UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Igor Augusto Reis Gomes

Seleção de Atributos para Monitoração de Cloud-Networks: Uma Análise do Compromisso entre Desempenho Preditivo e Complexidade Computacional

Uberlândia, Brasil

2026

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Igor Augusto Reis Gomes

Seleção de Atributos para Monitoração de Cloud-Networks: Uma Análise do Compromisso entre Desempenho Preditivo e Complexidade Computacional

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Rafael Pasquini

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2026

Igor Augusto Reis Gomes

Seleção de Atributos para Monitoração de Cloud-Networks: Uma Análise do Compromisso entre Desempenho Preditivo e Complexidade Computacional

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 27 de março de 2026:

Rafael Pasquini
Orientador

Luis Fernando Faina

Paulo Rodolfo da Silva Leite Coelho

Uberlândia, Brasil
2026

Resumo

A seleção de atributos é uma etapa fundamental na construção de modelos de aprendizado de máquina, especialmente em ambientes de monitoração de infraestruturas de redes e computação em nuvem, nos quais sistemas de orquestração podem coletar centenas ou milhares de métricas simultaneamente. Este trabalho investiga o compromisso entre desempenho preditivo e custo computacional em quatro métodos de seleção de atributos — busca exaustiva, seleção progressiva (*Forward Selection*), eliminação regressiva (*Backward Selection*) e *SelectKBest* — aplicados à predição de métricas de qualidade de serviço (QoS) em *cloud networks*. Para avaliar esse compromisso de forma sistemática, a busca exaustiva é utilizada como referência empírica para o melhor subconjunto de atributos identificado no espaço de busca considerado, enquanto os métodos heurísticos são comparados a esse referencial segundo as métricas MSE e NMAE. Os experimentos envolvem três algoritmos de aprendizado supervisionado — Regressão Linear, Árvore de Decisão e *Random Forest* — cinco configurações de conjuntos de dados e cinco sementes aleatórias distintas. Os resultados indicam que a seleção progressiva frequentemente encontra subconjuntos de atributos com desempenho muito próximo — e, em alguns casos, idêntico ao melhor resultado identificado pela busca exaustiva — para os modelos de Regressão Linear e Árvore de Decisão, apresentando pequena degradação no caso do *Random Forest*, ao mesmo tempo em que reduz significativamente o tempo de execução. A eliminação regressiva apresenta desempenho comparável para modelos lineares e florestas aleatórias, mas mostra maior sensibilidade no caso da Árvore de Decisão. O método *SelectKBest* destaca-se pelo custo computacional mínimo e por resultados adequados para modelos lineares, configurando-se como alternativa prática em cenários de altíssima dimensionalidade. A consistência dos resultados ao longo das diferentes sementes aleatórias reforça a robustez das conclusões experimentais.

Palavras-chave: seleção de atributos. aprendizado de máquina. qualidade de serviço. cloud-networks. busca exaustiva.

Lista de tabelas

Lista de abreviaturas e siglas

ANOVA	*Analysis of Variance* (Análise de Variância)
API	*Application Programming Interface* (Interface de Programação de Aplicações)
CPU	*Central Processing Unit* (Unidade Central de Processamento)
GPU	*Graphics Processing Unit* (Unidade de Processamento Gráfico)
IA	Inteligência Artificial
IoT	*Internet of Things* (Internet das Coisas)
ML	*Machine Learning* (Aprendizado de Máquina)
MSE	*Mean Squared Error* (Erro Quadrático Médio)
NaN	*Not a Number* (Não é um Número)
NMAE	*Normalized Mean Absolute Error* (Erro Absoluto Médio Normalizado)
OLS	*Ordinary Least Squares* (Mínimos Quadrados Ordinários)
QoE	*Quality of Experience* (Qualidade de Experiência)
QoS	*Quality of Service* (Qualidade de Serviço)
RAM	*Random Access Memory* (Memória de Acesso Aleatório)
RMSE	*Root Mean Squared Error* (Raiz do Erro Quadrático Médio)
SDN	*Software-Defined Networking* (Redes Definidas por Software)
SMT	*Simultaneous Multi-Threading* (Multithreading Simultâneo)
VoD	*Video-on-Demand* (Vídeo Sob Demanda)

Sumário

1 Introdução

1.1 Contextualização

A computação em nuvem tem revolucionado a maneira como recursos computacionais são provisionados e gerenciados, oferecendo escalabilidade, flexibilidade e eficiência sob demanda (ARMBRUST et al., 2010). Com o crescimento exponencial dos serviços digitais, as infraestruturas de nuvem tornaram-se progressivamente mais complexas, passando a incorporar milhares de servidores, dispositivos de rede e aplicações distribuídas que devem operar de forma coordenada para garantir níveis adequados de desempenho e disponibilidade (EL-GAZZAR, 2014).

Nesse contexto, o gerenciamento eficiente dessas infraestruturas é fundamental para atender aos requisitos de Qualidade de Serviço (*Quality of Service* – QoS), que englobam parâmetros como largura de banda, latência, taxa de perda de pacotes e disponibilidade (TANENBAUM; WETHERALL, 2011). A manutenção desses parâmetros é especialmente crítica em aplicações sensíveis ao tempo, como *streaming* de vídeo sob demanda, sistemas de armazenamento distribuído e serviços de computação científica, nas quais degradações pontuais podem comprometer diretamente a experiência do usuário ou a corretude dos serviços.

Para viabilizar o monitoramento e o controle dessas infraestruturas complexas, sistemas de orquestração coletam continuamente uma ampla variedade de métricas de desempenho provenientes de diferentes camadas da pilha computacional, incluindo servidores, máquinas virtuais, contêineres, switches de rede e aplicações (BELOGLAZOV; ABAWAJY; BUYYA, 2012). Em ambientes de produção, não é incomum que milhares de métricas sejam coletadas a cada segundo, resultando em volumes de dados que impõem desafios significativos tanto para armazenamento quanto para processamento e transmissão através da rede (STADLER; PASQUINI; FODOR, 2017).

Além disso, a transmissão contínua dessas métricas desde as fontes de coleta até os centros de processamento, onde algoritmos de análise e predição operam, pode consumir recursos expressivos de banda e introduzir latências indesejáveis, sobretudo em arquiteturas de orquestração distribuídas e geograficamente dispersas (PASQUINI; STADLER, 2017). Nesses cenários, a própria infraestrutura de monitoração passa a representar uma fonte adicional de sobrecarga, afetando negativamente a escalabilidade e a eficiência operacional do sistema como um todo.

Diante desse cenário, técnicas de Inteligência Artificial (IA), em especial aquelas baseadas em Aprendizado de Máquina (*Machine Learning*), emergem como ferramentas

centrais para apoiar a gestão e a otimização de ambientes de nuvem (GOODFELLOW; BENGIO; COURVILLE, 2016). Modelos de aprendizado supervisionado, como *Random Forest* e *Decision Trees*, têm demonstrado capacidade de estimar métricas de QoS a partir de estatísticas coletadas da infraestrutura, possibilitando predições acuradas e em tempo quase real (PASQUINI; STADLER, 2017; STADLER; PASQUINI; FODOR, 2017).

Entretanto, a eficácia desses modelos está diretamente associada não apenas à escolha do algoritmo de aprendizado, mas também à qualidade, relevância e quantidade dos dados de entrada utilizados. Nesse contexto, a seleção de atributos (*feature selection*) assume papel fundamental, ao permitir identificar subconjuntos reduzidos de métricas capazes de preservar a qualidade das predições, ao mesmo tempo em que diminuem a dimensionalidade dos dados, o custo computacional de treinamento e inferência, e a sobrecarga imposta pelos processos de coleta e transmissão (GUYON; ELISSEEFF, 2003; CHANDRASHEKAR; SAHIN, 2014).

1.2 Problema de Pesquisa

O problema central abordado neste trabalho consiste em analisar como diferentes métodos de seleção de atributos se comportam diante do compromisso entre desempenho preditivo e complexidade computacional no contexto da monitoração de infraestruturas de redes e computação em nuvem. Em particular, investiga-se em que medida estratégias heurísticas de seleção são capazes de aproximar o desempenho obtido por uma busca exaustiva, considerada ótima do ponto de vista preditivo, sem incorrer no custo computacional proibitivo associado a essa abordagem.

Em ambientes de orquestração de recursos em nuvem, sistemas de monitoramento podem coletar centenas ou milhares de métricas por segundo, incluindo estatísticas de CPU, memória, disco, rede e informações específicas de protocolos como OpenFlow (MCKEOWN et al., 2008). Embora esse conjunto amplo de métricas maximize o potencial informativo disponível, nem todas contribuem de forma relevante para a estimação de métricas de QoS fim-a-fim. A presença de atributos redundantes, irrelevantes ou ruidosos pode degradar o desempenho dos modelos preditivos e aumentar desnecessariamente o custo computacional (JOHN; KOHAVI; PFLEGER, 1994).

Do ponto de vista teórico, a avaliação exaustiva de todos os subconjuntos possíveis de atributos permite identificar o subconjunto ótimo em termos de desempenho preditivo. Entretanto, essa abordagem apresenta complexidade exponencial, uma vez que, para um conjunto com n atributos, existem $2^n - 1$ combinações possíveis, tornando sua aplicação inviável para valores moderados ou elevados de n (CHANDRASHEKAR; SAHIN, 2014). Como consequência, métodos heurísticos tornam-se necessários em cenários práticos, ainda que não garantam a obtenção do ótimo global.

Nesse contexto, surge uma questão fundamental: em que medida soluções heurísticas de seleção de atributos, sujeitas à convergência para ótimos locais, conseguem preservar a qualidade das predições quando comparadas ao desempenho obtido por uma busca exaustiva? Adicionalmente, qual é o ganho efetivo em termos de redução de complexidade computacional e sobrecarga operacional ao se adotar tais heurísticas em ambientes de produção?

Assim, o desafio deste trabalho não se limita à redução da dimensionalidade dos dados, mas envolve a avaliação sistemática do compromisso entre: (i) desempenho preditivo dos modelos de aprendizado de máquina; (ii) complexidade computacional associada aos processos de seleção, treinamento e inferência; e (iii) viabilidade prática de aplicação em sistemas de monitoração de larga escala. A partir dessa análise, busca-se compreender os limites e as vantagens das abordagens heurísticas em relação ao referencial ótimo estabelecido pela busca exaustiva.

1.3 Hipótese e Objetivos

1.3.1 Hipótese

A hipótese central deste trabalho é que métodos heurísticos de seleção de atributos, como *Forward Selection*, *Backward Selection* e técnicas univariadas baseadas em correlação (e.g., *SelectKBest*), são capazes de identificar subconjuntos de atributos que preservam desempenho preditivo próximo àquele obtido por meio de busca exaustiva (análise combinatória completa), ao mesmo tempo em que reduzem de forma significativa o custo computacional associado aos processos de seleção, treinamento e inferência.

1.3.2 Objetivo Geral

Analisar e comparar diferentes métodos de seleção de atributos aplicados à construção de modelos de aprendizado de máquina para predição de métricas de QoS em ambientes de monitoramento de infraestruturas de cloud-networks, investigando o compromisso entre desempenho preditivo e complexidade computacional.

1.3.3 Objetivos Específicos

1. Implementar e validar três abordagens distintas de seleção de atributos: análise combinatória exaustiva (força bruta), utilizada como referencial ótimo, métodos incrementais/decrementais (*Forward* e *Backward Selection*) e métodos baseados em heurísticas de correlação (*SelectKBest*);

2. Avaliar o desempenho dos modelos resultantes utilizando duas métricas complementares: *Mean Squared Error* (MSE) e *Normalized Mean Absolute Error* (NMAE);

3. Conduzir experimentos com múltiplos subconjuntos de dados gerados aleatoriamente a partir de um dataset base, de modo a permitir comparações estatisticamente mais robustas entre os métodos;

4. Comparar os subconjuntos de atributos selecionados pelos diferentes métodos em termos de: (i) desempenho preditivo; (ii) número de atributos selecionados; (iii) tempo computacional requerido; e (iv) sobreposição entre os conjuntos de atributos identificados;

5. Investigar a influência de diferentes algoritmos de aprendizado supervisionado (*Linear Regression*, *Decision Tree* e *Random Forest*) sobre o desempenho dos métodos de seleção de atributos avaliados.

1.4 Justificativa

A relevância deste trabalho fundamenta-se em três pilares principais: científico, prático e econômico, os quais se articulam em torno do desafio de conciliar desempenho preditivo e eficiência computacional em ambientes de monitoração de cloud-networks.

Do ponto de vista científico, a seleção de atributos permanece como um tema ativo de pesquisa em aprendizado de máquina, especialmente em cenários caracterizados por alta dimensionalidade dos dados (CHANDRASHEKAR; SAHIN, 2014). Embora a literatura apresente uma variedade de métodos para seleção de atributos, ainda são escassos estudos comparativos sistemáticos que avaliem, de forma quantitativa, o compromisso entre desempenho preditivo e complexidade computacional desses métodos no domínio de gerenciamento de redes e computação em nuvem. Este trabalho contribui para preencher essa lacuna ao conduzir uma análise empírica estruturada, utilizando a busca exaustiva como referencial ótimo e comparando-a com diferentes estratégias heurísticas aplicadas a dados reais de infraestruturas de rede.

Sob a perspectiva prática, sistemas de orquestração autônomos e adaptativos dependem criticamente de modelos preditivos que sejam não apenas acurados, mas também eficientes do ponto de vista computacional, a fim de apoiar decisões relacionadas à alocação de recursos, balanceamento de carga e detecção proativa de anomalias (MAO et al., 2016; LI et al., 2018). A redução do número de métricas monitoradas, quando realizada de forma criteriosa, pode diminuir a sobrecarga imposta pelos processos de coleta, transmissão e processamento de dados, viabilizando respostas mais rápidas e escaláveis frente à dinâmica da infraestrutura (AL-DHURAIBI et al., 2018).

No aspecto econômico, a otimização do monitoramento por meio da seleção inteligente de atributos pode resultar em economias substanciais de recursos computacionais e de rede. Em ambientes de nuvem de larga escala, nos quais milhares ou milhões de instâncias são monitoradas continuamente, reduções mesmo modestas no volume de dados coletados e processados podem traduzir-se em ganhos expressivos de eficiência operacional e redução de custos (BELOGLAZOV; ABAWAJY; BUYYA, 2012). Esses benefícios estendem-se também ao consumo energético da infraestrutura, alinhando-se a iniciativas contemporâneas de computação verde (*Green Computing*), nas quais a minimização do processamento desnecessário e da transmissão excessiva de dados contribui para a sustentabilidade de data centers.

Por fim, os resultados deste estudo possuem relevância direta para provedores de serviços de nuvem, operadores de rede e desenvolvedores de sistemas de orquestração, ao oferecerem subsídios quantitativos para a escolha de métodos de seleção de atributos mais adequados a diferentes cenários operacionais, considerando restrições de desempenho, complexidade computacional e escalabilidade.

1.4.1 Principais Resultados e Contribuições

Os resultados obtidos indicam que métodos heurísticos de seleção de atributos podem alcançar desempenho preditivo muito próximo ao referencial ótimo estabelecido pela busca exaustiva, ao mesmo tempo em que reduzem de forma significativa o custo computacional.

A *Forward Selection* destacou-se como a estratégia mais equilibrada, convergindo para soluções equivalentes ao ótimo global nos modelos de Regressão Linear e *Decision Tree*, e mantendo desempenho competitivo no *Random Forest*. Além da qualidade preditiva, apresentou expressiva vantagem em eficiência, reduzindo drasticamente o tempo de processamento em comparação com a busca exaustiva, o que evidencia uma relação custo-benefício altamente favorável. O método *SelectKBest*, por sua vez, mostrou excelente desempenho em modelos lineares, combinando rapidez de execução com elevada precisão. Entretanto, sua performance revelou-se menos consistente em modelos não lineares, sugerindo limitações inerentes ao seu critério univariado de seleção, que tende a desconsiderar interações mais complexas entre atributos.

De forma geral, a análise comparativa evidencia um gradiente claro de desempenho: enquanto a busca exaustiva permanece como referência ideal, o *Forward Selection* aproxima-se fortemente desse patamar; o *Backward Selection* apresenta desempenho intermediário; e o *SelectKBest* mostra maior sensibilidade ao tipo de modelo empregado. A consistência dos resultados ao longo de diferentes execuções reforça a robustez das conclusões e mitiga possíveis efeitos de aleatoriedade experimental.

Como principal contribuição, este trabalho demonstra que abordagens heurísticas adequadamente escolhidas podem substituir a busca exaustiva em cenários reais de predição de QoS em *cloud-networks*, preservando desempenho competitivo e ampliando a viabilidade prática em termos de escalabilidade. Adicionalmente, o estudo estabelece um referencial comparativo sistemático entre métodos com diferentes níveis de complexidade computacional, oferecendo subsídios consistentes para a tomada de decisão em pipelines de aprendizado de máquina sob restrições de tempo e recursos.

1.5 Estrutura do Trabalho

O restante deste trabalho está organizado da seguinte forma. O Capítulo 2 apresenta a fundamentação teórica e os trabalhos relacionados, abordando conceitos de monitoração em infraestruturas de cloud-networks, a aplicação de técnicas de inteligência artificial, os métodos de seleção de atributos considerados, os algoritmos de aprendizado de máquina empregados e as métricas de avaliação utilizadas. O Capítulo 3 descreve a metodologia adotada, detalhando o dataset, os procedimentos de pré-processamento, o desenho experimental, as estratégias de seleção de atributos avaliadas e o ambiente computacional. O Capítulo 4 discute os experimentos realizados e analisa os resultados obtidos, incluindo comparações de desempenho preditivo, interpretação dos atributos selecionados e considerações sobre custo computacional. Por fim, o Capítulo 5 apresenta as conclusões do trabalho, destacando suas principais contribuições, limitações e perspectivas para trabalhos futuros. Visando manter o corpo principal do texto mais objetivo, o detalhamento completo das configurações experimentais — incluindo as *features* utilizadas, os diferentes datasets, métodos de seleção, sementes e métricas avaliadas — é apresentado no Anexo A, que passa a ser referenciado ao longo do trabalho sempre que necessário.

2 Fundamentação Teórica

Este capítulo apresenta os fundamentos teóricos que sustentam a metodologia e a análise desenvolvidas neste trabalho. Inicialmente, são discutidos os principais desafios associados à monitoração de infraestruturas de cloud-networks, bem como o uso de técnicas de inteligência artificial nesse contexto. Em seguida, são descritos os métodos de seleção de atributos, os algoritmos de aprendizado de máquina considerados nos experimentos e as métricas de avaliação empregadas para a análise do desempenho preditivo.

2.1 Monitoração em Infraestruturas de Cloud-Network

A computação em nuvem consolidou-se como paradigma dominante para o provisionamento de recursos computacionais, transformando fundamentalmente a maneira como organizações gerenciam e operam suas infraestruturas de TI (ARMBRUST et al., 2010). Nesse contexto, as infraestruturas de cloud-networks caracterizam-se pela integração de recursos computacionais virtualizados, redes definidas por software (*Software-Defined Networking* – SDN) e sistemas de orquestração, resultando em ambientes altamente dinâmicos e complexos.

A monitoração dessas infraestruturas desempenha papel central na manutenção de níveis adequados de Qualidade de Serviço (QoS), ao permitir a detecção proativa de anomalias, o planejamento de capacidade e a tomada de decisões informadas sobre alocação de recursos (BELOGLAZOV; ABAWAJY; BUYYA, 2012). Sistemas de monitoramento contemporâneos coletam continuamente métricas de desempenho provenientes de múltiplas camadas da pilha tecnológica, incluindo estatísticas de hardware (CPU, memória, disco e interfaces de rede) e métricas de aplicação, como taxa de requisições, latência e taxa de erro.

Entretanto, a monitoração em larga escala impõe desafios significativos. Em ambientes de produção, milhares de métricas podem ser coletadas a cada segundo a partir de diferentes componentes da infraestrutura, resultando em grandes volumes de dados a serem armazenados, transmitidos e processados. A transmissão dessas métricas desde fontes de coleta distribuídas até centros de processamento pode consumir recursos expressivos de banda de rede e introduzir latências indesejáveis, enquanto o armazenamento e processamento contínuo desses dados acarretam custos computacionais e financeiros consideráveis, especialmente em cenários que demandam retenção de longo prazo para análises históricas e planejamento de capacidade.

Esses desafios são frequentemente descritos por meio dos chamados “três Vs” da

gestão de dados (LANEY, 2001): Volume, associado à quantidade de dados gerados; Velocidade, relacionada à taxa de geração e à necessidade de processamento em tempo quase real; e Variedade, referente à diversidade de tipos e formatos das métricas coletadas.

Nesse cenário, a identificação e a seleção de métricas verdadeiramente relevantes para a predição de QoS emergem como estratégias essenciais para a otimização do processo de monitoramento, ao possibilitar a redução da sobrecarga de coleta, transmissão e processamento sem comprometer a capacidade de inferência sobre o estado e o comportamento do sistema.

