UNIVERSIDADE FEDERAL FLUMINENSE

**FÁBIO OLIVEIRA DOS SANTOS**

# Uso de algoritmos de aprendizado de máquina na resolução de falhas em redes móveis

NITERÓI

2019

UNIVERSIDADE FEDERAL FLUMINENSE

**FÁBIO OLIVEIRA DOS SANTOS**

# Uso de algoritmos de aprendizado de máquina na resolução de falhas em redes móveis

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Orientadora:
NATALIA CASTRO FERNANDES

NITERÓI

2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

D722u Dos santos, Fábio Oliveira
Uso de algoritmos de aprendizado de máquina na resolução de falhas em redes móveis / Fábio Oliveira Dos santos ; Natalia Castro Fernandes, orientadora. Niterói, 2019.
100 f. : il.

Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: http://dx.doi.org/10.22409/PPGEET.2019.m.07519655709

1. Rede de comunicação de computadores. 2. Aprendizado de Máquina. 3. Produção intelectual. I. Fernandes, Natalia Castro, orientadora. II. Universidade Federal Fluminense. Escola de Engenharia. III. Título.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

FÁBIO OLIVEIRA DOS SANTOS

USO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA NA RESOLUÇÃO DE FALHAS EM REDES MÓVEIS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Telecomunicações da Universidade Federal Fluminense como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica e de Telecomunicações. Área de concentração: Sistemas de Telecomunicações

Aprovada em 30 de maio de 2019.

BANCA EXAMINADORA

---

Profª. NATALIA CASTRO FERNANDES, D.Sc.
Orientadora, Universidade Federal Fluminense (UFF)

---

Prof. MIGUEL ELIAS MITRE CAMPISTA, D.Sc.
Universidade Federal do Rio de Janeiro (UFRJ)

---

Profª. DIANNE SCHERLY VARELA DE MEDEIROS, D.Sc
Universidade Federal Fluminense (UFF)

Niterói
2019

# Agradecimentos

Agradeço a Deus pela permissão em realizar esse trabalho e presença nesse mundo.

À minha esposa Monique, pelo companheirismo, paciência, suporte e incentivo necessários para conclusão desse trabalho. Todo meu amor e gratidão.

Ao meu filho Caio Gabriel, que mostrou toda minha incapacidade de entender pequenas e simples coisas. E que não tenho como acelerar o tempo do qual sou apenas refém.

Aos meus pais Ana Maria e Luiz Carlos, que fizeram da minha vida o projeto de vida deles, me mostrando o caminho reto e direção segura.

Ao meu irmão Renan, que serviu de fonte de inspiração e perseverança.

À minha orientadora Natalia Castro, que foi uma verdadeira amiga e mentora, por generosamente compartilhar seu conhecimento e pela sua paciência e dedicação.

Finalizando, gostaria de agradecer a todos mais que eu não tenha citado nesta lista de agradecimentos, mas que de uma forma ou de outra contribuíram para a minha dissertação.

Cada sonho que você deixa pra trás, é um pedaço do seu futuro que deixa de existir.
*Steve Jobs*

# Resumo

As redes de acesso que fazem a interconexão das antenas das redes móveis com o núcleo da rede são de grande complexidade, por unirem diferentes tecnologias de comunicação, com requisitos muito distintos, sobre uma mesma infraestrutura de telecomunicações. A operação dessas redes se torna especialmente difícil, pois é comum observar falhas em serviços que não são identificadas pelas ferramentas de monitoração da rede. Assim, essa dissertação descreve o projeto, a implementação e um estudo de caso do sistema Monitoramento Avançado de REdes MÓveis de múlTiplas geraçÕes (MAREMOTO). O sistema proposto utiliza aprendizado de máquina para reconhecimento automatizado de causas de falhas redes IP/MPLS. A complexidade inerente dos serviços que usam o IP/MPLS atualmente dificulta encontrar uma solução de modelo fixo para todos os serviços que são executados pela rede. Além disso, a inspeção manual é extremamente dispendiosa e requer uma força de trabalho altamente qualificada, tornando-se impraticável à medida que o problema é dimensionado. Assim, o MAREMOTO coleta informações sobre roteadores e serviços, levantando o estado das variáveis de rede e correlacionando-os com as falhas que são observadas nos serviços. Uma coleta inicial foi realizada para avaliar diferentes algoritmos de aprendizado de máquina, observando quais são mais eficientes na detecção das causas das falhas de serviço. Assim, o MAREMOTO converte problemas de identificação das causas de falhas, ou seja, a tarefa de detectar e diagnosticar falhas em um problema de classificação de padrão viável. O estudo de caso realizado em uma rede de acesso em operação utilizando o sistema proposto mostrou sua potencialidade para auxiliar na detecção da origem do problema, o que permitiu propor uma solução viável para o problema com maior rapidez e efetividade.

