

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Detecção de *outliers* para o aumento da eficácia na fiscalização de contratos: um estudo de caso com a contratação de serviços de computação em nuvem

Victor Diego Medeiros Lino

Trabalho de Conclusão de Curso - MBA em Ciência de Dados (CEMEAI)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Victor Diego Medeiros Lino

Detecção de outliers para o aumento da eficácia na fiscalização de contratos: um estudo de caso com a contratação de serviços de computação em nuvem

Trabalho de conclusão de curso apresentado ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para conclusão do MBA em Ciência de Dados. Especialista em Ciências de Dados.

Área de Concentração: Ciências de Dados

Orientador: Prof. Dr. Jó Ueyama

USP – São Carlos
Janeiro de 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

L758d LINO, VICTOR DIEGO MEDEIROS
Detecção de outliers para o aumento da eficácia
na fiscalização de contratos: um estudo de caso com
a contratação de serviços de computação em nuvem /
VICTOR DIEGO MEDEIROS LINO; orientador JÓ UNEYAMA. --
São Carlos, 2021.
70 p.

Trabalho de conclusão de curso (MBA em Ciência
de Dados) -- Instituto de Ciências Matemáticas e de
Computação, Universidade de São Paulo, 2021.

1. Detecção de Outlier. 2. Série Temporal. 3.
Computação em Nuvem. 4. Fiscalização de Contrato. I.
UEYAMA, JÓ, orient. II. Título.

À minha amada esposa Denise, cujo incentivo e apoio incondicional foram essenciais para enfrentar mais esse desafio. Obrigado por sempre caminhar ao meu lado fazendo a jornada ser mais leve, meu amor.

RESUMO

LINO, V. D. M. **Detecção de *outliers* para o aumento da eficácia na fiscalização de contratos: um estudo de caso com a contratação de serviços de computação em nuvem.** 2021. 70 p. Trabalho de conclusão de curso (Conclusão em Ciências de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

A fiscalização de contratos de Tecnologia da Informação é um grande desafio, principalmente para aqueles nos quais uma grande quantidade de dados é gerada. Em grande parte desses contratos, são disponibilizados relatórios diários com a quantidade de cada recurso consumido, gerando séries temporais que necessitam passar pelo escrutínio do fiscal. As técnicas geralmente utilizadas na fiscalização de tais relatórios limitam-se a análise de valores extremos, que pode deixar de detectar pontos de interesse importantes para análise do fiscal. Algoritmos de detecção de outliers em séries temporais baseadas em abordagem estatística já são estudados há muitos anos e mostram resultados consolidados. Com o aumento do poder computacional nas últimas décadas, abordagens de aprendizado de máquina e até mesmo de redes neurais vem ganhando destaque nessa área de pesquisa. Este trabalho realiza um estudo de caso em 9 séries temporais de contrato de computação em nuvem, utilizando as três abordagens de detecção de *outliers* supracitadas, confirmando que, para o caso em análise, essas estratégias são mais eficazes. O modelo de redes neurais MLP apresentou os melhores resultados, com eficácia 64% maior que o método de Valores Extremos.

Palavras-chave: Detecção de *Outlier*, Série Temporal, Computação em Nuvem, Fiscalização de Contrato.

ABSTRACT

LINO, V. D. M. **Outlier detection for increase effectiveness in contracts inspection: a case study on cloud computing service.** 2021. 70 p. Trabalho de conclusão de curso (Conclusão em Ciências de Dados) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

The inspection of Information Technology contracts is a major challenge, especially for those in which a large volume of data is generated. In most of these contracts, daily reports are available with the amount of each resource consumed, generating time series that need to pass the scrutiny of the contract inspector. The techniques generally used in the inspection of such reports are limited to the analysis of extreme values, which may fail to detect important points of interest for the analysis of the inspector. Algorithms for detecting outliers in time series based on statistical approach have been studied for many years and show consolidated results. With the increase of computational power in the last decades, machine learning approaches and even neural networks have been gaining prominence in this area of research. This work performs a case study in 9 cloud computing time series, using the three aforementioned outlier detection approaches, confirming that, for the case under analysis, these strategies are more effective. The MLP neural network model showed the best results, with 64% more effectiveness than the Extreme Values method.

Keywords: Outlier Detection, Time Series, Cloud Computing, Contract Inspection.

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de dados com ruído	22
Figura 2 – Diferença de novidade e <i>outlier</i>	22
Figura 3 – À esquerda <i>outliers</i> pontuais; à direira <i>outliers</i> coletivos	23
Figura 4 – Exemplos de séries temporais	25
Figura 5 – Característica de uma série temporal	26
Figura 6 – Abordagens fundamentais de detecção de <i>outliers</i>	28
Figura 7 – Categorias de detecção de <i>outliers</i> em séries temporais	28
Figura 8 – Etapas desenvolvidas no trabalho	33
Figura 9 – Detalhamento dos itens do Pregão nº 29/2018	34
Figura 10 – Visualização do crescimento dos dados gerados	35
Figura 11 – Previsão do total de linhas geradas em Março de 2021	35
Figura 12 – Visualização dos dados no Jupyter Notebook	36
Figura 13 – Verificação do tipo de cada coluna e valores nulos	37
Figura 14 – Quantidade de linhas em cada conta	37
Figura 15 – Os 5 produtos mais utilizados	38
Figura 16 – Exemplo de série temporal obtida	39
Figura 17 – Exemplo de série temporal transformada	39
Figura 18 – Séries temporais resultantes após etapa de Pré-Processamento	40
Figura 19 – Pontos de interesse apontados pelo especialista	41
Figura 20 – Exemplo de Precisão e Revocação	44
Figura 21 – Exemplo de diferentes resultados de AUC-ROC	45
Figura 22 – Séries temporais com resultados opostos	47
Figura 23 – Limiar de separação claro entre observações normais e PdI	47
Figura 24 – Pontos de Interesse detectados pelos modelos Valores Extremos e SARIMA	48
Figura 25 – Pontos de Interesse detectados pelo modelo MLP	50
Figura 26 – Média da Medida- <i>F</i> ₁ para cada modelo	51
Figura 27 – Pontos de Interesse na série Amazon Simple Storage Service - Requests-Tier1	62
Figura 28 – Pontos de Interesse na série Amazon Simple Storage Service - Requests-Tier2	63
Figura 29 – Pontos de Interesse na série Amazon Simple Storage Service - SAE1-USE1-AWS-Out-Bytes	64
Figura 30 – Pontos de Interesse na série EC2 – Other - SAE1-DataTransfer-Regional-Bytes	65
Figura 31 – Pontos de Interesse na série EC2 – Other - SAE1-EU-AWS-In-Bytes	66
Figura 32 – Pontos de Interesse na série EC2 – Other - SAE1-NatGateway-Bytes	67

Figura 33 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-DataTransfer-In-Bytes	68
Figura 34 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-USW2-AWS-In-Bytes	69
Figura 35 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-USE1-AWS-In-Bytes	70

LISTA DE TABELAS

Tabela 1 – Abordagens e modelos utilizados nesse trabalho	29
Tabela 2 – Principais trabalhos de DdO e atendimento de pontos-chave	31
Tabela 3 – Descrição dos produtos e os 3 serviços escolhidos	38
Tabela 4 – Características das séries temporais	42
Tabela 5 – Parâmetros dos modelos da abordagem Estatística	42
Tabela 6 – Parâmetros dos modelos da abordagem de Aprendizado de Máquina	43
Tabela 7 – Parâmetros dos modelos da abordagem de Redes Neurais	43
Tabela 8 – AUC-ROC para cada série temporal em cada modelo	46
Tabela 9 – Medida- F_1 para cada série temporal em cada modelo	49
Tabela 10 – Médias das Métricas Medida- F_1 e AUC-ROC	51

LISTA DE ABREVIATURAS E SIGLAS

AUC-ROC	Área sob a Curva ROC
CGU	Controladoria-Geral da União
COVID-19	<i>Coronavirus disease 2019</i>
CSV	<i>Comma-separated values</i>
DBSCAN	<i>Density-based spatial clustering of applications with noise</i>
DdO	Detecção de <i>Outlier</i>
iForest	<i>Isolation Forest</i>
LOF	<i>Local outlier factor</i>
LSTM	<i>Long short-term memory</i>
MLP	<i>Multilayer perceptron</i>
PdI	Pontos de Interesse
ROC	<i>Receiver operating characteristic</i>
SARIMA	<i>Seasonal autoregressive integrated moving average</i>
TES	<i>Triple exponential smoothing</i>
TFP	Taxa de falsos positivos
TIC	Tecnologia da Informação e Comunicação
TVP	Taxa de verdadeiros positivos

SUMÁRIO

1	INTRODUÇÃO	17
1.1	Motivação	18
1.2	Objetivos	18
1.3	Metodologia	19
1.4	Organização do Trabalho	19
2	REVISÃO BIBLIOGRÁFICA	21
2.1	Conceitos Fundamentais	21
2.1.1	<i>Definição de outlier, ruído e novidade</i>	21
2.1.2	<i>Tipos de outliers</i>	22
2.1.3	<i>Cenários de detecção de outlier</i>	24
2.1.4	<i>Séries Temporais</i>	25
2.2	Detecção de outliers	27
2.2.1	<i>Abordagens de detecção de outliers</i>	27
2.2.2	<i>Abordagens de detecção de outliers em séries temporais</i>	28
2.3	Trabalhos relacionados e contribuição desse trabalho	29
3	MODELO PROPOSTO	33
3.1	Atividades Realizadas	33
3.1.1	<i>Coleta da base de dados</i>	34
3.1.2	<i>Análise Descritiva dos Dados</i>	36
3.1.3	<i>Pré-Processamento dos Dados</i>	39
3.1.3.1	<i>Transformação do campo de data em modelos espaciais</i>	39
3.1.3.2	<i>Normalização dos dados</i>	40
3.1.3.3	<i>Séries Temporais Resultantes</i>	40
3.1.4	Modelagem dos dados	40
3.1.4.1	<i>Parâmetros utilizados nos algoritmos</i>	42
3.1.5	Avaliação dos Modelos	43
3.1.5.1	<i>Medida-F</i>	44
3.1.5.2	<i>Área sob a curva ROC</i>	45
3.2	Resultados Obtidos	45
4	CONCLUSÃO	53
4.1	Conclusão	53

4.2	Dificuldades, Limitações e Trabalhos Futuros	54
REFERÊNCIAS		55
APÊNDICE A	PONTOS DE INTERESSE DETECTADOS POR CADA MODELO	61



INTRODUÇÃO

As despesas com contratos de Tecnologia da Informação e Comunicação (TIC) são responsáveis por parcela significativa das despesas do Governo Federal. Segundo o Portal da Transparência, em 2019, o Executivo Federal pagou cerca de cinco bilhões de reais em Serviços de TIC ([CGU, 2020](#)). Sendo assim, torna-se essencial garantir que os gastos nessa área sejam realizados de maneira eficaz.

A principal forma com a qual a Administração Pública adquire bens e serviços é por meio da licitação pública, conforme determina a Constituição Federal ([BRASIL, 1988](#)) . Após a conclusão do processo licitatório e assinatura de contrato entre o órgão público e a empresa vencedora do certame, servidores públicos são designados para fiscalização e liquidação das faturas emitidas ([BRASIL, 1993; MINISTÉRIO DA ECONOMIA, 2019b](#)).

Serviços de TIC como conexões de dados, hospedagem de equipamentos em centro de processamento de dados e serviços de computação em nuvem são monitorados constantemente para garantir que os níveis de serviço acordados no contrato sejam cumpridos pelos fornecedores. Assim, na fase de liquidação da fatura, os fiscais recebem o detalhamento dos serviços prestados e, em especial para serviços de TIC, esse detalhamento pode conter centenas ou milhares de dados que precisam ser auditados com escrutínio, para garantir que o valor pago pelo governo é realmente o valor devido.

Na maioria dos casos, a fiscalização dos contratos de TIC é realizada por meio de planilhas eletrônicas para identificação de Pontos de Interesse (PdI) e possíveis erros. Para tanto, a análise de valores extremos é uma técnica comumente utilizadas para filtrar o universo dos dados fiscalizados. Essa metodologia de fiscalização é pouco produtiva, demanda tempo considerável de trabalho do servidor para confirmação e aumentam as chances de haver pagamento por serviços não prestados ou superfaturados. Deste modo, a detecção de **pontos de interesse** de forma mais eficaz e automatizada é essencial para promover o uso apropriado dos recursos públicos.

1.1 Motivação

A contratação de computação em nuvem visando a substituição de compra e sustentação de equipamentos de TIC nos órgãos do Governo Federal é algo recente. Esse movimento ganhou força a partir da publicação da Instrução Normativa nº 1 de 2019 do Ministério da Economia ([MINISTÉRIO DA ECONOMIA, 2019a](#)), a qual determina que a criação ou ampliação de infraestrutura de centro de processamento de dados deverá ser realizada por meio de contratação de serviços de computação em nuvem, exceto em casos justificados. Se por um lado a expectativa é a diminuição do número de contratos administrativos, por meio da concentração de vários serviços em um só acordo de hospedagem de serviços em nuvem, por outro lado haverá o aumento da quantidade de itens que o fiscal terá que validar para atestar a efetiva prestação dos serviços pelos fornecedores. Isso reforça a necessidade de um método automatizado de seleção e evidenciação de pontos de interesse (ou *outliers*) para análise mais criteriosa por parte do fiscal do contrato.

Além disso, há um projeto no Ministério da Economia para padronizar e centralizar a contratação de serviços em nuvem (Consulta Pública nº 4/2020), já que 140 órgãos e entidades demonstraram em seu planejamento de contratações de 2020 o interesse em adquirir serviços em nuvem ([MINISTÉRIO DA ECONOMIA, 2020](#)). Caso essa iniciativa obtenha sucesso, as ferramentas de detecção de pontos de interesse para contratos de serviço em nuvem proposto por esse trabalho poderão ser compartilhadas com outros órgãos, aumentando significativamente o benefício esperado.

1.2 Objetivos

Existem inúmeros modelos e abordagens para se detectar *outliers*, sendo que as vantagens do uso de cada um deles variam de acordo com o conjunto de dados que se está trabalhando ([MANDHARE; IDATE, 2017](#)). Por esse motivo, torna-se necessário um estudo centrado em analisar e testar os principais métodos de detecção de *outliers* para detecção de pontos de interesse em faturamento de contratos de serviços em nuvem.

O propósito desse trabalho é indicar um algoritmo de detecção de *outlier* mais eficiente do que o método de valores extremos geralmente utilizado na fiscalização de contratos de serviço de nuvem no Governo Federal brasileiro, auxiliando os fiscais técnicos na tarefa de inspeção de pontos de interesse e detecção de erros de cobrança. Como consequência, isso pode gerar economia financeira para a Administração Pública, melhoria do serviço através da análise de anomalias que possam estar ocorrendo na prestação do serviço pela empresa fornecedora, além do direcionamento do tempo do fiscal para outras atividades igualmente importantes

1.3 Metodologia

Primeiramente, será feito um levantamento bibliográfico acerca das principais abordagens de detecção de *outliers* em séries temporais. Em seguida, serão selecionados os principais modelos em cada abordagem para serem testados em dados reais históricos de faturamento de contratos de computação em nuvem e o desempenho de cada um deles será avaliado através de gráficos e tabelas comparativas. Na sequência, o modelo de melhor desempenho será identificado e comparado com a técnica de valores extremos, buscando comprovar que o modelo apontado é mais eficiente que o processo atual de identificação de pontos de interesse.

1.4 Organização do Trabalho

O presente trabalho está organizado da seguinte maneira:

- **Capítulo 2** – São apresentados conceitos fundamentais em detecção de *outliers* e principais abordagens de detecção de *outliers* em séries temporais;
- **Capítulo 3** – Aplicação dos principais modelos de detecção de *outliers* em dados reais históricos de faturamento de serviço de computação em nuvem e comparação dos resultados obtidos para escolha do melhor método;
- **Capítulo 4** – Apresenta-se a conclusão do trabalho desenvolvido.



REVISÃO BIBLIOGRÁFICA

Neste capítulo são apresentados os conceitos fundamentais de *outliers*, séries temporais e detecção de *outliers* em séries temporais.

2.1 Conceitos Fundamentais

2.1.1 Definição de *outlier*, ruído e novidade

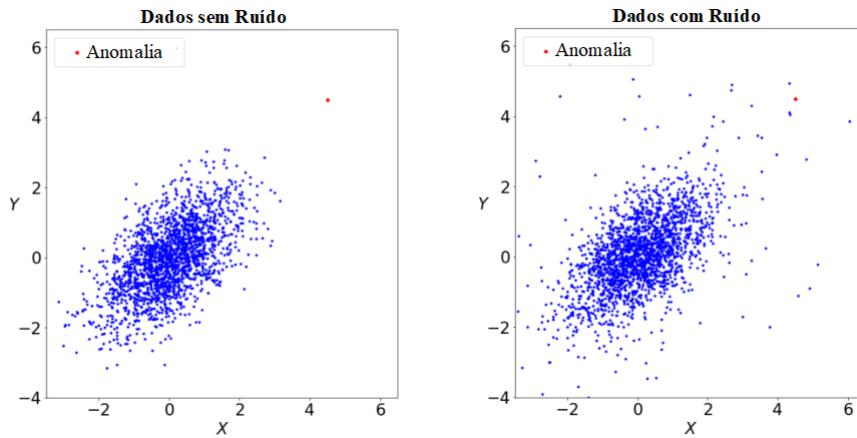
A área de Tecnologia da Informação gera e monitora a cada segundo uma grande quantidade de dados. Dentro desse conjunto de dados, configura-se fonte valiosa de informação a análise de observações que se destacam muito dos demais dados. Hawkins definiu esses pontos fora da curva como *outliers*:

Um *outlier* é uma observação que se desvia muito das outras observações a ponto de levantar suspeitas de que ela foi gerada por um mecanismo diferente. . ([HAWKINS, 1980](#))

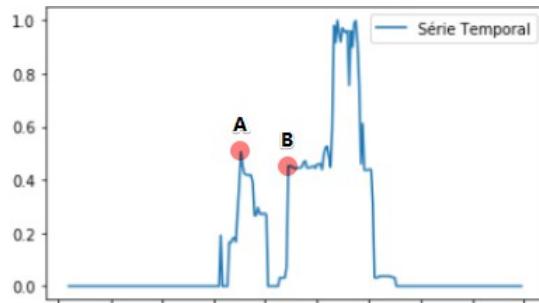
Nesse contexto, é possível identificar diversas áreas de pesquisa que possuem como objetivo a detecção de *outliers*: detecção de intrusão, detecção de fraudes em cartão de crédito, sensores de redes, diagnósticos médicos, entre outros ([CHANDOLA; BANERJEE; KUMAR, 2009](#)). Independente do campo em que a detecção de *outliers* é aplicada, em comum está o fato de que a maioria dos dados se encaixam no perfil considerado normal e menor número de instâncias se apresentam como anomalias, que serão objetos de análise posterior. Em termos estatísticos, a distribuição dos dados anômalos se distingue perceptivelmente dos demais dados.

Uma diferenciação importante a ser feita é entre ruído e *outlier*: o primeiro se trata de uma variação natural dos dados, um descolamento do modelo padrão que pode ser ignorado pelo observador; já o segundo apresenta uma variação significativa, contendo informação relevante para o contexto estudado.

Figura 1 – Exemplo de dados com ruído

Fonte: Adaptada de [Braei e Wagner \(2020\)](#).

Vários trabalhos também fazem distinção entre *outlier* e novidade ([BRAEI; WAGNER, 2020](#); [CHANDOLA; BANERJEE; KUMAR, 2009](#); [HODGE; AUSTIN, 2004](#)), sendo este último conceito definido como pontos nunca vistos no conjunto de dados até a observação atual, mas que após a primeira aparição são considerados normais e deixam de ser ponto de interesse para a análise.

Figura 2 – Diferença de novidade e *outlier*

Fonte: Elaborada pelo autor.

