneira: 1) Experimento 1: eliminação de amostras do conjunto de dados em intervalos de  $p$  segundos, com  $p$  variando exponencialmente entre 2 e 256 segundos; 2) Experimento 2: análise de janelas de dados em intervalos de  $j$  segundos variando exponencialmente entre 2 e 256 segundos, aplicando funções tradicionais de mínimo, máximo e média; 3) Experimento 3: engenharia de características com as informações da distribuição da métrica em intervalos de  $d$  segundos, variando exponencialmente de 8 a 256 segundos, considerando as seguintes estatísticas: média, desvio padrão, mediana, assimetria, curtose, percentil 25, percentil 75 e amplitude; e 4) Experimento 4: um experimento proposto com as 3 técnicas descritas anteriormente com cerca de apenas 20% dos dados disponíveis.

Com objetivo de comparar aos resultados de [Pasquini e Stadler \(2017\)](#), os experimentos utilizaram 120 estimadores para o Floresta Aleatória e 12 características para o algoritmo de seleção. As demais configurações são as padrões da biblioteca para as árvores, isto é, o mínimo de duas amostras nas partições, independentemente da profundidade, e utilizando a MSE como função de avaliação ([PEDREGOSA et al., 2011](#)). O K-Fold utiliza  $k = 3$  para aproximar do tamanho de partições de teste de referência (30% ([PASQUINI; STADLER, 2017](#))). O erro é mensurado com a NMAE.

A Figura 4 ilustra a arquitetura proposta por este trabalho, que estende o cenário inicial em [Pasquini e Stadler \(2017\)](#), no qual a coleta de  $X$  e  $Y$  não considerava etapas de pré-processamento. Neste trabalho, conforme ilustrado na Figura 4, propõe-se a inclusão de uma etapa de pré-processamento dos dados obtidos pelo sistema de monitoramento por meio da eliminação de amostras, agregação em janelas ou descrição estatística, abordadas na sequência deste capítulo. A arquitetura proposta considera ainda a possibilidade de utilizarmos técnicas de seleção de características, conforme a Figura 4. A nova arquitetura incorpora funcionalidades responsáveis por reduzir a carga computacional e volume de dados transferidos pelo monitoramento contínuo.

#### 3.1 Experimento 1: Eliminação de Amostras

A primeira abordagem proposta simula a mudança da frequência de coleta dos dados por um sistema de monitoramento. Dado um parâmetro  $p$ , este experimento escolhe as amostras de  $p$  em  $p$  segundos e elimina aquelas que estavam entre elas. Portanto, o conjunto de dados resultante é  $p$  vezes menor que o original e é possível analisar como os

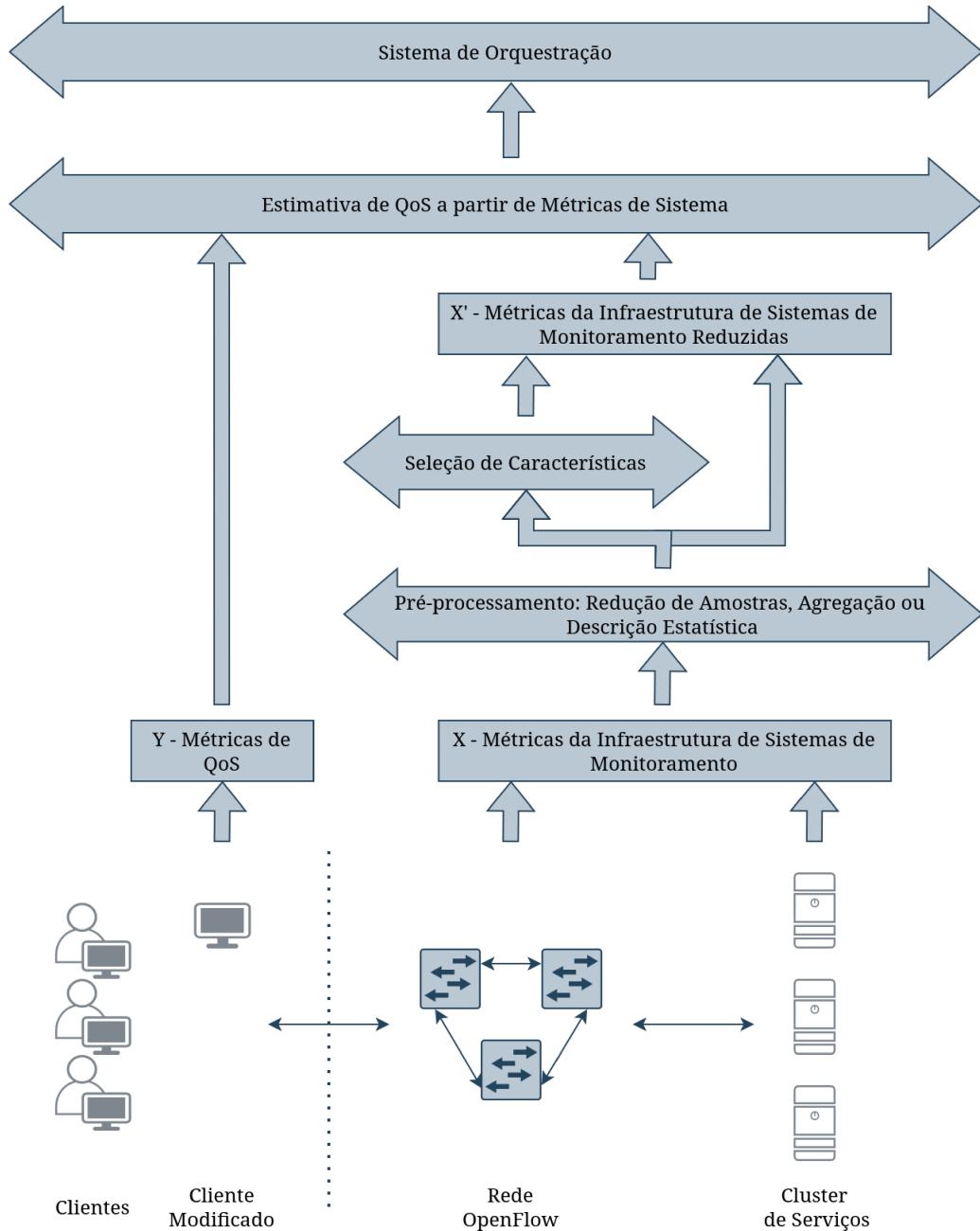


Figura 4 – Arquitetura proposta neste trabalho. Métricas brutas coletadas pelo sistema de monitoramento da infraestrutura ( $X$ ) são pré-processadas (eliminação de amostras, agregação em janelas ou descrição estatística, descritas neste capítulo) e opcionalmente submetidas à seleção de características, gerando  $X'$ . Esses dados são utilizados para estimar métricas de QoS associadas à cada amostra ( $Y$ ) que alimentam decisões de orquestração. Fonte: Elaborado pelo autor.

modelos se comportam na medida em que  $p$  aumenta.

O processo de eliminação das amostras avalia  $p$  segundos e apenas a primeira é mantida, as demais  $p - 1$  são removidas. A Figura 5 ilustra o processo para  $p = 4$ , onde uma amostra  $(x, y)$  é selecionada a cada 4 amostras do conjunto original, resultando nos conjuntos  $X'$  e  $Y'$  com 25% do tamanho original. Note que as amostras não sofrem

processamento e as amostras intermediárias são perdidas.

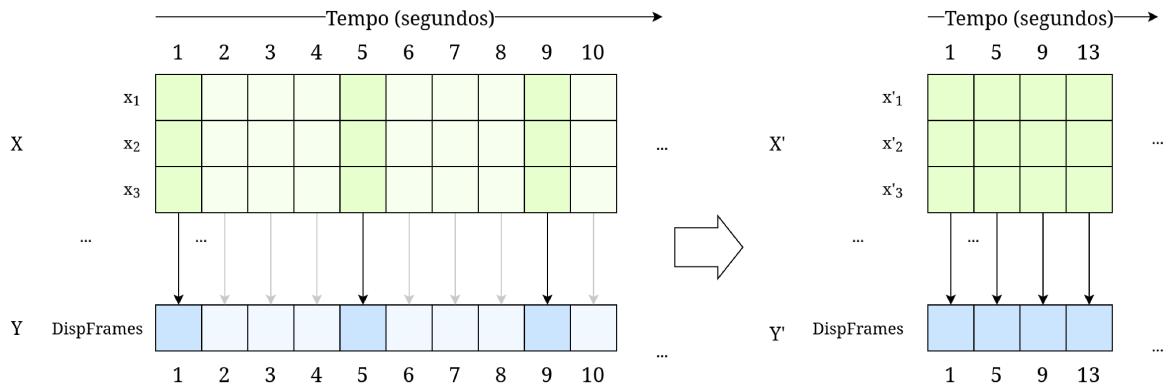


Figura 5 – Ilustração de como o Experimento 1 seleciona as amostras a cada  $p$  segundos e elimina as restantes, neste exemplo  $p = 4$ . Fonte: Elaborado pelo autor.

Os parâmetros  $p$  são definidos exponencialmente como 2, 4, 8, 16, 32, 64, 128 e 256. Assim, cada passo  $p$  simula uma redução de 50% do tamanho do conjunto de dados.

Embora seja a alternativa mais fácil de implementar e já tenha sido utilizada em outros estudos com diferentes valores de  $p$  (MARQUES, 2020), este método apresenta uma limitação significativa: a possibilidade de remover métricas importantes para a identificação de padrões de curta duração. Essa perda de informação pode impactar a capacidade de generalização do sistema ou ter os conjuntos de dados  $X'$  e  $Y'$  enviesados.

Tanto no conjunto de dados original quanto neste experimento, cada amostra  $(x, y)$  representa apenas um instante específico do sistema e não o que aconteceu no sistema entre as amostras. Ao alterar o valor de  $p$ , este intervalo não reportado aumenta de 1 segundo para  $p - 1$  segundos entre cada amostra. Isso implica que o comportamento do sistema pode ser distorcido em  $X'$  e  $Y'$ .

## 3.2 Experimento 2: Janelas

Dada a limitação identificada no Experimento 1, a ideia do Experimento 2 propõe uma alternativa para contornar a perda de informação decorrente da eliminação de amostras. A ideia é utilizar uma função  $f$  aplicada em janelas de tamanho  $j$  para criar novas amostra. Cada amostra  $x'$  representa o resultado da função  $f$  aplicada nos últimos  $j$  segundos de dados, enquanto  $y'$  corresponde à QoS do cliente no instante final da janela, ou seja, para obter  $y'$ , não se aplica nenhuma função de agregação.

