

Dissertação apresentada à Pró-Reitoria de Pós-Graduação do Instituto Tecnológico de Aeronáutica, como parte dos requisitos para obtenção do título de Mestre em Engenharia do Curso de Mestrado Profissional em Computação Aeronáutica.

Michel Santos da Silva

IDENTIFICAÇÃO DE EVENTOS DISRUPTIVOS EM REDES

CORE 5G SA

Dissertação aprovada em sua versão final pelos abaixo assinados:


Prof. Dr. Lourenço Alves Pereira Júnior
Orientador

Prof. Dr. André Valdetaro Gomes Cavalieri
Pró-Reitor de Pós-Graduação

Campo Montenegro
São José dos Campos, SP – Brasil
2024

Dados Internacionais de Catalogação-na-Publicação (CIP)
Divisão de Informação e Documentação

Santos da Silva, Michel

Identificação de eventos disruptivos em Redes Core 5G SA

São José dos Campos, 2024.

101f.

Dissertação de mestrado – Mestrado profissional em computação aeronáutica – Instituto Tecnológico de Aeronáutica, 2024. Orientador: Prof. Dr. Lourenço Alves Pereira Júnior

1. Detecção de Anomalia em Séries temporais. 2. Engenharia de Feature. 3. Modelos de Aprendizado.
I. Instituto Tecnológico de Aeronáutica. II. Título

REFERÊNCIA BIBLIOGRÁFICA

Silva, Michel Santos da. **Identificação de eventos disruptivos em redes core 5G SA**. Ano de depósito. 101 f. Dissertação de Mestrado Profissional em Computação de Missão Crítica – Instituto Tecnológico de Aeronáutica, São José dos Campos, 2024.

CESSÃO DE DIREITOS

NOME DO AUTOR: Michel Santos da Silva

TÍTULO DO TRABALHO: Identificação de eventos disruptivos em redes core 5G SA

TIPO DO TRABALHO/ANO: Dissertação / 2024

É concedida ao Instituto Tecnológico de Aeronáutica permissão para reproduzir cópias desta dissertação e para emprestar ou vender cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desta dissertação ou tese pode ser reproduzida sem a sua autorização (do autor).



Michel Santos da Silva

Rua Engenheiro Godofredo dos Santos, 93, Estoril

CEP: 30494-220, Belo Horizonte - MG

IDENTIFICAÇÃO DE EVENTOS DISRUPTIVOS EM REDES CORE 5G SA

MICHEL SANTOS DA SILVA

Composição da Banca Examinadora:

| | | |
|---|------------|--------|
| Prof. Dr. Lourenço Alves Pereira Júnior | Presidente | - ITA |
| Prof. Dr. Lourenço Alves Pereira Júnior | Orientador | - ITA |
| Prof. Dr. Paulo André Lima de Castro | | - ITA |
| Prof. Dr. Eduardo James Pereira Souto | | - UFAM |

ITA

Agradecimentos

Eu gostaria de agradecer primeiramente a minha família, em especial a minha esposa Cristina, que mesmo em meio a situações desfavoráveis nunca deixou de me apoiar e incentivar em todos os momentos. Aos meus pais que sempre me incentivaram a vencer através dos estudos. Aproveito para agradecer também as minhas filhas, Lívia e Luísa, que com sorrisos me dão forças em muitas situações.

Agradeço aos meus companheiros da Soka Gakkai, que sempre estão presentes para me incentivar a buscar os meus objetivos e ser uma pessoa melhor. Em especial, agradeço ao presidente Ikeda, que me fez acreditar desde novo que sou capaz de conquistar todos os meus sonhos e criou em mim um propósito.

Agradeço a diretoria de Core Operations da TIM Brasil que não somente permitiu a utilização dos dados para a realização desta pesquisa quanto me incentivou na realização desta. Em especial meu agradecimento ao colega Adriano Rocha, cujo papel foi importantíssimo durante toda etapa de codificação, testes e feedbacks a empresa acima citada.

Agradeço ao meu orientador, professor Dr. Lourenço Alves, que me orientou em todos os momentos com extrema atenção e paciência.

Aos meus colegas de mestrado que em todas as situações de dificuldade se mantiveram unidos e predispostos a ajudar. Em especial meus agradecimentos ao Carlos Travagini e Kenniston Arraes por todo suporte.

*"Não faz mal que seja pouco,
O que importa é que o avanço de hoje
Seja maior que o de ontem.
Que nossos passos de amanhã
Sejam mais largos que os de hoje.
Atuem agora e vivam o presente
Com a certeza que neste exato instante
Está se erguendo o futuro".*

(Daisaku Ikeda)

Resumo

A quinta geração das redes de telefonia móvel, a rede 5G, traz consigo grandes expectativas para novas aplicações devido a sua baixa latência, alta taxa de transferência de dados e grande disponibilidade. Esta nova arquitetura é muito mais flexível e escalonável, sendo a mesma baseada em tecnologias como Network Function Virtualization (NFV), Software-Defined Networking (SDN) e Service-Based Architecture (SBA), diferente das gerações anteriores que a expansão da rede era associada a alterações de hardware. No entanto, não houve melhorias no processo de monitoração da rede, o que abre brechas para a não identificação de eventos disruptivos que podem gerar interrupção do serviço. A rede core centraliza todo o tráfego, sendo responsável por funções como autenticação e gerenciamento de sessão dos usuários, sendo essa o cerne da nova especificação. Uma falha nesta camada da rede é capaz de deixar todo o sistema indisponível, logo o tempo de identificação de eventos disruptivos na rede deve ser o mais breve e assertivo possível. Monitorar completamente esta parte da rede traz desafios devido ao grande número de indicadores de desempenho disponíveis, cada um destes com um perfil de comportamento específico e mutável no tempo. Este trabalho propõe um framework cuja finalidade vai da diminuição da dimensão do que precisa ser analisado (engenharia de característica) à detecção de eventos disruptivos utilizando modelagem em séries temporais e métodos de inteligência artificial não supervisionado. Para a validação do framework proposto todos os testes são realizados com dados reais coletados de uma rede comercial em operação. Serão apresentados neste trabalho os resultados obtidos, indo desde as métricas tradicionais utilizadas em aprendizagem de máquina até o tempo gasto durante o processo, assim como a viabilidade de implementação do framework em uma grande empresa de telecomunicações. Por fim, também serão debatidas possibilidades de trabalhos futuros.

Abstract

The fifth generation of mobile phone networks, the 5G network, brings with it great expectations for new applications due to its low latency, high data transfer rate and great availability. This new architecture is much more flexible and scalable, being based on technologies such as Network Function Virtualization (NFV), Software-Defined Networking (SDN) and Service-Based Architecture (SBA), unlike previous generations in which network expansion was associated with hardware changes. However, there were no improvements in the network monitoring process, which leaves room for the non-identification of disruptive events that could lead to service interruption. The core network centralizes all traffic, being responsible for functions such as authentication and user session management, which is the core of the new specification. A failure at this layer of the network can leave the entire system unavailable, so the time for identifying disruptive events in the network must be as brief and assertive as possible. Completely monitoring this part of the network brings challenges due to the large number of performance indicators available, each with a specific behavior profile that changes over time. This work proposes a framework whose purpose ranges from reducing the size of what needs to be analyzed (feature engineering) to detecting disruptive events using time series modeling and unsupervised artificial intelligence methods. To validate the proposed framework, all tests are carried out with real data collected from an operating commercial network. The results obtained will be presented in this work, ranging from the traditional metrics used in machine learning to the time spent during the process, as well as the feasibility of implementing the framework in a large telecommunications company. Finally, possibilities for future work will also be discussed.

Lista de Figuras

| | |
|--|----|
| Figura 1 – Conceito de rede 5G..... | 16 |
| Figura 2 – Processo de identificação de falhas..... | 17 |
| Figura 3 – Tipo de Hardware utilizados em gerações anteriores de telecomunicações. | 18 |
| Figura 4 – Conceito de Zero Touch Network..... | 25 |
| Figura 5 – Exemplo de solução proposta que utiliza poucos KPIs (Chakraborty, Corici e Magedanz 2021). | 28 |
| Figura 6 – Exemplo de framework que utiliza dados coletados em tempo real (Zhou, Liu, Fu, Cheng, Fu e Zhao 2020) | 30 |
| Figura 7 – Comunicação entre NE e NWDAF para coleta de dados. | 31 |
| Figura 8 – Framework Genérico para automações na rede 5G. | 32 |
| Figura 9 – Conceito de Arquitetura utilizando NWDAF distribuído. | 32 |
| Figura 10 – Métodos de detecção de anomalia em séries temporais (Schmidl, Wenig e Papenbrock 2022). | 34 |
| Figura 11 – Arquitetura Conceitual da rede Core Padrão 3GPP. | 39 |
| Figura 12 – Descrição das funções de rede existentes na TIM BR. | 40 |
| Figura 13 – Rede Standalone e Non-standalone..... | 41 |
| Figura 14– Séries temporais do mesmo indicador de desempenho e elemento com comportamentos diferentes..... | 44 |
| Figura 15 – Série temporal com comportamento incremental. | 44 |
| Figura 16 – Série temporal com tendência estática. | 45 |
| Figura 17 – Série temporal com sazonalidade devido ao tráfego..... | 45 |
| Figura 18 – Série temporal com comportamento randômico. | 46 |
| Figura 19 – Série temporal com evento anômalo. | 46 |
| Figura 20 – Série Temporal que apresenta mudança de comportamento devido a configuração. | 47 |
| Figura 21 – Exemplo de KPI com mudanças de tendência. | 48 |

| | |
|--|----|
| Figura 22 – KPI que apresenta mudança de comportamento devido ao feriado de 12/10. | 48 |
| Figura 23 – Diagrama de Camadas do filtro dinâmico..... | 51 |
| Figura 24 – Exemplo de série temporal que seria filtrada pela entropia de Shannon. ... | 52 |
| Figura 25 – Decomposição multiplicativa..... | 53 |
| Figura 26 – Decomposição aditiva | 53 |
| Figura 27 – Série temporal excluída pela camada 2..... | 54 |
| Figura 28 – 2 séries diferentes com o mesmo parâmetro e com alta correlação durante evento anômalo..... | 55 |
| Figura 29 – Série temporal 1 | 61 |
| Figura 30 – Série temporal 2 | 62 |
| Figura 31 – Previsão da série temporal 1, predição de 7 dias antes do evento anômalo. | 64 |
| Figura 32 – Previsão da série temporal 1, predição de 7 dias antes do evento anômalo. | 65 |
| Figura 33 – Previsão da série temporal 1, predição de 15 dias antes do evento anômalo. | 66 |
| Figura 34 – Previsão da série temporal 2, predição de 15 dias antes do evento anômalo. | 67 |
| Figura 35 – Previsão da série temporal 1, predição de 30 dias antes do evento anômalo. | 68 |
| Figura 36 – Previsão da série temporal 2, predição de 30 dias antes do evento anômalo. | 69 |
| Figura 37 – Distribuição por família dos métodos utilização..... | 72 |
| Figura 38 – Exemplos de falhas DRRJO1..... | 74 |
| Figura 39 – Resultados da acurácia para os testes no DDRJO1..... | 78 |
| Figura 40 – Resultados da Precisão para os testes no DDRJO1..... | 79 |
| Figura 41 – Resultados do Recall para os testes no DDRJO1..... | 80 |
| Figura 42 – Resultados do F1 SCORE para os testes no DDRJO1..... | 81 |
| Figura 43– Comparativo de acurácia entre todos os algoritmos das classes estatístico e detecção de outliers. | 82 |
| Figura 44 – Comparativo de Precisão entre todos os algoritmos das classes estatístico e detecção de outliers. | 82 |

| | |
|--|----|
| Figura 45 – Comparativo de Recall entre todos os algoritmos das classes estatístico e detecção de outliers | 83 |
| Figura 46 – Comparativo de F1 Score entre todos os algoritmos das classes estatístico e detecção de outliers. | 83 |
| Figura 47 – Resultados do tempo para os testes no DDRJO1..... | 88 |
| Figura 48 – Comparativo de tempo entre todos os algoritmos das classes estatístico e detecção de outliers. | 89 |
| Figura 49 – Séries com alta correlação, grupo 0, mesma classe. | 93 |
| Figura 50 – Framework desenvolvido..... | 95 |

Lista de Tabelas

| | |
|---|----|
| Tabela 1 – Análise de Artigos..... | 26 |
| Tabela 2 – Comparativo entre abordagens encontradas em artigos e o nosso trabalho. | 30 |
| Tabela 3 – Descrição das funções de rede existentes no Core 5G. | 40 |
| Tabela 4 – Resumo quantitativo da rede core 5G SA da TIM Br. | 42 |
| Tabela 5 – Exemplo da tradução dos KPIs do NRF..... | 42 |
| Tabela 6 – Resultados da camada 1..... | 57 |
| Tabela 7 – Resultados da camada 2..... | 57 |
| Tabela 8 – Resultados da camada 3..... | 58 |
| Tabela 9 – Métodos utilizados neste trabalho para detecção de anomalia. | 72 |
| Tabela 10 – Resultados obtidos por método e intervalo de treinamento – Ponto Original. | 76 |
| Tabela 11 – Resultados obtidos por método e intervalo de treinamento – Primeiro Evento..... | 77 |
| Tabela 12– Tempos de execução aferidos por algoritmo e intervalo – Ponto original. . | 86 |
| Tabela 13– Tempos de execução aferidos por algoritmo e intervalo –Primeiro evento. | 87 |
| Tabela 14 – Tempos aferidos dos métodos de detecção de anomalia no cenário de 240 horas, tempo total previsto para 524 mil séries e tempo previsto após a saída dos filtros. | 90 |
| Tabela 15 – Séries temporais correlatas e suas classificações de falha, mesma classe.. | 91 |
| Tabela 16 – Séries temporais correlatas e suas classificações de falha, geral..... | 92 |

Lista de Abreviaturas e Siglas

| | |
|----------|---|
| 3GPP | 3rd Generation Partnership Project |
| 5GC | 5G Core |
| 5G NSA | 5G Non-standalone |
| 5G SA | 5G Standalone |
| AF | Application Function |
| AMF | Access Management Function |
| API | Application Programming Interface |
| ARIMA | Autoregressive integrated Moving average |
| AUSF | Authentication Server Function |
| CBLOF | Cluster based Local Outlier Factor |
| CN | Código Nacional |
| COF | Connectivity-Based Outlier factor |
| COPOD | Copula-Based Outlier Detection |
| DSPOT | Drift SPOT |
| EIF | Extended Isolation Forest |
| EWMA | Exponentially Weighted Moving Average |
| EWMA-STR | Exponentially Weighted Moving Average of season-trend |
| IF-LOF | Isolation Forest-Isolation Forest |
| KPI | Key Performance Indicator |
| LAC | Location Area Code |
| LOF | Local Outlier Factor |
| MTLF | Model Training Logical Function |
| NE | Network Element |
| NF | Network Function |
| NFV | Network Function Virtualization |
| NNSF | Network Slice Selection Function |
| NRF | Network Repository Function |
| NWDAF | Network Data Analytics Function |
| PCF | Policy Control Function |
| PCI | Prediction Confidence Interval |

| | |
|-----------------|---|
| pEWMA | Probabilistic Exponentially Weighted Moving Average |
| PMM | Período de Maior Movimento |
| SARIMA | Seasonal Autoregressive Integrated Moving Average |
| RAN | Radio Access Network |
| SBA | Service Based Architecture |
| SDN | Software Define Network |
| SMF | Session Management Function |
| Subsequence IF | Subsequence Isolation Forest |
| Subsequence LOF | Subsequence Local Outlier Factor |
| UDM | Unified Data Management |
| UE | User Equipment |
| UPF | User Plane Function |

Sumário

| | | |
|-------|--|----|
| 1 | Introdução | 16 |
| 1.1 | Redes Core 5G..... | 18 |
| 1.2 | Motivação | 19 |
| 1.3 | Objetivos..... | 20 |
| 1.3.1 | Problema de Pesquisa | 20 |
| 1.3.2 | Hipótese | 20 |
| 1.3.3 | Perguntas de Pesquisa..... | 21 |
| 1.3.4 | Objetivos..... | 22 |
| 1.4 | Organização | 23 |
| 2 | Fundamentos e Trabalhos Relacionados | 25 |
| 2.1 | Zero Touch Network | 25 |
| 2.1.1 | Identificação ou Predição de Falhas | 26 |
| 2.1.2 | Análise de Causa Raiz | 27 |
| 2.1.3 | Recuperação Automática..... | 27 |
| 2.2 | Aplicação de IA para detecção de eventos em redes..... | 28 |
| 2.2.1 | Escolha dos KPIs para análise | 28 |
| 2.2.2 | Algoritmos de Aprendizagem de máquina Supervisionado | 29 |
| 2.2.3 | Analisadores de tráfego em IA | 29 |
| 2.2.4 | Nossa abordagem..... | 30 |
| 2.3 | Network Data Analytics Function | 31 |
| 2.4 | Detecção de anomalia em séries temporais | 33 |
| 2.4.1 | Métodos baseados em Previsão | 34 |
| 2.4.2 | Métodos baseados em Distância..... | 36 |
| 2.4.3 | Métodos baseados em Árvores | 37 |
| 2.4.4 | Métodos baseados em Distribuição | 38 |
| 3 | Caracterização dos KPIs da rede core 5G | 39 |
| 3.1 | Análise quantitativa | 39 |
| 3.2 | Análise Descritiva..... | 43 |
| 3.3 | Mudanças de tendência..... | 47 |
| 3.4 | Considerações Finais | 48 |
| 4 | Filtros..... | 50 |
| 4.1 | Séries temporais estáticas | 51 |
| 4.2 | Séries temporais aleatórias | 52 |
| 4.3 | Séries temporais com alta correlação | 55 |

| | | |
|-------|--|----|
| 4.4 | Resultados..... | 56 |
| 4.5 | Considerações Finais | 59 |
| 5 | Abordagens de aprendizado..... | 61 |
| 5.1 | Testes realizados..... | 62 |
| 5.2 | Resultados..... | 63 |
| 5.3 | Considerações Finais | 70 |
| 6 | Deteção de Anomalias | 72 |
| 6.1 | Experimentos | 73 |
| 6.2 | Resultados..... | 75 |
| 6.2.1 | Desempenho dos modelos | 75 |
| 6.2.2 | Tempo de Execução..... | 85 |
| 6.3 | Considerações Finais | 93 |
| 7 | Conclusões..... | 95 |
| 7.1 | Contribuições..... | 95 |
| 7.2 | Possibilidade de Trabalhos Futuros | 96 |
| | Referências | 98 |

1 Introdução

A quinta geração das redes de telefonia móvel, a rede 5G, traz consigo grandes expectativas para novas aplicações como internet das coisas (do inglês, IoT – Internet of Things), carros inteligentes, cirurgias a distância, dentre várias outras. Isto torna-se possível devido a características de desempenho desta nova arquitetura, dentre elas:

- Taxas de latência inferiores a 1ms, número cerca de 10 vezes inferior aos da rede 4G.
- Taxa de dados de até 10 Gbps, número de 10 a 100 vezes superior as registradas nas redes 4G.
- Disponibilidade prevista superior a 99,99%.
- Redução no consumo de energia dos dispositivos na rede móvel.
- Maior cobertura.
- Maior quantidade de dispositivos conectados em uma mesma região.

Em uma rede de telecomunicações, sua arquitetura pode ser dividida em várias camadas, dentre elas o equipamento do usuário (do inglês, EU – User equipment), RAN (rede de acesso), Edge e Core. O núcleo centraliza todo o tráfego, sendo responsável por funções como autenticação, segurança e gerenciamento de sessão. Impactos nesta camada podem ter consequências catastróficas, desde perda na qualidade do serviço até indisponibilidade total do serviço.

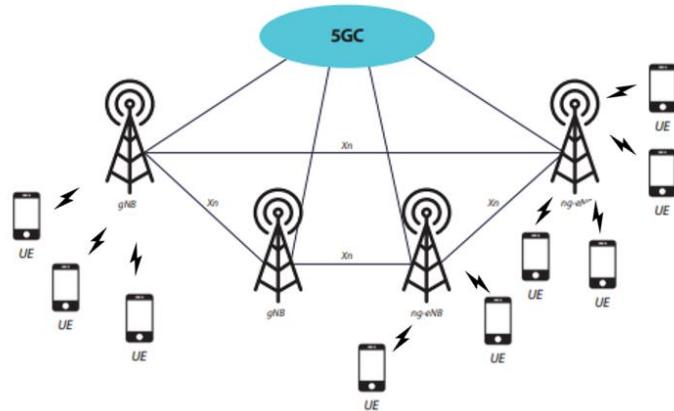


Figura 1 – Conceito de rede 5G.

Considerando que “apenas” o aumento da latência do sistema já seria o suficiente para prejudicar todo o serviço de carros autônomos, por exemplo, falhas não são toleradas. Se ocorrerem, o tempo entre detecção, análise e correção deve ser o menor possível. Tendo isto em vista, monitorar a qualidade do serviço e a saúde de cada equipamento conectado na rede é tarefa de extrema importância. Portanto, todo equipamento desta camada possui um conjunto de indicadores de desempenho (do inglês, Key Performance Indicator - KPI) que determinam o desempenho do sistema. Esses KPIs podem fazer referência a todo o equipamento (whole system), mas também a problemas setorizados, seja de um determinado Código Nacional (CN) ou Código de área de localização (do inglês, LAC - Location Area Code). Esta informação é fornecida através de sistemas à Operação, setor responsável pela manutenção e inclusões de configurações no Core, e é utilizada para a identificação e solução de falhas. No entanto, o processo de monitoração de falhas é incremental, e segue o processo abaixo:

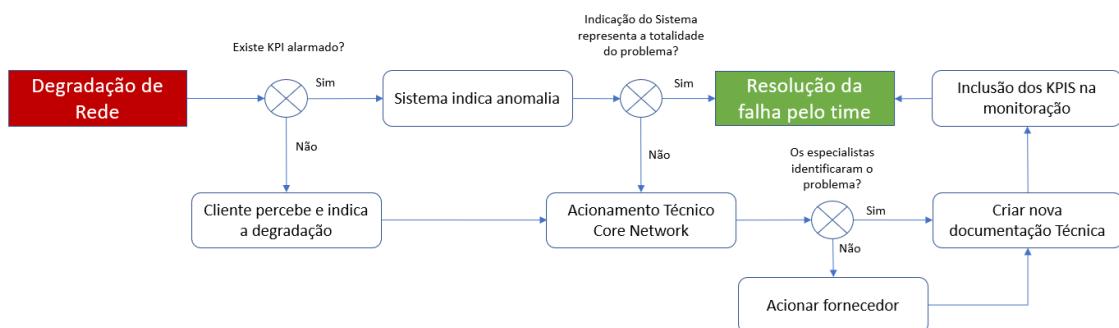


Figura 2 – Processo de identificação de falhas.

Eventos de falha de maior porte na rede core são raros, mas facilmente percebidos. Este tipo de situação gera impactos catastróficos normalmente reportado por portais de notícia e são passíveis de multas pelo órgão regulador, neste caso a Agência Nacional de Telecomunicações (Anatel). Logo, devido ao processo ilustrado na figura 2, indicadores de desempenho que indicam falhas de grande escala normalmente são incluídos na monitoração rapidamente. No entanto, falhas menores, com afetação setorizada, são comuns e muitas vezes não percebidas pela equipe técnica a não ser que exista feedback de clientes impactados. Ou seja, caso uma falha nunca tenha sido identificada anteriormente ou exista um procedimento prévio disponibilizado por um fornecedor, uma falha pode durar horas sem ser identificada pela equipe técnica. Com o passar do tempo,

os times envolvidos desenvolvem expertise e as monitorações se tornam mais seguras, apesar de ainda incompletas. No entanto, a cada mudança de geração o processo se repete.

1.1 Redes Core 5G

Em releases anteriores (2G e 3G), a operação, manutenção e a ampliação das redes dependiam muito da alteração de hardware, seja na definição de novos circuitos ou introduzindo novos elementos para adequação de tráfego. Consequentemente, mudanças eram incomuns ou no mínimo introduzidas lentamente. Um exemplo que pode ser citado é o processo de inclusão de novos elementos na rede. Este demandava semanas, e incluía desde a inserção física do bastidor no prédio industrial até testes exaustivos de desempenho do elemento.



Figura 3 – Tipo de Hardware utilizados em gerações anteriores de telecomunicações.

Esta nova geração de rede adiciona novas tecnologias como Network Function Virtualization (NFV), Software-Defined Networking (SDN) e Service-Based Architecture (SBA). Antes, toda nova implementação era fortemente dependente de hardware e conexões físicas, nesta nova geração tudo é configurável via software. Isso oferece maior flexibilidade, dinamismo e escalabilidade na expansão da infraestrutura, mas também apresenta vários desafios no aspecto de gerenciamento, segurança e manutenibilidade. Este tema é muito bem explorado pelos autores Bello, Hussein, Ulema

e Koilpillai (2022). No exemplo citado acima, o período para a inclusão de um novo elemento de rede é muito menor, pois é completamente virtualizado, ou seja, toda a etapa de instalação física não é mais necessária. Consequentemente, mudanças são mais frequentes, e, com isso, aumenta a probabilidade de inserção de erros, já que a configuração continua sendo realizada manualmente por operadores. Além disso, estas constantes alterações também impactam a monitoração, pois cada mudança pode gerar alterações de tendência no comportamento dos indicadores de desempenho, o que inviabiliza a utilização de regras de acionamento simples, como variação da média ou valores fixos.

1.2 Motivação

Uma alternativa a monitoração é a manutenção preventiva ou preditiva, que é focada em evitar que um problema venha a ocorrer. Porém, mesmo com esta abordagem implementada, existem situações difíceis de evitar, como o rompimento de uma fibra ótica ou até mesmo a execução de uma configuração inadequada na rede. Consequentemente, acompanhar a qualidade do serviço continua sendo vital para a operação. No entanto, monitorar a rede tornou-se uma tarefa mais complexa nesta nova geração, tanto devido as rápidas alterações que ocorrem na rede, gerando assim mudanças no perfil dos KPIs e consequentemente em métricas utilizadas na operação, quanto devido ao tipo de serviço planejado para utilização da infraestrutura, que não permite tempos grandes na identificação de falhas. Mais do que nunca, o processo de identificação de falhas deve ser preciso e rápido. Quando se fala em processo rápido, refere-se a capacidade de identificar o primeiro evento de falha e não necessariamente todo o subconjunto anômalo.

Uma solução simples para a monitoração seria acompanhar todos os KPIs disponibilizados pelos equipamentos, garantindo assim qualidade no serviço fornecido. No entanto esta abordagem tem alguns impeditivos. Primeiro, a quantidade de séries temporais geradas a cada período de medição ultrapassa a centena de milhares, o que inviabiliza a análise humana. Mesmo do ponto de vista computacional é um processo custoso. Outro desafio seria gerar regras de acionamento para cenários de falha genéricos, já que cada KPI tem um comportamento específico, sendo alguns constantes, outros variantes conforme o tráfego existente. A abordagem mais comum são regras baseadas

em médias, mas que em muitas situações não são o suficiente, pois existem KPI com perfil randômico, por exemplo, que acabariam gerando muitos alarmes falsos.

Na literatura acadêmica encontram-se vários artigos voltados para a identificação de eventos disruptivos na rede 5G utilizando inteligência artificial. No entanto, todos os artigos baseiam-se em emulação, não em testes feitos com dados reais de uma rede comercial e complexa em produção. Além disso, nos testes apresentados, normalmente são utilizados poucos KPIs, como a latência, que embora seja uma abordagem mais simples, não é capaz de classificar o problema em si. Uma abordagem comumente utilizada é a de métodos supervisionados, que podem ter grande assertividade, mas não são aplicados porque o conjunto de dados históricos não contém marcação de falhas anteriores.

Dado o cenário acima descrito, esta pesquisa tem como objetivo identificar o primeiro evento anômalo nos KPIs disponíveis em uma rede core comercial real com assertividade e tempo de inferência aceitável, adaptável as mudanças da rede e com uma abordagem caixa preta.

1.3 Objetivos

1.3.1 Problema de Pesquisa

É possível implementar um detector de anomalias, capaz de indicar o primeiro evento com alta assertividade e tempo de inferência aceitável, analisando todos os KPIs disponíveis, adaptável a mudanças de características da rede por meio de uma abordagem caixa-preta?

1.3.2 Hipótese

Uma abordagem centrada em dados é capaz de realizar a detecção de anomalia do primeiro evento com tempo de inferência aceitável.

1.3.3 Perguntas de Pesquisa

Para orientar esta dissertação e os passos para validar nossa hipótese, foram formuladas quatro perguntas de pesquisa, que serão exploradas nos capítulos subsequentes para alcançar nossos objetivos.

- **Pergunta de pesquisa 1:** *Qual é a caracterização dos KPIs de uma rede core 5G de uma empresa de telecomunicações?*

A primeira pergunta é sobre o comportamento dos KPIs e envolve entender a periodicidade da coleta de informação, quantos indicadores estão disponíveis assim como os valores quantitativos, se existe ou não sazonalidade nos dados, se o mesmo indicador em equipamentos diferentes tem ou não comportamentos análogos, se existem mudanças no comportamento estacionário dos dados por consequência de mudanças de rede. Esta análise é relevante para determinar se é possível diminuir a dimensão dos dados, se o modelo de treinamento pode ser apenas indutivo, assim como ajuda a determinar modelos de detecção de anomalia.

- **Pergunta de pesquisa 2:** *É possível isolar os KPIs que representam mudanças no estado da rede com o maior ganho de informação?*

Com base no entendimento do comportamento dos indicadores, esta pergunta de pesquisa visa entender quais as abordagens e métodos podem ser aplicados ao conjunto de dados de forma a diminuir a quantidade de informação a ser analisada. Considerando as mudanças constantes de configuração, é necessário também entender que esta etapa deve ser realizada dinamicamente, ou seja, também deve ser feita constantemente.

- **Pergunta de pesquisa 3:** *Qual a melhor abordagem de aprendizado para o problema proposto? Transdutivo ou indutivo?*

Considerando que o comportamento dos KPIs pode mudar com alterações na rede, treinar um modelo uma única vez e esperar bons resultados de assertividade em um curto período talvez não seja viável. Esta pergunta visa entender qual a melhor abordagem de treinamento para o problema proposto: puramente indutivo, ou seja, o treinamento é realizado apenas uma vez, híbrida com um período de retreinamento a

ser definido ou puramente transdutivo, ou seja, o treinamento do modelo deve ser refeito a cada previsão.

- **Pergunta de pesquisa 4:** *É possível implementar modelos de detecção de anomalia para o primeiro evento com alta assertividade e com tempo de inferência aceitável?*

Considerando todos os KPIs de rede e suas características, esta pergunta visa entender quais são os melhores métodos de detecção de anomalia não supervisionados para o problema proposto, levando em consideração não apenas a assertividade do modelo mas também o tempo de inferência, que não pode ser superior a periodicidade de atualização dos KPIs.

1.3.4 Objetivos

O objetivo geral deste trabalho é desenvolver um framework capaz de detectar anomalias para redes core 5G SA em tempo de inferência aceitável. Para alcançar os nossos objetivos utilizamos nesta pesquisa os dados da rede core reais disponibilizados pela operadora TIM Brasil, uma das 3 maiores empresas de telecomunicações operando em solo nacional. Os objetivos específicos são:

- **Objetivo específico 1:** *Fazer a caracterização da rede core 5G da TIM Brasil.*
Na literatura é comum a utilização de dados emulados para realizar estudos sobre a detecção de anomalia em redes. No entanto, escalar muitas das soluções propostas é inviável quando se trata de uma rede comercial complexa. Além disso, ao analisar todos os KPIs disponíveis encontra-se mais de um padrão de comportamento, sendo possível de mudança com o tempo. Nesta etapa o objetivo é definir, já com dados reais, as características de uma rede core 5G.
- **Objetivo específico 2:** *Implementar um filtro adaptativo capaz de excluir KPIs de baixo impacto.* Após a caracterização dos indicadores de desempenho da rede core, implementaremos um filtro dinâmico com base em três camadas. Este filtro visa retirar KPIs que são quase estáticos, não estáveis, e que possuem forte correlação com outros indicadores já analisados. O objetivo dessa etapa é reduzir a quantidade de indicadores a serem analisados, otimizando recursos

computacionais e diminuindo o tempo necessário para a aplicação de modelos de detecção de anomalia.

- **Objetivo específico 3:** *Avaliar o desempenho de modelos de aprendizado transdutivo e indutivo.* Os KPIs da rede Core 5G são mais susceptíveis a mudanças de comportamento que gerações anteriores. Consequentemente é necessário entender se um modelo de treinamento indutivo teria bons resultados no problema proposto, assim como um modelo transdutivo. Nesta etapa o objetivo é apresentar comparações entre as duas abordagens baseado em testes realizados com o conjunto de dados real.
- **Objetivo específico 4:** *Avaliar métodos estatísticos e de detecção de outliers no contexto da aplicação.* Para realizar uma avaliação eficaz dos modelos de detecção de anomalias, testaremos uma falha reportada pela operadora TIM BR em dois momentos diferente. Algoritmos estatísticos como ARIMA e SARIMA serão testados, juntamente com modelos de detecção de anomalias como LOF (Local Outlier Factor) e Isolation Forest. Esta etapa visa determinar a eficácia desses métodos em identificar e classificar anomalias em condições variáveis.

1.4 Organização

As seções subsequentes desta dissertação são organizadas da seguinte forma:

- Capítulo 2 apresenta os trabalhos correlacionados assim como o conceito de Zero Touch Network. Além disso, também serão apresentados gaps identificados em artigos quanto a aplicação de algumas abordagens no contexto de redes comerciais reais. Por fim é apresentado o embasamento teórico utilizado para a detecção de anomalias assim como todos uma breve descrição dos métodos utilizados neste trabalho.
- Capítulo 3 traz as caracterizações da rede, tanto do ponto de vista quantitativo como descritivo, dando assim dimensão do problema a ser tratado. Este capítulo responde à pergunta de pesquisa 1.

- Capítulo 4 aborda a implementação de um filtro dinâmico desenvolvido para diminuir a quantidade de indicadores a serem analisados, ganhando assim eficiência computacional. Este capítulo responde à pergunta de pesquisa 2.
- Capítulo 5 apresenta os testes realizados com modelo de treinamento indutivo, híbridos e totalmente transdutivo, respondendo assim à pergunta de pesquisa 3.
- Capítulo 6 aborda todos os testes realizados para a detecção de anomalias, assim como resultados tanto de tempo quanto de assertividade, comparando assim todos os métodos utilizados. Este capítulo responde à pergunta de pesquisa 4.
- Capítulo 7 apresenta as conclusões desta dissertação, assim como sugestões de trabalhos futuros.

2 Fundamentos e Trabalhos Relacionados

2.1 Zero Touch Network

O Zero Touch Network é um conceito muito desejado pelas operadoras de telecomunicações e consequentemente muitas pesquisas têm se concentrado em alguma etapa do seu processo. Trata-se de um sistema de malha fechada na qual a rede opera sem qualquer interferência humana, baseada em três fases: Identificação ou predição de falhas, identificação da causa raiz do problema e recuperação automática da rede. Este processo é descrito na imagem abaixo.

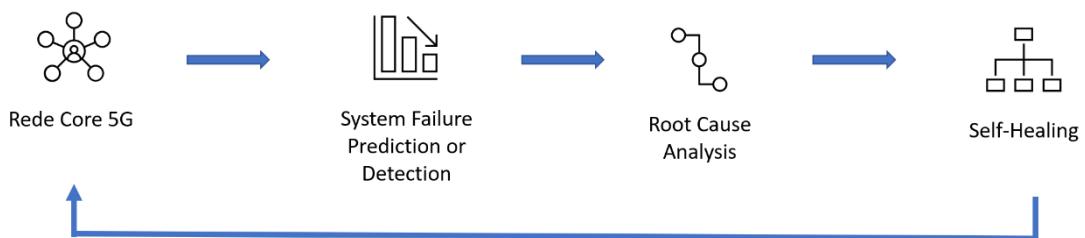


Figura 4 – Conceito de Zero Touch Network.

Este modelo, apesar de muito desejado, ainda está longe de ser implementado, pois possui diversos limitantes. Os autores Benzaid e Taleb (2020) mencionam que os investimentos previstos na área podem chegar a \$7.4 bilhões até 2025. Estes trazem um debate muito interessante sobre o conceito de zero touch que será apresentado a seguir.