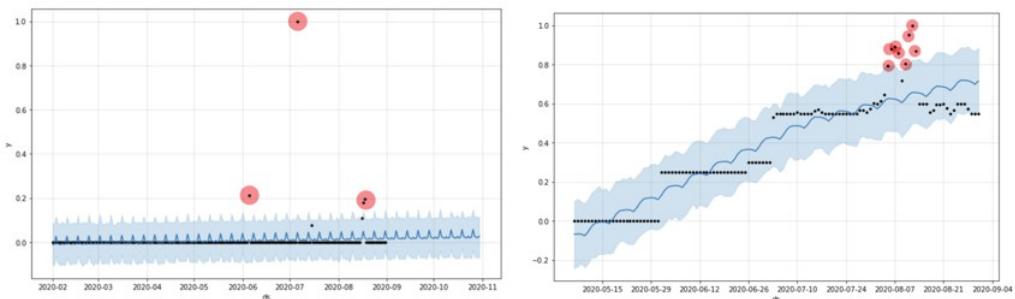
No contexto desse trabalho, **consideram-se equivalentes os termos *outlier*, anomalia, pontos de interesse e novidade.**

2.1.2 Tipos de outliers

Independente do cenário de análise, os métodos geralmente utilizam alguns critérios básicos para categorização de possíveis *outliers*. Em relação ao número de observações para se afirmar a existência de anomalia, [Chandola, Banerjee e Kumar \(2009\)](#) apresentam os seguintes tipos:

- **Outlier Pontual** – ocorre quando uma única instância é considerada uma anomalia pelo fato de existir discrepância significativa em relação ao restante dos dados. É o tipo mais simples de anomalia e foco da maioria das pesquisas no campo de detecção de *outlier*;
- **Outlier Coletivo** – caracterizado quando um conjunto de instâncias apresenta comportamento anômalo se comparado com o restante do *dataset* e uma observação isolada desse subconjunto não caracteriza anormalidade. Exemplo desse conceito é o processo de autenticação por senha: um ou dois erros podem ocorrer por falha na digitação e o sistema caracterizar tal fato como normal, mas 3 tentativas frustradas já levantam suspeitas e devem ser investigadas.;
- **Outlier Contextual** – alguns pontos podem ser identificados como normais ou *outliers* dependendo da circunstância ou contexto, por exemplo: ao se realizar a categorização de dados de temperatura de uma região brasileira, pontos atingindo 0 grau Celsius na região Sul do Brasil podem ser considerados dados normais, enquanto na região Nordeste estes mesmos dados poderiam ser classificados como anomalias.

Figura 3 – À esquerda *outliers* pontuais; à direira *outliers* coletivos



Fonte: Elaborada pelo autor.

No contexto de *outlier* pontual, existem duas categorizações importantes no que se refere ao escopo: (1) ***outlier* global**, que considera todas anomalias da mesma maneira, independente da região da distribuição dos dados (KNORR; NG, 1998). Nesse caso, pressupõe-se que existe somente um mecanismo de geração dos dados e o conjunto de referência é global; (2) ***outlier* local**, que aprimora a análise de detecção de *outlier* pontual e considera somente as diferenças entre a anomalia e seus vizinhos mais próximos (BREUNIG *et al.*, 2000), admitindo a possibilidade de existir mais de um mecanismo de geração de dados e vários conjuntos de dados de referência (AGGARWAL, 2017a).

Conhecer o tipo de *outlier* existente na base de dados é vantajoso para escolher o melhor método de detecção de anomalia, pois algumas abordagens funcionam melhor na detecção de *outliers* pontuais, enquanto outras performam melhor encontrando *outliers* coletivos ou contextuais. Como no cenário real de fiscalização de contratos não se sabe de antemão o tipo

de dados que serão analisados, buscou-se selecionar séries temporais com comportamentos mais variados possíveis, pretendendo encontrar os três tipos de *outliers*: **pontual**, **coletivo** e **contextual**. Na [Figura 3](#) são apresentados exemplos de *outliers* pontuais e contextuais.

2.1.3 Cenários de detecção de outlier

Encontrar anomalias é um processo custoso e desafiador, principalmente em campos onde se busca inovação. Isso porque a raridade de ocorrência de *outliers* e inexistência de dados categorizados inviabilizam uma comparação entre o conjunto de dados de estudo e o objetivo a ser alcançado. Nesse contexto, em relação à existência prévia de dados categorizados, pode-se dividir os métodos de detecção de *outliers* em três cenários:

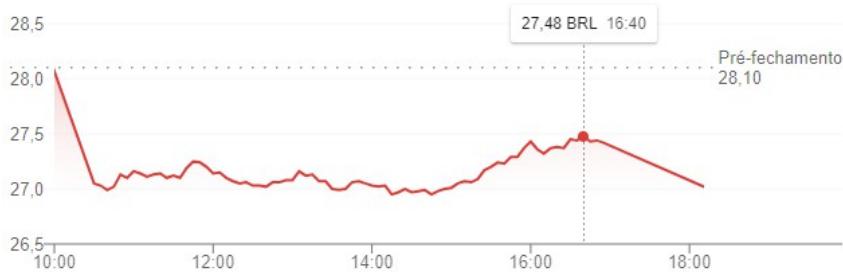
- **Não-supervisionado** – nesse cenário não há exemplos anteriores de anormalidade e os dados não estão diferenciados entre *outliers* ou normais. É uma abordagem mais genérica e comumente usada para remoção de ruído do *dataset* e detecção de anomalias na fase exploratória da análise. Como não há a necessidade de dados de treino classificados é o tipo mais comumente utilizado. Geralmente a resposta desse método não é apresentada de forma direta, indicando expressamente quais dados são *outliers*, mas atribui uma **pontuação/score**, que quantifica numericamente o quanto cada dado é uma anomalia, ranqueando cada ponto do conjunto de dados. Dessa forma, o resultado é disponibilizado para o especialista da área de estudo para análise e definição do valor limite entre o que é ruído e o que é anomalia;
- **Semi-supervisionado** – semelhante ao cenário não-supervisionado, mas há certo nível de interferência ou retroalimentação humana em alguma fase do processo. Nesse cenário os dados de teste são livres de *outliers* e um modelo é treinado, detectando qualquer ponto que se desvie do comportamento dos dados normais iniciais;
- **Supervisionado** – cenário em que há acesso a exemplos anteriores de anomalias, o que torna o método mais eficaz em comparação aos dois cenários anteriores. Por outro lado, como a comparação é o núcleo do método, pode ser mais custoso para obtenção de exemplos anteriores e mais exigente em termos computacionais. Os resultados geralmente são apresentados de **forma binária**, indicando de forma direta se cada dado é um *outlier* ou não.

As circunstâncias enfrentadas na maioria dos casos de fiscalização de contratos é de inexistência de dados rotulados, e o contexto do estudo de caso desse trabalho não é exceção. Entretanto, para podermos classificar de maneira objetiva o desempenho dos modelos comparados, um especialista da área de TIC irá rotular os pontos de interesse dos dados selecionados, transformando contexto de estudo em um **cenário supervisionado**.

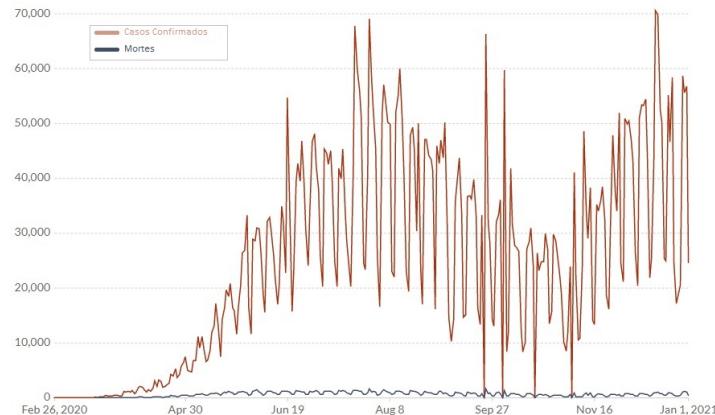
2.1.4 Séries Temporais

Uma série temporal é um conjunto de observações realizadas de forma sequencial e ao longo do tempo. Nos casos em que essas observações são realizadas de forma equidistante no tempo, forma-se um conjunto de dados discreto. Apesar do nome, o domínio do tempo pode ser substituído por outra variável como profundidade ou espaço. Exemplos de séries temporais são valores os a cada hora do preço de uma ação, valores diários de casos confirmados e de morte por *Coronavirus disease 2019* (COVID-19) no Brasil ou o volume acumulado de chuva por mês em determinada região .

Figura 4 – Exemplos de séries temporais



(a) Cotação da ação PETR4



(b) Valores diários de casos confirmados e de morte por COVID-19 no Brasil

Fonte: Elaborada pelo autor.

O exemplo da Figura 4a é considerado **univariado**, pois analisamos somente uma característica ou parâmetro na linha do tempo (preço da ação). Já a série da Figura 4b é classificada como **multivariada**, pois duas características (casos confirmados e mortes) são analisadas simultaneamente para cada unidade de tempo.

Ressalta-se que a característica mais relevante de uma série temporal é a dependência de uma observação com seus vizinhos, limitando o universo de modelos de detecção de anomalias que podem ser utilizados, pois dados correlacionados impõe desafios adicionais e demandam técnicas específicas para sua análise. Conforme destacado por (EHLERS, 2009): “Enquanto em

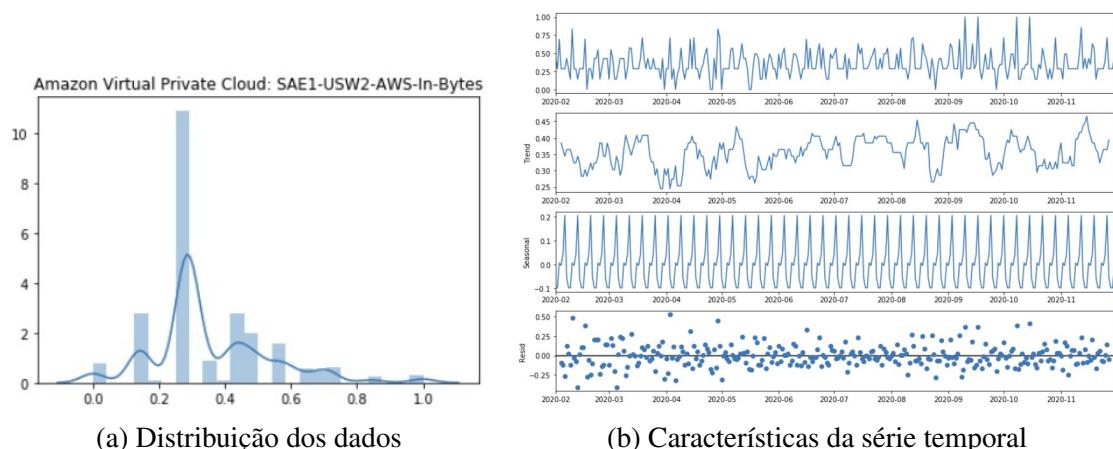
modelos de regressão por exemplo a ordem das observações é irrelevante para a análise, em séries temporais a ordem dos dados é crucial.”

Outros padrões presentes em séries temporais e especialmente relevantes para os métodos estatísticos mais utilizados em detecção de anomalias são:

- **Tendência** – característica presente quando a média dos valores no decorrer do tempo não é constante, apresentando inclinação positiva ou negativa, podendo ser linear ou não-linear;
- **Sazonalidade** – comportamento periódico recorrente das flutuações dos dados ao longo do tempo. Os principais tipos de sazonalidade são a aditiva e multiplicativa: no primeiro tipo as flutuações sazonais são aproximadamente constantes independentemente do nível global da série; já no segundo, há correlação entre o nível global da série e a amplitude da flutuação;
- **Estacionariedade** – qualidade que indica que as propriedades estatísticas como média e variância não variam ao longo do tempo, se comportando aleatoriamente em torno de uma média constante. Esse comportamento facilita o processamento de algoritmos de análise de séries temporais, mas é raro na maioria das séries encontradas na prática. Várias ferramentas e modelos de análise de séries temporais pressupõem que os dados são estacionários.

A base de dados utilizada nesse trabalho é um conjunto de **séries temporais discretas univariadas rotuladas com tendência variada, não-estacionárias e sazonalidade semanal**.

Figura 5 – Característica de uma série temporal



Fonte: Elaborada pelo autor.

2.2 Detecção de outliers

Na literatura a Detecção de *Outlier* (DdO) é definida como a tarefa de identificação de instâncias raras que se diferenciam consideravelmente da maioria do conjunto de dados (HODGE; AUSTIN, 2004; ZIMEK; SCHUBERT; KRIEGEL, 2012). A intensidade do desvio entre cada objeto e a distribuição geral do *dataset* é considerada como medida de força da anomalia, a qual é designada como **score de anomalia** (BRAEI; WAGNER, 2020). Várias abordagens de detecção de *outlier* buscam determinar um **valor limite** baseado nas propriedades do conjunto analisado para classificar cada instância como anomalia ou normal.

As principais formas de atribuir o *score* de anomalia em DdO é através da análise estatística do conjunto de dados ou pela análise de proximidade ou densidade entre os pontos no domínio do espaço. Nessas duas abordagens o principal pressuposto é que os dados são independentes entre si. Já no escopo da DdO em séries temporais os dados não são completamente independentes, pois a última observação influencia diretamente na observação seguinte. Essa dependência resulta em um comportamento brando e gradual e quando mudanças abruptas ocorrem teremos alta probabilidade de se tratar de um *outlier*. Outra consequência da dependência é que o tipo de *outlier* em séries temporais serão contextuais (mudanças abruptas) ou coletivas (AGGARWAL, 2017a).

Os métodos de DdO em séries temporais que mais se destacam são os: **baseados em previsões**, utilizando-se de análise estatística para calcular pontos futuros prováveis baseado no histórico de dados; **baseados em formas incomuns**, característica mais aproveitada por métodos de aprendizado de máquina e agrupamento.

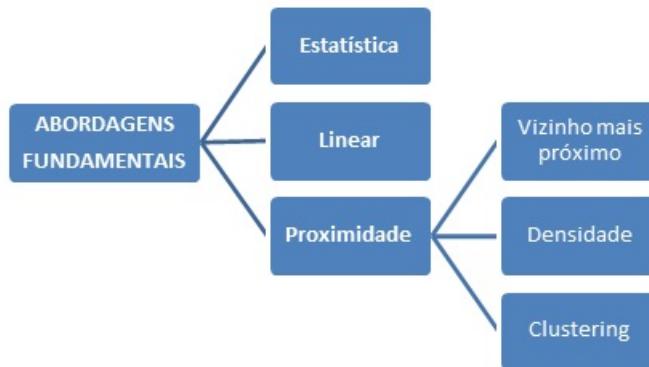
2.2.1 Abordagens de detecção de outliers

Diversos trabalhos na área de detecção de *outliers* fazem diferenciação quanto às abordagens de detecção (AGGARWAL, 2017a; BOUKERCHE; ZHENG; ALFANDI, 2020; CHANDOLA; BANERJEE; KUMAR, 2009), sendo as seguintes consideradas fundamentais (Figura 6): (1) **estatística**, (2) **linear** e (3) **proximidade**.

De maneira mais aprofundada, a abordagem por proximidade é dividida entre vizinhos mais próximos, densidade e *clustering*. Ressalta-se que a escolha do melhor modelo a ser utilizado dependerá de vários fatores, entre eles o contexto, tipo de dado, o tamanho do conjunto de dados, disponibilidade de exemplos de *outliers* e a interpretabilidade do modelo (AGGARWAL, 2017a).

Além de modelos fundamentais, Boukerche, Zheng e Alfandi (2020) citam em seu trabalho outras abordagens classificadas como avançadas, que utilizam os modelos fundamentais como ponto de partida e buscam resolver problemas mais atuais de alta dimensionalidade, fluxo de dados e *big data*.

Figura 6 – Abordagens fundamentais de detecção de *outliers*



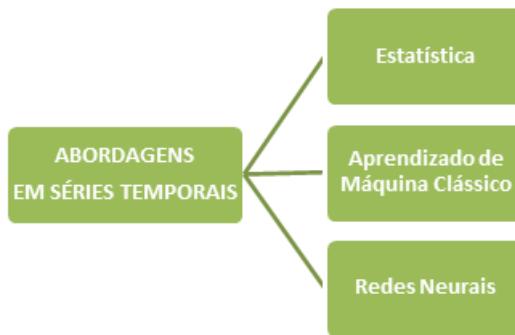
Fonte: Elaborada pelo autor.

2.2.2 Abordagens de detecção de *outliers* em séries temporais

O desafio de todo analista é entender bem o conjunto de dados que se está trabalhando e as vantagens e desvantagens de cada modelo. O uso de modelos mais complexos pode acarretar *overfitting* (sobreajuste), termo utilizado quando o modelo obtido se ajusta muito bem ao conjunto de dados, podendo classificar *outliers* como dados normais. Por outro lado, a utilização de modelos muito simples pode resultar no efeito contrário, a identificação de dados normais como *outliers*.

Da mesma forma, abordagens que funcionam muito bem para detecção de *outliers* em dados espaciais falham ao lidar com séries temporais. [Braei e Wagner \(2020\)](#), em seu trabalho com detecção de anomalias em séries temporais, dividiu as abordagens em três principais categorias:

Figura 7 – Categorias de detecção de *outliers* em séries temporais



Fonte: Elaborada pelo autor.

Também denominada como abordagem probabilística, na **abordagem estatística** a modelagem dos dados é realizada assumindo que o *dataset* possui uma distribuição bem definida da área estatística. A partir disso, os parâmetros da distribuição são assimilados pelo algoritmo e as

instâncias que apresentam discrepâncias significativas para o modelo encontrado são consideradas como *outliers*. Destaca-se que o termo “*outlier*” foi introduzido e primeiramente estudado na área da estatística (GRUBBS, 1974).

A vantagem dessa abordagem está no fato de que, uma vez encontrando o modelo estatístico adequado para o conjunto de dados, ela pode ser aplicada para praticamente qualquer tipo de dado. Não obstante, a grande desvantagem está justamente na dificuldade de encontrar esse modelo estatístico apropriado, o que pode resultar em erros na classificação de anomalias. Além disso, a própria existência do *outlier* pode ser um complicador na tarefa de busca pela melhor distribuição.

Por sua vez, a **abordagem de aprendizado de máquina** (*machine learning*) tenta encontrar anomalias sem assumir um modelo estatístico gerativo específico, considerando que o processo de geração dos dados é uma caixa preta, buscando o aprendizado de máquina analisando unicamente os dados disponibilizados aos modelos (BRAEI; WAGNER, 2020).

Por último, a **abordagem de redes neurais** é a mais recente e, similar às abordagens de aprendizado de máquina, não analisa o processo gerativo dos dados, sendo uma corrente popular devido aos resultados empíricos.

Para este trabalho foram selecionados modelos que tenham prestígio e vasta pesquisa na área acadêmica em cada uma das três principais abordagens, mostrados na Tabela 1. Além de modelos já consagrados, será testado dentro da abordagem estatística um algoritmo lançado em 2017 pelo time do Facebook Research (TAYLOR; LETHAM, 2017) denominado Prophet, ferramenta de código livre que está ganhando notoriedade dentro da área de predição.

Tabela 1 – Abordagens e modelos utilizados nesse trabalho

Abordagem	Modelo	Referencial Acadêmico
Estatística	SARIMA	(SAMAL <i>et al.</i> , 2019) (SOETRISNO <i>et al.</i> , 2019) (SOMBOONSAK, 2019)
	TES/Holt-Winters	(MCKENZIE, 1985) (KALEKAR, 2004) (SAMAL <i>et al.</i> , 2019)
	Prophet	(TAYLOR; LETHAM, 2017)
Aprendizado de Máquina	DBSCAN	(ESTER <i>et al.</i> , 1996)
	LOF	(BREUNIG <i>et al.</i> , 2000)
	Isolation Forest	(Liu; Ting; Zhou, 2008)
Redes Neurais	MLP	(HASELSTEINER; PFURTSCHELLER, 2001) (ROSA, 2010) (YAO <i>et al.</i> , 2006)
	LSTM	(KIM; CHO, 2018) (MALHOTRA <i>et al.</i> , 2016)

Fonte: Elaborada pelo autor.