As funções  $f$  escolhidas para o pré-processamento foram a média, o máximo e o mínimo. Portanto este experimento cria três conjuntos alterados  $X'_{\text{média}}$ ,  $X'_{\text{mínimo}}$  e  $X'_{\text{máximo}}$ , um para cada  $f$ , e são avaliados separadamente pelos modelos. Elas foram escolhidas pela simplicidade da implementação, mas também porque a média fornece uma visão geral do comportamento do sistema e as funções máximo e mínimo capturam os valores

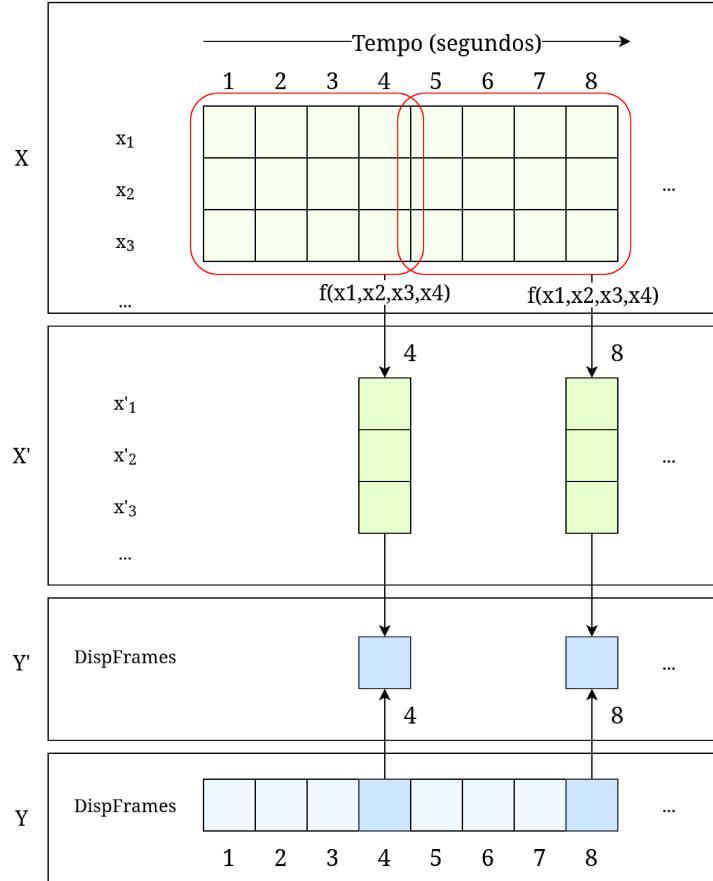


Figura 6 – Ilustração do pré-processamento realizado pelo Experimento 2 com  $j = 4$ .  
Fonte: Elaborado pelo autor.

extremos do intervalo, essenciais para identificar sobrecarga ou ociosidade. Essas funções são aplicadas em janelas de tamanho e passo  $j$ . No entanto, é importante destacar que as funções máximo e mínimo são mais sensíveis à *outliers*.

Os valores de  $j$ , assim como no primeiro experimento, são definidos exponencialmente como 2, 4, 8, 16, 32, 64, 128 e 256. Com isso, é possível comparar os resultados entre os experimentos e manter o comportamento de redução à metade do tamanho do conjunto de dados a cada aumento no valor de  $j$ .

Assim como no Experimento 1, o tamanho de cada  $X'$  e  $Y'$  é reduzido proporcionalmente ao tamanho de  $j$ . Entretanto, ao contrário do primeiro experimento, cada amostra  $x'$  representa de forma compacta os últimos  $j$  segundos e cada  $y'$  corresponde a QoS no instante final da janela.

A Figura 6 ilustra um exemplo em que  $j = 4$ . Observe que a aplicação das janelas ocorre apenas em  $X$ , enquanto as amostras intermediárias de  $Y$  são excluídas, assim como no Experimento 1. Além disso, cada amostra  $y'$  corresponde ao mesmo instante da última

amostra de uma janela em  $x$ .

### 3.3 Experimento 3: Janelas da Distribuição das Métricas

O Experimento 3 propõe uma abordagem alternativa ao Experimento 1 ao utilizar janelas, assim como o Experimento 2, mas ao invés de agregar com uma única função, este método representa cada janela por meio de 8 estatísticas descritivas: média, desvio padrão, mediana, assimetria, curtose, percentil 25, percentil 75 e amplitude. Portanto, essas estatísticas juntas apresentam diferentes informações da distribuição das métricas dentro da janela e também do comportamento do sistema neste intervalo de tempo.

Os valores de  $j$  diferem das demais e possuem tamanho exponencialmente definidos como 8, 16, 32, 64, 128 e 256, devido à criação de 8 novas métricas. O primeiro valor não resultaria em uma diminuição direta do tamanho do conjunto de dados, mas permitiria que um sistema condensasse informações e reportasse com menor frequência. Diferentemente dos demais, inicia-se com um conjunto de dados  $X'$  de mesmo tamanho que  $X$ , mas reduz à metade em cada incremento previsto pelo trabalho.

Uma diferença importante em relação ao Experimento 2 é a preservação e representação da variabilidade existente nas métricas que serão comprimidas nas 8 novas estatísticas. *Outliers*, por exemplo, seriam avaliados de acordo com o desvio padrão, enquanto a função máximo e mínimo do Experimento 2 consistentemente escolheriam aquele valor. Isso permite que os modelos treinados não percam informações da oscilação do sistema dentro de uma janela.

Uma importante constatação é que transformar cada característica em 8 novas estatísticas de certa forma contraria a *Manifold Hypothesis*, uma vez que isso aumentaria a dimensionalidade e poderia introduzir ruído. Entretanto, junto a uma técnica de seleção de características, é possível filtrar as melhores representações das métricas de acordo com as estatísticas obtidas. Dessa maneira, o Experimento 3 com utilização do *Univariate Feature Selection* se comportaria como o Experimento 2 também com seleção de características, mas com diferencial de que a função utilizada no pré-processamento de cada uma das 12 métricas seria a mais indicada de acordo com um teste estatístico.

A Figura 7 ilustra a aplicação das janelas para  $j = 16$ . Observe que cada métrica é transformada em 8 novas estatísticas, que são associadas à métrica de QoS  $y'$  correspondente ao último instante da janela.

### 3.4 Experimento 4: Diminuição do Tempo de Análise

Com o objetivo de avaliar como o tempo total de treinamento afeta o comportamento dos experimentos e a influência dos parâmetros definidos, o Experimento 4 realiza

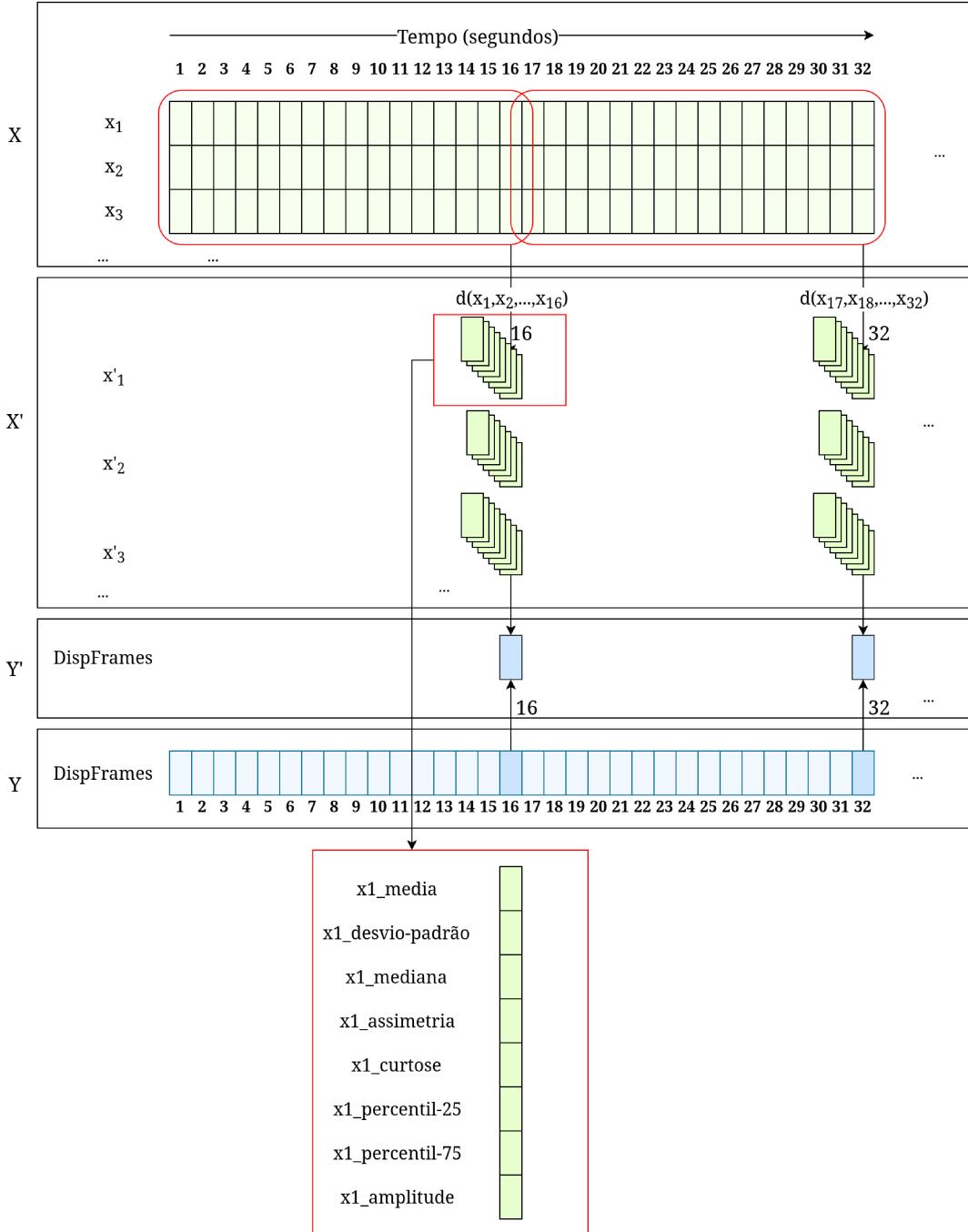


Figura 7 – Ilustração do pré-processamento do Experimento 3 realizado com  $j = 16$ . Cada métrica em uma janela de  $j$  segundos é transformada em 8 diferentes características estatísticas. Fonte: Elaborado pelo autor.

o pré-processamento dos três primeiros experimentos, mas utilizando um conjunto de entrada composto apenas pelas primeiras 7200 amostras do conjunto de dados (equivalente às duas primeiras horas do experimento ou aproximadamente 20% do total de amostras disponíveis), conforme ilustrado na Figura 8.