Do ponto de vista de artigos publicados na área, a grande maioria concentra-se na primeira etapa, ou seja, detecção ou predição de falhas. No entanto, nenhuma publicação encontrada utiliza dados de uma rede comercial complexa real, problema que será tratado no escopo desta dissertação. Isto é ilustrado na Tabela 1.

Tabela 1 – Análise de Artigos.

| Artigo | Testes | | Zero Touch Network | | |
|--|--------------|-----------|--------------------------------------|------------|--------------|
| | Rede emulada | Rede real | System Fault Detection or Prediction | Root Cause | Self Healing |
| Zhou, S.; Liu, X.; Fu, D.; Cheng, X.; Fu, B.; Zhao, Z (2020) | | | X | X | X |
| Chakraborty, P., Corici, M., & Magedanz, T (2020) | X | | X | | |
| Jeon, Y.; Jeong, H.; Seo, S.; Kim, T.; Ko, H.; Pack, S (2022) | | | X | | |
| Manias, D. M.; Chouman, A.; Shami, A (2022) | X | | X | | |
| Benzaid, C.; Taleb T (2020) | | | X | X | X |
| Ferreira D.; Senna C.; Sargent S (2020) | X | | X | | |
| Terra A.; Inam R.; Baskaran S.; Batista P.; Burdick I.; Fersman E (2020) | X | | | X | |
| Reshma, T. R.; Azath, M (2021) | X | | X | X | X |
| Panay, S.; Latkoski, P (2020) | X | | X | X | X |
| Chakraborty P.; Corici M.; Magedanz T (2021) | X | | X | | |
| Nosso trabalho | | X | X | | |

2.1.1 Identificação ou Predição de Falhas

A primeira etapa do processo de implementação de um Zero Touch Network envolve a predição ou detecção de um evento anômalo. Por predição subentende-se detectar que o evento pode ocorrer (probabilidade) dentro de um período determinado. Para o correto funcionamento deste modelo, os dados dos equipamentos ou indicadores de serviço (latência, por exemplo) devem ser interpretados em tempo real, e com base em diversos tipos de algoritmos e métricas determinar se está ocorrendo ou irá ocorrer um evento disruptivo.

Uma implementação eficaz deste modelo traria benefícios significativos tanto para os clientes quanto para as empresas de telecomunicações. Para os usuários, o ganho seria a melhoria da qualidade do serviço prestado devido a diminuição do tempo de indisponibilidade da rede no pior cenário (não é possível prever que um evento anômalo irá ocorrer, logo o foco é identificar o seu andamento o mais rápido possível). Para as operadoras de telefonia isso significa uma maior eficiência operacional, redução de custos e melhoria na imagem delas junto ao público.

Na literatura é comum encontrar artigos que utilizam algoritmos de aprendizagem supervisionados como metodologia de solução, normalmente utilizando poucos KPIs para análise e testes em redes emuladas. Também é possível encontrar exemplos utilizando modelagem em séries temporais. Exemplos de autores que abordam o tema são Chakraborty, Corici e Magedanz (2021) e Ferreira, Senna e Sargent (2020).

2.1.2 Análise de Causa Raiz

A segunda etapa do processo de Zero Touch Network é a análise da causa do problema. Trata-se de identificar o motivo pelo qual o evento disruptivo irá ocorrer ou está ocorrendo. Esta etapa é mais complexa que a anterior, já que é dependente de mais informações como registro de logs, de erros e de configurações, correlação com outras camadas da rede e registros anteriores de eventos que ocorreram. Em comparação com a identificação de falhas, é possível analisar a qualidade do serviço por apenas um indicador de qualidade, como latência medida a partir do terminal do usuário, mas não é impraticável identificar a causa raiz de um problema com tão pouca informação.

Ser capaz de identificar o motivo pelo qual um problema vai ou está acontecendo traz inúmeros ganhos, como maior eficiência operacional, diminuição da necessidade de conhecimento e maturidade de especialistas em rede Core, além de uma necessidade menor de acionar fornecedores em busca de apoio técnico. Para os clientes, isso se traduz em uma melhoria na qualidade e na confiabilidade da rede.

As pesquisas focadas nesta etapa costumam utilizar abordagens como explicabilidade em aprendizagem de máquina e redes bayesianas para fornecer informações mais precisas e interpretações comprehensíveis dos dados. Exemplos de autores que exploram diferentes abordagens e tecnologias para aprimorar a eficácia da análise de causa raiz são Terra, Inam, Baskaran, Batista, Burdick e Fersman (2020).

2.1.3 Recuperação Automática

Por fim, a última etapa do processo de Zero Touch Network é a recuperação automática da rede, que se baseia na causa raiz identificada para determinar a maneira mais eficaz de recuperar a rede. Este processo é facilitado pela arquitetura virtualizada do 5G, que permite uma maior flexibilidade e rapidez na resposta a incidentes.

Além disso, como trata-se de uma eventual configuração automática, a probabilidade de falhas humanas no processo é minimizada.

Exemplos de trabalhos que exploram essa metodologia incluem os estudos de Shiyu, Xiqing, Dengsheng, Xinzhou, Bin e Zhenqiao (2020), que demonstram como essas técnicas podem ser efetivamente aplicadas para melhorar a resiliência e a eficiência operacional das redes de telecomunicações.

2.2 Aplicação de IA para detecção de eventos em redes

A maioria dos artigos que abordam conceitos de Zero Touch em redes core 5G ou a utilização de IA em segurança em redes, baseiam-se em arquiteturas simples, normalmente emuladas, com poucos equipamentos e com a análise de poucos KPIs. Além disso, a maioria utiliza algoritmos supervisionados para detectar eventos anômalos enquanto outros utilizam dados extraídos diretamente da rede em operação. O problema destas abordagens são que a escalabilidade das soluções e modelos propostos não são simples. Nesta seção serão abordados desafios na utilização destes algoritmos e abordagens em uma rede core 5G real.

2.2.1 Escolha dos KPIs para análise

Cada equipamento de rede produz uma grande quantidade de indicadores de desempenho. Para exemplificar, um SMF tem mais de 800 indicadores de desempenho. Considerando que uma rede comercial complexa possui centenas de equipamentos, a quantidade de dados a serem analisados é grande. Além disso, a maioria destes KPIs representam informações de subconjuntos da rede, ou seja, são subdivididos em múltiplas séries temporais, o que torna este volume ainda maior (tema que será tratado no capítulo 3). No entanto, é comum na literatura modelos de previsões em redes core 5G que utilizam poucos KPIs, não dando o motivo desta escolha. Um exemplo de autores que fazem esta abordagem são Chakraborty, Corici e Magedanz (2021).

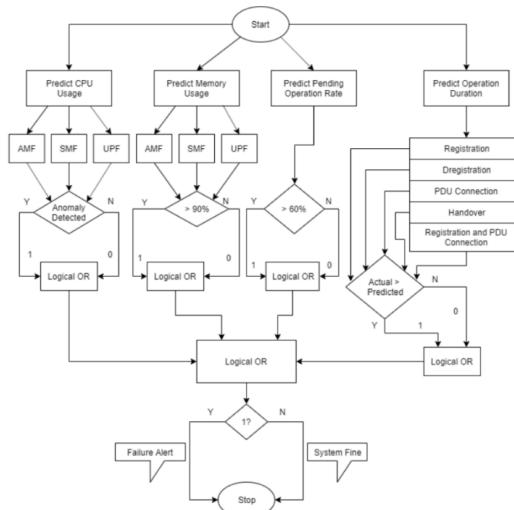


Figura 5 – Exemplo de solução proposta que utiliza poucos KPIs (Chakraborty, Corici e Magedanz 2021).

Ainda neste mesmo tópico, foca-se muito na assertividade do algoritmo utilizado, o que de fato é importante. Porém, considerando uma rede real, outras métrica são fundamentais como o tempo necessário para o treinamento e identificação da falha, considerando que se trata de um problema de big data. Até é possível a identificação de eventos disruptivos considerando indicadores de serviço, que são perceptíveis ao usuário final. Por exemplo, a medição da latência pode ser utilizada como um indicador de possível falha. No entanto, esta abordagem é ruim quando é necessário encontrar a causa raiz do problema, que necessita analisar a rede em cada um dos seus elementos, ou até mesmo identificar falhas menores, setorizadas em um bairro ou região.

2.2.2 Algoritmos de Aprendizagem de máquina Supervisionado

As redes 5G ainda estão sendo implementadas ao redor do mundo, sendo que a 5G SA existe em operação em poucos países até o presente momento. Além da falta de conhecimento completo quanto a operação destas redes, em muitas operadoras não existe um processo que mapeie falhas anteriores, criando assim um conjunto de dados classificados. Mesmo com este cenário é comum ver em artigos propostas de soluções que utilizam aprendizagem supervisionada.

Outro ponto relevante é que novos tipos de falhas e problemas são depurados constantemente. Com isso em vista, abordagens não supervisionadas tendem a detectar uma maior quantidade de problema a longo prazo.

2.2.3 Analisadores de tráfego em IA

Um ponto importante a ser observado são abordagens que utilizam analisadores de tráfego para tomadas de decisão como os autores Zhou, Liu, Fu, Cheng, Fu e Zhao (2020).

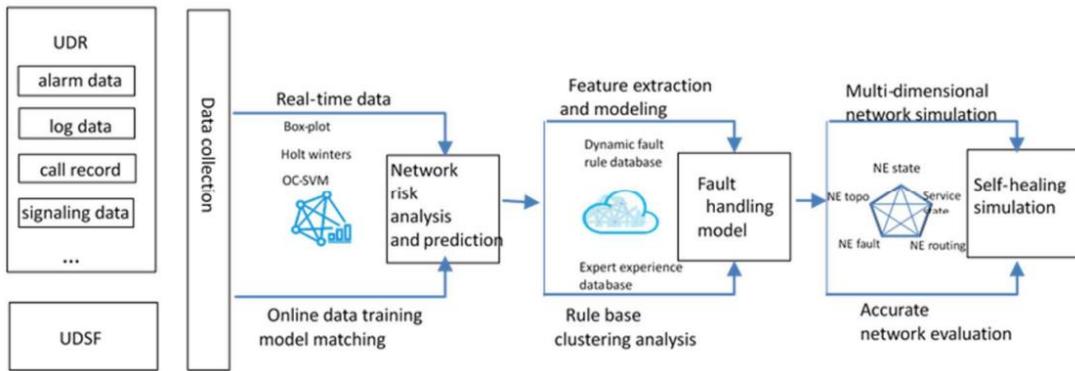


Figura 6 – Exemplo de framework que utiliza dados coletados em tempo real (Zhou, Liu, Fu, Cheng, Fu e Zhao 2020)

Analizar os pacotes trafegados na rede garante que toda a informação será analisada, o que parece uma abordagem interessante, considerando, por exemplo, a possibilidade de analisar comunicação e protocolos de rede. Esta abordagem é ainda mais interessante quando o foco são aspectos de segurança. No entanto, para redes core 5G reais, o volume de tráfego é gigantesco, o que torna esta abordagem inviável tanto do ponto de vista do esforço computacional quanto de tempo para análise de toda a informação em tempo real. Além disso, outra preocupação seria com a proteção dos dados pessoais, já que esta abordagem daria acesso a informações sensíveis, tais como quais foram chamadas realizadas por cada usuário, assim como o tempo de duração delas, localização do número de a, entre outras.

2.2.4 Nossa abordagem

Devido ao que foi apresentado nas seções anteriores, decidimos utilizar abordagens diferentes, sendo essas utilizar todos os indicadores de desempenho disponíveis na rede, baseados em contadores de desempenho summarizados, além de utilizar apenas algoritmos de inteligência artificial não supervisionados. A diferença entre a nossa abordagem e demais trabalhos está summarizado na Tabela 2,

Tabela 2 – Comparativo entre abordagens encontradas em artigos e o nosso trabalho.

| Artigos | Nosso trabalho |
|---------------------------------------|--|
| Poucos indicadores de desempenho | Utiliza todos os indicadores de desempenho disponíveis na rede |
| Utilizam aprendizado supervisionado | Utiliza aprendizado não supervisionado |
| Baseiam-se em analisadores de tráfego | Baseia-se nos contadores de desempenho summarizados |

Como nosso conjunto de dados, assim como a abordagem é muito diferente da encontrada na literatura, nesta dissertação não faremos comparações com resultados de outros trabalhos.

2.3 Network Data Analytics Function

A função de rede Network Data Analytics Function (NWDAF) faz parte da arquitetura da rede core 5G especificada no 3GPP TS 23.501. Foi introduzida pela primeira vez no release 15 para fornecer automaticamente análises específicas de dados de rede. Desde então vem sendo aprimorado para receber novas funcionalidades. Os autores Niu, Zhao, She e Chen (2022) abordam toda a evolução do NWDAF desde o release 15 até o 18.

A arquitetura 5G permite ao NWDAF coletar dados de qualquer elemento da rede através de requisições, no modelo de API. Este processo está descrito na imagem abaixo:



Figura 7 – Comunicação entre NE e NWDAF para coleta de dados.

Este processo centraliza os dados em um repositório único, o que simplifica o desenvolvimento de algoritmos de inteligência artificial. No release atual, o NWDAF é capaz de fazer análise de dados e retornar esta informação para os elementos de rede, possibilitando assim uma série de automações. Este processo está descrito na Figura 8.

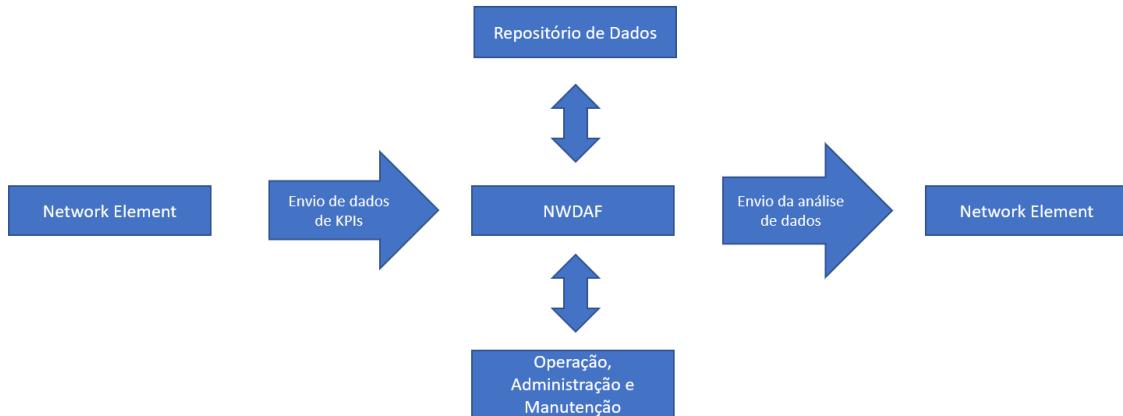


Figura 8 – Framework Genérico para automações na rede 5G.

É possível encontrar na literatura acadêmica uma série de estudos utilizando o NWDAF para arquiteturas baseadas em inteligência artificial. Dentre eles Chouman, Manias e Shami (2022) e Kim, Kim, Ko, Seo, Jcon e Jeong (2022). Muitos são os dados gerados pelo core 5G, logo coletá-los de maneira centralizada pode consumir uma grande quantidade de recursos além de poder causar falhas de segurança. Este tema é abordado por Jeon, Jeong, Seo, Kim, Ko e Pack (2022). Para mitigar esses problemas, o 3GPP, no release 17, introduziu a possibilidade de introduzir um sistema distribuído de NWDAFs. Esta abordagem é tratada na Figura 9, que demonstra como a descentralização do NWDAF pode aliviar a carga em infraestruturas centralizadas e contribuir para uma rede mais resiliente e adaptável.

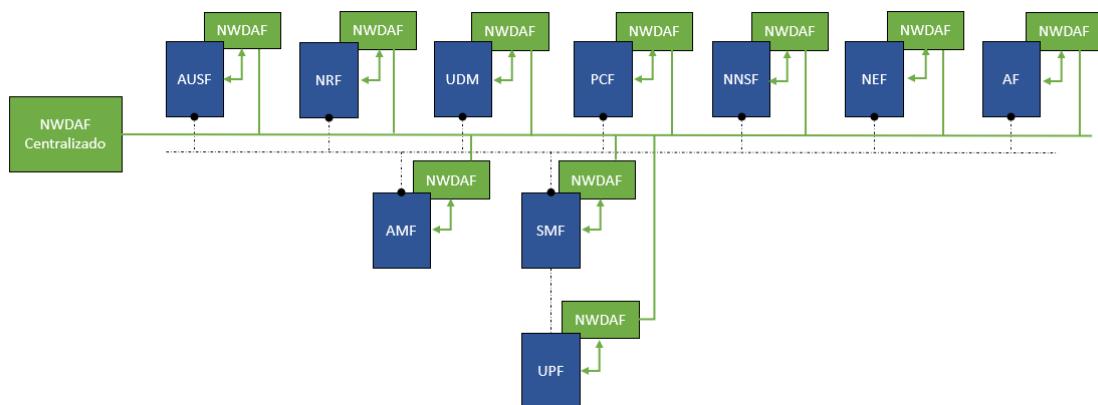


Figura 9 – Conceito de Arquitetura utilizando NWDAF distribuído.

Esta abordagem permite outros tipos de algoritmo a serem implementado, como o de aprendizado federado. Apesar de muito interessante, esta função de rede ainda não foi implementada na rede core da TIM Brasil, logo nos impossibilitando de testar este tipo de abordagem. Além disso, a utilização de aprendizado federado exigiria a execução de comandos em uma rede ativa, não se limitando apenas ao acesso aos contadores disponibilizados pelos equipamentos. Isso adiciona uma camada de complexidade, dificultando o acesso e a manipulação dos dados necessários para a pesquisa.

2.4 Detecção de anomalia em séries temporais

O tema detecção de anomalia em séries temporais é recorrente na literatura pois muitos são suas áreas de aplicação, como em processos de fabricação, segurança cibernética, mercado financeiro e cuidados com a saúde. Anomalias podem indicar problemas de segurança (invasão de um sistema), a tendência de um paciente ter ou não uma determinada doença através de medições cardíacas ou mesmo que uma falha pode ocorrer no futuro em um determinado equipamento monitorado dentro de uma fábrica. Estas séries podem ter comportamento mais simples, o que torna sua análise mais simples, ou ter comportamentos mais complexos, o que requer métodos mais sofisticados para sua análise. Por este motivo muitos são os algoritmos que vêm sendo desenvolvidos para detectar com precisão anomalias.

O principal artigo que utilizamos como embasamento para este tema é dos autores Schmidl, Wenig e Papenbrock (2022), que apresenta 158 métodos para detecção de anomalias, sendo estes de diferentes áreas, sendo estas: Aprendizagem de máquina Clássico, Análise de Sinais, Estatísticos, Data Mining, Deep Learning e Detecção de Outliers. Além disso, aborda diferentes famílias de algoritmos, como aqueles baseados em distância, previsão ou árvores. Nesse artigo é realizada uma análise comparativa em 71 algoritmos utilizando 976 conjuntos de dados distintos. A maioria dos códigos utilizados pelos autores utiliza bibliotecas bem conhecidas, tendo algumas exceções. Estas implementações encontram-se em um repositório no Git Hub e foram a base para o desenvolvimento deste trabalho.

Quanto a classificação dos algoritmos, utilizaremos a nomenclatura presente no artigo.

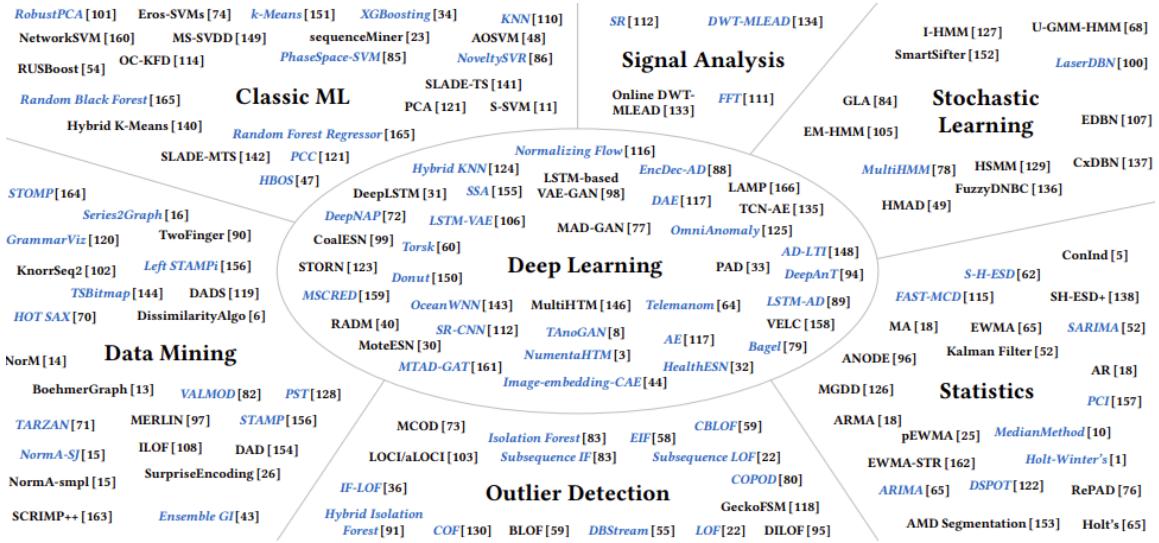


Figura 10 – Métodos de detecção de anomalia em séries temporais (Schmidl, Wenig e Papenbrock 2022).

Durante as etapas iniciais do nosso trabalho consideramos utilizar todos os métodos não supervisionados abordados neste artigo, o que não foi possível devido a limitação temporal para a conclusão do mestrado, pois não se trata apenas de executar os algoritmos, mas sim de adequá-los ao nosso conjunto de dados e extrair o melhor resultado possível. Por fim decidimos focar em duas áreas, sendo elas Detecção de Outliers e Estatísticos. A escolha deu-se devido a alguns motivos como aderência a nossa pesquisa, no caso da área de detecção de outliers, e a simplicidade da implementação, no caso dos métodos estatísticos.

Um breve resumo sobre os métodos de detecção de anomalia utilizados neste trabalho sera apresentado a seguir. Todas as referências para os artigos originais, que contém os detalhes matemáticos e lógicos, encontram-se na seção Referências.

2.4.1 Métodos baseados em Previsão

Os métodos baseados em previsão utilizam um modelo aprendido com base em intervalos anteriores para prever quais os valores da janela atual. Os pontos previstos são então comparados com os valores reais da série temporal através de um threshold para determinar o quanto anômalos são os valores reais. Os métodos desta família diferem mais no tipo de modelo de previsão que utilizam, na forma como constroem esse modelo e na métrica de cálculo para as pontuações de anomalia.

- ***Autoregressive integrated Moving average (ARIMA)***: Foi proposto em 1970 por George Box e Gwilym Jenkins como uma tentativa de descrever mudanças em séries temporais utilizando uma abordagem matemática. É um método estatístico clássico, composto por três partes: AR (p = auto regressão), I (d = grau de diferenciação) e MA (q = Média Móvel). Este modelo é uma generalização do modelo ARMA.
- ***Seasonal Autoregressive integrated Moving average (SARIMA)***: Foi proposto por George Box e Jenkins em 1976. O modelo ARIMA não é indicado para séries com sazonalidade, algo que foi trabalhado no modelo SARIMA. Neste modelo inclui-se mais um parâmetro, m, que indica a sazonalidade.
- ***Exponentially Weighted Moving Average (EWMA)***: Este modelo foi proposto por Roberts em 1956. O conceito de usar uma média móvel foi projetado para dar mais peso aos pontos de dados mais recentes e menos aos mais antigos. Os pesos diminuem exponencialmente à medida que os pontos dos dados envelhecem.
- ***Probabilistic Exponentially Weighted Moving Average (pEWMA)***: Este modelo foi proposto por Carter e Streilein (2012). O conceito deste é ajustar a parametrização existente no EWMA baseado na probabilidade da observação dada.
- ***Exponentially Weighted Moving Average of season-trend model Residuals (EWMA-STR)***: Modelo proposto em 2016 por Zeng-Guang Zhou e Ping Tang. Este modelo é baseado no EWMA e utiliza como parâmetros partes da decomposição das séries temporais, mais especificamente a sazonalidade e a tendência.
- ***Holt-Winters***: Trata-se da extensão do algoritmo Holt's para capturar sazonalidade. Foi proposto por Winter em 1960. Este modelo incorpora tanto a tendência quanto o efeito sazonal.
- ***Prediction Confidence Interval (PCI)***: Método apresentado por Yu, Zhu, Li e Wan (2014). O método usa intervalo de confiança de predição (PCI) como limite

em consideração à incerteza nos parâmetros da série de dados no modelo de previsão. Os dados são classificados como anômalos/não anômalos com base no fato de estarem ou não fora de um determinado PCI.

2.4.2 Métodos baseados em Distância

Os métodos baseados em distância utilizam métricas de distância para comparar pontos ou subsequências de uma série temporal. Pontos disruptivos devem ter distâncias maiores para outros pontos com comportamento normal. Para os cálculos de distância, os algoritmos desta família podem tomar todas as outras subsequências, apenas alguns vizinhos mais próximos ou certos centróides de cluster como pontos de referência. O objetivo dos métodos pode ser classificar apenas um ou todo o cluster de pontos. Estes algoritmos baseados em distância geralmente não requerem dados de treinamento e, portanto, não são supervisionados.

- ***Local Outlier Factor (LOF)***: Algoritmo proposto por Breunig, Kriegel, Raymond e Sander (2000) para encontrar pontos de dados anômalos medindo o desvio local de um determinado ponto de dados em relação aos seus vizinhos. O LOF utiliza conceitos como distância para o centro para estimar a densidade local, consequentemente indicando como pontos anômalos aqueles distantes deste centro.
- ***Cluster Based Local Outlier Factor (CBLOF)***: Este algoritmo foi proposto He, Xu, and Deng em 2002. Enquanto o LOF está preocupado em identificar um único evento anômalo, o CBLOF utiliza a mesma lógica de cálculo de distâncias até o centro para classificar toda a região de pontos próximas como anômalo.
- ***Connectivity-Based Outlier Factor (COF)***: Algoritmo proposto por Tang e Chen (2002). É um método baseado na densidade local para lidar com valores discrepantes que se desviam dos padrões de densidade esférica. É uma versão melhorada do LOF. A ideia do algoritmo é atribuir um nível de outlier para cada ponto. Este nível é chamado de COF do ponto. Quanto maior este ponto maior a probabilidade de o ponto ser anômalo.

- ***Subsequence Local Outlier Factor (subsequence LOF)***: Algoritmo proposto por Schmidl, Wenig e Papenbrock (2022). Ao invés de aplicar o LOF em todo o conjunto de dados, a série temporal é subdividida em tamanhos fixos e o algoritmo então é aplicado nesta subsequência.

2.4.3 Métodos baseados em Árvores

Os métodos baseados em árvores constroem um conjunto de árvores aleatórias, podendo ser binárias ou não, particionando as amostras da série temporal randomicamente buscando isolar pontos anômalos. Como as amostras anômalas são mais fáceis de separar do que as amostras normais, elas estão, em média, mais próximas da raiz da árvore e têm caminhos visivelmente mais curtos.

- ***Isolation Forest (IF)***: O algoritmo Isolation Forest foi inicialmente proposto por Liu, Ting e Zhou (2008). Este algoritmo baseia-se no fato que eventos anômalos costumam ser poucos e diferentes, buscando assim anormalidades. Seu funcionamento é baseado em árvores binárias, sendo que a série vai sendo dividida randomicamente buscando-se isolar o ponto disruptivo.
- ***Extended Isolation Forest (EIF)***: Algoritmo proposto por Hariri, Kind e Brunner (2019). A diferença entre o IF e o EIF está principalmente na maneira com que os dados são subdivididos.
- ***Subsequence Isolation Forest (subsequence IF)***: Algoritmo proposto por Schmidl, Wenig e Papenbrock (2022). Ao invés de aplicar o IF em todo o conjunto de dados, a série temporal é subdividida em tamanhos fixos e o algoritmo então é aplicado nesta subsequência.
- ***Isolation Forest-Isolation Forest (IF-LOF)***: Algoritmo proposto por Cheng, Zou e Dong (2019). O algoritmo LOF tem alta complexidade, pois precisa calcular todas as distâncias entre os pontos, ou seja, $O(n^2)$. Este método propõe primeiro aplicar o algoritmo Isolation Forest para diminuir a quantidade de pontos a serem analisados, e em sequência aplicar o LOF.

2.4.4 Métodos baseados em Distribuição

Os métodos de distribuição estimam a distribuição dos dados ou ajustam um modelo de distribuição aos dados. As distribuições são calculadas sobre pontos de dados ou subsequências obtidas por meio de janelas. Embora a similaridade de pontos e subsequências possa ser um fator para o ajuste da distribuição, a anormalidade é julgada pela frequência e não pela distância nesta família de algoritmos. Em geral, esta é uma abordagem não supervisionada, pois anomalias podem ser encontradas nos extremos das distribuições.

- ***Copula-Based Outlier Detection* (COPOD).** Algoritmo proposto por Li, Zhao, Botta, Ionescu e Hu (2020). Em estatística copula é uma função utilizada para formular distribuições multivariadas, representando dependências entre os dados. Este método é inspirado por copulas, construindo inicialmente uma copula empírica, então utiliza a mesma para prever as probabilidades de cada dado para determinar o seu nível de extremidade.
- ***Drift SPOT (DSPOT)*:** Proposto por Siffer Fouque, Termiere e Largouët (2017). Este algoritmo considera que a distribuição dos dados pode mudar com o tempo, logo não necessita de uma série temporal estacionaria para obter bons resultados.

3 Caracterização dos KPIs da rede core 5G

3.1 Análise quantitativa

Apesar de ser muitas vezes vista como uma caixa preta, a rede core é composta por múltiplos tipos de equipamento, sendo cada um destes responsável por uma função. Existem equipamentos voltados para o gerenciamento do tráfego, outros para o armazenamento de informações de clientes, assim como aqueles voltados para a disponibilidade dos dados enquanto alguns são voltados para chamadas de voz. A Figura 11 ilustra a arquitetura padrão adotada pelo 3GPP:

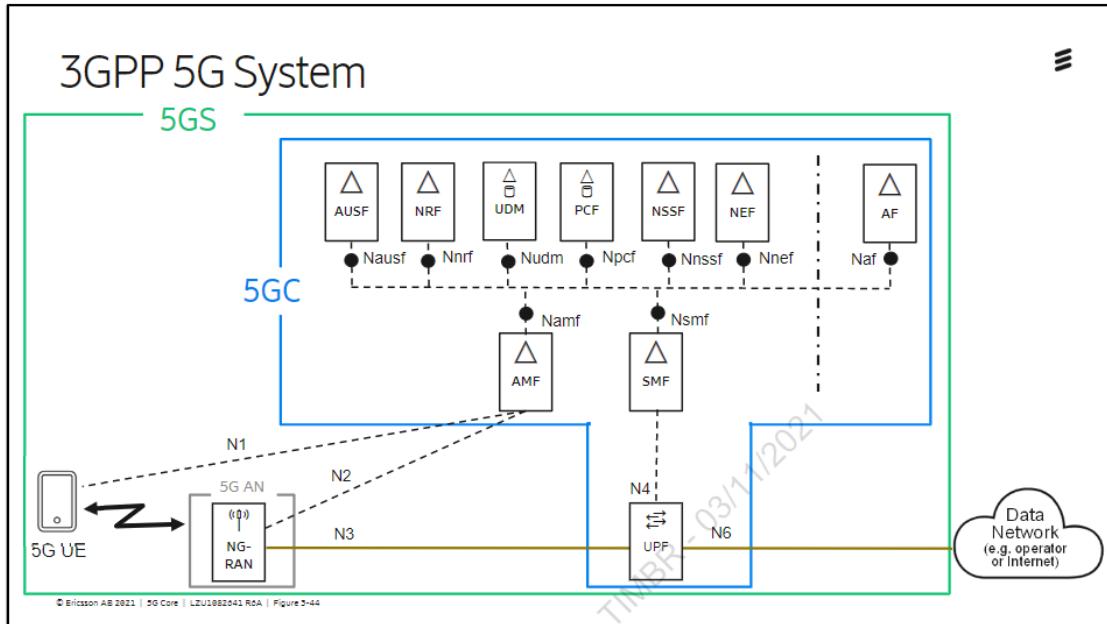


Figura 11 – Arquitetura Conceitual da rede Core Padrão 3GPP.

Abaixo a descrição de cada uma das funções de rede existentes no padrão 3GPP.

Tabela 3 – Descrição das funções de rede existentes no Core 5G.

| 5G Core NF | Função |
|---|--|
| Application Function (AF) | Fornecer serviços de aplicativos para o assinante. Exemplo pode ser para serviço de streaming de vídeo. Se um AF for confiável, ele pode interagir diretamente com as funções de rede 5GC ou, se for de terceiros, deve interagir com um NEF. |
| Access Management Function (AMF) | Funciona como um ponto de acesso entre os UEs e o control plane em uma rede core e fornece funções como gerenciamento de registro de UE, gerenciamento de conexão, gerenciamento de acessibilidade, gerenciamento de mobilidade, autenticação e autorização e short message. |
| Authentication Server Function (AUSF) | Prove serviços de autenticação unificada para acesso. |
| Network Exposure Function (NEF) | Uma entidade de rede que expõe os principais recursos de rede 3GPP a terceiros ou ambientes não 3GPP. O NEF também fornece segurança quando serviços ou funções de aplicativo (AFs) acessam os nós 5G Core. Pode ser considerado um proxy, ou ponto de API. |
| Network Slice Selection Function (NNSF) | Seleciona um grupo de Network Slice para o EU. |
| Network Repository Function (NRF) | Prove serviços de registro, descoberta, e autorizações de funções e mantém disponibilizado informações sobre outros NF. |
| Policy Control Function (PCF) | Controla as políticas de qualidade de serviço e de tarifação. |
| Session Management Function (SMF) | Prove funções como o gerenciamento de sessões, alocação e gerenciamento de endereços IP. |
| Unified Data Management (UDM) | Proporciona a gestão dos dados do assinante, como autenticação, identificação, autorização, registo e gestão de localização. |
| User Plane Function (UPF) | Encaminha pacotes, processa políticas de qualidade de serviço e fornece relatórios de uso de tráfego. |

Em uma rede comercial real são necessários mais de um elemento do mesmo tipo para ser capaz de suportar todo o tráfego gerado. Neste trabalho utilizaremos os dados fornecidos pela empresa TIM Brasil, uma das 3 maiores empresas de telefonia operando em solo nacional. Apesar de ter uma arquitetura próxima a descrita pelo 3GPP, existem algumas diferenças apresentadas na Figura 12. Dentro de cada função de rede está descrito a quantidade de elementos existentes em fevereiro de 2023, poucos meses após a inicialização do serviço. Hoje é sabido que os números são maiores.

3GPP 5G SYSTEM

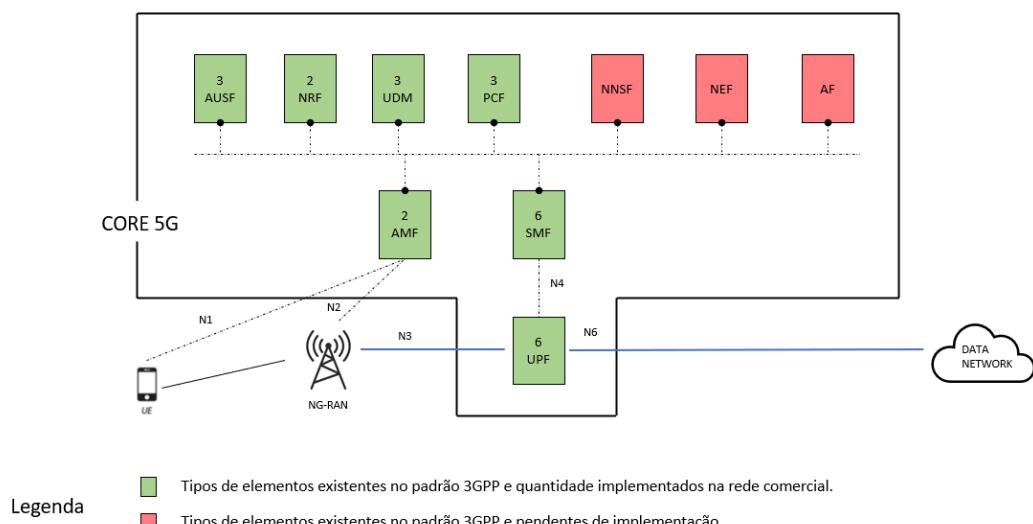


Figura 12 – Descrição das funções de rede existentes na TIM BR.