2.3 Trabalhos relacionados e contribuição desse trabalho

O desafio de detectar *outliers* é estudado desde o início da década de 70 (ANDERBERG, 1973; FOX, 1972; GRUBBS, 1974), na maioria dos casos com foco na perspectiva estatística.

Desses trabalhos pioneiros surgiram livros clássicos que serviram de base para toda área de detecção de *outliers* (BARNETT; LEWIS, 1978; HAWKINS, 1980; ROUSSEEUW; LEROY, 1987). Nesse ínterim, o poder computacional evoluiu de forma exponencial, dando surgimento a novos desafios e criando diferentes abordagens e possibilidades inovadoras. As obras de Chandola, Banerjee e Kumar (2009) e de Aggarwal (2017b) buscaram endereçar essas evoluções e se tornaram referências no assunto.

Várias pesquisas buscaram revisitar o tema de detecção de *outliers* (BOUKERCHE; ZHENG; ALFANDI, 2020; AKOGLU; TONG; KOUTRA, 2015; GUPTA *et al.*, 2014; SINGH; AGGARWAL, 2013; ZHANG, 2013). Outros trabalhos exploraram sua aplicação em áreas específicas: (a) **diagnóstico médico** (HAUSKRECHT *et al.*, 2012; GAO; WU, 2020); (b) **detecção de intrusão** (JABEZ; MUTHUKUMAR, 2015; MINGQIANG; HUI; QIAN, 2012); (c) **dados de alta dimensionalidade** (RO *et al.*, 2015; LIU *et al.*, 2017; ZIMEK; SCHUBERT; KRIEGEL, 2012); (d) tráfego/transporte (LI *et al.*, 2009; CHEN; WANG; ZUYLEN, 2010); (e) **fraudes em cartão de crédito** (MALINI; PUSHPA, 2017; PAWAR; KALAVADEKAR; TAMBE, 2014; CHAUDHARY; YADAV; MALLICK, 2012); (f) **fraudes em sistema de saúde** (CAPELLEVEEN *et al.*, 2016; BAUDER; KHOSHGOFTAAR, 2016).

Na perspectiva da área mais específica de detecção de anomalias em séries temporais, inicialmente os estudos eram focados na identificação e remoção de ruídos do conjunto de dados, com o objetivo de obter dados mais comportados, o que contribui para regressões e previsões mais precisas (CHANG; TIAO; CHEN, 1988; CHEN; LIU, 1993; FOX, 1972). Mais recentemente, vários trabalhos se concentraram no uso de métodos de aprendizado de máquina para a detecção de *outliers* em séries temporais (ALMAGUER-ANGELES *et al.*, 2019; LAZAREVIC *et al.*, 2003; MUNIR *et al.*, 2018). Outros, focaram em redes neurais com esse mesmo objetivo (BONTEMPS *et al.*, 2017; MALHOTRA *et al.*, 2016).

Alguns artigos concentram-se no tema de detecção de anomalias em contratos (CAMPOS, 2018; SHAN; MURRAY; SUTINEN, 2009), outros na detecção de *outliers* em serviços de computação em nuvem (HUANG *et al.*, 2017; NAVAZ; SANDEETHA; PRABHADEVI, 2013; PANDEESWARI; KUMAR, 2015), mas nenhum trabalho pesquisado por este autor possui foco em fiscalização de contratos administrativos de serviços de computação em nuvem.

A Tabela 2 busca relacionar os principais trabalhos no tema detecção de *outliers* mencionados nesse capítulo e quais abordam os seguintes pontos-chave:

1. Utiliza a abordagem estatística?
2. Utiliza a abordagem de aprendizado de máquina?
3. Utiliza a abordagem de redes neurais?
4. Possui foco em séries temporais?

5. Aborda a fiscalização de contratos administrativos?
6. Os dados são de serviço de computação em nuvem?

Tabela 2 – Principais trabalhos de DdO e atendimento de pontos-chave

TRABALHOS	PONTOS-CHAVE					
	1	2	3	4	5	6
Grubbs (1974)	✓					
Barnett e Lewis (1978)	✓					
Chandola, Banerjee e Kumar (2009)	✓	✓				
Aggarwal (2017b)	✓	✓	✓			
Lazarevic <i>et al.</i> (2003)	✓	✓	✓	✓		
Munir <i>et al.</i> (2018)	✓	✓	✓	✓		
Malhotra <i>et al.</i> (2016)			✓	✓		
Shan, Murray e Sutinen (2009)		✓			✓	
Campos (2018)	✓				✓	
Huang <i>et al.</i> (2017)		✓		✓		✓
Pandeeswari e Kumar (2015)		✓	✓	✓		✓
ESTE TRABALHO	✓	✓	✓	✓	✓	✓

A contribuição buscada por esse trabalho é abordar todos os 6 pontos-chave da Tabela 2 em uma única obra e fornecer ferramenta para auxiliar o fiscal do contrato a encontrar pontos de interesse para aumentar a eficácia do supervisionamento, além de automatizar a tarefa de detecção de pontos de interesse em séries temporais de serviço de computação em nuvem. O Capítulo 3 detalha como isso foi alcançado, comparando 9 modelos de detecção de *outliers* com 3 diferentes abordagens em 9 séries temporais, apontando aquele que apresenta melhor desempenho para verificação de faturas de contrato de *cloud computing*.

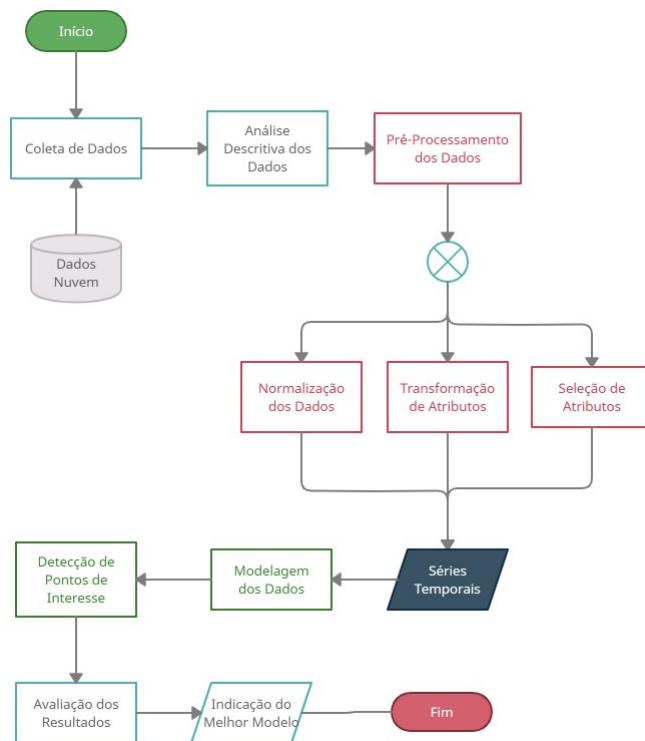
CAPÍTULO
3

MODELO PROPOSTO

3.1 Atividades Realizadas

Após a revisão bibliográfica do tema de detecção de *outliers* em séries temporais e a seleção dos métodos a serem utilizados nesse trabalho, o projeto foi segmentado em etapas sequenciais, conforme [Figura 8](#).

Figura 8 – Etapas desenvolvidas no trabalho



Fonte: Elaborada pelo autor.

Cada etapa será detalhada nas próximas sessões.

3.1.1 Coleta da base de dados

A empresa prestadora do serviço de computação em nuvem contratada pela administração pública disponibiliza em sítio eletrônico ferramenta para a gestão do contrato. Nesse portal, de acesso restrito aos gestores e fiscais do contrato, é possível extrair os dados de utilização diária de cada item consumido. No caso específico do contrato escopo desse trabalho (Pregão nº 29/2018 do Ministério do Planejamento, Desenvolvimento e Gestão), são oferecidos 32 itens que são cobrados sob demanda ([Figura 9](#)), podendo ser adquiridos de forma escalável. Dessa forma, quanto maior o uso dos serviços em nuvem, maior a quantidade de dados que deve ser monitorada e registrada para posterior avaliação do fiscal.

Figura 9 – Detalhamento dos itens do Pregão nº 29/2018

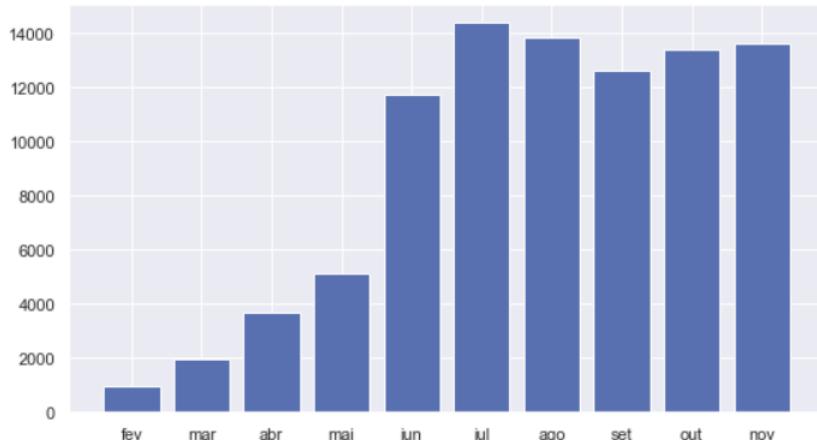
Item	Descrição do serviço (por reserva de recurso)	Unidade	Valor de referência (USN)	
1.	Máquina virtual padrão - adquirida por meio de vCPU, reservada por 1 ano	Unidade de vCPU/hora	0,0300	
2.	Máquina virtual padrão - adquirida por meio de memória, reservada por 1 ano	Gigabyte de memória/hora	0,0079	
3.	Máquina virtual Windows - adquirida por meio de vCPU, reservada por 1 ano	Unidade de vCPU/hora	0,0702	
4.	Máquina virtual Windows - adquirida por meio de memória, reservada por 1 ano	Gigabyte de memória/hora	0,0185	
5.	Máquina virtual com serviço de hospedagem de container gerenciado - adquirida por meio de vCPU, reservada por 1 ano	Unidade de vCPU/hora	0,0300	
Item	Descrição do serviço (por demanda)	Unidade	Valor de referência (USN)	
6.	Máquina virtual padrão - adquirida por meio de vCPU (por demanda)	Unidade de vCPU/hora	0,0507	
7.	Máquina virtual padrão - adquirida por meio de memória (por demanda)	Gigabyte de memória/hora	0,0135	
8.	Máquina virtual Windows - adquirida por meio de vCPU (por demanda)	Unidade de vCPU/hora	0,0927	
9.	Máquina virtual Windows - adquirida por meio de memória (por demanda)	Gigabyte de memória/hora	0,0245	
10.	Serviço de armazenamento de blocos (SSD)	Gigabyte/mês	0,2067	
11.	Serviço de armazenamento de blocos (HDD)	Gigabyte/mês	0,0437	
12.	Serviço de armazenamento de objetos	Gigabyte/mês	0,0227	
13.	Tráfego de saída da rede	Gigabyte/mês	0,0808	
14.	Tráfego de rede do平衡ador de carga	Gigabyte/mês	0,0070	
15.	Tráfego de rede do CDN	Gigabyte/mês	0,1175	
16.	Serviço de balanceamento de carga (*)	Unidade/hora	0,0250	
17.	Serviço de balanceamento de carga utilizando gerenciador de tráfego (*)	DNS Queries Milhão/Mês	0,4700	
18.	Porta de conexão de fibra 10Gbps	Unidade/hora	3,8518	
19.	Serviço de DNS – Hospedagem de zonas	Zona/mês	0,1000	
20.	Serviço de DNS – Consultas	Milhão de consulta/mês	0,4000	
21.	Serviço de VPN	Gigabyte/Mês	0,0100	
22.	VPN Gateway	Hora de Conexão	0,0467	
23.	Serviço de BI	Node/mês	253,3033	
24.	Serviço de Cofre de Senhas	Por operação (a cada 10.000)	0,6567	
25.	Serviço Web Application Firewall adquirido por ACL (**)	ACL/hora	0,0085	
26.	Serviço Web Application Firewall adquirido por hora (**)	Gateway/hora	0,0250	
27.	Serviço de Backup	Instância/mês	10,0000	
28.	Serviço de armazenamento de Backup	Gigabyte/mês	0,0114	
29.	Serviço de Autenticação (Integração com AD) adquirido por usuário (***)	Por usuário/Mês	3,1650	
30.	Serviço de Autenticação (Integração com AD) adquirido por mês (***)	Gigabyte/Mês	0,0663	
31.	Serviço de Auditoria e Análise de Logs	Gigabyte/Mês	0,5000	
32.	IP Público	Unidade/Mês	0,0017	

Fonte: [ComprasNet \(2018\)](#).

Para a construção da base de dados desse trabalho, um arquivo em formato *Comma-separated values* (CSV) foi extraído do portal de gerenciamento, contendo dados registrados desde o início do uso pela Controladoria-Geral da União (CGU), em fevereiro de 2020, até final de novembro de 2020, resultando em mais de 91 mil linhas de dados, conforme detalhado na [Figura 10](#).

Observa-se o volume crescente da quantidade de dados registrados. É importante ressaltar que, por ser um modelo de serviço novo no Governo Federal, estima-se que mais da metade dos serviços ainda poderão ser migrados para computação em nuvem. Fazendo-se uma projeção com os dados disponíveis até o novembro de 2020 para o contrato da CGU, mais de 23 mil linhas de

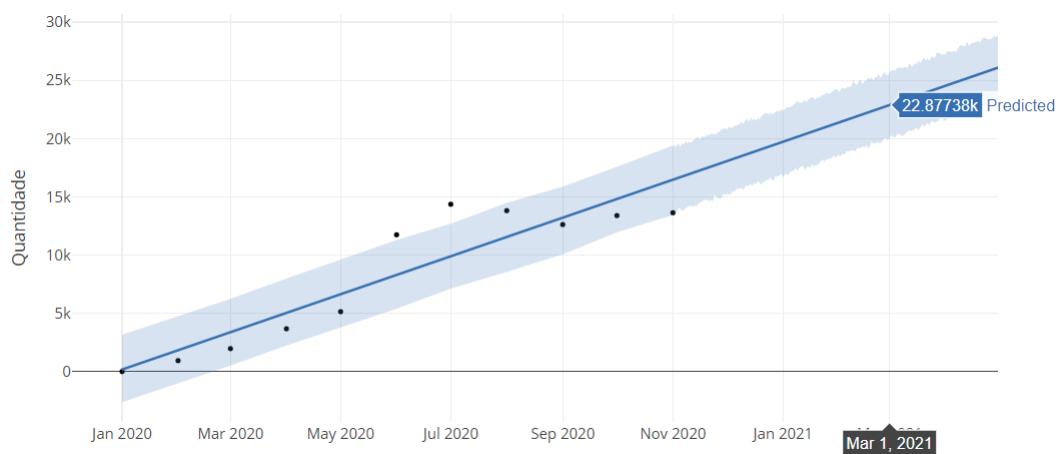
Figura 10 – Visualização do crescimento dos dados gerados



Fonte: Elaborada pelo autor.

detalhamento seriam obtidas somente no faturamento do mês de março de 2021 (Figura 11). A expectativa da área de TIC é que com o amadurecimento do serviço de nuvem no Brasil e com a comprovação prática da eficiência e redução de custos em relação a hospedagem *on-premises*, mais serviços serão migrados e o volume de dados cresça de maneira considerável, o que poderá gerar quantidade ainda maior de dados a ser validada mensalmente pelos fiscais.

Figura 11 – Previsão do total de linhas geradas em Março de 2021



Fonte: Elaborada pelo autor.

Cada linha dessa base de dados contém os seguintes atributos/colunas (Figura 12):

- **Conta** – Foram criadas três contas diferentes no provedor de serviços de nuvem, delimitando o escopo de cada ambiente: conta **principal**, onde estão hospedados os serviços de produção; conta **sandbox**, utilizada para homologação e testes; conta de **segurança**, utilizada para serviços de proteção e administração dos ambientes;

- **Início** – data de início do uso do recurso em ambiente de nuvem da contratada;
- **Fim** – data fim de uso do recurso em ambiente de nuvem da contratada;
- **Produto** – descrição do produto consumido;
- **Serviço** – subtipo do produto;
- **Quantidade** – valor numérico do consumo do recurso;
- **Unidade** – descrição da unidade de medida de acordo com o tipo de serviço. Por exemplo: item de serviço de armazenamento é cobrado por gigabyte/mês, já item de serviço de serviço de balanceamento de carga é cobrado por unidade/hora.

Figura 12 – Visualização dos dados no Jupyter Notebook

	inicio	fim	produto	servico	qnt	unidade	conta
0	2020-08-01	2020-08-02	AWS CloudTrail	APN1-FreeEventsRecorded	4.430000e+12	Events	principal
1	2020-08-01	2020-08-02	AWS CloudTrail	APN1-InsightsEvents	1.300000e+11	Events	principal
2	2020-08-01	2020-08-02	AWS CloudTrail	APN2-FreeEventsRecorded	4.440000e+12	Events	principal
3	2020-08-01	2020-08-02	AWS CloudTrail	APN2-InsightsEvents	1.300000e+11	Events	principal
4	2020-08-01	2020-08-02	AWS CloudTrail	APS1-FreeEventsRecorded	4.420000e+12	Events	principal
...
91269	2020-11-30	2020-12-01	Amazon Simple Storage Service	USE1-EUW3-AWS-Out-Bytes	5.178000e-07	GB	segurança
91270	2020-11-30	2020-12-01	Amazon Simple Storage Service	USE1-SAE1-AWS-Out-Bytes	1.553400e-06	GB	segurança
91271	2020-11-30	2020-12-01	Amazon Simple Storage Service	USE1-USE2-AWS-Out-Bytes	5.178000e-07	GB	segurança
91272	2020-11-30	2020-12-01	Amazon Simple Storage Service	USE1-USW1-AWS-Out-Bytes	5.178000e-07	GB	segurança
91273	2020-11-30	2020-12-01	Amazon Simple Storage Service	USE1-USW2-AWS-Out-Bytes	5.178000e-07	GB	segurança

91274 rows × 7 columns

Fonte: Elaborada pelo autor.

3.1.2 Análise Descritiva dos Dados

Após a coleta dos dados, utilizou-se a linguagem de programação Python para visualização e manipulação do dataset. Os códigos, bibliotecas e dados utilizados podem ser consultados no repositório do autor deste trabalho ([LINO, 2021](#)).

A primeira verificação realizada foi em relação a existência de dados nulos na base de dados. Não foram encontrados valores nulos e foi constatado que o tipo dos dados de cada coluna está adequado para a metodologia proposta, conforme detalhado na [Figura 13](#).

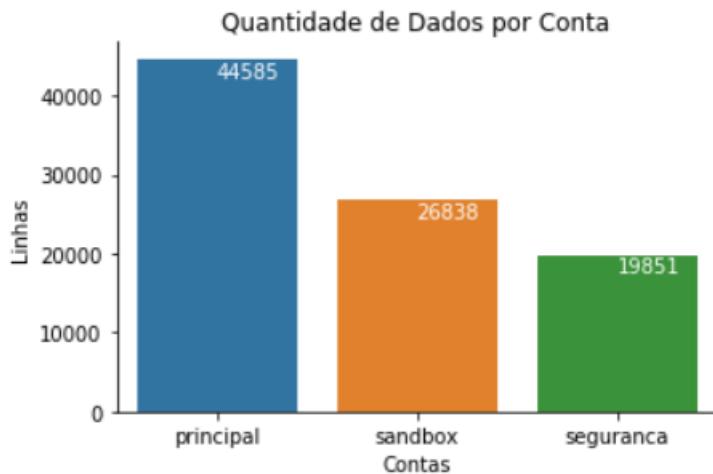
Como detalhado na sessão anterior, dados de diferentes serviços, unidades de medida e contas estão unificados em uma base de dados única. Sendo assim, foi necessário realizar algumas manipulações nos dados originais objetivando a obtenção de 9 séries temporais diferentes que serão analisadas pelos modelos de detecção de *outliers* propostos. Para esse fim, foram escolhidos 3 serviços em três produtos diferentes. O primeiro atributo analisado foi a coluna ‘conta’, através da visualização da quantidade de dados em cada conta ([Figura 14](#)).