A redução da quantidade de amostras disponíveis para treinamento e teste pode impactar diretamente a precisão dos modelos. Um experimento associado a um parâmetro  $p$  ou  $j$  pode apresentar menor precisão devido a uma quantidade de dados de treinamento

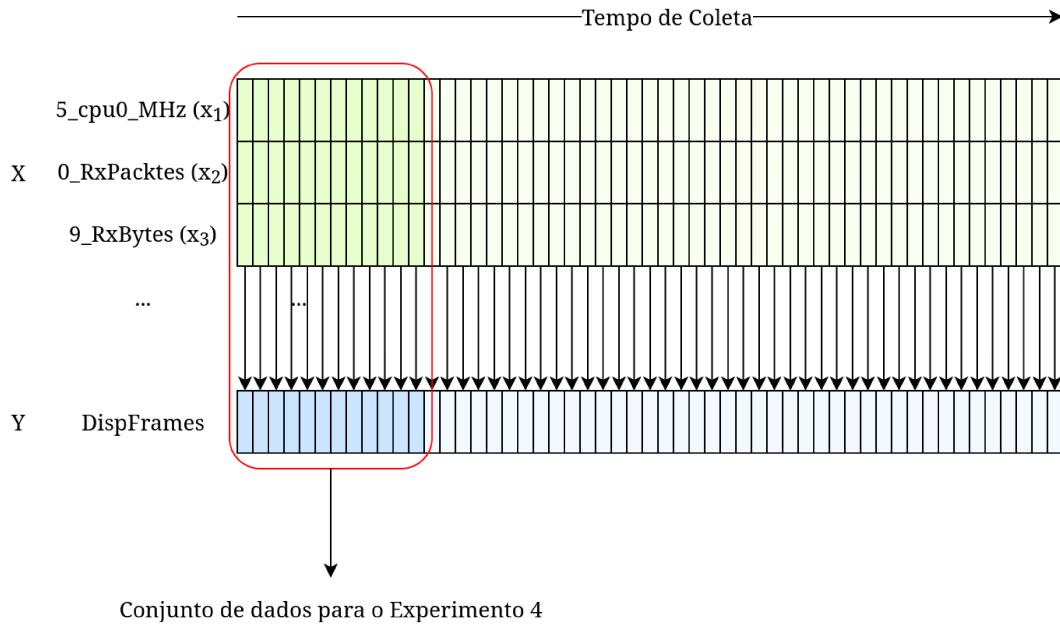


Figura 8 – Exemplificação do corte realizado nos conjuntos de dados em 7200 amostras, equivalente a aproximadamente 20% dos conjuntos de dados originais. Fonte: Elaborado pelo autor.

insuficiente, ainda que, com mais amostras, pudesse convergir para a mesma precisão observada nos experimentos anteriores. Dessa forma, um conjunto de dados incompleto é utilizado neste experimento, e a diferença entre os resultados dos modelos treinados com ele e os demais experimentos pode indicar que determinados parâmetros  $p$  ou  $j$  requerem um volume maior de amostras para garantir convergência.

# 4 Resultados

Este capítulo apresenta os resultados e as análises de cada experimento realizado. O objetivo é avaliar como cada experimento se comportou à medida em que o tamanho do conjunto de dados diminuía e o período de reportagem aumentava. Primeiramente, cada experimento proposto no Capítulo 3 é analisado em sua respectiva seção e os resultados são avaliados com tabelas com granularidade por carga, aplicações em execução, algoritmo de Aprendizado de Máquina e por período, também são apresentados gráficos *boxplot* e comparações entre os casos com e sem a utilização da técnica de seleção de características. Ao final da explicação do Experimento 3 existe uma comparação entre todos os experimentos 1, 2 e 3 na Tabela 8. Em seguida, os resultados do Experimento 4 são reportados e permitem notar a relevância do tempo total do experimento para as técnicas de pré-processamento propostos. Finalmente avalia-se o desempenho da técnica de validação cruzada que obteve resultados mais consistentes e com amostragens mais fiéis ao conjunto de dados original, visualizada por meio de gráficos das densidades das amostras, ilustradas na Figura 14. Todos os resultados e o código utilizado estão publicamente disponíveis em um repositório *git* ([VILARINHO, 2025](#)).

## 4.1 Experimento 1: Eliminação de Amostras

Os resultados encontrados pelo Experimento 1 estão disponíveis na Tabela 1 e são comparadas aos resultados originais da literatura ([PASQUINI; STADLER, 2017](#)) em  $p = 1$ . Nela, estão presentes os resultados percentuais da NMAE (do inglês, *Normalized Mean Absolute Error*), métrica usada em ([PASQUINI; STADLER, 2017](#)), para cada um dos períodos avaliados com base no algoritmo de aprendizado de máquina utilizado, quantas aplicações em execução pelo *cluster* e qual o tipo da carga.

Quando comparado ao caso mais difícil, numericamente determinado como periódico com ambas aplicações em execução paralela, nenhum outro conjunto de dados, em qualquer  $p$  ou algoritmo, apresentou resultados piores do que ele. Ou seja, a mera exclusão das amostras não tornou um caso anteriormente fácil em mais difícil de estimar que o conjunto de dados mais complexo originalmente.

Entretanto, dentro de um mesmo conjunto de dados, é possível observar a degradação das estimativas. A diferença entre o máximo e o mínimo da NMAE é fortemente impactada pelo método escolhido, o *Random Forest* apresentou amplitude igual ou menor à do *Regression Tree*, exceto no cenário mais difícil, onde teve maior amplitude.

Assim como [Pasquini e Stadler \(2017\)](#) encontrou, o *Random Forest* apresentou

Carga	Aplicações	Algoritmo	NMAE (%) por $p$								
			1	2	4	8	16	32	64	128	256
Periódica	Streaming	RT	10	10	11	11	11	12	11	9	11
		RF	9	9	10	10	10	10	10	9	11
	Ambas	RT	13	14	14	14	16	16	15	15	14
		RF	12	13	13	14	14	15	16	14	16
Repentina	Streaming	RT	10	10	11	10	11	11	12	11	10
		RF	9	9	10	10	10	11	11	11	9
	Ambas	RT	10	9	9	10	11	9	11	12	13
		RF	8	8	9	9	10	9	10	11	11

Tabela 1 – Experimento 1: Tabela de NMAE para diferentes cargas, aplicações e métodos para cada período de raleamento.  $p = 1$  é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). RT representa o algoritmo *Regression Tree* e RF o *Random Forest*. Fonte: elaborado pelo autor.

melhores resultados consistentemente, a exceção de  $p = 64$  e  $p = 256$ , para o caso periódico com ambas aplicações em execução paralelamente. Este algoritmo é bem mais caro computacionalmente e a precisão em diferentes condições de raleamento justificam este custo, na medida em que a quantidade de métricas necessárias para treinamento reduz, mas mantém uma boa precisão.

Uma observação importante é que no caso repentino, para ambas aplicações, o raleamento até permitiu melhorar a estimativa original em 1 ponto percentual para  $p = 2, 4$  e 32. Uma possível explicação é que a remoção de algumas amostras intermediárias poderia omitir estatísticas de instantes muito próximos, as quais, se não forem significativas, podem impactar o treinamento como ruído.

A Tabela 2 permite visualizar os resultados obtidos quando utilizado a seleção de características, onde  $p = 1$  representa o resultado original da literatura ([PASQUINI; STADLER, 2017](#)), também com seleção de características. É possível observar que houve uma grande diferença quanto ao valor de comparação no primeiro e último caso, provavelmente em decorrência da técnica utilizada. Entretanto, também é possível observar como os modelos apresentaram um comportamento bem uniforme, excluído novamente o caso periódico com ambas aplicações em paralelo. Em todo o intervalo de  $4 \leq p \leq 32$  (exceto o caso periódico com ambas aplicações em paralelo), apenas uma das 24 NMAEs não foi 10 ou 11.

Baseado nas informações presentes na Tabela 1 e 2, as janelas com 128 e 256 segundos são intervalos grandes o suficiente para que os resultados obtidos sejam bem menos previsíveis. O caso periódico com ambas aplicações alcançou um resultado muito pior, com ou sem a utilização de seleção de características, já a partir de valores de  $p \geq 8$ .

A Figura 9 ilustra a diferença entre utilizar ou não a seleção de características no Experimento 1. A adoção da representação em *boxplot* na Figura 9 visa permitir ao

Carga	Aplicações	Algoritmo	NMAE (%) por $p$								
			1	2	4	8	16	32	64	128	256
Periódica	Streaming	RT	14	10	11	11	11	10	10	9	10
		RF	11	9	10	10	10	10	9	8	9
	Ambas	RT	12	14	15	16	16	16	18	13	18
		RF	11	13	14	14	14	15	16	12	14
Repentina	Streaming	RT	11	11	11	11	11	12	10	11	8
		RF	9	10	10	11	11	11	10	11	8
	Ambas	RT	10	11	11	11	11	10	11	13	10
		RF	6	10	10	11	11	10	10	11	10

Tabela 2 – Experimento 1 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada período de raleamento.  $p = 1$  é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: elaborado pelo autor.

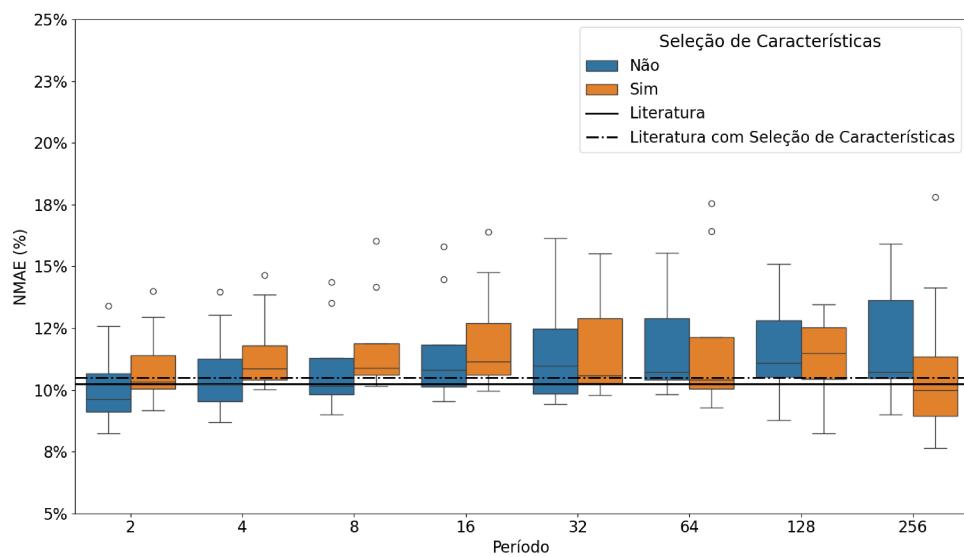


Figura 9 – *Boxplot* da NMAE do Experimento 1 comparando a utilização de Seleção de Características. As linhas representam os resultados médios da literatura ([PASQUINI; STADLER, 2017](#)), a contínua quando não utilizada seleção de características e a tracejada com uso da técnica. Fonte: Elaborado pelo autor.

leitor condensar as informações das Tabelas 1 e 2 para cada valor de  $p$  e variação de métricas (com e sem seleção de características). Por exemplo, o *boxplot* azul para  $p = 2$  na Figura 9 representa os oito valores da Tabela 1, na coluna  $p = 2$ . As linhas que cortam horizontalmente o gráfico correspondem aos resultados extraídos da literatura ([PASQUINI; STADLER, 2017](#)), para  $p = 1$  com e sem a seleção de características. A mesma metodologia aplica-se aos demais gráficos *boxplot* deste capítulo.