A implementação de uma rede 5G pode utilizar parte da infraestrutura da geração anterior, o que traz custos menores para a sua implantação, mesmo com perda de desempenho da rede. A esta solução é dada o nome de non-standalone (NSA). Nesta arquitetura a parte de control plane utiliza a rede 4G, o que faz com que a latência seja mais alta que no padrão Standalone (5G puro, ou SA). Neste trabalho iremos focar apenas no core 5G SA, excluindo elementos voltados para database, ou seja: UDM, AUSF e UDM, que a priori poderiam ser utilizados no Volte.

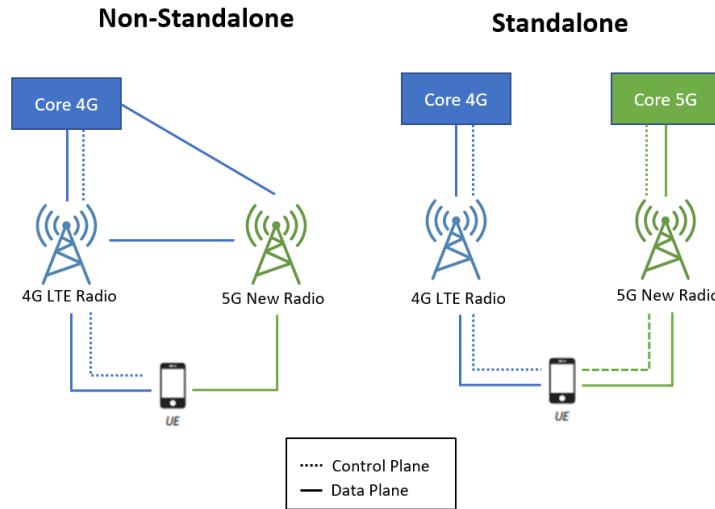


Figura 13 – Rede Standalone e Non-standalone.

Cada função de rede tem uma quantidade de KPIs diferente, estes representando eventos ou status que ocorrem no decorrer do tempo. Estes valores podem estar relacionados a todo o elemento, o que chamamos de whole system, mas também pode representar eventos em menores unidades da rede, informação vital para determinar problemas menores ou localizados. Alguns exemplos são:

- Uso de CPU médio e de pico do elemento de rede.
- Volume médio e máximo de tráfego de dados em um bairro de São Paulo.
- Quantidade de assinantes registrados no Código Nacional (CN) 22.

Cada KPI pode então ser interpretado como um conjunto de séries temporais, cada uma destas contendo um parâmetro associado que indica se a informação pertence a todo o elemento ou a uma determinada máquina virtual, site, rota etc. Além disso cada

indicador também tem um número de classe a ele associado. Este valor indica KPIs com proximidade de sentido como, por exemplo, CPU média e de pico. Analisando todos os contadores existentes nas funções de rede, quantos elementos efetivamente estão em operação e a quantidade de séries temporais geradas alcançamos os números apresentados na Tabela 4. Vale citar que os KPIs fornecidos tem periodicidade fixa de 30 minutos, sendo todos contadores numéricos, sem a inclusão de eventuais dados de clientes. Todos os elementos analisados são do fornecedor Huawei e foram coletados entre os meses de fevereiro/2023 e janeiro/2024.

Tabela 4 – Resumo quantitativo da rede core 5G SA da TIM Br.

| Network Functions | Quantity of KPIs | Network Elements | Temporal Series | % kpis Not Whole System |
|---|------------------|------------------|-----------------|-------------------------|
| AMF (Access and Mobility Management Function) | 815 | 2 | 11064 | 81,72% |
| NRF (Network Repository Function) | 315 | 2 | 4874 | 75,67% |
| SMF (Session Management Function) | 1928 | 6 | 255490 | 89,30% |
| UPF (User Plane Function) | 150 | 6 | 252591 | 94,45% |
| Total | 3208 | 16 | 524019 | 91,50% |

Cada KPI disponibilizado no core da rede 5G SA é representado por uma sequência numérica (código), sendo possível encontrar seu significado em materiais disponibilizados pelo fornecedor. Abaixo um exemplo do tipo de elemento NRF.

Tabela 5 – Exemplo da tradução dos KPIs do NRF.

| Classe | Tipo | KPI | Counter Name | Counter Unit |
|------------|---------------|------------|---|--------------|
| 1929379840 | Entire system | 1929471436 | Number of NF registration requests | times |
| 1929379840 | Entire system | 1929471437 | Number of successful NF registrations | times |
| 1929379840 | Entire system | 1929471438 | Number of NF failed registrations (other error) | times |
| 1929379840 | Entire system | 1929471439 | Number of NF failed registrations (invalid request) | times |
| 1929379840 | Entire system | 1929471440 | Number of NF failed registrations (system internal error) | times |
| 1929379840 | Entire system | 1929471441 | Number of NF deregistration requests | times |
| 1929379840 | Entire system | 1929471442 | Number of successful NF deregistrations | times |
| 1929379840 | Entire system | 1929471443 | Number of failed NF deregistrations (other error) | times |
| 1929379840 | Entire system | 1929471444 | Number of failed NF deregistrations (invalid request) | times |
| 1929379840 | Entire system | 1929471445 | Number of failed NF deregistrations (system internal error) | times |
| 1929379840 | Entire system | 1929471446 | Real-time number of suspended NF instances | number |
| 1929379840 | Entire system | 1929471447 | Real-time number of registered NF Instances | number |
| 1929379840 | Entire system | 1929471448 | Real-time number of NF instances not allowed for discovery | number |
| 1929379840 | Entire system | 1929471449 | Real-time number of suspended NFSs | number |
| 1929379840 | Entire system | 1929471450 | Real-time number of registered NFSs | number |
| 1929379840 | Entire system | 1929471451 | Real-time number of NFSs not allowed for discovery | number |

3.2 Análise Descritiva

Como apresentado na seção anterior, um KPI pode possuir múltiplas séries temporais associados, a depender das configurações existentes no elemento, ou seja, da quantidade de parâmetros existentes. Estas séries temporais pertencentes a um mesmo KPI não necessariamente tem o mesmo comportamento, a depender do tipo de tráfego existente, processos de ativação de rotas entre operadoras, dentre inúmeros outros motivos. Alguns exemplos:

- Em um feriado prolongado a tendência é que haja maior tráfego em regiões com perfil turístico e queda em outras cidades. Seria o caso, por exemplo, da cidade de Belo Horizonte em relação a cidades históricas próximas, como Ouro Preto.
- Durante o processo de ativação de um novo site na rede, esta passa por um caderno de testes para garantir a qualidade de sua inclusão. Durante estes testes é gerado um pequeno tráfego, que tem comportamento completamente diferente de um site já em operação.

Neste capítulo todos os exemplos utilizados serão de KPIs do elemento DDRJO01, um equipamento do tipo NRF. Todos os gráficos apresentados nesta seção são do mesmo período, entre os dias 15/07/23 às 00:00hs e 21/07/23 às 23:30hs. Apesar dos dados apresentados neste capítulo serem da mesma função de rede, os resultados apresentados aqui podem ser aplicados a outras sem perda de caracterização.

A Figura 14 ilustra a diferença de comportamento de duas séries temporais do mesmo indicador de desempenho. Analisando graficamente, conclui-se que não é possível agrupar o comportamento das séries temporais apenas verificando a qual KPI essas pertencem.

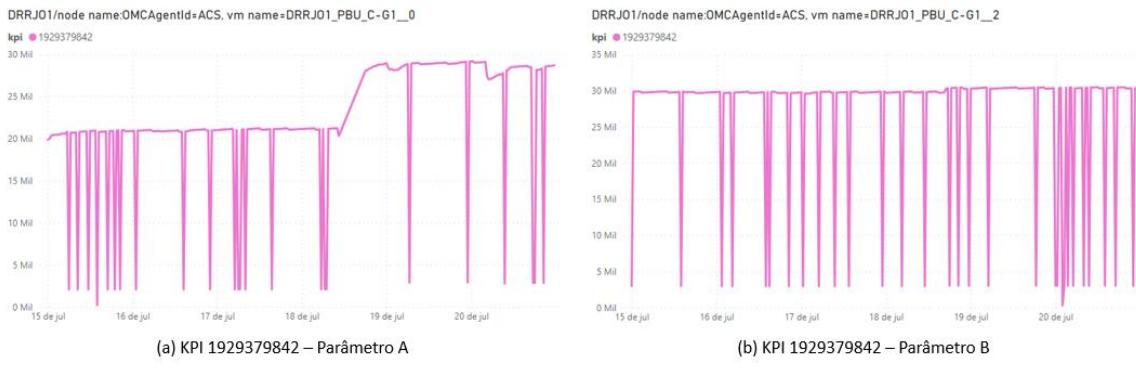


Figura 14– Séries temporais do mesmo indicador de desempenho e elemento com comportamentos diferentes.

Existem séries temporais com tendência incremental, como o exemplo abaixo, o KPI 1907953696 (*Number of discarded packets*). Estes, mais raros, não costumam ser monitorados pela operação, além de que normalmente existem outros KPIs com significado semelhante mais sumarizados dentro de um período definido. A Figura 15 ilustra este comportamento, sendo esse o KPI 1907953720 (*Number of discarded packets within this period*).

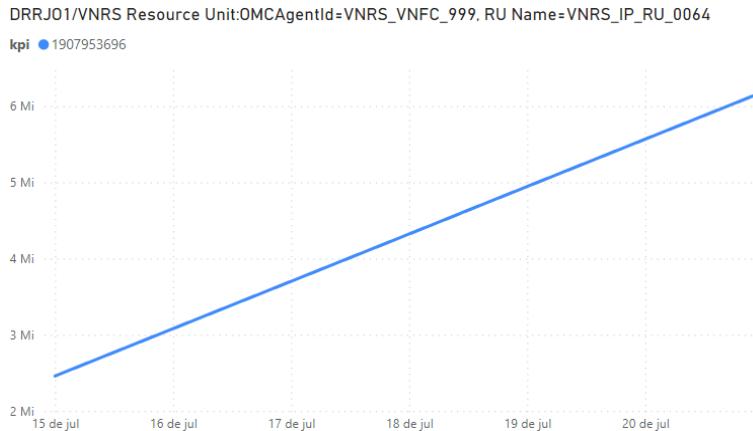


Figura 15 – Série temporal com comportamento incremental.

Por outro lado, existem séries temporais com comportamento mais estático, como o 1907953744 (*Number of IPv4 packets dropped due to exceptions within this period*), ilustrado na Figura 16. Vários indicadores têm este comportamento, como disponibilidade da rede, que deve estar sempre em 100%, ou indicadores de falha de serviço, que

idealmente devem ser séries nulas. Quanto à monitoração, neste tipo de cenário costuma-se criar thresholds fixos (acima ou abaixo de um valor pré-determinado).

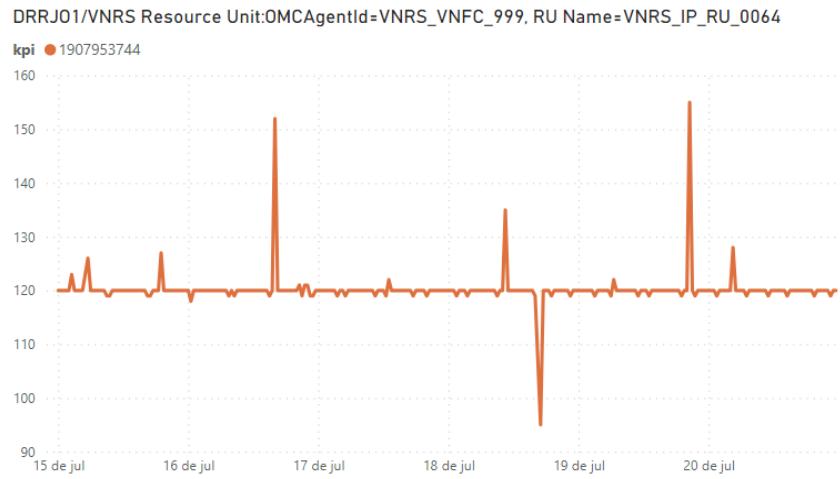


Figura 16 – Série temporal com tendência estática.

Muitas séries, por outro lado, têm sazonalidade, sendo estas baseadas no tráfego. Isto ocorre porque durante o dia a quantidade de usuários que utilizam a rede aumenta progressivamente até cerca das 13hs, cai levemente até as 17hs, cresce novamente até as 19hs, e por fim tem seu vale durante a madrugada. A maioria dos KPIs existentes tem este comportamento, sendo que sua monitoração costuma ser realizada por variação da média ponto a ponto em relação a dias anteriores. Um exemplo deste comportamento é o KPI 1929472037 (*Number of successful NFS/NF discoveries for NFs of a specific type*), ilustrado na Figura 17.

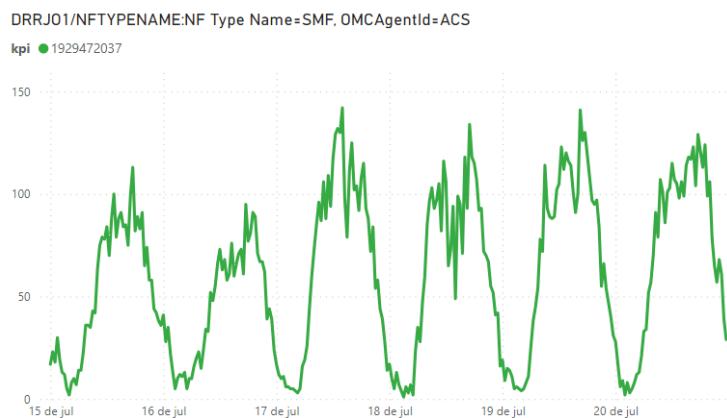


Figura 17 – Série temporal com sazonalidade devido ao tráfego.

Outras séries temporais não têm tendência e não são estáticas, e, portanto, tem comportamento mais randômico. Isto costuma acontecer em cenários de ativação de um novo serviço ou rota (em que o tráfego existente é experimental), ou em cenários de provisionamento, ou seja, inclusão de novos usuários na rede (costuma ser realizado em rajadas sem frequência pré-determinada). Nestes casos, nenhuma monitoração costuma ser aplicada. Um exemplo deste comportamento é indicado na Figura 18.

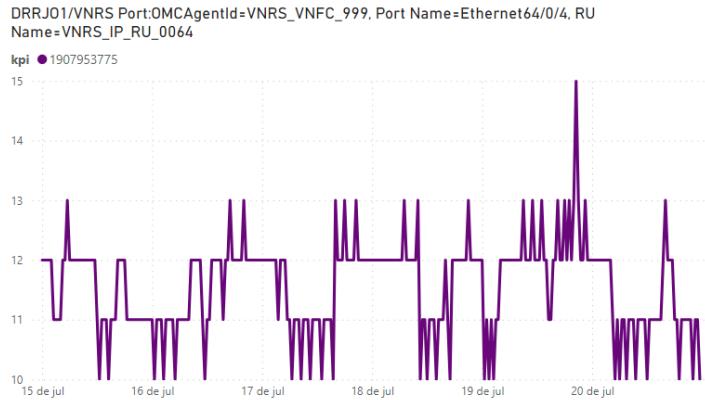


Figura 18 – Série temporal com comportamento randômico.

Eventos anômalos em indicadores de desempenho das redes core 5G são caracterizados principalmente por mudanças em sua amplitude, normalmente gerando picos ou vales e posteriormente retornando aos patamares anteriores. Normalmente quanto maior esta mudança de amplitude maior é o impacto sendo gerado na rede. Este fato é ilustrado na Figura 19.

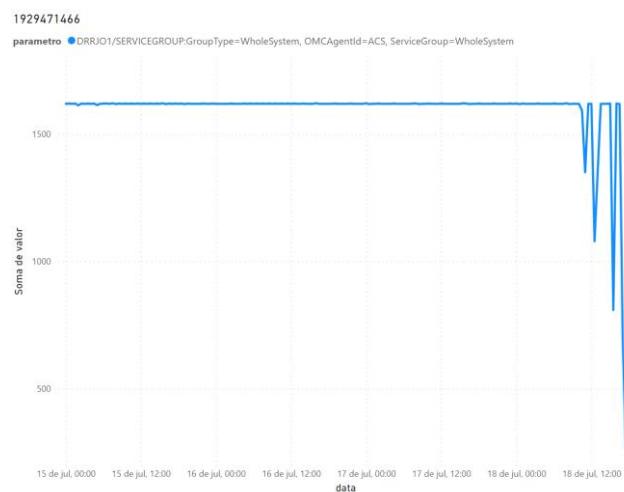


Figura 19 – Série temporal com evento anômalo.

3.3 Mudanças de tendência

Como abordado anteriormente, mudanças nas configurações da rede são esperadas, consequentemente é provável que mudanças na tendência dos KPIs ocorra sem uma periodicidade definida. Na Figura 20, entre os dias 18/07 e 19/07 ocorre uma mudança de tendência do kpi 1929379841(*Average Node CPU usage*), que não se trata de falha.

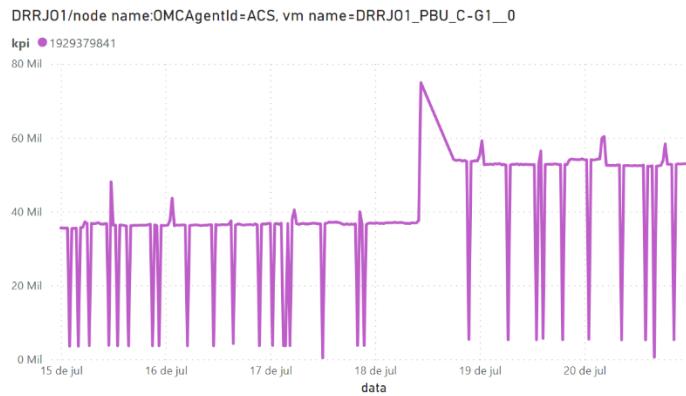


Figura 20 – Série Temporal que apresenta mudança de comportamento devido a configuração.

Este tipo de situação não é um cenário isolado, já que dezenas de configurações são realizadas todos os dias: ativações e desativações de sites, ampliação da capacidade do elemento (ou da sua VM), mudança entre elemento ativo ou stand-by, ativação de novos serviços, dentre várias outras. Outra observação pertinente, é que alterações em qualquer camada da rede, seja acesso ou backbone, também resultam em mudanças na tendência dos KPIs da rede core, já que ela centraliza toda a informação.

Estas mudanças no comportamento dos indicadores da rede em muitas situações não são possíveis de antecipar, mesmo considerando eventuais rollouts (sequência de implementações previstas) na infraestrutura. Isto deve-se ao fato de que a grande maioria dos KPIs não são monitorados ou em casos de novas redes, conhecidos. Abaixo outro exemplo de KPI do elemento DDRJ01, do tipo NRF, no período do dia 01/07/23 até o dia 30/07/23. Neste período ocorreram 6 mudanças de tendência no KPI, sendo que neste intervalo foi reportado uma falha que gerou impacto no serviço (dia 18/07/23 às 16:30hs).

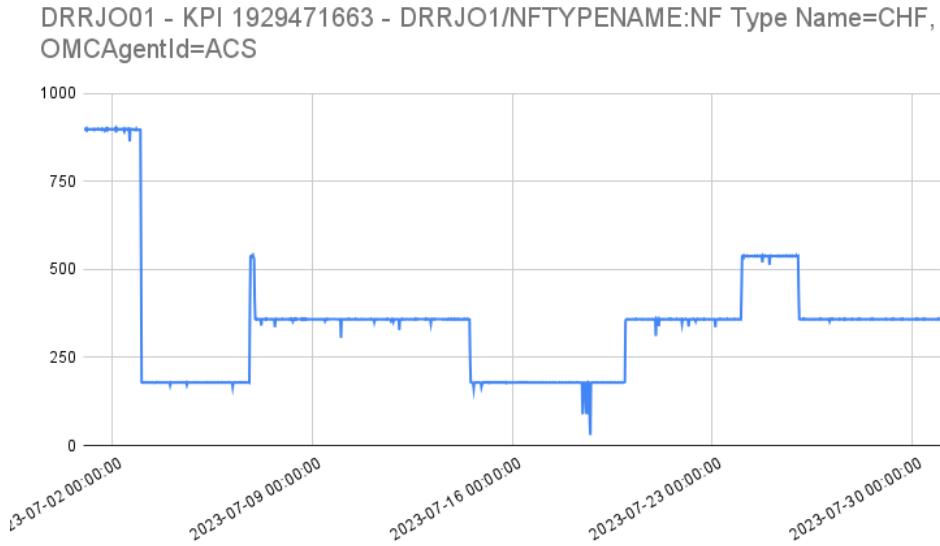


Figura 21 – Exemplo de KPI com mudanças de tendência.

Na seção anterior foi mencionado a alteração do tráfego a depender de eventuais feriados ou fins de semana. Na abordagem apresentada neste trabalho partimos do pressuposto de que não conhecemos a rede e nem o significado de cada um dos indicadores e seus respectivos parâmetros, ou seja, não é possível determinar qual a tendência que um determinado KPI vai assumir em uma situação como a descrita.

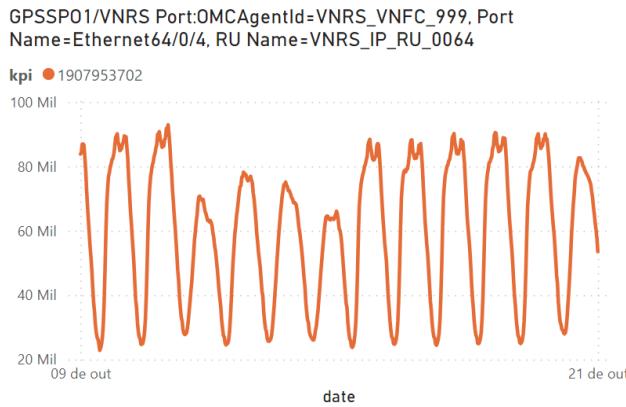


Figura 22 – KPI que apresenta mudança de comportamento devido ao feriado de 12/10.

3.4 Considerações Finais

Para o problema proposto, cada série temporal deve ser analisada individualmente, mesmo que pertença ao mesmo KPI, pois não existe uniformidade nas características

destas. Do ponto de vista quantitativo significa que é necessário analisar cerca de 524 mil séries temporais a cada 30 minutos, considerando apenas o cenário de fevereiro de 2023, poucos meses após o início da operação da rede 5G na TIM Brasil. Devido a este grande volume, é importante diminuir a quantidade de indicadores a serem analisados, considerando também que muitos não trazem informações relevantes, como aqueles que permanecem constantes durante todo o intervalo.

Sobre a detecção de eventos disruptivos em si, mudanças de comportamento podem ocorrer devido a configurações nos equipamentos, o que significa que é necessário definir o intervalo para o treinamento dos modelos, já que intervalos muito grandes de dados podem não ser o indicado, assim como intervalos muito pequenos. Além disso é necessário definir qual a melhor abordagem de treinamento, indutivo ou transdutivo.

Por fim, cada sequência numérica pode ou não ter sazonalidade ou uma tendência bem definida. Caso a abordagem fosse memorizar as características das séries, seria possível definir um método de detecção mais apropriado para cada cenário. Por exemplo, nas séries com sazonalidade aplica-se o SARIMA. No entanto, a rede tem comportamento dinâmico, sendo possível a mudança destes comportamentos. Por este motivo nossa proposta é uma solução caixa preta, sem conhecimento prévio de cada KPI. Logo, na aplicação dos métodos de detecção de anomalia, iremos utilizar diversos métodos em todas as séries buscando aquele que melhor se adequa ao cenário proposto.

4 Filtros

Como descrito no capítulo anterior, em termos de análise quantitativa, o número de indicadores de desempenho da rede core 5G SA da TIM BR que precisam ser analisado com periodicidade de 30 minutos são cerca de 524 mil. Do ponto de vista descritivo, existem KPIs que não são significativos, como por exemplo, indicadores que se mantiveram constante durante todo o intervalo analisado, ou KPIs com forte comportamento randômico. Além disso, existem indicadores que representam informações similares, como é o caso de valores medidos de CPU máxima e CPU média do mesmo parâmetro (apesar dos números diferentes em escala, neste caso a correlação é alta e um comportamento anômalo em um indica comportamento anômalo no outro).

Neste capítulo será descrito nossa proposta para a criação de filtros dinâmicos para diminuir a quantidade de KPIs a serem analisados pela etapa de detecção de anomalia. O desenvolvimento deste filtro foi realizado em conjunto com Adriano Guilherme Silva Rocha, na época colaborador da TIM Brasil, como parte do seu trabalho de fim de curso na USP sobre o título “Classificação dinâmica de KPIs relevantes em redes Core 5G”, ainda a ser apresentado. Neste trabalho, além da abordagem dos filtros o autor também propõe a classificação das séries temporais quanto a sua tendência, sazonalidade e ruido, o que possibilita a escolha mais assertiva de um método de detecção de anomalia. Esta segunda parte, de classificação, não faz parte do escopo deste trabalho.

Considerando as características de comportamento dos indicadores de desempenho apresentados no capítulo anterior, foi idealizado um filtro com 3 camadas, sendo estas:

- **Camada 1:** Eliminação de séries temporais “estáticas”.
- **Camada 2:** Eliminação de séries temporais randômicos.
- **Camada 3:** Eliminação de séries temporais com alta correlação.

A proposta é que a execução deste filtro seja sequencial, onde uma série que foi filtrada pela camada anterior não siga para a etapa posterior, ou seja, um pipeline. Este conceito está apresentado na Figura 23.

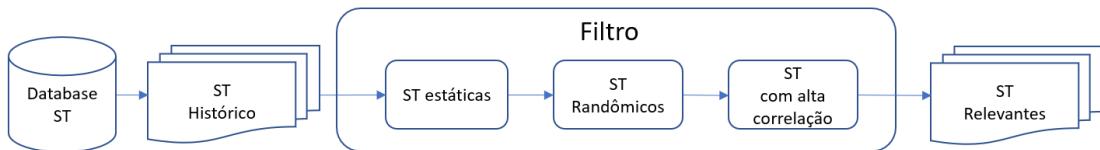


Figura 23 – Diagrama de Camadas do filtro dinâmico.

Com base no histórico existente das séries temporais, o objetivo é gerar uma lista de séries temporais relevantes, a ser enviada para a etapa de detecção de anomalia. É importante ressaltar que neste caso a análise não é feita por KPI tão somente, mas sim para cada uma das séries temporais geradas pelo mesmo, considerando os seus respectivos parâmetros. Como a rede está em constante mudança o filtro não pode ser estático, ou seja, ser executado uma única vez. Ao mesmo tempo, identificamos que sua execução não precisa ser em tempo real já que múltiplas mudanças de comportamento não ocorrem diariamente. Sendo assim definimos que uma nova lista de séries temporais relevantes seria gerada diariamente. Esta informação será utilizada no capítulo 6, Detecção de Anomalia. Para cada uma das camadas foram testados métodos ou abordagens diferentes, que serão explicados nas seções subsequentes. Quanto aos dados utilizados, foram testados diferentes intervalos sendo estes **5, 10, 15 e 30 dias**.

Em redes de telecomunicações existe o período de maior movimento, comumente chamado de PMM. Normalmente, são dois os PMMs observados durante o dia, sendo estes entre 10hs e 14hs e 18hs as 22hs. Estes intervalos, inclusive, são utilizados pelo órgão regulador Anatel para realizar medições oficiais de desempenho. Para o filtro decidimos utilizar estes intervalos para obtenção dos dados. Esta abordagem tem duas vantagens. A primeira diminui o volume de dados a ser analisado e a segunda vantagem é que justamente por ser o período de maior tráfego, e consequentemente mais estável, o filtro tende a ter melhor resultado (vários indicadores de desempenho ficam instáveis a noite e estáveis durante o dia).

4.1 Séries temporais estáticas

Nesta camada foram testadas duas abordagens diferentes, a serem executadas separadamente, sendo estas um **algoritmo de constantes** e a **entropia de Shannon**. O algoritmo de constante trata-se de um algoritmo simples, que verifica se todos os valores do intervalo analisado são iguais, tendo assim complexidade $O(n)$. A entropia de Shannon

pode ser utilizada para determinar o nível de incerteza de um sistema, o que consequentemente permite a quantificação da informação. Em um cenário de certeza, ou seja, todos os números da série temporal são iguais, a entropia torna-se 0.

$$H(X) := - \sum_{x \in \mathcal{X}} p(x) \log p(x)$$

A vantagem em utilizar a entropia de Shannon no problema proposto é que se pode filtrar séries que não são completamente estáticas, mas próximas o suficiente para serem consideradas como tal. Para a implementação do filtro consideramos que a entropia não poderia ser superior a 0,3.

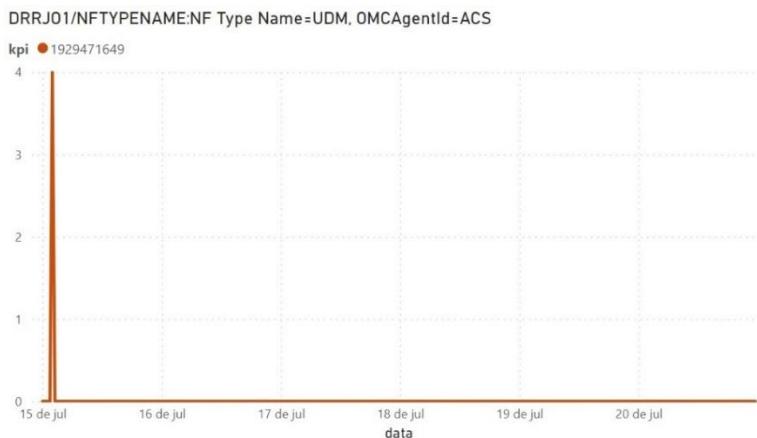


Figura 24 – Exemplo de série temporal que seria filtrada pela entropia de Shannon.

4.2 Séries temporais aleatórias

Como mencionado no capítulo 3, algumas séries podem apresentar comportamento randômicos. Estas devem ser eliminadas, já que não trazem informação útil. Para tal testamos duas abordagens diferentes, a primeira de **decompor a série temporal** e medir o seu ruido e outra de **contabilizar a quantidade de inversões** de sentido nas séries junto a distância euclidiana entre esses pontos.

Uma série temporal pode ser decomposta em 3 partes, sendo estas:

- T_t : Componente de tendência
- S_t : Componente sazonal
- R_t : Componente aleatória

A decomposição pode ser realizada de maneira aditiva ou multiplicativa.

$$y_t = T_t + S_t + R_t \text{ ou } y_t = T_t * S_t * R_t$$

As Figuras 25 e 26 ilustram a decomposição de séries temporais, sendo ambas da mesma série. A Figura 25 trata-se de decomposição multiplicativa enquanto a Figura 26, decomposição aditiva. Neste trabalho decidimos adotar o método aditivo.

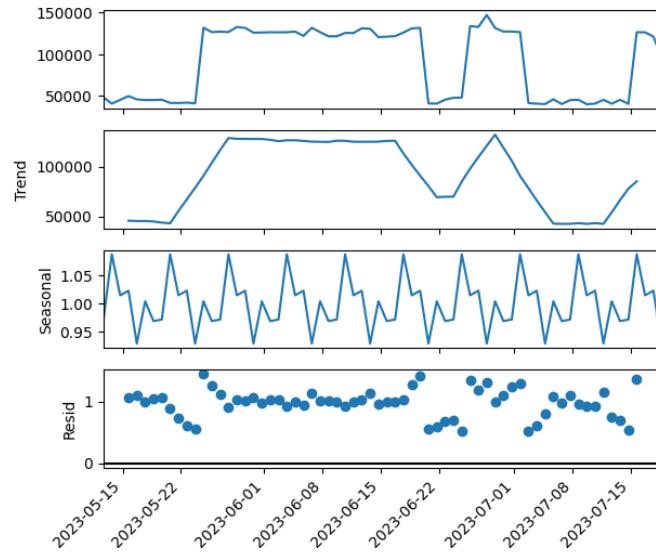


Figura 25 – Decomposição multiplicativa.

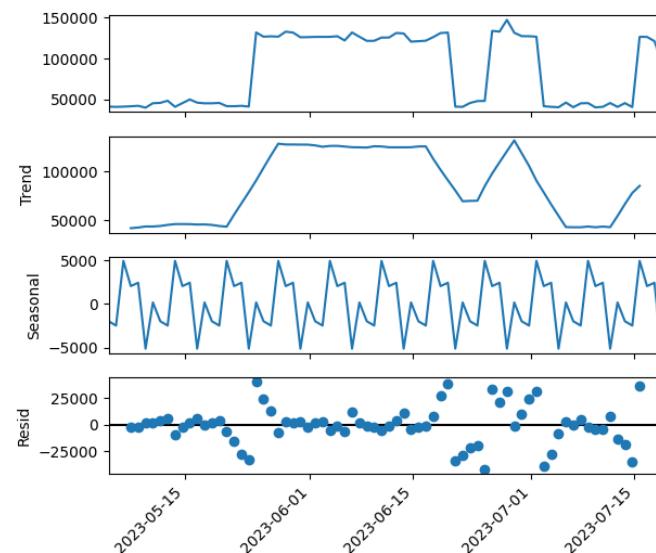


Figura 26 – Decomposição aditiva.

O algoritmo desenvolvido para esta primeira abordagem inicia-se normalizando a série temporal em valores que variam de 0 a 1, sendo 1 o maior valor no intervalo observado. Em sequência é realizada a decomposição da série normalizada em suas componentes, tendência, sazonalidade e ruído. Por fim, verificamos o desvio padrão do componente ruído. Através de experimentação definimos que o threshold para eliminação deveria ser séries com desvio padrão superior a 0.4, valor alto justamente com o objetivo de ser conservador, garantindo assim que apenas séries muito randômicas seriam excluídas.

A segundo algoritmo também inicia-se normalizando a série temporal (mesmo processo descrito anteriormente). Em seguida é verificado a quantidade de inversões de sentido (crescente ou decrescente) que a série possui no intervalo observado. Para cada uma destas inversões é calculado a distância euclidiana entre os pontos (inicial e o próximo pôs inversão). De maneira análoga utilizamos experimentação para encontrar a melhor maneira de realizar a classificação, sendo este:

*Se inversões > intervalo da amostra *10 e distância > intervalo da amostra * 6:*

Retorne “aleatório”

Se não:

Retorne “não aleatório”

A Figura 27 exemplifica o tipo de série temporal a ser filtrada na camada 2.

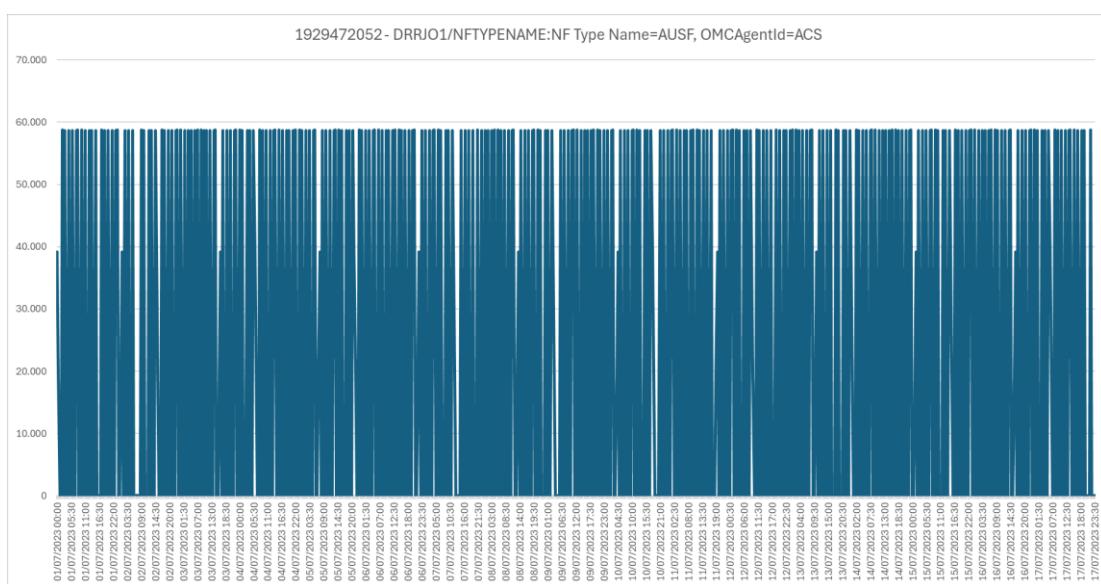


Figura 27 – Série temporal excluída pela camada 2.