2.2 Inteligência Artificial Aplicada à Monitoração

A aplicação de técnicas de Inteligência Artificial, em especial aquelas baseadas em aprendizado de máquina, à monitoração e ao gerenciamento de infraestruturas de nuvem tem sido amplamente investigada na literatura recente (MAO et al., 2016; LI et al., 2018). Essas abordagens permitem a construção de modelos orientados a dados, capazes de capturar padrões complexos a partir de observações históricas, viabilizando a predição de métricas de Qualidade de Serviço (QoS) e o suporte à tomada de decisões em sistemas de orquestração autônomos.

Em contraste com abordagens baseadas em modelagem analítica, que exigem conhecimento detalhado das interações entre os diversos componentes do sistema e frequentemente resultam em modelos de alta complexidade, métodos baseados em aprendizado de máquina adotam um paradigma orientado a dados. Nesse contexto, o comportamento do sistema é inferido a partir de exemplos observados, sem a necessidade de especificação explícita de regras ou equações que descrevam as relações entre variáveis.

A predição de métricas de QoS fim-a-fim a partir de estatísticas de infraestrutura pode ser formulada como um problema de aprendizado supervisionado, no qual modelos são treinados a partir de pares de entrada-saída conhecidos, compostos por métricas coletadas da infraestrutura como variáveis de entrada (X) e métricas de QoS observadas como variáveis de saída (Y). Essa formulação tem sido amplamente adotada em estudos sobre monitoração inteligente de sistemas distribuídos e de computação em nuvem.

Contudo, a eficácia de modelos de aprendizado de máquina está fortemente associada à qualidade e à relevância dos dados de entrada utilizados. A presença de atributos irrelevantes ou redundantes pode comprometer o desempenho dos modelos, fenômeno conhecido como *curse of dimensionality* (maldição da dimensionalidade) (CHANDRASHEKAR; SAHIN, 2014). Em espaços de alta dimensionalidade, os dados tendem a se tornar esparsos, a noção de distância perde poder discriminativo e a quantidade de amostras necessárias para um treinamento adequado cresce exponencialmente com o número de dimensões.

Além de impactar a qualidade das predições, a alta dimensionalidade afeta diretamente o custo computacional de treinamento e inferência. Modelos que utilizam centenas ou milhares de atributos de entrada demandam maior capacidade de processamento e memória, aumentando o tempo de resposta e os custos operacionais. Nesse cenário, a redução da dimensionalidade por meio de técnicas de seleção de atributos emerge como etapa fundamental para o desenvolvimento de soluções práticas, eficientes e escaláveis de monitoração inteligente em infraestruturas de nuvem.

2.3 Seleção de Atributos (Feature Selection)

A seleção de atributos (*feature selection*) constitui etapa fundamental no desenvolvimento de modelos de aprendizado de máquina, especialmente em cenários caracterizados por alta dimensionalidade dos dados e restrições de custo computacional (GUYON; ELISSEEFF, 2003). Seu objetivo é identificar subconjuntos de características capazes de preservar o desempenho preditivo dos modelos, ao mesmo tempo em que reduzem a complexidade associada aos processos de treinamento e inferência.

A relevância da seleção de atributos fundamenta-se em três aspectos principais. Primeiramente, a presença de atributos irrelevantes ou ruidosos pode degradar o desempenho preditivo dos modelos, introduzindo padrões espúrios que comprometem a capacidade de generalização (JOHN; KOHAVI; PFLEGER, 1994). Em segundo lugar, atributos redundantes, altamente correlacionados entre si, não agregam informação adicional ao modelo e aumentam desnecessariamente sua complexidade. Por fim, em cenários práticos de monitoração de infraestruturas de cloud-networks, cada atributo coletado implica custos associados à aquisição, transmissão e armazenamento de dados, tornando essencial a identificação de um conjunto mínimo de métricas suficiente para a obtenção de predições acuradas.

2.3.1 Taxonomia de Métodos

A literatura apresenta diversas taxonomias para classificação de métodos de seleção de atributos. A classificação mais amplamente adotada, proposta por Guyon e Elisseeff (2003) e consolidada por Chandrashekar e Sahin (2014), categoriza os métodos em três famílias principais: *filter*, *wrapper* e *embedded*.

Os métodos *filter* avaliam a relevância de cada atributo independentemente do algoritmo de aprendizado que será utilizado posteriormente, baseando-se exclusivamente em propriedades estatísticas dos dados. Essas técnicas constroem rankings de atributos utilizando medidas como correlação, ganho de informação, estatística qui-quadrado ou coeficientes de determinação. A principal vantagem dos métodos *filter* reside em sua eficiência computacional e independência do modelo preditivo, permitindo aplicação rápida

mesmo em conjuntos de dados de grande dimensionalidade. Por outro lado, ao não considerar as interações entre atributos nem as características específicas do algoritmo de aprendizado, podem resultar em seleções subótimas.

Os métodos *wrapper* utilizam o próprio algoritmo de aprendizado como função de avaliação, treinando modelos com diferentes subconjuntos de atributos e selecionando aquele que produz melhor desempenho preditivo. Essa abordagem tende a produzir seleções mais acuradas e específicas para o modelo em questão, capturando interações entre atributos relevantes para a tarefa. Entretanto, o custo computacional é substancialmente maior, proporcional ao número de subconjuntos avaliados multiplicado pelo custo de treinamento do modelo. Os métodos *stepwise* (incremental e decremental) constituem instâncias importantes dessa categoria.

Por fim, os métodos *embedded* integram a seleção de atributos diretamente no processo de treinamento do modelo, representando um compromisso entre as abordagens *filter* e *wrapper*. Exemplos incluem técnicas de regularização (Lasso, Ridge, Elastic Net) que penalizam coeficientes associados a atributos menos relevantes, e algoritmos baseados em árvores que calculam medidas de importância de atributos durante a construção das estruturas de decisão.

2.3.2 Análise Combinatória Exaustiva

A análise combinatória exaustiva, também conhecida como busca completa ou *brute force*, consiste na avaliação sistemática de todos os subconjuntos possíveis de atributos. Para um conjunto com n atributos, existem $2^n - 1$ subconjuntos não-vazios a serem considerados (CHANDRASHEKAR; SAHIN, 2014). Este método garante a identificação do subconjunto ótimo global, isto é, aquele que maximiza (ou minimiza) a função objetivo escolhida, seja ela acurácia preditiva, erro quadrático médio, ou outra métrica de desempenho.

Matematicamente, o problema de seleção ótima de atributos pode ser formulado como:

$$S^* = \underset{S \subseteq \{1,2,\ldots,n\}}{\arg\min} \; E(M_S),$$

onde S^* representa o subconjunto ótimo, M_S denota o modelo treinado com os atributos em S, e $E(\cdot)$ é a função de erro ou perda em um conjunto de validação.

A complexidade computacional da busca exaustiva, $O(2^n \cdot C)$, onde C representa o custo de treinamento e avaliação do modelo, torna essa abordagem impraticável para valores moderados ou elevados de n. Por exemplo, para $n = 20$ atributos, seria necessário treinar e avaliar mais de um milhão de modelos; para $n = 30$, esse número ultrapassa um bilhão. Consequentemente, a busca exaustiva é viável apenas para conjuntos pequenos

de atributos (tipicamente $n < 15$) ou quando utilizada como referência para comparação com métodos heurísticos em estudos controlados.

Apesar de sua inviabilidade prática em cenários reais, a análise combinatória exaustiva serve como *baseline* teórico, estabelecendo um limite superior de desempenho que métodos heurísticos aspiram aproximar com custo computacional reduzido.

2.3.3 Forward Stepwise Selection

O *Forward Stepwise Selection* (seleção incremental passo-a-passo) constitui método heurístico de seleção de atributos da família *wrapper*, caracterizado por estratégia construtiva *bottom-up* (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). O algoritmo inicia com um conjunto vazio de atributos e, em cada iteração, adiciona ao conjunto o atributo que, dentre os ainda não selecionados, produz maior melhoria no desempenho do modelo.

Formalmente, o processo pode ser descrito como:

1. Inicializar $S_0 = \emptyset$ (conjunto vazio de atributos selecionados)
2. Para $k = 1, 2, \ldots$ até critério de parada:
   a) Para cada atributo $j \notin S_{k-1}$:
      - Treinar modelo $M_{S_{k-1} \cup \{j\}}$
      - Calcular erro $E_{S_{k-1} \cup \{j\}}$ em conjunto de validação

   b) Selecionar $j^* = \arg\min_j E_{S_{k-1} \cup \{j\}}$

   c) $S_k = S_{k-1} \cup \{j^*\}$

   d) Se melhoria no erro $< \epsilon$ (limiar predefinido), parar
3. Retornar S_k

A complexidade temporal do *Forward Selection*, no pior caso, é $O(n^2 \cdot C)$, onde n é o número total de atributos e C o custo de treinamento do modelo. Esta complexidade deriva do fato de que, na k-ésima iteração, $(n - k + 1)$ modelos são treinados, resultando em aproximadamente $\frac{n(n+1)}{2}$ avaliações no total.

Comparado à busca exaustiva, o *Forward Selection* oferece redução dramática de complexidade, de exponencial para quadrática. Entretanto, essa eficiência vem ao custo de não garantir a identificação do subconjunto ótimo global. O método é suscetível ao fenômeno de ótimo local: uma vez que um atributo é adicionado ao conjunto, ele não é mais reavaliado, mesmo que sua relevância diminua com a inclusão de outros atributos posteriormente. Ademais, o método não captura adequadamente situações onde a relevância de um atributo se manifesta apenas em combinação com outros (*feature interactions*).

Trabalhos como o de Pasquini e Stadler (2017) demonstraram a eficácia prática do *Forward Selection* no contexto de estimação de QoS em redes OpenFlow, obtendo subconjuntos reduzidos (tipicamente 5-12 atributos de um total de centenas) com desempenho preditivo comparável ao do conjunto completo.

2.3.4 Backward Stepwise Selection

O *Backward Stepwise Selection* (seleção decremental passo-a-passo) adota estratégia complementar ao método incremental, seguindo abordagem destrutiva *top-down* (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). O algoritmo inicia com o conjunto completo de atributos e, iterativamente, remove o atributo cuja exclusão resulta em menor degradação (ou maior melhoria) no desempenho do modelo.

O procedimento formal é descrito como:

1. Inicializar $S_0 = \{1, 2, \ldots, n\}$ (conjunto completo de atributos)
2. Para $k = 1, 2, \ldots$ até critério de parada:
   a) Para cada atributo $j \in S_{k-1}$:
      - Treinar modelo $M_{S_{k-1}\setminus\{j\}}$
      - Calcular erro $E_{S_{k-1}\setminus\{j\}}$ em conjunto de validação

   b) Selecionar $j^* = \arg\min_j E_{S_{k-1}\setminus\{j\}}$

   c) $S_k = S_{k-1} \setminus \{j^*\}$

   d) Se degradação no erro $> \epsilon$ (limiar predefinido), parar
3. Retornar S_k

A complexidade temporal do *Backward Selection* é equivalente ao método incremental, $O(n^2 \cdot C)$, com a diferença de que as primeiras iterações, que trabalham com conjuntos maiores de atributos, tendem a ser computacionalmente mais custosas que as iterações finais do *Forward Selection*.

Uma vantagem potencial do *Backward Selection* é sua capacidade de considerar interações entre atributos desde o início, dado que parte do modelo completo. Atributos que são individualmente fracos mas relevantes em combinação têm maior probabilidade de serem preservados. Por outro lado, este método requer que seja possível treinar um modelo com todos os n atributos simultaneamente, o que pode ser problemático quando n é muito grande relativamente ao número de amostras disponíveis, situação comum em problemas de alta dimensionalidade onde $n \gg m$ (número de atributos maior que número de amostras).

Na prática, a escolha entre *Forward* e *Backward Selection* depende das características do problema. Para conjuntos de dados onde se espera que apenas uma pequena fração dos atributos seja relevante, o método incremental tende a ser mais eficiente. Quando se acredita que a maioria dos atributos contém informação útil, ou quando interações complexas são esperadas, o método decremental pode ser preferível, desde que a dimensionalidade inicial permita seu uso.

2.3.5 Métodos Univariados: SelectKBest

Os métodos univariados de seleção de atributos pertencem à família *filter*, avaliando cada atributo individualmente em relação à variável alvo, sem considerar possíveis interações entre atributos (GUYON; ELISSEEFF, 2003). O *SelectKBest* constitui abordagem simples e computacionalmente eficiente dessa categoria, selecionando os k atributos que apresentam maiores escores segundo uma função de pontuação estatística predefinida.

Para problemas de regressão, função de pontuação comumente empregada é a estatística F da análise de variância univariada (ANOVA), denotada por `f_regression` em implementações como scikit-learn (PEDREGOSA et al., 2011). Para cada atributo X_j, ajusta-se um modelo de regressão linear simples:

$$Y = \beta_0 + \beta_1 X_j + \epsilon,$$

e calcula-se a estatística F (MONTGOMERY; PECK; VINING, 2012):

$$F_j = \frac{SS_{\text{reg}}/1}{SS_{\text{res}}/(m-2)} = \frac{m-2}{1} \cdot \frac{SS_{\text{reg}}}{SS_{\text{res}}},$$

onde SS_{reg} é a soma de quadrados da regressão (variância explicada pelo modelo), SS_{res} é a soma de quadrados dos resíduos (variância não explicada), e m é o número de amostras. Valores elevados de F_j indicam forte associação linear entre X_j e Y.

O procedimento *SelectKBest* ordena todos os atributos em ordem decrescente de seus escores F_j e seleciona os k primeiros. A complexidade computacional é $O(n \cdot m)$ para calcular todos os escores, mais $O(n \log n)$ para ordenação, resultando em $O(n \cdot m + n \log n)$, substancialmente inferior aos métodos *wrapper*.

As principais vantagens do *SelectKBest* incluem: (i) eficiência computacional, permitindo aplicação rápida mesmo em conjuntos de alta dimensionalidade; (ii) simplicidade de implementação e interpretação; (iii) independência do modelo preditivo final, possibilitando uso como etapa de pré-processamento genérica.

As limitações são igualmente significativas: (i) a avaliação univariada ignora completamente interações entre atributos, podendo descartar variáveis individualmente fracas mas coletivamente importantes; (ii) pode selecionar múltiplos atributos redundantes,

altamente correlacionados entre si; (iii) a escolha do parâmetro k é arbitrária e pode requerer validação cruzada para otimização; (iv) a premissa de relações lineares (no caso de `f_regression`) pode ser inadequada para dados com associações não-lineares.

Apesar dessas limitações, métodos univariados como *SelectKBest* frequentemente servem como *baseline* de referência em estudos comparativos e como técnica de pré-seleção, reduzindo drasticamente a dimensionalidade antes da aplicação de métodos mais sofisticados (CHANDRASHEKAR; SAHIN, 2014).

2.4 Algoritmos de Aprendizado de Máquina Utilizados

Este trabalho emprega três algoritmos de aprendizado supervisionado para tarefas de regressão — Regressão Linear, *Decision Tree* e *Random Forest* — escolhidos por representarem paradigmas de modelagem distintos e apresentarem comportamentos complementares frente à seleção de atributos. Esses algoritmos diferem em termos de complexidade computacional, interpretabilidade e capacidade de capturar relações não-lineares, tornando-os adequados para a análise comparativa proposta neste trabalho.

2.4.1 Regressão Linear

A Regressão Linear constitui um dos métodos estatísticos mais fundamentais e amplamente utilizados para modelagem de relações entre variáveis (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Dada uma matriz de entrada $\mathbf{X} \in \mathbb{R}^{m \times n}$ com m amostras e n atributos, e um vetor alvo $\mathbf{y} \in \mathbb{R}^m$, o modelo linear assume que a relação entre entrada e saída pode ser aproximada por uma combinação linear:

$$\hat{y} = \beta_0 + \sum_{j=1}^{n} \beta_j x_j = \beta_0 + \mathbf{x}^T \boldsymbol{\beta},$$

onde β_0 é o intercepto, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_n]^T$ é o vetor de coeficientes (pesos), e $\mathbf{x}$ é o vetor de atributos de entrada.

Os parâmetros do modelo são tipicamente estimados pelo método dos mínimos quadrados ordinários (*Ordinary Least Squares* – OLS), que minimiza a soma dos quadrados dos resíduos:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{m} (y_i - \hat{y}_i)^2 = \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2.$$

A solução analítica, quando $\mathbf{X}^T\mathbf{X}$ é inversível, é dada por:

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}.$$

As principais vantagens da Regressão Linear incluem: (i) simplicidade conceitual e facilidade de implementação; (ii) interpretabilidade direta dos coeficientes, que indicam a contribuição marginal de cada atributo; (iii) eficiência computacional, com complexidade $O(n^2m + n^3)$ para o cálculo exato; (iv) existência de vasta teoria estatística sobre propriedades dos estimadores, intervalos de confiança e testes de hipóteses.

As limitações fundamentais residem na premissa de linearidade: o modelo assume que a relação entre entrada e saída é linear, o que pode ser inadequado para fenômenos complexos. Ademais, o método é sensível a multicolinearidade (alta correlação entre atributos), que pode tornar a matriz $\mathbf{X}^T\mathbf{X}$ próxima de singular, resultando em estimativas instáveis dos coeficientes. A presença de *outliers* também pode influenciar desproporcionalmente o ajuste, dado que o critério de mínimos quadrados penaliza erros quadraticamente.

2.4.2 Decision Tree

Decision Trees para regressão (*Regression Trees*) representam paradigma não-paramétrico que particiona recursivamente o espaço de entrada em regiões disjuntas, atribuindo a cada região um valor de predição constante (BREIMAN et al., 1984; QUINLAN, 1986). A estrutura resultante assemelha-se a uma árvore binária, onde nós internos representam testes sobre atributos, arestas representam resultados dos testes, e folhas representam predições.

Formalmente, o algoritmo constrói um modelo da forma:

$$\hat{y} = \sum_{k=1}^{K} c_k \cdot \mathbb{I}(\mathbf{x} \in R_k),$$

onde K é o número de regiões (folhas), R_k é a k-ésima região do espaço de entrada, c_k é o valor de predição para a região R_k (tipicamente a média dos valores alvo das amostras em R_k), e $\mathbb{I}(\cdot)$ é a função indicadora.

O processo de construção emprega estratégia gulosa recursiva, selecionando em cada nó o atributo e o limiar que produzem a melhor divisão dos dados segundo um critério de impureza (BREIMAN et al., 1984). Para problemas de regressão, o critério comumente empregado é a variância ou o erro quadrático médio. Especificamente, busca-se minimizar:

$$\sum_{i:\mathbf{x}_i \in R_L} (y_i - \bar{y}_L)^2 + \sum_{i:x_i \in R_R} (y_i - \bar{y}_R)^2,$$

onde R_L e R_R são as regiões esquerda e direita resultantes da divisão, e $\bar{y}_L$, $\bar{y}_R$ são as médias dos valores alvo em cada região.

As *Decision Trees* apresentam diversas vantagens práticas: (i) capacidade de modelar relações não-lineares e interações complexas entre atributos sem necessidade de especificação explícita; (ii) interpretabilidade visual, com a estrutura da árvore fornecendo insights sobre quais atributos e limiares são mais discriminativos; (iii) robustez a transformações monotônicas dos atributos, não requerendo normalização; (iv) capacidade de lidar nativamente com atributos categóricos e numéricos; (v) seleção automática de atributos relevantes, dado que apenas atributos que melhoram as divisões são incorporados.

Por outro lado, árvores individuais sofrem de alta variância: pequenas perturbações nos dados de treinamento podem resultar em estruturas significativamente diferentes. Além disso, são propensas a *overfitting*, especialmente quando crescem profundas e complexas, memorizando ruído nos dados de treinamento. A partição recursiva em regiões retangulares pode ser ineficiente para aproximar fronteiras de decisão suaves ou diagonais no espaço de entrada.

2.4.3 Random Forest

Random Forest, proposto por Breiman (2001), constitui método de aprendizado em conjunto (*ensemble learning*) que combina múltiplas *Decision Trees* para produzir predições mais robustas e acuradas. O algoritmo baseia-se em dois princípios fundamentais: *bagging* (abreviação de *bootstrap aggregating*) e aleatorização da seleção de atributos.

O procedimento de treinamento pode ser sumarizado como:

1. Para $t = 1, 2, \ldots, T$ (onde T é o número de árvores na floresta):
   a) Gerar amostra *bootstrap* $\mathcal{D}_t$ de tamanho m com reposição do conjunto de treinamento original
   b) Construir *Decision Tree* h_t utilizando $\mathcal{D}_t$, com a seguinte modificação:
      - Em cada nó, em vez de considerar todos os n atributos para divisão, selecionar aleatoriamente um subconjunto de m_{try} atributos (m_{try} tipicamente definido como $\sqrt{n}$ em classificação ou $n/3$ em regressão) e escolher a melhor divisão apenas entre esses
2. A predição final para uma nova amostra $\mathbf{x}$ é a média das predições individuais:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} h_t(\mathbf{x})$$

A combinação de *bagging* e aleatorização de atributos induz diversidade entre as árvores da floresta, reduzindo a correlação entre elas. Este mecanismo é fundamental para a eficácia do método, dado que a variância de uma média de T variáveis aleatórias com variância σ^2 e correlação ρ é:

$$\text{Var}\left(\frac{1}{T}\sum_{t=1}^{T} h_t\right) = \rho\sigma^2 + \frac{1-\rho}{T}\sigma^2.$$

Conforme T aumenta, o segundo termo tende a zero, mas o primeiro permanece. Portanto, reduzir ρ através da decorrelação das árvores é essencial para diminuir a variância do conjunto.