**Palavras-chave**: gerência de redes, redes móveis, aprendizado de máquina, detecção e diagnóstico de falhas.

# Abstract

Access networks that interconnect the antennas of mobile networks with the network core are very complex because they combine different communication technologies with very different requirements over the same telecommunications infrastructure. The operation of these networks becomes especially difficult, as it is common to observe faults in services that cannot be identified by the standard network monitoring tools. Thus, this dissertation describes the design, implementation, and a case study of the Advanced Monitoring System for Multi-Generation Mobile Networks (MAREMOTO). The proposed system uses machine learning for automated recognition of failure causes in IP/MPLS networks. The inherent complexity of services using IP/MPLS currently makes it difficult to find a fixed-model solution for all services that run across the network. In addition, manual inspection is extremely costly and requires a highly skilled workforce, making it impractical as the problem is dimensioned. Thus, MAREMOTO collects information about routers and services, raising the state of the network variables and correlating them with the faults that are observed in the services. An initial data gathering was performed to evaluate different machine learning algorithms, observing which is more efficient in detecting the causes of service failures. Thus, MAREMOTO converts problems of identifying the causes of failures, that is, the task of detecting and diagnosing failures, in a problem of viable standard classification. The case study was carried out in an operational access network using the proposed system and it showed the system potential to assist in the detection of the origin of the problem. Indeed, the system made it possible to propose a viable solution to the service failure in a faster and more effective way.

**Keywords**: network management, mobile networks, machine learning, fault detection and diagnosis.

# Lista de Figuras

# Lista de Tabelas

# Lista de Abreviaturas e Siglas

| | | |
|---|---|---|
| AdaBoost | : | Adaptative Boosting; |
| AMPS | : | Advanced Mobile Phone Service; |
| ASIC | : | Application Specific Integrated Circuits; |
| BSC | : | Base Station controller; |
| BTS | : | Base transceiver station; |
| CDMA | : | Code Division Multiple Access; |
| CEM | : | Circuit Emulation; |
| CEoP | : | Circuit Emulation over Packet; |
| CLI | : | Command Line Interface; |
| CPU | : | Central Processing Unit; |
| CSV | : | Comma-separated values; |
| DECT | : | Digital Enhanced Cordless Telecommunication; |
| DR | : | Designated Router; |
| DT | : | Direct tunnel; |
| EPC | : | Evolved Packet Core; |
| ETACS | : | Extended Total Access Comunications System; |
| FDMA | : | Frequency Division Multiple Access; |
| GGSN | : | Gateway GPRS Support Node; |
| GPRS | : | General Packet Radio Services; |
| GPU | : | Graphics Processing Unit; |
| GSM | : | Global System for Mobile Communications; |
| GTP | : | GPRS tunnelling protocol; |
| HLR | : | Home Location Register; |
| HSS | : | Home Subscriber Server; |
| ICMP | : | Internet Control Message Protocol; |
| IP | : | Internet Protocol; |
| IPSLA | : | Internet Protocol Service Level Agreement; |
| LDP | : | Label distribution protocol; |
| LSP | : | Label-Switched Path; |
| LTE | : | Long Term Evolution; |
| MIB | : | Management Information Base; |
| MME | : | Home Subscriber Server; |
| MPLS | : | MultiProtocol Label Switching; |
| MS | : | Mobile Station; |

MSC : Mobile Switching System;
MTU : Maximum Transmission Unit;
NFV : Network Function Virtualization;
NMT : Nordic Mobile Telecomunications;
NTP : Network Time Protocol;
OIDS : Object identifiers;
OSPF : Open Shortest Path First;
PCM : Puse-Code Modulation;
PCU : Packet Control Unit;
PDC : Personal Digital Celular;
PDH : Plesiochronous Digital Hierarchy;
PDN-G : Packet Data Network Gateway;
PSTN : Public Switched Telephone Network;
PTP : Precision Time Protocol;
PWS : PseudoWires;
QOS : Qualidade de Serviço;
RAM : Random Access Memory;
RFC : Request for Comments;
RID : Router ID;
RNC : Radio Network Controller;
SAE : System Architecture Evolution;
SAToP : Structure-agnostic TDM over Packet;
SDH : Synchronous Digital Hierarchy;
SDN : Software Defined Network;
SDR : Software Defined Routers;
SGSN : Serving GPRS Support Node;
SGW : Serving Gateway;
SNMP : Simple Network Management Protocol;
SMS-C : Short Message Service Center;
SON : Self-organized network;
SPF : Shortest Path Firs;
SRAM : Static Random Access Memory;
TACS : Total Access Communications System;
TCP : Transmission Control Protocol;
TDM : Time-division multiplexing;

| | | |
|---|---|---|
| TDMA | : | Time Division Multiple Access; |
| UDP | : | User Datagram Protocol; |
| UFF | : | Universidade Federal Fluminense; |
| WCDMA | : | Wide-Band Code-Division Multiple Access; |
| 3GPP | : | 3rd Generation Partnership Project; |

# Sumário

# Capítulo 1

# Introdução

As redes de telecomunicações atuais estão se tornando cada vez mais complexas em função da grande quantidade de elementos, protocolos e sistemas necessários para prover a infraestrutura necessária para as aplicações atuais. Essa complexidade se torna mais visível nas redes móveis, que precisam possuir tecnologia suficiente para acomodar e transportar todas as gerações de redes móveis, bem como serem escaláveis para gerações futuras. A parte da rede que interliga as antenas entre si bem como aos seus controladores são chamadas de rede de acesso. É nessa parte da rede que temos uma grande diversidade de protocolos, assim como um grande número de elementos e meios de transmissão diferenciados. O fato de possuir um grande número de elementos interconectados por diferentes meios de transmissão, por si só, já é de grande complexidade, dificultando o projeto e atualizações dessa rede. Por se tratar de rede de acesso, os roteadores usados são de média ou baixa capacidade, sendo que alguns apresentam restrições em relação ao tamanho da tabela de roteamento que podem comportar outros com o tipo de tráfego que conseguem comutar etc. Além de possuir uma grande complexidade em relação ao seu projeto, tais redes também precisam acomodar diferentes tipos de interfaces e tráfegos vindos das diferentes gerações de redes movem, como pode ser visto na Figura 1.1.