Figura 13 – Verificação do tipo de cada coluna e valores nulos

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51633 entries, 0 to 51632
Data columns (total 7 columns):
inicio      51633 non-null datetime64[ns]
fim         51633 non-null datetime64[ns]
produto     51633 non-null object
servico     51633 non-null object
qnt         51633 non-null float64
unidade    51633 non-null object
conta       51633 non-null object
dtypes: datetime64[ns](2), float64(1), object(4)
memory usage: 2.8+ MB
```

Fonte: Elaborada pelo autor.

Figura 14 – Quantidade de linhas em cada conta



Fonte: Elaborada pelo autor.

Como a conta ‘principal’ possui a maior quantidade de dados do *dataset* e registra o consumo dos principais produtos de computação em nuvem utilizados, os dados das contas ‘sandbox’ e ‘segurança’ foram retirados da base de dados.

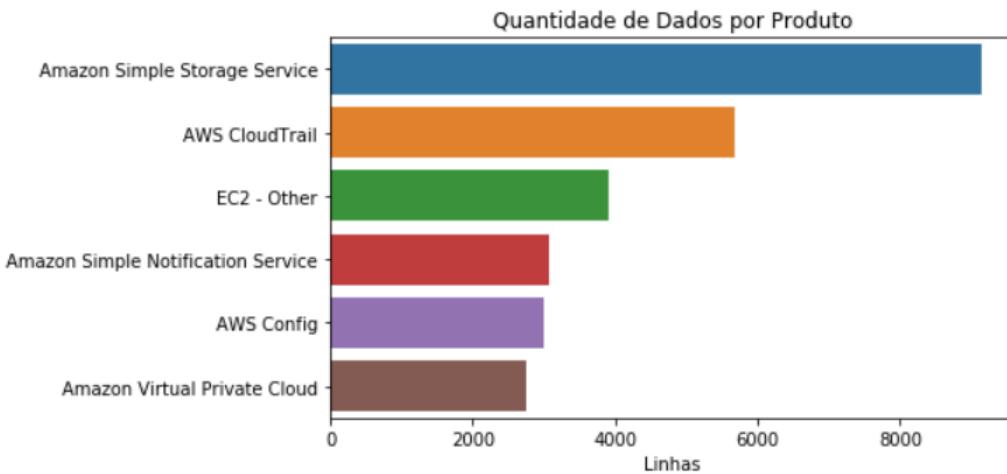
As colunas de ‘inicio’ e ‘fim’ informam o período em que cada serviço foi utilizado. Ao se analisar essas informações, constatou-se que os dados de consumo de cada serviço são calculados e contabilizados ao final de cada dia, ou seja, mesmo que o recurso seja utilizado por 3 dias seguidos, serão registradas no relatório três entradas com o total consumido no final de cada dia. Sendo assim, a coluna ‘fim’ sempre será igual a data de ‘inicio’ adicionada de 1 dia, não agregando informação a base de dados e por isso foi eliminada.

Como a unidade de medida da coluna ‘unidade’ sempre será a mesma para cada serviço selecionado, também iremos eliminar esse atributo do *dataset*.

Com a base de dados resultante dos procedimentos anteriores, foi necessário filtrar os atributos ‘produto’ e ‘servico’ para se obter as séries temporais que serão as entradas para cada

modelo escopo desse trabalho. Foram utilizados no total 27 serviços diferentes nos 08 primeiros meses de contrato, sendo necessário filtrar 3 serviços em 3 produtos diferentes a fim de obter as 9 séries temporais. O primeiro fator considerado para a escolha dos produtos foi o número de ocorrências na base de dados, buscando privilegiar os mais utilizados [Figura 15](#).

Figura 15 – Os 5 produtos mais utilizados



Fonte: Elaborada pelo autor.

Além da avaliação da frequência na base de dados, foi considerada a opinião do especialista técnico e fiscal do contrato, que indicaram os três produtos mais importantes entre os mais utilizados. Para cada produto também foram selecionados os serviços mais interessantes e variados para a análise desse trabalho ([Tabela 3](#)), considerando aspectos como distribuição dos dados, comportamento da série temporal e quantidade de *outliers*.

Tabela 3 – Descrição dos produtos e os 3 serviços escolhidos

PRODUTO	DESCRIÇÃO PRODUTO	SERVIÇO SELECIONADO
Amazon Simple Storage Service	Serviço de armazenamento de dados	Requests-Tier1 Requests-Tier2 SAE1-USE1-AWS-Out-Bytes
EC2 – Other	Serviço de máquina virtual	SAE1-DataTransfer-Regional-Bytes SAE1-EU-AWS-In-Bytes SAE1-NatGateway-Bytes
Amazon Virtual Private Cloud	Serviço de VPN (Virtual Private Network)	SAE1-DataTransfer-In-Bytes SAE1-USW2-AWS-In-Bytes SAE1-USE1-AWS-In-Bytes

Após todas as manipulações realizadas, para cada produto e serviço selecionado para a análise, foi obtida uma série temporal com as seguintes características: coluna ‘**inicio**’, com as datas em que houve consumo do serviço de computação em nuvem; coluna ‘**qnt**’, com a quantidade consumida do serviço analisado ([Figura 16](#)).

Figura 16 – Exemplo de série temporal obtida

	inicio	qnt
0	2020-02-01	7.820000e-08
1	2020-02-02	5.220000e-08
2	2020-02-03	1.267000e-07
3	2020-02-04	5.220000e-08
4	2020-02-05	5.220000e-08
...
208	2020-08-27	7.830000e-08
209	2020-08-28	7.830000e-08
210	2020-08-29	5.220000e-08
211	2020-08-30	5.220000e-08
212	2020-08-31	5.220000e-08

213 rows × 2 columns

Fonte: Elaborada pelo autor.

3.1.3 Pré-Processamento dos Dados

3.1.3.1 Transformação do campo de data em modelos espaciais

Alguns modelos testados nesse trabalho são usualmente utilizados com dados espaciais, não lidando muito bem com séries temporais. Sendo assim, para ser utilizada nos modelos de aprendizado de máquina escolhidos para esse trabalho (DBSCAN, LOF e Isolation Forest), a coluna ‘**inicio**’ da série temporal obtida precisou ser transformada em dados numéricos, buscando transformar os dados temporais em dados espaciais ([Figura 17](#)).

Figura 17 – Exemplo de série temporal transformada

	inicio	qnt
172	15962400000000000000	5.220000e-08
438	15963264000000000000	5.220000e-08
617	15964128000000000000	5.220000e-08
839	15964992000000000000	8.950000e-08
1069	15965856000000000000	5.220000e-08
...
42494	15958080000000000000	5.220000e-08
42724	15958944000000000000	8.950000e-08
42961	15959808000000000000	8.940000e-08
43148	15960672000000000000	1.044000e-07
43320	15961536000000000000	5.220000e-08

204 rows × 2 columns

Fonte: Elaborada pelo autor.

3.1.3.2 Normalização dos dados

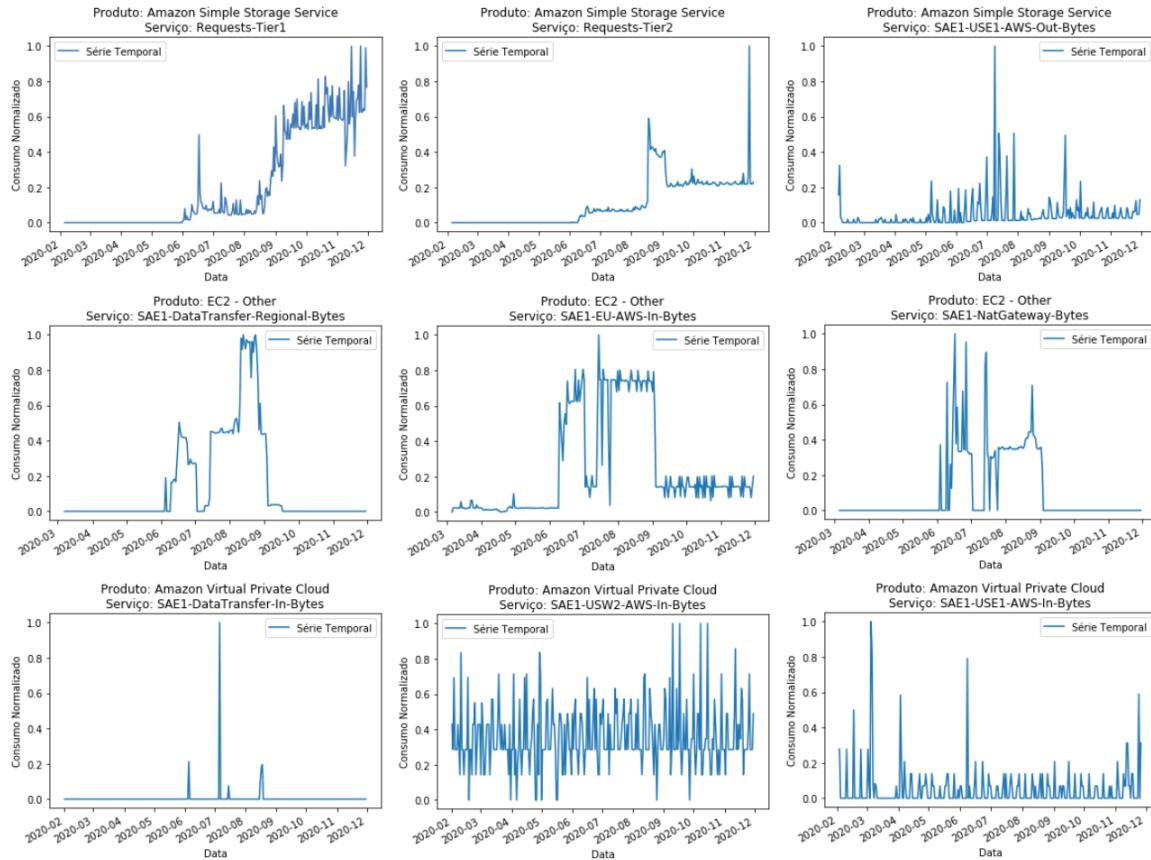
Um requisito comum para muitos métodos de aprendizado de máquina é a normalização dos dados, evitando que os algoritmos sejam excessivamente influenciados pelos dados com maior ordem de grandeza. Foi utilizada a **normalização Min-Max**, que transforma os valores da variável alvo em valores decimais entre 0 e 1, conforme a seguinte fórmula:

$$X_{\text{normalizado}} = \frac{(X - X_{\min})}{(X_{\max} - X_{\min})}$$

3.1.3.3 Séries Temporais Resultantes

Após todas as manipulações realizadas nos dados, separando as séries temporais e normalizando os dados, obtemos as 9 séries temporais a serem utilizadas neste trabalho (Figura 18).

Figura 18 – Séries temporais resultantes após etapa de Pré-Processamento



Fonte: Elaborada pelo autor.

3.1.4 Modelagem dos dados

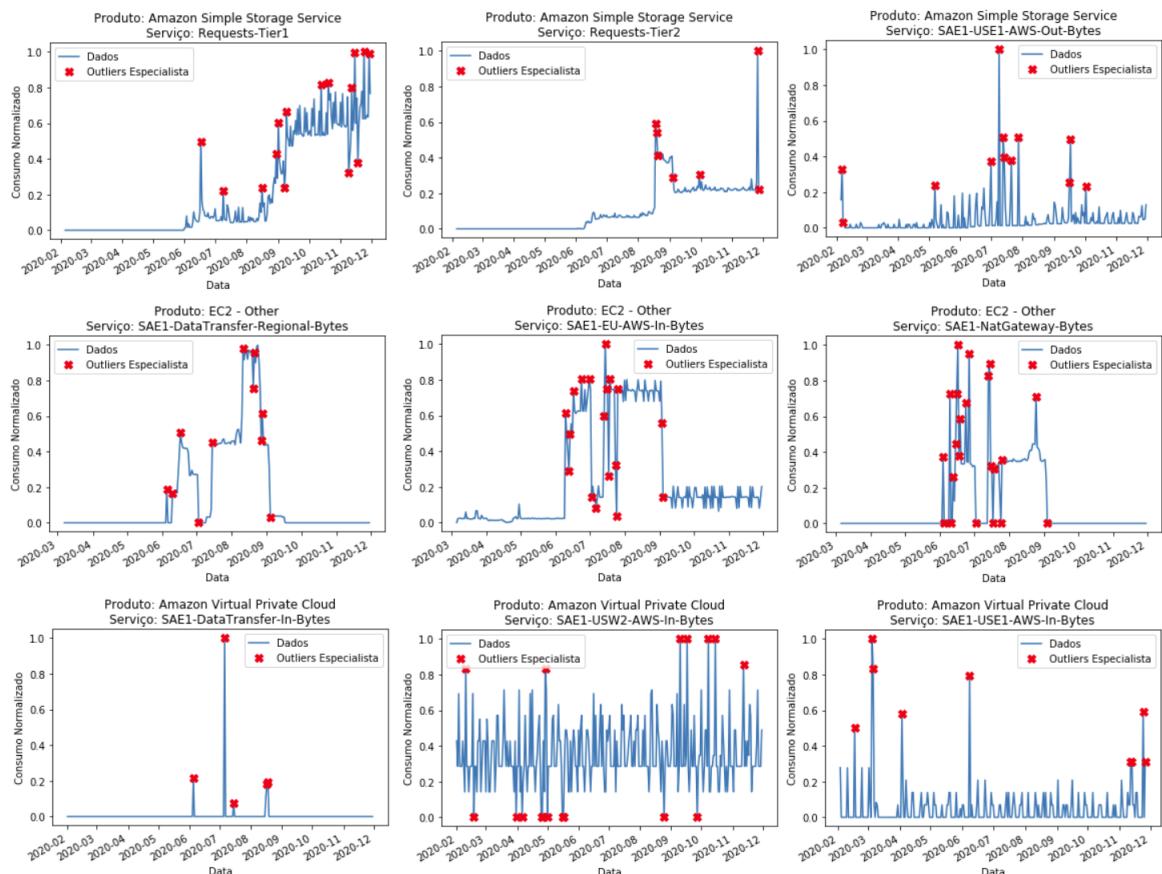
Após a obtenção das séries temporais para cada par de produto-serviço, os dados foram inseridos em cada modelo escopo deste trabalho (Tabela 1). Optou-se por não utilizar o processo

de separação dos dados em treino e teste, utilizando todos os dados de cada série temporal na modelagem.

O processo utilizado para a detecção de pontos de interesse nos modelos das abordagens estatísticas e de redes neurais foi a utilização de todos os dados de cada série temporal como entrada para a predição de uma nova série temporal, que posteriormente foi comparada com a série temporal original. Os pontos mais discrepantes (acima de um limite determinado pelo especialista) foram identificados como *outliers*. Já para os modelos de aprendizado de máquina, utilizando as séries temporais como entrada, os métodos utilizados nesse trabalho informam de forma mais direta se um determinado ponto é um *outlier* (DBSCAN) ou informam o *score* de anomalia de cada ponto (LOF e Isolation Forest). Nesse último caso, novamente é utilizado um *threshold* para classificar observações como anomalias.

Dessa forma, para cada modelo obtivemos uma lista de instâncias consideradas como pontos de interesse. Essas listas foram comparadas com a lista de **pontos de interesse identificadas de forma visual pelo especialista e fiscal do contrato** (Figura 19), informação esta que será considerada como a linha de base, ou seja, consideraremos que somente as instâncias desta lista são os verdadeiros pontos de interesse.

Figura 19 – Pontos de interesse apontados pelo especialista



Fonte: Elaborada pelo autor.

A [Tabela 4](#) mostra as principais características das séries temporais escolhidas para avaliação dos modelos.

Tabela 4 – Características das séries temporais

Produto	Serviço	Pontos Normais	PdI	% PdI	Sazonalidade
Amazon Simple Storage Service	Requests-Tier1	285	15	5,26	Semanal
	Requests-Tier2	293	7	5,12	Semanal
	SAE1-USE1-AWS-Out-Bytes	288	12	5,21	Semanal
EC2 – Other	SAE1-DataTransfer-Regional-Bytes	258	11	5,81	Semanal
	SAE1-EU-AWS-In-Bytes	252	18	5,95	Semanal
	SAE1-NatGateway-Bytes	248	22	6,05	Semanal
Amazon Virtual Private Cloud	SAE1-DataTransfer-In-Bytes	299	5	5,02	Semanal
	SAE1-USW2-AWS-In-Bytes	287	17	5,23	Semanal