É possível observar na Figura 9 que a utilização do método de seleção de características (ou seja, redução de dimensionalidade) apresenta maior uniformidade quando o grau de raleamento do conjunto de dados aumenta. Isso pode ser observado pela diferença do intervalo de confiança em comparação à não utilização dele.

Pela *Manifold Hypothesis* ([SHAHABSAMANI, 2024](#)), infere-se que para valores menores de  $p$ , onde temos conjuntos de dados maiores, ou seja, com mais amostras, a alta dimensionalidade não se demonstra como um problema. No outro extremo, onde temos poucas amostras (valores maiores de  $p$ ), a redução de dimensionalidade é indispensável, uma vez que traz balanceamento entre o número de características e o número de amostras. Neste contexto, os resultados do Experimento 1 sugerem que a frequência para coleta de amostras guarda relação com a quantidade de características utilizadas no treinamento dos modelos. Os conjuntos de dados em sua integralidade (aproximadamente 10.500 características) possuem boa relação características/amostragem para valores de  $p \leq 16$ . Os conjuntos de dados resultantes do processo de seleção de características (neste trabalho, com 12 características) possuem boa relação características/amostragem para valores de  $p \geq 64$ . O cenário onde  $p = 32$  parece desbalanceado para ambas as situações investigadas no Experimento 1, conforme detalhado na Figura 9.

## 4.2 Experimento 2: Janelas

O Experimento 2 tem como objetivo unir as informações de uma janela em um valor só, de forma que o modelo tenha informação de como o sistema se comportou desde o último valor reportado, ao contrário do Experimento 1, que apenas elimina as amostras intermediárias. Os resultados do Experimento 2 para a função média estão presentes na Tabela 3. Nela são avaliadas a NMAE de acordo com o conjunto de dados e algoritmo utilizado na medida em que  $j$  aumenta, também comparado à literatura (caso  $j = 1$ ). Destaca-se que os resultados para as demais funções, mínimo e máximo, serão relatados a partir da Tabela 5.

Carga	Aplicações	Algoritmo	NMAE (%) por $j$ função média								
			1	2	4	8	16	32	64	128	256
Periódica	Streaming	RT	10	10	11	10	11	10	12	10	12
		RF	9	9	10	10	10	10	10	11	12
	Ambas	RT	13	14	14	15	14	15	15	18	19
		RF	12	13	13	14	13	14	13	16	17
Repentina	Streaming	RT	10	10	10	11	11	11	11	14	16
		RF	9	9	10	10	10	10	11	12	14
	Ambas	RT	10	9	9	10	9	10	13	13	13
		RF	8	8	9	9	9	10	12	13	12

Tabela 3 – Experimento 2 com função média: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas.  $j = 1$  é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: elaborado pelo autor.

É possível observar que, assim como no Experimento 1, os cenários para  $j = 128$  e  $j = 256$  obtiveram resultados mais imprevisíveis. Porém, neste experimento também é possível atribuir este efeito às NMAEs obtidas em  $j = 64$ , já que há degradação clara em

um caso que anteriormente era de fácil estimativa (carga repentina com ambas aplicações em paralelo), onde partiu de um valor base de 10% e 8% e atingiu 13% e 12% para *Regression Tree* e *Random Forest*, respectivamente.

Outra observação importante é que no cenário repentino, para ambas aplicações utilizando o *Regression Tree*, o Experimento 2 conseguiu um resultado melhor que a literatura para  $p = 2, 4$  e  $16$ .

A Tabela 4 apresenta os resultados, equivalentes à Tabela 3, para os cenários avaliados com a utilização de seleção de características. Um resultado importante neste experimento com seleção de características foi a manutenção de estimativas consistentemente próximas à original por mais tempo. Isso fica claro tanto em cenários de fácil estimativa para  $j < 128$ , mas também pelo fato de que o caso mais difícil (o periódico com ambas aplicações), com o *Random Forest*, variou apenas 2 p.p. de diferença ao original, de forma consistente para  $j < 128$ .

Carga	Aplicações	Algoritmo	NMAE (%) por $j$ função média								
			1	2	4	8	16	32	64	128	256
Periódica	Streaming	RT	14	10	10	10	10	12	12	11	12
		RF	11	9	10	10	10	11	10	11	11
	Ambas	RT	12	14	15	17	15	14	14	15	11
		RF	11	13	14	15	14	13	13	18	15
Repentina	Streaming	RT	11	11	11	12	12	12	11	16	13
		RF	9	10	10	11	11	11	10	11	13
	Ambas	RT	10	11	10	11	10	9	11	12	8
		RF	6	10	10	11	9	9	11	9	10

Tabela 4 – Experimento 2 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas com utilizando função média quando submetido à técnica de redução de dimensionalidade *Univariate Feature Selection*.  $j = 1$  é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: elaborado pelo autor.

Outro resultado importante da utilização de seleção de características é que para  $j < 128$ , o *Random Forest* apresentou melhor resultado consistentemente. Entretanto, para as duas maiores janelas houve uma inversão deste resultado. O fato dessa diferença ser observada apenas para janelas significativamente grandes pode indicar que o pré-processamento introduziu mudanças na distribuição dos dados que alteraram a natureza do problema. Em suma, aplicar a média dos valores amostrados para cada característica, em intervalos  $j \geq 128$ , agrupa amostras de ambos estados do sistema (baixa e alta resolução), resultando em uma média entre esses extremos que não se enquadra, precisamente, em nenhuma das regiões da natureza original do problema - próximo de 13 ou de 24 quadros por segundo ( $Y$ , originalmente, apresenta distribuição bimodal, conforme ilustrado na Figura 2).

Janela	Sem Seleção de Características			Com Seleção de Características		
	Média	Mínimo	Máximo	Média	Mínimo	Máximo
1 (lit)	10 ± 1	10 ± 1	10 ± 1	11 ± 2	11 ± 2	11 ± 2
2	10 ± 2	10 ± 2	10 ± 2	11 ± 1	11 ± 2	11 ± 1
4	11 ± 2	11 ± 2	11 ± 2	12 ± 2	12 ± 2	12 ± 2
8	11 ± 2	11 ± 2	11 ± 2	12 ± 2	12 ± 2	12 ± 2
16	11 ± 2	11 ± 1	11 ± 2	12 ± 2	12 ± 2	12 ± 2
32	11 ± 2	11 ± 2	11 ± 1	11 ± 2	11 ± 1	12 ± 2
64	12 ± 1	12 ± 1	12 ± 1	12 ± 1	12 ± 1	12 ± 2
128	13 ± 2	13 ± 2	14 ± 2	13 ± 3	14 ± 3	14 ± 3
256	15 ± 2	14 ± 2	15 ± 3	12 ± 2	14 ± 4	14 ± 2

Tabela 5 – NMAE média e intervalo de confiança (teste  $t$  de Student) em valores porcentuais para o Experimento 2, comparando a função utilizada no pré-processamento e quanto à utilização de seleção de características.  $j = 1$  é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: Elaborado pelo Autor.

A Tabela 5 apresenta os resultados do Experimento 2 de acordo com função de agregação utilizada nas janelas no pré-processamento, comparados quanto à utilização da seleção de características. Cada linha representa o tamanho da janela e  $j = 1$  representa a referência da literatura, os valores presentes na tabela são a NMAE média calculada das NMAEs obtidas em cada um dos quatro conjuntos de dados, dos dois métodos (*Random Forest* e *Regression Tree*) quando utilizada a função de agregação em questão, o intervalo de confiança é calculado a partir do teste  $t$  de Student.

É possível inferir pela Tabela 5 que os resultados para cada função de agregação sem seleção de características foram similares. Para  $j \geq 128$  a degradação é clara, enquanto para  $j < 128$  as NMAEs em cada janela foram idênticas, com diferenças apenas quanto ao intervalo de confiança da função mínimo em  $j = 16$  e máximo em  $j = 32$ , as quais foram 1 e não 2 como nas demais reportadas para esses parâmetros.

Por outro lado, a Tabela 5 mostra que o resultado foi diferente quando submetido à seleção de características. Para  $j \leq 16$  a NMAE também foi idêntica em cada  $j$ , com alteração apenas no intervalo de confiança para a função mínimo com  $j = 2$ . Entretanto, para  $j > 16$  já existe uma maior divergência nos resultados, e a degradação para a média é ligeiramente melhor do que para as funções mínimo e máximo para  $j \geq 128$ .

A Figura 10 compara a NMAE obtida de acordo com a função de agregação e quanto ao uso de seleção de características para cada tamanho de janela  $j$  avaliado. Os resultados consideram todos os conjuntos de dados utilizados pelo trabalho. Novamente, destaca-se que as duas linhas horizontais correspondem aos resultados da literatura ([PASQUINI; STADLER, 2017](#)) para  $p = 1$ .

Analizando a Figura 10 é possível inferir que as janelas de 128 e 256 segundos são

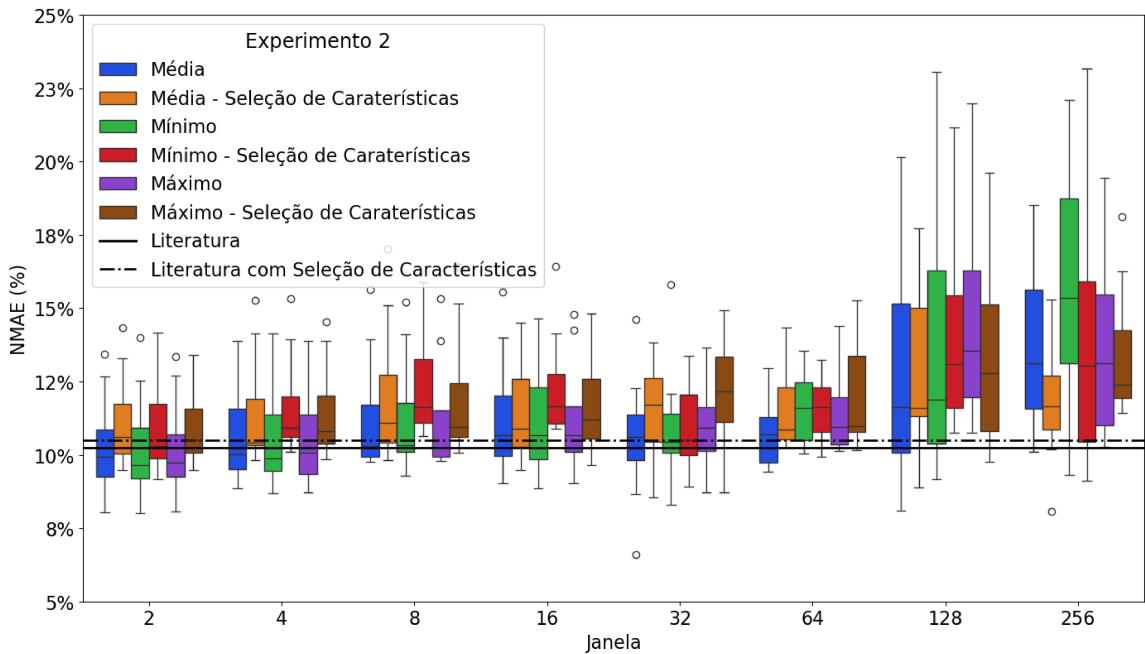


Figura 10 – *Boxplot* da NMAE para diferentes tamanhos de janela  $j$  Experimento 2. Cada caixa representa uma função de agregação utilizada no pré-processamento e se considera a utilização de seleção de características ou não para o valor de  $j$  associado. As linhas representam os resultados médios da (PASQUINI; STADLER, 2017), onde a contínua é a NMAE média sem utilizar a seleção de característica e a tracejada com uso da técnica. Fonte: Elaborado pelo autor.

grandes demais para os algoritmos e quantidade de amostras utilizadas. Mesmo assim, com seleção de características, é nítido que a função média em conjunto com uma técnica de seleção de características teve bons resultados, relativamente estáveis e com intervalos de confiança melhores que as outras funções em geral.