Dada a grande quantidade de elementos conectados por uma complexa estrutura de roteamento, tratando diferentes tipos de tráfego em diferentes interfaces, percebe-se o grau de dificuldade na operação das redes de acesso. Assim, não é incomum ocorrer quedas totais ou parciais dos seus serviços e/ou os mesmos apresentarem quedas de qualidade sem que as ferramentas de apoio à gestão e operação dessa rede possam fornecer informações precisas em relação às falhas que ocorreram. Por conta do grau de dificuldade na operação dessa rede, surge a necessidade de uso de outras ferramentas no auxílio operacional.

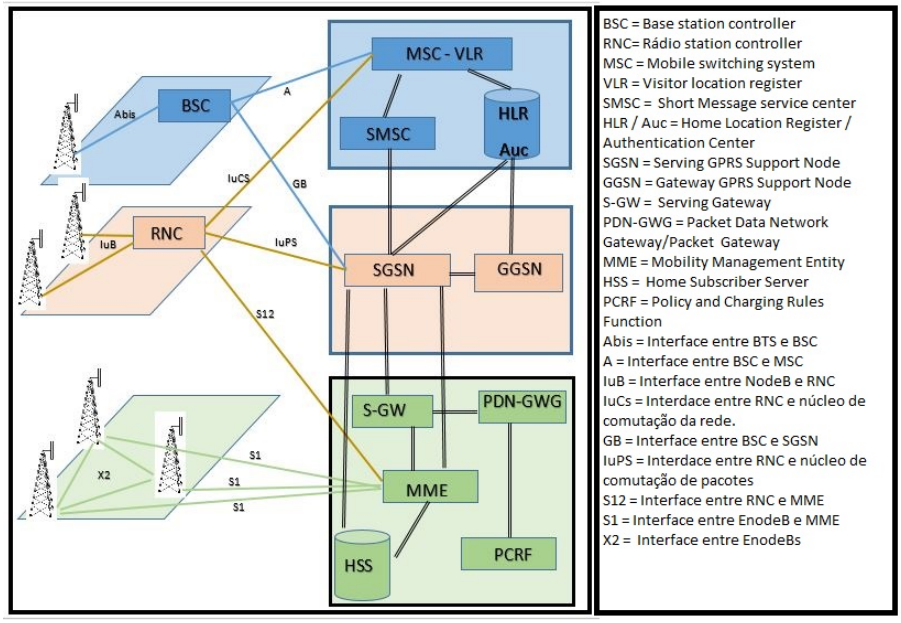



Figura 1.1: Topologia básica com as principais interfaces da rede de acesso.

## 1.1 Motivação

Um dos serviços mais complexos de incorporar nas redes atuais é o serviço legado de telefonia móvel de segunda geração, conhecido pela sigla 2G. A complexidade do mesmo está atrelada a necessidade de trafegar a estrutura 2G que utiliza o *Time-division multiplexing*(TDM) dentro das redes de transporte de pacotes. A acomodação do tráfego 2G (TDM) sobre uma rede de pacotes é provida pelos serviços de emulação de circuitos, também conhecida como *Circuit Emulation*(CEM), e essa acomodação se faz necessária em função dos diversos usuários corporativos *Machine-to-machine*(M2M) que possuem um alto valor agregado e ainda utilizam tecnologia 2G. Para uma correta emulação de um circuito TDM dentro de uma rede de pacotes, os controladores CEM presentes dentro dos roteadores terminadores dos circuitos de emulação geram um pacote de 256 bytes a cada 1 ms. Esse padrão de geração gera um perfil de tráfego bem definido, comportado e sem grandes alterações. Tal emulação é sensível à variação de atraso e necessidade de ordenação de pacotes, bem como é necessário que os relógios internos de cada roteador CEM estejam sincronizados. Para isso, o protocolo de *Precision Time Protocol*(PTP) pode ser utilizado. A dificuldade na operação desse tipo de serviço ocorre, porque variações de atraso na ordem milissegundos são suficientes para levar ao descarte do pacote e a falhas na geração do quadro TDM no controlador, o que gera degradação no sinal *Puse-Code Modulation*(PCM) gerado.

Identificar e correlacionar falhas nos serviços de CEM com falhas na rede de trans-

porte de acesso pode se tornar uma tarefa complexa, pois as ferramentas de avaliação de desempenho atuais que funcionam com precisão de milissegundos não são de fácil acesso e nem sempre estão disponíveis em toda porção da rede. Atualmente, os elementos que compõem as redes IP/MPLS usadas no transporte de tráfego no acesso das redes móveis usam o protocolo *Simple Network Manamage protocol* (SNMP) e acessos via linha de comando para coleta das informações de gerência relacionadas aos diversos contadores e estatísticas disponíveis nos elementos. Como tais informações são geradas em cada elemento de forma individual e necessitam de recursos de processamento e memória em cada elemento de rede, não é viável que informações sejam geradas em uma escala temporal na ordem de milissegundos. Mesmo o uso de comandos para testes nos meios de transmissão e/ou elementos de rede em momentos de falhas não são eficazes na constatação e relação direta da origem dos problemas de CEM, visto que os elementos de rede usam recursos centralizados para execução e respostas de tais comandos, de forma que normalmente comandos que realizam testes de conectividade ficam limitados a testes executados em segundos não milissegundos.