As principais vantagens do *Random Forest* incluem: (i) redução substancial de variância comparado a árvores individuais, resultando em menor propensão a *overfitting*; (ii) manutenção da capacidade de modelar não-linearidades e interações complexas; (iii) fornecimento de medidas de importância de atributos, derivadas da redução média de impureza proporcionada por cada atributo em todas as árvores; (iv) desempenho competitivo com métodos estado-da-arte em ampla variedade de domínios, com pouco ou nenhum ajuste de hiperparâmetros (BREIMAN, 2001).

As desvantagens incluem: (i) perda de interpretabilidade em relação a árvores individuais, dado que o modelo consiste em centenas ou milhares de estruturas; (ii) maior custo computacional de treinamento e inferência, proporcional ao número de árvores; (iii) potencial ineficiência em dados de muito alta dimensionalidade com atributos majoritariamente irrelevantes.

2.5 Métricas de Avaliação

A avaliação quantitativa do desempenho preditivo de modelos de regressão requer métricas capazes de mensurar a discrepância entre valores preditos e valores observados. Este trabalho emprega duas métricas complementares: *Mean Squared Error* (MSE) e *Normalized Mean Absolute Error* (NMAE), cada uma com propriedades e interpretações específicas.

2.5.1 Mean Squared Error (MSE)

O *Mean Squared Error* (HASTIE; TIBSHIRANI; FRIEDMAN, 2009) é definido como a média dos quadrados das diferenças entre valores preditos e observados:

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2,$$

onde m é o número de amostras no conjunto de teste, y_i é o valor observado da i-ésima amostra, e $\hat{y}_i$ é o valor predito correspondente.

O MSE possui interpretação direta em termos de teoria estatística de decisão: sob premissas de erros gaussianos e perda quadrática, a minimização do MSE corresponde

à estimação de máxima verossimilhança. A penalização quadrática confere maior peso a erros grandes comparados a erros pequenos, o que pode ser desejável quando grandes desvios são particularmente custosos ou indesejáveis.

Matematicamente, o MSE é diferenciável e convexo em relação aos parâmetros do modelo (para modelos lineares), o que facilita sua otimização por métodos baseados em gradiente. Ademais, em termos esperados, o MSE pode ser decomposto como:

$$\mathbb{E}[\text{MSE}] = \text{Bias}^2 + \text{Variância} + \text{Ruído Irredutível},$$

explicitando o compromisso fundamental entre viés e variância (*bias–variance trade-off*) em aprendizado de máquina (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

A principal limitação do MSE reside em sua sensibilidade a *outliers*: dado que os erros são elevados ao quadrado, amostras com desvios extremos podem dominar o valor da métrica, potencialmente distorcendo a percepção de desempenho geral do modelo. Além disso, o MSE não é interpretável em termos das unidades originais da variável alvo (sua unidade é o quadrado da unidade de y), dificultando comunicação intuitiva dos resultados.

Uma variante comumente reportada é a raiz do erro quadrático médio (*Root Mean Squared Error* – RMSE), definida como $\text{RMSE} = \sqrt{\text{MSE}}$, que retorna à escala original das observações.

2.5.2 Normalized Mean Absolute Error (NMAE)

O *Normalized Mean Absolute Error* (WILLMOTT; MATSUURA, 2005) é definido como:

$$\text{NMAE} = \frac{1}{\bar{y}} \cdot \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i| = \frac{\text{MAE}}{\bar{y}},$$

onde $\text{MAE} = \frac{1}{m} \sum_{i=1}^{m} |y_i - \hat{y}_i|$ é o erro absoluto médio e $\bar{y} = \frac{1}{m} \sum_{i=1}^{m} y_i$ é a média dos valores observados no conjunto de teste.

A normalização pela média dos valores observados torna o NMAE adimensional e facilita comparações entre diferentes contextos ou escalas. Um NMAE de 0,10 (ou 10%) indica que, em média, as predições desviam-se de 10% do valor médio observado, proporcionando interpretação intuitiva e percentual do erro.

O uso do erro absoluto (em vez de quadrático) confere ao NMAE maior robustez a *outliers*: desvios extremos contribuem proporcionalmente ao seu valor absoluto, não ao seu quadrado. Isso pode ser vantajoso quando *outliers* são considerados anomalias ou quando se deseja uma métrica que reflita o erro típico da maioria das predições, sem inflação por casos excepcionais.

Por outro lado, o MAE (e consequentemente NMAE) não é diferenciável na origem ($y = \hat{y}$), o que pode complicar otimização por métodos de primeira ordem em certas situações. Além disso, a métrica trata todos os erros igualmente, independentemente de sua magnitude, o que pode não refletir adequadamente a gravidade relativa de grandes versus pequenos desvios em domínios onde a penalização deve ser não-linear.

2.5.3 Complementaridade das Métricas

A utilização conjunta de MSE e NMAE permite avaliação mais abrangente do desempenho dos modelos. Diferenças substanciais entre as duas métricas podem indicar presença de *outliers* ou assimetrias na distribuição dos erros. Por exemplo, se o RMSE é significativamente maior que o MAE, isso sugere que alguns erros grandes estão inflando o RMSE, sinalizando possível presença de predições extremamente errôneas para um subconjunto de amostras.

A escolha da métrica primária para comparação de modelos deve considerar o contexto aplicacional. Quando a minimização de erros extremos é prioritária (e.g., em sistemas de controle onde grandes desvios podem ter consequências severas), o MSE pode ser mais apropriado. Quando se busca uma medida representativa do erro típico, menos influenciada por casos atípicos, o NMAE oferece perspectiva mais equilibrada.

No contexto deste trabalho, onde o objetivo é comparar métodos de seleção de atributos em relação à sua capacidade de manter a acurácia preditiva com subconjuntos reduzidos, a utilização de ambas as métricas proporciona visão complementar, evidenciando tanto o comportamento médio (NMAE) quanto a sensibilidade a desvios extremos (MSE).

2.6 Trabalhos Relacionados

Esta seção revisa os principais trabalhos relacionados à aplicação de técnicas de aprendizado de máquina para a monitoração e o gerenciamento de infraestruturas de redes e computação em nuvem, bem como estudos que abordam o uso de seleção de atributos e redução de dimensionalidade nesse contexto. O objetivo é situar o presente trabalho no estado da arte, destacando abordagens, resultados e lacunas existentes na literatura.

2.6.1 Estimação de QoS a partir de Estatísticas de Rede

O trabalho apresentado por Pasquini e Stadler (2017), publicado na conferência IEEE NetSoft 2017, investigou a viabilidade de estimar métricas de Qualidade de Serviço (QoS) fim-a-fim de aplicações a partir de estatísticas coletadas exclusivamente em dispositivos de rede OpenFlow. Os autores avaliaram essa abordagem em um *testbed* ex-

perimental composto por um cluster de servidores executando duas aplicações distintas: um serviço de vídeo sob demanda (*Video-on-Demand* – VoD) baseado em VLC e um sistema de armazenamento chave-valor (*Key-Value Store* – KV) utilizando Voldemort.

A infraestrutura de rede foi implementada por meio de uma topologia OpenFlow de três camadas (núcleo, agregação e borda), totalizando 14 switches virtualizados com Open vSwitch. Durante experimentos com duração de 12 horas, foram coletados três conjuntos principais de métricas: (i) estatísticas do nível de sistema operacional dos servidores (X_{cluster}), abrangendo métricas relacionadas a CPU, memória, disco e rede; (ii) estatísticas de portas dos switches OpenFlow (X_{port}), incluindo contadores de pacotes e bytes transmitidos e recebidos; e (iii) estatísticas de fluxos OpenFlow (X_{flow}), capturando métricas bidirecionais de tráfego entre clientes e servidores.

As métricas de QoS consideradas incluíram, para o serviço VoD, a taxa de quadros de vídeo (*frame rate*) e métricas relacionadas à reprodução de áudio; e, para o serviço KV, os tempos de resposta de operações de leitura e escrita. Os experimentos foram conduzidos sob diferentes padrões de carga, incluindo cenários periódicos e eventos do tipo *flash crowd*, permitindo avaliar o comportamento dos modelos em condições dinâmicas de tráfego.

Utilizando modelos de aprendizado supervisionado, como *Decision Trees* e *Random Forest*, os resultados indicaram que métricas coletadas no nível da rede eram suficientes para estimar métricas de QoS fim-a-fim com erros relativamente baixos. Em particular, observou-se que estatísticas provenientes das portas dos switches OpenFlow (X_{port}) apresentavam desempenho preditivo comparável àquele obtido com conjuntos mais abrangentes de métricas, incluindo estatísticas coletadas diretamente dos servidores de aplicação.

No que se refere à seleção de atributos, o estudo empregou o método *Forward Stepwise Selection* para reduzir conjuntos originalmente compostos por milhares de métricas para subconjuntos significativamente menores. Os resultados sugeriram que tais subconjuntos reduzidos eram capazes de preservar o desempenho preditivo dos modelos, ao mesmo tempo em que diminuíam substancialmente o custo computacional associado ao treinamento e à inferência.

2.6.2 Extensões na Seleção de Atributos e Análise de Custo Computacional

O trabalho foi subsequentemente expandido e publicado no *Journal of Network and Systems Management* por Stadler, Pasquini e Fodor (2017), em edição especial comemorativa do 25º aniversário da revista. Esta versão estendida incluiu contribuições adicionais significativas: (i) experimentos com tráfego concorrente, avaliando o impacto de múltiplas aplicações compartilhando a mesma infraestrutura; (ii) aplicação de método *baseline* ingênuo para estabelecer referência mínima de desempenho; (iii) análise detalhada da importância relativa de diferentes tipos de estatísticas; e (iv) avaliação da viabilidade

de utilizar estatísticas de um único switch na rota entre cliente e servidores, em vez de toda a topologia.

Os autores aplicaram também método univariado de seleção de atributos baseado em correlação, além do *Forward Stepwise Selection*, permitindo comparação entre abordagens *filter* e *wrapper*. Análise dos atributos selecionados revelou que features com maiores escores de relevância tendiam a corresponder a estatísticas de dispositivos localizados no caminho de rede entre o cluster de servidores e o cliente, corroborando a hipótese de que informação relevante concentra-se na trajetória do tráfego da aplicação.

Um aspecto importante destacado pelos autores refere-se ao custo computacional da seleção de atributos. Embora o *Forward Stepwise Selection* reduza drasticamente o custo de treinamento dos modelos finais (de dezenas de milhares de segundos para centenas), o processo de seleção em si é computacionalmente custoso, requerendo entre 500 e 6.000 segundos para *Decision Tree* e entre 100.000 e 300.000 segundos para *Random Forest*. Os autores reconhecem esta limitação e sugerem que métodos alternativos de menor custo computacional poderiam ser investigados.

2.6.3 Lacunas e Oportunidades de Pesquisa

Não obstante as contribuições relevantes dos trabalhos de Pasquini e Stadler (2017) e Stadler, Pasquini e Fodor (2017), é possível identificar algumas limitações e lacunas que motivam investigações adicionais. Embora esses estudos tenham empregado Forward Selection e métodos univariados baseados em correlação, não foi realizada uma comparação sistemática entre diferentes abordagens de seleção de atributos. Em particular, técnicas como Backward Selection, métodos univariados alternativos (e.g., *SelectKBest* com estatística F) e análise combinatória exaustiva para subconjuntos de pequena cardinalidade não foram consideradas.

Adicionalmente, os estudos concentraram-se principalmente na demonstração da viabilidade da redução do número de atributos, sem uma análise quantitativa detalhada do compromisso entre diferentes métodos em termos de desempenho preditivo, número de atributos selecionados e custo computacional. Aspectos como o custo do próprio processo de seleção em relação ao ganho obtido no tempo de treinamento dos modelos, bem como a estabilidade dos subconjuntos identificados frente a variações nos dados, não foram explorados de forma sistemática. A ausência de uma busca exaustiva como referência limita, por exemplo, a quantificação de quão próximos os métodos heurísticos se encontram do subconjunto ótimo global, aspecto relevante para avaliar sua eficácia relativa.

Outro ponto diz respeito ao desenho experimental. Os experimentos foram conduzidos com uma única execução para cada configuração avaliada, sem a realização de múltiplas repetições ou de diferentes amostragens aleatórias do conjunto de dados. Como

consequência, não é possível avaliar a estabilidade dos subconjuntos de atributos identificados ou sua sensibilidade a particularidades específicas das amostras utilizadas. Ademais, embora métricas como NMAE e tempo de treinamento tenham sido reportadas, análises complementares envolvendo outras métricas de erro (e.g., MSE, RMSE e coeficiente de determinação R^2), bem como visualizações da distribuição dos erros e da dispersão das predições, poderiam proporcionar uma compreensão mais aprofundada do comportamento dos modelos.

2.6.4 Posicionamento do Trabalho Atual

O presente trabalho insere-se no contexto de pesquisas que investigam a aplicação de técnicas de aprendizado de máquina para a predição de métricas de Qualidade de Serviço (QoS) em infraestruturas de cloud-networks, buscando avançar na compreensão do papel da seleção de atributos nesse domínio. Em particular, o estudo aborda lacunas identificadas na literatura quanto à ausência de análises comparativas sistemáticas entre diferentes métodos de seleção de atributos, especialmente no que se refere ao compromisso entre desempenho preditivo e complexidade computacional.

A principal contribuição deste trabalho consiste na implementação e avaliação comparativa de quatro abordagens distintas de seleção de atributos — análise combinatória exaustiva, Forward Selection, Backward Selection e *SelectKBest* — aplicadas ao problema de predição de QoS. Essa abordagem permite uma comparação quantitativa rigorosa das vantagens, limitações e aplicabilidade prática de cada método, considerando múltiplas dimensões de avaliação.

Os experimentos utilizam dados experimentais disponibilizados publicamente por Stadler, Pasquini e Fodor (2017), o que assegura reprodutibilidade e comparabilidade direta com resultados previamente reportados na literatura, ao mesmo tempo em que viabiliza investigações adicionais não exploradas nos estudos originais. A validade estatística dos resultados é reforçada por meio da geração de múltiplos subconjuntos aleatórios a partir do dataset base, possibilitando a avaliação da variabilidade, estabilidade dos subconjuntos selecionados e robustez dos métodos frente a diferentes amostragens dos dados.

A perspectiva de avaliação é ampliada pela utilização conjunta das métricas MSE e NMAE, bem como pela aplicação de três algoritmos de aprendizado supervisionado com características distintas — Regressão Linear, *Decision Tree* e *Random Forest*. Essa combinação permite analisar o comportamento dos métodos de seleção de atributos em diferentes contextos algorítmicos. Sempre que viável computacionalmente, a adoção de busca exaustiva fornece um referencial ótimo, possibilitando quantificar o desvio de otimalidade das abordagens heurísticas.

Diferentemente de trabalhos anteriores, nos quais a seleção de atributos foi em-

pregada predominantemente como técnica auxiliar para demonstração de viabilidade, o presente estudo a posiciona como objeto central de investigação. São conduzidas análises detalhadas de complexidade computacional, escalabilidade e interpretabilidade dos subconjuntos selecionados, com o objetivo de fornecer diretrizes práticas e fundamentadas para a escolha de métodos de seleção de atributos em função de restrições de tempo, recursos computacionais e requisitos de desempenho preditivo.

3 Metodologia e Desenho Experimental

Este capítulo apresenta a metodologia adotada para a condução dos experimentos de seleção de atributos no contexto da monitoração de infraestruturas de redes de cloud computing. São descritos o *dataset* utilizado, os procedimentos de pré-processamento dos dados, a geração de subconjuntos aleatórios, as estratégias de seleção de atributos implementadas, os algoritmos de aprendizado de máquina empregados e a configuração geral do desenho experimental.

3.1 Dataset Original e Caracterização

Os experimentos deste trabalho foram conduzidos utilizando um *dataset* derivado dos dados experimentais disponibilizados por Pasquini e Stadler (2017), coletados em um ambiente de experimentação controlado com infraestrutura de redes definidas por software (SDN) baseada no protocolo OpenFlow. O *dataset* corresponde a uma aplicação de *Video-on-Demand* (VoD), caracterizada por tráfego de natureza periódica e previsível, conforme descrito no estudo original.

O arquivo principal, denominado `X_cluster.csv`, possui dimensões originais de 37.036 amostras $\times$ 10.375 atributos. Para os experimentos deste trabalho, foram utilizadas as primeiras 10.800 amostras temporais, correspondentes a aproximadamente 3 horas de execução da aplicação com coletas em intervalos de 1 segundo. Cada amostra representa um instante de tempo e agrega métricas provenientes de múltiplas camadas da infraestrutura, incluindo estatísticas de dispositivos de rede OpenFlow, servidores e switches. O conjunto inicial de atributos é composto por variáveis numéricas relacionadas ao desempenho de rede, utilização de CPU, memória, disco e interrupções de hardware. Exemplos de atributos incluem `cpu0_.idle`, `eth1_rxkB.s`, `dev252.0_await` e `kbmemused`, dentre os 10.375 disponíveis no *dataset* original.

A variável alvo (`Y`), disponibilizada no arquivo `Y.csv`, corresponde à métrica de qualidade de experiência (*Quality of Experience* – QoE) denominada `DispFrames`, que mensura o número de quadros de vídeo exibidos por segundo na camada de aplicação. Essa métrica reflete diretamente a percepção de desempenho do usuário final, sendo amplamente utilizada como indicador crítico de qualidade em aplicações de streaming de vídeo.

3.2 Pré-processamento dos Dados

Antes da aplicação das técnicas de seleção de atributos, os dados passaram por uma etapa de pré-processamento com o objetivo de garantir qualidade, consistência e adequação aos algoritmos de aprendizado empregados. Inicialmente, foram removidas todas as colunas que continham valores ausentes (NaN, *Not a Number*), a fim de evitar a introdução de vieses ou instabilidades associadas a procedimentos de imputação durante o treinamento dos modelos.

Em seguida, atributos cujos valores permaneceram constantes ao longo de todas as amostras — em particular, colunas com valores iguais a zero em todo o período observado — foram excluídos por não apresentarem poder discriminativo para os modelos de regressão considerados. Adicionalmente, atributos categóricos ou textuais (dos tipos *object* ou *string*) foram removidos, uma vez que os algoritmos avaliados neste trabalho operam exclusivamente sobre atributos numéricos.

O campo de marcação temporal (`TimeStamp`) também foi descartado. Embora atue como chave de sincronização entre os arquivos `X_cluster.csv` e `Y.csv` — alinhando as métricas de infraestrutura coletadas em cada instante com o correspondente impacto na QoE da aplicação —, esse atributo não incorpora informação relevante sobre o estado intrínseco do sistema para fins de modelagem preditiva. O *timestamp* alimita-se a estabelecer a ordenação temporal das amostras, sem contribuir como variável explicativa na estimação da métrica alvo.

Dessa forma, optou-se por sua remoção antes da etapa de seleção de atributos. Ao término do pré-processamento, o *dataset* resultante passou a conter 1.808 atributos numéricos, representando uma redução de 10.375 para 1.808 variáveis após a exclusão de colunas com valores ausentes, constantes e categóricas. Com isso, os dados encontram-se adequadamente preparados para a aplicação de métodos de seleção de atributos e de modelos de regressão.

3.3 Seleção de Subconjuntos Aleatórios

Com o objetivo de avaliar a robustez e a capacidade de generalização dos métodos de seleção de atributos sob diferentes configurações de entrada, foram gerados cinco subconjuntos aleatórios a partir do *dataset* pré-processado. Cada subconjunto foi construído por meio da seleção aleatória, sem repetição, de 15 atributos distintos, garantindo diversidade na composição das variáveis de entrada e viabilizando a aplicação de análise combinatória exaustiva.

O processo de geração dos subconjuntos foi implementado em Python, utilizando as bibliotecas `pandas`, `numpy` e `random`. Inicialmente, o *dataset* pré-processado foi carre-

gado, e uma semente foi definida para assegurar a reprodutibilidade dos experimentos. Para cada um dos cinco subconjuntos, 15 atributos distintos foram selecionados aleatoriamente por meio da função `random.sample()`, e as colunas correspondentes foram extraídas e armazenadas em arquivos individuais. Os atributos selecionados em cada subconjunto também foram registrados em arquivos de texto, com o objetivo de facilitar a documentação e a rastreabilidade dos experimentos.

Cada subconjunto manteve as mesmas 10.800 amostras temporais do *dataset* original, variando apenas a dimensionalidade do espaço de atributos. Essa estratégia permitiu avaliar o comportamento dos métodos de seleção de atributos em múltiplas configurações de entrada, reduzindo o viés associado à análise de um único conjunto inicial de variáveis.

3.4 Estratégias de Seleção de Features

Foram implementadas e avaliadas quatro estratégias distintas de seleção de atributos, cada uma com características algorítmicas e complexidade computacional específicas. A estratégia de *Forward Selection* inicia com um conjunto vazio e adiciona iterativamente o atributo que mais contribui para a redução do erro do modelo. A cada etapa, treina-se um modelo para cada atributo candidato ainda não selecionado, escolhendo aquele que minimiza a métrica de erro no conjunto de teste. O processo continua enquanto houver melhoria na métrica de desempenho avaliada (MSE ou NMAE), terminando quando a adição de novos atributos não resulta em redução significativa do erro. Nos experimentos, o *Forward Selection* avaliou iterativamente todos os atributos restantes e adicionou aquele de melhor desempenho até atingir o critério de convergência, resultando em subconjuntos com diferentes quantidades de atributos dependendo do dataset e do algoritmo empregado.