Porém, pela Tabela 5, os resultados das diferentes funções  $f$  utilizadas pela agregação das janelas foram similares, exceto pela função média, especificamente quando utilizada seleção de características para os períodos maiores, quando obteve um resultado melhor que as demais. Uma possível explicação é que na medida em que a janela cresce, maior a chance de que nela esteja presente um valor *outlier* que a função mínimo ou máximo poderiam retornar. Em algumas métricas estes valores podem ser fundamentais para identificar um padrão de sobrecarga ou ociosidade, mas em outras podem ser ruído da operação do sistema, o que implica na diferença observada dos resultados da função média e das demais utilizadas pelo Experimento 2.

Janelas com valores de  $p \leq 64$ , mesmo sem a seleção de características, apresentam desempenho satisfatório, indicando que a transformação das métricas através de funções (média, mínimo e máximo) tendem a preservar a distribuição bimodal no conjunto  $X$ , incorporando valores de amostras antes desconsiderados no Experimento 1. Similar ao reportado no Experimento 1, a alta dimensionalidade encontra-se balanceada com o nú-

mero de amostras, neste caso do Experimento 2, para valores um pouco mais elevados de  $p$  (antes para  $p \leq 16$  no Experimento 1, agora para  $p \leq 64$  no Experimento 2). Por outro lado, ao contrário do comportamento identificado no Experimento 1, os resultados para a seleção de característica com valores maiores de  $p$  não indicam capacidade de treinamento de modelos com bom desempenho. Novamente, a aplicação de funções em janelas maiores de dados tende a distorcer a distribuição bimodal original de  $Y$ .

### 4.3 Experimento 3: Janelas da Distribuição das Métricas

Assim como o Experimento 2, o Experimento 3 propõe uma forma de fornecer informações do período entre as reportagens, ao invés de apenas removê-las, como fez o Experimento 1. Porém, no experimento desta seção, uma janela tem cada métrica descrita em 8 novas métricas, que explicam o comportamento dela nos últimos  $j$  segundos. As NMAEs obtidas com a utilização desta técnica de pré-processamento estão disponíveis na Tabela 6, separadas por carga, aplicações e algoritmo utilizados para cada parâmetro  $j$ . Porém, como indicado na descrição do experimento proposto, a criação e utilização de mais métricas pelo modelo vai contra a *Manifold Hypothesis*. Por isso, os resultados com a utilização da Seleção de Características também estão presentes na Tabela 7, de formato similar, e é recomendado analisá-las em conjunto, uma vez que o aumento da dimensionalidade pode incorrer em ruído.

Carga	Aplicações	Algoritmo	NMAE (%) por $j$ amostras descritas						
			(lit)	8	16	32	64	128	256
Periódica	Streaming	RT	10	10	11	11	11	10	13
		RF	9	10	10	10	10	11	12
	Ambas	RT	13	15	15	14	15	15	18
		RF	12	14	14	14	13	16	17
Repentina	Streaming	RT	10	11	11	11	11	14	12
		RF	9	10	10	10	11	12	13
	Ambas	RT	10	10	9	10	13	14	12
		RF	8	9	9	10	11	13	12

Tabela 6 – Experimento 3 sem Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas. A primeira coluna da NMAE é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: elaborado pelo autor.

Pela Tabela 6, seria possível estabelecer  $p \leq 64$  como um limite da técnica de pré-processamento proposta. Entretanto, quando ocorre a filtragem das melhores características (reportados na Tabela 7), observa-se já em  $j = 8$  um erro de 16% para o caso mais difícil, quando utilizado o *Regression Tree*, assim como no Experimento 1, diferenciando-se apenas que este resultado só volta a ocorrer, ainda pior, em  $j = 256$ . Portanto, é importante destacar que a redução das amostras com esta técnica de pré-processamento não

somente pode exigir a utilização de redução da dimensionalidade, como também pode exigir algoritmos computacionalmente mais caros para os casos mais difíceis e, nessas condições, seria possível estabelecer um limite de até 128 segundos entre reportagens.

Carga	Aplicações	Algoritmo	NMAE (%) por $j$ amostras descritas						
			(lit)	8	16	32	64	128	256
Periódica	Streaming	RT	14	11	11	11	11	10	15
		RF	11	10	10	10	10	10	12
	Ambas	RT	12	16	15	14	13	15	20
		RF	11	15	14	13	13	14	17
Repentina	Streaming	RT	11	11	12	12	13	11	9
		RF	9	11	11	11	12	11	10
	Ambas	RT	10	12	10	11	10	9	9
		RF	6	11	10	10	11	11	9

Tabela 7 – Experimento 3 com Seleção de Características: NMAE para diferentes cargas, aplicações e métodos para cada tamanho das janelas. A primeira coluna da NMAE é o valor referência da literatura ([PASQUINI; STADLER, 2017](#)). Fonte: elaborado pelo autor.

Os resultados obtidos com seleção de características presentes na Tabela 7 apresentaram valores consistentes em  $8 \leq j \leq 128$ . Neste período, é possível observar que para metade dos experimentos realizados, o caso repentino com *Random Forest* e o caso periódico com apenas *Streaming*, obtiveram amplitude de no máximo 1 ponto percentual. Em especial, para este último caso, os resultados foram os mesmos para o *Random Forest*. Observe que, matematicamente, aumentar a janela até 128 segundos não representa o mesmo grau de redução espacial nos Experimentos 3 e 2. No Experimento 3, a janela ter 128 segundos representa uma redução do conjunto de dados em  $2^4$  vezes, enquanto no Experimento 2 isso seria  $2^7$ . Essa diferença ocorre pelo fato de que o Experimento 3 transforma cada métrica em 8 novas, o que aumenta o tamanho total do conjunto de dados. Este aspecto, portanto, apresenta influência na desejável relação de equilíbrio entre características/amostragem.

A Figura 11 apresenta o *boxplot* das NMAEs observadas pelo Experimento 3. É possível concluir que, assim como observado pelas análises das Tabelas 6 e 7, os resultados são consistentes para uma janela de até 128 segundos e a seleção de características é essencial para a precisão do método, enquanto à falta dela implica em condições de maior degradação da NMAE.

No caso de Experimento 3, conforme detalhado na Figura 11 o equilíbrio características/amostragem faz-se presente para valores de  $p \leq 32$  quando não utilizamos a seleção de características, e com desempenho um pouco inferior, mas ainda assim presente com a seleção de características para valores de  $p \leq 128$ .

Finalmente, a Tabela 8 sumariza os resultados encontrados em todos experimen-

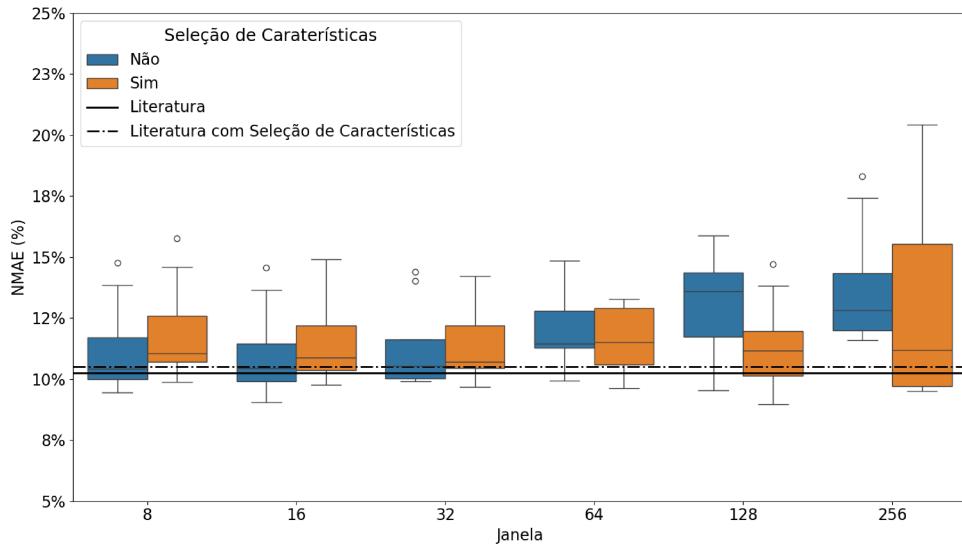


Figura 11 – *Boxplot* da NMAE do Experimento 3 comparando a utilização de Seleção de Características. As linhas representam a média dos resultados da literatura ([PASQUINI; STADLER, 2017](#)), a contínua sem utilização de características e a tracejada com utilização da técnica. Fonte: Elaborado pelo autor.

tos 1, 2 e 3 que foram propostos e, também, quanto à utilização ou não da seleção de características. Os valores presentes nela são a NMAE média que foi calculada a partir da NMAE obtida em cada um dos quatro conjuntos de dados, para cada um dos dois métodos, *Random Forest* e *Regression Tree*, e o intervalo de confiança é calculado à partir do teste *t* de Student. Destaca-se que para o Experimento 2, o valor reportado considera todas as três funções: média, mínimo e máximo.

Pela Tabela 8 é possível concluir que: 1) 256 é um período grande demais entre reportagens para este problema; 2) Os experimentos obtiveram resultados similares; 3) O Experimento 1 aparenta ter bons resultados, mas quando submetido à seleção de características apresenta uma degradação, quando comparado ao Experimento 2 para  $p \geq 8$ ; 4) O Experimento 2 teve o melhor resultado sem a utilização de seleção de características para  $j \leq 64$ ; e 5) O Experimento 3 teve desempenho pior que o Experimento 2 sem utilização de Seleção de Características, porém, junto ao *Univariate Feature Selection*, alcançou bons resultados com até 128 segundos entre reportagens.