Embora existam protocolos mais rebuscados para gerência dos serviços de rede, esses protocolos precisam estar implementados dentro dos elementos de rede e serem viáveis mesmo com as diversas limitações de recursos desses elementos. Um exemplo desses protocolos é o *Internet Protocol Service Level Agreement* (IPSLA), que usa duas marcações de tempo, possibilitando a medição de perdas e atrasos na ordem de milissegundos. A Figura 1.2 mostra o funcionamento básico desse protocolo. O IPSLA inicia criando os pacotes na *Central Processor Unit*(CPU) e os envia para interface adequada a fim de que os mesmo cheguem ao roteador de destino, nessa interface o pacote recebe a primeira marcação de tempo (timetamp) e em seguida é enviado para o roteador de destino. Ao chegar à interface do roteador de destino o mesmo recebe uma nova marcação de tempo, é enviado para CPU do roteador de destino que realiza o processamento do mesmo e então uma resposta é gerada para o roteador de origem, contendo as marcações anteriores. Essa resposta passa pela interface de saída em direção ao roteador de origem do IPSLA, onde recebe a segunda marcação do roteador de destino e em seguida é enviado ao roteador de origem. Dessa forma as duas macacões de tempo realizadas no roteador de destino permite verificar separadamente o atraso no envio do pacote e a perda de pacote nos dois sentidos, bem como o tempo de processamento do mesmo no roteador de destino, o que auxilia a avaliação dos dados de rede para os serviços CEM, uma vez que o comportamento da rede é unidirecional. Para o correto funcionamento do IPSLA, ambos os roteadores (origem e destino) devem estar com a data e hora sincronizadas. Isso pode ser obtido com

o uso do *Network Time Protocol* (NTP), usando um mesmo servidor de NTP para ambos os roteadores.

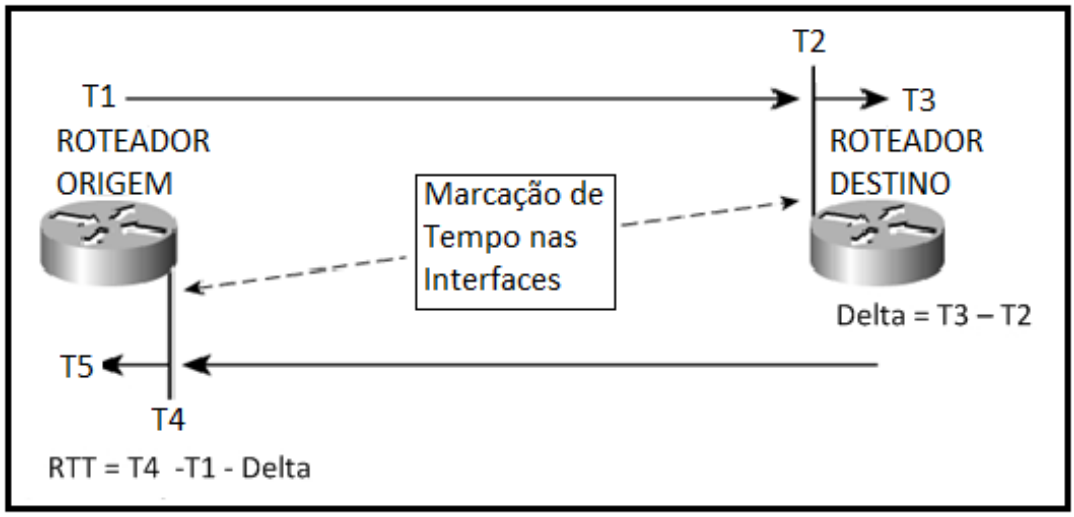



Figura 1.2: IPSLA e o uso de marcação de dois timestamp no roteador de destino.

Porém, mesmo com informações na ordem de milissegundos, nem sempre é possível diagnosticar e relacionar problemas em meios de transmissão ou elementos de rede com falhas no serviço de CEM. Tal dificuldade advém do fato de que as *probes* do *Internet protocol service level agreement*(IPSLA), configuradas internamente nos elementos, também usam recursos dos elementos de rede, possuindo tempos de execução de operação na ordem de segundos e tempo entre envio de pacotes na ordem de 20ms. Tais restrições de tempo inviabilizam analisar as falhas nos serviços de CEM apropriadamente.

## 1.2 Objetivos

Esse trabalho avalia o uso de algoritmos de aprendizado de máquina para obtenção de uma melhor monitoração dos serviços críticos, como os serviços CEM, que requerem alto desempenho da rede, embora as ferramentas de gestão tradicionais não forneçam os dados adequados para a correta operação da rede. Os serviços CEM geram automaticamente informações de qualidade de cada controlador TDM nos mesmos intervalos de tempo que os controladores removem as informações de seus espaços de memória para reconstruírem o sinal TDM. Hoje, esse intervalo é de 5ms. Vale ressaltar que a geração dessas informações na ordem de 5ms ocorre sem o uso demasiado dos recursos nos elementos de rede, em função da geração das estatísticas nos próprios controladores e não nos recursos centralizados dos elementos de rede, visto que cada controlador normalmente possui um *Application Specific Integrated Circuits*(ASIC) próprio.