4.3 Séries temporais com alta correlação

Todos os KPIs Huawei possuem a informação de classe, ou seja, indicadores de desempenho com significados próximos. O exemplo citado no capítulo anterior foi o de uso de CPU, que tem vários indicadores associados (médio, pico, minimo). Neste caso, se a correlação for alta, acima de 96%, um evento anômalo pode ser detectado apenas por uma série, assumindo assim que este evento também está ocorrendo na outra. É importante ressaltar que esta abordagem é válida para o mesmo parâmetro do mesmo elemento. Por exemplo, no caso dos KPIs derivados da utilização da CPU, para considerar a correlação as séries temporais envolvidas devem ser dos mesmos elementos de rede e ter o mesmo parâmetro.

Nesta camada foram implementadas duas abordagens, a primeira sendo **alta correlação dentro da mesma classe** e outra **correlação geral** entre todos as séries temporais. Nesta camada o objetivo é de fato eliminar as séries temporais, apenas indicando a quais outras ela está associada. Na detecção de anomalia, caso uma série tenha sido filtrada, mas está associada a outra de alta correlação, ambas terão a mesma classificação. No exemplo abaixo, apenas uma série seria avaliada mais ambas teriam a marcação de evento anômalo.

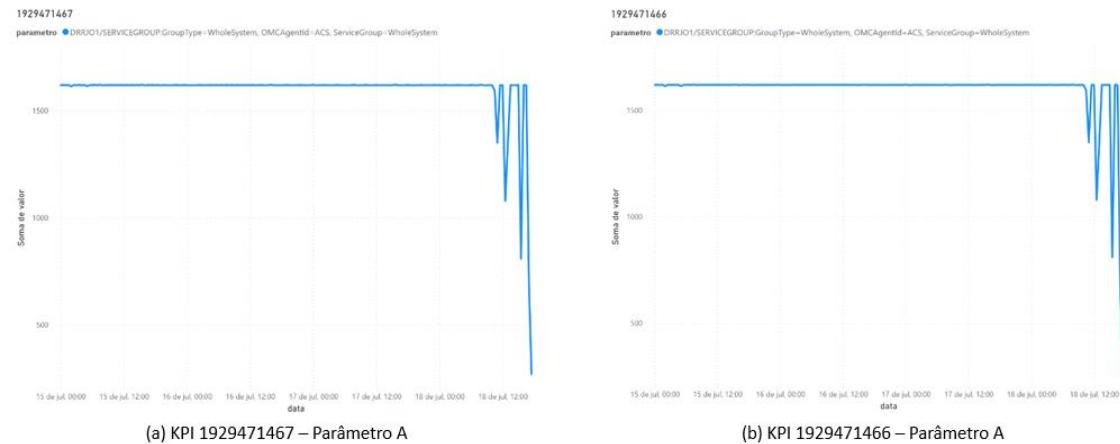


Figura 28 – 2 séries diferentes com o mesmo parâmetro e com alta correlação durante evento anômalo.

4.4 Resultados

Os testes foram executados em notebook com as seguintes características:

- Processador i7-1165G7 de 11º geração
- 16 GB de memória ram DDR5
- SSD de 512MB
- Sistema Operacional Windows 11 Home

Os experimentos foram realizados em todos os elementos da rede Core 5G SA e sumarizados nas Tabelas 5, 6 e 7. São apresentadas as janelas de tempo utilizadas, sendo que no caso do NRF o fim deste intervalo é D-1 (24 horas antes) em relação ao ponto de falha reportado, a ser explorado no capítulo 6. Por exemplo, em um dos pontos testados a falha ocorreu no dia 18/07/23. Logo o intervalo de 5 dias utilizado para o filtro foi entre os dias 12/07/23 e 17/07/23.

São apresentados os o tempo de execução, quantidade de séries que entraram no filtro, quantidade de séries que saíram e a percentagem filtrada. Como o filtro é sequencial, a segunda coluna (*Approach*), indica a sequência utilizada para o teste. Exemplo: Same Class – Std Resíduos – Shannon indica que a sequência utilizada foi: entropia de Shannon (camada 1), decomposição da série temporal (camada 2) e correlação dentro da mesma classe (camada 3).

Tabela 6 – Resultados da camada 1.

| Period (days) | Approach | CAMADA 1 | | | | | | | | | | | | | | | |
|----------------|--------------------|----------|--------|-----------|-------|----------|--------|-----------|-------|-----------|--------|-----------|-------|-----------|--------|--------|-------|
| | | AMF | | | | NRF | | | | SMF | | | | UPF | | | |
| Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | | |
| 5 | Constant Algorithm | 227,568 | 11058 | 87,66% | 1365 | 100,246 | 4874 | 92,94% | 344 | 4933,261 | 255336 | 90,32% | 24722 | 5474,801 | 252591 | 84,81% | 38362 |
| 10 | Constant Algorithm | 432,579 | 11060 | 87,50% | 1383 | 188,781 | 4874 | 92,82% | 350 | 10220,082 | 255336 | 89,81% | 26027 | 10022,385 | 252591 | 84,06% | 40266 |
| 15 | Constant Algorithm | 565,7 | 11064 | 87,34% | 1401 | 260,418 | 4874 | 92,74% | 354 | 15227,383 | 255490 | 89,50% | 26822 | 15531,587 | 252591 | 83,57% | 41492 |
| 30 | Constant Algorithm | 1198,649 | 11072 | 87,13% | 1425 | 581,066 | 4874 | 91,46% | 416 | 52704,253 | 255681 | 88,97% | 28212 | 30661,039 | 252601 | 82,39% | 44479 |
| 5 | Shannon Entropy | 624,169 | 11058 | 89,74% | 1135 | 280,323 | 4874 | 94,73% | 257 | 14721,065 | 255336 | 91,52% | 21659 | 14637,58 | 252591 | 86,67% | 33680 |
| 10 | Shannon Entropy | 1034,846 | 11060 | 89,41% | 1171 | 448,716 | 4874 | 94,60% | 263 | 23403,294 | 255336 | 91,34% | 22124 | 22136,936 | 252591 | 86,48% | 34147 |
| 15 | Shannon Entropy | 1418,663 | 11064 | 89,11% | 1205 | 560,384 | 4874 | 94,48% | 269 | 32355,482 | 255490 | 91,35% | 22112 | 30111,656 | 252591 | 86,52% | 34057 |
| 30 | Shannon Entropy | 2526,496 | 11072 | 88,92% | 1227 | 1126,297 | 4874 | 94,17% | 284 | 64704,416 | 255681 | 91,57% | 21561 | 54763,037 | 252601 | 86,83% | 33262 |

Tabela 7 – Resultados da camada 2.

| Period (days) | Approach | CAMADA 2 | | | | | | | | | | | | | | | |
|----------------|----------------------------------|----------|--------|-----------|-------|----------|--------|-----------|-------|-----------|--------|-----------|-------|-----------|--------|-------|-------|
| | | AMF | | | | NRF | | | | SMF | | | | UPF | | | |
| Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | | |
| 5 | Deteta Aleatoriedade - Constante | 68,602 | 1365 | 0,29% | 1361 | 18,233 | 344 | 3,78% | 331 | 1250,621 | 24722 | 0,31% | 24645 | 2048,883 | 38362 | 0,59% | 38137 |
| 10 | Deteta Aleatoriedade - Constante | 147,026 | 1383 | 0,29% | 1379 | 34,682 | 350 | 3,14% | 339 | 2472,502 | 26027 | 0,18% | 25979 | 4086,044 | 40266 | 0,38% | 40113 |
| 15 | Deteta Aleatoriedade - Constante | 199,345 | 1401 | 0,00% | 1401 | 51,819 | 354 | 1,98% | 347 | 3895,593 | 26822 | 0,15% | 26781 | 6246,356 | 41492 | 0,25% | 41390 |
| 30 | Deteta Aleatoriedade - Constante | 406,055 | 1425 | 0,00% | 1425 | 113,893 | 416 | 1,44% | 410 | 7419,077 | 28212 | 0,09% | 28186 | 12614,785 | 44479 | 0,10% | 44436 |
| 5 | Std Resíduos - Constante | 446,959 | 1365 | 0,00% | 1365 | 118,762 | 344 | 0,00% | 344 | 8528,627 | 24722 | 0,00% | 24722 | 13014,589 | 38362 | 0,00% | 38362 |
| 10 | Std Resíduos - Constante | 511,915 | 1383 | 0,00% | 1383 | 135,647 | 350 | 0,00% | 350 | 9335,351 | 26027 | 0,00% | 26027 | 15773,368 | 40266 | 0,00% | 40266 |
| 15 | Std Resíduos - Constante | 551,337 | 1401 | 0,14% | 1399 | 144,913 | 354 | 0,00% | 354 | 11361,78 | 26822 | 0,08% | 26800 | 17265,548 | 41492 | 0,00% | 41492 |
| 30 | Std Resíduos - Constante | 706,33 | 1425 | 0,14% | 1423 | 221,991 | 416 | 0,00% | 416 | 14596,527 | 28212 | 0,00% | 28212 | 22688,265 | 44479 | 0,00% | 44477 |
| 5 | Deteta Aleatoriedade - Shannon | 72,103 | 1135 | 0,35% | 1131 | 13,937 | 257 | 5,06% | 244 | 1180,832 | 21659 | 0,36% | 21582 | 1814,212 | 33680 | 0,67% | 33455 |
| 10 | Deteta Aleatoriedade - Shannon | 130,872 | 1171 | 0,34% | 1167 | 35,631 | 263 | 4,18% | 252 | 2328,822 | 22124 | 0,22% | 22076 | 3568,059 | 34147 | 0,45% | 33994 |
| 15 | Deteta Aleatoriedade - Shannon | 194,16 | 1205 | 0,00% | 1205 | 46,855 | 269 | 2,60% | 262 | 3262,192 | 22112 | 0,19% | 22071 | 5270,898 | 34057 | 0,30% | 33955 |
| 30 | Deteta Aleatoriedade - Shannon | 419,359 | 1227 | 0,00% | 1227 | 105,682 | 284 | 2,11% | 278 | 6530,437 | 21561 | 0,12% | 21535 | 10086,626 | 33262 | 0,13% | 33219 |
| 5 | Std Resíduos - Shannon | 415,34 | 1135 | 0,00% | 1135 | 84,691 | 257 | 0,00% | 257 | 7417,907 | 21659 | 0,00% | 21659 | 11187,623 | 33680 | 0,00% | 33680 |
| 10 | Std Resíduos - Shannon | 425,469 | 1171 | 0,00% | 1171 | 108,793 | 263 | 0,00% | 263 | 8082,955 | 22124 | 0,00% | 22124 | 12667,074 | 34147 | 0,00% | 34147 |
| 15 | Std Resíduos - Shannon | 486,48 | 1205 | 0,08% | 1204 | 118,574 | 269 | 0,00% | 269 | 9050,347 | 22112 | 0,09% | 22092 | 13511,158 | 34057 | 0,00% | 34057 |
| 30 | Std Resíduos - Shannon | 622,324 | 1227 | 0,16% | 1225 | 168,313 | 284 | 0,00% | 284 | 10640,864 | 21561 | 0,00% | 21561 | 15789,428 | 33262 | 0,00% | 33262 |

Tabela 8 – Resultados da camada 3.

| Period (days) | Approach | CAMADA 3 | | | | | | | | | | | | | | | |
|----------------|--|-----------|-------|----------|--------|-----------|-------|----------|--------|------------|-------|----------|--------|------------|-------|----------|--------|
| | | AMF | | | | NRF | | | | SMF | | | | UPF | | | |
| | | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out | Time (ms) | TS IN | TS DEL % | TS Out |
| 5 | Same Class-Detecta Aleatoriedade - Constante | 561,552 | 1361 | 32,04% | 925 | 73,381 | 331 | 40,79% | 196 | 31688,192 | 24645 | 61,03% | 9605 | 54111,173 | 38137 | 59,61% | 15402 |
| 10 | Same Class-Detecta Aleatoriedade - Constante | 566,838 | 1379 | 29,15% | 977 | 95,677 | 339 | 38,64% | 208 | 37451,41 | 25979 | 61,09% | 10108 | 67347,249 | 40113 | 59,44% | 16269 |
| 15 | Same Class-Detecta Aleatoriedade - Constante | 750,012 | 1401 | 28,12% | 1007 | 102,424 | 347 | 38,62% | 213 | 41375,929 | 26781 | 61,51% | 10308 | 72567,228 | 41390 | 61,98% | 15736 |
| 30 | Same Class-Detecta Aleatoriedade - Constante | 845,286 | 1425 | 27,93% | 1027 | 200,135 | 410 | 39,76% | 247 | 52041,517 | 28186 | 64,36% | 10046 | 97498,78 | 44436 | 67,39% | 14491 |
| 5 | Same Class- Std Residuos - Constante | 557,259 | 1365 | 32,16% | 926 | 83,402 | 344 | 40,41% | 205 | 31973,35 | 24722 | 61,01% | 9638 | 56710,358 | 38362 | 59,64% | 15484 |
| 10 | Same Class-Std Residuos - Constante | 624,555 | 1383 | 29,28% | 978 | 100,107 | 350 | 38,29% | 216 | 37113,418 | 26027 | 61,08% | 10131 | 67694,749 | 40266 | 59,48% | 16314 |
| 15 | Same Class-Std Residuos - Constante | 697,314 | 1399 | 28,16% | 1005 | 116,877 | 354 | 37,85% | 220 | 42380,456 | 26800 | 61,54% | 10306 | 72764,528 | 41492 | 62,00% | 15769 |
| 30 | Same Class-Std Residuos - Constante | 919,832 | 1423 | 27,97% | 1025 | 171,21 | 416 | 39,18% | 253 | 52046,112 | 28212 | 64,31% | 10068 | 99903,179 | 44477 | 67,39% | 14505 |
| 5 | Geral correlacao - Detecta Aleatoriedade - Constante | 6361,749 | 1361 | 41,29% | 799 | 375,253 | 331 | 44,71% | 183 | 717671,919 | 24645 | 70,27% | 7326 | 1659525,48 | 38137 | 70,75% | 11154 |
| 10 | Geral correlacao - Detecta Aleatoriedade - Constante | 6565,65 | 1379 | 38,14% | 853 | 410,358 | 339 | 40,41% | 202 | 843006,943 | 25979 | 70,14% | 7757 | 1968245,09 | 40113 | 70,29% | 11919 |
| 15 | Geral correlacao - Detecta Aleatoriedade - Constante | 6831,709 | 1401 | 36,26% | 893 | 452,029 | 347 | 40,35% | 207 | 901864,182 | 26781 | 70,68% | 7853 | 2103577,5 | 41390 | 71,95% | 11610 |
| 30 | Geral correlacao - Detecta Aleatoriedade - Constante | 7719,396 | 1425 | 35,58% | 918 | 737,919 | 410 | 43,66% | 231 | 1069430,28 | 28186 | 73,03% | 7603 | 2592754,77 | 44436 | 75,96% | 10683 |
| 5 | Geral correlacao -- Std Residuos - Constante | 6287,954 | 1365 | 41,39% | 800 | 405,769 | 344 | 44,19% | 192 | 722558,308 | 24722 | 70,23% | 7359 | 1667330,64 | 38362 | 70,74% | 11226 |
| 10 | Geral correlacao -- Std Residuos - Constante | 6579,703 | 1383 | 38,25% | 854 | 443,174 | 350 | 40,00% | 210 | 815994,351 | 26027 | 70,11% | 7780 | 1958175,87 | 40266 | 70,30% | 11957 |
| 15 | Geral correlacao -- Std Residuos - Constante | 6890,234 | 1399 | 36,31% | 891 | 468,063 | 354 | 39,55% | 214 | 886550,59 | 26800 | 70,71% | 7851 | 2112160,16 | 41492 | 71,95% | 11637 |
| 30 | Geral correlacao -- Std Residuos - Constante | 7791,405 | 1423 | 35,63% | 916 | 718,507 | 416 | 43,03% | 237 | 1024710,91 | 28212 | 72,97% | 7625 | 2469180,38 | 44477 | 75,95% | 10695 |
| 5 | Same Class-Detecta Aleatoriedade - Shannon | 424,301 | 1131 | 29,35% | 799 | 43,374 | 244 | 33,20% | 163 | 26138,489 | 21582 | 62,33% | 8129 | 37917,686 | 33455 | 60,08% | 13356 |
| 10 | Same Class-Detecta Aleatoriedade - Shannon | 484,181 | 1167 | 28,02% | 840 | 49,271 | 252 | 31,75% | 172 | 28362,743 | 22076 | 63,42% | 8075 | 41747,022 | 33994 | 60,10% | 13563 |
| 15 | Same Class-Detecta Aleatoriedade - Shannon | 560,057 | 1205 | 27,63% | 872 | 67,746 | 262 | 31,68% | 179 | 29280,134 | 22071 | 64,36% | 7867 | 43647,771 | 33955 | 62,97% | 12574 |
| 30 | Same Class-Detecta Aleatoriedade - Shannon | 832,081 | 1227 | 28,93% | 872 | 143,136 | 278 | 33,45% | 185 | 32622,629 | 21535 | 67,06% | 7094 | 49636,457 | 33219 | 68,16% | 10578 |
| 5 | Same Class- Std Residuos - Shannon | 432,366 | 1135 | 29,52% | 800 | 49,766 | 257 | 33,07% | 172 | 26282,244 | 21659 | 62,32% | 8162 | 39283,255 | 33680 | 60,10% | 13438 |
| 10 | Same Class-Std Residuos - Shannon | 492,011 | 1171 | 28,18% | 841 | 65,182 | 263 | 31,56% | 180 | 28338,633 | 22124 | 63,40% | 8098 | 42269,755 | 34147 | 60,15% | 13608 |
| 15 | Same Class-Std Residuos - Shannon | 585,485 | 1204 | 27,66% | 871 | 86,101 | 269 | 30,86% | 186 | 29814,912 | 22092 | 64,39% | 7867 | 43669,457 | 34057 | 62,98% | 12607 |
| 30 | Same Class-Std Residuos - Shannon | 808,642 | 1225 | 28,98% | 870 | 154,679 | 284 | 32,75% | 191 | 32621,91 | 21561 | 67,00% | 7116 | 50242,429 | 33262 | 68,15% | 10594 |
| 5 | Geral correlacao - Detecta Aleatoriedade - Shannon | 4440,138 | 1131 | 37,49% | 707 | 216,267 | 244 | 34,84% | 159 | 554905,703 | 21582 | 70,79% | 6305 | 1264957,2 | 33455 | 70,74% | 9790 |
| 10 | Geral correlacao - Detecta Aleatoriedade - Shannon | 4718,714 | 1167 | 35,56% | 752 | 245,687 | 252 | 33,33% | 168 | 605834,197 | 22076 | 71,63% | 6263 | 1313332,17 | 33994 | 70,59% | 9999 |
| 15 | Geral correlacao - Detecta Aleatoriedade - Shannon | 5198,916 | 1205 | 35,19% | 781 | 275,995 | 262 | 33,21% | 175 | 615053,649 | 22071 | 72,52% | 6065 | 1357226,5 | 33955 | 72,60% | 9302 |
| 30 | Geral correlacao - Detecta Aleatoriedade - Shannon | 5842,7 | 1227 | 36,35% | 781 | 403,144 | 278 | 37,41% | 174 | 612112,749 | 21535 | 74,43% | 5506 | 1425131,6 | 33219 | 76,24% | 7894 |
| 5 | Geral correlacao -- Std Residuos - Shannon | 4516,982 | 1135 | 37,62% | 708 | 229,271 | 257 | 34,63% | 168 | 564173,964 | 21659 | 70,74% | 6338 | 1289810,07 | 33680 | 70,72% | 9862 |
| 10 | Geral correlacao -- Std Residuos - Shannon | 4758,693 | 1171 | 35,70% | 753 | 272,818 | 263 | 33,08% | 176 | 610380,332 | 22124 | 71,59% | 6286 | 1340674,25 | 34147 | 70,61% | 10037 |
| 15 | Geral correlacao -- Std Residuos - Shannon | 5179,667 | 1204 | 35,22% | 780 | 268,11 | 269 | 32,34% | 182 | 615376,486 | 22092 | 72,55% | 6065 | 1360931,28 | 34057 | 72,61% | 9329 |
| 30 | Geral correlacao -- Std Residuos - Shannon | 5890,299 | 1225 | 36,41% | 779 | 398,308 | 284 | 36,62% | 180 | 619944,887 | 21561 | 74,36% | 5528 | 1400724,21 | 33262 | 76,23% | 7908 |

Sobre a camada 1 pode-se observar, como esperado, que a entropia de Shannon filtrou mais séries temporais em todos os cenários testados. Além disso, em média foi possível diminuir cerca de 87,93% das séries, o que indica que a maior parte dos KPIs tem comportamento constante. É importante ressaltar que um evento anômalo pode alterar a amplitude das séries temporais, logo um conjunto constante não pode ser simplesmente descartado. Por este motivo está camada não necessariamente elimina a série, e sim indica um valor de constante (moda), que deve ser utilizado para comparar com o ponto a ser analisado. Caso os valores sejam iguais, a série deve ser descartada, caso contrário deve passar pelo detector de anomalia. Quanto ao tempo de execução, a entropia de Shannon demanda mais tempo para executar, que o algoritmo de constantes, sendo que a diferença de tempo entre as abordagens vai aumentando conforme o tamanho da janela de dados aumenta. No entanto, este comportamento não é um problema pois o filtro não necessita ser executado em tempo real.

Na camada seguinte, pode-se observar que a quantidade de séries temporais excluídas foi muito menor que na camada 1, cerca de 0,16% (média). Em alguns tipos de elemento não houve eliminação, o que sugere que os KPIs destes elementos estão mais estáveis. Ao analisar o gráfico das séries eliminadas, percebeu-se que o comportamento foi o esperado, excluindo apenas aquelas com alta aleatoriedade. Como exemplo temos a figura 27.

Na última camada houve grande variação na quantidade de séries temporais excluídas, indo de 28% em relação a sua entrada até 71%, a depender do tipo do elemento e de sua entrada. Este comportamento indica que muitos KPIs de rede tendem a ter significado semelhante. Além disso, é possível perceber que a abordagem geral filtrou mais séries que a abordagem na mesma classe, no entanto seu tempo de execução é muito superior, outro comportamento que também é esperado.

4.5 Considerações Finais

Em média utilizando um intervalo de 15 dias de dados foi o suficiente para obter resultados satisfatórios de eliminação de séries temporais, sendo esta redução cerca de 96,71% do total, ou seja, de um cenário de 524 mil séries temporais foi possível chegar ao número de 18 mil após a aplicação das 3 camadas. Considerando este resultado obtido,

é possível isolar as séries temporais que representam mudanças no estado da rede com maior ganho de informação.

No entanto, até este momento não podemos afirmar que o que foi retirado não irá impactar na detecção de eventos anômalos, pois o filtro pode mascarar falhas de rede em si. Portanto, para analisar por completo o desempenho do que foi apresentado neste capítulo é necessário avaliá-lo após a implementação dos métodos de detecção de anomalia. Por este motivo, no capítulo 6, detecção de eventos anômalos, os resultados do filtro serão levados em consideração para medir a assertividade do framework proposto.

Pelo mesmo motivo apresentado, a definição de qual abordagem utilizar em cada camada deste filtro, será realizado após a apresentação dos resultados da detecção de anomalia, desta forma não avaliando apenas a quantidade de séries temporais retiradas, mas também a assertividade.

5 Abordagens de aprendizado

Como abordado no capítulo 3, mudanças de tendência podem acontecer nas séries temporais da rede core 5G, sendo a maioria destas alterações imprevisíveis. Não se trata de uma falha em si, mas de uma mudança de comportamento devido a alteração de configuração. Devido a este fato, a abordagem de treinamento indutiva, ou seja, aquela em que se treina o modelo com base nos dados históricos e reutiliza o mesmo indefinidamente para realizar previsões, apresentou resultados ruins mesmo variando a janela de dados nas quais obtém-se os dados de treinamento e o período antes da predição em si. Neste capítulo iremos explorar mais as características mutáveis das séries temporais da rede, tendo foco na nossa abordagem de treinamento escolhida, a transdutiva, o que significa que para cada nova predição iremos treinar o modelo novamente, sempre buscando apenas o próximo ponto. A abordagem híbrida, ou seja, definir uma frequência para treinar novamente o modelo buscando aprimorar os resultados não faz parte do escopo desta pesquisa.

Para a demonstração das abordagens utilizaremos duas séries temporais do NRF como exemplo, sendo estas de dois KPIs completamente diferentes (classes distintas), uma sendo whole system e outra subdividida por parâmetro. Para simplificar, chamaremos estas de série temporal 1 e 2. O intervalo escolhido foi o mesmo para ambas, entre os dias 18/05/23 e 18/07/23. O último ponto das séries, dia 18/07/23 às 17:00 trata-se de falha reportada pela TIM BR, ou seja, as duas séries são relevantes para a identificação de eventos anômalos.

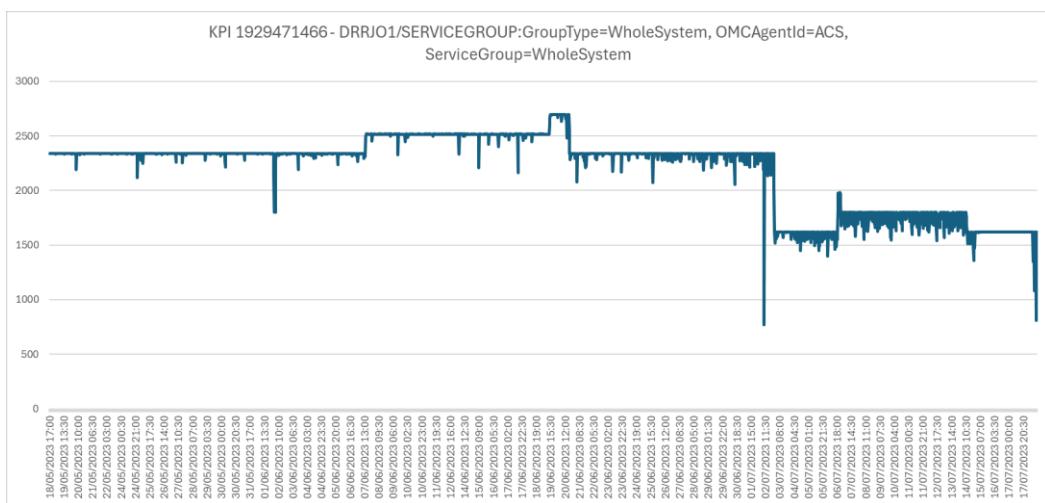


Figura 29 – Série temporal 1.

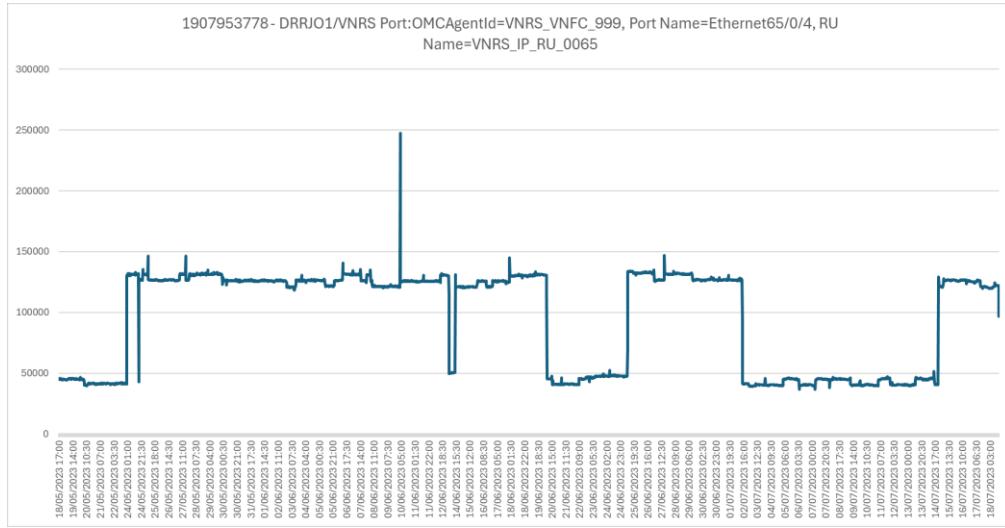


Figura 30 – Série temporal 2.

5.1 Testes realizados

Para as demonstrações utilizamos apenas um algoritmo, o ARIMA. É sabido que este método necessita de uma série temporal sem tendência, sendo assim necessário adequação dos dados ou a divisão das séries nos seus intervalos. No entanto, nosso objetivo neste trabalho é avaliar a possibilidade de um framework caixa-preta no qual não se tem o conhecimento prévio de cada série temporal. Portanto, utilizamos o método apenas para demonstrar a diferença entre as abordagens de treinamento, e por isso o ajuste realizado foi apenas nos parâmetros AR (auto regressão), I (integração) e MA (média móvel), obtidos através da biblioteca Auto Arima. Os testes foram estruturados da seguinte forma:

- **Dias antes do evento anômalo:** Utilizamos 3 possibilidades: 7, 15 e 30 dias. Ou seja, esta é a quantidade de dias que o algoritmo deve prever. Intuitivamente quanto menor a quantidade de dias antes do evento mais assertiva deve ser o valor encontrado.
- **Janela de dados para treinamento:** Utilizamos 4 possibilidades: 7, 15, 30 e 60 dias antes do primeiro ponto a ser predito. Intervalos pequenos de dados tendem a dar resultados ruins, e no nosso problema proposto um intervalo muito grande também pode significar resultados menos precisos.

Além disso, foram testadas 2 abordagens indutivas e uma abordagem transdutiva. Como não temos os dados classificados (evento anômalo ou não) para todos os intervalos definidos, toda a análise realizada foi feita graficamente.

- **Transdutivo:** Para cada ponto a ser previsto o modelo é treinado novamente com a janela de dados imediatamente anterior disponível. Exemplo: para prever o valor da série do dia 18/07/23 17:00 utilizamos a janela de treinamento entre os dias 18/06/23 16:30 até o dia 18/07/23 16:30.
- **Indutivo:** O modelo é treinado apenas uma vez utilizando os dados da primeira janela de dados disponível. Exemplo: Em uma janela de 30 dias com a previsão de 7 dias, para prever o valor da série do dia 18/07/23 17:00 utilizamos os dados 11/06/23 16:30 até o dia 11/07/23 16:30.
- **Indutivo com janela deslizante:** Como nosso objetivo é prever apenas um ponto, treinamos o modelo com o tamanho da janela fixo, no entanto deslizante no tempo. Exemplo: Em uma janela de 30 dias com a previsão de 7 dias, para prever o valor da série do dia 18/07/23 17:00 utilizamos os dados 11/06/23 16:30 até o dia 11/07/23 16:30. Neste mesmo cenário para prever o valor da série do dia 18/07/23 16:30 utilizamos os dados 11/06/23 16:00 até o dia 11/07/23 16:00.

5.2 Resultados

Para simplificar a apresentação dos resultados, esta será feita com base na quantidade de previsões a serem realizadas. Como a periodicidade dos dados é de 30 minutos, em um intervalo de 7 dias serão realizadas 336 previsões, enquanto em um intervalo de 30 dias serão realizadas 1440.

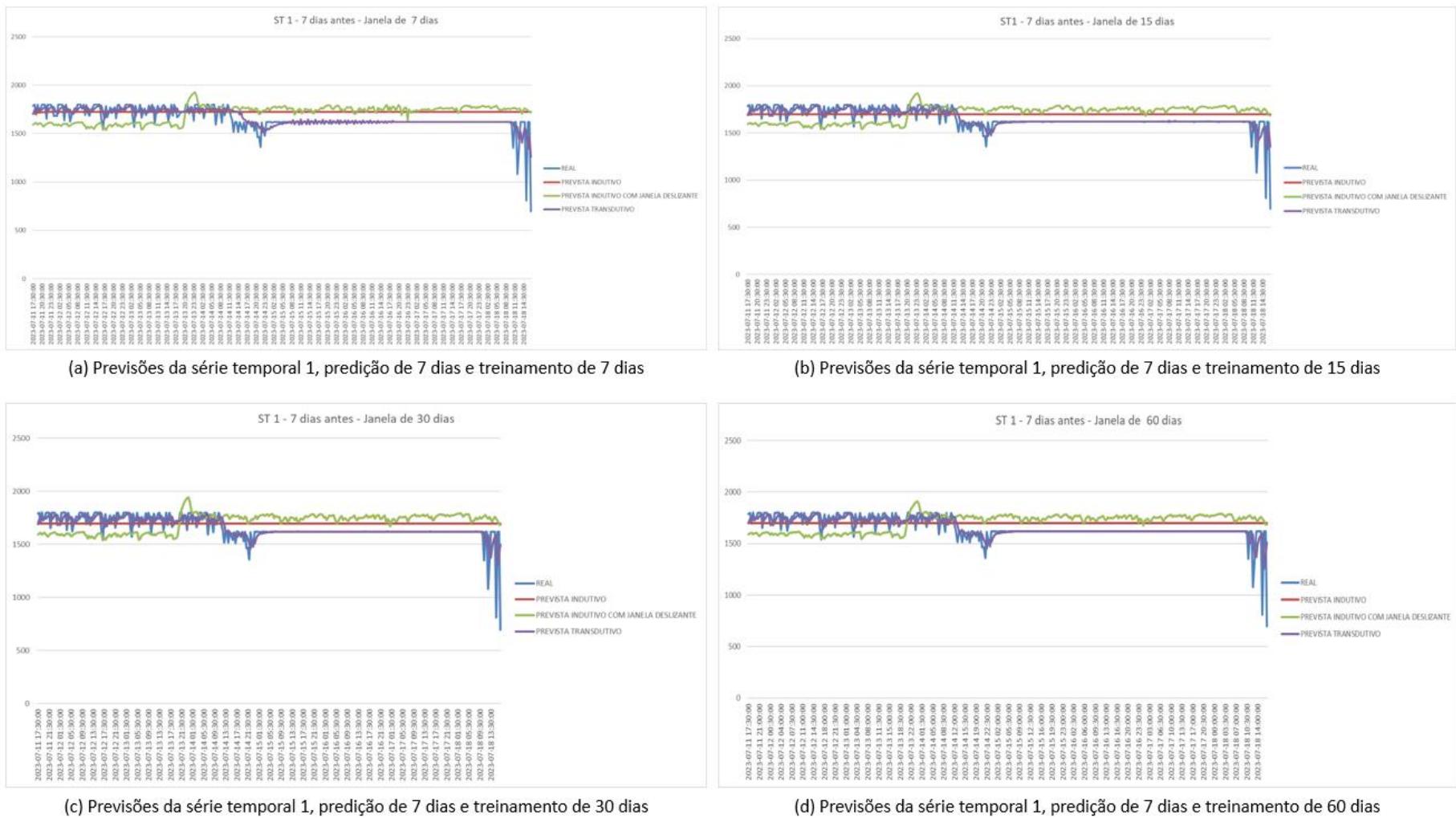


Figura 31 – Previsão da série temporal 1, predição de 7 dias antes do evento anômalo.