	SAE1-USE1-AWS-In-Bytes	289	9	5,19	Semanal

Métricas de avaliação, detalhadas na [Subseção 3.1.5](#), serão utilizadas para obter de forma objetiva qual modelo consegue melhor identificar os pontos de interesse para a base de dados de serviço em computação em nuvem.

3.1.4.1 Parâmetros utilizados nos algoritmos

As próximas tabelas ([Tabela 5](#), [Tabela 6](#) e [Tabela 7](#)) detalham os parâmetros utilizados em cada um dos modelos utilizados nesse trabalho.

Tabela 5 – Parâmetros dos modelos da abordagem Estatística

Modelo	Parâmetro	Valor
Holt-Winters	trend	add
	seasonal	add
	seasonal_periods	7
SARIMA (auto_arima)	start_p	0
	max_p	10
	start_q	0
	max_q	10
	m	7
	seasonal	True
	stepwise	True
Prophet	interval_width	0.9
	changepoint_range	0.6
	weekly_seasonality	True
	seasonality_mode	additive
	growth	linear

Tabela 6 – Parâmetros dos modelos da abordagem de Aprendizado de Máquina

Modelo	Parâmetro	Valor
DBSCAN	eps	0.1
	min_samples	2
LOF	n_neighbors	20
	metric	euclidean
Isolation Forest	n_estimators	100
	contamination	auto

Tabela 7 – Parâmetros dos modelos da abordagem de Redes Neurais

Modelo	Parâmetro	Valor
MLP	Arquitetura	2 camadas ocultas
	neurônios	100, 50
	total de parâmetros	5301
	optimizer	Adam
	loss	MSE
	activation	ReLU
	batch_size	1
	epochs	100
	limite	0.2
	dropout	0.2
	n_input	1
	n_features	1
LSTM	Arquitetura	1 camada LSTM
	neuronios	100
	total de parâmetros	40901
	Optimizers	Adam
	loss	MSE
	activation	ReLU
	batch_size	1
	epochs	100
	limite	0.2

3.1.5 Avaliação dos Modelos

Apesar de originalmente o cenário deste estudo ser não-supervisionado, ou seja, não existir rótulos nos dados históricos para distinguir os dados normais dos *outliers*, decidiu-se que o especialista iria apontar os principais pontos de interesse para cada série temporal analisada neste trabalho, visando o uso de métricas que informam de modo objetivo através de valor numérico o desempenho de cada método.

As métricas escolhidas foram a **Medida-F** e a **Área sob a Curva ROC (AUC-ROC)**. Conforme constatado em alguns trabalhos (DAVIS; GOADRICH, 2006; Jeni; Cohn; De La Torre, 2013), a métrica AUC-ROC pode mascarar performances insatisfatórias em *datasets* desbalanceados, sugerindo que a Medida-F é mais informativa e enfatiza a performance da classe

minoritária (*outliers*). Sendo assim, serão reportadas as duas métricas, mas a **Medida-F** será utilizada prioritariamente para a classificação dos modelos.

3.1.5.1 Medida-F

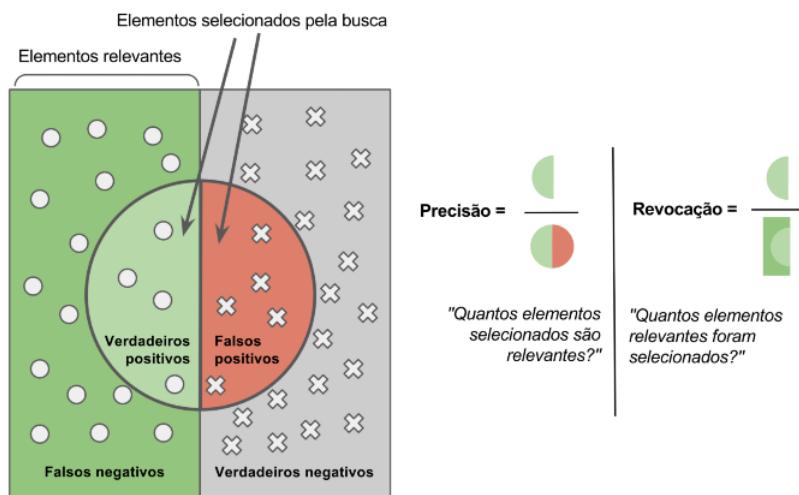
Métrica utilizada em vários trabalhos (MAYA; UENO; NISHIKAWA, 2019; MUNIR *et al.*, 2018), a Medida-F é um indicador de desempenho para modelos e algoritmos e é definida como a média harmônica da precisão e revocação, dada pela fórmula:

$$\text{Medida-F}_\beta = (1 + \beta^2) \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\beta^2 \cdot \text{Precisão} + \text{Revocação}}$$

- **Precisão** – métrica que informa, das classificações positivas obtidas pelo modelo, quantas foram corretas;
- **Revocação** – métrica que informa, das observações positivas existentes, quantas o modelo classificou acertadamente.

Na [Figura 20](#) é mostrado um exemplo visual da diferença entre precisão e revocação.

Figura 20 – Exemplo de Precisão e Revocação



Fonte: [Wikipédia \(2020\)](#).

Nesse trabalho será utilizado o valor de beta igual a 1, designando a métrica como Medida-F₁ (*F₁-score*):

$$\text{Medida-F}_1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}$$

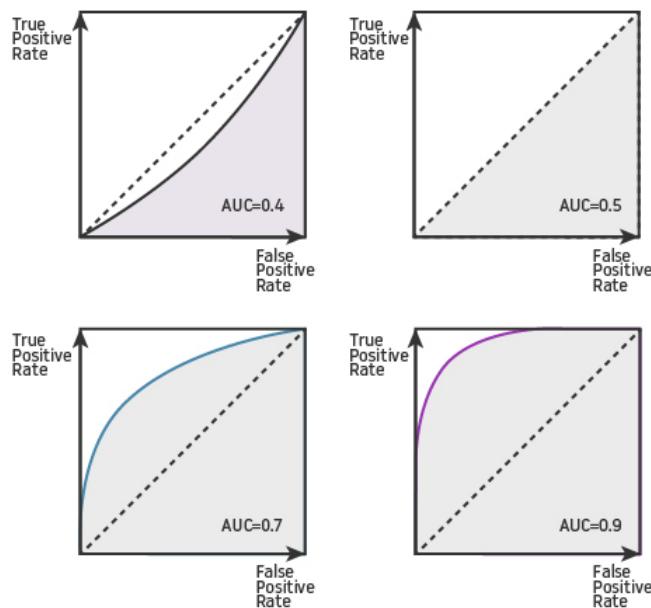
Os valores da Medida-F₁ variam entre 0 e 1, sendo o último valor quando temos o melhor cenário, em que a precisão e a revocação é perfeita.

3.1.5.2 Área sob a curva ROC

Outra métrica muito utilizada para a avaliação de modelos de detecção de anomalia ([MALHOTRA et al., 2016; PRATI; BATISTA; MONARD, 2008](#)) é a área sob a curva *Receiver operating characteristic* (ROC), que mostra a relação de custo (Taxa de falsos positivos (TFP)) e benefício (Taxa de verdadeiros positivos (TVP)). A curva ROC é o gráfico de desempenho de um modelo classificador em diferentes limiares de classificação. O valor da AUC nos informa com um valor numérico entre 0 e 1 o quanto competente é o modelo de classificação, sendo que quanto mais próximo do valor 1 melhor é o modelo.

Na [Figura 21](#) é mostrado um exemplo visual de diferentes resultados de AUC-ROC e o comportamento de suas curvas no gráfico.

Figura 21 – Exemplo de diferentes resultados de AUC-ROC



Fonte: [Lapix \(2020\)](#).

3.2 Resultados Obtidos

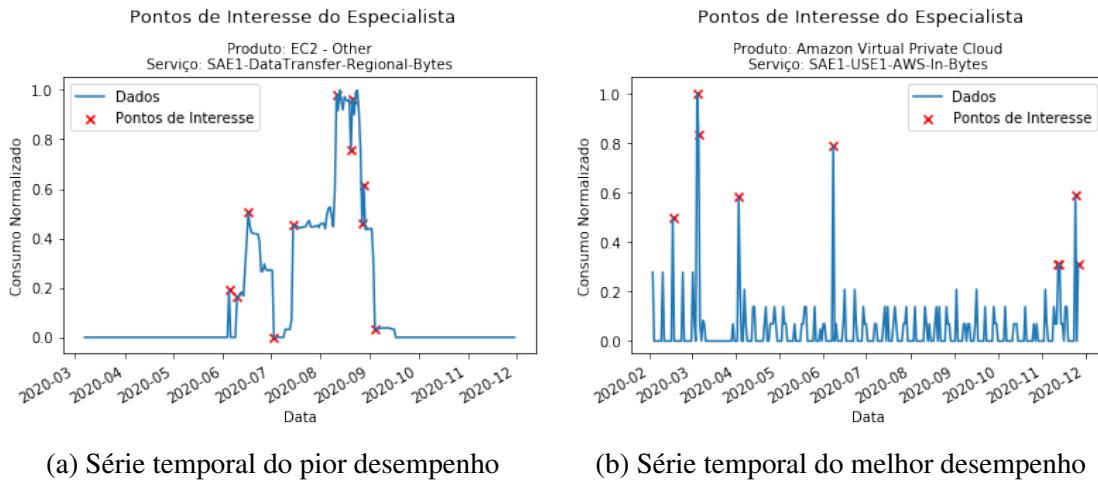
Nesta seção são apresentados os resultados da detecção de pontos de interesse nas séries temporais apresentadas na [Tabela 3](#), utilizando o método de Valores Extremos e os diferentes modelos descritos na [Tabela 1](#) com os parâmetros especificados na [Subsubseção 3.1.4.1](#). As visualizações dos PdI encontrados por cada modelo podem ser examinadas no [Apêndice A](#). As tabelas [Tabela 8](#) e [Tabela 9](#) detalham os resultados da **AUC-ROC** e **Medida- F_1** para cada par série-modelo e a [Tabela 10](#) apresenta a média dessas métricas, realizando a comparação final entre todos os algoritmos. Foram criadas escalas de cores nas duas primeiras tabelas para facilitar a visualização, sendo valores em vermelhos os piores e em azul os melhores resultados.

Tabela 8 – AUC-ROC para cada série temporal em cada modelo

Produto	Serviço	Valores Extremos	Prophet	Holt-Winters	SARIMA	DBSCAN	LOF	iForest	MLP	LSTM
Amazon Simple Storage Service	Requests-Tier1	0,500	0,832	0,765	0,765	0,695	0,732	0,795	0,865	0,761
	Requests-Tier2	0,714	0,762	0,714	0,786	0,786	0,786	0,711	0,643	0,643
	SAE1-USE1-AWS-Out-Bytes	0,925	0,957	0,663	0,873	0,832	0,950	0,995	0,917	0,750
EC2 - Other	SAE1-DataTransfer-Regional-Bytes	0,611	0,657	0,727	0,818	0,583	0,632	0,655	0,773	0,727
	SAE1-EU-AWS-In-Bytes	0,500	0,673	0,778	0,776	0,751	0,619	0,599	0,861	0,806
	SAE1-NatGateway-Bytes	0,636	0,721	0,839	0,860	0,755	0,690	0,715	0,932	0,795
Amazon Virtual Private Cloud	SAE1-DataTransfer-In-Bytes	0,995	0,898	0,600	0,700	0,998	0,998	0,995	0,700	0,700
	SAE1-USW2-AWS-In-Bytes	0,976	0,965	0,822	0,901	0,804	0,701	0,907	0,883	0,892
	SAE1-USE1-AWS-In-Bytes	0,981	0,998	0,882	0,995	0,819	0,990	0,984	0,993	0,887
Ranking do Modelo		8º	3º	9º	2º	6º	5º	4º	1º	7º

Observando a [Tabela 8](#), nota-se que os modelos encontraram mais dificuldades em encontrar os PdI na série do serviço **SAE1-DataTransfer-Regional-Bytes** ([Figura 22a](#)). Por outro lado, o serviço em que os modelos apresentaram melhor desempenho foi a **AE1-USE1-AWS-In-Bytes** ([Figura 22b](#)).

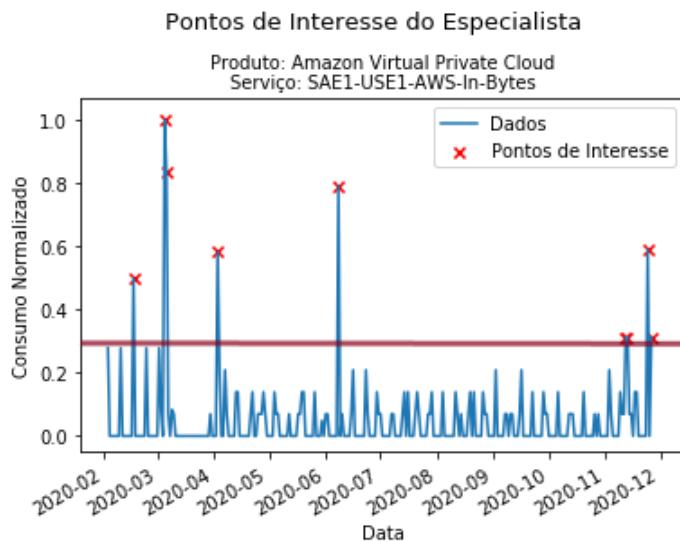
Figura 22 – Séries temporais com resultados opostos



Fonte: Elaborada pelo autor.

Comparando os gráficos dessas duas séries, pode-se atribuir essa diferença de performance à variação do posicionamento dos PdI, sendo que para séries em que é possível traçar um limiar claro entre os pontos normais e anomalias, como ilustrado na [Figura 23](#), o desempenho dos modelos é acentuado. Para a série dessa figura, por exemplo, o método de Valores Extremos atingiu valor de AUC-ROC de 0,981, quantia maior do que modelos como o Holt-Winters e DBSCAN.

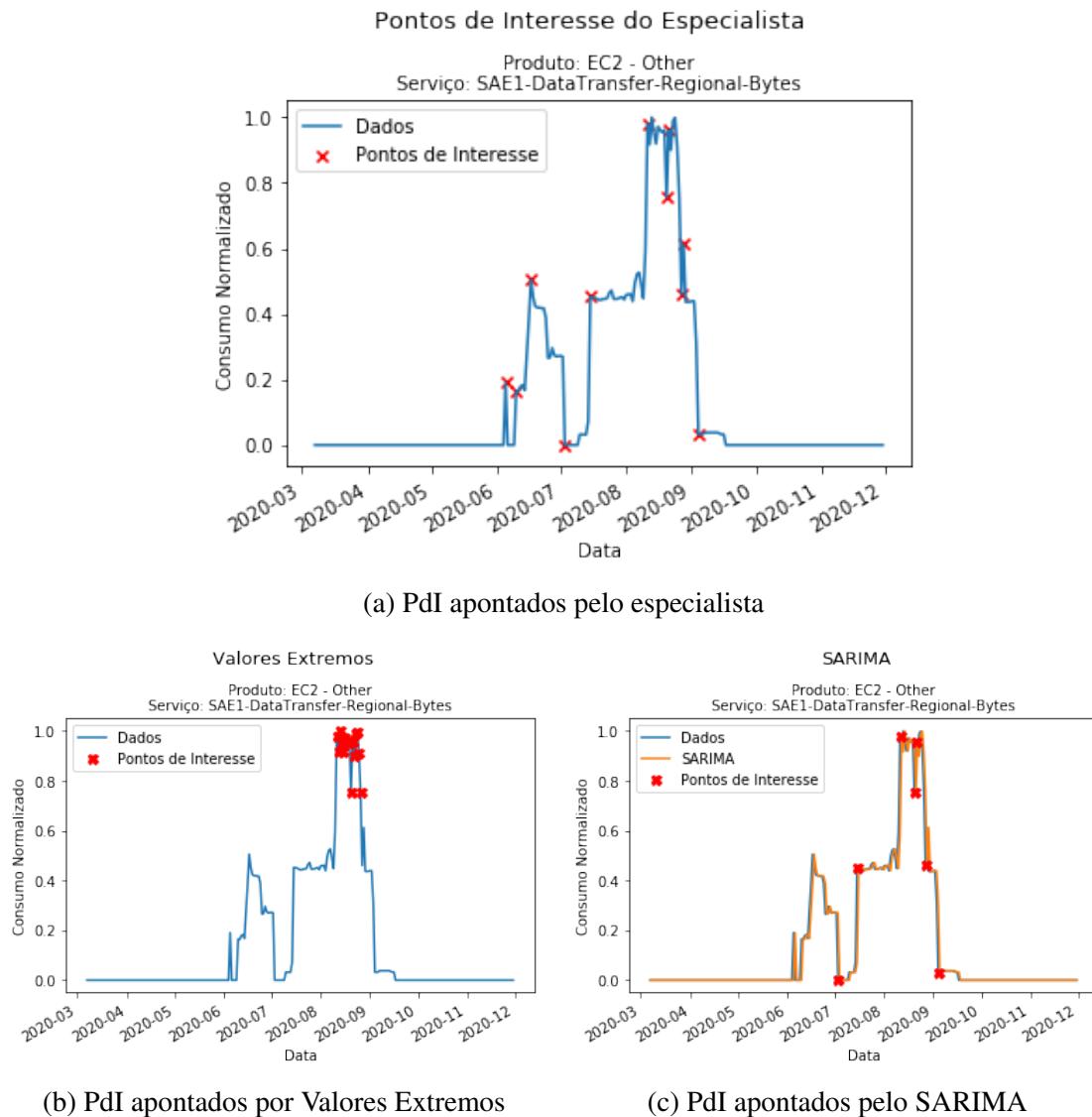
Figura 23 – Limiar de separação claro entre observações normais e PdI



Fonte: Elaborada pelo autor.

Em contrapartida, o modelo de Valores Extremos falha gravemente ao identificar os PdI da série da [Figura 24a](#), apontando erroneamente os pontos da [Figura 24b](#), falhando em detectar *outliers* quando há mudanças abruptas ou novidades, obtendo um resultado de AUC-ROC de somente 0,611. Observe que o modelo que apresentou melhor desempenho para essa série conseguiu detectar alguns desses pontos fora do conjunto dos valores extremos ([Figura 24c](#)), alcançando AUC-ROC de 0,818.

Figura 24 – Pontos de Interesse detectados pelos modelos Valores Extremos e SARIMA



Fonte: Elaborada pelo autor.

Tabela 9 – Medida- F_1 para cada série temporal em cada modelo

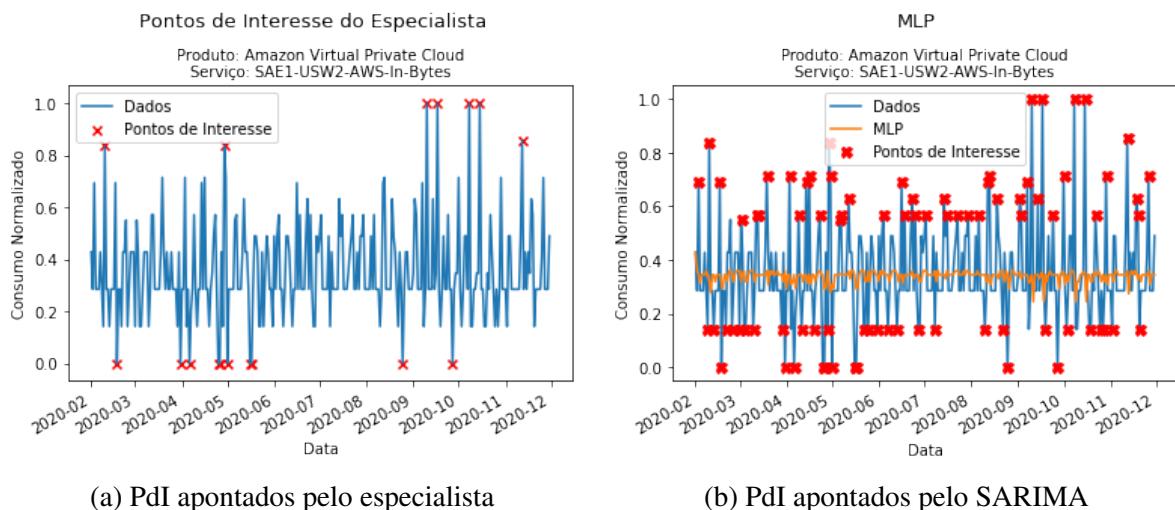