Devido aos serviços CEM disponibilizarem informações de desempenho na ordem de 5ms e sua gestão ser complexa, a correlação com as falhas não diretamente relacionadas no ambiente de rede se torna muito difícil. Isso faz com que os serviços de emulação de circuitos via CEM sejam uma excelente opção para avaliação de sistemas de aprendizado de maquina na verificação de falhas nos ambientes de rede ou das variáveis da rede e suas relações com o a desempenho do serviço.

Por essas razões, os objetivos desse trabalho são:

- levantamento de requisitos e definição de arquitetura de ferramenta para monitoramento de serviços críticos em redes de telefonia móvel;
- desenvolvimento de um sistema de monitoramento avançado baseado em aprendizado de máquina para detecção de causas de falhas de serviços da rede, chamado de Monitoramento Avançado de REdes MÓveis de múlTiplas geraçÕes (MAREMOTO).
- realização de coletas de informações de desempenho dos serviços CEM e de informações de rede como: topologia da rede usada por cada serviço, variáveis de rede relacionadas com a topologia, informações dos elementos de rede etc. para armazenamento em banco de dados;
- utilização dos dados coletados em algoritmos de aprendizado de máquina, a fim de diagnosticar quais variáveis de rede possuem maior correlação com os das falhas nos serviços CEM;
- desenvolvimento de um estudo de caso utilizando o MAREMOTO em uma rede de acesso real para análise de um problema em serviço crítico.

Para realizar a coleta dos dados, um conjunto de scripts foi criado utilizando a linguagem Python, a fim de criar um diretório para cada coleta, realizar as coletas propriamente ditas e armazenar as informações correlacionadas no seu respectivo diretório. Os resultados indicam que os algoritmos de aprendizado de maquina são capazes de encontrar, com grande assertividade, as relações entre as diversas variáveis de rede e os status dos serviços. Os testes foram executados em uma rede em operação, e observou-se grande eficácia na classificação dos serviços, na relação estabelecida pelos algoritmos, bem como na normalização dos serviços quando se atua nas variáveis indicadas pelos algoritmos de aprendizado de máquinas.

## 1.3 Organização do trabalho

Essa dissertação está organizada em sete capítulos, conforme descrito a seguir.

O Capítulo 2 descreve como as redes de comutação de pacotes são implementadas a fim de alcançarem os requisitos dos diversos serviços prestados pelas operadoras. Atualmente tais operadoras utilizam ao máximo uma mesma infraestrutura de rede para prestação de diversos serviços. De forma que a rede precisa ser implementada para prover todos os recursos necessários para todos os serviços.

O Capítulo 3 apresenta uma breve introdução sobre os algoritmos de aprendizado de máquinas, bem como demonstra como os mesmos podem auxiliar no desenvolvimento de novos sistemas de gestão das redes atuais trazendo grandes benefícios para o diagnostico de falhas e degradações nos serviços prestados pela rede.

O Capítulo 4 discute os atuais usos e estudos realizados a respeito dos algoritmos de aprendizado de máquinas e dos mesmos em ambientes de rede. Onde se percebe que não os algoritmos de aprendizado de maquinas estão recebendo uma grande atenção, como também o uso dos mesmos nos ambientes de rede.

O Capítulo 5 apresenta uma proposta de uso de algoritmos de aprendizado de maquinas nos ambientes de rede. Onde os mesmos podem ser utilizados na correlação das diversas variáveis de rede e os status de indisponibilidade ou degradação dos serviços.

O Capítulo 6 descreve os testes de avaliação de uso dos algoritmos de aprendizado de maquinas executados em uma rede em operação com o intuído de obter correlações entre as diversas variáveis de rede e os status dos serviços que utilizam a mesma.

Por fim, o Capítulo 7 conclui o trabalho, refletindo os resultados dos testes e o Capítulo 8 indica futuras implementações e alterações na ferramenta proposta.

# Capítulo 2

# Redes de Telecomunicações Móveis

O tráfego de dados em redes móveis cresceu 71% no ano de 2017, alcançando a marca de 11.5 exabytes por mês no final de 2017, o que corresponde a um acréscimo de 6.7 exabytes por mês se compararmos com 2016. Hoje, as redes 4G (quarta geração de rede móvel) correspondem a 72% do tráfego total de Internet presente nas redes móveis e a tendência é que esse tráfego continue crescendo [10]. De fato, a crescente demanda de transmissão de dados norteou as evoluções das redes móveis. Hoje, o tráfego de dados já supera o tráfego de voz em diversas redes móvel no mundo. Com isso, a necessidade de melhores formas de transportar esse tráfego com segurança e transparência para o usuário resultou em novas arquiteturas de redes móveis, bem como nas redes de transporte usadas atualmente.

## 2.1 Evolução das redes móveis e suas arquiteturas

Os primeiros sistemas móveis tinham como objetivo alcançar uma grande área de cobertura através de um único transmissor de alta potência e utilizavam a técnica de acesso conhecida como *Frequency Division Multiple Access* (FDMA), onde cada usuário era alocado em uma frequência distinta. Embora essa abordagem gerasse uma cobertura muito boa, o número de usuários simultâneos era limitado, mostrando a ineficácia da rede. Como cada canal possui uma largura de banda definida e cada usuário usando efetivamente a rede recebia um canal para uso, logo se alcançava o número máximo de canais nos transmissores de alta potência, que muitas vezes possuíam áreas de cobertura na ordem de centenas de quilômetros, o que permitia poucos usuários, logo surgiu o conceito de uso de células como solução do problema de congestionamento espectral e de limitação de capacidade de usuários. Neste momento, as novas redes móveis passaram a

usar transmissores de potência mais baixa, abrangendo áreas menores de cobertura, na ordem de dezenas de quilômetros e não mais centenas, permitindo o reuso de frequências do espectro, pois, com baixa potência, canais usados em um determinado transmissor eram reutilizados em outros transmissores distantes sem problemas de interferência. Com isso, surgia o conceito de células, que são áreas de cobertura menores, originando o nome de sistema celular. O agrupamento dessas células, bem como o reuso de frequências no espectro, deu origem aos primeiros sistemas celulares. A seguir, são apresentadas as diferenças entre as gerações dos sistemas móveis e seu impacto sobre o núcleo da rede.