A estratégia de *Backward Selection* adota a abordagem oposta, iniciando com o conjunto completo de atributos e removendo iterativamente aquele cuja ausência menos impacta o desempenho do modelo. A cada etapa, o algoritmo treina modelos excluindo cada atributo individualmente e identifica qual exclusão resulta no menor erro. O processo continua enquanto a remoção de atributos proporciona melhoria ou manutenção da métrica de desempenho, terminando quando a exclusão de qualquer atributo remanescente resulta em degradação significativa do erro. Assim como no *Forward Selection*, o número final de atributos selecionados varia conforme o dataset e o algoritmo empregado.

O método `SelectKBest`, disponível na biblioteca `scikit-learn`, implementa uma abordagem de seleção univariada baseada em testes estatísticos. Cada atributo é avaliado individualmente quanto à sua associação linear com a variável alvo por meio da função `f_regression`, que calcula a estatística F da ANOVA entre cada atributo e o alvo. Os k atributos com maiores escores são então selecionados. Neste trabalho, adotou-se

$k = 5$, selecionando os cinco atributos mais fortemente associados à métrica `DispFrames`. Essa abordagem apresenta complexidade computacional linear em relação ao número de atributos, sendo significativamente mais eficiente que os métodos *wrapper* progressivo e regressivo.

Por fim, a estratégia de busca exaustiva (*Exhaustive Search*) avalia todas as combinações possíveis de k atributos dentre o conjunto disponível, testando desde subconjuntos unitários até o conjunto completo, treinando e avaliando um modelo para cada combinação. O subconjunto que resulta no menor erro é selecionado como ótimo. Para 15 atributos de entrada, o número total de combinações não vazias é dado por $2^{15} - 1 = 32.767$ combinações. Apesar de sua complexidade combinatória, essa abordagem garante a identificação do subconjunto ótimo global para o modelo e métrica considerados, servindo como referência para comparação com os métodos heurísticos.

3.5 Algoritmos de Aprendizado de Máquina

Para cada subconjunto de atributos selecionado, foram treinados e avaliados três algoritmos de regressão amplamente empregados na literatura de aprendizado de máquina. Os modelos considerados foram Regressão Linear, *Decision Tree* e *Random Forest*, escolhidos por apresentarem diferentes níveis de complexidade e capacidade de modelagem.

A Regressão Linear foi implementada por meio da classe `LinearRegression` da biblioteca `scikit-learn`. Trata-se de um modelo paramétrico de baixo custo computacional, utilizado neste estudo como *baseline* para comparação de desempenho preditivo. Todos os parâmetros foram mantidos em seus valores padrão.

A *Decision Tree* foi implementada utilizando a classe `DecisionTreeRegressor`, também da biblioteca `scikit-learn`. Esse modelo não paramétrico permite capturar relações não lineares entre os atributos de entrada e a variável alvo. Para garantir reprodutibilidade, o parâmetro `random_state` foi fixado, enquanto os demais hiperparâmetros permaneceram com valores padrão.

Por fim, o método *Random Forest* foi implementado por meio da classe `RandomForestRegressor`. O modelo consiste em um conjunto de *Decision Trees* treinadas sobre amostras *bootstrap* dos dados, cujas predições são agregadas por meio da média aritmética. Para explorar paralelismo durante o treinamento, o parâmetro `n_jobs` foi configurado como `-1`, permitindo o uso de todos os núcleos disponíveis. O parâmetro `random_state` também foi fixado, e os demais hiperparâmetros foram mantidos em seus valores padrão.

3.6 Configuração dos Experimentos

Os experimentos foram estruturados de forma sistemática com o objetivo de avaliar as quatro estratégias de seleção de atributos em múltiplos cenários, reduzindo a influência de variações aleatórias e aumentando a confiabilidade dos resultados. As 10.800 amostras disponíveis foram particionadas em conjuntos de treinamento, correspondendo a 70% dos dados (7.560 amostras), e de teste, com os 30% restantes (3.240 amostras), utilizando a função `train_test_split` da biblioteca `scikit-learn`. Essa proporção é amplamente adotada na literatura e permite equilibrar adequadamente a capacidade de aprendizado dos modelos com a avaliação independente de desempenho.

Para reduzir a variabilidade associada à divisão aleatória dos dados e à inicialização estocástica dos modelos, cada configuração experimental foi executada com cinco sementes distintas (38, 42, 70, 128 e 527). O parâmetro `random_state` foi configurado de forma consistente em todas as etapas do processo experimental, incluindo a divisão entre treino e teste e a inicialização dos modelos baseados em árvores, garantindo plena reprodutibilidade dos resultados.

Dois critérios de desempenho foram considerados na avaliação dos modelos. O primeiro, o erro quadrático médio (MSE), penaliza fortemente predições com grandes desvios em relação aos valores observados. O segundo, o erro absoluto médio normalizado (NMAE), fornece uma interpretação percentual do erro ao normalizar o erro absoluto médio pelo valor médio da variável alvo. Cada estratégia de seleção de atributos foi avaliada separadamente em cenários de otimização baseados em MSE e em NMAE, resultando em dois conjuntos independentes de experimentos.

A combinação dos fatores experimentais considerados — cinco subconjuntos de dados de entrada, três algoritmos de aprendizado de máquina (Regressão Linear, *Decision Tree* e *Random Forest*), cinco sementes (38, 42, 70, 128 e 527) e duas métricas de otimização (NMAE e MSE) — resultou, para cada estratégia de seleção, em 75 execuções por métrica ($5 \times 3 \times 5$), totalizando 150 experimentos por estratégia. Considerando as quatro estratégias (Exaustiva, *Forward*, *Backward* e *SelectKBest*) avaliadas, o número total de experimentos conduzidos neste trabalho foi de 600.

O detalhamento completo das combinações experimentais avaliadas — abrangendo as *features* consideradas em cada dataset, os métodos de seleção aplicados, as sementes utilizadas e as métricas de avaliação — encontra-se sistematizado no Anexo A, garantindo a rastreabilidade e a reprodutibilidade dos experimentos conduzidos.

3.7 Ambiente de Execução

Os experimentos foram conduzidos em um ambiente computacional equipado com um processador AMD Ryzen 5 7600, composto por seis núcleos físicos e doze *threads* lógicos por meio da tecnologia SMT, 32 GB de memória RAM DDR5 e sistema operacional Windows 11 Pro de 64 bits. Essa configuração foi suficiente para a execução dos experimentos dentro de tempos de processamento compatíveis com o escopo do trabalho, especialmente considerando a possibilidade de paralelização no treinamento de *Random Forests*.

A implementação foi desenvolvida na linguagem Python, versão 3.13.2, utilizando o gerenciador de pacotes `uv` para a criação e o gerenciamento de ambientes virtuais e dependências. As principais bibliotecas empregadas incluem `scikit-learn` versão 1.6.1, responsável pela implementação dos algoritmos de aprendizado de máquina e dos métodos de seleção de atributos; `pandas` versão 2.2.3, utilizada para manipulação e organização dos dados tabulares; `numpy` versão 2.2.4, empregada em operações de computação numérica; e as bibliotecas de visualização `matplotlib` versão 3.10.1 e `seaborn` versão 0.13.2, utilizadas para a geração de gráficos e análises visuais dos resultados.

3.8 Procedimento de Execução

O fluxo de execução dos experimentos seguiu um procedimento sistemático composto por seis etapas principais. Inicialmente, realizou-se o carregamento dos dados por meio da leitura dos arquivos `X_cluster_subset_N.csv` e `Y.csv`, onde N representa o identificador do subconjunto de atributos (variando de 1 a 5). Em seguida, as 10.800 amostras foram particionadas em conjuntos de treinamento (70%) e teste (30%), utilizando a função `train_test_split` com o parâmetro `random_state` fixado de acordo com a semente do experimento em questão.

A terceira etapa consistiu na aplicação da estratégia de seleção de atributos correspondente. Para os métodos de *Forward Selection* e *Backward Selection*, os algoritmos foram executados de forma iterativa, treinando modelos intermediários exclusivamente com os dados do conjunto de treinamento e avaliando seu desempenho em um processo interno de validação até que o critério de convergência baseado na métrica de erro fosse atingido. No caso do método `SelectKBest`, os escores estatísticos F foram calculados para todos os atributos disponíveis no conjunto de treinamento, sendo selecionados os cinco com maior associação linear com a variável alvo. Para a estratégia de busca exaustiva, todas as 32.767 combinações possíveis foram geradas, e um modelo foi treinado e avaliado, no conjunto de treinamento, para cada combinação.

Após a definição do subconjunto de atributos, procedeu-se ao treinamento final

do modelo de aprendizado de máquina — Regressão Linear, *Decision Tree* ou *Random Forest* — utilizando exclusivamente os dados do conjunto de treinamento e o subconjunto de atributos selecionados. O modelo treinado foi então aplicado ao conjunto de teste para a geração de predições, e as métricas MSE e NMAE foram calculadas por meio da comparação entre os valores preditos e os valores observados. Por fim, os valores das métricas de erro, a lista de atributos selecionados, o modelo empregado e a semente utilizada foram armazenados em arquivos estruturados para posterior análise.

Todos os experimentos foram executados de forma automatizada por meio de notebooks Jupyter organizados por estratégia de seleção e métrica de otimização, assegurando consistência na aplicação dos procedimentos experimentais e facilitando a reprodutibilidade dos resultados. O código-fonte, os *notebooks* e os *datasets* utilizados estão disponíveis publicamente no repositório do *GitHub* (GOMES, 2026).

4 Resultados e Discussão

Este capítulo apresenta e discute os resultados obtidos nos experimentos de seleção de atributos descritos no Capítulo 3. A análise é organizada em seções correspondentes a cada uma das quatro estratégias de seleção avaliadas — busca exaustiva, seleção progressiva (*Forward Selection*), eliminação regressiva (*Backward Selection*) e seleção univariada (*SelectKBest*) —, seguidas de uma seção dedicada à comparação entre os métodos e de uma discussão geral dos principais achados. Para cada estratégia, são apresentados os resultados obtidos com os três algoritmos de aprendizado de máquina considerados (Regressão Linear, *Decision Tree* e *Random Forest*) segundo as duas métricas de desempenho empregadas (MSE e NMAE), bem como considerações sobre o custo computacional associado.

Os resultados apresentados neste capítulo são derivados de um conjunto abrangente de experimentos, considerando diferentes combinações de datasets, estratégias de seleção de *features*, algoritmos de aprendizado de máquina e sementes experimentais. Para evitar redundâncias, o detalhamento completo dessas configurações é apresentado no Anexo A, sendo referenciado sempre que necessário ao longo da análise dos resultados.

4.1 Resultados da Combinação Completa

A estratégia de busca exaustiva, também denominada análise combinatória completa ou *brute force*, consiste na avaliação sistemática de todas as combinações possíveis de atributos, garantindo a identificação do subconjunto ótimo global para cada modelo e métrica. Para 15 atributos de entrada, foram avaliadas $2^{15} - 1 = 32.767$ combinações não vazias por dataset. Considerando os três algoritmos de aprendizado de máquina, o total de modelos treinados e avaliados por dataset foi de $32.767 \times 3 = 98.301$. Com cinco datasets no total, cada experimento resultou em $98.301 \times 5 = 491.505$ modelos treinados por semente aleatória. Multiplicando-se pelas cinco sementes utilizadas (38, 42, 70, 128 e 527), o número total de experimentos conduzidos apenas para a estratégia de busca exaustiva atingiu 2.457.525 treinamentos de modelos. As configurações específicas de *features* associadas a cada cenário experimental analisado nesta seção estão detalhadas no Anexo A.

Esse volume de experimentos implicou custo computacional substancial. Para a métrica MSE com semente 42, o tempo total de execução foi de aproximadamente 9 horas (539,9 minutos), com tempos similares observados para as demais sementes e para a métrica NMAE. Esse custo computacional elevado constitui característica intrínseca da abordagem exaustiva e representa a principal limitação prática à sua aplicação em

cenários com maior número de atributos.

Em face disso, vale ressaltar que a escolha de 15 atributos por subconjunto não foi arbitrária, mas resultado de análise cuidadosa da viabilidade computacional da busca exaustiva. Inicialmente, tentou-se conduzir os experimentos com 20 atributos, visando maior representatividade do espaço de características. Entretanto, para $n = 20$, o número de combinações atingiria $2^{20} - 1 = 1.048.575$, aproximadamente 32 vezes superior ao caso com 15 atributos. Estimativas preliminares indicaram que o tempo de execução excederia 200 horas por experimento (mais de 8 dias), tornando a abordagem impraticável dado o escopo temporal deste trabalho. Tentativas subsequentes com 19, 18 e 17 atributos também se mostraram inviáveis, com tempos estimados superiores a 100, 50 e 25 horas, respectivamente. A redução para 15 atributos representou o maior valor de n que permitiu conclusão dos experimentos em tempo razoável (aproximadamente 9 horas), mantendo complexidade suficiente para avaliar comparativamente os métodos heurísticos. Essa limitação evidencia a natureza exponencial da busca exaustiva e reforça a importância de métodos alternativos para cenários de alta dimensionalidade.

4.1.1 Desempenho com MSE

Os resultados obtidos pela busca exaustiva quando otimizada segundo o MSE são apresentados na Tabela 1. Os valores reportados correspondem à média dos erros obtidos nos cinco datasets para cada combinação de modelo e semente aleatória.

Tabela 1 – Resultados MSE da busca exaustiva (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	12,8297	11,8978	12,2648	12,7166	11,9080
Decision Tree	13,2476	12,4767	12,8484	13,1511	12,3647
Random Forest	12,3022	11,3694	11,6571	12,0229	11,4314

Observa-se que o *Random Forest* consistentemente apresentou os menores valores de MSE entre os três algoritmos considerados, com médias variando entre 11,3694 (seed 70) e 12,3022 (seed 42). Esse resultado está alinhado com Breiman (2001), o qual aponta a capacidade de modelos baseados em *ensemble* de *Decision Trees* de capturar relações complexas e não lineares nos dados, além de maior robustez a ruído e *overfitting* em comparação com *Decision Trees* individuais.

A Regressão Linear apresentou desempenho intermediário, com valores de MSE entre 11,8978 e 12,8297, demonstrando que, mesmo sendo um modelo linear simples, é capaz de capturar adequadamente as relações entre os atributos selecionados e a variável alvo. A proximidade entre os desempenhos da Regressão Linear e do *Random Forest* sugere

que, para o problema em questão, relações predominantemente lineares estão presentes nos dados.

As *Decision Trees* individuais apresentaram desempenho ligeiramente inferior aos demais modelos em termos de MSE, com valores entre 12,3647 e 13,2476. Esse comportamento é esperado, visto que *Decision Trees* únicas tendem a apresentar maior variância e menor capacidade de generalização quando comparadas a métodos de *ensemble*.

Quanto ao número de atributos selecionados, a busca exaustiva identificou subconjuntos com diferentes cardinalidades dependendo do dataset e do algoritmo empregado. Para a Regressão Linear, por exemplo, os subconjuntos ótimos variaram entre 5 e 11 atributos, com médias de 9 atributos para o dataset 1, 8 para o dataset 2 e 11 para o dataset 3. Essa variação reforça a inexistência de um número ótimo universal de atributos, sendo essa quantidade dependente das características específicas de cada conjunto de dados.

4.1.2 Desempenho com NMAE

Os resultados para a métrica NMAE, apresentados na Tabela 2, seguem padrão distinto daquele observado com MSE. O NMAE, por normalizar o erro absoluto médio pelo valor médio da variável alvo, fornece interpretação percentual do desvio entre predições e valores observados.

Tabela 2 – Resultados NMAE da busca exaustiva (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	0,1054	0,1026	0,1035	0,1048	0,1029
Decision Tree	0,0872	0,0848	0,0871	0,0861	0,0852
Random Forest	0,0875	0,0844	0,0857	0,0862	0,0848

Diferentemente do observado com MSE, as *Decision Trees* individuais apresentaram desempenho ligeiramente superior ao *Random Forest* quando avaliadas por NMAE, com valores entre 0,0848 e 0,0872, em comparação com 0,0844 a 0,0875 para o *Random Forest*. Essa inversão na ordenação dos algoritmos evidencia que a métrica de avaliação influencia diretamente os subconjuntos de atributos selecionados e, consequentemente, o desempenho relativo dos modelos.

A Regressão Linear apresentou desempenho inferior aos modelos baseados em árvores, com valores de NMAE entre 0,1026 e 0,1054, aproximadamente 20% superiores aos obtidos pelos demais algoritmos. Esse resultado sugere que, embora a Regressão Linear seja adequada segundo a métrica MSE, que penaliza fortemente erros grandes, ela apre-

senta limitações ao lidar com a magnitude absoluta dos erros de forma uniforme, como capturado pelo NMAE.

4.1.3 Estabilidade entre Sementes

A análise da variação entre as cinco sementes aleatórias permite avaliar a estabilidade dos métodos frente a diferentes partições treino-teste e inicializações estocásticas dos modelos. Para o MSE, o desvio padrão entre as sementes foi de aproximadamente 0,4 para a Regressão Linear, 0,4 para as *Decision Trees* e 0,3 para o *Random Forest*, indicando estabilidade relativamente alta. Para o NMAE, os desvios foram ainda menores, da ordem de 0,001, reforçando a robustez da busca exaustiva.

Essa estabilidade elevada é esperada, uma vez que a busca exaustiva, ao avaliar todas as combinações possíveis, sempre identifica o subconjunto ótimo para cada configuração específica de dados, independentemente de heurísticas ou estratégias de busca sequencial que possam ser influenciadas pela ordem de avaliação dos atributos.

4.2 Resultados com Forward Stepwise

A estratégia de *Forward Selection* adota abordagem incremental, iniciando com um conjunto vazio e adicionando iterativamente o atributo que produz maior melhoria no desempenho do modelo. O processo continua até que não haja melhoria significativa na métrica de avaliação. Essa abordagem apresenta complexidade computacional substancialmente inferior à busca exaustiva, da ordem de $O(n^2 \cdot C)$, onde n é o número de atributos e C o custo de treinamento do modelo.

4.2.1 Desempenho com MSE

Os resultados obtidos pelo *Forward Selection* para a métrica MSE são apresentados na Tabela 3. Observa-se que, para a Regressão Linear e as *Decision Trees*, os valores de MSE são idênticos aos obtidos pela busca exaustiva, indicando que o método incremental convergiu para o ótimo global nesses casos.

Tabela 3 – Resultados MSE do *Forward Selection* (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	12,8297	11,8978	12,2648	12,7166	11,9080
Decision Tree	13,2476	12,4767	12,8498	13,1511	12,3647
Random Forest	13,1921	12,4034	12,7347	12,4966	12,3010

Para a Regressão Linear, a igualdade de resultados entre o *Forward Selection* e a busca exaustiva pode ser explicada pela natureza convexa do problema de otimização subjacente, que facilita a convergência para o ótimo global mesmo com estratégias gulosas. Para as *Decision Trees*, os valores praticamente idênticos (diferenças inferiores a 0,002 em alguns casos) sugerem que o método incremental identificou os mesmos subconjuntos de atributos ou subconjuntos com desempenho equivalente.

O *Random Forest*, por outro lado, apresentou desempenho ligeiramente inferior ao da busca exaustiva, com diferenças absolutas variando entre 0,12 (seed 70: 12,4034 vs. 11,3694) e 0,89 (seed 42: 13,1921 vs. 12,3022). Essas diferenças, embora modestas em termos absolutos (representando aproximadamente 7% de degradação), indicam que o método incremental convergiu para ótimos locais nesse caso, não identificando os mesmos subconjuntos ótimos encontrados pela busca exaustiva.

No entanto, esse resultado demonstra comportamento notavelmente favorável da estratégia gulosa, na qual, ao serem adicionados iterativamente os atributos mais informativos, o método convergiu para regiões muito próximas do mínimo global, com degradação inferior a 10% em todos os cenários. Esse achado é particularmente relevante, pois evidencia que, para o problema em questão, o espaço de busca apresenta estrutura favorável à exploração por heurísticas incrementais, possivelmente devido à redundância e correlação entre atributos de monitoramento em ambientes de cloud-networks.

O número de atributos selecionados pelo *Forward Selection* variou entre os datasets e algoritmos. Para a Regressão Linear com seed 42, foram selecionados 9 atributos para o dataset 1, 6 para o dataset 2, 10 para o dataset 3, 10 para o dataset 4 e 7 para o dataset 5, refletindo adaptação às características específicas de cada conjunto de dados. Esse padrão de variação é consistente com o observado na busca exaustiva, reforçando a ausência de cardinalidade ótima universal.

4.2.2 Desempenho com NMAE

Os resultados para NMAE, apresentados na Tabela 4, mostram comportamento similar ao observado com MSE, com convergência para valores próximos ou idênticos aos da busca exaustiva para todos os algoritmos considerados.