Produto	Serviço	Valores	Prophet	Holt-Winters	SARIMA	DBSCAN	LOF	iForest	MLP	LSTM
		Extremos								
Amazon Simple Storage Service	Requests-Tier1	0,000	0,478	0,667	0,667	0,500	0,609	0,667	0,815	0,615
	Requests-Tier2	0,600	0,320	0,600	0,727	0,727	0,727	0,500	0,444	0,444
	SAE1-USE1-AWS-Out-Bytes	0,524	0,917	0,444	0,818	0,762	0,786	0,889	0,909	0,667
EC2 - Other	SAE1-DataTransfer-Regional-Bytes	0,222	0,286	0,625	0,778	0,235	0,195	0,276	0,706	0,625
	SAE1-EU-AWS-In-Bytes	0,000	0,389	0,714	0,690	0,571	0,303	0,286	0,839	0,759
	SAE1-NatGateway-Bytes	0,429	0,571	0,789	0,800	0,632	0,514	0,588	0,927	0,743
	Amazon Virtual Private Cloud	0,769	0,800	0,333	0,571	0,909	0,909	0,769	0,571	0,571
SAE1-USW2-AWS-In-Bytes	0,708	0,630	0,511	0,438	0,421	0,519	0,824	0,337	0,500	
	SAE1-USE1-AWS-In-Bytes	0,621	0,947	0,700	0,857	0,522	0,750	0,667	0,818	0,824
Ranking do Modelo		9º	6º	2º	8º	7º	4º	1º	3º	

Conforme ressaltado na Subseção 3.1.5, os valores da métrica AUC-ROC foram superiores aos da Medida- F_1 , havendo inclusive mudança na determinação do melhor e pior desempenho para cada série temporal. Isso demonstra que realmente há um mascaramento de performance para dados desbalanceados usando AUC-ROC, por isso a Medida- F_1 será privilegiada. Vale destacar que, ao se considerar a métrica Medida- F_1 ao invés da AUC-ROC, houve mudança na determinação do melhor e/ou pior desempenho em 4 séries:

- **Request-Tier2** – pior desempenho para a métrica AUC-ROC são os modelos MLP e LSTM, já na Medida- F_1 é o modelo **Prophet**;
- **SAE1-DataTransfer-Regional-Bytes** – pior desempenho para a métrica AUC-ROC é o modelo DBSCAN, já na Medida- F_1 é o modelo **LOF**;
- **SAE1-USE1-AWS-Out-Bytes** – melhor desempenho para a métrica AUC-ROC é o modelo Isolation Forest, já na Medida- F_1 é o modelo **Prophet**;
- **SAE1-USW2-AWS-In-Bytes** – melhor e pior desempenho para a métrica AUC-ROC são respectivamente os modelos Valores Extremos e LOF, já na Medida- F_1 são respectivamente os modelos **Isolation Forest** e **MLP**.

Além dessas mudanças, é importante ressaltar a diferença de resultado do pior desempenho na série do serviço SAE1-USW2-AWS-In-Bytes (Figura 25a) ao se utilizar a métrica Medida- F_1 : o modelo MLP demonstrou maior sensibilidade na identificação dos PdI, classificando vários pontos intermediários como *outliers*, aumentando a quantidade de falsos positivos e, por consequência, degradando fortemente a componente **Precisão** da Medida- F_1 (definição abordada na Subsubseção 3.1.5.1).

Figura 25 – Pontos de Interesse detectados pelo modelo MLP



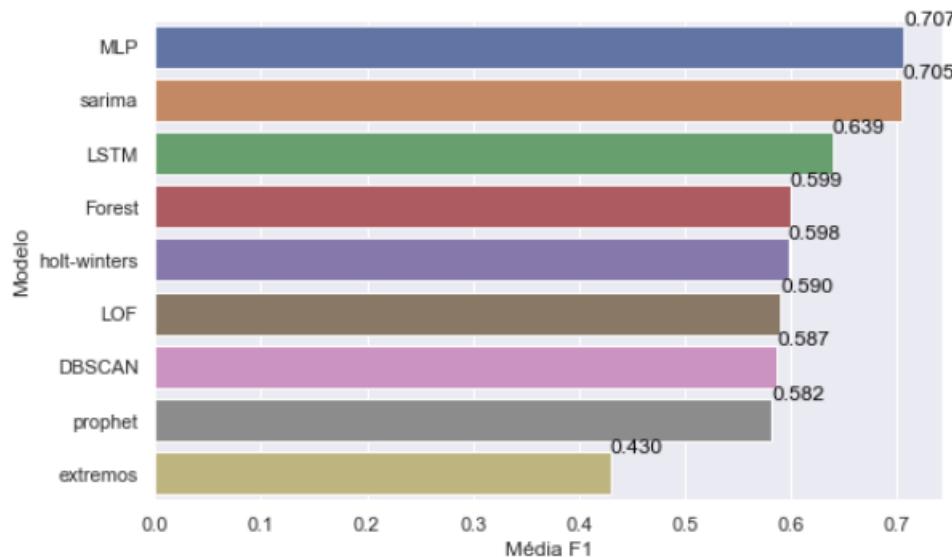
Fonte: Elaborada pelo autor.

Tabela 10 – Médias das Métricas Medida- F_1 e AUC-ROC

Modelo	Ranking	Medida- F_1	AUC-ROC
MLP	1º	0.707	0.841
SARIMA	2º	0.705	0.830
LSTM	3º	0.639	0.774
iForest	4º	0.601	0.816
Holt-Winters	5º	0.598	0.754
LOF	6º	0.590	0.788
Prophet	7º	0.590	0.822
DBSCAN	8º	0.587	0.780
Valores Extremos	9º	0.430	0.760

A Tabela 10 mostra as médias dos resultados de cada modelo para as duas métricas, classificando em ordem decrescente pela métrica Medida- F_1 .

A Figura 26 exibe as médias da Medida- F_1 em todas as séries para cada modelo em ordem decrescente.

Figura 26 – Média da Medida- F_1 para cada modelo

Fonte: Elaborada pelo autor.

Verifica-se que todos modelos testados se sobressaíram ao modelo de Valores Extremos, com destaque para o modelo **MLP**, que obteve os melhores índices considerando as duas métricas de avaliação. Em termos de abordagens, os melhores resultados foram obtidos na área das Redes Neurais, com dois representantes nas 3 primeiras posições.

Interessante observar que, para alguns casos, os modelos de Redes Neurais obtiveram desempenho inferior a outros modelos, inclusive ao método de Valores Extremos. Esse comportamento foi observado quando a série analisada contém pontos de interesse pontuais e bem destacados. Por outro lado, quando a série retrata comportamento mais variado e contém pontos

de interesse mais contextuais, a abordagem de Redes Neurais se mostrou mais competente.



CONCLUSÃO

4.1 Conclusão

O presente trabalho teve como objetivo apontar um modelo de aprendizado de máquina automatizado e mais eficaz que a técnica de Valores Extremos para a identificação de pontos de interesse em faturamento de contrato de computação em nuvem. Três abordagens de detecção de *outliers* foram utilizadas, totalizando 9 modelos treinados nesse trabalho. Foi demonstrado para o estudo de caso que todos os modelos propostos obtiveram melhor performance do que o método de Valores Extremos. Conforme apresentado ao final da [Seção 3.2](#), o modelo MLP apresentou o melhor desempenho, com um ganho médio de eficácia de mais de 64% em relação ao método de Valores Extremos ao se utilizar a métrica Medida- F_1 .

Constatou-se também que a abordagem de Redes Neurais apresenta maior adaptabilidade aos diferentes tipos de séries temporais analisadas, característica especialmente importante quando não é conhecido de antemão o comportamento dos dados e os tipos de pontos de interesse que serão analisados.

Embora o foco tenha sido detecção de pontos de interesse na área específica de dados de nuvem, a metodologia proposta é genérica e suficiente para ser aplicada com a finalidade de encontrar o modelo mais eficaz em outros conjuntos de dados de séries temporais, como, por exemplo, dados de faturamento de contas telefônicas, consumo diário de contrato de Internet, entre outros. Ressalta-se, mais uma vez, que o modelo mais eficaz poderá divergir do modelo encontrado neste trabalho.

Além de indicar um método automatizado e eficaz para detecção de pontos de interesse, este trabalho contribui com uma revisão bibliográfica abrangente e atualizada sobre detecção de *outliers* em diferentes áreas, com destaque em séries temporais obtidas do monitoramento dos serviços de *cloud computing* ([Seção 2.3](#)).

Outra contribuição relevante é a disponibilização de uma base de dados consolidada

do consumo de 10 meses de serviço em computação em nuvem na Administração Pública Federal. Apesar de aparentar um período sucinto, poucos órgãos aderiram aos serviços em nuvem com tamanho volume e período de tempo, fazendo com que o *dataset* disponibilizado facilite e incentive o trabalho de outros pesquisadores nesse tema. Ademais, os **códigos da implementação de todos os modelos na linguagem Python** ([LINO, 2021](#)) também estão disponibilizados para implantação e adaptação de outros analistas, desmistificando e impulsionando o assunto de computação em nuvem no serviço público.

4.2 Dificuldades, Limitações e Trabalhos Futuros

Primeira dificuldade a ser mencionada, é a incipiente utilização de serviços de computação em nuvem na Administração Pública Federal, o que resulta em demanda eminentemente flutuante e em escassez de dados. Outro obstáculo, relaciona-se à escolha dos parâmetros dos algoritmos ao se comparar diferentes modelos, uma vez que até mesmo pequenas modificações podem gerar resultados显著mente distintos.

No campo das limitações dos recursos computacionais, o tempo de execução de alguns modelos como SARIMA, MLP e LSTM foi obstáculo para a expansão e investigação de novos parâmetros e arquiteturas. Nesse sentido, em relação a trabalhos futuros, vislumbra-se a possibilidade de inserção do tempo de execução de cada modelo como métrica de avaliação, privilegiando modelos mais rápidos. Além disso, aumentar o número de especialistas rotulando os pontos de interesse e acrescentar séries com novas características pode promover maior robustez aos resultados.

Por fim, novas funcionalidades podem ser incorporadas na implementação dos modelos, como a utilização da técnica de Janelas Deslizantes para a definição do valor limite de demarcação de *outliers* e a criação de função de previsão de consumo futuro baseado nos dados históricos. Essa última ferramenta pode auxiliar de forma decisiva o gerenciamento de capacidade de utilização dos recursos tecnológicos e o controle financeiro dos contratos de serviço de computação em nuvem.