### 2.1.1 Primeira Geração

A primeira geração das comunicações móveis utilizava sistemas analógicos com tecnologia FDMA, onde, para cada usuário utilizando a rede, é necessário realizar a alocação de um canal de transmissão. Essa rede era projetada para trafegar somente voz. Os primeiros sistemas desenvolvidos foram o *Nordic Mobile Telecomunications* (NMT), o *Advanced Mobile Phone Service* (AMPS), o *Total Access Communications System* (TACS), o *Extended Total Access Communications System* (ETACS), o C450 e o Radicom 2000. Estes sistemas possuíam vários problemas, tais como:

- limitação de capacidade, pois a alocação um canal de voz de 64 kbits/s para cada usuário em conversação limitava muito o uso do espectro, o qual era muitas vezes completamente ocupado.
- incompatibilidade entre os sistemas, pois se utilizavam interfaces que não eram padronizadas;
- baixa qualidade nas ligações, visto que a modulação era analógica;
- ausência de segurança na transmissão das informações;
- aparelhos não portáteis, em função da grande potência necessária para transmissão entre o aparelho móvel e a antena, que poderia estar a centenas de quilômetros de distância; com isso, o aparelho móvel precisava de uma grande fonte de energia, sendo normalmente instalado em automóveis.

### 2.1.2 Segunda geração de redes móveis

Devido à baixa qualidade nas ligações, ao uso não otimizado do espectro, à falta de segurança e à necessidade de padronização para o sistema celular, bem como a crescente

demanda pelo serviço móvel, foi necessário dar início ao desenvolvimento dos sistemas digitais móveis. Com isso, deu-se início à comercialização e à utilização dos sistemas de 2ª geração, sendo os principais sistemas o *Global System for Mobile Communications* (GSM) e o *Digital Enhanced Cordless Telecommunication* (DECT) na Europa; o *Time Division Multiple Access* (TDMA), também conhecido como IS-54 e IS-136; o *Code Division Multiple Access* (CDMA IS-95) nos EUA; e o *Personal Digital Celular* (PDC) no Japão.

Alguns sistemas de segunda geração das redes móveis já permitiam baixas taxas de transmissões de dados, usando a mesma infraestrutura de voz. Os sistemas de segunda geração possuíam, normalmente, a seguinte arquitetura de rede/elementos:

- *Móbile Station* (MS) - é a estação móvel, ou seja, o aparelho celular do usuário;
- *Base transceiver station* (BTS) - é a estação base transceptora, que transmite e recebe os sinais de rádio celular. Possui todo o hardware necessário para conectar o terminal do usuário (MS) à rede móvel.
- *Base Station controller* (BSC) - é o controlador de estação base, que realiza o controle das BTS, incluindo o controle de potência do sinal de rádio transmitido, a alocação de canais e o estabelecimento de conexão para provimento de serviços aos usuários.
- *Mobile Switching System* (MSC) - é a central de comutação da rede móvel, que faz a ligação entre os usuários da rede móvel, bem como fornece uma interface entre a rede móvel e a rede de telefonia fixa (*Public Switched Telephone Network* - PSTN). É na MSC que as chamadas são comutadas entre os usuários da rede móvel e entre a rede móvel e as demais redes.
- *Home Location Register* (HLR) - é o banco de dados que armazena localização do MS e informações sobre o usuário. Assim, quando um usuário inicia uma chamada, o HLR é consultado para identificar se o usuário pode ou não realizar essa chamada. Da mesma forma, o HLR é consultado para obtenção da localização do MS, a fim de que uma chamada possa ser entregue ao usuário, no caso de recebimento de chamadas.
- *Short Message Service Center*(SMS-C) - é o centro de serviço de mensagens curtas, mais conhecidas como *Short Messages*. Já no inicio das redes celulares, o serviço

de envio de mensagens curtas de texto estavam disponíveis através dos serviços prestados pela SMS-C.

A Figura 2.1 mostra um esquema geral dos equipamentos de uma rede de segunda geração. Nas redes de segunda geração (2G), as interfaces entre BTS e BSC utilizam enlaces dentro do padrão *Pulse Code Modulation* (PCM) sendo muito comum interfaces de TDM de 2mbps/s. Contudo, normalmente, não havia interoperabilidade entre fornecedores.

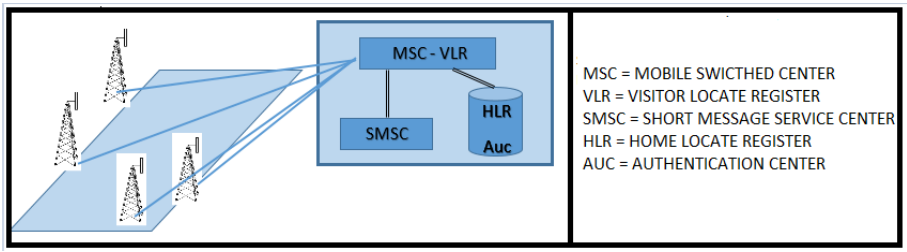



Figura 2.1: Topologia básica de interligação de serviços em redes móveis 2G.