Tabela 4 – Resultados NMAE do *Forward Selection* (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	0,1054	0,1026	0,1035	0,1048	0,1029
Decision Tree	0,0904	0,0877	0,0886	0,0891	0,0879
Random Forest	0,0880	0,0845	0,0859	0,0863	0,0850

Novamente, a Regressão Linear apresentou valores idênticos aos da busca exaustiva, confirmando convergência para o ótimo global. As *Decision Trees* apresentaram diferenças ligeiramente maiores (por exemplo, 0,0904 vs. 0,0872 para seed 42), representando degradação de aproximadamente 3,7%, ainda assim modesta em termos práticos. O *Random Forest* apresentou diferenças ainda menores (inferiores a 0,6%), demonstrando que, para a métrica NMAE, o método incremental é altamente eficaz.

4.2.3 Custo Computacional e Estabilidade

O tempo de execução do *Forward Selection* foi substancialmente inferior ao da busca exaustiva, embora valores exatos não tenham sido registrados de forma sistemática. Estimativas baseadas na complexidade algorítmica sugerem redução de várias ordens de grandeza, permitindo conclusão dos experimentos em minutos, em contraste com as 9 horas requeridas pela busca exaustiva.

A estabilidade entre sementes permaneceu alta, com desvios padrão comparáveis aos da busca exaustiva, indicando que o método incremental não introduziu variabilidade adicional significativa. Esse resultado é importante, pois demonstra que a eficiência computacional não foi obtida às custas de maior sensibilidade às condições iniciais ou partições dos dados.

4.3 Resultados com Backward Stepwise

A estratégia de *Backward Selection* adota abordagem complementar ao *Forward Selection*, iniciando com o conjunto completo de atributos e removendo iterativamente aquele cuja ausência menos impacta o desempenho do modelo. Embora apresente complexidade computacional equivalente ao *Forward Selection*, $O(n^2 \cdot C)$, as primeiras iterações tendem a ser mais custosas por operarem com conjuntos maiores de atributos.

4.3.1 Desempenho com MSE

Os resultados do *Backward Selection* para MSE, apresentados na Tabela 5, revelam comportamento heterogêneo entre os algoritmos de aprendizado de máquina. Para a Regressão Linear, os valores são idênticos aos obtidos pela busca exaustiva e pelo *Forward Selection*, reforçando a convergência para o ótimo global nesse caso.

As *Decision Trees* apresentaram degradação substancial de desempenho, com valores de MSE entre 20,6850 e 22,6696, aproximadamente 60% superiores aos obtidos pela busca exaustiva. A inspeção dos subconjuntos selecionados mostrou que o método decremental reteve consistentemente mais atributos (11-13) do que os métodos baseados em busca global ou progressiva (7-10).

Tabela 5 – Resultados MSE do *Backward Selection* (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	12,8297	11,8978	12,2648	12,7166	11,9080
Decision Tree	20,6850	21,6335	22,0605	22,6696	21,9021
Random Forest	12,4268	11,4519	11,8592	12,1642	11,6867

Esse comportamento está associado à alta variância e à instabilidade estrutural inerentes às *Decision Trees*: pequenas alterações no conjunto de atributos podem modificar significativamente a estrutura da árvore desde o nó raiz. Durante o *Backward Selection*, essa sensibilidade gera avaliações de desempenho ruidosas, dificultando a identificação de atributos verdadeiramente dispensáveis.

Além disso, como *Decision Trees* operam por particionamentos hierárquicos do espaço de atributos, variáveis aparentemente fracas de forma isolada podem ser decisivas em divisões profundas ou em combinação com outras. A avaliação decremental, ao considerar remoções individuais, captura de forma limitada essas dependências condicionais. Assim, o efeito é amplificado pelo uso do MSE, cuja penalização quadrática intensifica erros localizados gerados por particionamentos subótimos.

Já o *Random Forest* apresentou desempenho muito próximo ao da busca exaustiva, com diferenças absolutas inferiores a 0,3 em todas as sementes, representando degradação inferior a 2,5%, indicando que o método decremental é altamente eficaz para esse algoritmo no contexto do MSE, possivelmente devido à capacidade do *ensemble* de capturar interações entre atributos desde as iterações iniciais, quando todos estão presentes.

4.3.2 Desempenho com NMAE

Os resultados para NMAE, apresentados na Tabela 6, mostram padrão mais uniforme entre os algoritmos, embora ainda com particularidades.

Tabela 6 – Resultados NMAE do *Backward Selection* (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	0,1054	0,1026	0,1035	0,1048	0,1029
Decision Tree	0,0916	0,0876	0,0914	0,0913	0,0891
Random Forest	0,0880	0,0852	0,0863	0,0863	0,0851

Para NMAE, as *Decision Trees* apresentaram degradação muito mais modesta em comparação com o MSE, com valores entre 0,0876 e 0,0916, representando aumento de aproximadamente 3% a 5% em relação aos resultados da busca exaustiva (0,0848 a 0,0872). Esse comportamento distinto entre as métricas sugere que o MSE, ao penalizar quadraticamente os erros, é mais sensível à presença de atributos redundantes ou ruidosos que podem ser preservados indevidamente pelo método decremental.

Dessa forma, o NMAE, por tratar erros de forma linear e sem amplificação, mostra-se mais robusto a imperfeições na seleção de atributos, sugerindo que o problema não reside tanto na magnitude média dos erros, mas sim na presença de predições ocasionalmente muito distantes dos valores observados, que são fortemente penalizadas pela métrica, mas afetam moderadamente a segunda. Por outro lado, a Regressão Linear e o *Random Forest* mantiveram desempenho consistente com o observado para MSE, reforçando a adequação do *Backward Selection* para esses algoritmos, mesmo sob a métrica NMAE.

4.3.3 Interpretação da Discrepância

A discrepância observada entre as *Decision Trees* e os demais algoritmos na eliminação regressiva com MSE constitui achado importante deste estudo. Ela evidencia que a escolha do método de seleção de atributos não pode ser dissociada das características do algoritmo de aprendizado de máquina empregado. Para modelos de alta variância, como *Decision Trees* individuais, métodos incrementais (que iniciam com conjuntos pequenos e crescem controladamente) tendem a ser mais adequados que métodos decrementais (que partem do conjunto completo e podem preservar atributos desnecessários por instabilidade nas avaliações).

4.4 Resultados com Univariate (Select K Best)

O método *SelectKBest* implementa abordagem de seleção univariada da família *filter*, avaliando cada atributo individualmente em relação à variável alvo por meio da estatística F da regressão linear univariada. Neste trabalho, foi configurado para selecionar os cinco atributos mais correlacionados com a métrica `DispFrames`, independentemente do algoritmo de aprendizado de máquina empregado posteriormente.

4.4.1 Desempenho com MSE

Os resultados do *SelectKBest* para MSE são apresentados na Tabela 7. Observa-se que a Regressão Linear apresentou desempenho muito próximo ao dos métodos *wrapper* (busca exaustiva, *Forward Selection* e *Backward Selection*), com diferenças absolutas inferiores a 0,1, representando degradação de menos de 1%.

Tabela 7 – Resultados MSE do SelectKBest (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	12,8724	11,9336	12,3002	12,7687	11,9453
Decision Tree	25,0823	23,7411	24,6964	24,2095	23,5810
Random Forest	13,2542	12,1465	12,4811	12,7736	12,3466

Esse resultado é notável, pois evidencia que, para a Regressão Linear, a seleção univariada baseada em correlações lineares é suficiente para identificar atributos altamente informativos, sem necessidade de avaliação de interações entre variáveis. Isso está alinhado com a natureza linear do modelo, que fundamentalmente assume relações aditivas entre os atributos.

Por outro lado, os Decision Trees demonstraram resultados consistentes com o padrão já observado no *Backward Selection*, com degradação ainda mais pronunciada, apresentando MSE entre 23,5810 e 25,0823, aproximadamente 90% superior à busca exaustiva. Esse resultado evidencia a sensibilidade das árvores de decisão à forma como os atributos são selecionados, especialmente quando métodos univariados são empregados.

Diferentemente dos métodos *wrapper*, o *SelectKBest* avalia atributos de forma univariada por meio de associações lineares com o alvo. Essa suposição é desalinhada com a natureza não linear e baseada em interações das *Decision Trees*. Variáveis relevantes apenas em combinação com outras tendem a receber escores baixos, enquanto atributos correlacionados mas redundantes podem ser simultaneamente selecionados.

Assim, embora por mecanismos distintos, tanto o *SelectKBest* quanto o *Backward Selection* sofrem de um desalinhamento estrutural em relação às *Decision Trees*: o primeiro por ignorar interações entre variáveis, o segundo por realizar avaliações instáveis sob pequenas perturbações no conjunto de atributos. Em ambos os casos, o impacto é exacerbado quando avaliado por MSE devido à penalização quadrática dos erros, efeito atenuado sob NMAE.

Por fim, o *Random Forest* apresentou degradação intermediária, com valores entre 12,1465 e 13,2542, aproximadamente 7% superiores aos da busca exaustiva. Embora o *ensemble* de árvores tenha capacidade de mitigar parcialmente as limitações da seleção univariada, o desempenho permaneceu inferior aos métodos *wrapper*.

4.4.2 Desempenho com NMAE

Os resultados para NMAE, apresentados na Tabela 8, seguem padrão similar ao observado com MSE, embora com degradações percentuais ligeiramente menores.

Tabela 8 – Resultados NMAE do SelectKBest (média sobre cinco datasets).

	Seeds				
Modelo	42	70	38	128	527
Linear Regression	0,1063	0,1034	0,1043	0,1055	0,1040
Decision Tree	0,1010	0,0957	0,0997	0,0978	0,0948
Random Forest	0,0928	0,0885	0,0905	0,0910	0,0897

A Regressão Linear manteve degradação inferior a 1,5%, reforçando sua adequação ao método univariado. As *Decision Trees* apresentaram degradação de aproximadamente 12% em relação à busca exaustiva, substancialmente inferior aos 90% observados com MSE, mas ainda significativa. O *Random Forest* apresentou degradação de aproximadamente 6%, consistente com o observado para MSE.

4.4.3 Eficiência Computacional

A principal vantagem do método reside em sua eficiência computacional significativamente superior em relação aos demais métodos avaliados. O cálculo dos escores F para os 15 atributos e a seleção dos cinco melhores foi praticamente instantâneo (ordem de milissegundos), representando redução de aproximadamente cinco ordens de grandeza em relação à busca exaustiva (9 horas para milissegundos) e três ordens em relação aos métodos *stepwise*, cujo tempo de execução foi estimado em menos de uma hora.

Essa diferença expressiva decorre diretamente da complexidade algorítmica associada a cada abordagem. Enquanto a busca exaustiva apresenta complexidade exponencial $O(2^n \cdot C)$ e os métodos *stepwise* exibem complexidade quadrática $O(n^2 \cdot C)$, o *SelectKBest* opera com complexidade linear $O(n \cdot m)$, onde m denota o número de amostras. Na prática, para o conjunto de dados com 15 atributos e 10.800 amostras, o método realizou apenas 15 regressões lineares univariadas, em contraste com os 491.505 modelos treinados pela busca exaustiva ou com os aproximadamente 5.050 modelos treinados pelos métodos *stepwise* (estimativa para *Forward Selection*).

Nesse contexto, tal eficiência torna o método particularmente atrativo como técnica de pré-seleção em cenários de alta dimensionalidade, nos quais mesmo métodos incrementais ou decrementais podem se tornar computacionalmente proibitivos. Conforme discutido na Seção 4.6, para o conjunto de dados original — composto por centenas de atributos —, o *SelectKBest* representa não apenas uma alternativa viável, mas possivelmente a única abordagem prática, dado que a busca exaustiva seria inviável do ponto de vista combinatório e os métodos *stepwise* demandariam semanas ou meses de processamento contínuo.

4.5 Comparação Geral entre Métodos

A Tabela 9 apresenta síntese comparativa dos métodos de seleção de atributos avaliados, considerando desempenho preditivo médio (MSE e NMAE), custo computacional relativo e estabilidade entre sementes. Os valores de desempenho correspondem às médias entre todas as sementes e algoritmos de aprendizado de máquina.

Tabela 9 – Comparação geral entre métodos de seleção de atributos.

	Desempenho (médias)		**Custo Computacional**	**Estabilidade**
Método	**MSE**	**NMAE**		
Busca Exaustiva	12,28	0,0908	Muito Alto (9h)	Muito Alta
Forward Selection	12,52	0,0919	Moderado (<1h)	Alta
Backward Selection	14,76	0,0934	Moderado (<1h)	Média
SelectKBest	16,38	0,0958	Muito Baixo (<1min)	Alta

4.5.1 Compromisso Desempenho-Custo

A análise comparativa revela clara relação de compromisso entre desempenho preditivo e custo computacional. A busca exaustiva, embora garanta identificação do ótimo global, apresenta custo proibitivo (aproximadamente 9 horas por métrica por semente) que inviabiliza sua aplicação em cenários práticos com maior número de atributos.

O *Forward Selection* emerge como método mais equilibrado, apresentando degradação média de apenas 2% em relação à busca exaustiva para MSE e menos de 1,5% para NMAE, com redução drástica de custo computacional (estimado em menos de 1 hora). Para aplicações que exigem alta acurácia preditiva mas dispõem de recursos computacionais limitados, o *Forward Selection* representa escolha altamente recomendável.

O *Backward Selection* apresentou desempenho heterogêneo, altamente dependente do algoritmo de aprendizado de máquina empregado. Para *Random Forest* e Regressão Linear, o desempenho foi comparável ao *Forward Selection*, mas para *Decision Trees* houve degradação severa (até 60% para MSE), limitando sua aplicabilidade geral.

O *SelectKBest*, embora apresente degradação mais pronunciada (33% para MSE e 5,5% para NMAE em média), oferece eficiência computacional sem precedentes, sendo adequado como método de pré-seleção ou para cenários onde eficiência é prioritária em detrimento de acurácia máxima.

4.5.2 Influência do Algoritmo de Aprendizado

A análise dos resultados evidencia que a adequação de um método de seleção de atributos não pode ser avaliada independentemente do algoritmo de aprendizado de

máquina empregado. A Regressão Linear, devido à sua natureza convexa e linear, apresentou desempenho consistentemente alto em todos os métodos de seleção, inclusive no univariado, demonstrando menor dependência da qualidade do processo de seleção.

O *Random Forest*, por sua robustez intrínseca e capacidade de lidar com redundância e interações complexas, apresentou degradação modesta mesmo com métodos subótimos, embora ainda se beneficie de seleção baseada em *wrapper* para desempenho máximo.

As *Decision Trees* individuais, caracterizadas por alta variância, mostraram-se altamente sensíveis ao método de seleção, com desempenho fortemente dependente da qualidade dos subconjuntos identificados. Para esse algoritmo, métodos incrementais demonstraram clara superioridade sobre métodos decrementais e univariados.

4.5.3 Influência da Métrica de Avaliação

A métrica de avaliação (MSE versus NMAE) exerceu influência significativa sobre o desempenho relativo dos métodos. O MSE, ao penalizar quadraticamente os erros de predição, mostrou-se mais sensível à presença de atributos inadequados, resultando em degradações percentuais maiores para métodos subótimos. Em contraste, o NMAE, ao tratar os erros de forma linear e normalizada, apresentou maior tolerância a escolhas não ideais de atributos, refletida em degradações percentuais sistematicamente menores.

Essa diferença sugere que a escolha da métrica de otimização durante a seleção de atributos deve considerar não apenas a natureza da aplicação em questão — como a severidade associada a grandes erros de predição —, mas também a robustez desejada frente a escolhas subótimas de atributos. Em particular, métricas mais sensíveis a erros extremos tendem a amplificar os efeitos de escolhas inadequadas de atributos, enquanto métricas mais robustas podem mitigar tais impactos.

4.6 Discussão Geral

Os resultados obtidos neste trabalho corroboram a hipótese central de que métodos heurísticos de seleção de atributos são capazes de aproximar o desempenho da busca exaustiva com reduções substanciais de custo computacional. Em particular, o *Forward Selection* demonstrou ser altamente eficaz, convergindo para o ótimo global ou para soluções muito próximas na maioria dos cenários avaliados.

4.6.1 Mínimo Global versus Mínimo Local

Do ponto de vista teórico, a busca exaustiva garante identificação do mínimo global da função objetivo (MSE ou NMAE) sobre o espaço de todos os subconjuntos possíveis de

atributos. Essa garantia, entretanto, vem ao custo de complexidade exponencial, tornando o método impraticável para problemas de dimensionalidade moderada ou alta.

Os métodos heurísticos, por sua natureza gulosa (*greedy*), não garantem convergência para o mínimo global, estando sujeitos a ótimos locais. Entretanto, os resultados empíricos demonstram que, para o problema específico de seleção de atributos em contexto de predição de QoS em cloud-networks, a diferença entre ótimos locais identificados por métodos heurísticos e o ótimo global é frequentemente negligível.

Essa observação pode ser explicada pela estrutura do espaço de busca: em muitos casos, múltiplos subconjuntos de atributos apresentam desempenho similar, formando "platôs" de qualidade próxima ao ótimo. Nessas situações, mesmo estratégias gulosas têm alta probabilidade de identificar soluções de qualidade satisfatória. Adicionalmente, a redundância e correlação entre atributos de monitoramento em cloud-networks implica que diferentes subconjuntos podem capturar informação essencialmente similar, reduzindo a importância de identificar exatamente o subconjunto ótimo global.

4.6.2 Adequação Prática dos Métodos

Para aplicações práticas em sistemas de monitoramento de cloud-networks em produção, a escolha do método de seleção de atributos deve considerar múltiplos fatores além do desempenho preditivo puro. O tempo de execução, a escalabilidade para maior número de atributos, a interpretabilidade dos subconjuntos selecionados e a robustez frente a mudanças na distribuição dos dados são aspectos igualmente relevantes.

Nesse sentido, o *Forward Selection* destaca-se como o método mais adequado para a maioria dos cenários avaliados, ao oferecer um compromisso favorável entre desempenho preditivo e eficiência computacional. Sua natureza incremental também contribui para a interpretabilidade dos resultados, uma vez que a ordem de inclusão dos atributos reflete diretamente sua contribuição individual e marginal para o modelo. Já o *Backward Selection* deve ser empregado com cautela, especialmente quando o algoritmo de aprendizado de máquina apresenta alta variância. Para *Random Forest* e Regressão Linear, entretanto, demonstrou desempenho competitivo, podendo ser preferível em contextos nos quais se espera a presença de interações complexas entre atributos e se deseja considerá-las desde as etapas iniciais do processo de seleção.

O *SelectKBest*, embora apresente degradação de desempenho mais pronunciada (33% para MSE e 5,5% para NMAE em média), oferece um *trade-off* relevante no contexto de seleção de atributos. Com tempo de execução na ordem de milissegundos — muito inferior à busca exaustiva e aos métodos *stepwise* —, o método torna-se a alternativa viável em cenários de altíssima dimensionalidade, onde outros métodos seriam computacionalmente inviáveis. Nesse contexto, o *SelectKBest* não apenas se mostra viável, mas essencial.

Para modelos lineares, a degradação é mínima (inferior a 1%), enquanto para algoritmos não lineares, como *Decision Trees*, ela é maior, mas aceitável diante da impossibilidade prática de aplicar outros métodos. Uma estratégia híbrida promissora consiste em usar o *SelectKBest* para reduzir a dimensionalidade inicial, seguida de *Forward Selection* sobre o subconjunto reduzido, combinando eficiência com capacidade de capturar interações entre atributos.

4.6.3 Limitações e Trabalhos Futuros

Este estudo apresenta algumas limitações que devem ser devidamente reconhecidas. Primeiramente, os experimentos foram conduzidos com subconjuntos compostos por apenas 15 atributos, o que viabilizou a utilização da busca exaustiva como referência ao ótimo global. Embora essa escolha tenha sido necessária para permitir comparações rigorosas entre os métodos avaliados, ela não representa plenamente cenários reais de alta dimensionalidade, nos quais milhares de métricas podem ser coletadas simultaneamente. Nesse sentido, trabalhos futuros devem investigar o comportamento dos métodos heurísticos em contextos de maior escala, adotando referências alternativas ao ótimo global, como limites teóricos ou aproximações computacionalmente viáveis.

Adicionalmente, este trabalho concentrou-se exclusivamente em métodos de seleção de atributos das famílias *wrapper* e *filter*, não explorando técnicas *embedded*, tais como a regularização Lasso ou métodos baseados na importância de atributos em modelos de árvores. Essas abordagens constituem um compromisso interessante entre eficiência computacional e a capacidade de considerar interações entre atributos, o que justifica sua inclusão em análises comparativas em estudos futuros.

Por fim, não foi avaliada a estabilidade temporal dos subconjuntos de atributos selecionados. Em ambientes reais de monitoramento, a distribuição dos dados pode variar ao longo do tempo em função de mudanças na carga de trabalho, na configuração da infraestrutura ou da introdução de novos serviços. Assim, a investigação da necessidade e da frequência de re-seleção de atributos em cenários sujeitos a *concept drift* configura uma direção promissora para pesquisas futuras.

5 Conclusões e Trabalhos Futuros

Este trabalho investigou o compromisso entre desempenho preditivo e complexidade computacional em métodos de seleção de atributos aplicados à predição de métricas de qualidade de serviço (QoS) em ambientes de cloud-networks. A questão central era saber se estratégias heurísticas de seleção de atributos seriam capazes de preservar a qualidade preditiva próxima à da busca exaustiva enquanto reduzem de forma significativa o custo computacional. Os resultados obtidos respondem afirmativamente a essa questão, com nuances importantes dependentes do algoritmo de aprendizado e do método de seleção empregado.