REFERÊNCIAS

- AGGARWAL, C. C. **Outlier Analysis**. 2. ed. Springer International Publishing, 2017. Disponível em: <<https://doi.org/10.1007/978-3-319-47578-3>>. Citado nas páginas 23 e 27.
- _____. **Outlier Analysis**. 1. ed. [S.l.]: Springer International Publishing, 2017. Citado nas páginas 30 e 31.
- AKOGLU, L.; TONG, H.; KOUTRA, D. Graph based anomaly detection and description: A survey. **Data Min. Knowl. Discov.**, Kluwer Academic Publishers, USA, v. 29, n. 3, p. 626–688, maio 2015. ISSN 1384-5810. Disponível em: <<https://doi.org/10.1007/s10618-014-0365-y>>. Citado na página 30.
- ALMAGUER-ANGELES, F.; MURPHY, J.; MURPHY, L.; PORTILLO, O. Choosing machine learning algorithms for anomaly detection in smart building iot scenarios. In: . [S.l.: s.n.], 2019. p. 491–495. Citado na página 30.
- ANDERBERG, M. **Cluster Analysis for Applications**. New York: Academic Press, 1973. Citado na página 29.
- BARNETT, V.; LEWIS, T. **Outliers in statistical data**. 2nd edition. ed. [S.l.]: John Wiley & Sons Ltd., 1978. Citado nas páginas 30 e 31.
- BAUDER, R.; KHOSHGOFTAAR, T. A probabilistic programming approach for outlier detection in healthcare claims. In: . [S.l.: s.n.], 2016. p. 347–354. Citado na página 30.
- BONTEMPS, L.; CAO, V. L.; McDERMOTT, J.; LE-KHAC, N.-A. Collective anomaly detection based on long short term memory recurrent neural network. 03 2017. Citado na página 30.
- BOUKERCHE, A.; ZHENG, L.; ALFANDI, O. Outlier detection: Methods, models, and classification. **ACM Computing Surveys**, v. 53, p. 1–37, 06 2020. Citado nas páginas 27 e 30.
- BRAEI, M.; WAGNER, S. Anomaly detection in univariate time-series: A survey on the state-of-the-art. 04 2020. Citado nas páginas 22, 27, 28 e 29.
- BRASIL. **Constituição da República Federativa do Brasil de 1988**, , art. 37, inc. XXI. Brasília, DF: [s.n.], 1988. Disponível em: <http://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em: 01 out. 2020. Citado na página 17.
- BRASIL. **Lei nº 8.666, de 21 de Junho de 1993. Artigo 67**. 1993. Citado na página 17.
- BREUNIG, M.; KRIEGEL, H.-P.; NG, R.; SANDER, J. Lof: Identifying density-based local outliers. In: . [S.l.: s.n.], 2000. v. 29, p. 93–104. Citado nas páginas 23 e 29.
- CAMPOS, O. **Data Analytics Transparente para Descoberta de Padrões e Anomalias na Realização de Convênios e Contratos de Repasse Federais**. Dissertação (Mestrado) — Instituto Federal de Sergipe, 2018. Citado nas páginas 30 e 31.

CAPELLEVEEN, G. van; POEL, M.; MUELLER, R.; THORNTON, D.; HILLEGERSBERG, J. Outlier detection in healthcare fraud: A case study in the medicaid dental domain. **International Journal of Accounting Information Systems**, v. 21, p. 18–31, 06 2016. Citado na página 30.

CGU. **Detalhamento da Despesa Pública**. 2020. Disponível em: <<http://transparencia.gov.br/url/9306958e>>. Acesso em: 07/04/2020. Citado na página 17.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: A survey. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 41, n. 3, jul. 2009. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/1541880.1541882>>. Citado nas páginas 21, 22, 27, 30 e 31.

CHANG, I.; TIAO, G.; CHEN, C. Estimation of time series parameters in the presence of outliers. **Technometrics**, v. 30, p. 193–204, 05 1988. Citado na página 30.

CHAUDHARY, K.; YADAV, J.; MALLICK, B. A review of fraud detection techniques: Credit card. **International Journal of Computer Applications**, v. 45, 01 2012. Citado na página 30.

CHEN, C.; LIU, L.-M. Joint estimation of model parameters and outlier effects in time series. **Journal of the American Statistical Association**, Taylor & Francis, v. 88, n. 421, p. 284–297, 1993. Disponível em: <<https://doi.org/10.1080/01621459.1993.10594321>>. Citado na página 30.

CHEN, S.; WANG, W.; ZUYLEN, H. van. A comparison of outlier detection algorithms for its data. **Expert Syst. Appl.**, v. 37, p. 1169–1178, 03 2010. Citado na página 30.

COMPRAZNET. **Consulta Licitações. Pregão nº 29/2018**. 2018. Disponível em: <<http://www.comprasnet.gov.br/ConsultaLicitacoes/Download/Download.asp?coduasg=201004&numprp=292018&modprp=5&bidbird=N>>. Acesso em: 01/10/2020. Citado na página 34.

DAVIS, J.; GOADRICH, M. The relationship between precision-recall and roc curves. In: **Proceedings of the 23rd International Conference on Machine Learning**. New York, NY, USA: Association for Computing Machinery, 2006. (ICML '06), p. 233–240. ISBN 1595933832. Disponível em: <<https://doi.org/10.1145/1143844.1143874>>. Citado na página 43.

EHLERS, R. **Análise de Séries Temporais**. 2009. Disponível em: <<http://www.icmc.usp.br/eplers/stemp/stemp.pdf>>. Acesso em: 10/10/2020. Citado na página 25.

ESTER, M.; KRIEGEL, H.-P.; SANDER, J.; XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In: . [S.l.: s.n.], 1996. v. 96, p. 226–231. Citado na página 29.

FOX, A. J. Outliers in time series. **Journal of the Royal Statistical Society. Series B (Methodological)**, v. 34, n. 3, p. 350–363, 1972. Citado nas páginas 29 e 30.

GAO, L.; WU, S. Response score of deep learning for out-of-distribution sample detection of medical images. **Journal of Biomedical Informatics**, v. 107, p. 103442, 05 2020. Citado na página 30.

GRUBBS, F. Procedure for detecting outlying observations in samples. **Technometrics**, v. 11, p. 53, 04 1974. Citado nas páginas 29 e 31.

- GUPTA, M.; GAO, J.; AGGARWAL, C.; HAN, J. Outlier detection for temporal data: A survey. **IEEE Transactions on Knowledge and Data Engineering**, IEEE Computer Society, v. 26, n. 9, p. 2250–2267, set. 2014. ISSN 1041-4347. Citado na página 30.
- HASELSTEINER, E.; PFURTSCHELLER, G. Using time-dependent neural networks for eeg classification. **IEEE transactions on rehabilitation engineering : a publication of the IEEE Engineering in Medicine and Biology Society**, v. 8, p. 457–63, 01 2001. Citado na página 29.
- HAUSKRECHT, M.; BATAL, I.; VALKO, M.; VISWESWARAN, S.; COOPER, G.; CLERMONT, G. Outlier detection for patient monitoring and alerting. **Journal of biomedical informatics**, v. 46, 08 2012. Citado na página 30.
- HAWKINS, D. **Identification of outliers**. London [u.a.]: Chapman and Hall, 1980. (Monographs on applied probability and statistics). Citado nas páginas 21 e 30.
- HODGE, V.; AUSTIN, J. A survey of outlier detection methodologies. **Artificial Intelligence Review**, v. 22, p. 85–126, 10 2004. Citado nas páginas 22 e 27.
- HUANG, C.; MIN, G.; WU, Y.; YING, Y.; PEI, K.; XIANG, Z. Time series anomaly detection for trustworthy services in cloud computing systems. **IEEE Transactions on Big Data**, PP, p. 1–1, 06 2017. Citado nas páginas 30 e 31.
- JABEZ, J.; MUTHUKUMAR, B. Intrusion detection system (IDS): Anomaly detection using outlier detection approach. **Procedia Computer Science**, Elsevier BV, v. 48, p. 338–346, 2015. Disponível em: <<https://doi.org/10.1016/j.procs.2015.04.191>>. Citado na página 30.
- Jeni, L. A.; Cohn, J. F.; De La Torre, F. Facing imbalanced data—recommendations for the use of performance metrics. In: **2013 Humaine Association Conference on Affective Computing and Intelligent Interaction**. [S.l.: s.n.], 2013. p. 245–251. Citado na página 43.
- KALEKAR, P. Time series forecasting using holt-winters exponential smoothing. **Time Series Forecasting Using Holt-Winters Exponential Smoothing**, 01 2004. Citado na página 29.
- KIM, T.-Y.; CHO, S. Web traffic anomaly detection using c-lstm neural networks. **Expert Systems with Applications**, v. 106, 04 2018. Citado na página 29.
- KNORR, E.; NG, R. Algorithms for mining distance-based outliers in large datasets. **VLDB**, 06 1998. Citado na página 23.
- LAPIX. **Avaliando, Validando e Testando o seu Modelo: Metodologias de Avaliação de Performance**. 2020. Disponível em: <<http://www.lapix.ufsc.br/wp-content/uploads/2019/08/AUC-curve-98.jpg>>. Acesso em: 10/12/2020. Citado na página 45.
- LAZAREVIC, A.; ERTÖZ, L.; KUMAR, V.; OZGUR, A.; SRIVASTAVA, J. A comparative study of anomaly detection schemes in network intrusion detection. In: . [S.l.: s.n.], 2003. v. 3. Citado nas páginas 30 e 31.