#### 4.4 Experimento 4: o Impacto da Quantidade de Amostras

O Experimento 4 tem o objetivo de avaliar a possibilidade dos resultados, para cada tamanho das janelas, serem impactados pela quantidade inicial de amostras disponíveis para o pré-processamento. Assim como esperado, a Tabela 9 apresenta resultados que confirmaram essa hipótese, já que os resultados com apenas 7.200 amostras poderiam ter descartados janelas que se provaram aceitáveis pelo Experimento 3, como  $j = 64$ , bem

Período	Sem Seleção de Características			Com Seleção de Características		
	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3
1 (lit)	10 ± 1	10 ± 1	-	11 ± 2	11 ± 2	-
2	10 ± 2	10 ± 1	-	11 ± 1	11 ± 1	-
4	11 ± 2	11 ± 1	-	12 ± 1	12 ± 1	-
8	11 ± 2	11 ± 1	11 ± 2	12 ± 2	12 ± 1	12 ± 2
16	12 ± 2	11 ± 1	11 ± 2	12 ± 2	12 ± 1	12 ± 2
32	12 ± 2	11 ± 1	11 ± 2	12 ± 2	12 ± 1	11 ± 1
64	12 ± 2	12 ± 1	12 ± 1	12 ± 3	12 ± 1	12 ± 1
128	12 ± 2	13 ± 1	13 ± 2	11 ± 2	14 ± 1	11 ± 2
256	12 ± 2	14 ± 1	14 ± 2	11 ± 3	13 ± 1	13 ± 4

Tabela 8 – Resultados da NMAE (valores percentuais) com intervalos de confiança calculados via teste  $t$  de Student para os experimentos 1 a 3, comparando a utilização de seleção de características. Fonte: Elaborado pelo Autor.

como apresenta valores muito distintos daqueles obtidos ao final de todo o processo de treinamento.

Período	Primeiras 7200 Amostras			Todas Amostras		
	Exp. 1	Exp. 2	Exp. 3	Exp. 1	Exp. 2	Exp. 3
2	8 ± 1	8 ± 1	-	10 ± 2	10 ± 2	-
4	9 ± 1	9 ± 1	-	11 ± 2	11 ± 2	-
8	9 ± 2	10 ± 1	10 ± 1	11 ± 2	11 ± 2	11 ± 2
16	9 ± 2	9 ± 1	9 ± 0	12 ± 2	11 ± 2	11 ± 2
32	10 ± 4	10 ± 1	9 ± 1	12 ± 3	11 ± 2	11 ± 2
64	9 ± 2	11 ± 1	12 ± 4	12 ± 2	12 ± 1	12 ± 2
128	9 ± 4	13 ± 2	14 ± 1	12 ± 2	13 ± 3	13 ± 2
256	12 ± 4	12 ± 2	13 ± 4	12 ± 2	14 ± 3	14 ± 3

Tabela 9 – Comparação dos valores da NMAE em percentual e intervalo de confiança baseado no teste T entre subconjuntos com 7200 amostras e o completo (presentes na Tabela 8 na parte sem seleção de características). Fonte: Elaborado pelo Autor.

A Figura 12 apresenta valores de NMAE, para os distintos experimentos, partindo com valores menores (melhor desempenho) que o resultado original da literatura ([PASQUINI; STADLER, 2017](#)), representado pela linha horizontal. Este resultado, embora pareça benéfico, representa falta de generalização do modelo, ou seja, tendência a *overfitting*, indicando que o conjunto de dados tende a apresentar distribuição unimodal nestas duas primeiras horas dos experimentos.

Dessa forma, o Experimento 4 mostra dois efeitos importantes. Primeiramente, a geração de carga durante os experimentos deve ser capaz de conduzir o serviço (neste caso, *streaming* de vídeo) entre todos os possíveis estados, desde baixa carga até elevada carga, para que o conjunto de dados seja suficientemente representativo dos estados possíveis do sistema, para fins de treinamento de modelos de qualidade e que possam generalizar. Em

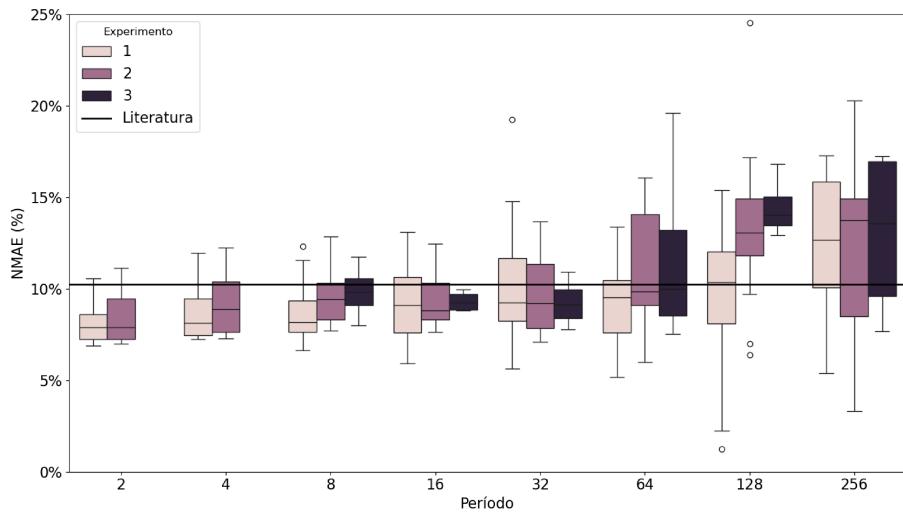


Figura 12 – Boxplot da NMAE para cada experimento com apenas as 7200 amostras disponíveis para o pré-processamento. A linha contínua representa o resultado médio da literatura sem utilização de seleção de características ([PASQUINI; STADLER, 2017](#)). Fonte: Elaborado pelo autor.

segundo lugar, deve-se buscar o equilíbrio entre dimensionalidade do conjunto de dados e o volume de amostras.

Neste contexto, os resultados obtidos anteriormente e reportados em ([PASQUINI; STADLER, 2017](#)) podem também estar limitados à quantidade total de amostras disponíveis em  $X$ . Caso o conjunto de dados e o estudo original ([PASQUINI; STADLER, 2017](#)) abrangesse um tempo de observação superior a dez horas, as janelas com maior período para os experimentos 1, 2 e 3 poderiam ter um desempenho melhor.

Abre-se, portanto, uma importante discussão sobre o quanto célere o treinamento dos modelos precisa ser. Eventualmente, sistemas adaptativos de monitoramento podem fornecer métricas mais rapidamente para o treinamento inicial dos modelos, ao preço de movimentação de maiores volumes de dados de monitoramento (telemetria) e, após um determinado nível de treinamento, gradualmente reduzir a movimentação de dados, visando, tão somente, atualizar o modelo, economizando recursos com a movimentação de menores volumes telemétricos.

## 4.5 Avaliação das Técnicas de Particionamento

Assim como descrito anteriormente, o estudo base ([PASQUINI; STADLER, 2017](#)) utilizou o conjunto de validação com a função `train_test_split` ([PEDREGOSA et al., 2011](#)), este trabalho, porém, utilizou a técnica *K-Fold* para avaliação dos resultados. Esta seção apresenta a diferença dos resultados e dos conjuntos de treinamento obtidos pelas duas técnicas.

A Figura 13 ilustra a diferença dos resultados obtidos pelos experimentos quando:

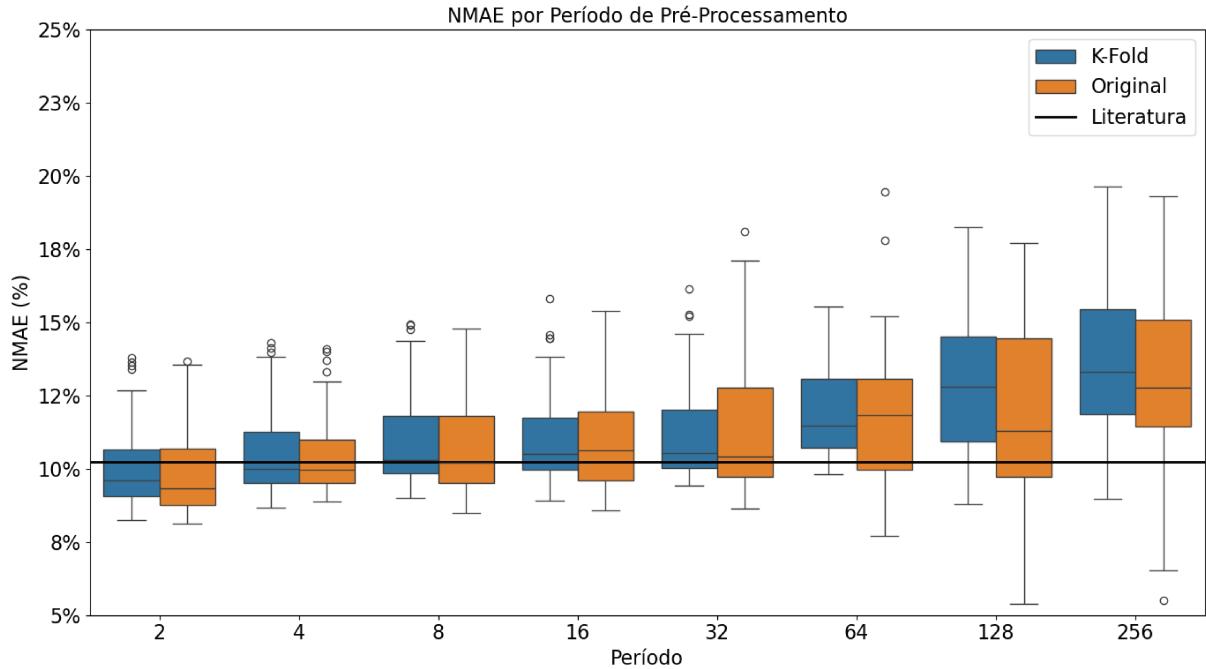


Figura 13 – Diferença da NMAE obtida por todos experimentos sem utilização de seleção de características utilizando o método de validação cruzada *K-Fold* e `train_test_split`. A linha contínua referencia a média da NMAE obtida pela literatura ([PASQUINI; STADLER, 2017](#)) também sem utilização de seleção de características. Fonte: Elaborado pelo autor.

1) original - submetidos à avaliação com particionamento de treinamento (70%) e teste (30%) ([PASQUINI; STADLER, 2017](#)); e 2) *K-Fold* - a utilização de validação cruzada *K-Fold* com  $k = 3$ . É possível observar que o intervalo de confiança dos resultados aumentam na medida em que os parâmetros  $p$  e  $j$  aumentam, assim como esperado. Entretanto, o limite inferior e o primeiro quartil tem uma tendência de degradação mais clara com o *K-Fold*, enquanto o método original tem a média reduzida para os últimos 2 períodos (128 e 256 segundos). Esse efeito pode ser visualizado na Figura 13 pela diferença abrupta entre os resultados com parâmetros 64 e 128.

Uma explicação para a diferença dos resultados das duas formas de particionamento e avaliação no escopo deste trabalho é visualizada pela Figura 14. Esta figura apresenta a densidade do conjunto de  $Y$ ,  $Y_{treinamento}$  e  $Y_{teste}$  para  $p, j = 128$  e para o conjunto original sem pré-processamento. No conjunto original já havia uma diferença na forma como ambas técnicas particionavam, onde o *K-Fold* apresentou menor diferença em relação ao  $Y$  original. Com o pré-processamento, mesmo com uma alta redução da quantidade de amostras, o *K-Fold* mantém as partições com comportamento bimodal mais acentuado, enquanto o `train_test_split` atenua a distância da densidade original de  $Y$ . A diferença dos métodos fica nítida quando comparadas as densidades no vale onde  $13 \leq y' \leq 24$  e para  $y' \leq 10$ . Uma observação também é que a quantidade menor de amostras muda a densidade como um todo em ambos cenários.