A busca exaustiva cumpriu seu papel como referência ótima, avaliando todos os subconjuntos possíveis e identificando o de desempenho global mínimo para cada combinação de algoritmo e métrica. Com tempo de execução em torno de nove horas por semente, sua aplicação direta é inviável em cenários reais de produção, mas forneceu o parâmetro de comparação rigoroso que fundamentou todas as análises realizadas.

O *Forward Selection* destacou-se como o método mais eficaz dentre as heurísticas avaliadas. Para a Regressão Linear e o *Decision Tree*, o método convergiu exatamente para o ótimo global em todos os datasets e sementes avaliados. Para o *Random Forest*, a degradação manteve-se em patamar notável considerando que o método requer apenas uma fração do tempo computacional da busca exaustiva. Esse desempenho confirma a eficácia das estratégias incrementais em espaços de busca nos quais a contribuição marginal de cada atributo pode ser avaliada de forma razoavelmente independente.

O *Backward Selection* apresentou comportamento heterogêneo entre algoritmos. Para a Regressão Linear e o *Random Forest*, o método demonstrou desempenho comparável ao *Forward Selection*. Entretanto, para o *Decision Tree*, observou-se uma degradação expressiva em MSE que contrasta com valores controlados em NMAE. Esse comportamento divergente indica que o método, ao remover atributos de forma regressiva em modelos de alta variância, tende a reter subconjuntos subótimos que introduzem erros extremos pontuais — amplificados pelo MSE — sem comprometer a tendência geral das predições capturada pelo NMAE. Essa observação reforça a importância de considerar a interação entre o método de seleção e o algoritmo de aprendizado na prática.

O *SelectKBest* revelou-se uma alternativa de dupla face. Para a Regressão Linear, a degradação foi mínima, confirmando que a relevância univariada dos atributos é altamente informativa para modelos lineares. Para o *Decision Tree*, a degradação foi substancial, evidenciando a limitação fundamental dos métodos *filter*: ao avaliarem atributos de forma independente, ignoram interações entre variáveis que são cruciais para modelos

não lineares (CHANDRASHEKAR; SAHIN, 2014). Para o *Random Forest*, o resultado ficou em patamar aceitável. A principal vantagem do *SelectKBest* reside em seu custo computacional ínfimo, tornando-o a única alternativa viável quando o espaço de atributos possui centenas ou milhares de dimensões.

Considerando os resultados médios sobre todos os algoritmos e datasets, o *Forward Selection* consolidou-se como o método de melhor relação custo-benefício dentre as heurísticas avaliadas, com degradação marginal em relação à busca exaustiva. O *Backward Selection* e o *SelectKBest* apresentaram perdas progressivamente maiores, sendo o impacto mais pronunciado em MSE do que em NMAE — padrão consistente com as diferenças de sensibilidade entre as duas métricas.

Do ponto de vista prático, esses resultados têm implicações diretas para o projeto de sistemas de monitoramento inteligente em cloud-networks. A viabilidade de substituir a busca exaustiva pelo *Forward Selection* sem perda perceptível de desempenho — em especial para modelos lineares — reduz dramaticamente o tempo de configuração dos modelos preditivos, o que viabiliza re-seleções periódicas em ambientes dinâmicos.

A condução dos experimentos com cinco sementes distintas permitiu verificar a robustez estatística dos achados. Em todos os métodos avaliados, a variabilidade entre sementes mostrou-se baixa, indicando que os resultados não são artefatos de uma partição específica dos dados, mas refletem propriedades estruturais consistentes da relação entre os atributos de monitoramento e as métricas de QoS avaliadas. Essa estabilidade reforça a confiança nas conclusões e sugere que os métodos se comportam de forma previsível em diferentes amostras do dataset original.

Apesar dos resultados consistentes obtidos, este estudo apresenta limitações que devem ser reconhecidas para contextualizar adequadamente as conclusões e orientar pesquisas futuras.

Os experimentos foram conduzidos sobre subconjuntos de 15 atributos extraídos do dataset de cloud-networks. Essa restrição foi necessária para viabilizar a busca exaustiva como referência, mas cenários reais de monitoramento frequentemente envolvem centenas ou milhares de métricas simultâneas, condição na qual nem mesmo o *Forward Selection* e o *Backward Selection* podem ser aplicados diretamente com eficiência. Os resultados obtidos neste trabalho caracterizam, portanto, o comportamento dos métodos em um regime de dimensionalidade moderada, não devendo ser extrapolados para espaços de atributos muito maiores sem experimentação adicional.

O trabalho concentrou-se em métodos das famílias *wrapper* e *filter*, não contemplando métodos *embedded*, como a regularização Lasso (GUYON; ELISSEEFF, 2003) ou a importância de atributos derivada diretamente de modelos de *Random Forest* (BREIMAN, 2001). Esses métodos representam um compromisso diferente entre eficiência e

qualidade, e sua ausência constitui uma lacuna nos resultados comparativos.

Os experimentos utilizaram exclusivamente o dataset de VoD em cloud-networks de Pasquini e Stadler (2017), o que limita a generalização das conclusões para outros domínios — como redes móveis, computação de borda ou ambientes de Internet das Coisas — sem estudos adicionais. Da mesma forma, os experimentos foram conduzidos em regime estático, sem considerar variações temporais na distribuição dos dados. Em ambientes de produção, a distribuição de métricas de QoS pode sofrer *concept drift* ao longo do tempo, e a adequação dos subconjuntos selecionados a essas variações não foi avaliada neste trabalho.

Por fim, a análise de desempenho preditivo utilizou MSE e NMAE como critérios de comparação. Embora essas métricas sejam amplamente adotadas na literatura de predição de QoS (PASQUINI; STADLER, 2017), outras perspectivas — como estabilidade dos subconjuntos selecionados ao longo de sementes, interpretabilidade ou desempenho em cenários de distribuição assimétrica — não foram exploradas.

As limitações identificadas apontam diretamente para direções promissoras de pesquisa, das quais as mais relevantes são descritas a seguir.

Uma extensão natural é a avaliação em alta dimensionalidade, estendendo os experimentos para datasets com centenas ou milhares de atributos. Uma abordagem híbrida que combine o *SelectKBest* como etapa de pré-filtragem seguida de *Forward Selection* sobre o subconjunto reduzido pode oferecer melhor equilíbrio entre custo e qualidade em cenários de maior escala. Nesse mesmo sentido, o uso de bibliotecas com aceleração em GPU compatíveis com a API do `scikit-learn`, como a `cuML` da NVIDIA, representa uma direção promissora para contornar a principal restrição computacional do trabalho, potencialmente tornando a busca exaustiva viável para espaços de atributos maiores e permitindo comparações mais representativas de cenários reais.

A incorporação de métodos *embedded* — como regressão Lasso, *Ridge* e *Elastic Net* (GUYON; ELISSEEFF, 2003), bem como a seleção implícita realizada por *Random Forests* por meio da importância de variáveis (BREIMAN, 2001) — permitiria ampliar o escopo comparativo e avaliar abordagens que realizam seleção e treinamento de forma conjunta, potencialmente mais eficientes e robustas.

Investigar o comportamento dos subconjuntos selecionados ao longo do tempo em ambientes com variação de distribuição constitui outra frente relevante, em especial para sistemas de monitoramento autônomo em produção. Nessa direção, o desenvolvimento de mecanismos de seleção adaptativa que ajustem dinamicamente o subconjunto de atributos em resposta a mudanças detectadas nos dados — integrando técnicas de detecção de *drift* com pipelines de re-seleção automática — tem potencial direto de aplicação em orquestradores inteligentes de recursos em cloud-networks.

Por fim, a aplicação da mesma metodologia a outros domínios de redes, como tráfego móvel (5G/6G), computação de borda e IoT, permitiria avaliar em que medida as conclusões obtidas para cloud-networks se generalizam para topologias e padrões de tráfego distintos. Uma análise mais aprofundada dos próprios subconjuntos selecionados — sua sobreposição entre métodos e estabilidade ao longo das sementes — também poderia revelar atributos de monitoramento estruturalmente relevantes para a predição de QoS, independentemente do método empregado, contribuindo para uma melhor compreensão das relações subjacentes ao problema.

Referências

AL-DHURAIBI, Y. et al. Elasticity in cloud computing: State of the art and research challenges. **IEEE Transactions on Services Computing**, v. 11, n. 2, p. 430–447, 2018. Disponível em: <https://doi.org/10.1109/TSC.2017.2711009>. Citado na página 13.

ARMBRUST, M. et al. A view of cloud computing. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 53, n. 4, p. 50–58, abr. 2010. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/1721654.1721672>. Citado 2 vezes nas páginas 10 e 16.

BELOGLAZOV, A.; ABAWAJY, J.; BUYYA, R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. **Future Generation Computer Systems**, v. 28, n. 5, p. 755–768, 2012. ISSN 0167-739X. Special Section: Energy efficiency in large-scale distributed systems. Disponível em: <https://doi.org/10.1016/j.future.2011.04.017>. Citado 3 vezes nas páginas 10, 14 e 16.

BREIMAN, L. Random forests. **Machine Learning**, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A%3A1010933404324>. Citado 5 vezes nas páginas 25, 26, 41, 55 e 56.

BREIMAN, L. et al. **Classification and Regression Trees**. Andover, UK: Taylor & Francis/Chapman & Hall, 1984. 368 p. (Wadsworth Statistics/Probability Series). ISBN 9780412048418. Disponível em: <https://books.google.com/books?id=JwQx-WOmSyQC>. Citado na página 24.

CHANDRASHEKAR, G.; SAHIN, F. A survey on feature selection methods. **Comput. Electr. Eng.**, Pergamon Press, Inc., USA, v. 40, n. 1, p. 16–28, jan. 2014. ISSN 0045-7906. Disponível em: <https://doi.org/10.1016/j.compeleceng.2013.11.024>. Citado 7 vezes nas páginas 11, 13, 17, 18, 19, 23 e 55.

EL-GAZZAR, R. F. A literature review on cloud computing adoption issues in enterprises. In: BERGVALL-KÅREBORN, B.; NIELSEN, P. A. (Ed.). **Creating Value for All Through IT**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 214–242. ISBN 978-3-662-43459-8. Citado na página 10.

GOMES, I. A. R. **ml-feature-selection-cloud-networks: Reproducible code and datasets for feature selection in cloud-networks QoS prediction**. 2026. <https://github.com/IgorAugust0/ml-feature-selection-cloud-networks>. Acesso em: 21 abr. 2026. Citado na página 39.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 11.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 1157–1182, mar. 2003. ISSN 1532-4435. Citado 5 vezes nas páginas 11, 18, 22, 55 e 56.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition**. 2. ed. [S.l.]: Springer New York, NY, 2009. XXII, 745 p. (Springer Series in Statistics). 54 b/w illustrations, 604 colour illustrations. ISBN 978-0-387-84857-0. Citado 5 vezes nas páginas 20, 21, 23, 26 e 27.

JOHN, G. H.; KOHAVI, R.; PFLEGER, K. Irrelevant features and the subset selection problem. In: **Proceedings of the Eleventh International Conference on International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1994. (ICML'94), p. 121–129. ISBN 1558603352. Citado 2 vezes nas páginas 11 e 18.

LANEY, D. **3D Data Management: Controlling Data Volume, Velocity,and Variety**. [S.l.], 2001. Original research note introducing the 3Vs of big data. Disponível em: <https://diegonogare.net/wp-content/uploads/2020/08/3D-Data-Management-Controlling-Data-Volume-Velocity-and-Variety.pdf>. Citado na página 17.

LI, R. et al. Deep reinforcement learning for resource management in network slicing. **IEEE Access**, v. 6, p. 74429–74441, 2018. Disponível em: <https://doi.org/10.1109/ACCESS.2018.2881964>. Citado 2 vezes nas páginas 13 e 17.

MAO, H. et al. Resource management with deep reinforcement learning. In: **Proceedings of the 15th ACM Workshop on Hot Topics in Networks**. New York, NY, USA: Association for Computing Machinery, 2016. (HotNets '16), p. 50–56. ISBN 9781450346610. Disponível em: <https://doi.org/10.1145/3005745.3005750>. Citado 2 vezes nas páginas 13 e 17.

MCKEOWN, N. et al. Openflow: enabling innovation in campus networks. **SIGCOMM Comput. Commun. Rev.**, Association for Computing Machinery, New York, NY, USA, v. 38, n. 2, p. 69–74, mar. 2008. ISSN 0146-4833. Disponível em: <https://doi.org/10.1145/1355734.1355746>. Citado na página 11.

MONTGOMERY, D. C.; PECK, E. A.; VINING, G. G. **Introduction to Linear Regression Analysis**. 5th. ed. Hoboken, NJ: John Wiley & Sons, 2012. ISBN 978-0-470-54281-1. Citado na página 22.

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://doi.org/10.1109/NETSOFT.2017.8004198>. Citado 7 vezes nas páginas 10, 11, 21, 28, 30, 33 e 56.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado na página 22.

QUINLAN, J. Induction of decision trees. **Machine Learning**, Springer, v. 1, n. 1, p. 81–106, 1986. Disponível em: <https://doi.org/10.1007/BF00116251>. Citado na página 24.

STADLER, R.; PASQUINI, R.; FODOR, V. Learning from network device statistics. **J. Netw. Syst. Manage.**, Plenum Press, USA, v. 25, n. 4, p. 672–698, out. 2017. ISSN 1064-7570. Disponível em: <https://doi.org/10.1007/s10922-017-9426-z>. Citado 5 vezes nas páginas 10, 11, 29, 30 e 31.

TANENBAUM, A.; WETHERALL, D. **Computer Networks**. [S.l.]: Pearson Prentice Hall, 2011. ISBN 9780132126953. Citado na página 10.

WILLMOTT, C. J.; MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. **Climate Research**, v. 30, n. 1, p. 79–82, 2005. Disponível em: <https://www.int-res.com/articles/cr2005/30/c030p079.pdf>. Citado na página 27.

Anexos

ANEXO A – Detalhamento das Configurações e Resultados Experimentais

Este anexo apresenta o detalhamento completo dos resultados experimentais, incluindo as features selecionadas por cada método e os valores de desempenho (MSE e NMAE) obtidos pelos algoritmos de aprendizado de máquina. Para cada combinação de dataset e semente aleatória, são apresentadas duas tabelas: uma para os resultados com MSE e outra para os resultados com NMAE.

A estrutura das tabelas está organizada em duas seções principais:

- **Seção 1 - Seleção de Features:** Lista as 15 features de cada dataset e indica com um "X" quais foram selecionadas por cada método (Busca Exaustiva, Forward Selection, Backward Selection e SelectKBest).
- **Seção 2 - Resultados dos Modelos:** Apresenta os valores de MSE ou NMAE obtidos por cada algoritmo (Linear Regression, Decision Tree e Random Forest) considerando as features selecionadas por cada método.

A.1 Resultados para Semente 42

A.1.1 Dataset 1

Tabela 10 – Dataset 1, Semente 42, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle			X			X		X	X	
0_dev8.5_tps	X		X	X			X	X	X	X
0_eth1_rxkB.s	X		X	X			X		X	X
0_tcp6sck			X			X	X	X	X	
1_all_..usr	X		X	X			X	X	X	
1_dev252.0_svctm			X					X	X	
1_igmbq6.s			X						X	
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X		X	X			X	X	X	X
4_cpu4_.iowait	X	X	X	X	X	X	X	X	X	
4_dev8.5_.util	X			X			X	X		
4_i022_intr.s			X					X	X	
5_cpu0_.soft			X					X	X	
5_cpu2_.idle	X		X	X			X		X	X
5_cpu2_.irq	X		X	X			X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.3929			12.3929			12.3929			12.3976
Decision Tree	14.1073			14.1073			22.1639			25.6043
Random Forest	12.8567			13.9117			12.8567			13.4579

Tabela 11 – Dataset 1, Semente 42, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle		X	X		X			X	X	
0_dev8.5_tps	X	X	X	X		X	X	X	X	X
0_eth1_rxkB.s	X			X			X			X
0_tcp6sck		X					X	X		
1_all_..usr	X	X	X	X			X	X	X	
1_dev252.0_svctm		X	X			X		X	X	
1_igmbq6.s	X		X	X			X		X	
2_cpu0_.sys	X	X	X	X		X	X	X	X	X
3_iseg.s	X	X	X	X	X	X	X	X	X	X
4_cpu4_.iowait	X	X	X	X			X	X	X	
4_dev8.5_.util	X	X	X	X			X	X	X	
4_i022_intr.s		X	X			X		X	X	
5_cpu0_.soft		X						X		
5_cpu2_.idle	X		X	X			X		X	X
5_cpu2_.irq		X	X					X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1024			0.1024			0.1024			0.1025
Decision Tree	0.0909			0.0982			0.0909			0.1024
Random Forest	0.0908			0.0911			0.0908			0.0923

A.1.2 Dataset 2

Tabela 12 – Dataset 2, Semente 42, Métrica MSE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X		X	X			X		X	X
0_cpu12_.idle			X						X	
0_cpu12_.usr			X						X	
0_dev252.0_await	X		X	X			X		X	X
1_dev8.5_avgqu.sz	X		X	X			X		X	
1_frmpg.s			X						X	
1_tcpsck		X			X	X		X	X	
2_X.dev.sda1_.Iused	X		X				X	X		
3_cpu22_MHz	X						X	X	X	
3_iseg.s	X		X	X			X		X	X
3_kbmemused	X		X	X			X		X	X
4_cpu20_MHz			X					X		
4_cpu9_.usr	X		X	X			X		X	X
4_i151_intr.s			X							
4_tcp6sck			X				X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.3590			12.3590			12.3590			12.3614
Decision Tree	15.1591			15.1591			15.1591			24.5120
Random Forest	12.8355			15.1588			12.8668			13.2286

Tabela 13 – Dataset 2, Semente 42, Métrica NMAE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X	X	X	X		X	X	X	X	X
0_cpu12_.idle		X						X	X	
0_cpu12_.usr		X						X	X	
0_dev252.0_await			X			X		X	X	X
1_dev8.5_avgqu.sz	X	X		X			X	X	X	
1_frmpg.s		X							X	
1_tcpsck	X	X		X	X		X	X		
2_X.dev.sda1_.Iused	X	X	X				X	X	X	
3_cpu22_MHz	X	X	X				X	X	X	
3_iseg.s	X	X	X	X	X	X	X		X	X
3_kbmemused	X	X	X	X	X	X	X	X	X	X
4_cpu20_MHz			X	X				X	X	
4_cpu9_.usr	X			X			X	X	X	X
4_i151_intr.s								X		
4_tcp6sck							X	X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1025			0.1025			0.1025			0.1026
Decision Tree	0.0885			0.0937			0.0954			0.0988
Random Forest	0.0899			0.0900			0.0906			0.0907

A.1.3 Dataset 3

Tabela 14 – Dataset 3, Semente 42, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait								X		
1_cpu6_.sys	X			X			X	X	X	
1_cpu8_.sys									X	
1_dev252.0_tps	X			X			X	X	X	
2_X.dev.sda1_MBfsused	X					X	X	X	X	
2_cpu1_.sys	X			X			X	X	X	X
2_i129_intr.s	X			X			X			
3_all_..usr	X			X			X	X	X	X
3_cpu9_.nice		X	X		X	X		X		
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s	X			X			X	X	X	
5_bwrtn.s								X	X	
5_cpu10_.soft	X			X			X	X	X	X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X			X			X			X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	13.2900			13.2900			13.2900			13.3187
Decision Tree	11.0283			11.0283			20.4114			24.8747
Random Forest	10.9607			10.9607			11.5245			13.4982

Tabela 15 – Dataset 3, Semente 42, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait								X	X	
1_cpu6_.sys		X	X		X	X		X	X	
1_cpu8_.sys										
1_dev252.0_tps	X			X			X	X		
2_X.dev.sda1_MBfsused								X	X	
2_cpu1_.sys	X			X			X	X		X
2_i129_intr.s	X	X	X	X	X	X	X		X	
3_all_..usr	X			X			X	X		X
3_cpu9_.nice		X	X		X	X		X	X	
4_cpu3_.idle	X			X			X	X		X
4_passive.s	X			X			X	X		
5_bwrtn.s								X		
5_cpu10_.soft								X		X
5_kbbuffers		X	X		X	X		X	X	
5_temp1_degC	X			X			X			X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1085			0.1085			0.1085			0.1086
Decision Tree	0.0763			0.0763			0.0842			0.1017
Random Forest	0.0772			0.0772			0.0772			0.0944

A.1.4 Dataset 4

Tabela 16 – Dataset 4, Semente 42, Métrica MSE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft			X					X	X	
0_dev252.0_avgrq.sz	X		X	X			X	X	X	
0_dev252.1_await	X			X			X			
0_eth1_rxkB.s	X		X	X			X		X	X
1_bus3_maxpower			X	X				X	X	
1_dev8.0_await										
2_X..swpcad							X	X	X	
2_eth1_rxkB.s	X		X	X			X	X	X	
2_i138_intr.s	X		X	X			X	X	X	X
2_sum_intr.s	X		X	X			X	X	X	X
4_X.dev.sda1_.ufsused							X	X	X	
4_cpu9_.idle	X	X	X	X	X	X	X	X	X	X
5_active.s	X		X	X			X	X	X	
5_bufpg.s	X		X	X			X	X		
5_cpu12_.soft										X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	13.4959			13.4959			13.4959			13.6014
Decision Tree	13.4266			13.4266			23.0361			26.7404
Random Forest	12.4351			13.4229			12.4628			13.2018