- LI, X.; LI, Z.; HAN, J.; LEE, J.-G. Temporal outlier detection in vehicle traffic data. In: . [S.l.: s.n.], 2009. p. 1319–1322. Citado na página 30.
- LINO, V. **Repositório de Detecção de Outliers em Contrato de Nuvem**. 2021. Disponível em: <https://github.com/victordml/Deteccao_Outliers_Nuvem.git>. Acesso em: 07/01/2021. Citado nas páginas 36 e 54.

Liu, F. T.; Ting, K. M.; Zhou, Z. Isolation forest. In: **2008 Eighth IEEE International Conference on Data Mining**. [S.l.: s.n.], 2008. p. 413–422. Citado na página 29.

LIU, H.; LI, X.; LI, J.; ZHANG, S. Efficient outlier detection for high-dimensional data. **IEEE Transactions on Systems, Man, and Cybernetics: Systems**, PP, p. 1–11, 07 2017. Citado na página 30.

MALHOTRA, P.; RAMAKRISHNAN, A.; ANAND, G.; VIG, L.; AGARWAL, P.; SHROFF, G. Lstm-based encoder-decoder for multi-sensor anomaly detection. 07 2016. Citado nas páginas 29, 30, 31 e 45.

MALINI, N.; PUSHPA, M. Analysis on credit card fraud identification techniques based on knn and outlier detection. In: . [S.l.: s.n.], 2017. p. 255–258. Citado na página 30.

MANDHARE, H.; IDATE, S. A comparative study of cluster based outlier detection, distance based outlier detection and density based outlier detection techniques. In: . [S.l.: s.n.], 2017. p. 931–935. Citado na página 18.

MAYA, S.; UENO, K.; NISHIKAWA, T. dlstm: a new approach for anomaly detection using deep learning with delayed prediction. **International Journal of Data Science and Analytics**, v. 8, 09 2019. Citado na página 44.

MCKENZIE, E. Comments on ‘exponential smoothing: The state of the art’ by e. s. gardner jr. **Journal of Forecasting**, v. 4, 01 1985. Citado na página 29.

MINGQIANG, Z.; HUI, H.; QIAN, W. A graph-based clustering algorithm for anomaly intrusion detection. In: . [S.l.: s.n.], 2012. p. 1311–1314. ISBN 978-1-4673-0241-8. Citado na página 30.

MINISTÉRIO DA ECONOMIA. **Instrução Normativa nº 1, de 4 de abril de 2019**. Brasília, DF: [s.n.], 2019. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/70267659/do1-2019-04-05-instrucao-normativa-n-1-de-4-de-abril-de-2019-70267535>. Acesso em: 01 out. 2020. Citado na página 18.

_____. **Instrução Normativa nº 1, de 4 de abril de 2019, art. 29**. Brasília, DF: [s.n.], 2019. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/70267659/do1-2019-04-05-instrucao-normativa-n-1-de-4-de-abril-de-2019-70267535>. Acesso em: 01 out. 2020. Citado na página 17.

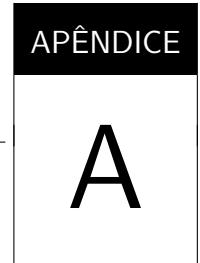
_____. **Economia abre consulta pública para contratar serviços de computação em nuvem**. 2020. Disponível em: <<https://www.gov.br/economia/pt-br/assuntos/noticias/2020/junho/economia-abre-consulta-publica-para-contratar-servicos-de-computacao-em-nuvem>>. Acesso em: 01/10/2020. Citado na página 18.

MUNIR, M.; SIDDIQUI, S.; DENGEL, A.; AHMED, S. Deepant: A deep learning approach for unsupervised anomaly detection in time series. **IEEE Access**, PP, p. 1–1, 12 2018. Citado nas páginas 30, 31 e 44.

NAVAZ, A. S. S.; SANGEETHA, V.; PRABHADEVI, C. Entropy based anomaly detection system to prevent ddos attacks in cloud. **International Journal of Computer Applications**, v. 62, p. 42–47, 08 2013. Citado na página 30.

- PANDEESWARI, N.; KUMAR, G. Anomaly detection system in cloud environment using fuzzy clustering based ann. **Mobile Networks and Applications**, v. 21, 08 2015. Citado nas páginas 30 e 31.
- PAWAR, M.; KALAVADEKAR, P.; TAMBE, M. A survey on outlier detection techniques for credit card fraud detection. **IOSR Journal of Computer Engineering**, v. 16, p. 44–48, 01 2014. Citado na página 30.
- PRATI, R.; BATISTA, G.; MONARD, M. **Curvas ROC para avaliação de classificadores**. 2008. 215-222 p. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.do?arnumber=4609920&isnumber=4609907>>. Citado na página 45.
- RO, K.; ZOU, C.; WANG, Z.; YIN, G. Outlier detection for high-dimensional data. **Biometrika**, v. 102, n. 3, p. 589–599, 2015. Disponível em: <<https://EconPapers.repec.org/RePEc:oup:biomet:v:102:y:2015:i:3:p:589-599>>. Citado na página 30.
- ROSA, G. The elements of statistical learning: Data mining, inference, and prediction by hastie, t., tibshirani, r., and friedman, j. **Biometrics**, v. 66, 12 2010. Citado na página 29.
- ROUSSEEUW, P. J.; LEROY, A. M. **Robust Regression and Outlier Detection**. USA: John Wiley & Sons, Inc., 1987. ISBN 0471852333. Citado na página 30.
- SAMAL, K.; BABU, K.; DAS, S.; ACHARYA, A. Time series based air pollution forecasting using sarima and prophet model. In: . [S.l.: s.n.], 2019. p. 80–85. ISBN 978-1-4503-7228-2. Citado na página 29.
- SHAN, Y.; MURRAY, D.; SUTINEN, A. Discovering inappropriate billings with local density based outlier detection method. In: . [S.l.: s.n.], 2009. v. 101, p. 93–98. Citado nas páginas 30 e 31.
- SINGH, J.; AGGARWAL, S. Survey on outlier detection in data mining. **International Journal of Computer Applications**, v. 67, p. 29–32, 04 2013. Citado na página 30.
- SOETRISNO, Y. A. A.; HANDOYO, E.; ILYASA, M.; DENIS, M.; SINURAYA, E. T-series analysis for predicting apple prices in indonesian market using the sarima method. In: . [S.l.: s.n.], 2019. p. 1–6. Citado na página 29.
- SOMBOONSAK, P. Time series analysis of dengue fever cases in thailand utilizing the sarima model. In: . [S.l.: s.n.], 2019. p. 439–444. Citado na página 29.
- TAYLOR, S. J.; LETHAM, B. Forecasting at scale. PeerJ, set. 2017. Disponível em: <<https://doi.org/10.7287/peerj.preprints.3190v2>>. Citado na página 29.
- WIKIPÉDIA. **Precisão e revocação**. 2020. Disponível em: <https://pt.wikipedia.org/wiki/Precis%C3%A3o_e_revoc%C3%A3o>. Acesso em: 11/10/2020. Citado na página 44.
- YAO, Y.; WEI, Y.; FU-XIANG, G.; GE, Y. Anomaly intrusion detection approach using hybrid mlp/cnn neural network. In: . [S.l.: s.n.], 2006. v. 2, p. 1095 – 1102. Citado na página 29.
- ZHANG, J. Advancements of outlier detection: A survey. **ICST Transactions on Scalable Information Systems**, v. 13, p. e2, 02 2013. Citado na página 30.

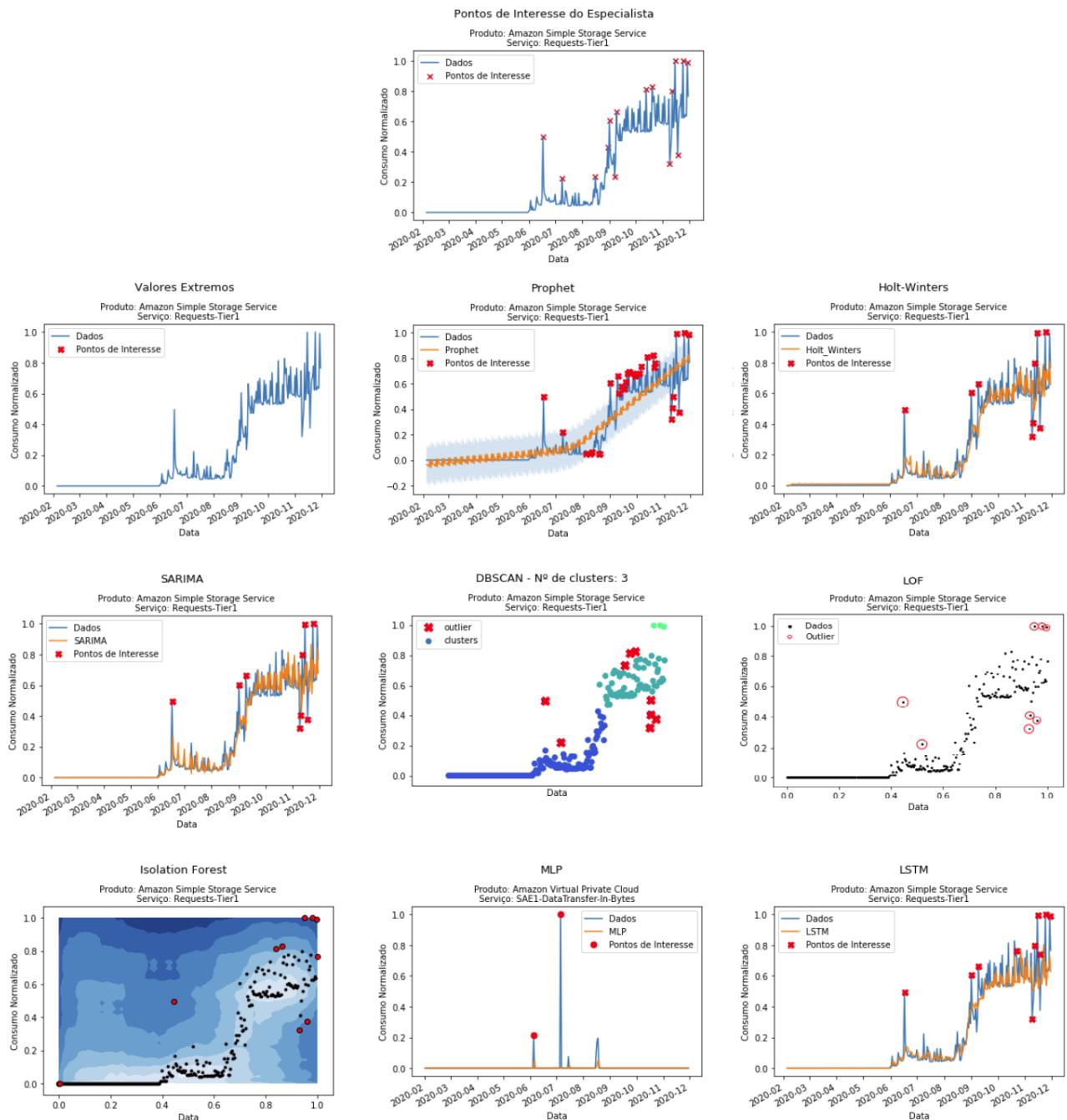
ZIMEK, A.; SCHUBERT, E.; KRIEGEL, H.-P. A survey on unsupervised outlier detection in high-dimensional numerical data. **Statistical Analysis and Data Mining**, v. 5, p. 363–387, 10 2012. Citado nas páginas [27](#) e [30](#).



PONTOS DE INTERESSE DETECTADOS POR CADA MODELO

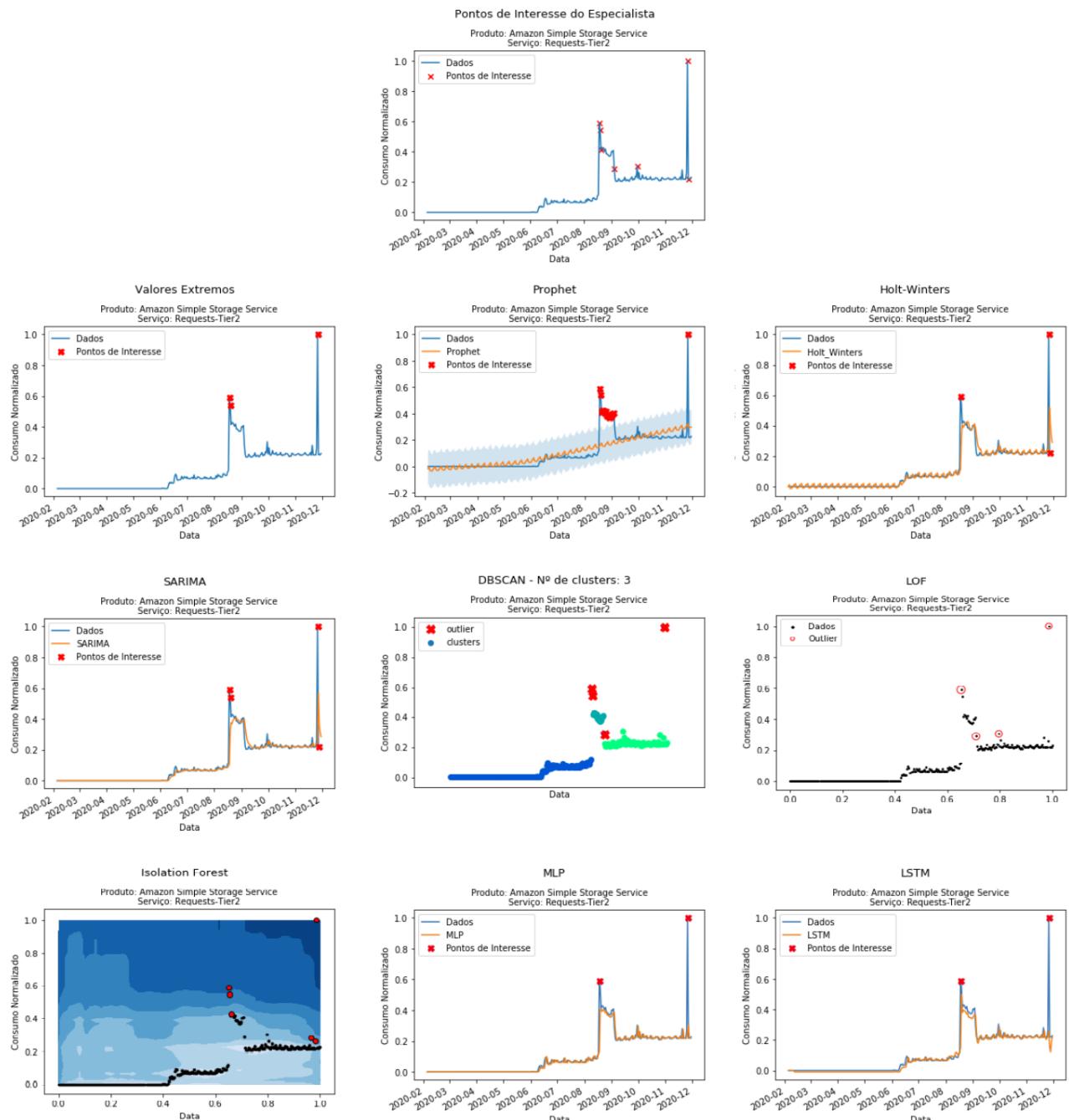
Este apêndice apresenta a visualização de cada série temporal e os pontos de interesse apontados pelo especialista na parte superior, com as figuras dos pontos de interesse obtidos por cada modelo nas linhas inferiores.

Figura 27 – Pontos de Interesse na série Amazon Simple Storage Service - Requests-Tier1



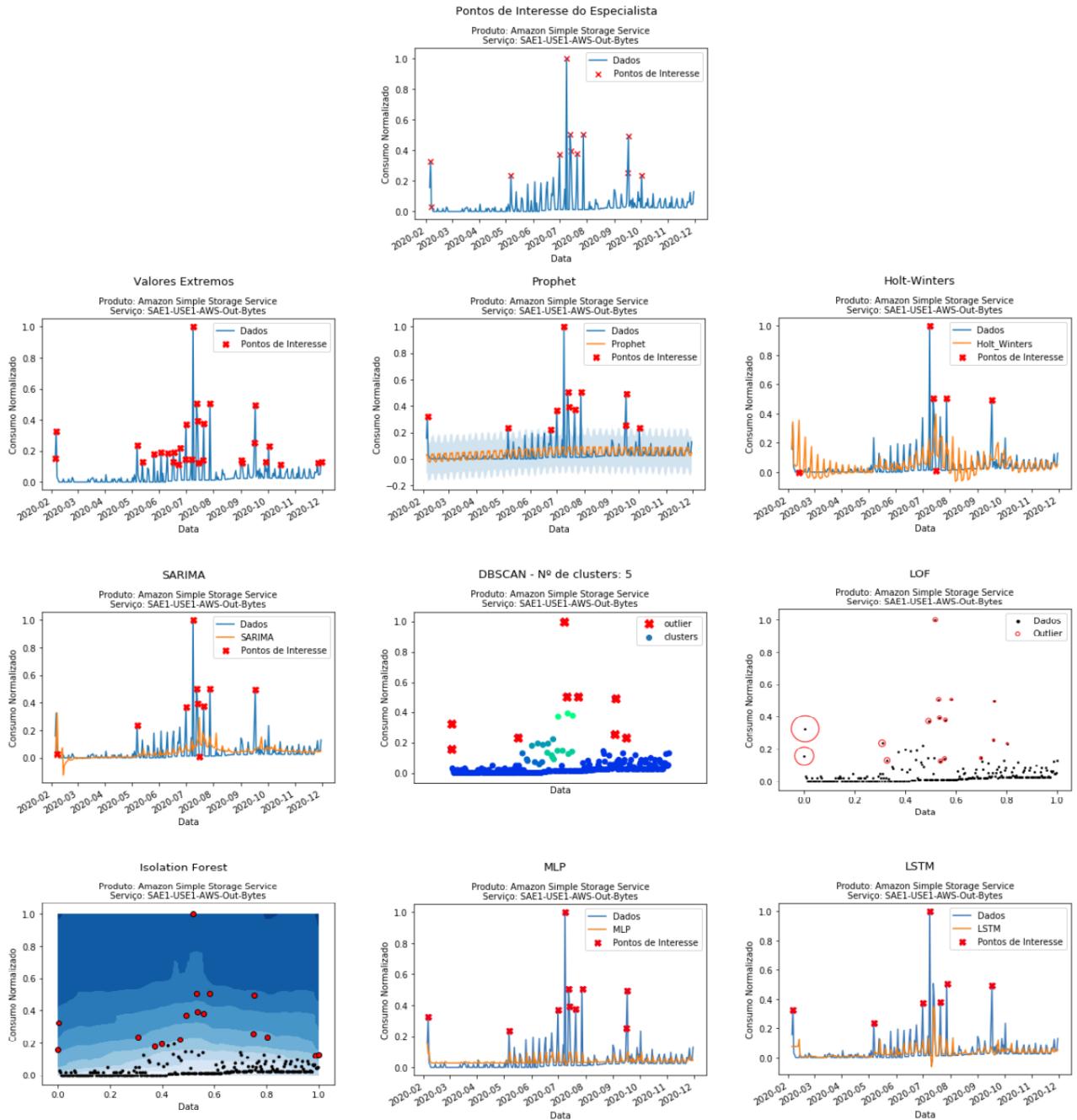
Fonte: Elaborada pelo autor.

Figura 28 – Pontos de Interesse na série Amazon Simple Storage Service - Requests-Tier2



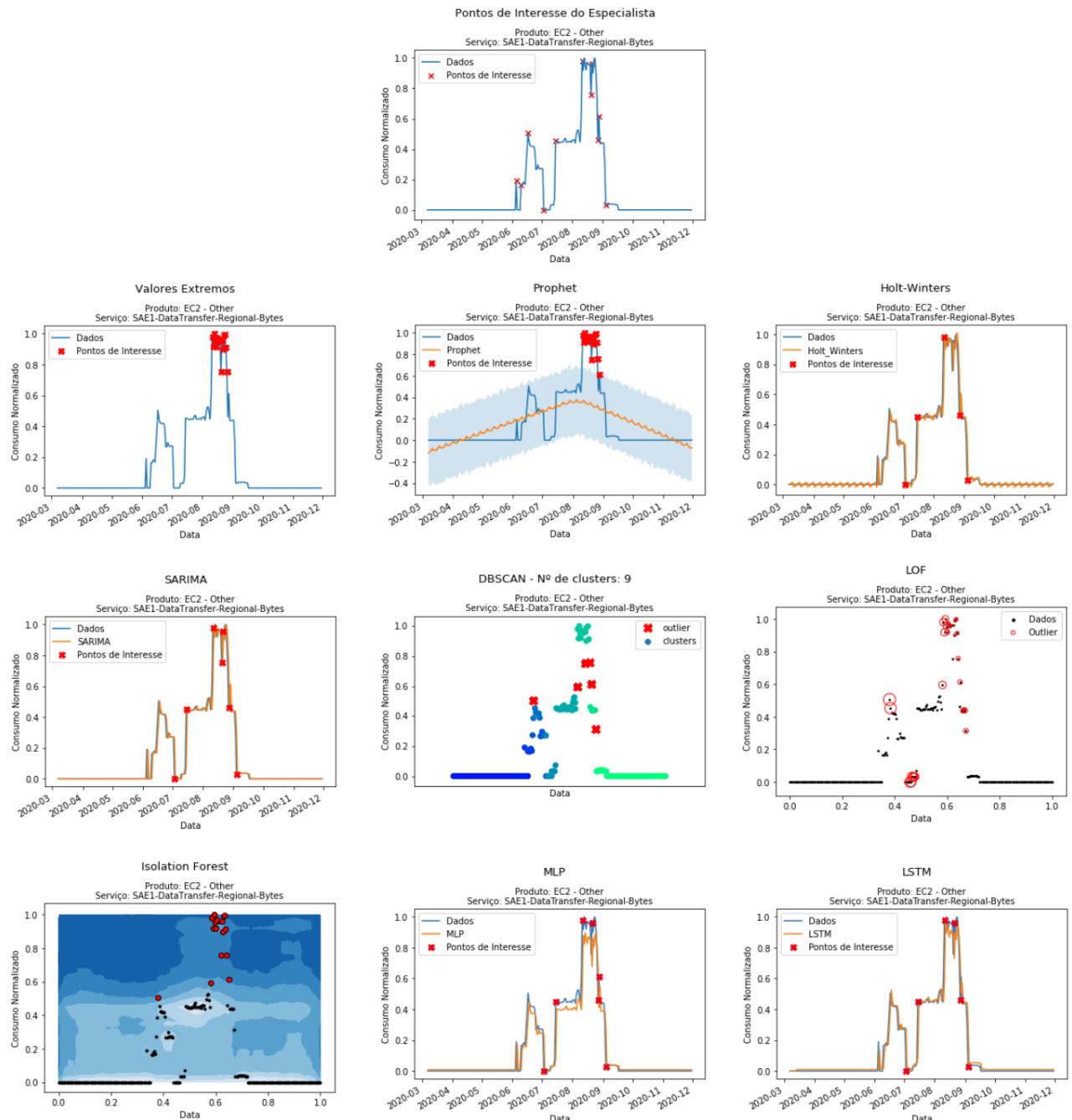
Fonte: Elaborada pelo autor.

Figura 29 – Pontos de Interesse na série Amazon Simple Storage Service - SAE1-USE1-AWS-Out-Bytes



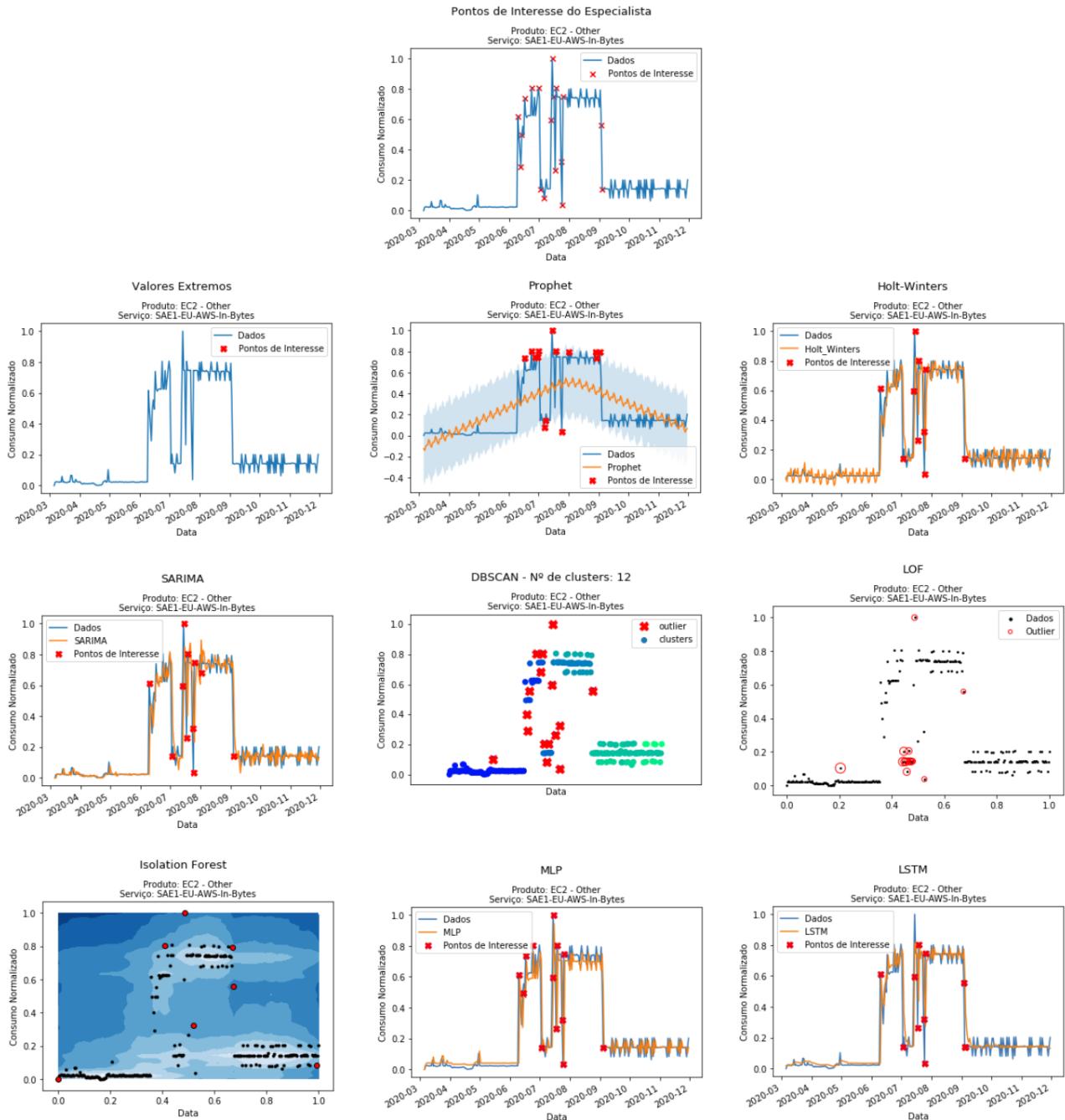
Fonte: Elaborada pelo autor.

Figura 30 – Pontos de Interesse na série EC2 – Other - SAE1-DataTransfer-Regional-Bytes



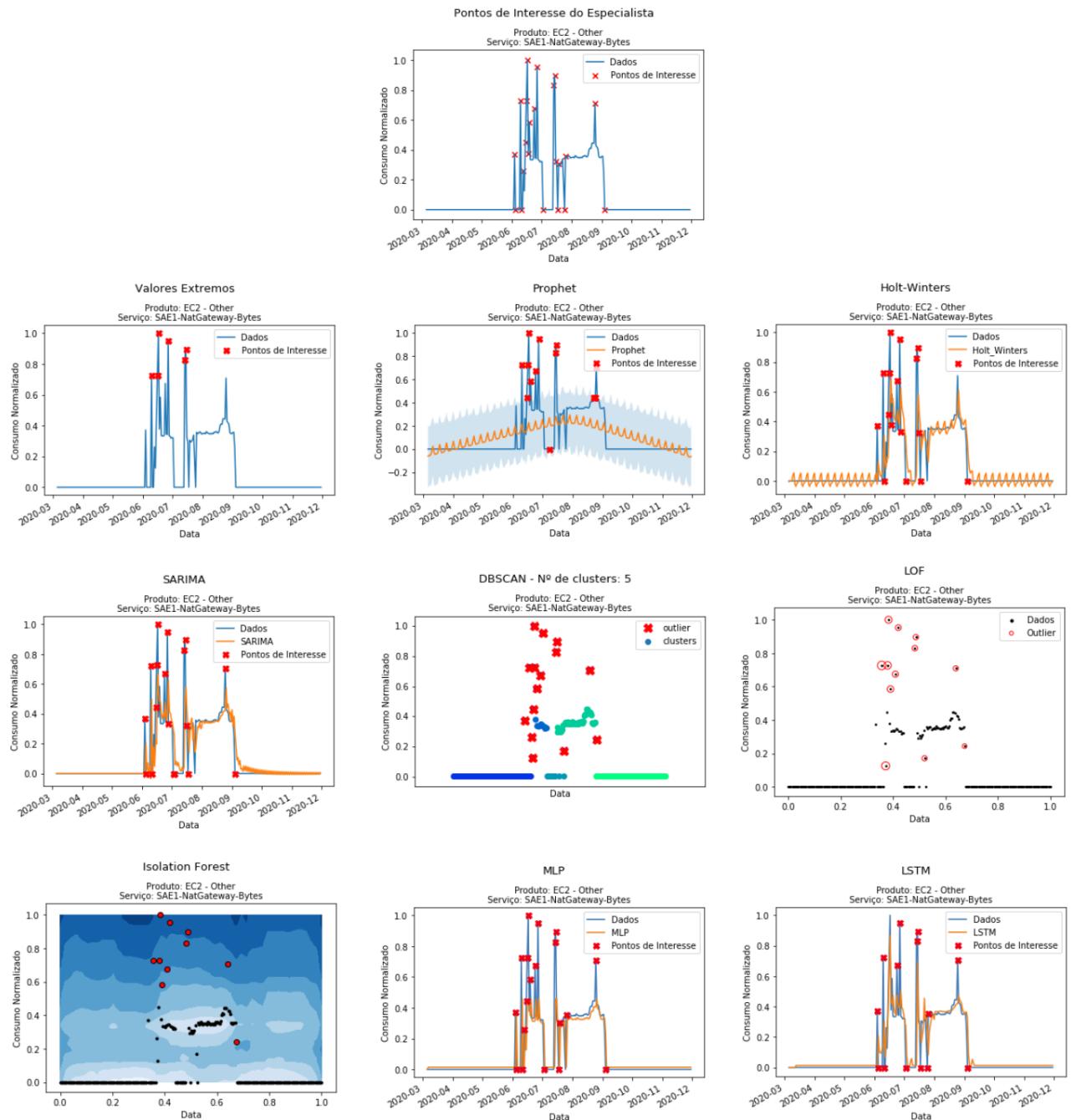
Fonte: Elaborada pelo autor.

Figura 31 – Pontos de Interesse na série EC2 – Other - SAE1-EU-AWS-In-Bytes



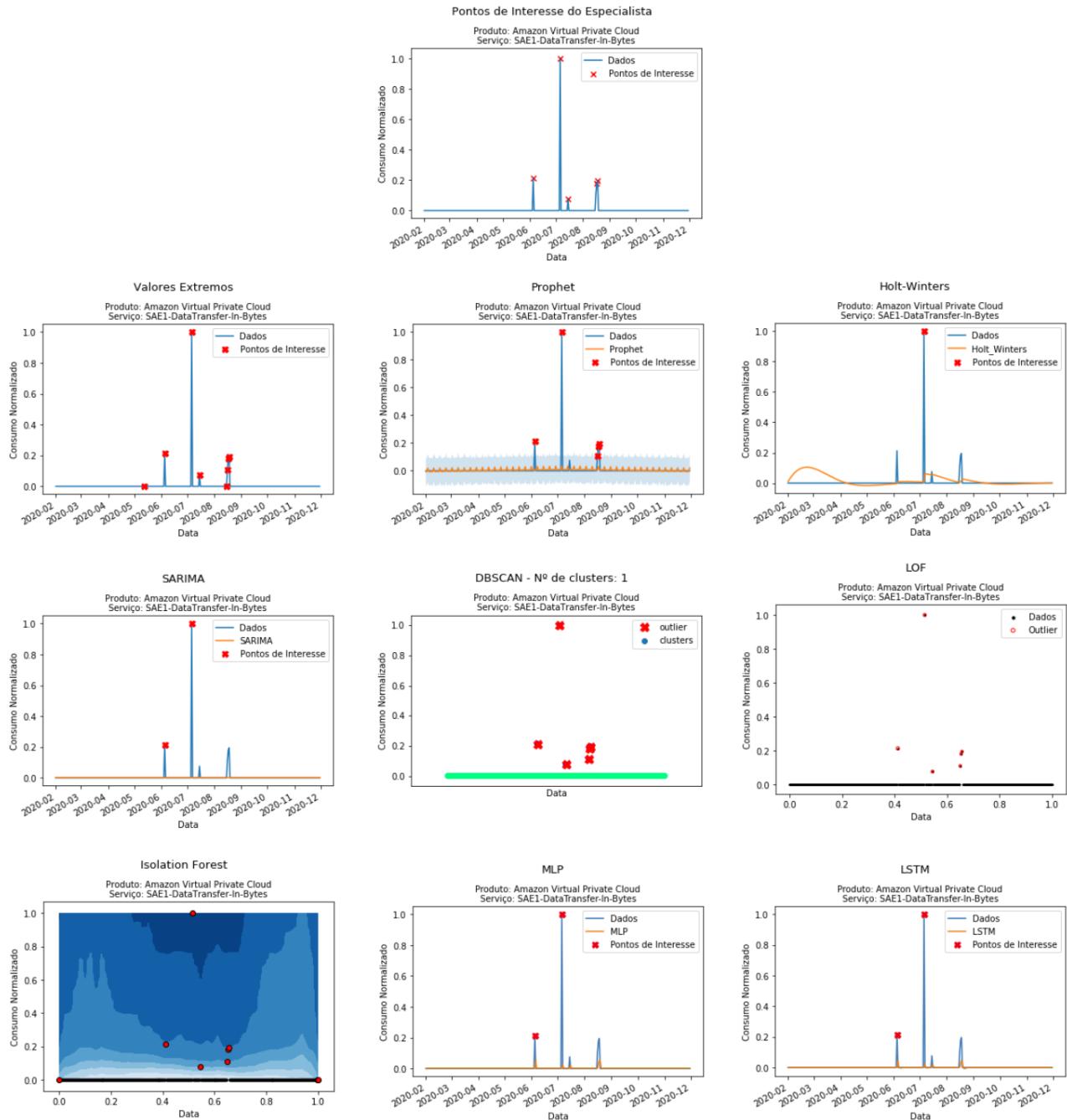
Fonte: Elaborada pelo autor.

Figura 32 – Pontos de Interesse na série EC2 – Other - SAE1-NatGateway-Bytes



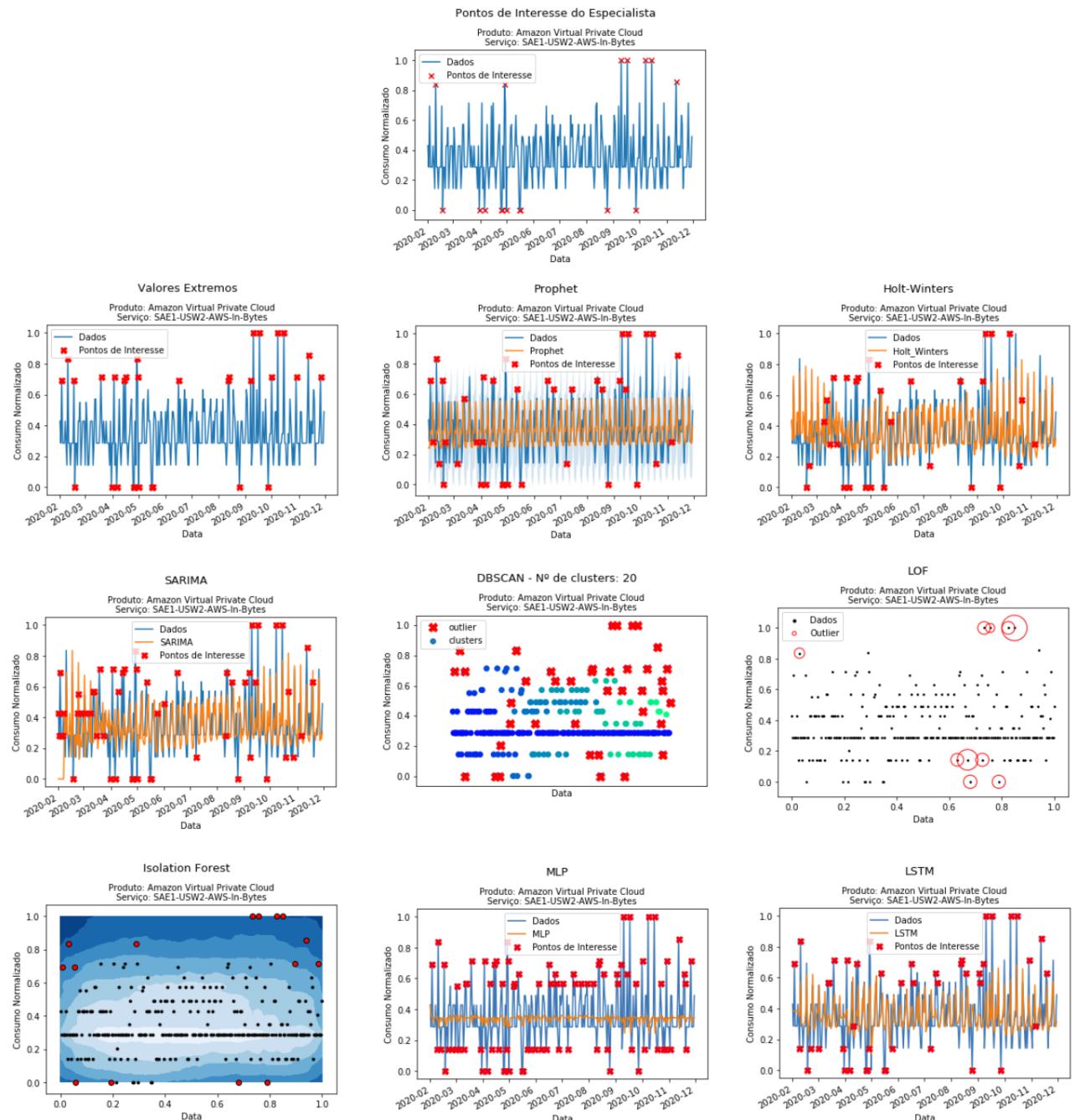
Fonte: Elaborada pelo autor.

Figura 33 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-DataTransfer-In-Bytes



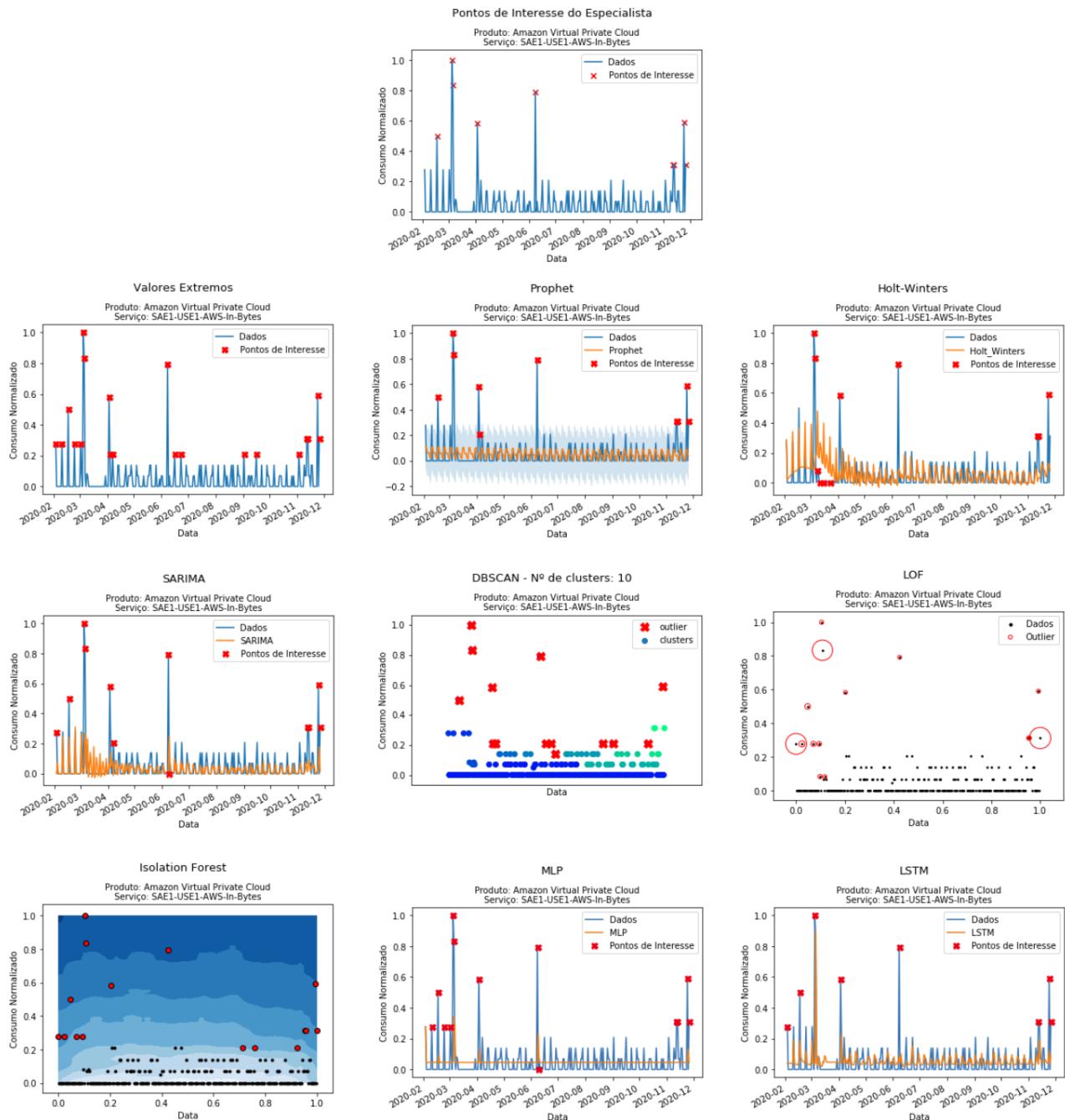
Fonte: Elaborada pelo autor.

Figura 34 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-USW2-AWS-In-Bytes



Fonte: Elaborada pelo autor.

Figura 35 – Pontos de Interesse na série Amazon Virtual Private Cloud - SAE1-USE1-AWS-In-Bytes



Fonte: Elaborada pelo autor.