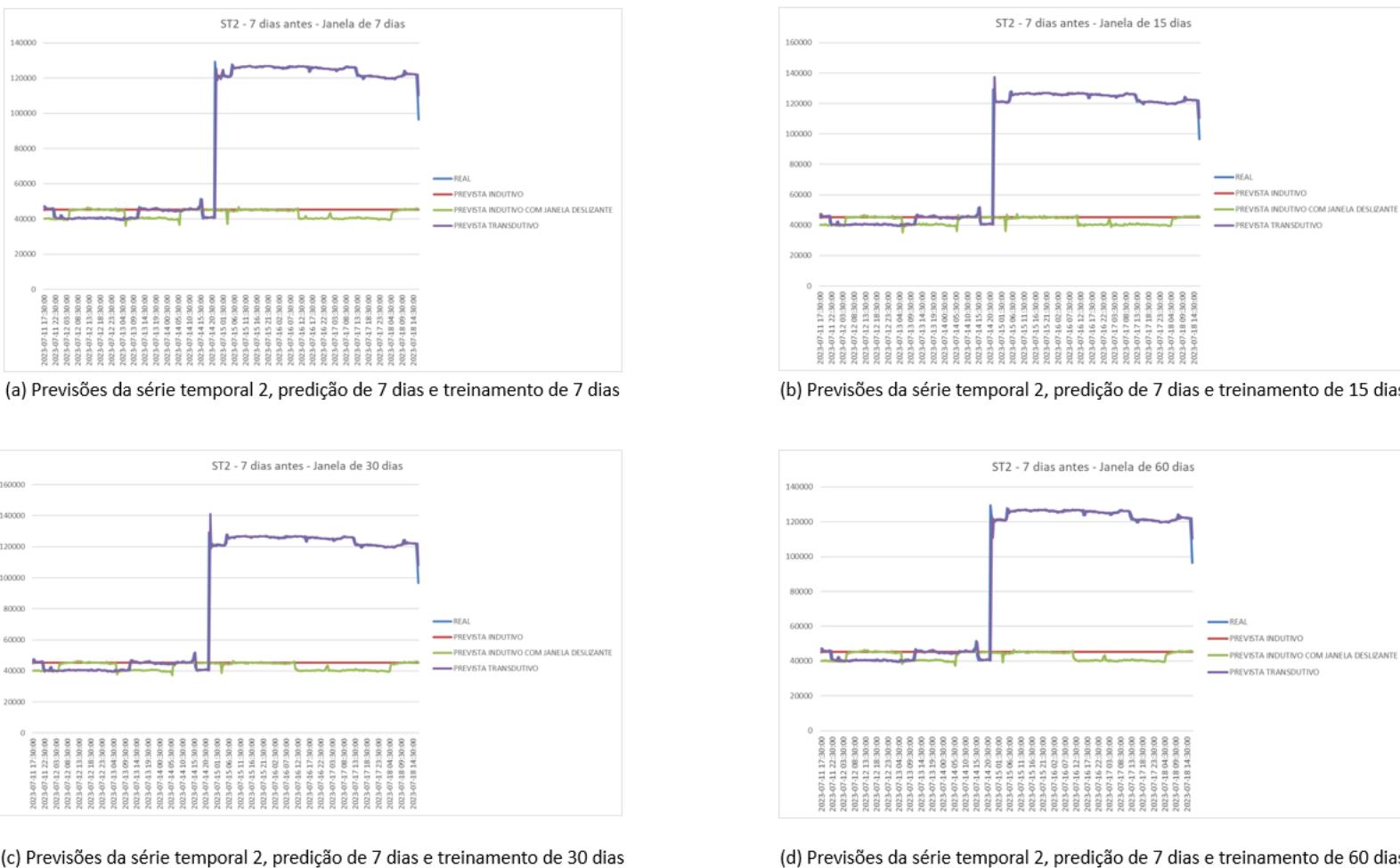
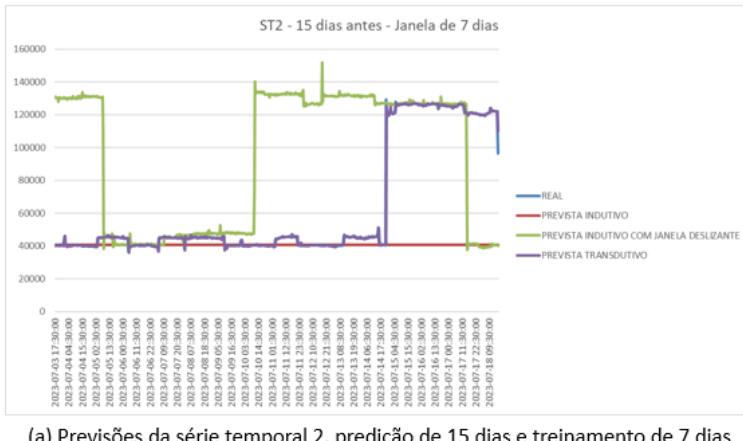


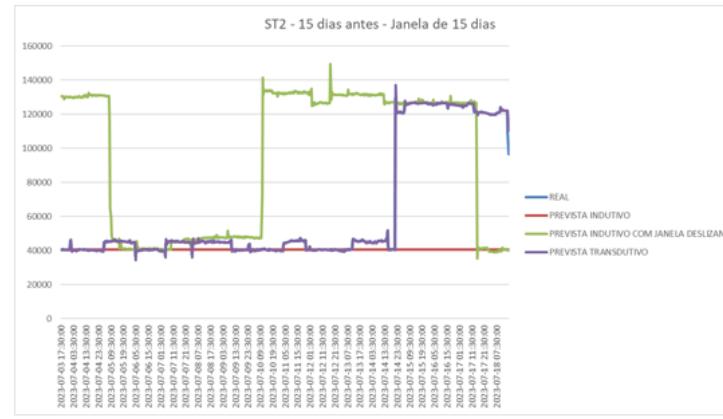
Figura 32 – Previsão da série temporal 1, predição de 7 dias antes do evento anômalo.



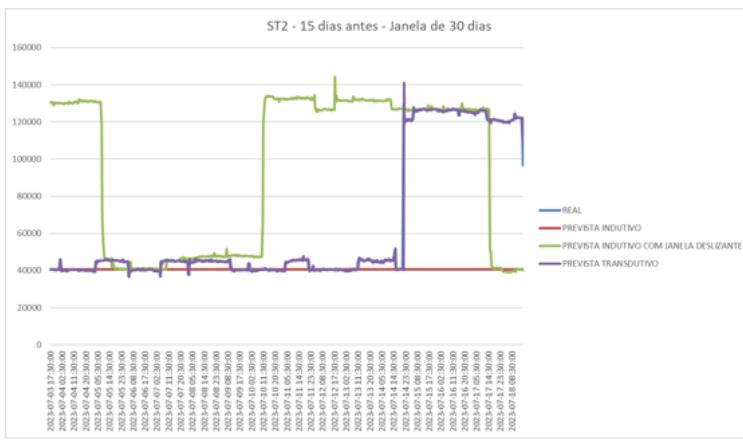
Figura 33 – Previsão da série temporal 1, predição de 15 dias antes do evento anômalo.



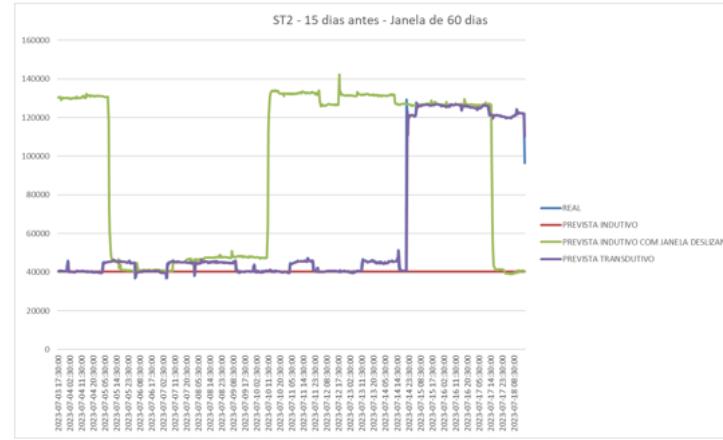
(a) Previsões da série temporal 2, predição de 15 dias e treinamento de 7 dias



(b) Previsões da série temporal 2, predição de 15 dias e treinamento de 15 dias



(c) Previsões da série temporal 2, predição de 15 dias e treinamento de 30 dias



(d) Previsões da série temporal 2, predição de 15 dias e treinamento de 60 dias

Figura 34 – Previsão da série temporal 2, predição de 15 dias antes do evento anômalo.

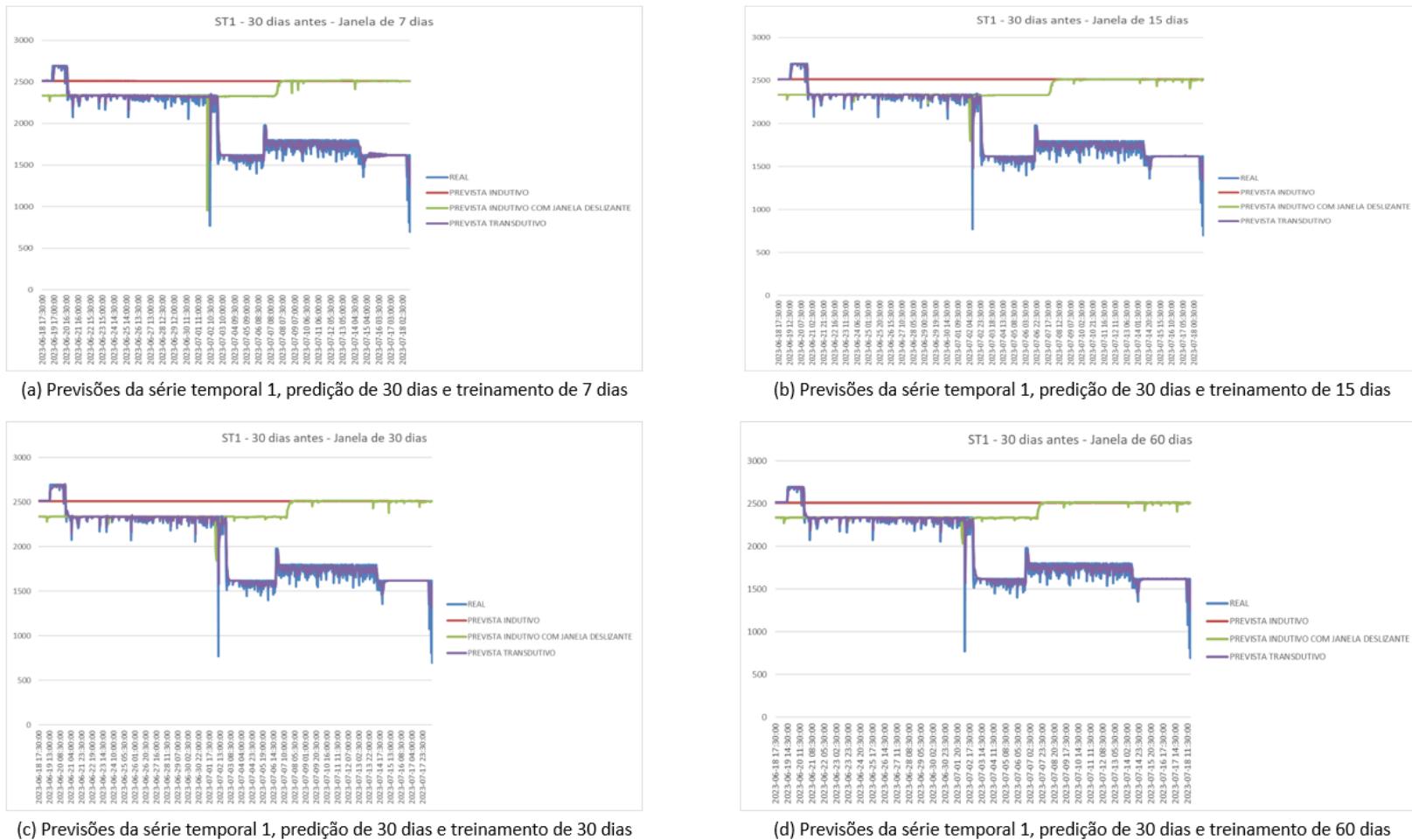


Figura 35 – Previsão da série temporal 1, predição de 30 dias antes do evento anômalo.

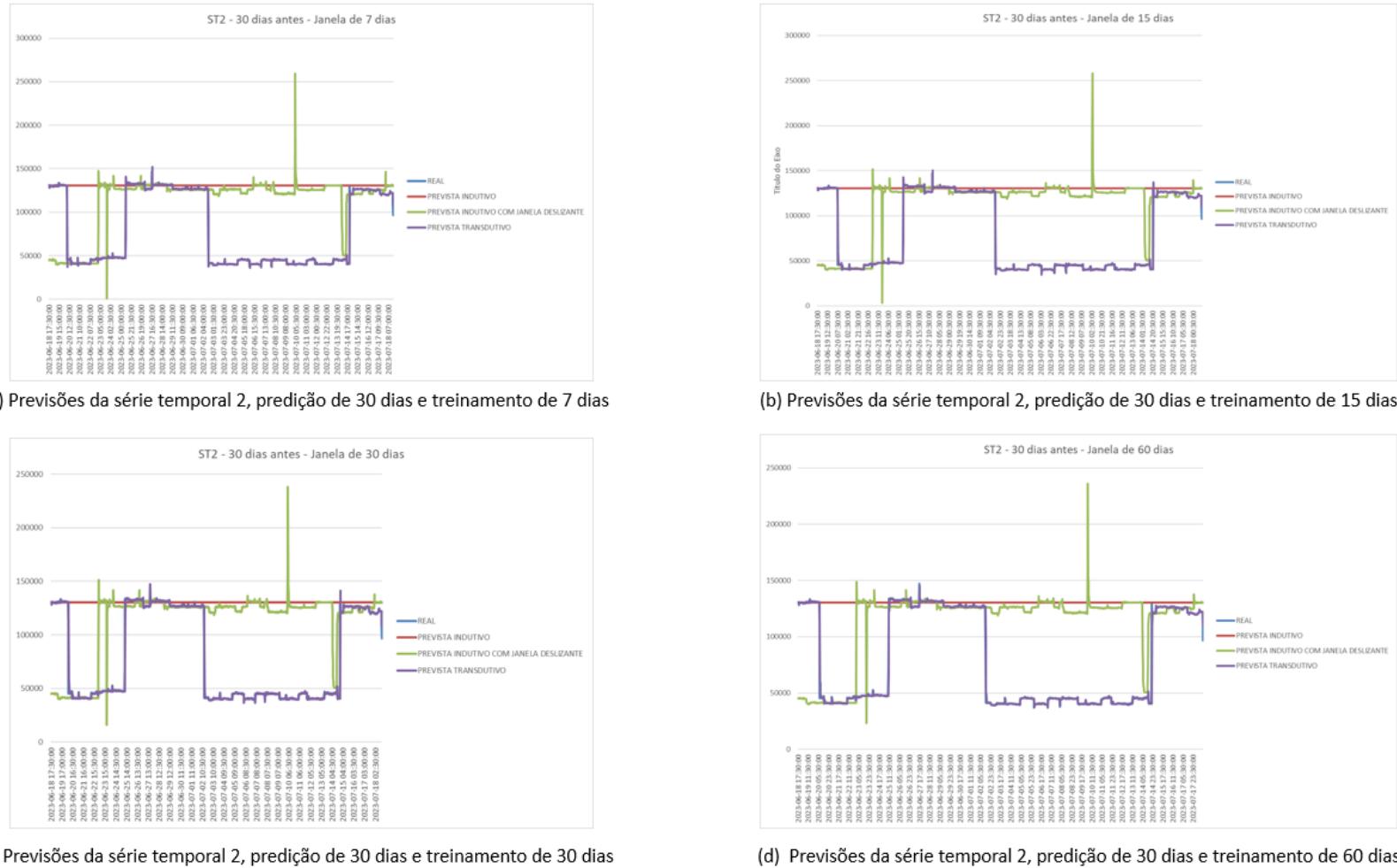


Figura 36 – Previsão da série temporal 2, predição de 30 dias antes do evento anômalo.

Na série temporal 1, o período de 7 dias de previsões coincide com um momento de maior estabilidade da série, consequentemente o valor indutivo está mais próximo do valor real da série, apesar de visivelmente ter resultado muito abaixo da abordagem transdutiva. Outro ponto interessante a se observar, é que a janela de tempo para o treinamento teve pouca influência no perfil dos gráficos obtidos.

Na série temporal 2 ocorre uma mudança de tendência no meio do intervalo de 7 dias, o que torna a abordagem indutiva extremamente imprecisa. Neste cenário, a abordagem transdutiva continua tendo um excelente desempenho, já que se adapta rapidamente a nova tendência. De maneira semelhante o tamanho da janela de dados interfere pouco nos resultados obtidos.

Aumentar a quantidade de dias de previsão resultou em uma piora significativa do método indutivo com janela deslizante, no caso da série temporal 1. Quanto a série temporal 2, o mesmo método tem resultados aceitáveis em determinados períodos, no entanto no geral o seu desempenho não é apropriado para o problema proposto.

Em ambas as séries temporais as previsões indutivas coincidem com alguns intervalos do valor real, no entanto nas mudanças de tendência a diferença torna-se muito grande. A abordagem transdutiva consegue acompanhar essas mudanças já que o treinamento ocorre mais rapidamente.

5.3 Considerações Finais

Com base nos resultados obtidos é nítido que a abordagem transdutiva tem resultados muito superiores para o problema proposto. Além disso também é possível concluir que quanto menor a quantidade de previsões a serem realizadas mais preciso será o valor encontrado. Sobre a janela de dados utilizados para o treinamento, o tamanho desta teve pouca interferência, considerando que os resultados encontrados são similares. Este ponto é levado em consideração no próximo capítulo, que é sobre a detecção de anomalias em si. A abordagem indutiva permite maior generalização dos modelos, no entanto para o problema proposto a especialização é necessária para obter melhores resultados, lembrando que cada série temporal é única.

A abordagem transdutiva é mais custosa do ponto de vista computacional, considerando a grande quantidade de treinamentos a serem realizados a cada período. No entanto contornamos o seu impacto implementando a camada de filtro, tornando assim

possível sua utilização. Logo, cada um dos métodos de detecção de anomalia abordados no próximo capítulo leva em consideração o tempo de treinamento como uma das métricas de desempenho.

Durante o desenvolvimento deste trabalho consideramos implementar algoritmos vistos como o estado da arte para a previsão de séries temporais, como o N-BEATS, proposto por Oreshkin, Carpow, Chapados e Bengio (2019) e o N-HITS proposto por Challu, Olivares, Oreshkin, Ramirez, Canseco e Dubrawski (2023). No entanto, estes métodos são da classe de deep learning, e sabidamente tem um tempo de treinamento considerável, que pode facilmente ultrapassar a periodicidade de atualização das séries temporais (30 minutos). Por este motivo estes não foram implementados e estão fora do escopo deste trabalho.

6 Detecção de Anomalias

Neste capítulo iremos abordar a detecção de anomalia em si. Para tal, como abordado no capítulo 2, utilizaremos métodos da família de detecção de anomalia e métodos estatísticos. A Tabela 9 apresenta os métodos implementados, todos na linguagem Python, assim como a família no qual esses pertencem. A Figura 37 apresenta a distribuição das famílias utilizadas neste trabalho.

Tabela 9 – Métodos utilizados neste trabalho para detecção de anomalia.

| Método | Classe | Dimensionality | Família |
|------------------|-------------------|----------------|----------------|
| Isolation Forest | Outlier Detection | multivariate | trees |
| EIF | Outlier Detection | multivariate | trees |
| CBLOF | Outlier Detection | multivariate | distance |
| Subsequence IF | Outlier Detection | univariate | trees |
| Subsequence LOF | Outlier Detection | univariate | distance |
| COPOD | Outlier Detection | multivariate | distribution |
| IF-LOF | Outlier Detection | multivariate | trees |
| COF | Outlier Detection | multivariate | distance |
| LOF | Outlier Detection | multivariate | distance |
| EWMA | Statistics | univariate | forecasting |
| SARIMA | Statistics | univariate | forecasting |
| PCI | Statistics | univariate | reconstruction |
| pEWMA | Statistics | univariate | forecasting |
| EWMA-STR | Statistics | univariate | forecasting |
| Holt-Winter's | Statistics | univariate | forecasting |
| ARIMA | Statistics | univariate | forecasting |
| DSPOT | Statistics | univariate | distribution |

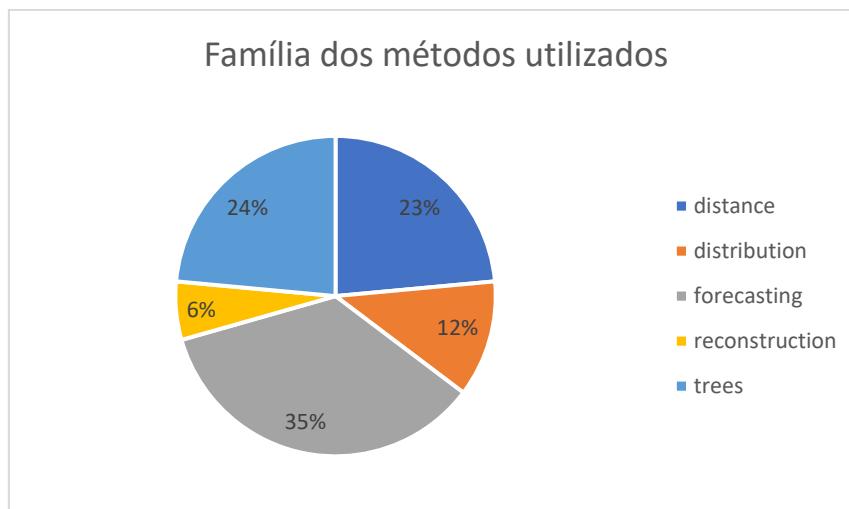


Figura 37 – Distribuição por família dos métodos utilização.

6.1 Experimentos

Todos os experimentos foram realizados em um notebook, que no momento dos testes eram dedicados. Cada teste foi repetido uma segunda vez com o objetivo de ter a média dos tempos. Para garantir que a repetibilidade dos resultados da detecção o gerador de números randômicos foi fixado (seed). As características do hardware utilizado foram:

- Processador i7-11800H de 11° geração
- 16 GB de memória ram DDR5
- SSD de 1TB
- Sistema Operacional Windows 11 Home

Utilizar uma quantidade menor de dados pode gerar vantagens pensando no problema proposto, com um menor tempo de treinamento e diminuir a probabilidade de mudança de tendência no intervalo analisado. Por este motivo foram testadas várias janelas de dados.

- **Janela de dados para treinamento:** 8, 16, 24, 48 ,120, 360 e 720 horas. Como a periodicidade é de 30 minutos, estes intervalos significam 16, 32, 48, 96, 240, 720 e 1440 medições para cada série temporal. Esta janela sempre é a imediatamente anterior ao ponto a ser analisado (abordagem transdutiva).

Como mencionado na introdução deste trabalho, eventos de maior criticidade são percebidos rapidamente, no entanto eventos menores, principalmente em uma rede em implantação, costumam não ser identificados. Por este motivo, o time técnico da TIM BR, no intervalo da coleta de dados (ano de 2023), nos informou apenas 1 evento anômalo, no DRRJO1, elemento do tipo NRF. Para medir a assertividade dos métodos realizamos análise gráfica e classificamos manualmente apenas o ponto informado (dia e horário) como falha ou não.

Ao analisar graficamente as séries fornecidas percebemos um aspecto importantes. No caso do DRRJO1 havia ocorrido um evento anterior no mesmo dia, e em sequência a falha ficou intermitente. Como nosso objetivo é identificar o primeiro evento anômalo, incluímos este evento nos testes.

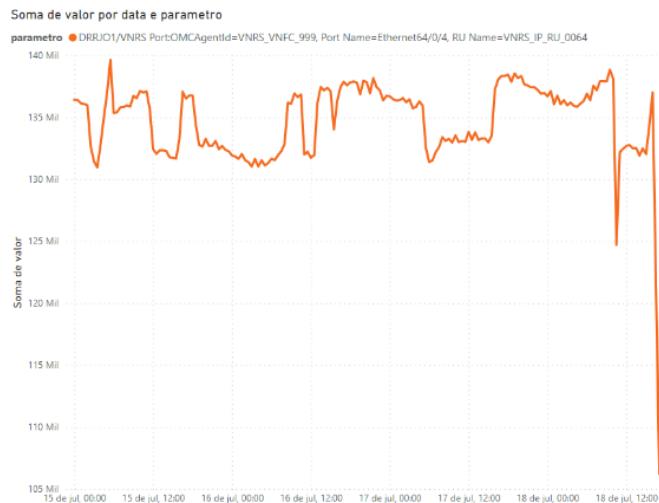


Figura 38 – Exemplos de falhas DRRJO1.

Assim chegamos aos números abaixo:

- **DRRJO1 – Ponto Informado:** 40 séries com falhas em um conjunto de 2399 no dia 18/07/23 às 17:00hs.
- **DRRJO1 – Primeiro evento:** 32 séries com falhas em um conjunto de 2399 no dia 18/07/23 às 11:00hs.

Como temos apenas duas falhas confirmadas, houve uma tentativa de percorrer as séries temporais em busca de outros eventos anômalos e em sequência informar a operadora os eventos anômalos identificados, confirmando assim a veracidade dos resultados obtidos. Para tanto escolhemos dias randômicos no ano de 2023 e os feriados nacionais do mesmo ano. No entanto, não existe o registro de falhas menores, consequentemente não nos foi confirmado se os resultados encontrados eram de fato anômalos ou não. Por este motivo estes resultados não serão apresentados.

Sobre a parametrização e threshold para cada método testado, estes são hiperparâmetros que devem ser configurados conforme o contexto da rede em estudo. No nosso caso buscamos estes valores através de experimentação, considerando que eventos anômalos em redes Core normalmente geram uma variação considerável na amplitude das séries temporais. No caso dos métodos de detecção de outliers que utilizam distância, por exemplo, o score próximo a 1 indica normalidade. No problema proposto o valor setado foi próximo a 2. No caso de métodos de previsão, como EWMA, variação de mais

de 15% em relação ao valor predito. Uma observação sobre o ARIMA, o mesmo foi executado de duas formas diferentes, utilizando o método auto, mais lento mas que busca otimizar os parâmetros de regressão, diferenciação e média móvel (p,d,q) e outra com este valor padrão para todas as séries (1,1,1).

Os testes foram executados nas séries temporais que não foram filtradas pelo algoritmo de constantes da etapa 1 do filtro apresentado no capítulo 4. As demais camadas serão avaliadas após a detecção de anomalia, no contexto do tempo de execução. Esta decisão tem como objetivo diminuir o tempo de execução dos métodos de maneira assertiva (séries sem variabilidade não podem ser anômalas).

Para cada um dos testes realizado foi gerado uma matriz de confusão e as métricas utilizadas para avaliar os resultados foram precisão, recall, F1-Score e acurácia. Também será apresentado o tempo aferido para a detecção (somando-se o tempo de treinamento). Os resultados obtidos serão apresentados a seguir.

6.2 Resultados

Primeiro serão apresentados os resultados de desempenho dos modelos utilizados para detecção de anomalia. Em sequência será apresentado os resultados do tempo de execução, sendo este o momento que será debatido a assertividade da utilização dos filtros.

6.2.1 Desempenho dos modelos

As tabelas 9 e 10 apresentam os resultados obtidos por método, descrevendo os resultados por ponto dos 234 testados (saída do algoritmo de constante). As figuras 39, 40, 41 e 42 apresentam os resultados de acurácia, precisão, recall e F1 Score por método testado, subdivididos em detecção de outliers e métodos estatísticos. As figuras 43, 44, 45 e 46 apresentam os resultados médios para as duas áreas de algoritmos testados.

Tabela 10 – Resultados obtidos por método e intervalo de treinamento – Ponto Original.

| METODO | ÁREA | INTERVALO | TP | TN | FP | FN | METODO | ÁREA | INTERVALO | TP | TN | FP | FN |
|------------------|-------------------|-----------|----|-----|----|----|--------------|------------|-----------|----|-----|-----|----|
| lof | Outlier Detection | 8 | 17 | 104 | 23 | 23 | auto_arima | Statistics | 8 | 28 | 88 | 39 | 12 |
| lof | Outlier Detection | 16 | 33 | 117 | 19 | 7 | auto_arima | Statistics | 16 | 37 | 96 | 40 | 3 |
| lof | Outlier Detection | 24 | 40 | 124 | 14 | 0 | auto_arima | Statistics | 24 | 37 | 94 | 44 | 3 |
| lof | Outlier Detection | 48 | 40 | 132 | 6 | 0 | auto_arima | Statistics | 48 | 37 | 94 | 44 | 3 |
| lof | Outlier Detection | 120 | 40 | 145 | 5 | 0 | auto_arima | Statistics | 120 | 36 | 115 | 35 | 4 |
| lof | Outlier Detection | 240 | 40 | 149 | 1 | 0 | auto_arima | Statistics | 240 | 37 | 113 | 37 | 3 |
| lof | Outlier Detection | 360 | 40 | 159 | 0 | 0 | auto_arima | Statistics | 360 | 37 | 121 | 38 | 3 |
| lof | Outlier Detection | 720 | 40 | 194 | 0 | 0 | auto_arima | Statistics | 720 | 37 | 160 | 34 | 3 |
| cblof | Outlier Detection | 8 | 0 | 126 | 1 | 40 | arima | Statistics | 8 | 28 | 90 | 37 | 12 |
| cblof | Outlier Detection | 16 | 7 | 135 | 1 | 33 | arima | Statistics | 16 | 40 | 97 | 39 | 0 |
| cblof | Outlier Detection | 24 | 7 | 137 | 1 | 33 | arima | Statistics | 24 | 40 | 98 | 40 | 0 |
| cblof | Outlier Detection | 48 | 33 | 136 | 2 | 7 | arima | Statistics | 48 | 40 | 93 | 45 | 0 |
| cblof | Outlier Detection | 120 | 39 | 148 | 2 | 1 | arima | Statistics | 120 | 39 | 114 | 36 | 1 |
| cblof | Outlier Detection | 240 | 39 | 146 | 4 | 1 | arima | Statistics | 240 | 40 | 113 | 37 | 0 |
| cblof | Outlier Detection | 360 | 40 | 155 | 4 | 0 | arima | Statistics | 360 | 40 | 121 | 38 | 0 |
| cblof | Outlier Detection | 720 | 40 | 189 | 5 | 0 | arima | Statistics | 720 | 40 | 159 | 35 | 0 |
| cof | Outlier Detection | 8 | 40 | 120 | 7 | 0 | dspot | Statistics | 8 | 29 | 97 | 30 | 11 |
| cof | Outlier Detection | 16 | 40 | 135 | 1 | 0 | dspot | Statistics | 16 | 29 | 110 | 26 | 11 |
| cof | Outlier Detection | 24 | 40 | 136 | 2 | 0 | dspot | Statistics | 24 | 31 | 114 | 24 | 9 |
| cof | Outlier Detection | 48 | 40 | 138 | 0 | 0 | dspot | Statistics | 48 | 40 | 108 | 30 | 0 |
| cof | Outlier Detection | 120 | 40 | 150 | 0 | 0 | dspot | Statistics | 120 | 40 | 119 | 31 | 0 |
| cof | Outlier Detection | 240 | 40 | 150 | 0 | 0 | dspot | Statistics | 240 | 40 | 122 | 28 | 0 |
| cof | Outlier Detection | 360 | 40 | 159 | 0 | 0 | dspot | Statistics | 360 | 40 | 129 | 30 | 0 |
| cof | Outlier Detection | 720 | 40 | 194 | 0 | 0 | dspot | Statistics | 720 | 40 | 141 | 53 | 0 |
| subsequence_lof | Outlier Detection | 8 | 9 | 96 | 31 | 31 | ewma_str | Statistics | 8 | 40 | 52 | 75 | 0 |
| subsequence_lof | Outlier Detection | 16 | 19 | 114 | 22 | 21 | ewma_str | Statistics | 16 | 40 | 52 | 84 | 0 |
| subsequence_lof | Outlier Detection | 24 | 38 | 97 | 41 | 2 | ewma_str | Statistics | 24 | 40 | 52 | 86 | 0 |
| subsequence_lof | Outlier Detection | 48 | 35 | 96 | 42 | 5 | ewma_str | Statistics | 48 | 40 | 52 | 86 | 0 |
| subsequence_lof | Outlier Detection | 120 | 36 | 115 | 35 | 4 | ewma_str | Statistics | 120 | 40 | 54 | 96 | 0 |
| subsequence_lof | Outlier Detection | 240 | 36 | 111 | 39 | 4 | ewma_str | Statistics | 240 | 40 | 54 | 96 | 0 |
| subsequence_lof | Outlier Detection | 360 | 36 | 122 | 37 | 4 | ewma_str | Statistics | 360 | 40 | 57 | 102 | 0 |
| subsequence_lof | Outlier Detection | 720 | 36 | 169 | 25 | 4 | ewma_str | Statistics | 720 | 40 | 86 | 108 | 0 |
| isolation_forest | Outlier Detection | 8 | 10 | 107 | 20 | 30 | pewma | Statistics | 8 | 40 | 53 | 74 | 0 |
| isolation_forest | Outlier Detection | 16 | 36 | 97 | 39 | 4 | pewma | Statistics | 16 | 40 | 53 | 83 | 0 |
| isolation_forest | Outlier Detection | 24 | 40 | 102 | 36 | 0 | pewma | Statistics | 24 | 40 | 53 | 85 | 0 |
| isolation_forest | Outlier Detection | 48 | 40 | 100 | 38 | 0 | pewma | Statistics | 48 | 40 | 53 | 85 | 0 |
| isolation_forest | Outlier Detection | 120 | 36 | 106 | 44 | 4 | pewma | Statistics | 120 | 40 | 55 | 95 | 0 |
| isolation_forest | Outlier Detection | 240 | 38 | 106 | 44 | 2 | pewma | Statistics | 240 | 40 | 55 | 95 | 0 |
| isolation_forest | Outlier Detection | 360 | 40 | 116 | 43 | 0 | pewma | Statistics | 360 | 40 | 58 | 101 | 0 |
| isolation_forest | Outlier Detection | 720 | 40 | 154 | 40 | 0 | pewma | Statistics | 720 | 40 | 93 | 101 | 0 |
| eif | Outlier Detection | 8 | 15 | 126 | 1 | 25 | ewma | Statistics | 8 | 40 | 54 | 73 | 0 |
| eif | Outlier Detection | 16 | 28 | 133 | 3 | 12 | ewma | Statistics | 16 | 40 | 54 | 82 | 0 |
| eif | Outlier Detection | 24 | 33 | 133 | 5 | 7 | ewma | Statistics | 24 | 40 | 54 | 84 | 0 |
| eif | Outlier Detection | 48 | 40 | 132 | 6 | 0 | ewma | Statistics | 48 | 40 | 54 | 84 | 0 |
| eif | Outlier Detection | 120 | 40 | 145 | 5 | 0 | ewma | Statistics | 120 | 40 | 56 | 94 | 0 |
| eif | Outlier Detection | 240 | 40 | 150 | 0 | 0 | ewma | Statistics | 240 | 40 | 56 | 94 | 0 |
| eif | Outlier Detection | 360 | 34 | 159 | 0 | 6 | ewma | Statistics | 360 | 40 | 59 | 100 | 0 |
| eif | Outlier Detection | 720 | 29 | 193 | 1 | 11 | ewma | Statistics | 720 | 40 | 65 | 129 | 0 |
| subsequence_if | Outlier Detection | 8 | 9 | 104 | 23 | 31 | pci | Statistics | 8 | 28 | 66 | 61 | 12 |
| subsequence_if | Outlier Detection | 16 | 6 | 89 | 47 | 34 | pci | Statistics | 16 | 28 | 76 | 60 | 12 |
| subsequence_if | Outlier Detection | 24 | 27 | 78 | 60 | 13 | pci | Statistics | 24 | 28 | 78 | 60 | 12 |
| subsequence_if | Outlier Detection | 48 | 37 | 80 | 58 | 3 | pci | Statistics | 48 | 28 | 78 | 60 | 12 |
| subsequence_if | Outlier Detection | 120 | 14 | 104 | 46 | 26 | pci | Statistics | 120 | 28 | 90 | 60 | 12 |
| subsequence_if | Outlier Detection | 240 | 23 | 103 | 47 | 17 | pci | Statistics | 240 | 28 | 90 | 60 | 12 |
| subsequence_if | Outlier Detection | 360 | 24 | 115 | 44 | 16 | pci | Statistics | 360 | 28 | 97 | 62 | 12 |
| subsequence_if | Outlier Detection | 720 | 18 | 157 | 37 | 22 | pci | Statistics | 720 | 28 | 129 | 65 | 12 |
| if_lof | Outlier Detection | 8 | 8 | 123 | 4 | 32 | sarima | Statistics | 8 | 29 | 34 | 93 | 11 |
| if_lof | Outlier Detection | 16 | 33 | 122 | 14 | 7 | sarima | Statistics | 16 | 40 | 48 | 88 | 0 |
| if_lof | Outlier Detection | 24 | 40 | 126 | 12 | 0 | sarima | Statistics | 24 | 0 | 138 | 0 | 40 |
| if_lof | Outlier Detection | 48 | 40 | 134 | 4 | 0 | sarima | Statistics | 48 | 40 | 50 | 88 | 0 |
| if_lof | Outlier Detection | 120 | 40 | 149 | 1 | 0 | sarima | Statistics | 120 | 40 | 67 | 83 | 0 |
| if_lof | Outlier Detection | 240 | 40 | 150 | 0 | 0 | sarima | Statistics | 240 | 40 | 66 | 84 | 0 |
| if_lof | Outlier Detection | 360 | 40 | 159 | 0 | 0 | sarima | Statistics | 360 | 40 | 70 | 89 | 0 |
| if_lof | Outlier Detection | 720 | 40 | 193 | 1 | 0 | sarima | Statistics | 720 | 40 | 99 | 95 | 0 |
| copod | Outlier Detection | 8 | 0 | 127 | 0 | 40 | holt_winters | Statistics | 8 | 37 | 107 | 20 | 3 |
| copod | Outlier Detection | 16 | 0 | 136 | 0 | 40 | holt_winters | Statistics | 16 | 35 | 116 | 20 | 5 |
| copod | Outlier Detection | 24 | 0 | 138 | 0 | 40 | holt_winters | Statistics | 24 | 35 | 118 | 20 | 5 |
| copod | Outlier Detection | 48 | 0 | 138 | 0 | 40 | holt_winters | Statistics | 48 | 36 | 120 | 18 | 4 |
| copod | Outlier Detection | 120 | 22 | 135 | 15 | 18 | holt_winters | Statistics | 120 | 35 | 128 | 22 | 5 |
| copod | Outlier Detection | 240 | 32 | 130 | 20 | 8 | holt_winters | Statistics | 240 | 34 | 128 | 22 | 6 |
| copod | Outlier Detection | 360 | 37 | 130 | 29 | 3 | holt_winters | Statistics | 360 | 36 | 137 | 22 | 4 |
| copod | Outlier Detection | 720 | 35 | 151 | 43 | 5 | holt_winters | Statistics | 720 | 35 | 174 | 20 | 5 |