Tabela 17 – Dataset 4, Semente 42, Métrica NMAE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft					X	X		X	X	
0_dev252.0_avgrq.sz	X		X	X			X	X	X	
0_dev252.1_await	X			X		X	X		X	
0_eth1_rxkB.s	X	X		X			X			X
1_bus3_maxpower								X	X	
1_dev8.0_await		X								
2_X..swpcad	X					X	X	X	X	
2_eth1_rxkB.s		X	X					X	X	
2_i138_intr.s	X			X			X	X	X	X
2_sum_intr.s		X	X		X	X		X	X	X
4_X.dev.sda1_.ufsused	X						X	X	X	
4_cpu9_.idle		X	X					X	X	X
5_active.s	X		X	X			X	X	X	
5_bufpg.s	X		X	X			X	X	X	
5_cpu12_.soft	X			X			X			X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1081			0.1081			0.1081			0.1119
Decision Tree	0.0921			0.0949			0.0940			0.1059
Random Forest	0.0918			0.0940			0.0920			0.0948

A.1.5 Dataset 5

Tabela 18 – Dataset 5, Semente 42, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X						X	X		
0_file.nr	X		X	X			X		X	
1_i128_intr.s			X					X	X	
2_cpu7_.sys	X		X	X			X	X	X	X
2_eth0_.ifutil								X		
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice								X		
3_cpu3_MHz			X			X			X	
3_eth1_txkB.s								X		X
4_all_..irq	X	X	X	X	X	X	X	X	X	
4_cpu16_.sys	X		X	X			X	X	X	X
4_cpu20_.usr	X		X	X			X	X	X	X
4_omsg6.s	X			X			X	X		
5_cpu8_MHz	X		X				X	X	X	
5_dev252.0_avgqu.sz								X		
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.6108			12.6108			12.6108			12.6828
Decision Tree	12.5166			12.5166			22.6543			23.6799
Random Forest	12.4231			12.5066			12.4231			12.8846

Tabela 19 – Dataset 5, Semente 42, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X	X					X		X	
0_file.nr	X			X			X		X	
1_i128_intr.s								X	X	
2_cpu7_.sys	X			X			X	X		X
2_eth0_.ifutil		X						X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice		X	X		X	X		X	X	
3_cpu3_MHz	X	X	X				X		X	
3_eth1_txkB.s	X			X			X	X		X
4_all_..irq	X		X	X	X		X	X	X	
4_cpu16_.sys	X	X	X	X	X	X	X	X	X	X
4_cpu20_.usr	X			X			X	X		X
4_omsg6.s	X			X	X		X	X	X	
5_cpu8_MHz	X	X	X				X	X	X	
5_dev252.0_avgqu.sz								X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1055			0.1055			0.1055			0.1061
Decision Tree	0.0885			0.0886			0.0935			0.0965
Random Forest	0.0878			0.0878			0.0894			0.0919

A.2 Resultados para Semente 70

A.2.1 Dataset 1

Tabela 20 – Dataset 1, Semente 70, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle	X			X			X	X		
0_dev8.5_tps	X		X	X			X	X	X	X
0_eth1_rxkB.s	X		X	X			X	X	X	X
0_tcp6sck								X	X	
1_all_..usr								X		
1_dev252.0_svctm	X			X			X		X	
1_igmbq6.s								X		
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X		X	X			X	X	X	X
4_cpu4_.iowait	X			X			X	X		
4_dev8.5_.util	X		X	X			X	X	X	
4_i022_intr.s			X					X	X	
5_cpu0_.soft	X		X	X			X	X	X	
5_cpu2_.idle			X					X	X	X
5_cpu2_.irq		X			X	X		X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		11.4574			11.4574			11.4574		11.4687
Decision Tree		13.1390			13.1390			22.2966		23.4093
Random Forest		11.6770			13.0785			11.7123		12.1247

Tabela 21 – Dataset 1, Semente 70, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle	X	X		X		X	X			
0_dev8.5_tps	X	X		X			X	X		X
0_eth1_rxkB.s	X			X			X			X
0_tcp6sck		X	X			X		X	X	
1_all_..usr		X						X		
1_dev252.0_svctm	X			X			X			
1_igmbq6.s			X					X	X	
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X	X	X	X	X	X	X	X	X	X
4_cpu4_.iowait	X	X	X	X	X		X	X	X	
4_dev8.5_.util	X			X			X			
4_i022_intr.s		X						X		
5_cpu0_.soft	X	X	X	X		X	X	X	X	
5_cpu2_.idle	X	X	X	X		X	X	X	X	X
5_cpu2_.irq								X		
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1002			0.1002			0.1002		0.1003
Decision Tree		0.0867			0.0920			0.0875		0.0949
Random Forest		0.0865			0.0865			0.0865		0.0880

A.2.2 Dataset 2

Tabela 22 – Dataset 2, Semente 70, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X	X	X	X	X	X	X	X	X	X
0_cpu12_.idle	X			X			X	X	X	
0_cpu12_.usr	X			X			X		X	
0_dev252.0_await	X		X	X			X	X	X	X
1_dev8.5_avgqu.sz		X	X		X	X		X	X	
1_frmpg.s	X		X	X			X	X	X	
1_tcpsck	X			X			X	X		
2_X.dev.sda1_.Iused			X					X	X	
3_cpu22_MHz	X		X				X	X	X	
3_iseg.s	X		X	X			X	X	X	X
3_kbmemused	X		X	X			X	X	X	X
4_cpu20_MHz								X	X	
4_cpu9_.usr	X		X	X			X	X	X	X
4_i151_intr.s								X	X	
4_tcp6sck							X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	11.4645			11.4645			11.4645			11.4700
Decision Tree	14.0678			14.0678			21.7219			23.1355
Random Forest	11.3004			14.0890			11.3709			11.7046

Tabela 23 – Dataset 2, Semente 70, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X	X	X	X		X	X	X	X	X
0_cpu12_.idle			X		X			X		
0_cpu12_.usr	X	X	X	X			X		X	
0_dev252.0_await	X	X	X	X	X	X	X	X	X	X
1_dev8.5_avgqu.sz		X						X		
1_frmpg.s	X			X			X			
1_tcpsck	X	X		X		X	X	X	X	
2_X.dev.sda1_.Iused		X	X		X			X		
3_cpu22_MHz		X	X					X	X	
3_iseg.s	X	X	X	X		X	X	X	X	X
3_kbmemused	X	X	X	X	X	X	X	X	X	X
4_cpu20_MHz	X	X	X				X	X	X	
4_cpu9_.usr	X	X	X	X		X	X	X	X	X
4_i151_intr.s			X			X		X	X	
4_tcp6sck		X					X	X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1003			0.1003			0.1003			0.1003
Decision Tree	0.0865			0.0909			0.0888			0.0917
Random Forest	0.0847			0.0847			0.0847			0.0851

A.2.3 Dataset 3

Tabela 24 – Dataset 3, Semente 70, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait		X	X		X	X		X	X	
1_cpu6_.sys	X			X			X	X	X	
1_cpu8_.sys	X			X			X	X		
1_dev252.0_tps								X	X	
2_X.dev.sda1_MBfsused										
2_cpu1_.sys	X			X			X	X	X	X
2_i129_intr.s								X		
3_all_..usr	X			X			X	X		X
3_cpu9_.nice	X			X			X	X	X	
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s	X			X			X		X	
5_bwrtn.s								X		
5_cpu10_.soft	X			X			X	X	X	X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X			X			X	X	X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.3072			12.3072			12.3072			12.3216
Decision Tree	10.6204			10.6204			20.2475			23.5250
Random Forest	10.5529			10.5529			10.7846			12.7009

Tabela 25 – Dataset 3, Semente 70, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait			X			X			X	
1_cpu6_.sys								X	X	
1_cpu8_.sys								X	X	
1_dev252.0_tps	X			X			X		X	
2_X.dev.sda1_MBfsused			X			X		X	X	
2_cpu1_.sys	X			X			X	X		X
2_i129_intr.s										
3_all_..usr	X			X			X			X
3_cpu9_.nice	X		X	X		X	X	X	X	
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s	X			X			X	X	X	
5_bwrtn.s								X		
5_cpu10_.soft	X			X			X	X		X
5_kbbuffers		X	X		X	X		X	X	
5_temp1_degC	X		X	X		X	X	X	X	X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1052			0.1052			0.1052			0.1053
Decision Tree	0.0764			0.0764			0.0800			0.0962
Random Forest	0.0761			0.0761			0.0795			0.0913

A.2.4 Dataset 4

Tabela 26 – Dataset 4, Semente 70, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft								X	X	
0_dev252.0_avgrq.sz	X			X		X	X	X	X	
0_dev252.1_await	X	X		X	X	X	X	X	X	
0_eth1_rxkB.s	X		X	X			X	X	X	X
1_bus3_maxpower			X	X			X	X	X	
1_dev8.0_await			X						X	
2_X..swpcad			X				X	X	X	
2_eth1_rxkB.s	X		X	X			X	X	X	
2_i138_intr.s	X		X	X			X	X	X	X
2_sum_intr.s	X		X	X			X	X	X	X
4_X.dev.sda1_.ufsused			X				X	X		
4_cpu9_.idle	X	X	X	X	X	X	X	X	X	X
5_active.s	X		X	X			X	X	X	
5_bufpg.s									X	
5_cpu12_.soft	X		X	X			X	X	X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		12.4982			12.4982			12.4982		12.5753
Decision Tree		12.8929			12.8929			21.5293		24.5441
Random Forest		11.7351			12.6532			11.7713		12.1246

Tabela 27 – Dataset 4, Semente 70, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft	X		X	X			X	X	X	
0_dev252.0_avgrq.sz	X	X	X	X	X	X	X	X	X	
0_dev252.1_await	X	X	X	X	X	X	X	X	X	
0_eth1_rxkB.s	X	X	X	X			X	X	X	X
1_bus3_maxpower	X	X			X		X	X	X	
1_dev8.0_await			X						X	
2_X..swpcad	X		X				X	X	X	
2_eth1_rxkB.s		X						X	X	
2_i138_intr.s	X	X		X			X	X	X	X
2_sum_intr.s		X	X			X		X	X	X
4_X.dev.sda1_.ufsused		X	X				X	X		
4_cpu9_.idle		X	X		X	X		X	X	X
5_active.s	X		X	X		X	X	X	X	
5_bufpg.s	X			X			X		X	
5_cpu12_.soft	X	X		X		X	X	X	X	X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1050			0.1050			0.1050		0.1082
Decision Tree		0.0874			0.0916			0.0892		0.0976
Random Forest		0.0883			0.0886			0.0891		0.0903

A.2.5 Dataset 5

Tabela 28 – Dataset 5, Semente 70, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X						X	X	X	
0_file.nr	X		X	X			X	X	X	
1_i128_intr.s			X					X	X	
2_cpu7_.sys	X			X			X	X	X	X
2_eth0_.ifutil			X					X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice			X							
3_cpu3_MHz								X	X	
3_eth1_txkB.s			X						X	X
4_all_..irq	X	X		X	X	X	X		X	
4_cpu16_.sys	X		X	X			X	X	X	X
4_cpu20_.usr	X		X	X			X	X	X	X
4_omsg6.s	X			X			X			
5_cpu8_MHz	X		X				X		X	
5_dev252.0_avgqu.sz			X					X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		11.7616			11.7616			11.7616		11.8325
Decision Tree		11.6634			11.6634			22.3722		24.0917
Random Forest		11.5815			11.6433			11.6203		12.0778

Tabela 29 – Dataset 5, Semente 70, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X	X	X	X	X	X	X	X	X	
0_file.nr	X		X	X			X	X	X	
1_i128_intr.s			X					X	X	
2_cpu7_.sys	X			X			X	X		X
2_eth0_.ifutil	X	X		X			X	X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice					X	X		X	X	
3_cpu3_MHz	X		X				X	X	X	
3_eth1_txkB.s	X		X	X			X		X	X
4_all_..irq	X		X	X	X		X	X	X	
4_cpu16_.sys	X	X		X	X		X	X		X
4_cpu20_.usr	X		X	X		X	X	X	X	X
4_omsg6.s			X						X	
5_cpu8_MHz	X		X				X			
5_dev252.0_avgqu.sz			X					X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1021			0.1021			0.1021		0.1028
Decision Tree		0.0870			0.0875			0.0925		0.0980
Random Forest		0.0862			0.0866			0.0863		0.0881

A.3 Resultados para Semente 38

A.3.1 Dataset 1

Tabela 30 – Dataset 1, Semente 38, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle						X		X		
0_dev8.5_tps	X		X	X			X	X	X	X
0_eth1_rxkB.s			X					X	X	X
0_tcp6sck						X	X	X		
1_all_..usr										
1_dev252.0_svctm	X			X			X	X		
1_igmbq6.s	X		X	X			X	X	X	
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X		X	X			X	X	X	X
4_cpu4_.iowait			X						X	
4_dev8.5_.util	X			X			X	X		
4_i022_intr.s	X		X	X			X	X	X	
5_cpu0_.soft	X		X	X			X	X	X	
5_cpu2_.idle			X					X	X	X
5_cpu2_.irq		X			X	X		X		
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		11.8712			11.8712			11.8712		11.8800
Decision Tree		14.0581			14.0581			22.3000		24.1086
Random Forest		12.0166			13.7432			12.0166		12.3943

Tabela 31 – Dataset 1, Semente 38, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle		X	X					X		
0_dev8.5_tps	X	X	X	X		X	X	X	X	X
0_eth1_rxkB.s	X		X	X		X	X	X	X	X
0_tcp6sck			X	X	X			X		
1_all_..usr		X								
1_dev252.0_svctm		X						X		
1_igmbq6.s	X			X		X	X	X	X	
2_cpu0_.sys	X	X	X	X		X	X	X	X	X
3_iseg.s	X	X	X	X	X	X	X	X	X	X
4_cpu4_.iowait			X						X	
4_dev8.5_.util	X	X		X		X	X	X		
4_i022_intr.s	X	X	X	X			X	X	X	
5_cpu0_.soft	X	X		X			X	X	X	
5_cpu2_.idle	X			X	X		X	X	X	X
5_cpu2_.irq		X	X					X		
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1012			0.1012			0.1012		0.1013
Decision Tree		0.0902			0.0950			0.0927		0.0987
Random Forest		0.0883			0.0884			0.0887		0.0895

A.3.2 Dataset 2

Tabela 32 – Dataset 2, Semente 38, Métrica MSE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X		X	X			X	X	X	X
0_cpu12_.idle	X			X			X	X	X	
0_cpu12_.usr									X	
0_dev252.0_await	X		X	X			X		X	X
1_dev8.5_avgqu.sz	X	X	X	X			X	X	X	
1_frmpg.s	X		X	X			X	X	X	
1_tcpsck					X	X		X	X	
2_X.dev.sda1_.Iused	X							X	X	
3_cpu22_MHz							X	X	X	
3_iseg.s	X		X	X			X	X	X	X
3_kbmemused	X		X	X			X	X	X	X
4_cpu20_MHz								X	X	
4_cpu9_.usr	X		X	X			X	X	X	X
4_i151_intr.s		X	X					X	X	
4_tcp6sck			X				X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	11.9246			11.9246			11.9246			11.9334
Decision Tree	14.5832			14.5901			22.4407			22.8423
Random Forest	11.6423			14.5902			11.6900			11.8080

Tabela 33 – Dataset 2, Semente 38, Métrica NMAE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X		X	X			X	X	X	X
0_cpu12_.idle		X						X	X	
0_cpu12_.usr									X	
0_dev252.0_await		X	X		X			X	X	X
1_dev8.5_avgqu.sz	X			X			X		X	
1_frmpg.s	X			X			X	X	X	
1_tcpsck	X	X		X			X	X	X	
2_X.dev.sda1_.Iused	X	X			X		X	X	X	
3_cpu22_MHz	X						X	X	X	
3_iseg.s	X	X	X	X	X	X	X	X	X	X
3_kbmemused	X	X	X	X	X	X	X	X	X	X
4_cpu20_MHz		X		X				X	X	
4_cpu9_.usr	X	X	X	X		X	X	X	X	X
4_i151_intr.s			X		X	X		X	X	
4_tcp6sck	X	X					X	X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1012			0.1012			0.1012			0.1014
Decision Tree	0.0892			0.0906			0.0926			0.0925
Random Forest	0.0861			0.0865			0.0872			0.0862

A.3.3 Dataset 3

Tabela 34 – Dataset 3, Semente 38, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait								X	X	
1_cpu6_.sys								X	X	
1_cpu8_.sys									X	
1_dev252.0_tps	X			X			X	X	X	
2_X.dev.sda1_MBfsused							X	X	X	
2_cpu1_.sys	X			X			X	X	X	X
2_i129_intr.s								X	X	
3_all_..usr								X	X	X
3_cpu9_.nice							X	X	X	
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s	X			X			X	X	X	
5_bwrtn.s								X		
5_cpu10_.soft	X			X			X	X	X	X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X			X			X	X	X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.5097			12.5097			12.5097			12.5329
Decision Tree	10.7316			10.7316			21.2596			25.6485
Random Forest	10.6769			10.6769			11.3337			12.8614

Tabela 35 – Dataset 3, Semente 38, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait								X		
1_cpu6_.sys										
1_cpu8_.sys										
1_dev252.0_tps	X			X			X			
2_X.dev.sda1_MBfsused	X		X			X	X		X	
2_cpu1_.sys	X			X			X	X		X
2_i129_intr.s										
3_all_..usr								X		X
3_cpu9_.nice	X		X			X	X	X	X	
4_cpu3_.idle	X			X			X	X		X
4_passive.s								X		
5_bwrtn.s								X		
5_cpu10_.soft	X			X			X	X		X
5_kbbuffers		X	X		X	X		X	X	
5_temp1_degC	X	X	X	X	X	X	X	X	X	X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1051			0.1051			0.1051			0.1051
Decision Tree	0.0757			0.0757			0.0838			0.1033
Random Forest	0.0756			0.0756			0.0756			0.0916

A.3.4 Dataset 4

Tabela 36 – Dataset 4, Semente 38, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft			X					X	X	
0_dev252.0_avgrq.sz	X		X	X		X	X	X	X	
0_dev252.1_await	X			X		X	X	X		
0_eth1_rxkB.s	X		X	X			X	X	X	X
1_bus3_maxpower	X		X				X	X		
1_dev8.0_await								X		
2_X..swpcad	X						X	X	X	
2_eth1_rxkB.s	X		X	X			X		X	
2_i138_intr.s	X		X	X			X		X	X
2_sum_intr.s	X		X	X			X	X	X	X
4_X.dev.sda1_.ufsused								X		
4_cpu9_.idle		X	X		X	X		X	X	X
5_active.s			X					X	X	
5_bufpg.s			X					X	X	
5_cpu12_.soft			X					X	X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		12.8855			12.8855			12.8855		12.9436
Decision Tree		13.1194			13.1194			22.2025		25.7330
Random Forest		12.2134			12.9270			12.2134		12.7807

Tabela 37 – Dataset 4, Semente 38, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft		X	X		X			X	X	
0_dev252.0_avgrq.sz		X	X			X		X	X	
0_dev252.1_await	X	X	X	X	X		X		X	
0_eth1_rxkB.s	X			X			X	X		X
1_bus3_maxpower	X	X	X		X		X	X		
1_dev8.0_await		X								
2_X..swpcad	X		X	X			X	X		
2_eth1_rxkB.s		X	X			X		X	X	
2_i138_intr.s	X	X		X			X	X		X
2_sum_intr.s		X	X		X	X		X	X	X
4_X.dev.sda1_.ufsused		X						X	X	
4_cpu9_.idle		X	X			X		X	X	X
5_active.s								X	X	
5_bufpg.s		X						X		
5_cpu12_.soft	X			X			X	X	X	X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1065			0.1065			0.1065		0.1094
Decision Tree		0.0908			0.0917			0.0952		0.1040
Random Forest		0.0901			0.0904			0.0902		0.0934

A.3.5 Dataset 5

Tabela 38 – Dataset 5, Semente 38, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz						X	X		X	
0_file.nr	X			X			X	X	X	
1_i128_intr.s								X	X	
2_cpu7_.sys	X			X			X	X	X	X
2_eth0_.ifutil									X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice								X	X	
3_cpu3_MHz								X	X	
3_eth1_txkB.s									X	X
4_all_..irq	X	X	X	X	X	X	X	X	X	
4_cpu16_.sys	X			X			X	X		X
4_cpu20_.usr	X			X			X	X	X	X
4_omsg6.s								X	X	
5_cpu8_MHz							X		X	
5_dev252.0_avgqu.sz								X		
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.1330			12.1330			12.1330			12.2113
Decision Tree	11.7495			11.7495			22.0997			25.1497
Random Forest	11.7362			11.7362			12.0422			12.5608