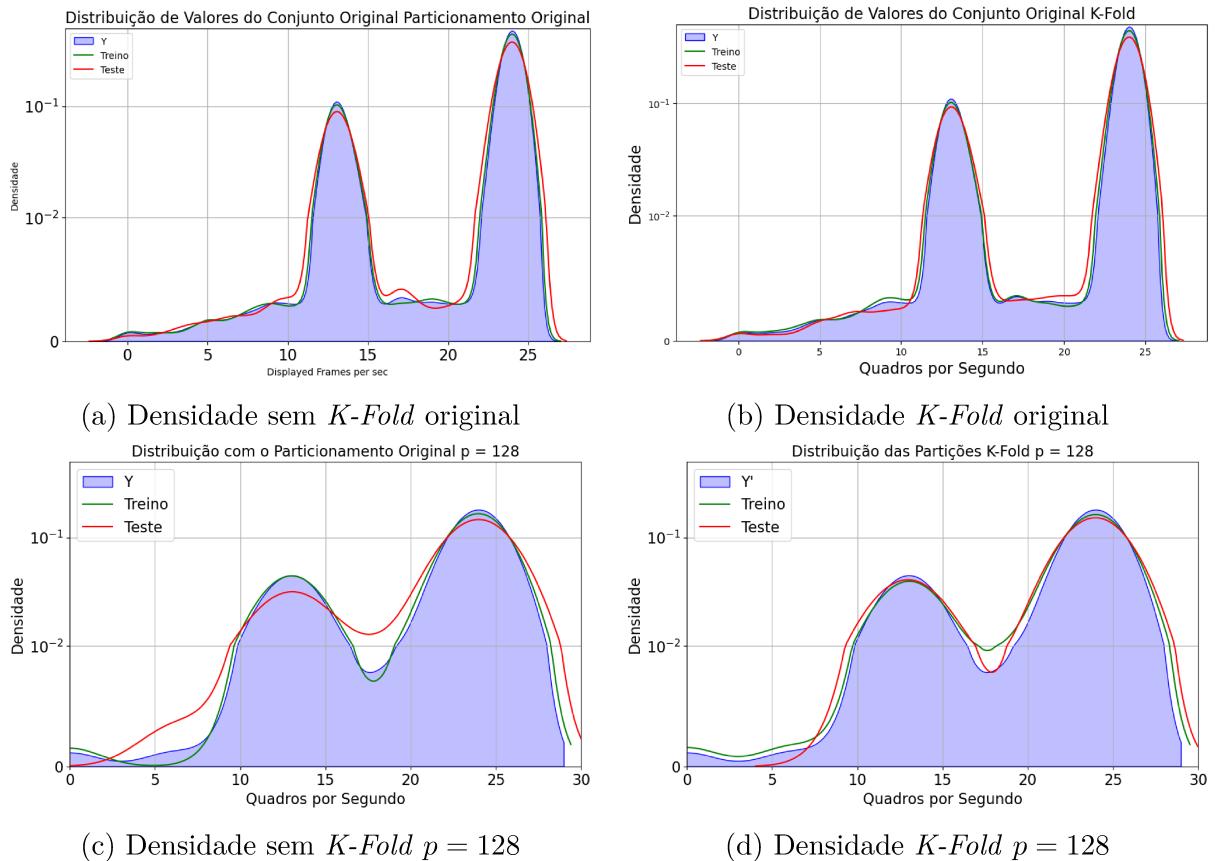


Figura 14 – Comparação das densidades dos conjuntos  $Y'$  em um mesmo grau de raleamento para as partições de teste e treinamento com a utilização do `train_test_split` e  $K$ -Fold (*fold* arbitrário) para o conjunto sem pré-processamento e com  $p = 128$ . Eixo  $Y$  em escala logarítmica para ressaltar as variações de densidade. Fonte: Elaborado pelo autor.

Os resultados da Figura 14 confirmam a preservação bimodal dos conjuntos de dados, mesmo com severas reduções das amostras (valores de  $p \geq 128$ , por exemplo). Nota-se claramente a mudança na “resolução” das distribuições nas Figuras 14 (c) e (d), que podem ser resolvidas, simplesmente, com mais amostras, ou seja, com mais tempo para que os dados cheguem na “resolução” das Figuras 14 (a) e (b). Conforme descrito anteriormente, os resultados originalmente publicados em (PASQUINI; STADLER, 2017) poderiam suportar valores maiores de  $p$  para os experimentos propostos neste trabalho, caso os conjuntos de dados coletados apresentassem duração superior às dez horas publicamente disponibilizadas no repositório.

## 5 Conclusão

Este trabalho partiu do problema da sobrecarga que o monitoramento pode provocar na infraestrutura de um sistema que utiliza orquestração baseada em aprendizado de máquina. A proposta era encontrar técnicas capazes de reduzir a quantidade de amostras total de um modelo e ainda mantê-lo preciso o suficiente para avaliar a QoS do cliente. Baseando-se na parametrização de modelos utilizada pela literatura ([PASQUINI; STADLER, 2017](#)), três técnicas foram avaliadas de forma a reduzir um conjunto de dados disponibilizado ([PASQUINI, 2017](#)): a eliminação de amostras entre os períodos reportados; a agregação em janelas; e a descrição da janela de cada métrica por meio de 8 novas características.

O Experimento 1 avaliou a eliminação de amostras e os resultados permitiram observar uma intrínseca relação da frequência de coleta com a quantidade de características disponíveis. Para  $p \leq 16$ , observou-se uma boa relação métricas/característica, mas na medida em que os valores de  $p$  aumentavam, o modelo treinado com todas as métricas apresentava uma degradação da previsão. Enquanto isso, para  $p \geq 64$  os resultados foram uniformes quando utilizada a técnica de seleção de características. Infere-se, pela *Manifold Hypothesis*, que o *Univariate Feature Selection* permitiu balancear a quantidade de métricas para uma quantidade menor de amostras, portanto a mudança no parâmetro  $p$ , que reduz a quantidade de amostras, deve estar acompanhada da seleção de características.

A agregação em janelas proposta pelo Experimento 2 dá a informação de como o sistema se comportou desde o último valor reportado, ao contrário do primeiro que apenas elimina as amostras. Em comparação ao Experimento 1, o comportamento dos modelos treinados com a integralidade das métricas era mais uniforme, e alcançou um intervalo maior de  $j \leq 64$ , anteriormente 16. Isso indica que o equilíbrio da relação das amostras com a quantidade de características é mantido com menos amostras neste cenário. Porém, nem com a utilização de seleção de características os resultados para  $j > 64$  tiveram bom desempenho e concluiu-se que a utilização das funções podem distorcer a amostragem que originalmente seria bimodal para as características com maior relação com  $Y$ .

O Experimento 3, assim como o segundo, busca fornecer informações de cada métrica no intervalo desde a última reportagem, porém com 8 novas métricas descritivas para cada característica original. Os resultados sem a utilização de seleção de características foram satisfatórios apenas para  $j \leq 32$ , onde teve melhor equilíbrio características/amostragem, enquanto a utilização do *Univariate Feature Selection* permitiu alcançar  $j \leq 128$ . Pela *Manifold Hypothesis*, conclui-se que o acréscimo das novas estatísticas impactam os modelos na medida em que  $j$  aumenta porque muda a relação da quantidade de

amostras disponíveis com a quantidade de características existentes, mas quando submetido à seleção de características é possível selecionar aquelas que ainda guardam a melhor relação com os dois estados principais encontrados na distribuição bimodal de  $Y$ .

Existiram resultados iguais ou melhores à literatura, principalmente para os primeiros níveis de raleamento das amostras. Isso alinha com o observado por Baig et al. (2020), já que algumas amostras ruidosas foram processadas ou eliminadas pela etapa de pré-processamento e, por consequência, a qualidade das estimativas dos modelos melhora.

Foi possível atestar a diferença da utilização de uma técnica de validação cruzada para os casos de menor volume de dados, em que mantinha valores mais consistentes que o particionamento convencional do *train\_test\_split*. Além disso, o Experimento 4 permitiu analisar e ressaltar que os parâmetros encontrados por este trabalho correspondem ao pré-processamento realizado em um experimento que executou por mais de 10 horas. Dessa forma, os resultados indicam que o resultado obtido com uma janela pode melhorar na medida em que mais amostras são apresentadas na mesma granularidade. Ainda, a observação do equilíbrio entre a quantidade de características e a quantidade de amostras permite que sistemas adaptativos de monitoramento possam iniciar o treinamento de modelos com uma quantidade maior de amostras, mas gradualmente seria possível atualizá-lo com menor frequência (embora na mesma granularidade).

Conclui-se que o objetivo deste trabalho foi alcançado: é possível utilizar técnicas para reduzir a frequência de reportagem das métricas por um sistema de monitoramento. É possível construir modelos com até 128 segundos de tempo entre reportagens quando aliado com seleção de características, o que representa uma redução de 16 vezes o tamanho original, e *Random Forest* para casos mais complexos, ou 64 segundos com pré-processamento com janelas, com redução também de 64 vezes. Isso implica que os nós de um sistema de orquestração não precisam reportar a todo instante e impactar o tráfego da infraestrutura.

Entretanto, é importante ressaltar que o objetivo deste trabalho não era de refutar a utilidade de modelos com granularidade por segundo, mas propor uma alternativa que utilizasse menos recursos da rede com uma precisão ainda relevante. Isso possibilita a combinação de diferentes granularidades para avaliar a QoS em diferentes pontos do sistema, considerando o *tradeoff* de desempenho e precisão do intervalo utilizado, assim como abre discussão para uma abordagem híbrida onde, mesmo com uma dada granularidade, é possível treinar um modelo com uma certa quantidade de amostras e em seguida o fornecimento de novas amostras teria o mero objetivo de atualizá-lo, mas com menor frequência.

Este trabalho tem como limitação o fato de que não repetiu o experimento original sob as diferentes técnicas de coleta e transporte das métricas. Os experimentos propostos por este trabalho foram aplicação de métodos de pré-processamento em um

momento posterior à execução dos experimentos originais, com os dados disponibilizados pela literatura ([PASQUINI; STADLER, 2017](#)). Outra limitação foi que, embora a técnica de particionamento tenha sido a Validação Cruzada, ainda seria possível utilizar técnicas mais adequadas para séries temporais, as quais não foram utilizadas para manter o mesmo comportamento que [Pasquini e Stadler \(2017\)](#) e pelo fato de que as amostras representam os diferentes possíveis estados de carga da aplicação, independentemente da causalidade temporal.