Tabela 11 – Resultados obtidos por método e intervalo de treinamento – Primeiro Evento.

| METODO | Área | INTERVALO | TP | TN | FP | FN |
|------------------|-------------------|-----------|----|-----|-----|----|
| lof | Outlier Detection | 8 | 32 | 97 | 35 | 0 |
| lof | Outlier Detection | 16 | 32 | 128 | 15 | 0 |
| lof | Outlier Detection | 24 | 32 | 132 | 14 | 0 |
| lof | Outlier Detection | 48 | 32 | 137 | 9 | 0 |
| lof | Outlier Detection | 120 | 32 | 152 | 6 | 0 |
| lof | Outlier Detection | 240 | 32 | 150 | 8 | 0 |
| lof | Outlier Detection | 360 | 32 | 167 | 0 | 0 |
| lof | Outlier Detection | 720 | 32 | 200 | 2 | 0 |
| isolation_forest | Outlier Detection | 8 | 32 | 117 | 15 | 0 |
| isolation_forest | Outlier Detection | 16 | 32 | 108 | 35 | 0 |
| isolation_forest | Outlier Detection | 24 | 32 | 116 | 30 | 0 |
| isolation_forest | Outlier Detection | 48 | 32 | 117 | 29 | 0 |
| isolation_forest | Outlier Detection | 120 | 32 | 131 | 27 | 0 |
| isolation_forest | Outlier Detection | 240 | 32 | 135 | 23 | 0 |
| isolation_forest | Outlier Detection | 360 | 32 | 148 | 19 | 0 |
| isolation_forest | Outlier Detection | 720 | 32 | 184 | 18 | 0 |
| eif | Outlier Detection | 8 | 32 | 123 | 9 | 0 |
| eif | Outlier Detection | 16 | 32 | 134 | 9 | 0 |
| eif | Outlier Detection | 24 | 32 | 140 | 6 | 0 |
| eif | Outlier Detection | 48 | 32 | 139 | 7 | 0 |
| eif | Outlier Detection | 120 | 32 | 151 | 7 | 0 |
| eif | Outlier Detection | 240 | 32 | 154 | 4 | 0 |
| eif | Outlier Detection | 360 | 20 | 165 | 2 | 12 |
| eif | Outlier Detection | 720 | 8 | 201 | 1 | 24 |
| cblof | Outlier Detection | 8 | 0 | 131 | 1 | 32 |
| cblof | Outlier Detection | 16 | 12 | 142 | 1 | 20 |
| cblof | Outlier Detection | 24 | 15 | 142 | 4 | 17 |
| cblof | Outlier Detection | 48 | 26 | 142 | 4 | 6 |
| cblof | Outlier Detection | 120 | 26 | 152 | 6 | 6 |
| cblof | Outlier Detection | 240 | 26 | 149 | 9 | 6 |
| cblof | Outlier Detection | 360 | 26 | 157 | 10 | 6 |
| cblof | Outlier Detection | 720 | 26 | 192 | 10 | 6 |
| subsequence_if | Outlier Detection | 8 | 29 | 56 | 76 | 3 |
| subsequence_if | Outlier Detection | 16 | 22 | 81 | 62 | 10 |
| subsequence_if | Outlier Detection | 24 | 27 | 99 | 47 | 5 |
| subsequence_if | Outlier Detection | 48 | 17 | 122 | 24 | 15 |
| subsequence_if | Outlier Detection | 120 | 2 | 138 | 20 | 30 |
| subsequence_if | Outlier Detection | 240 | 2 | 131 | 27 | 30 |
| subsequence_if | Outlier Detection | 360 | 0 | 136 | 31 | 32 |
| subsequence_if | Outlier Detection | 720 | 0 | 167 | 35 | 32 |
| subsequence_lof | Outlier Detection | 8 | 32 | 78 | 54 | 0 |
| subsequence_lof | Outlier Detection | 16 | 32 | 102 | 41 | 0 |
| subsequence_lof | Outlier Detection | 24 | 32 | 110 | 36 | 0 |
| subsequence_lof | Outlier Detection | 48 | 32 | 119 | 27 | 0 |
| subsequence_lof | Outlier Detection | 120 | 32 | 133 | 25 | 0 |
| subsequence_lof | Outlier Detection | 240 | 32 | 131 | 27 | 0 |
| subsequence_lof | Outlier Detection | 360 | 32 | 139 | 28 | 0 |
| subsequence_lof | Outlier Detection | 720 | 21 | 176 | 26 | 11 |
| copod | Outlier Detection | 8 | 0 | 132 | 0 | 32 |
| copod | Outlier Detection | 16 | 0 | 143 | 0 | 32 |
| copod | Outlier Detection | 24 | 0 | 146 | 0 | 32 |
| copod | Outlier Detection | 48 | 0 | 146 | 0 | 32 |
| copod | Outlier Detection | 120 | 24 | 148 | 10 | 8 |
| copod | Outlier Detection | 240 | 32 | 144 | 14 | 0 |
| copod | Outlier Detection | 360 | 32 | 152 | 15 | 0 |
| copod | Outlier Detection | 720 | 30 | 180 | 22 | 2 |
| if_lof | Outlier Detection | 8 | 32 | 124 | 8 | 0 |
| if_lof | Outlier Detection | 16 | 32 | 129 | 14 | 0 |
| if_lof | Outlier Detection | 24 | 32 | 134 | 12 | 0 |
| if_lof | Outlier Detection | 48 | 32 | 138 | 8 | 0 |
| if_lof | Outlier Detection | 120 | 32 | 152 | 6 | 0 |
| if_lof | Outlier Detection | 240 | 32 | 156 | 2 | 0 |
| if_lof | Outlier Detection | 360 | 26 | 167 | 0 | 6 |
| if_lof | Outlier Detection | 720 | 26 | 200 | 2 | 6 |
| cof | Outlier Detection | 8 | 32 | 122 | 10 | 0 |
| cof | Outlier Detection | 16 | 32 | 135 | 8 | 0 |
| cof | Outlier Detection | 24 | 32 | 139 | 7 | 0 |
| cof | Outlier Detection | 48 | 32 | 146 | 0 | 0 |
| cof | Outlier Detection | 120 | 32 | 158 | 0 | 0 |
| cof | Outlier Detection | 240 | 32 | 158 | 0 | 0 |
| cof | Outlier Detection | 360 | 32 | 167 | 0 | 0 |
| cof | Outlier Detection | 720 | 32 | 202 | 0 | 0 |
| ewma | Statistics | 8 | 21 | 64 | 68 | 11 |
| ewma | Statistics | 16 | 21 | 67 | 76 | 11 |
| ewma | Statistics | 24 | 21 | 68 | 78 | 11 |
| ewma | Statistics | 48 | 21 | 68 | 78 | 11 |
| ewma | Statistics | 120 | 21 | 70 | 88 | 11 |
| ewma | Statistics | 240 | 21 | 70 | 88 | 11 |
| ewma | Statistics | 360 | 21 | 73 | 94 | 11 |
| ewma | Statistics | 720 | 21 | 79 | 123 | 11 |
| pci | Statistics | 8 | 20 | 62 | 70 | 12 |
| pci | Statistics | 16 | 20 | 74 | 69 | 12 |
| pci | Statistics | 24 | 20 | 77 | 69 | 12 |
| pci | Statistics | 48 | 20 | 77 | 69 | 12 |
| pci | Statistics | 120 | 20 | 89 | 69 | 12 |
| pci | Statistics | 240 | 20 | 89 | 69 | 12 |
| pci | Statistics | 360 | 20 | 96 | 71 | 12 |
| pci | Statistics | 720 | 20 | 128 | 74 | 12 |
| pewma | Statistics | 8 | 32 | 57 | 75 | 0 |
| pewma | Statistics | 16 | 32 | 57 | 86 | 0 |
| pewma | Statistics | 24 | 32 | 58 | 88 | 0 |
| pewma | Statistics | 48 | 32 | 58 | 88 | 0 |
| pewma | Statistics | 120 | 32 | 60 | 98 | 0 |
| pewma | Statistics | 240 | 32 | 60 | 98 | 0 |
| pewma | Statistics | 360 | 32 | 63 | 104 | 0 |
| pewma | Statistics | 720 | 32 | 98 | 104 | 0 |
| ewma_str | Statistics | 8 | 32 | 57 | 75 | 0 |
| ewma_str | Statistics | 16 | 32 | 57 | 86 | 0 |
| ewma_str | Statistics | 24 | 32 | 58 | 88 | 0 |
| ewma_str | Statistics | 48 | 32 | 58 | 88 | 0 |
| ewma_str | Statistics | 120 | 32 | 60 | 98 | 0 |
| ewma_str | Statistics | 240 | 32 | 60 | 98 | 0 |
| ewma_str | Statistics | 360 | 32 | 63 | 104 | 0 |
| ewma_str | Statistics | 720 | 32 | 92 | 110 | 0 |
| dspot | Statistics | 8 | 20 | 92 | 40 | 12 |
| dspot | Statistics | 16 | 21 | 110 | 33 | 11 |
| dspot | Statistics | 24 | 25 | 110 | 36 | 7 |
| dspot | Statistics | 48 | 25 | 108 | 38 | 7 |
| dspot | Statistics | 120 | 31 | 120 | 38 | 1 |
| dspot | Statistics | 240 | 31 | 116 | 42 | 1 |
| dspot | Statistics | 360 | 31 | 123 | 44 | 1 |
| dspot | Statistics | 720 | 31 | 134 | 68 | 1 |
| auto_arima | Statistics | 8 | 29 | 83 | 49 | 3 |
| auto_arima | Statistics | 16 | 29 | 101 | 42 | 3 |
| auto_arima | Statistics | 24 | 32 | 101 | 45 | 0 |
| auto_arima | Statistics | 48 | 32 | 109 | 37 | 0 |
| auto_arima | Statistics | 120 | 32 | 128 | 30 | 0 |
| auto_arima | Statistics | 240 | 32 | 130 | 28 | 0 |
| auto_arima | Statistics | 360 | 32 | 145 | 22 | 0 |
| auto_arima | Statistics | 720 | 32 | 178 | 24 | 0 |
| arima | Statistics | 8 | 32 | 80 | 52 | 0 |
| arima | Statistics | 16 | 32 | 89 | 54 | 0 |
| arima | Statistics | 24 | 32 | 98 | 48 | 0 |
| arima | Statistics | 48 | 32 | 102 | 44 | 0 |
| arima | Statistics | 120 | 32 | 133 | 25 | 0 |
| arima | Statistics | 240 | 32 | 133 | 25 | 0 |
| arima | Statistics | 360 | 32 | 143 | 24 | 0 |
| arima | Statistics | 720 | 32 | 179 | 23 | 0 |
| sarima | Statistics | 8 | 31 | 40 | 92 | 1 |
| sarima | Statistics | 16 | 32 | 49 | 94 | 0 |
| sarima | Statistics | 24 | 0 | 146 | 0 | 32 |
| sarima | Statistics | 48 | 32 | 46 | 100 | 0 |
| sarima | Statistics | 120 | 32 | 62 | 96 | 0 |
| sarima | Statistics | 240 | 32 | 59 | 99 | 0 |
| sarima | Statistics | 360 | 32 | 61 | 106 | 0 |
| sarima | Statistics | 720 | 32 | 95 | 107 | 0 |
| holt_winters | Statistics | 8 | 20 | 118 | 14 | 12 |
| holt_winters | Statistics | 16 | 20 | 128 | 15 | 12 |
| holt_winters | Statistics | 24 | 20 | 132 | 14 | 12 |
| holt_winters | Statistics | 48 | 20 | 134 | 12 | 12 |
| holt_winters | Statistics | 120 | 20 | 145 | 13 | 12 |
| holt_winters | Statistics | 240 | 20 | 147 | 11 | 12 |
| holt_winters | Statistics | 360 | 20 | 156 | 11 | 12 |
| holt_winters | Statistics | 720 | 20 | 189 | 13 | 12 |

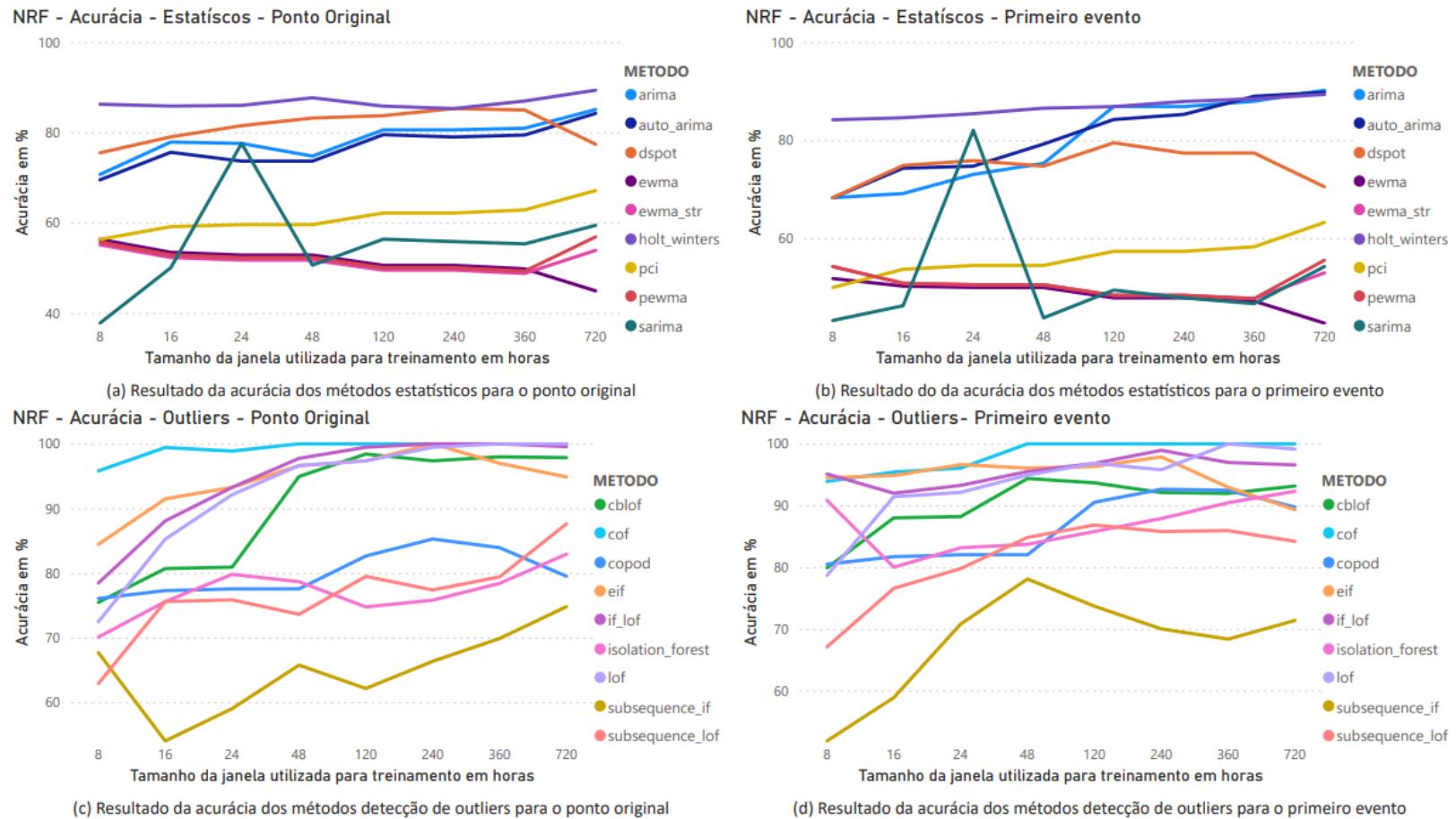


Figura 39 – Resultados da acurácia para os testes no DDRJO1.

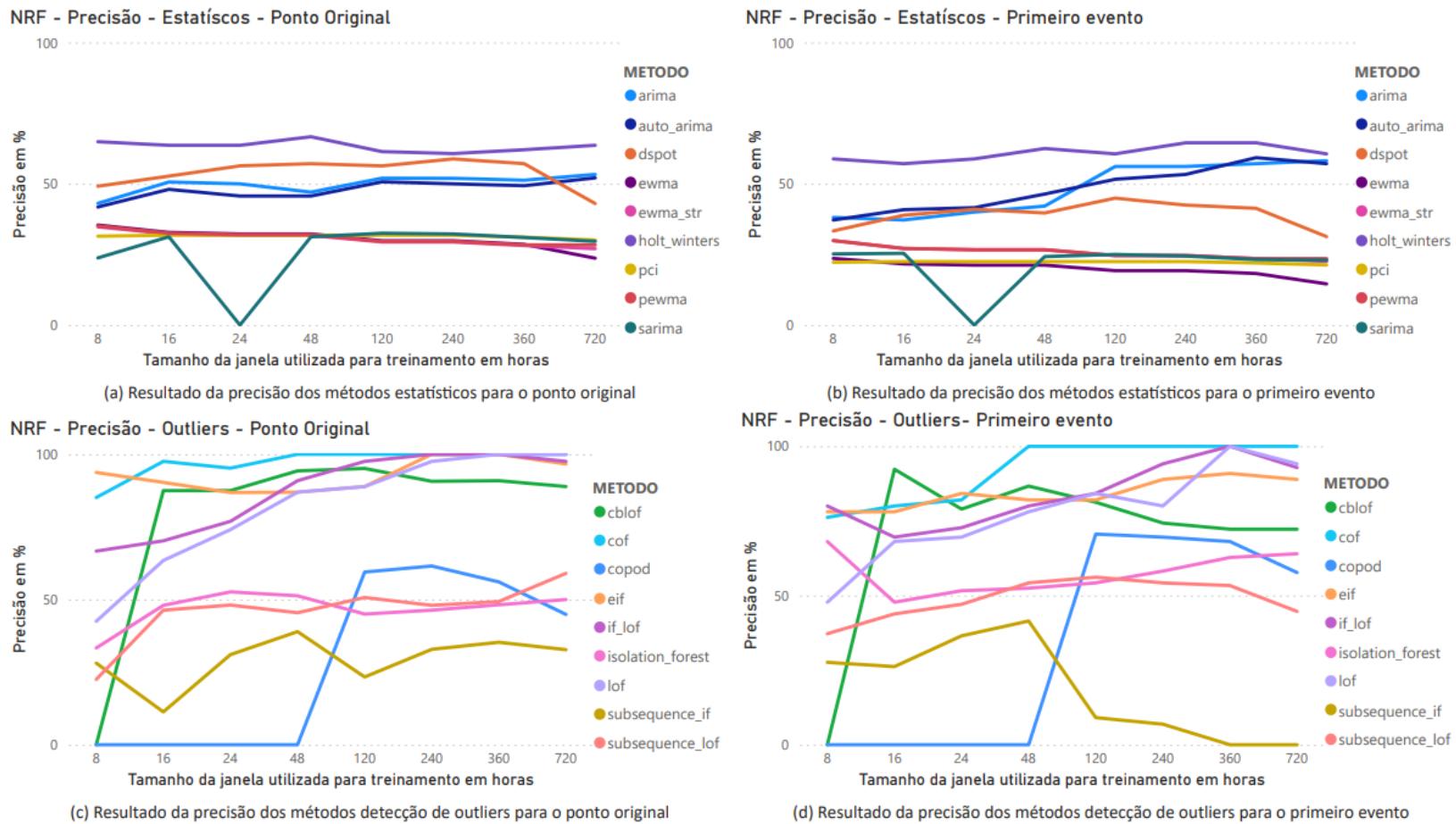


Figura 40 – Resultados da Precisão para os testes no DDRJO1.

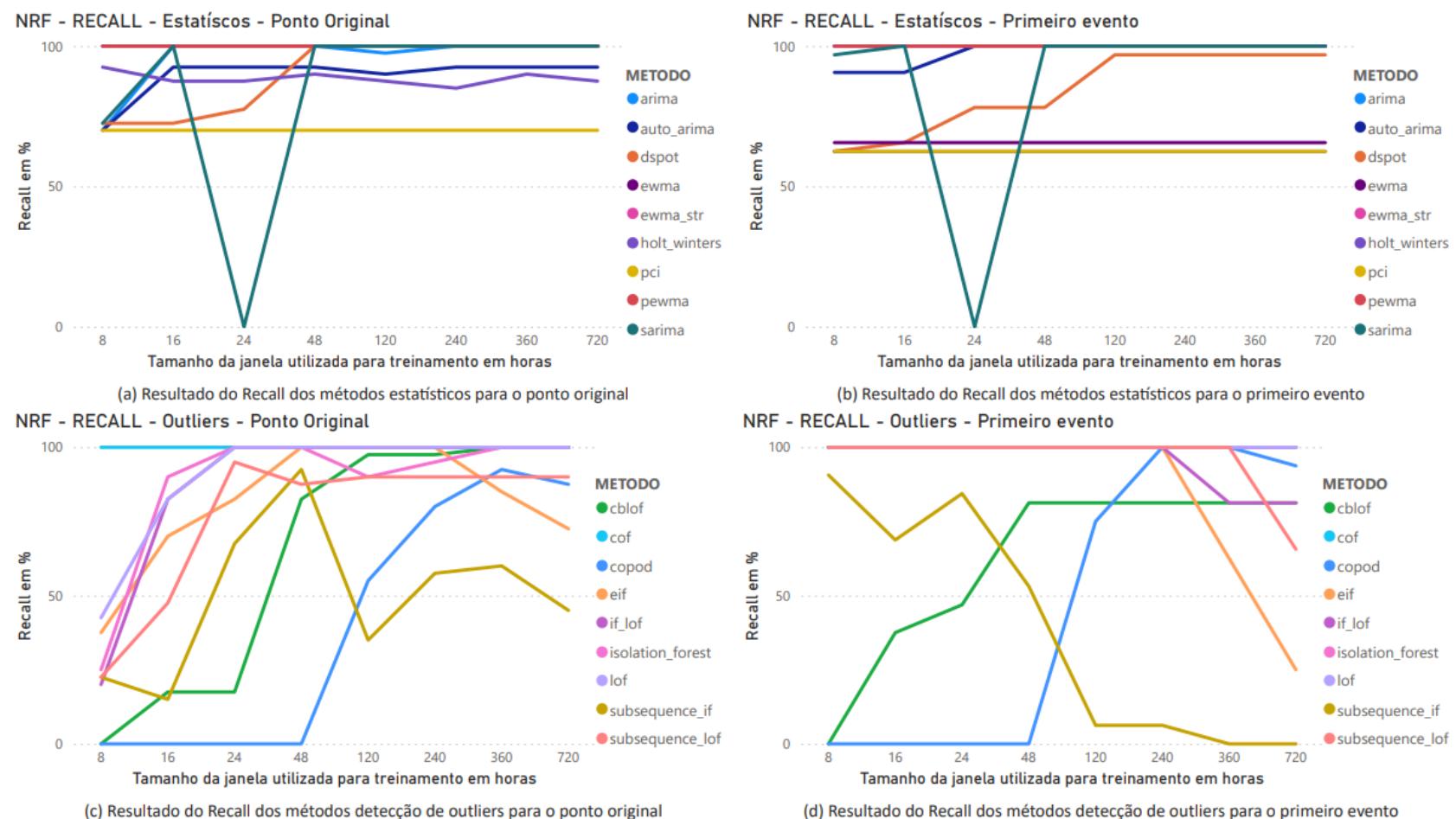


Figura 41 – Resultados do Recall para os testes no DDRJO1.

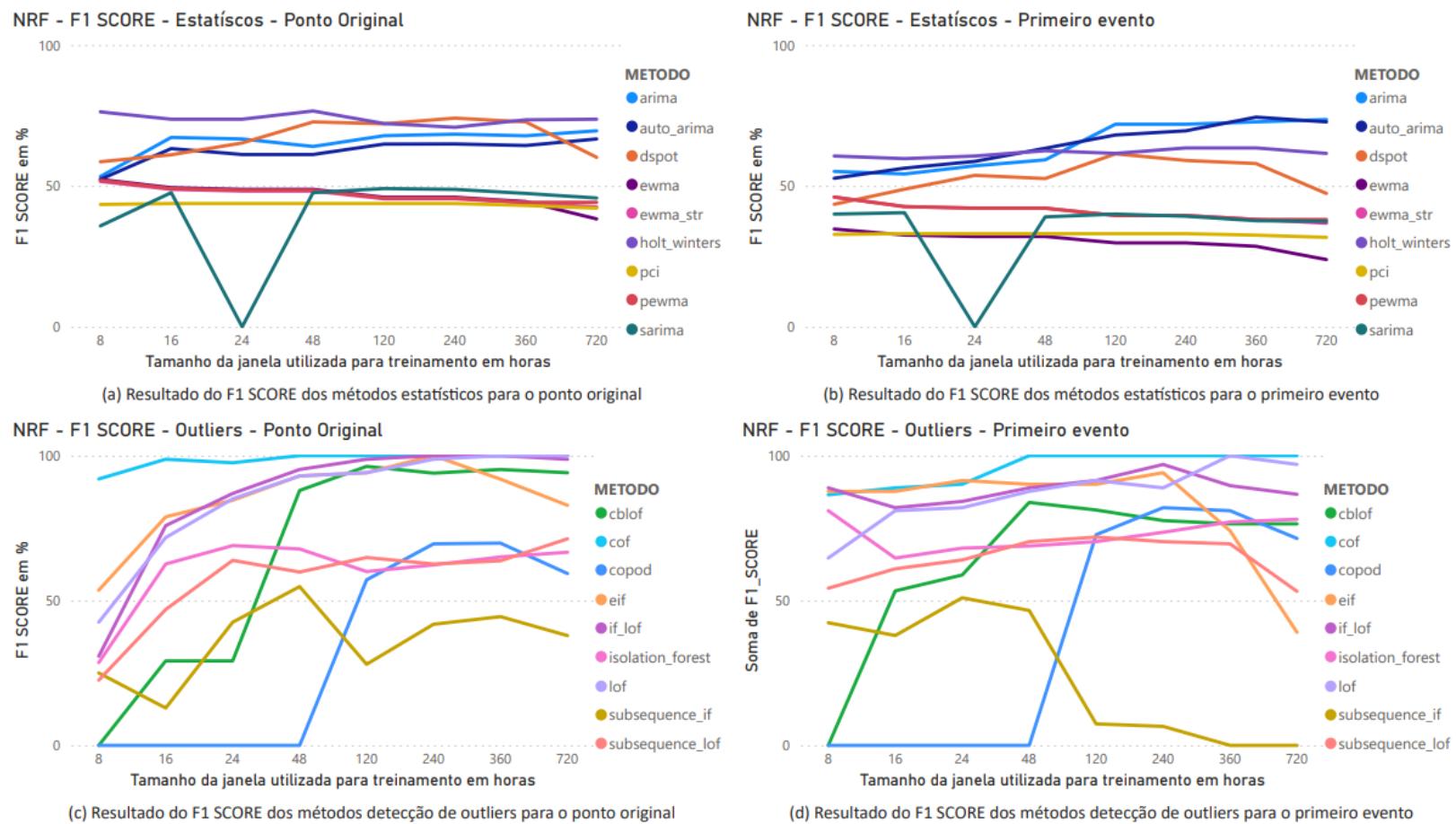


Figura 42 – Resultados do F1 SCORE para os testes no DDRJO1.

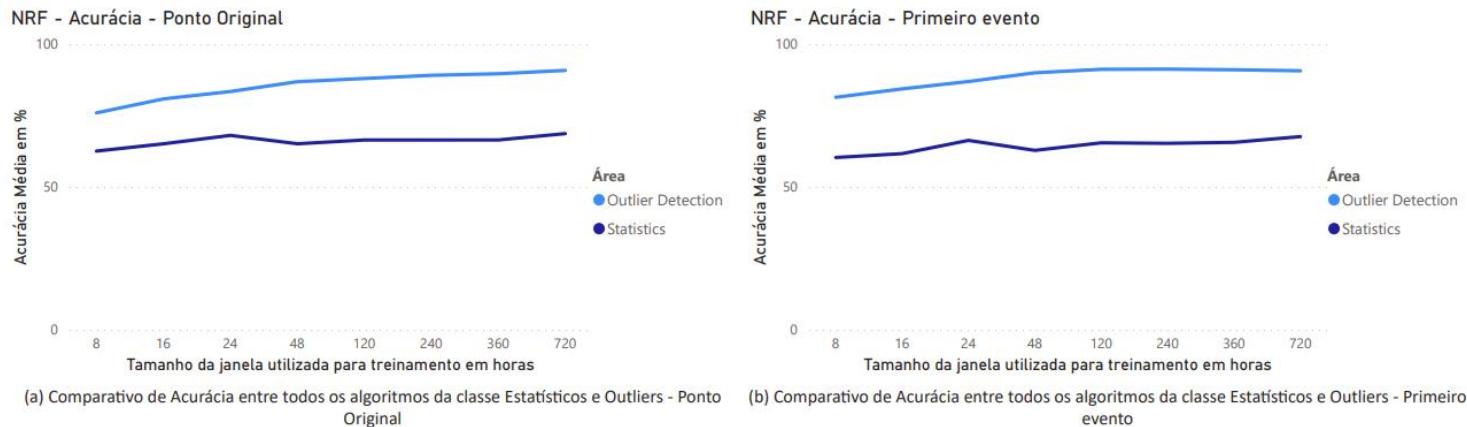


Figura 43– Comparativo de acurácia entre todos os algoritmos das classes estatístico e detecção de outliers.

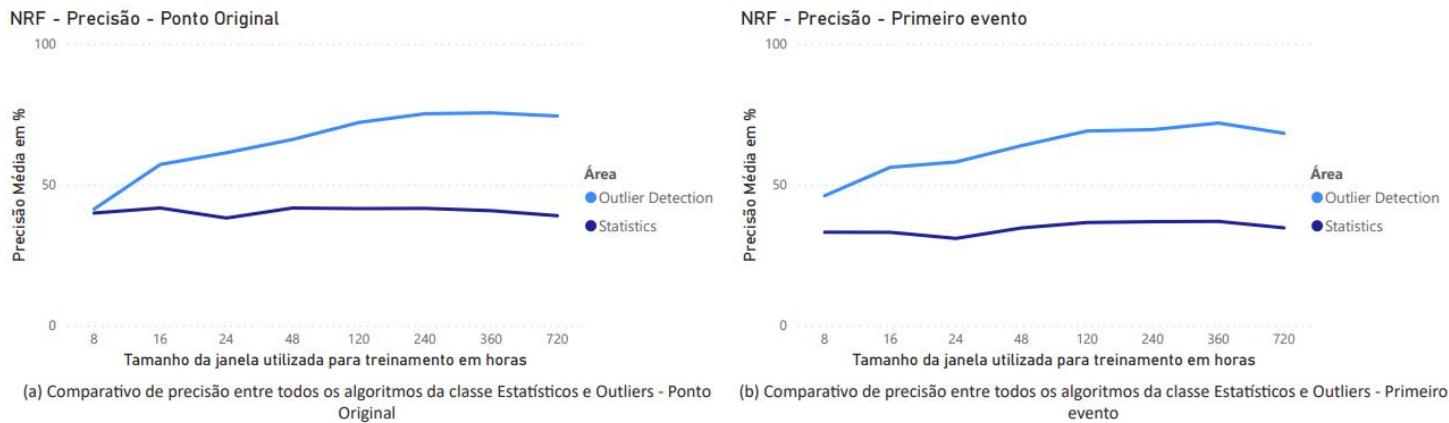


Figura 44 – Comparativo de Precisão entre todos os algoritmos das classes estatístico e detecção de outliers.

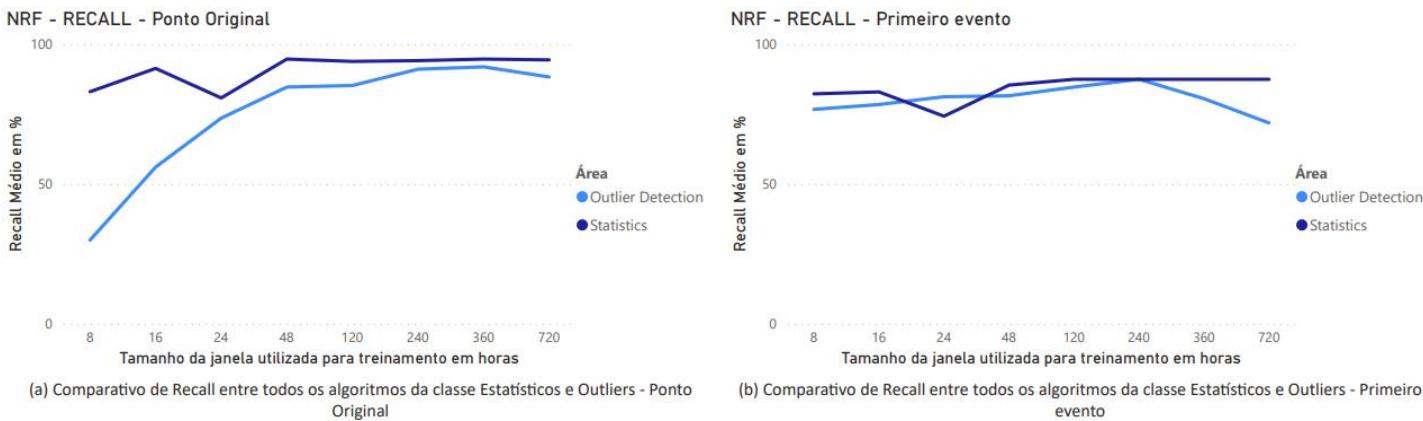


Figura 45 – Comparativo de Recall entre todos os algoritmos das classes estatístico e detecção de outliers

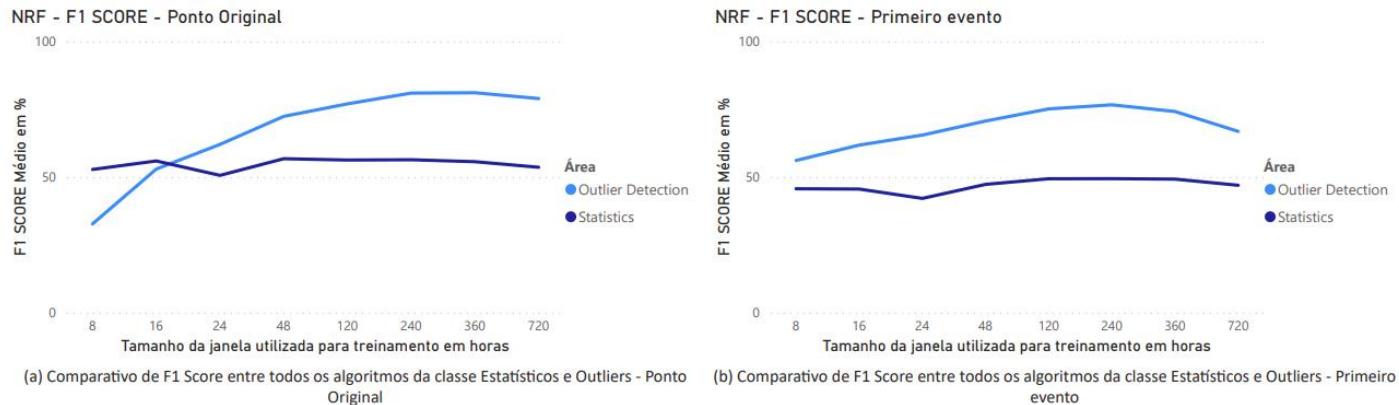


Figura 46 – Comparativo de F1 Score entre todos os algoritmos das classes estatístico e detecção de outliers.