### 2.1.3 Redes 2.5G

Com a evolução dos serviços móveis e a com a crescente demanda de serviços de dados para usuários, se fizeram necessárias melhorias nas redes 2G para melhor comportar o tráfego de dados, visto que o serviço de dados provido nessa geração utilizavam adaptações dos canais e estrutura de voz para prover os serviços de dados. Nas redes 2.5G, criou-se a infraestrutura necessária para tráfego de dados separado da estrutura de voz. A essa estrutura foi dado o nome de *General Packet Radio Services*(GPRS), onde o tráfego de dados dos usuários é enviado das BSCs para a rede GPRS via interface dedicada. Essa interface não é usada para nenhum outro tipo de tráfego e é chamada de GPRS *GB interface*. Interfaces GB, inicialmente, usavam o protocolo *frame-relay* e, posteriormente, começaram a usar o protocolo IP. Esse foi o primeiro ponto de separação do tráfego de voz e dados nas redes móveis. Dessa forma, se fez necessária a criação de uma estrutura para controle e comutação dos pacotes de usuários na rede móvel. Nas redes 2,5G, essa estrutura era composta pelos seguintes elementos:

- *Gateway GPRS Support Node* (GGSN) - provê a interconexão com outras redes, como a Internet ou as redes privadas. Desse ponto em diante, tem-se apenas o protocolo IP.
- *Serving GPRS Support Node* (SGSN) - mantém o registro da localização dos dispositivos móveis dentro da cobertura da rede móvel, além de realizar a interface entre

a rede de pacotes móvel e as BSCs via interface GB. Nas BSCs, a interface GB é disponibilizada pelo *Packet Control Unit* (PCU). É por essa interface que o SGSN envia os pacotes para um determinado MS. Entre o SGSN e os elementos da rede móvel, tem-se encapsulamentos adicionais, a fim de prover uma camada de abstração da mobilidade do usuário. Assim, entre o SGSN e o *Gateway GPRS Support Node* (GGSN), tem-se o protocolo de tunelamento chamado de textitGPRS tunnelling protocol (GTP), e, entre o SGSN e a BSC, tem-se o tunelamento da interface GB.

Na Figura 2.2, se observa a arquitetura criada para o tráfego de pacotes nas redes 2.5G.

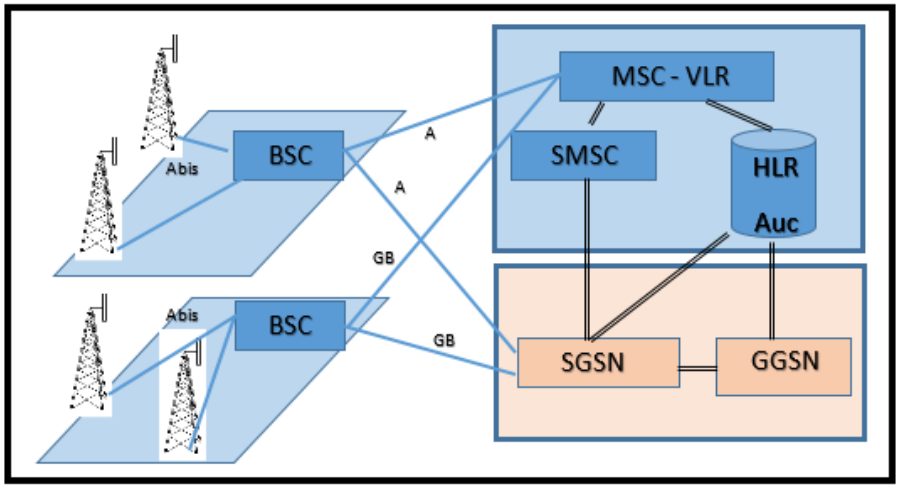



Figura 2.2: Topologia básica de interligação de serviços de pacotes em redes móveis 2,5G.

### 2.1.4 Redes 3G

Em consequência da evolução natural dos elementos e tecnologia de redes móveis, bem como a necessidade cada vez maior de conectividade, velocidade de transmissão de dados e novos serviços por parte dos usuários, uma nova arquitetura foi estabelecida para a terceira geração dos serviços móveis, como mostrado na Figura 2.3. Do ponto de vista de estrutural, não houve modificações significativas para o tráfego de pacotes dos usuários, pois os elementos SGSN e GGSN continuam realizando o mesmo tipo de controle e comutação de pacotes. Porém, na terceira geração das redes móveis, grandes avanços foram realizados na interface aérea, a fim de torná-la mais eficaz na transmissão de dados e no controle de seu uso. A BTS passou a se chamar NodeB, assumindo funções de transmissão e recepção dos sinais rádio; modulação e demodulação; codificação no canal físico com a técnica *Wide-Band Code-Division Multiple Access* (WCDMA); monitoramento de taxa de erros; controle de potência; etc.