Como destacado pela Figura 4, este trabalho concentrou-se em alterações no sistema de monitoramento para auxiliar o de orquestração. Para trabalhos futuros, é esperada a implementação das técnicas propostas nos sistemas de monitoramento ainda nos nós fonte de coleta das métricas, alternativamente sugere-se: 1) a utilização de *autoencoders* como alternativa às métricas descritivas do Experimento 3, que seriam capazes de encontrar uma quantidade arbitrária de parâmetros ([SHAHABSAMANI, 2024](#)), ela substituiria a definição das 8 estatísticas por este trabalho para representações intermediárias; 2) janelas de tamanho dinâmico: possibilidade de redução de *overfitting* ([BAIG et al., 2020](#)) ao condensar amostras similares e melhorar a relação de características e quantidade de amostras; 3) utilizar janelas para o conjunto  $Y$ , baseada na solução pragmática de que a QoS não depende de estimar uma métrica a cada segundo, mas sim em estimar se os requisitos concordados são cumpridos. Seria implementada por meio de uma janela que identificasse se a qualidade mínima foi garantida em um intervalo de tempo; e 4) implementação de um sistema adaptativo de orquestração baseado na premissa de que é possível treinar com mais amostras inicialmente e diminuir gradualmente a frequência de atualização do modelo, embora em uma mesma granularidade.

Os resultados e técnicas utilizadas estão disponibilizadas publicamente ([VILARINHO, 2025](#)) com o intuito de permitir a reproduzibilidade dos experimentos realizados e facilitar trabalhos futuros.

# Referências

- AHMED, J.; NASCIMENTO, A. D.; KILINC, C.; PAN, D.; RIU, J. R. I.; GUSTAFSSON, J. Using blackbox ml techniques to diagnose qoe problems for an iptv service. In: **NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium**. Budapest, Hungary: IEEE, 2020. p. 1–8. ISBN 978-1-7281-4973-8. Disponível em: <<https://ieeexplore.ieee.org/document/9110375/>>. Citado na página 18.
- BAIG, S.-u.-R.; IQBAL, W.; BERRAL, J. L.; CARRERA, D. Adaptive sliding windows for improved estimation of data center resource utilization. **Future Generation Computer Systems**, v. 104, p. 212–224, mar. 2020. ISSN 0167-739X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167739X19309203>>. Citado 6 vezes nas páginas 12, 22, 23, 26, 49 e 50.
- BREIMAN, L. Random Forests. **Machine Learning**, v. 45, n. 1, p. 5–32, out. 2001. ISSN 1573-0565. Peguei do Ian. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>. Citado 3 vezes nas páginas 11, 16 e 17.
- DOMINGOS, P. A few useful things to know about machine learning. **Communications of the ACM**, v. 55, n. 10, p. 78–87, out. 2012. ISSN 0001-0782, 1557-7317. Disponível em: <<https://dl.acm.org/doi/10.1145/2347736.2347755>>. Citado na página 23.
- GODARD, S. **sar(1) - Linux man page**. [S.l.], 2007. Disponível em: <<https://linux.die.net/man/1/sar>>. Citado 3 vezes nas páginas 18, 19 e 25.
- HASTIE, T.; FRIEDMAN, J.; TIBSHIRANI, R. **The Elements of Statistical Learning**. New York, NY: Springer, 2001. (Springer Series in Statistics). ISBN 978-1-4899-0519-2 978-0-387-21606-5. Disponível em: <<http://link.springer.com/10.1007/978-0-387-21606-5>>. Citado 2 vezes nas páginas 22 e 23.
- IOSTAT(1) - Linux man page. [S.l.], 2023. Disponível em: <<https://linux.die.net/man/1/iostat>>. Citado na página 18.
- JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R.; TAYLOR, J. **An introduction to statistical learning: With applications in python**. Springer Nature, 2023. Disponível em: <<https://books.google.com/books?hl=en&lr=&id=ygzJEAAAQBAJ&oi=fnd&pg=PR7&dq=An+introduction+to+statistical+learning:+With+applications+in+python&ots=WzkaoWF4Zn&sig=0YUkd-rZn6Uyd6bvYLQz7ZTDq5s>>. Citado 8 vezes nas páginas 11, 12, 14, 15, 16, 17, 22 e 23.
- KEPHART, J.; CHESS, D. The vision of autonomic computing. **Computer**, v. 36, n. 1, p. 41–50, jan. 2003. ISSN 1558-0814. Conference Name: Computer. Disponível em: <[https://ieeexplore.ieee.org/abstract/document/1160055?casa\\_token=WNI-J1vTqagAAAAA:u4Usn\\_P3Zj5\\_TCN5jomOfqbHH0uDG3k5etInwPGAMo1gEaaJBHRjHazkdOGv\\_qH1-EgC0OAA0gOW](https://ieeexplore.ieee.org/abstract/document/1160055?casa_token=WNI-J1vTqagAAAAA:u4Usn_P3Zj5_TCN5jomOfqbHH0uDG3k5etInwPGAMo1gEaaJBHRjHazkdOGv_qH1-EgC0OAA0gOW)>. Citado 3 vezes nas páginas 10, 17 e 18.
- KUHN, M.; JOHNSON, K. **Applied predictive modeling**. Corrected at 5th printing. New York: Springer, 2016. ISBN 978-1-4614-6848-6. Citado na página 23.

- KUROSE, J. F. **Redes de computadores e a internet: uma abordagem top-down.** [S.l.]: Pearson, 2013. ISBN 978-85-8143-677-7. Citado 2 vezes nas páginas 10 e 20.
- LACERDA, M. **DASH sobre OpenFlow: estimando métricas de QoS a partir da rede.** Tese (Doutorado) — Universidade Federal de Uberlândia, fev. 2020. Disponível em: <<https://repositorio.ufu.br/handle/123456789/30457>>. Citado na página 20.
- LACSON, R.; ESKIAN, M.; LICAROS, A.; KAPOOR, N.; KHORASANI, R. Machine learning model drift: predicting diagnostic imaging follow-up as a case example. **Journal of the American College of Radiology**, Elsevier, v. 19, n. 10, p. 1162–1169, 2022. Citado na página 16.
- MARQUES, G. **Proposta de um sistema inteligente de monitoramento baseado em seleção de características.** Tese (Doutorado) — Universidade Federal de Uberlândia, dez. 2020. Disponível em: <<https://repositorio.ufu.br/handle/123456789/35690>>. Citado na página 29.
- MATOS, J. R. F. Construção de um serviço conteinerizado de vídeo sob demanda baseado em DASH para experimentação e coleta de métricas de desempenho. dez. 2018. Accepted: 2019-01-25T09:56:28Z Publisher: Universidade Federal de Uberlândia. Disponível em: <<https://repositorio.ufu.br/handle/123456789/24068>>. Citado na página 20.
- MELLO, R. F. D.; PONTI, M. A. **Machine Learning: A Practical Approach on the Statistical Learning Theory.** Cham: Springer International Publishing, 2018. ISBN 978-3-319-94988-8 978-3-319-94989-5. Disponível em: <<http://link.springer.com/10.1007/978-3-319-94989-5>>. Citado 2 vezes nas páginas 14 e 16.
- MITCHELL, T. M. **Machine learning.** [S.l.]: McGraw-hill, 1997. Citado na página 16.
- MPEG. **Standards – MPEG.** [S.l.], 2022. Disponível em: <<https://www.mpeg.org/standards/>>. Citado na página 20.
- PASQUINI, R. [rafaelpasquini/traces-netsoft-2017](https://github.com/rafaelpasquini/traces-netsoft-2017). 2017. Original-date: 2017-03-29T08:49:12Z. Disponível em: <<https://github.com/rafaelpasquini/traces-netsoft-2017>>. Citado 5 vezes nas páginas 11, 19, 21, 22 e 48.
- PASQUINI, R.; STADLER, R. Learning end-to-end application QoS from openflow switch statistics. In: **2017 IEEE Conference on Network Softwarization ( NetSoft).** [s.n.], 2017. p. 1–9. Disponível em: <<https://ieeexplore.ieee.org/document/8004198?denied=>>>. Citado 31 vezes nas páginas 5, 6, 11, 12, 15, 16, 17, 18, 20, 21, 23, 24, 25, 26, 27, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48 e 50.
- PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 4 vezes nas páginas 15, 17, 27 e 45.
- REZENDE, A. **Orquestração de cloud-network slices orientada à predição de métricas de serviço a partir do monitoramento da infraestrutura.** Tese

- (Doutorado) — Universidade Federal de Uberlândia, ago. 2020. Disponível em: <<https://repositorio.ufu.br/handle/123456789/32692>>. Citado na página 18.
- SALEHIE, M.; TAHVILDARI, L. Self-adaptive software: Landscape and research challenges. **ACM Transactions on Autonomous and Adaptive Systems**, v. 4, n. 2, p. 1–42, maio 2009. ISSN 1556-4665, 1556-4703. Disponível em: <<https://dl.acm.org/doi/10.1145/1516533.1516538>>. Citado 3 vezes nas páginas 10, 17 e 18.
- SHAHABSAMANI, F. **End-to-end performance prediction and automated resource management of cloud services**. [S.l.]: KTH Royal Institute of Technology, 2024. ISBN 9789180409179. Citado 6 vezes nas páginas 12, 23, 25, 26, 37 e 50.
- SODAGAR, I. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. **IEEE MultiMedia**, v. 18, n. 4, p. 62–67, abr. 2011. ISSN 1941-0166. Conference Name: IEEE MultiMedia. Disponível em: <[https://ieeexplore.ieee.org/abstract/document/6077864?casa\\_token=TtqWe7XP1AkAAAAA:h93s\\_Z71xGyeBAzqEB49Ca2RfD\\_eTISr\\_07G4XZAE8VYkkDmT-epBRzp8VVR\\_CB3C-RXSOnk823](https://ieeexplore.ieee.org/abstract/document/6077864?casa_token=TtqWe7XP1AkAAAAA:h93s_Z71xGyeBAzqEB49Ca2RfD_eTISr_07G4XZAE8VYkkDmT-epBRzp8VVR_CB3C-RXSOnk823)>. Citado na página 20.
- SYSSTAT Home Page. [S.l.], 2023. Disponível em: <<https://sysstat.github.io/>>. Citado na página 18.
- VIDEOLAN. **Download oficial do Reprodutor de Mídias VLC, o melhor reprodutor livre - VideoLAN**. [S.l.], 2024. Disponível em: <<https://www.videolan.org/vlc/>>. Citado na página 19.
- VILARINHO, D. F. R. d. O. **DaviVilarinho/ic-intervalo-qos**. 2025. Original-date: 2025-04-25T00:27:12Z. Disponível em: <<https://github.com/DaviVilarinho/ic-intervalo-qos>>. Citado 3 vezes nas páginas 13, 34 e 50.
- YANGGRATOKE, R.; AHMED, J.; ARDELIUS, J.; FLINTA, C.; JOHNSSON, A.; GILLBLAD, D.; STADLER, R. Predicting real-time service-level metrics from device statistics. In: **2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)**. [s.n.], 2015. p. 414–422. ISSN: 1573-0077. Disponível em: <<https://ieeexplore.ieee.org/abstract/document/7140318>>. Citado 2 vezes nas páginas 20 e 21.