Iniciando a análise pelo ponto original informado pela TIM BR, dentro dos modelos estatísticos quatro métodos tiveram resultados superiores a 80%, sendo eles ARIMA, Auto ARIMA, Holt Winters e DSPOT. No entanto, analisando o primeiro evento de falha, o ARIMA e o AUTO ARIMA aumentam seu desempenho, chegando a 90% de acurácia. Este comportamento era esperado já que o treinamento não levaria em consideração um evento anômalo anterior, logo tornando-se mais assertivo. Os demais métodos tiveram desempenho próximo ou abaixo de 60% nos dois experimentos.

Quanto aos algoritmos de detecção de outliers, vários alcançaram 100% de acurácia, seja na janela de 240 ou de 360 horas, sendo eles LOF, COF, CBLOF, IF-LOF e EIF. Mesmo considerando o pior resultado este não é abaixo de 70% em nenhum dos dois experimentos. Para o primeiro evento de falha apenas dois métodos não alcançaram valores próximos ou superiores a 90%, sendo eles o subsequence-if e o subsequence-lof.

Comparativamente (figura 43), os métodos de detecção de outliers alcançaram em média mais de 90% de acurácia, enquanto métodos estatísticos cerca de 68%, uma diferença de aproximadamente 22%, valor significativo.

Na operação de rede, acionamentos indevidos podem gerar problemas, principalmente se forem múltiplos. Neste cenário a equipe técnica tende a ignorar todos os eventos, já que na sua maioria são falsos, o que pode mascarar um problema real em andamento. Ou seja, falso positivo é um problema que deve ser evitado. Neste contexto a precisão é uma das métricas mais importantes, aliado ao tempo para a execução dos modelos.

Para o ponto de falha informado pela TIM BR, os métodos estatísticos como um todo tiveram resultados de precisão próximos ao 50%, sendo que o melhor desempenho foi o Holt Winters, superior a 60%. Ao mesmo tempo houve métodos que não ficaram próximos a 30% em todos os testes realizados, como o EWMA e surpreendentemente o SARIMA. Até pela proporção de dados removidos pela camada 1 conclui-se que a proporção de séries temporais com sazonalidade é menor em relação a séries mais estáticas. Quanto ao primeiro evento, há uma melhora em alguns métodos estatísticos, como o ARIMA e o Auto ARIMA, no entanto estes continuam abaixo do resultado do Holt Winters.

Nós métodos de detecção de outliers há 4 métodos que alcançaram 100%, sendo eles LOF, COF, EIF e IF-LOF. Outros algoritmos também tiveram ótimo resultado, como o CBLOF, que foi superior a 90%, ou seja, a maioria dos métodos baseados em cálculos de distância tem excelente desempenho no cenário apresentado. Uma característica

interessante observada é que as janelas de treinamento entre 240 horas e 360 horas apresentam os melhores resultados. Mesmo em volumes menores de intervalos de treinamento os resultados dos métodos de detecção já são muito superiores aos estatísticos. Por exemplo, o COF alcançou patamares de 100% de precisão nos dois pontos medidos com apenas 48 horas de dados para treinamento. Falando especificamente sobre o primeiro evento, ocorre uma queda na precisão em relação ao primeiro, no entanto considerando a janela de 360 horas 3 algoritmos continuam apresentando resultado de 100%, sendo eles o LOF, COF e o IF-LOF, sendo que o EIF continua tendo um ótimo desempenho (acima de 90%).

Comparativamente é possível perceber uma grande diferença entre as duas áreas de métodos, chegando a mais de 35% de diferença na precisão. Este resultado está representado na figura 44.

Quanto ao Recall, a situação inverte-se se compararmos os métodos estatísticos e os de detecção de outliers (figura 41). Algoritmos como ARIMA, SARIMA, DSPOT, EWMA apresentam valores de 100%, tendo comportamento bem estático, tanto para o ponto informado quanto para o primeiro evento. No caso dos métodos de detecção alguns algoritmos oscilam, como o próprio EIF. Comparativamente, os métodos estatísticos têm desempenho melhor nesta métrica (figura 45).

6.2.2 Tempo de Execução

As tabelas 11 e 12 apresentam os resultados de tempo obtidos por método, incluindo tempo medido em *ms* e o tempo normalizado após a aplicação da função log. A figura 47 apresenta os gráficos de tempo normalizado, subdivididos em detecção de outliers e métodos estatísticos. A figura 48 apresenta os tempos médios para as duas áreas de algoritmos testados.

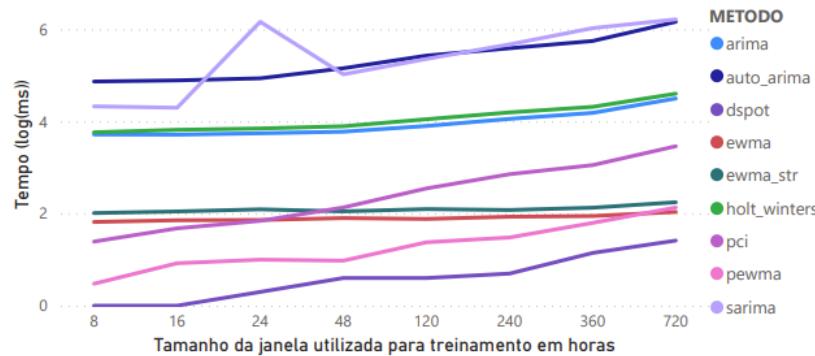
Tabela 12– Tempos de execução aferidos por algoritmo e intervalo – Ponto original.

| METODO | Área | INTERVALO | TEMPO (ms) | TEMPO LOG(ms) | METODO | Área | TEMPO (ms) | TEMPO LOG(ms) |
|------------------|-------------------|-----------|------------|---------------|--------------|------------|------------|---------------|
| lof | Outlier Detection | 8 | 74,74 | 1,873553094 | auto_arima | Statistics | 73623,65 | 4,867017344 |
| lof | Outlier Detection | 16 | 79,3 | 1,899273187 | auto_arima | Statistics | 77984,97 | 4,892010909 |
| lof | Outlier Detection | 24 | 76,35 | 1,882809041 | auto_arima | Statistics | 86483,45 | 4,936933006 |
| lof | Outlier Detection | 48 | 104,52 | 2,019199401 | auto_arima | Statistics | 143011,61 | 5,155371296 |
| lof | Outlier Detection | 120 | 216,76 | 2,335979142 | auto_arima | Statistics | 268908,63 | 5,42960474 |
| lof | Outlier Detection | 240 | 519,14 | 2,715284493 | auto_arima | Statistics | 388817,76 | 5,589746094 |
| lof | Outlier Detection | 360 | 1216,36 | 3,08506213 | auto_arima | Statistics | 559105,31 | 5,747493617 |
| lof | Outlier Detection | 720 | 3117,78 | 3,493845467 | auto_arima | Statistics | 1459151,98 | 6,164100529 |
| cblmf | Outlier Detection | 8 | 1243,42 | 3,094617849 | arima | Statistics | 5218,54 | 3,717549017 |
| cblmf | Outlier Detection | 16 | 543,65 | 2,735319392 | arima | Statistics | 5173,13 | 3,713753392 |
| cblmf | Outlier Detection | 24 | 662,06 | 2,82089735 | arima | Statistics | 5544,04 | 3,743826355 |
| cblmf | Outlier Detection | 48 | 778,56 | 2,891292087 | arima | Statistics | 6000,54 | 3,778190335 |
| cblmf | Outlier Detection | 120 | 1111,67 | 3,045975886 | arima | Statistics | 7981,9 | 3,902106282 |
| cblmf | Outlier Detection | 240 | 1724,44 | 3,236648088 | arima | Statistics | 11383,23 | 4,056265511 |
| cblmf | Outlier Detection | 360 | 2557,07 | 3,407742617 | arima | Statistics | 15364,43 | 4,186516453 |
| cblmf | Outlier Detection | 720 | 5816,09 | 3,764631118 | arima | Statistics | 31503,76 | 4,49836239 |
| cof | Outlier Detection | 8 | 189,22 | 2,276967038 | dspot | Statistics | 1 | 0 |
| cof | Outlier Detection | 16 | 639,81 | 2,806051024 | dspot | Statistics | 1 | 0 |
| cof | Outlier Detection | 24 | 1329,72 | 3,123760201 | dspot | Statistics | 1,99 | 0,298853076 |
| cof | Outlier Detection | 48 | 5215,85 | 3,717325093 | dspot | Statistics | 3,99 | 0,600972896 |
| cof | Outlier Detection | 120 | 15581,49 | 4,192608985 | dspot | Statistics | 3,99 | 0,600972896 |
| cof | Outlier Detection | 240 | 33877,88 | 4,529916225 | dspot | Statistics | 4,98 | 0,697229343 |
| cof | Outlier Detection | 360 | 59310,32 | 4,773130267 | dspot | Statistics | 13,98 | 1,145507171 |
| cof | Outlier Detection | 720 | 171987,95 | 5,23549802 | dspot | Statistics | 25,92 | 1,413634997 |
| subsequence_lof | Outlier Detection | 8 | 462,19 | 2,664820545 | ewma_str | Statistics | 102,74 | 2,011739561 |
| subsequence_lof | Outlier Detection | 16 | 440,86 | 2,644300696 | ewma_str | Statistics | 111,14 | 2,045870392 |
| subsequence_lof | Outlier Detection | 24 | 437,78 | 2,641255917 | ewma_str | Statistics | 123,6 | 2,092018471 |
| subsequence_lof | Outlier Detection | 48 | 453,46 | 2,656538984 | ewma_str | Statistics | 111,63 | 2,047780925 |
| subsequence_lof | Outlier Detection | 120 | 791,27 | 2,8983247 | ewma_str | Statistics | 125,1 | 2,09725731 |
| subsequence_lof | Outlier Detection | 240 | 1905,93 | 3,280106946 | ewma_str | Statistics | 119,61 | 2,07776749 |
| subsequence_lof | Outlier Detection | 360 | 3229,39 | 3,509120496 | ewma_str | Statistics | 134,16 | 2,12762305 |
| subsequence_lof | Outlier Detection | 720 | 10936,51 | 4,038878754 | ewma_str | Statistics | 176,15 | 2,245882648 |
| isolation_forest | Outlier Detection | 8 | 12212,16 | 4,086792486 | pewma | Statistics | 3 | 0,477121255 |
| isolation_forest | Outlier Detection | 16 | 12795,36 | 4,107052509 | pewma | Statistics | 8,35 | 0,921686475 |
| isolation_forest | Outlier Detection | 24 | 13033,63 | 4,115065388 | pewma | Statistics | 9,98 | 0,999130541 |
| isolation_forest | Outlier Detection | 48 | 13401,06 | 4,127139152 | pewma | Statistics | 9,49 | 0,977266212 |
| isolation_forest | Outlier Detection | 120 | 15007,09 | 4,176296487 | pewma | Statistics | 23,82 | 1,376941757 |
| isolation_forest | Outlier Detection | 240 | 15419,21 | 4,188062123 | pewma | Statistics | 30,27 | 1,481012421 |
| isolation_forest | Outlier Detection | 360 | 16398,86 | 4,214813658 | pewma | Statistics | 62,8 | 1,797959644 |
| isolation_forest | Outlier Detection | 720 | 20349,15 | 4,308546273 | pewma | Statistics | 135,31 | 2,131329894 |
| eif | Outlier Detection | 8 | 17591,88 | 4,245312254 | ewma | Statistics | 65,78 | 1,818093869 |
| eif | Outlier Detection | 16 | 34843,07 | 4,542116413 | ewma | Statistics | 71,68 | 1,855397997 |
| eif | Outlier Detection | 24 | 45143,26 | 4,654592918 | ewma | Statistics | 72,63 | 1,861116044 |
| eif | Outlier Detection | 48 | 87854,75 | 4,943765247 | ewma | Statistics | 79,77 | 1,901839592 |
| eif | Outlier Detection | 120 | 217344,52 | 5,337148695 | ewma | Statistics | 76,42 | 1,883207033 |
| eif | Outlier Detection | 240 | 319982,26 | 5,505125901 | ewma | Statistics | 85,55 | 1,932220014 |
| eif | Outlier Detection | 360 | 436171,74 | 5,639657524 | ewma | Statistics | 88,51 | 1,946992341 |
| eif | Outlier Detection | 720 | 864539,09 | 5,936784635 | ewma | Statistics | 108,47 | 2,03530964 |
| subsequence_if | Outlier Detection | 8 | 924,57 | 2,965939798 | pci | Statistics | 24,66 | 1,391993072 |
| subsequence_if | Outlier Detection | 16 | 1746,13 | 3,242076574 | pci | Statistics | 47,8 | 1,679427897 |
| subsequence_if | Outlier Detection | 24 | 2529,35 | 3,403008929 | pci | Statistics | 69,57 | 1,842422003 |
| subsequence_if | Outlier Detection | 48 | 4110,32 | 3,613875634 | pci | Statistics | 135,87 | 2,133123575 |
| subsequence_if | Outlier Detection | 120 | 8737,75 | 3,941399615 | pci | Statistics | 351,75 | 2,546234106 |
| subsequence_if | Outlier Detection | 240 | 9182,43 | 3,962957626 | pci | Statistics | 716,83 | 2,855416173 |
| subsequence_if | Outlier Detection | 360 | 13164,05 | 4,119389523 | pci | Statistics | 1128,61 | 3,052543894 |
| subsequence_if | Outlier Detection | 720 | 19771,04 | 4,296029515 | pci | Statistics | 2898,5 | 3,462173305 |
| if_lof | Outlier Detection | 8 | 12011,93 | 4,079612793 | sarima | Statistics | 21283,22 | 4,328037334 |
| if_lof | Outlier Detection | 16 | 13000,01 | 4,113943686 | sarima | Statistics | 19877,4 | 4,298359577 |
| if_lof | Outlier Detection | 24 | 12944,36 | 4,112080583 | sarima | Statistics | 1459373,23 | 6,164166376 |
| if_lof | Outlier Detection | 48 | 13272,11 | 4,122939972 | sarima | Statistics | 105596,21 | 5,023648331 |
| if_lof | Outlier Detection | 120 | 15060,99 | 4,17785352 | sarima | Statistics | 227613,84 | 5,357198666 |
| if_lof | Outlier Detection | 240 | 16292,01 | 4,211974668 | sarima | Statistics | 471751,77 | 5,673713538 |
| if_lof | Outlier Detection | 360 | 18345,32 | 4,263525292 | sarima | Statistics | 1063283,26 | 6,026648977 |
| if_lof | Outlier Detection | 720 | 26618,21 | 4,425178847 | sarima | Statistics | 1644920,23 | 6,216144842 |
| copod | Outlier Detection | 8 | 364,65 | 2,561876218 | holt_winters | Statistics | 5815,81 | 3,76461021 |
| copod | Outlier Detection | 16 | 197,87 | 2,296379954 | holt_winters | Statistics | 6586,2 | 3,818634915 |
| copod | Outlier Detection | 24 | 200,38 | 2,301854372 | holt_winters | Statistics | 7042,31 | 3,847715139 |
| copod | Outlier Detection | 48 | 209,08 | 2,320312491 | holt_winters | Statistics | 7870,15 | 3,89598301 |
| copod | Outlier Detection | 120 | 255,95 | 2,408155134 | holt_winters | Statistics | 11180,1 | 4,048445688 |
| copod | Outlier Detection | 240 | 298,63 | 2,475133434 | holt_winters | Statistics | 15724,96 | 4,196589549 |
| copod | Outlier Detection | 360 | 364,88 | 2,562150059 | holt_winters | Statistics | 20758,98 | 4,31720601 |
| copod | Outlier Detection | 720 | 559,49 | 2,747792329 | holt_winters | Statistics | 40125,43 | 4,6034197 |

Tabela 13– Tempos de execução aferidos por algoritmo e intervalo –Primeiro evento.

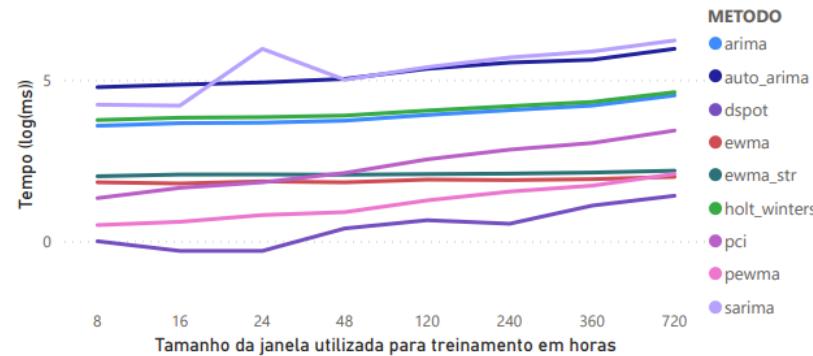
| METODO | Área | INTERVALO | TEMPO (ms) | TEMPO LOG(ms) | METODO | Área | INTERVALO | TEMPO (ms) | TEMPO LOG(ms) |
|------------------|-------------------|-----------|------------|---------------|--------------|------------|-----------|------------|---------------|
| lof | Outlier Detection | 8 | 68,29 | 1,834357113 | ewma | Statistics | 8 | 67,32 | 1,828144107 |
| lof | Outlier Detection | 16 | 73,76 | 1,867820908 | ewma | Statistics | 16 | 61,64 | 1,78986263 |
| lof | Outlier Detection | 24 | 84,24 | 1,925518358 | ewma | Statistics | 24 | 71,74 | 1,855761372 |
| lof | Outlier Detection | 48 | 100,17 | 2,000737674 | ewma | Statistics | 48 | 67,03 | 1,826269219 |
| lof | Outlier Detection | 120 | 204,39 | 2,310459644 | ewma | Statistics | 120 | 81,51 | 1,911210893 |
| lof | Outlier Detection | 240 | 518,52 | 2,714765512 | ewma | Statistics | 240 | 78,73 | 1,896140251 |
| lof | Outlier Detection | 360 | 1134,41 | 3,054770046 | ewma | Statistics | 360 | 84,04 | 1,924486044 |
| lof | Outlier Detection | 720 | 3134,87 | 3,496219536 | ewma | Statistics | 720 | 99,22 | 1,996599223 |
| isolation_forest | Outlier Detection | 8 | 11747,52 | 4,069946193 | pci | Statistics | 8 | 21,69 | 1,336259552 |
| isolation_forest | Outlier Detection | 16 | 12812,52 | 4,107634556 | pci | Statistics | 16 | 45,15 | 1,654657755 |
| isolation_forest | Outlier Detection | 24 | 12967,96 | 4,112871662 | pci | Statistics | 24 | 67,14 | 1,826981337 |
| isolation_forest | Outlier Detection | 48 | 13494,82 | 4,130167096 | pci | Statistics | 48 | 130,86 | 2,116806916 |
| isolation_forest | Outlier Detection | 120 | 14792,06 | 4,17002866 | pci | Statistics | 120 | 346,21 | 2,539339608 |
| isolation_forest | Outlier Detection | 240 | 15495,46 | 4,190204473 | pci | Statistics | 240 | 692,46 | 2,840394691 |
| isolation_forest | Outlier Detection | 360 | 16267,64 | 4,211324553 | pci | Statistics | 360 | 1111,6 | 3,045948538 |
| isolation_forest | Outlier Detection | 720 | 20291,04 | 4,307304307 | pci | Statistics | 720 | 2711,46 | 3,433203202 |
| eif | Outlier Detection | 8 | 16905,22 | 4,228020827 | pewma | Statistics | 8 | 3,17 | 0,501059262 |
| eif | Outlier Detection | 16 | 34650,73 | 4,539712388 | pewma | Statistics | 16 | 3,99 | 0,600972896 |
| eif | Outlier Detection | 24 | 45207,32 | 4,655208762 | pewma | Statistics | 24 | 6,48 | 0,811575006 |
| eif | Outlier Detection | 48 | 89504,65 | 4,951845599 | pewma | Statistics | 48 | 7,98 | 0,902002891 |
| eif | Outlier Detection | 120 | 220957,97 | 5,344309671 | pewma | Statistics | 120 | 18,47 | 1,266466895 |
| eif | Outlier Detection | 240 | 324402,27 | 5,511083885 | pewma | Statistics | 240 | 34,69 | 1,5402043 |
| eif | Outlier Detection | 360 | 430867,5 | 5,634343737 | pewma | Statistics | 360 | 52,89 | 1,723373567 |
| eif | Outlier Detection | 720 | 845311,28 | 5,927016664 | ewma_str | Statistics | 8 | 103,47 | 2,014814449 |
| cblof | Outlier Detection | 8 | 1002,84 | 3,001231648 | ewma_str | Statistics | 16 | 116,97 | 2,06807449 |
| cblof | Outlier Detection | 16 | 536,31 | 2,729415895 | ewma_str | Statistics | 24 | 117,71 | 2,07081336 |
| cblof | Outlier Detection | 24 | 657,29 | 2,817757025 | ewma_str | Statistics | 48 | 115,4 | 2,062205809 |
| cblof | Outlier Detection | 48 | 770,39 | 2,886710637 | ewma_str | Statistics | 120 | 120,65 | 2,081527326 |
| cblof | Outlier Detection | 120 | 1086,63 | 3,036081691 | ewma_str | Statistics | 240 | 124,88 | 2,09649289 |
| cblof | Outlier Detection | 240 | 1683,16 | 3,226125402 | ewma_str | Statistics | 360 | 134,59 | 2,129012793 |
| cblof | Outlier Detection | 360 | 2448,86 | 3,388963957 | ewma_str | Statistics | 720 | 154,03 | 2,187605315 |
| cblof | Outlier Detection | 720 | 5733,83 | 3,758444813 | dspot | Statistics | 8 | 1 | 0 |
| subsequence_if | Outlier Detection | 8 | 897,12 | 2,952850539 | dspot | Statistics | 16 | 0,5 | -0,301029996 |
| subsequence_if | Outlier Detection | 16 | 1706,03 | 3,231986664 | dspot | Statistics | 24 | 0,5 | -0,301029996 |
| subsequence_if | Outlier Detection | 24 | 2492,16 | 3,396575921 | dspot | Statistics | 48 | 2,49 | 0,396199347 |
| subsequence_if | Outlier Detection | 48 | 4092,44 | 3,611982321 | dspot | Statistics | 120 | 4,48 | 0,651278014 |
| subsequence_if | Outlier Detection | 120 | 8764,81 | 3,942742506 | dspot | Statistics | 240 | 3,49 | 0,542825427 |
| subsequence_if | Outlier Detection | 240 | 9134,69 | 3,960693813 | dspot | Statistics | 360 | 12,72 | 1,104487111 |
| subsequence_if | Outlier Detection | 360 | 12953,24 | 4,112378412 | dspot | Statistics | 720 | 25,66 | 1,409256652 |
| subsequence_if | Outlier Detection | 720 | 19853,58 | 4,29783883 | auto_arima | Statistics | 8 | 59852,97 | 4,777085706 |
| subsequence_lof | Outlier Detection | 8 | 451,59 | 2,654744316 | auto_arima | Statistics | 16 | 71900,71 | 4,856733179 |
| subsequence_lof | Outlier Detection | 16 | 434,6 | 2,638089722 | auto_arima | Statistics | 24 | 84188,22 | 4,925251327 |
| subsequence_lof | Outlier Detection | 24 | 447,97 | 2,651248931 | auto_arima | Statistics | 48 | 108307,35 | 5,03465793 |
| subsequence_lof | Outlier Detection | 48 | 449,86 | 2,653077379 | auto_arima | Statistics | 120 | 223766,52 | 5,349795108 |
| subsequence_lof | Outlier Detection | 120 | 786,98 | 2,895963696 | auto_arima | Statistics | 240 | 347035,03 | 5,540373315 |
| subsequence_lof | Outlier Detection | 240 | 1905,85 | 3,280088716 | auto_arima | Statistics | 360 | 424798,44 | 5,628182913 |
| subsequence_lof | Outlier Detection | 360 | 3275,12 | 3,515227217 | auto_arima | Statistics | 720 | 930070,4 | 5,968515823 |
| subsequence_lof | Outlier Detection | 720 | 11154,79 | 4,047461399 | arima | Statistics | 8 | 3801,48 | 3,57995271 |
| copod | Outlier Detection | 8 | 288,1 | 2,459543258 | arima | Statistics | 16 | 4592,81 | 3,662078479 |
| copod | Outlier Detection | 16 | 193,12 | 2,285827253 | arima | Statistics | 24 | 4751,15 | 3,676798742 |
| copod | Outlier Detection | 24 | 210,09 | 2,322405381 | arima | Statistics | 48 | 5486,3 | 3,739279552 |
| copod | Outlier Detection | 48 | 213,63 | 2,32966224 | arima | Statistics | 120 | 8257,94 | 3,916871723 |
| copod | Outlier Detection | 120 | 255,74 | 2,407798661 | arima | Statistics | 240 | 11624,31 | 4,065367183 |
| copod | Outlier Detection | 240 | 293,57 | 2,467711673 | arima | Statistics | 360 | 16091,18 | 4,206587893 |
| copod | Outlier Detection | 360 | 351,04 | 2,545356606 | arima | Statistics | 720 | 33269,05 | 4,5220404 |
| copod | Outlier Detection | 720 | 554,62 | 2,743995526 | sarima | Statistics | 8 | 17137,75 | 4,233953803 |
| if_lof | Outlier Detection | 8 | 11676,68 | 4,067319379 | sarima | Statistics | 16 | 16019,49 | 4,204648686 |
| if_lof | Outlier Detection | 16 | 12683,77 | 4,103248358 | sarima | Statistics | 24 | 927132,4 | 5,967141758 |
| if_lof | Outlier Detection | 24 | 12889,39 | 4,110232365 | sarima | Statistics | 48 | 102023,76 | 5,008701325 |
| if_lof | Outlier Detection | 48 | 13150,99 | 4,118958448 | sarima | Statistics | 120 | 250563,92 | 5,398918535 |
| if_lof | Outlier Detection | 120 | 15061,11 | 4,17785698 | sarima | Statistics | 240 | 501219,15 | 5,700027656 |
| if_lof | Outlier Detection | 240 | 16285,31 | 4,21179603 | sarima | Statistics | 360 | 767065,71 | 5,884832569 |
| if_lof | Outlier Detection | 360 | 18439,7 | 4,265753851 | sarima | Statistics | 720 | 1686637,75 | 6,227021816 |
| if_lof | Outlier Detection | 720 | 26112,08 | 4,416841468 | holt_winters | Statistics | 8 | 5731,9 | 3,758298605 |
| cof | Outlier Detection | 8 | 181,92 | 2,259880447 | holt_winters | Statistics | 16 | 6766,46 | 3,830361519 |
| cof | Outlier Detection | 16 | 633,95 | 2,802055006 | holt_winters | Statistics | 24 | 7047,94 | 3,848062198 |
| cof | Outlier Detection | 24 | 1368,52 | 3,136251149 | holt_winters | Statistics | 48 | 7881,95 | 3,896633676 |
| cof | Outlier Detection | 48 | 5222,32 | 3,71786348 | holt_winters | Statistics | 120 | 11292,6 | 4,052793945 |
| cof | Outlier Detection | 120 | 15583,83 | 4,192674202 | holt_winters | Statistics | 240 | 15414,93 | 4,187941557 |
| cof | Outlier Detection | 240 | 34243,84 | 4,534582459 | holt_winters | Statistics | 360 | 20889,64 | 4,319930956 |
| cof | Outlier Detection | 360 | 60243,31 | 4,779908826 | holt_winters | Statistics | 720 | 41849,67 | 4,621692038 |
| cof | Outlier Detection | 720 | 174049,04 | 5,240671632 | | | | | |

NRF - TEMPO - Estatísticos - Ponto Original



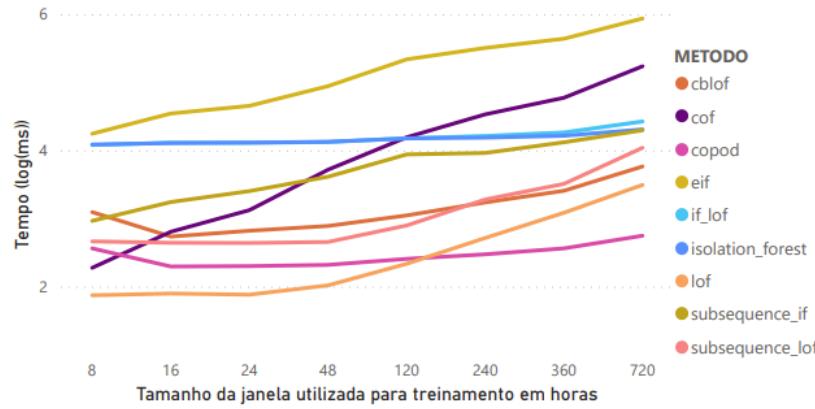
(a) Resultado do tempo dos métodos estatísticos para o ponto original

NRF - TEMPO - Estatísticos - Primeiro evento



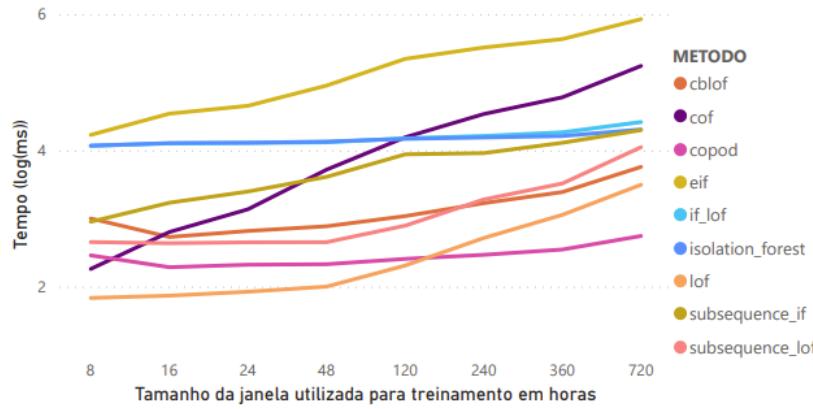
(b) Resultado do tempo da acurácia dos métodos estatísticos para o primeiro evento

NRF - TEMPO - Outliers - Ponto Original



(c) Resultado do tempo dos métodos detecção de outliers para o ponto original

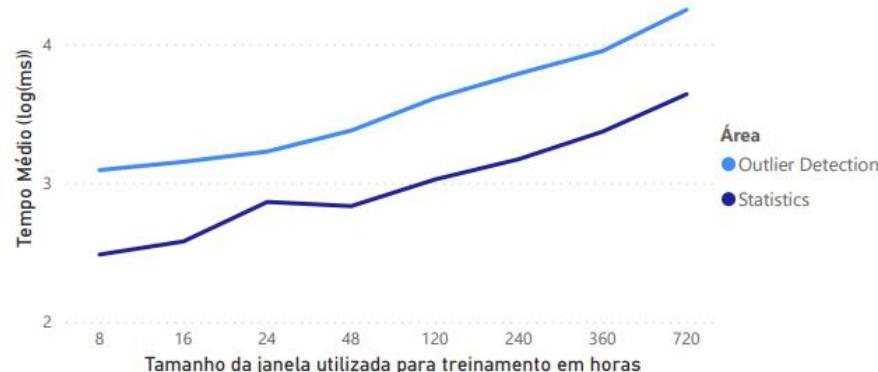
NRF - TEMPO - Estatísticos - Primeiro evento



(d) Resultado do tempo da acurácia - métodos estatísticos - primeiro evento

Figura 47 – Resultados do tempo para os testes no DDRJ01.

NRF - TEMPO - Ponto Original



(a) Comparativo de tempo entre todos os algoritmos da classe Estatísticos e Outliers - Ponto Original

NRF - TEMPO - Primeiro evento



(b) Comparativo de tempo entre todos os algoritmos da classe Estatísticos e Outliers - Primeiro evento

Figura 48 – Comparativo de tempo entre todos os algoritmos das classes estatístico e detecção de outliers.

Métodos de detecção de outliers tendem a ter um tempo de execução maior que métodos estatísticos, já que normalmente estes necessitam calcular a distância entre todos os pontos para assim determinar o score de cada medição. O LOF, por exemplo, tem ordem de complexidade n^2 . No entanto, as janelas de treinamento propostos neste trabalho são pequenas, consequentemente a diferença de tempo entre estas abordagens não é tão significativo. Isto pode ser verificado na figura 48. Analisando os métodos individualmente, alguns foram excessivamente lentos, como o SARIMA e o EIF, enquanto outros como o LOF tiveram tempos surpreendentemente positivos.

O problema proposto é ser capaz de identificar eventos anômalos em tempo de inferência aceitável, o que significa analisar as séries temporais disponíveis em 30 minutos (periodicidade de atualização dos KPIs). Na tabela 13 apresentamos o resultado do tempo de execução do método para uma janela de 240 horas de treinamento e 190 séries temporais (após saída do filtro 1), incluindo uma previsão para o cenário de 524 mil séries (total existente no core 5G SA) e após as 3 camadas do filtro.

Tabela 14 – Tempos aferidos dos métodos de detecção de anomalia no cenário de 240 horas, tempo total previsto para 524 mil séries e tempo previsto após a saída dos filtros.

| Método | Área | Tempo (ms) | Tempo previsto para execução | | | |
|------------------|-------------------|------------|------------------------------|---------------|---------------|---------------|
| | | | Total de séries | Após camada 1 | Após camada 2 | Após camada 3 |
| | | | 524019 | 63249 | 62237 | 17240 |
| lof | Outlier Detection | 519,14 | 23,86 | 2,88 | 2,83 | 0,79 |
| cblof | Outlier Detection | 1724,44 | 79,27 | 9,57 | 9,41 | 2,61 |
| cof | Outlier Detection | 33877,88 | 1557,25 | 187,96 | 184,95 | 51,23 |
| subsequence_lof | Outlier Detection | 1905,93 | 87,61 | 10,57 | 10,41 | 2,88 |
| isolation_forest | Outlier Detection | 15419,21 | 708,77 | 85,55 | 84,18 | 23,32 |
| eif | Outlier Detection | 319982,26 | 14708,49 | 1775,31 | 1746,91 | 483,91 |
| subsequence_if | Outlier Detection | 9182,43 | 422,08 | 50,95 | 50,13 | 13,89 |
| if_lof | Outlier Detection | 16292,01 | 748,89 | 90,39 | 88,94 | 24,64 |
| copod | Outlier Detection | 298,63 | 13,73 | 1,66 | 1,63 | 0,45 |
| auto_arima | Statistics | 388817,76 | 17872,62 | 2157,23 | 2122,71 | 588,01 |
| arima | Statistics | 11383,23 | 523,25 | 63,16 | 62,15 | 17,21 |
| dspot | Statistics | 4,98 | 0,23 | 0,03 | 0,03 | 0,01 |
| ewma_str | Statistics | 119,61 | 5,50 | 0,66 | 0,65 | 0,18 |
| pewma | Statistics | 30,27 | 1,39 | 0,17 | 0,17 | 0,05 |
| ewma | Statistics | 85,55 | 3,93 | 0,47 | 0,47 | 0,13 |
| pci | Statistics | 716,83 | 32,95 | 3,98 | 3,91 | 1,08 |
| sarima | Statistics | 471751,77 | 21684,81 | 2617,36 | 2575,48 | 713,43 |
| holt_winters | Statistics | 15724,96 | 722,82 | 87,24 | 85,85 | 23,78 |

Por esta análise é possível verificar que mesmo sem o filtro a maioria dos métodos estatísticos são aderentes ao problema proposto, no entanto neste cenário dos métodos de detecção de outliers apenas o LOF e o COPOD executariam em tempo inferior a 30

minutos. No entanto, através da utilização das camadas de filtragem a maioria dos métodos utilizados tornam-se viáveis.

Retornando a integração com o filtro, algumas séries indicadas como constantes por nossa abordagem utilizando a entropia de Shannon apresentaram evento anômalo, o que não é um problema já que como mencionado no capítulo 4 esta etapa é mais focada em classificar, sendo necessário comparar se o valor da moda é igual ao valor atual, o que tem baixa complexidade. Nenhuma série temporal filtrada na camada 2 foi identificada como anômala, como esperado. Quanto à camada 3, abaixo as séries indicadas como correlatas e suas respectivas classificações de falha.