Tabela 39 – Dataset 5, Semente 38, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X	X		X			X	X	X	
0_file.nr	X			X			X	X	X	
1_i128_intr.s								X	X	
2_cpu7_.sys	X			X			X	X	X	X
2_eth0_.ifutil								X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice		X	X					X	X	
3_cpu3_MHz			X					X	X	
3_eth1_txkB.s	X	X		X			X		X	X
4_all_..irq	X			X	X		X	X	X	
4_cpu16_.sys	X	X	X	X		X	X	X		X
4_cpu20_.usr	X			X			X	X	X	X
4_omsg6.s		X	X		X	X		X	X	
5_cpu8_MHz				X			X		X	
5_dev252.0_avgqu.sz	X	X	X	X		X	X	X		
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1036			0.1036			0.1036			0.1042
Decision Tree	0.0896			0.0900			0.0930			0.0999
Random Forest	0.0887			0.0887			0.0899			0.0916

A.4 Resultados para Semente 128

A.4.1 Dataset 1

Tabela 40 – Dataset 1, Semente 128, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle	X		X	X			X	X	X	
0_dev8.5_tps	X		X	X			X	X	X	X
0_eth1_rxkB.s	X		X	X			X	X	X	X
0_tcp6sck	X		X				X	X	X	
1_all_..usr			X					X	X	
1_dev252.0_svctm	X			X			X	X		
1_igmbq6.s	X		X	X			X	X	X	
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X		X	X			X	X	X	X
4_cpu4_.iowait			X					X	X	
4_dev8.5_.util			X					X	X	
4_i022_intr.s								X		
5_cpu0_.soft			X					X	X	
5_cpu2_.idle			X					X	X	X
5_cpu2_.irq	X			X			X	X		
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.3057			12.3057			12.3057			12.3073
Decision Tree	14.0565			14.0565			23.7707			23.8799
Random Forest	12.4670			13.9502			12.4670			12.7170

Tabela 41 – Dataset 1, Semente 128, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle			X		X			X	X	
0_dev8.5_tps	X	X	X	X		X	X	X	X	X
0_eth1_rxkB.s	X	X	X	X		X	X	X	X	X
0_tcp6sck		X					X	X		
1_all_..usr								X		
1_dev252.0_svctm	X			X			X	X		
1_igmbq6.s	X		X	X			X	X	X	
2_cpu0_.sys	X	X	X	X		X	X	X	X	X
3_iseg.s	X	X	X	X	X	X	X	X	X	X
4_cpu4_.iowait		X	X					X	X	
4_dev8.5_.util	X	X		X			X	X		
4_i022_intr.s								X		
5_cpu0_.soft	X	X	X	X		X	X	X	X	
5_cpu2_.idle	X		X	X			X	X	X	X
5_cpu2_.irq	X	X	X	X			X	X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1026			0.1026			0.1026			0.1027
Decision Tree	0.0891			0.0948			0.0960			0.0975
Random Forest	0.0889			0.0891			0.0889			0.0897

A.4.2 Dataset 2

Tabela 42 – Dataset 2, Semente 128, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait			X					X	X	X
0_cpu12_.idle	X		X	X		X	X	X	X	
0_cpu12_.usr	X		X	X		X	X	X	X	
0_dev252.0_await	X		X	X		X	X	X	X	X
1_dev8.5_avgqu.sz	X		X	X			X		X	
1_frmpg.s			X			X		X	X	
1_tcpsck	X			X			X	X	X	
2_X.dev.sda1_.Iused							X	X	X	
3_cpu22_MHz							X	X	X	
3_iseg.s	X		X	X		X	X	X	X	X
3_kbmemused	X		X	X		X	X	X	X	X
4_cpu20_MHz								X	X	
4_cpu9_.usr	X	X	X	X	X	X	X	X	X	X
4_i151_intr.s			X			X		X	X	
4_tcp6sck							X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		12.2306			12.2306			12.2306		12.2523
Decision Tree		15.1411			15.1411			22.5290		23.2525
Random Forest		12.0036			12.0689			12.0182		12.3673

Tabela 43 – Dataset 2, Semente 128, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X		X	X			X		X	X
0_cpu12_.idle	X	X		X			X	X	X	
0_cpu12_.usr	X		X	X		X	X		X	
0_dev252.0_await	X	X	X	X		X	X	X	X	X
1_dev8.5_avgqu.sz	X			X			X			
1_frmpg.s		X						X		
1_tcpsck	X	X		X	X		X	X	X	
2_X.dev.sda1_.Iused	X	X	X			X	X	X	X	
3_cpu22_MHz	X	X	X				X	X	X	
3_iseg.s	X	X	X	X	X	X	X	X	X	X
3_kbmemused	X	X	X	X		X	X	X	X	X
4_cpu20_MHz				X					X	
4_cpu9_.usr	X	X	X	X		X	X	X	X	X
4_i151_intr.s		X	X			X		X	X	
4_tcp6sck		X					X	X		
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1023			0.1023			0.1023		0.1024
Decision Tree		0.0881			0.0949			0.0881		0.0933
Random Forest		0.0857			0.0859			0.0858		0.0869

A.4.3 Dataset 3

Tabela 44 – Dataset 3, Semente 128, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait		X			X			X	X	
1_cpu6_.sys								X	X	
1_cpu8_.sys									X	
1_dev252.0_tps	X			X			X	X	X	
2_X.dev.sda1_MBfsused				X				X	X	
2_cpu1_.sys	X			X			X	X	X	X
2_i129_intr.s	X			X			X	X	X	
3_all_..usr								X	X	X
3_cpu9_.nice	X	X	X	X	X	X	X	X	X	
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s	X			X			X		X	
5_bwrtn.s	X			X			X	X		
5_cpu10_.soft								X	X	X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X			X			X	X	X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		13.1531			13.1531			13.1531		13.2426
Decision Tree		11.0126			11.0126			21.2463		24.5791
Random Forest		10.9382			10.9382			11.5821		12.9069

Tabela 45 – Dataset 3, Semente 128, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu1_.iowait			X					X	X	
1_cpu6_.sys		X	X			X		X	X	
1_cpu8_.sys								X		
1_dev252.0_tps	X			X			X	X		
2_X.dev.sda1_MBfsused			X					X	X	
2_cpu1_.sys	X			X			X	X		X
2_i129_intr.s	X	X	X	X		X	X	X	X	
3_all_..usr								X		X
3_cpu9_.nice			X		X	X		X	X	
4_cpu3_.idle	X			X			X	X		X
4_passive.s	X			X			X			
5_bwrtn.s								X		
5_cpu10_.soft								X		X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X	X		X			X	X		X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1072			0.1072			0.1072		0.1075
Decision Tree		0.0764			0.0765			0.0875		0.1003
Random Forest		0.0771			0.0771			0.0771		0.0927

A.4.4 Dataset 4

Tabela 46 – Dataset 4, Semente 128, Métrica MSE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft								X		
0_dev252.0_avgrq.sz	X		X	X			X		X	
0_dev252.1_await			X					X	X	
0_eth1_rxkB.s	X		X	X			X	X	X	X
1_bus3_maxpower			X					X	X	
1_dev8.0_await			X					X		
2_X..swpcad							X	X	X	
2_eth1_rxkB.s	X		X	X			X	X	X	
2_i138_intr.s	X		X	X			X	X	X	X
2_sum_intr.s	X		X	X			X	X	X	X
4_X.dev.sda1_.ufsused			X				X	X	X	
4_cpu9_.idle	X	X	X	X	X	X	X	X	X	X
5_active.s	X			X			X	X		
5_bufpg.s								X		
5_cpu12_.soft								X		X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	13.4228			13.4228			13.4228			13.5214
Decision Tree	13.1677			13.1677			23.6636			25.8762
Random Forest	12.4593			13.1648			12.4828			13.1252

Tabela 47 – Dataset 4, Semente 128, Métrica NMAE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva**			**Forward**			**Backward**			**SelectKBest**
	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft	X	X		X			X		X	
0_dev252.0_avgrq.sz	X		X	X		X	X		X	
0_dev252.1_await		X	X		X	X			X	
0_eth1_rxkB.s	X			X		X	X	X	X	X
1_bus3_maxpower	X							X	X	
1_dev8.0_await	X			X		X	X	X	X	
2_X..swpcad							X	X	X	
2_eth1_rxkB.s	X	X	X	X	X	X	X	X	X	
2_i138_intr.s	X	X	X	X	X		X	X		X
2_sum_intr.s		X	X		X	X		X	X	X
4_X.dev.sda1_.ufsused			X				X			
4_cpu9_.idle			X			X		X	X	X
5_active.s	X			X			X	X		
5_bufpg.s						X		X		
5_cpu12_.soft	X			X			X			X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1078			0.1078			0.1078			0.1106
Decision Tree	0.0887			0.0891			0.0943			0.1031
Random Forest	0.0908			0.0909			0.0910			0.0939

A.4.5 Dataset 5

Tabela 48 – Dataset 5, Semente 128, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X		X				X	X	X	
0_file.nr	X		X	X			X	X	X	
1_i128_intr.s			X						X	
2_cpu7_.sys	X		X	X			X	X	X	X
2_eth0_.ifutil						X		X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice			X					X	X	
3_cpu3_MHz								X		
3_eth1_txkB.s	X		X	X			X	X	X	X
4_all_..irq	X	X	X	X	X	X	X	X		
4_cpu16_.sys	X		X	X			X	X	X	X
4_cpu20_.usr	X			X			X	X		X
4_omsg6.s								X		
5_cpu8_MHz			X			X			X	
5_dev252.0_avgqu.sz	X			X			X	X		
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.4707			12.4707			12.4707			12.5199
Decision Tree	12.3779			12.3779			22.1383			23.4596
Random Forest	12.2466			12.3610			12.2711			12.7518

Tabela 49 – Dataset 5, Semente 128, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X						X		X	
0_file.nr	X	X	X	X	X	X	X	X	X	
1_i128_intr.s		X						X		
2_cpu7_.sys	X	X	X	X			X	X	X	X
2_eth0_.ifutil	X	X	X	X		X	X			
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice	X			X			X	X	X	
3_cpu3_MHz		X						X	X	
3_eth1_txkB.s		X						X		X
4_all_..irq	X		X	X			X		X	
4_cpu16_.sys	X	X	X	X			X	X	X	X
4_cpu20_.usr	X	X	X	X		X	X	X	X	X
4_omsg6.s										
5_cpu8_MHz	X						X	X	X	
5_dev252.0_avgqu.sz			X			X		X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1040			0.1040			0.1040			0.1046
Decision Tree	0.0884			0.0903			0.0905			0.0947
Random Forest	0.0882			0.0883			0.0885			0.0916

A.5 Resultados para Semente 527

A.5.1 Dataset 1

Tabela 50 – Dataset 1, Semente 527, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle	X		X	X			X	X	X	
0_dev8.5_tps	X		X	X			X	X	X	X
0_eth1_rxkB.s	X		X	X			X	X	X	X
0_tcp6sck			X			X		X	X	
1_all_..usr			X						X	
1_dev252.0_svctm			X					X	X	
1_igmbq6.s	X		X	X			X	X	X	
2_cpu0_.sys	X	X	X	X	X	X	X	X	X	X
3_iseg.s	X		X	X			X	X	X	X
4_cpu4_.iowait	X			X			X	X		
4_dev8.5_.util			X					X	X	
4_i022_intr.s			X					X	X	
5_cpu0_.soft	X		X	X			X	X	X	
5_cpu2_.idle	X		X	X			X		X	X
5_cpu2_.irq		X	X		X	X		X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression		11.4823			11.4823			11.4823		11.4863
Decision Tree		13.3711			13.3711			21.6772		23.1336
Random Forest		11.7477			13.2998			11.7477		12.2601

Tabela 51 – Dataset 1, Semente 527, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu9_.idle	X		X	X	X	X	X	X	X	
0_dev8.5_tps	X	X	X	X	X	X	X	X	X	X
0_eth1_rxkB.s	X			X			X	X		X
0_tcp6sck		X	X					X	X	
1_all_..usr	X	X		X			X			
1_dev252.0_svctm		X						X	X	
1_igmbq6.s	X	X	X	X		X	X	X		
2_cpu0_.sys	X	X	X	X		X	X	X	X	X
3_iseg.s	X	X	X	X	X	X	X	X	X	X
4_cpu4_.iowait	X	X	X	X			X	X	X	
4_dev8.5_.util			X			X		X	X	
4_i022_intr.s								X		
5_cpu0_.soft	X	X		X	X		X	X	X	
5_cpu2_.idle	X			X			X			X
5_cpu2_.irq		X	X					X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression		0.1006			0.1006			0.1006		0.1007
Decision Tree		0.0868			0.0897			0.0889		0.0933
Random Forest		0.0868			0.0868			0.0870		0.0890

A.5.2 Dataset 2

Tabela 52 – Dataset 2, Semente 527, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X		X	X			X	X	X	X
0_cpu12_.idle			X						X	
0_cpu12_.usr	X			X			X			
0_dev252.0_await	X		X	X			X	X	X	X
1_dev8.5_avgqu.sz	X	X	X	X	X	X	X	X	X	
1_frmpg.s	X		X	X			X		X	
1_tcpsck			X					X	X	
2_X.dev.sda1_.Iused						X		X	X	
3_cpu22_MHz								X	X	
3_iseg.s	X		X	X			X	X	X	X
3_kbmemused	X		X	X			X	X	X	X
4_cpu20_MHz	X		X				X	X		
4_cpu9_.usr	X		X	X			X	X	X	X
4_i151_intr.s								X		
4_tcp6sck							X	X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	11.5936			11.5936			11.5936			11.6057
Decision Tree	13.8215			13.8215			22.5975			23.3944
Random Forest	11.7234			13.8219			11.7522			12.1129

Tabela 53 – Dataset 2, Semente 527, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu10_.iowait	X	X		X	X	X	X	X	X	X
0_cpu12_.idle	X			X			X			
0_cpu12_.usr	X	X	X	X		X	X		X	
0_dev252.0_await		X	X			X		X	X	X
1_dev8.5_avgqu.sz	X			X			X	X		
1_frmpg.s										
1_tcpsck	X		X	X			X	X	X	
2_X.dev.sda1_.Iused		X	X					X		
3_cpu22_MHz		X					X	X	X	
3_iseg.s	X		X	X	X	X	X	X	X	X
3_kbmemused	X	X	X	X		X	X	X	X	X
4_cpu20_MHz		X		X			X	X	X	
4_cpu9_.usr	X		X	X		X	X	X	X	X
4_i151_intr.s		X	X		X	X		X	X	
4_tcp6sck		X	X				X	X	X	
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1006			0.1006			0.1006			0.1009
Decision Tree	0.0884			0.0932			0.0907			0.0930
Random Forest	0.0860			0.0863			0.0862			0.0867

A.5.3 Dataset 3

Tabela 54 – Dataset 3, Semente 527, Métrica MSE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva** LR	DT	RF	**Forward** LR	DT	RF	**Backward** LR	DT	RF	**SelectKBest**
0_cpu1_.iowait			X			X			X	
1_cpu6_.sys			X					X	X	
1_cpu8_.sys	X			X			X	X	X	
1_dev252.0_tps	X			X			X	X	X	
2_X.dev.sda1_MBfsused						X		X	X	
2_cpu1_.sys	X			X			X	X	X	X
2_i129_intr.s	X			X			X	X		
3_all_..usr	X			X			X	X	X	X
3_cpu9_.nice	X		X	X		X	X	X		
4_cpu3_.idle	X			X			X	X	X	X
4_passive.s								X	X	
5_bwrtn.s								X	X	
5_cpu10_.soft	X			X			X	X	X	X
5_kbbuffers	X	X	X	X	X	X	X	X	X	
5_temp1_degC	X			X			X		X	X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.2772			12.2772			12.2772			12.2979
Decision Tree	10.4961			10.4961			20.1466			23.7795
Random Forest	10.3762			10.3768			11.1051			13.1401

Tabela 55 – Dataset 3, Semente 527, Métrica NMAE.

SELEÇÃO DE FEATURES										
Feature	**Busca Exaustiva** LR	DT	RF	**Forward** LR	DT	RF	**Backward** LR	DT	RF	**SelectKBest**
0_cpu1_.iowait	X		X	X		X	X		X	
1_cpu6_.sys								X		
1_cpu8_.sys	X			X			X	X		
1_dev252.0_tps								X		
2_X.dev.sda1_MBfsused								X		
2_cpu1_.sys	X			X			X	X		X
2_i129_intr.s								X		
3_all_..usr	X			X			X	X		X
3_cpu9_.nice			X			X		X	X	
4_cpu3_.idle	X			X			X	X		X
4_passive.s								X		
5_bwrtn.s								X		
5_cpu10_.soft	X		X	X		X	X	X	X	X
5_kbbuffers		X	X		X	X		X	X	
5_temp1_degC	X			X			X			X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1059			0.1059			0.1059			0.1059
Decision Tree	0.0747			0.0747			0.0837			0.0963
Random Forest	0.0749			0.0749			0.0749			0.0931

A.5.4 Dataset 4

Tabela 56 – Dataset 4, Semente 527, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft	X		X	X			X	X	X	
0_dev252.0_avgrq.sz			X						X	
0_dev252.1_await		X	X		X			X	X	
0_eth1_rxkB.s			X					X	X	X
1_bus3_maxpower							X	X	X	
1_dev8.0_await			X					X	X	
2_X..swpcad							X	X	X	
2_eth1_rxkB.s	X		X	X			X	X	X	
2_i138_intr.s	X		X	X			X	X	X	X
2_sum_intr.s	X		X	X			X	X	X	X
4_X.dev.sda1_.ufsused							X	X	X	
4_cpu9_.idle	X	X	X	X	X	X	X	X	X	X
5_active.s	X		X	X			X	X	X	
5_bufpg.s	X			X			X	X	X	
5_cpu12_.soft	X			X			X	X		X
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	12.4544			12.4544			12.4544			12.5246
Decision Tree	12.7441			12.7441			23.4259			24.8370
Random Forest	11.9257			12.6226			11.9573			12.4580

Tabela 57 – Dataset 4, Semente 527, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu18_.soft		X	X			X		X	X	
0_dev252.0_avgrq.sz		X	X		X	X		X	X	
0_dev252.1_await	X		X	X	X		X	X	X	
0_eth1_rxkB.s			X			X		X	X	X
1_bus3_maxpower		X	X			X			X	
1_dev8.0_await			X			X		X	X	
2_X..swpcad			X						X	
2_eth1_rxkB.s		X	X			X			X	
2_i138_intr.s	X			X			X	X	X	X
2_sum_intr.s		X	X			X		X	X	X
4_X.dev.sda1_.ufsused			X				X	X	X	
4_cpu9_.idle		X	X		X	X		X	X	X
5_active.s	X	X		X			X	X	X	
5_bufpg.s	X			X		X	X	X	X	
5_cpu12_.soft	X	X		X			X	X		X
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1044			0.1044			0.1044			0.1088
Decision Tree	0.0899			0.0932			0.0928			0.0989
Random Forest	0.0895			0.0895			0.0896			0.0919

A.5.5 Dataset 5

Tabela 58 – Dataset 5, Semente 527, Métrica MSE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz	X							X	X	
0_file.nr	X			X			X			
1_i128_intr.s								X	X	
2_cpu7_.sys	X			X			X	X	X	X
2_eth0_.ifutil								X	X	
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice								X	X	
3_cpu3_MHz							X	X	X	
3_eth1_txkB.s								X	X	X
4_all_..irq	X	X	X	X	X	X	X	X		
4_cpu16_.sys	X			X			X	X	X	X
4_cpu20_.usr	X			X			X	X	X	X
4_omsg6.s	X			X			X	X	X	
5_cpu8_MHz							X		X	
5_dev252.0_avgqu.sz								X	X	
RESULTADOS DOS MODELOS (MSE)										
Linear Regression	11.7327			11.7327			11.7327			11.8120
Decision Tree	11.3908			11.3908			21.6633			22.7602
Random Forest	11.3841			11.3841			11.8714			11.7619

Tabela 59 – Dataset 5, Semente 527, Métrica NMAE.

SELEÇÃO DE FEATURES										
	Busca Exaustiva			**Forward**			**Backward**			**SelectKBest**
Feature	LR	DT	RF	LR	DT	RF	LR	DT	RF	
0_cpu22_MHz			X						X	
0_file.nr	X		X	X			X		X	
1_i128_intr.s								X		
2_cpu7_.sys	X			X			X	X		X
2_eth0_.ifutil		X								
2_runq.sz	X	X	X	X	X	X	X	X	X	X
3_cpu13_.nice								X		
3_cpu3_MHz							X	X	X	
3_eth1_txkB.s	X	X		X		X	X	X	X	X
4_all_..irq	X	X	X	X	X	X	X	X	X	
4_cpu16_.sys	X	X	X	X		X	X	X	X	X
4_cpu20_.usr	X			X		X	X	X	X	X
4_omsg6.s								X	X	
5_cpu8_MHz							X	X	X	
5_dev252.0_avgqu.sz		X	X					X		
RESULTADOS DOS MODELOS (NMAE)										
Linear Regression	0.1030			0.1030			0.1030			0.1035
Decision Tree	0.0860			0.0885			0.0895			0.0927
Random Forest	0.0870			0.0874			0.0875			0.0877