Tabela 15 – Séries temporais correlatas e suas classificações de falha, mesma classe

| ID ST | FALHA | ID GRUPO | ST CORRELATAS | ST COM FALHA | ST SEM FALHA | ACERTO(%) |
|-------|-------|----------|---------------|--------------|--------------|-----------|
| 0 | 1 | 0 | 4 | 3 | 1 | 75% |
| 1 | 1 | 0 | 4 | 3 | 1 | 75% |
| 2 | 0 | 0 | 4 | 3 | 1 | 25% |
| 3 | 1 | 0 | 4 | 3 | 1 | 75% |
| 4 | 0 | 1 | 4 | 2 | 2 | 50% |
| 5 | 0 | 1 | 4 | 2 | 2 | 50% |
| 6 | 1 | 1 | 4 | 2 | 2 | 50% |
| 7 | 1 | 1 | 4 | 2 | 2 | 50% |
| 8 | 1 | 2 | 2 | 2 | 0 | 100% |
| 9 | 1 | 2 | 2 | 2 | 0 | 100% |
| 10 | 0 | 3 | 2 | 0 | 2 | 100% |
| 11 | 0 | 3 | 2 | 0 | 2 | 100% |
| 12 | 0 | 4 | 2 | 0 | 2 | 100% |
| 13 | 0 | 5 | 2 | 0 | 2 | 100% |
| 14 | 0 | 5 | 2 | 0 | 2 | 100% |
| 15 | 0 | 6 | 2 | 0 | 2 | 100% |
| 16 | 0 | 6 | 2 | 0 | 2 | 100% |
| 17 | 1 | 7 | 4 | 4 | 0 | 100% |
| 18 | 1 | 7 | 4 | 4 | 0 | 100% |
| 19 | 1 | 7 | 4 | 4 | 0 | 100% |
| 20 | 1 | 7 | 4 | 4 | 0 | 100% |
| 21 | 0 | 8 | 2 | 0 | 2 | 100% |
| 22 | 0 | 8 | 2 | 0 | 2 | 100% |
| 23 | 0 | 9 | 3 | 0 | 3 | 100% |
| 24 | 0 | 9 | 3 | 0 | 3 | 100% |
| 25 | 0 | 9 | 3 | 0 | 3 | 100% |
| 26 | 1 | 10 | 2 | 2 | 0 | 100% |
| 27 | 1 | 10 | 2 | 2 | 0 | 100% |
| 28 | 1 | 11 | 2 | 2 | 0 | 100% |
| 29 | 1 | 11 | 2 | 2 | 0 | 100% |
| 30 | 0 | 12 | 2 | 0 | 2 | 100% |
| 33 | 0 | 12 | 2 | 0 | 2 | 100% |
| 31 | 0 | 13 | 2 | 0 | 2 | 100% |
| 34 | 0 | 13 | 2 | 0 | 2 | 100% |
| 32 | 0 | 14 | 4 | 0 | 4 | 100% |
| 35 | 0 | 14 | 4 | 0 | 4 | 100% |
| 40 | 0 | 14 | 4 | 0 | 4 | 100% |
| 45 | 0 | 14 | 4 | 0 | 4 | 100% |
| 37 | 0 | 15 | 6 | 2 | 4 | 67% |
| 38 | 0 | 15 | 6 | 2 | 4 | 67% |
| 42 | 0 | 15 | 6 | 2 | 4 | 67% |
| 43 | 0 | 15 | 6 | 2 | 4 | 67% |
| 49 | 1 | 15 | 6 | 2 | 4 | 33% |
| 55 | 1 | 15 | 6 | 2 | 4 | 33% |
| 36 | 0 | 16 | 2 | 0 | 2 | 100% |
| 41 | 0 | 16 | 2 | 0 | 2 | 100% |
| 39 | 0 | 17 | 2 | 0 | 2 | 100% |
| 44 | 0 | 17 | 2 | 0 | 2 | 100% |
| 47 | 1 | 18 | 8 | 8 | 0 | 100% |
| 51 | 1 | 18 | 8 | 8 | 0 | 100% |

| ID ST | FALHA | ID GRUPO | ST CORRELATAS | ST COM FALHA | ST SEM FALHA | ACERTO(%) |
|-------|-------|----------|---------------|--------------|--------------|-----------|
| 53 | 1 | 18 | 8 | 8 | 0 | 100% |
| 57 | 1 | 18 | 8 | 8 | 0 | 100% |
| 59 | 1 | 18 | 8 | 8 | 0 | 100% |
| 63 | 1 | 18 | 8 | 8 | 0 | 100% |
| 65 | 1 | 18 | 8 | 8 | 0 | 100% |
| 69 | 1 | 18 | 8 | 8 | 0 | 100% |
| 46 | 0 | 19 | 2 | 0 | 2 | 100% |
| 52 | 0 | 19 | 2 | 0 | 2 | 100% |
| 48 | 1 | 20 | 4 | 4 | 0 | 100% |
| 54 | 1 | 20 | 4 | 4 | 0 | 100% |
| 60 | 1 | 20 | 4 | 4 | 0 | 100% |
| 66 | 1 | 20 | 4 | 4 | 0 | 100% |
| 50 | 1 | 21 | 4 | 4 | 0 | 100% |
| 56 | 1 | 21 | 4 | 4 | 0 | 100% |
| 62 | 1 | 21 | 4 | 4 | 0 | 100% |
| 68 | 1 | 21 | 4 | 4 | 0 | 100% |
| 58 | 0 | 22 | 2 | 0 | 2 | 100% |
| 64 | 0 | 22 | 2 | 0 | 2 | 100% |
| 61 | 1 | 23 | 2 | 2 | 0 | 100% |
| 67 | 1 | 23 | 2 | 2 | 0 | 100% |
| 70 | 0 | 24 | 5 | 0 | 5 | 100% |
| 71 | 0 | 24 | 5 | 0 | 5 | 100% |
| 72 | 0 | 24 | 5 | 0 | 5 | 100% |
| 73 | 0 | 24 | 5 | 0 | 5 | 100% |
| 76 | 0 | 24 | 5 | 0 | 5 | 100% |
| 74 | 0 | 25 | 2 | 0 | 2 | 100% |
| 75 | 0 | 25 | 2 | 0 | 2 | 100% |
| 77 | 0 | 26 | 3 | 0 | 3 | 100% |
| 84 | 0 | 26 | 3 | 0 | 3 | 100% |
| 95 | 0 | 26 | 3 | 0 | 3 | 100% |
| 78 | 0 | 27 | 3 | 0 | 3 | 100% |
| 85 | 0 | 27 | 3 | 0 | 3 | 100% |
| 96 | 0 | 27 | 3 | 0 | 3 | 100% |
| 79 | 0 | 28 | 2 | 0 | 2 | 100% |
| 86 | 0 | 28 | 2 | 0 | 2 | 100% |
| 80 | 1 | 29 | 3 | 3 | 0 | 100% |
| 87 | 1 | 29 | 3 | 3 | 0 | 100% |
| 97 | 1 | 29 | 3 | 3 | 0 | 100% |
| 81 | 0 | 30 | 3 | 0 | 3 | 100% |
| 88 | 0 | 30 | 3 | 0 | 3 | 100% |
| 98 | 0 | 30 | 3 | 0 | 3 | 100% |
| 82 | 0 | 31 | 3 | 0 | 3 | 100% |
| 89 | 0 | 31 | 3 | 0 | 3 | 100% |
| 99 | 0 | 31 | 3 | 0 | 3 | 100% |
| 83 | 0 | 32 | 2 | 0 | 2 | 100% |
| 90 | 0 | 32 | 2 | 0 | 2 | 100% |
| 91 | 0 | 33 | 2 | 0 | 2 | 100% |
| 93 | 0 | 33 | 2 | 0 | 2 | 100% |
| 92 | 0 | 34 | 2 | 0 | 2 | 100% |
| 94 | 0 | 34 | 2 | 0 | 2 | 100% |

Tabela 16 – Séries temporais correlatas e suas classificações de falha, geral

| ID ST | FALHA | ID GRUPO | SERIES CORRELATAS | SERIES COM FALHA | SERIES SEM FALHA | ACERTO(%) |
|-------|-------|----------|-------------------|------------------|------------------|-----------|
| 0 | 1 | 0 | 8 | 5 | 3 | 62,00% |
| 1 | 1 | 0 | 8 | 5 | 3 | 62,00% |
| 3 | 0 | 0 | 8 | 5 | 3 | 38,00% |
| 4 | 1 | 0 | 8 | 5 | 3 | 62,00% |
| 5 | 0 | 0 | 8 | 5 | 3 | 38,00% |
| 6 | 0 | 0 | 8 | 5 | 3 | 38,00% |
| 7 | 1 | 0 | 8 | 5 | 3 | 62,00% |
| 8 | 1 | 0 | 8 | 5 | 3 | 62,00% |
| 9 | 1 | 1 | 2 | 2 | 0 | 100,00% |
| 10 | 1 | 1 | 2 | 2 | 0 | 100,00% |
| 11 | 0 | 2 | 2 | 0 | 2 | 100,00% |
| 12 | 0 | 2 | 2 | 0 | 2 | 100,00% |
| 13 | 0 | 3 | 3 | 0 | 3 | 100,00% |
| 14 | 0 | 4 | 2 | 0 | 2 | 100,00% |
| 15 | 0 | 4 | 2 | 0 | 2 | 100,00% |
| 16 | 0 | 5 | 2 | 0 | 2 | 100,00% |
| 17 | 0 | 5 | 2 | 0 | 2 | 100,00% |
| 18 | 1 | 6 | 4 | 4 | 0 | 100,00% |
| 19 | 1 | 6 | 4 | 4 | 0 | 100,00% |
| 20 | 1 | 6 | 4 | 4 | 0 | 100,00% |
| 21 | 1 | 6 | 4 | 4 | 0 | 100,00% |
| 22 | 0 | 7 | 2 | 0 | 2 | 100,00% |
| 23 | 0 | 7 | 2 | 0 | 2 | 100,00% |
| 24 | 0 | 8 | 3 | 0 | 3 | 100,00% |
| 25 | 0 | 8 | 3 | 0 | 3 | 100,00% |
| 26 | 0 | 8 | 3 | 0 | 3 | 100,00% |
| 27 | 1 | 9 | 2 | 2 | 0 | 100,00% |
| 28 | 1 | 9 | 2 | 2 | 0 | 100,00% |
| 29 | 1 | 10 | 2 | 2 | 0 | 100,00% |
| 30 | 1 | 10 | 2 | 2 | 0 | 100,00% |
| 31 | 0 | 11 | 2 | 0 | 2 | 100,00% |
| 34 | 0 | 11 | 2 | 0 | 2 | 100,00% |
| 32 | 0 | 12 | 2 | 0 | 2 | 100,00% |
| 35 | 0 | 12 | 2 | 0 | 2 | 100,00% |
| 33 | 0 | 13 | 4 | 0 | 4 | 100,00% |
| 36 | 0 | 13 | 4 | 0 | 4 | 100,00% |
| 41 | 0 | 13 | 4 | 0 | 4 | 100,00% |
| 46 | 0 | 13 | 4 | 0 | 4 | 100,00% |
| 2 | 0 | 14 | 7 | 2 | 5 | 71,00% |
| 38 | 0 | 14 | 7 | 2 | 5 | 71,00% |
| 39 | 0 | 14 | 7 | 2 | 5 | 71,00% |
| 43 | 0 | 14 | 7 | 2 | 5 | 71,00% |
| 44 | 0 | 14 | 7 | 2 | 5 | 71,00% |
| 50 | 1 | 14 | 7 | 2 | 5 | 29,00% |
| 56 | 1 | 14 | 7 | 2 | 5 | 29,00% |
| 37 | 0 | 15 | 2 | 0 | 2 | 100,00% |
| 42 | 0 | 15 | 2 | 0 | 2 | 100,00% |
| 40 | 0 | 16 | 2 | 0 | 2 | 100,00% |
| 45 | 0 | 16 | 2 | 0 | 2 | 100,00% |
| 93 | 0 | 33 | 2 | 0 | 2 | 100,00% |
| 95 | 0 | 33 | 2 | 0 | 2 | 100,00% |

| ID ST | FALHA | ID GRUPO | SERIES CORRELATAS | SERIES COM FALHA | SERIES SEM FALHA | ACERTO(%) |
|-------|-------|----------|-------------------|------------------|------------------|-----------|
| 48 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 52 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 54 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 58 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 60 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 64 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 66 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 70 | 1 | 17 | 8 | 8 | 0 | 100,00% |
| 47 | 0 | 18 | 2 | 0 | 2 | 100,00% |
| 53 | 0 | 18 | 2 | 0 | 2 | 100,00% |
| 49 | 1 | 19 | 4 | 4 | 0 | 100,00% |
| 55 | 1 | 19 | 4 | 4 | 0 | 100,00% |
| 61 | 1 | 19 | 4 | 4 | 0 | 100,00% |
| 67 | 1 | 19 | 4 | 4 | 0 | 100,00% |
| 51 | 1 | 20 | 4 | 4 | 0 | 100,00% |
| 57 | 1 | 20 | 4 | 4 | 0 | 100,00% |
| 63 | 1 | 20 | 4 | 4 | 0 | 100,00% |
| 69 | 1 | 20 | 4 | 4 | 0 | 100,00% |
| 59 | 0 | 21 | 2 | 0 | 2 | 100,00% |
| 65 | 0 | 21 | 2 | 0 | 2 | 100,00% |
| 62 | 1 | 22 | 2 | 2 | 0 | 100,00% |
| 68 | 1 | 22 | 2 | 2 | 0 | 100,00% |
| 71 | 0 | 23 | 5 | 0 | 5 | 100,00% |
| 72 | 0 | 23 | 5 | 0 | 5 | 100,00% |
| 73 | 0 | 23 | 5 | 0 | 5 | 100,00% |
| 74 | 0 | 23 | 5 | 0 | 5 | 100,00% |
| 77 | 0 | 23 | 5 | 0 | 5 | 100,00% |
| 75 | 0 | 24 | 2 | 0 | 2 | 100,00% |
| 76 | 0 | 24 | 2 | 0 | 2 | 100,00% |
| 78 | 0 | 25 | 3 | 0 | 3 | 100,00% |
| 85 | 0 | 25 | 3 | 0 | 3 | 100,00% |
| 96 | 0 | 25 | 3 | 0 | 3 | 100,00% |
| 79 | 0 | 26 | 3 | 0 | 3 | 100,00% |
| 86 | 0 | 26 | 3 | 0 | 3 | 100,00% |
| 97 | 0 | 26 | 3 | 0 | 3 | 100,00% |
| 80 | 0 | 27 | 2 | 0 | 2 | 100,00% |
| 87 | 0 | 27 | 2 | 0 | 2 | 100,00% |
| 81 | 1 | 28 | 3 | 3 | 0 | 100,00% |
| 88 | 1 | 28 | 3 | 3 | 0 | 100,00% |
| 98 | 1 | 28 | 3 | 3 | 0 | 100,00% |
| 82 | 0 | 29 | 3 | 0 | 3 | 100,00% |
| 89 | 0 | 29 | 3 | 0 | 3 | 100,00% |
| 99 | 0 | 29 | 3 | 0 | 3 | 100,00% |
| 83 | 0 | 30 | 3 | 0 | 3 | 100,00% |
| 90 | 0 | 30 | 3 | 0 | 3 | 100,00% |
| 100 | 0 | 30 | 3 | 0 | 3 | 100,00% |
| 84 | 0 | 31 | 2 | 0 | 2 | 100,00% |
| 91 | 0 | 31 | 2 | 0 | 2 | 100,00% |
| 92 | 0 | 32 | 2 | 0 | 2 | 100,00% |
| 94 | 0 | 32 | 2 | 0 | 2 | 100,00% |

Através destes resultados é possível perceber que a camada 3 não filtra corretamente alguns cenários, tendo os resultados de acurácia abaixo:

- **Mesma classe:** 91,40%.
- **Correlação Geral:** 88,97%.

Analizando o grupo 0 da correlação na mesma classe (informação referente a tabela 14) é possível verificar que 4 séries são correlatas, no entanto houve evento anômalo classificado em 3 destas. A figura 49 é referente as séries temporais deste grupo.

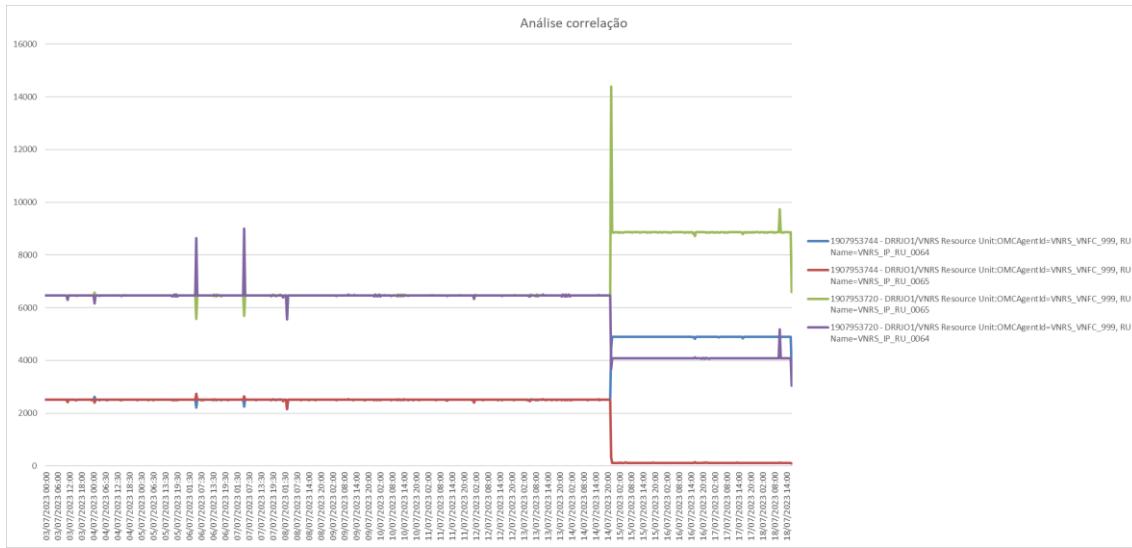


Figura 49 – Séries com alta correlação, grupo 0, mesma classe.

De fato, é possível identificar queda acentuada em 3 das 4 séries temporais, motivo pelo qual estas foram classificadas como anômalas, o que indica que a abordagem de fato não funcionaria em todos os cenários existentes.

6.3 Considerações Finais

Com base em tudo o que foi exposto, é possível concluir que é possível detectar anomalias em uma rede core 5G SA com alta assertividade e com tempo de inferência aceitável.

Sobre a precisão dos métodos é perceptível que os métodos de detecção de outliers tiveram um desempenho superior aos métodos estatísticos. O método LOF em particular teve tanto tempo de execução aceitável como um dos melhores desempenhos para os intervalos testados, sendo que em 240 horas a precisão alcançou 100% nos dois testes realizados. É importante ressaltar que a maioria dos métodos de previsão precisam de séries temporais sem tendência, fato este que não é garantido na abordagem que adotamos, o que justifica também o desempenho mais baixo em alguns cenários testados.

Sobre os filtros, as camadas 1 e 2 tiveram os resultados esperados, eliminando mais de 88% das séries temporais com alta precisão. No entanto, a utilização da camada 3 resultaria em classificação errônea em alguns cenários. Sobre sua utilização ou não no problema proposto, depende de qual é a maior necessidade, diminuir o volume de séries a serem analisadas ou ter o melhor desempenho possível na detecção de anomalia. Com

base nas informações da tabela 13, muitos dos métodos executariam em tempo aceitável sem a necessidade da camada 3, dentre eles o próprio LOF. Isto indica que apenas as camadas 1 e 2 já são diminuiriam o volume de séries temporais o suficiente para a maioria dos algoritmos.

7 Conclusões

Com base em tudo o que foi apresentado neste trabalho pode-se concluir que é possível implementar um detector de anomalias para a rede core 5G SA com alta assertividade e com tempo de inferência aceitável, analisando todos os KPIs, adaptável a mudanças de características da rede por meio de uma abordagem caixa-preta. Para tanto propomos o framework abaixo.

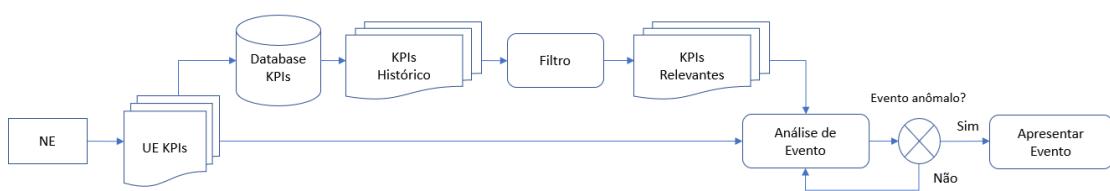


Figura 50 – Framework desenvolvido.

É importante ressaltar que todo o nosso trabalho foi considerando a realidade existente na TIM em 2023. Uma eventual alteração, seja na quantidade de elementos de rede disponíveis ou na periodicidade dos KPIs, pode afetar diretamente nos resultados. No entanto, todo nosso trabalho foi desenvolvido sem considerar facilitadores, como paralelismo para processar simultaneamente múltiplos elementos.

Este trabalho foi apresentado a diretoria de Core Operations da TIM BR, e como consequência dos resultados obtidos, está sendo estudado o desenvolvimento de um sistema tendo esta pesquisa como base.

7.1 Contribuições

1. Caracterização dos KPIs da rede core 5G SA de uma rede comercial complexa real, o que facilita futuras pesquisas na área.
2. Desenvolvimento de um filtro dinâmico com o objetivo de diminuir a quantidade de séries temporais a serem analisadas, simplificando assim a aplicação de métodos de detecção de anomalia.

3. Implementação de métodos de inteligência superficial não supervisionado para detecção de anomalias em rede core 5G SA real, levando-se em consideração todos os KPIs existente dentro da arquitetura.

7.2 Possibilidade de Trabalhos Futuros

- **Expansão da pesquisa para outras áreas da rede:** Todo trabalho aqui desenvolvido focou-se exclusivamente na rede core 5G SA. No entanto, existe potencial de generalização para outras camadas da rede, como a rede de acesso, que tem o mesmo processo de identificação de falhas do core mas que do ponto de vista de quantidade de equipamentos é consideravelmente maior, o que significa um enorme ganho operacional para empresas de telecomunicações e uma melhora na qualidade do serviço prestado aos clientes.
- **Testar abordagem de aprendizado híbrida entre indutiva e transdutiva:** Apesar dos excelentes resultados encontrados nos testes realizados para a detecção de anomalia, é indiscutível que a abordagem transdutiva utiliza mais recursos computacionais que a indutiva. Neste trabalho, devido a introdução da camada de filtragem, o tempo de detecção foi dentro dos limites propostos de 30 minutos. No entanto caso a periodicidade seja menor, cenário possível em outras camadas de rede ou fornecedores, a abordagem transdutiva pode se tornar um gargalo. Logo, testes em uma abordagem híbrida, definindo-se assim a melhor frequência de atualização do modelo, são interessantes. N-BEATS
- **Utilização do NWDAF:** Esta nova função de rede, quando implementada, visa auxiliar toda a obtenção e análise de dados, assim como devolver a rede informações analisadas para tomada de decisão. Seria interessante implementar o trabalho aqui desenvolvido utilizando este equipamento, visando entender gargalos de desempenho assim como utilização de recursos da própria rede. Além disso, a próxima geração, a rede 6G, já está em desenvolvimento e provavelmente irá contar com este elemento desde sua concepção. Ou seja, ter o framework aqui desenvolvido integrado com o NWDAF poderia evitar problemas de desempenho logo no início da operação do novo padrão.

- **Utilizar o filtro apresentado como uma camada de caracterização:** A camada de filtro apresentada neste trabalho tem como finalidade diminuir a quantidade de séries temporais a serem analisadas pela camada de detecção de anomalia, no entanto existe mais potencial nesta etapa. Sabendo algumas características de cada série, como tendência e sazonalidade, seria possível escolher um algoritmo para detecção que melhor atenda a estas características, podendo assim melhorar o desempenho da detecção.
- **Root Cause Analysis:** Identificar um evento anômalo logo no seu início traz grande agilidade as áreas de operação e manutenção, no entanto ainda existe um gargalo importante a ser resolvido, o da identificação da causa raiz do problema. Em muitos casos esta etapa pode ser ainda mais demorada, envolvendo muitas vezes a intervenção de fornecedores. Realizar pesquisas nesta próxima etapa do Zero Touch Network ofereceria ainda mais agilidade no processo de recuperação da rede em uma eventual falha.
- **Self Healing:** Correções de falha em muitos casos envolve alterações de configuração, que em sua totalidade são realizadas manualmente hoje. Neste caso, além do tempo maior em si está envolvido a inserção de novos problemas na rede, algo que poderia ser evitado com a implementação de um framework para a etapa de self healing.
- **Manutenção Preditiva:** Evitar que um evento anômalo venha a ocorrer nem sempre é viável do ponto de vista de uma rede Core. No entanto, existem sim indicadores de desempenho que se degradam com o tempo, e consequentemente poderiam ser evitados. Logo uma das possibilidades de expansão desta pesquisa é implementar um modelo de manutenção preditiva.

Referências

3GPP TS 23.288, Architecture enhancements for 5G System (5GS) to support network data analytics services. Release 16.

3GPP TS 23.288, Architecture enhancements for 5G System (5GS) to support network data analytics services. Release 17.

3GPP TS 29.520, Network data analytics services. Release 15.

Afaq A.; Haider N.; Baig M.; Khan K., Imran M.; I Razzak. Aprendizagem de máquina for 5G security: Architecture, recent advances, and challenges. **Ad Hoc Networks**, v. 123, p. 102667. Dez. 2021

Bello, Y.; Hussein, A. R.; Ulema, M.; Koilpillai, J. On Sustained Zero Trust Conceptualization Security for Mobile Core Networks in 5G and Beyond, **IEEE Transactions on Network and Service Management**, v. 19, n. 2, p. 1876-1889. Jun. 2022.

Benzaid, C.; Taleb T. AI-Driven Zero Touch Network and Service Management in 5G and Beyond: Challenges and Research Directions. **IEEE Network**, v. 34, n. 2, p. 186-194. Fev. 2020.

Carter, K.M.; Strelein, W.W. Probabilistic reasoning for streaming anomaly detection. In: **2012 IEEE Statistical Signal Processing Workshop (SSP)**, IEEE, 2012 (pp. 377-380).

Chakraborty P.; Corici M.; Magedanz T. System Failure Prediction within Software 5G Core Networks using Time Series. In: **2021 IEEE International Conference on Communications Workshops (ICC Workshops)**, IEEE, 2021. p. 1-7.

Chakraborty, P., Corici, M., & Magedanz, T. A comparative study for Time Series Forecasting within software 5G networks. In: **2020 14th International Conference on Signal Processing and Communication Systems (ICSPCS)**. IEEE, 2020. p. 1-7.

Challu, C.; Olivares, K.G.; Oreshkin, B.N.; Ramirez, F.G.; Canseco, M.M.; Dubrawski, A. Nhits: Neural hierarchical interpolation for time series forecasting. In: **Proceedings of the AAAI Conference on Artificial Intelligence**, Junho 2023 (Vol. 37, No. 6, pp. 6989-6997).

Cheng, Z.; Zou, C; Dong, J. Outlier detection using isolation forest and local outlier factor. In: **Proceedings of the conference on research in adaptive and convergent systems**, 2019 (pp. 161-168).

Chouman, A.; Manias, D. M.; Shami, A. Towards Supporting Intelligence in 5G/6G Core Networks: NWDAF Implementation and Initial Analysis. **arXiv preprint arXiv:2205.15121**. Maio. 2022.

Ferreira D.; Senna C.; Sargent S. Distributed Real-time Forecasting Framework for IoT Network and Service Management. In: **NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium**, IEEE, 2020. p. 1-4.

Hariri, S.; Kind, M.C.; Brunner, R.J. Extended isolation forest. **IEEE transactions on knowledge and data engineering**, 2019, 33(4), pp.1479-1489.

He, Z.; Xu, X.; Deng, S. Discovering cluster-based local outliers. **Pattern recognition letters**, 2023, 24(9-10), pp.1641-1650.

Jeon, Y.; Jeong, H.; Seo, S.; Kim, T.; Ko, H.; Pack, S. A Distributed NWDAF Architecture for Federated Learning in 5G. In: **2022 IEEE International Conference on Consumer Electronics (ICCE)**. IEEE, 2022. p. 1-2.

Kim, T.; Kim, J.; Ko, H.; Seo, S.; Jeon, Y.; Jeong, H.; Lee, S.; Pack, S. An Implementation Study of Network Data Analytic Function in 5G. In: **2022 IEEE International Conference on Consumer Electronics (ICCE)**. IEEE, 2022. p. 1-3.

Lee, S.; Lee, J.; Kim, T.; Jung, D.; Cha, I.; Cha, D.; Ko, H.; Pack, S. Design and Implementation of Network Data Analytics Function in 5G. In: **2022 13th International Conference on Information and Communication Technology Convergence (ICTC)**. IEEE, 2022. p. 757-759.

Li, Z.; Zhao, Y.; Botta, N.; Ionescu, C.; Hu, X. COPOD: copula-based outlier detection. In: **2020 IEEE international conference on data mining (ICDM)**, IEEE, 2020 (pp. 1118-1123).

Lingga, P.; Kim, J. J.; Jeong, J. P. Intent-Based Network Management in 6G Core Networks. In: **2022 13th International Conference on Information and Communication Technology Convergence (ICTC)**. IEEE, 2022. p. 760-762.

Lv, Z.; Singh, A. K.; Li, J. Deep learning for security problems in 5G heterogeneous networks. **IEEE Network**, v. 35, n. 2, p. 67-73. Jan. 2021.

Manias, D. M.; Chouman, A.; Shami, A. A Model Drift Detection and Adaptation Framework for 5G Core Networks. In: **2022 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)**. IEEE, 2022. p. 197-202.

Niu, Y.; Zhao, S.; She, X.; Chen, P. A Survey of 3GPP Release 18 on Network Data Analytics Function Management . In: **2022 IEEE/CIC International Conference on Communications in China (ICCC Workshops)**. IEEE, 2022. p. 146-151.

Oreshkin, B.N.; Carpow, D.; Chapados, N.; Bengio, Y. NBEATS: Neural basis expansion analysis for interpretable time series forecasting. **CoRR abs/1905.10437** (2019).

Panev, S.; Latkoski, P. SDN-based failure detection and recovery mechanism for 5G core networks. **Transactions on Emerging Telecommunications Technologies**, v. 31, n. 2, p. e3721, 2020.

Park, S.; Kim, D.; Park, Y.; Cho, H.; Kim, D.; Kwon, S. 5G Security Threat Assessment in Real Networks. **Sensors**, v. 21, n. 16, p. 5524, Ago. 2021.

Reshma, T. R.; Azath, M. Improved self-healing technique for 5G networks using predictive analysis. **Peer-to-Peer Networking and Applications**, v. 14, n. 1, p. 375-391, 2021.

Schmidl, S.; Wenig, P.; Papenbrock, T. Anomaly detection in time series: a comprehensive evaluation. **Proceedings of the VLDB Endowment**, 2022, 15(9), pp.1779-1797.

Sedjelmaci H. Cooperative attacks detection based on artificial intelligence system for 5G networks. **Computers & Electrical Engineering**, v. 91, p. 107045. Maio. 2021.
Sevgican, S., Turan, M., Gökarslan, K., Yilmaz, H. B., & Tugcu, T. Intelligent network data analytics function in 5G cellular networks using aprendizagem de máquina. **Journal of Communications and Networks**, v. 22, n. 3, p. 269-280. Junho. 2020.

Siffer, A.; Fouque, P.A.; Termier, A; Largouet, C. Anomaly detection in streams with extreme value theory. In **Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining**, 2017 (pp. 1067-1075).

Tang, J.; Chen, Z.; Fu, A.W.C.; Cheung, D.W. Enhancing effectiveness of outlier detections for low density patterns. In: **Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings**, Springer Berlin Heidelberg, 2002, 6 (pp. 535-548).

Tang, Q.; Ermis, O.; Nguyen, C. D.; De Oliveira, A.; Hirtzig, A. A Systematic Analysis of 5G Networks With a Focus on 5G Core Security. **IEEE Access**, v. 10, p. 18298-18319. Fev. 2022.

Terra A.; Inam R.; Baskaran S.; Batista P.; Burdick I.; Fersman E. Explainability Methods for Identifying Root-Cause of SLA Violation Prediction in 5G Network. In: **GLOBECOM 2020 - 2020 IEEE Global Communications Conference**, IEEE, 2020. p. 1-7.

Wei, Y.; Peng, M.; Liu, Y. Intent-based networks for 6G: Insights and challenges. **Digital Communications and Networks**, v. 6, n. 3, p. 270-280. Ago. 2020.

Yu, Y.; Zhu, Y.; Li, S.; Wan, D. Time series outlier detection based on sliding window prediction. **Mathematical problems in Engineering**, 2014.

Zhou, S.; Liu, X.; Fu, D.; Cheng, X.; Fu, B.; Zhao, Z. 5G Core Network Framework Based on Artificial Intelligence. In: **2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)**, IEEE, 2020. p. 1460-1464.

Zhou, Z.G.; Tang, P. Improving time series anomaly detection based on exponentially weighted moving average (EWMA) of season-trend model residuals. In: **2016 IEEE**

International Geoscience and Remote Sensing Symposium (IGARSS), IEEE,
2016 (pp. 3414-3417).

| FOLHA DE REGISTRO DO DOCUMENTO | | | |
|--|--------------------------------|--|-------------------------|
| 1. CLASSIFICAÇÃO/TIPO DP | 2. DATA 30 de abril de 2024 | 3. REGISTRO N° DCTA/ITA/DP-031/2024 | 4. N° DE PÁGINAS 101 |
| 5. TÍTULO E SUBTÍTULO: Identificação de eventos disruptivos em redes core 5G SA | | | |
| 6. AUTOR(ES): Michel Santos da Silva | | | |
| 7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica – ITA | | | |
| 8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: 1. Detecção de Anomalia em Séries temporais. 2. Engenharia de Feature. 3. Modelos de Aprendizado. | | | |
| 9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Sistemas de comunicação móvel; Estruturas (processamento de dados); Análise de séries temporais; Comunicação sem fio; Aprendizagem (inteligência artificial); Avaliação de ameaças; Segurança de dados; Computação. | | | |
| 10. APRESENTAÇÃO: (X) Nacional () Internacional ITA, São José dos Campos. Curso de Mestrado Profissional em Computação Aeronáutica. Área de Sistemas de Computação. Orientador: Prof. Dr. Lourenço Alves Pereira Júnior. Defesa em 26/04/2024. Publicada em 2024. | | | |
| 11. RESUMO: A quinta geração das redes de telefonia móvel, a rede 5G, traz consigo grandes expectativas para novas aplicações devido a sua baixa latência, alta taxa de transferência de dados e grande disponibilidade. Esta nova arquitetura é muito mais flexível e escalonável, sendo a mesma baseada em tecnologias como Network Function Virtualization (NFV), Software-Defined Networking (SDN) e Service-Based Architecture (SBA), diferente das gerações anteriores que a expansão da rede era associada a alterações de hardware. No entanto, não houve melhorias no processo de monitoração da rede, o que abre brechas para a não identificação de eventos disruptivos que podem gerar interrupção do serviço. A rede core centraliza todo o tráfego, sendo responsável por funções como autenticação e gerenciamento de sessão dos usuários, sendo essa o cerne da nova especificação. Uma falha nesta camada da rede é capaz de deixar todo o sistema indisponível, logo o tempo de identificação de eventos disruptivos na rede deve ser o mais breve e assertivo possível. Monitorar completamente esta parte da rede traz desafios devido ao grande número de indicadores de desempenho disponíveis, cada um destes com um perfil de comportamento específico e mutável no tempo. Este trabalho propõe um framework cuja finalidade vai da diminuição da dimensão do que precisa ser analisado (engenharia de característica) à detecção de eventos disruptivos utilizando modelagem em séries temporais e métodos de inteligência artificial não supervisionado. Para a validação do framework proposto todos os testes são realizados com dados reais coletados de uma rede comercial em operação. Serão apresentados neste trabalho os resultados obtidos, indo desde as métricas tradicionais utilizadas em aprendizagem de máquina até o tempo gasto durante o processo, assim como a viabilidade de implementação do framework em uma grande empresa de telecomunicações. Por fim, também serão debatidas possibilidades de trabalhos futuros. | | | |
| 12. GRAU DE SIGILO: (X) OSTENSIVO () RESERVADO () SECRETO | | | |