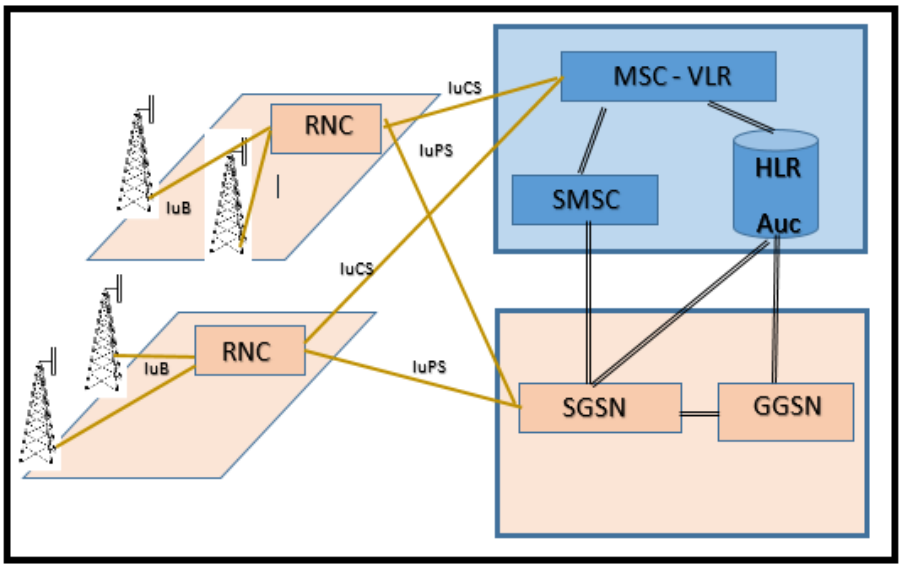



Figura 2.3: Topologia básica de interligação de serviços de pacotes em redes móveis 3G.

No 3G, a BSC passou a ser chamada de *Radio Network Controller* (RNC), apresentando como funcionalidades o controle dos recursos dos rádios; o controle de admissão; a alocação de canais; o controle de potência; o controle dos mecanismos de *handover*; criptografia; etc. Até então, parte dessas funcionalidades estava divida entra as BTS e BSCs, o que tornava o sistema mais complexo, visto que as funções realizadas pelas antenas (BTS) não eram totalmente centralizadas nas BSCs. Assim, parte do controle precisava ser distribuída. Nessa nova geração, as interfaces SGSN-RNC e GGSN-SGSN são agora interfaces puramente IP e necessitam apenas de uma conectividade IP a fim de prover os serviços necessários. No 3G, tem-se também a possibilidade da RNC enviar o tráfego de dados do usuário diretamente para o GGSN. Isso se dá após a fase de autenticação do usuário e auxilia no objetivo de alcançar taxas de transmissão de dados mais elevadas. A técnica que permite à RNC enviar todo de dados do usuário diretamente para o GGSN é chamada de *Direct tunnel* (DT).

### 2.1.5 Quarta geração de redes móveis - *Long Term Evolution* (LTE) e *System Architecture Evolution* (SAE)

O SAE é o padrão definido pelo *3rd Generation Partnership Project* (3GPP) para o núcleo de rede LTE. A rede LTE é à base da quarta geração de redes móveis. O SAE é a evolução das redes GPRS usadas nas versões anteriores, porém com algumas diferenças. Primeiramente, a arquitetura proposta pelo SAE é bem mais simples, pois todos os elementos usam o IP como protocolo de roteamento e endereçamento. Além

disso, todos os elementos da rede de acesso possuem altas taxas de dados com baixa latência.

A principal parte da arquitetura SAE é o *Evolved Packet Core* (EPC). O EPC é um sistema equivalente ao GPRS, e seus serviços são providos pelos seguintes elementos:

- *Home Subscriber Server* (HSS) - O HSS possui funções semelhantes ao HLR das gerações anteriores, visto que funciona como um banco de dados para os assinantes da rede.

- *Mobility Management Entity* (MME) - É o elemento que controla a rede de acesso, sendo responsável por gerenciar a mobilidade do usuário através dos procedimentos de *location* e *paging*. O MME também tem a tarefa de autenticar o usuário na rede, visto que possui uma interface com o HSS. O MME é o ponto de terminação da rede para questões de cifragem e proteção da integridade. Pode-se dizer que o MME realiza funções semelhantes às realizadas pelo SGSN nas arquiteturas anteriores.

- *Serving Gateway* (SGW) - É o elemento que realiza o roteamento dos pacotes de usuário, além de atuar como controlador da mobilidade durante o processo de *handover* entre as antenas da rede de quarta geração, que agora são chamadas de *eNodeBs*. O SGW gerencia a conexão de dados de *downlink* dos usuários, bem como o processo de *paging* para os aparelhos móveis que agora se chamam *User Equipment*(UEs) em estado inativo quando a conexão é estabelecida.

- *Packet Data Network Gateway/Packet Gateway*(PDN/PGW) é o responsável por roteador os pacotes IPs dos usuários da rede móvel para rede de dados externa ou interna, sendo o nó de saída e entrada do tráfego de dados de usuário. O PGW também permite a implementação de políticas de controle, filtragem de pacotes por usuário, tarifação, espelhamento de tráfego e interceptação legal.

- *Home Subscriber Server* (HSS): é a uma base de dados central, que contém informações relativas ao usuário. É consultado para verificação de diversas informações de usuários. O HSS suporta funcionalidades como gerenciamento de mobilidade, chamada e sessão, autenticação e autorização de usuários.

A arquitetura das redes 4G com seus principais elementos são mostrados na Figura 2.4.

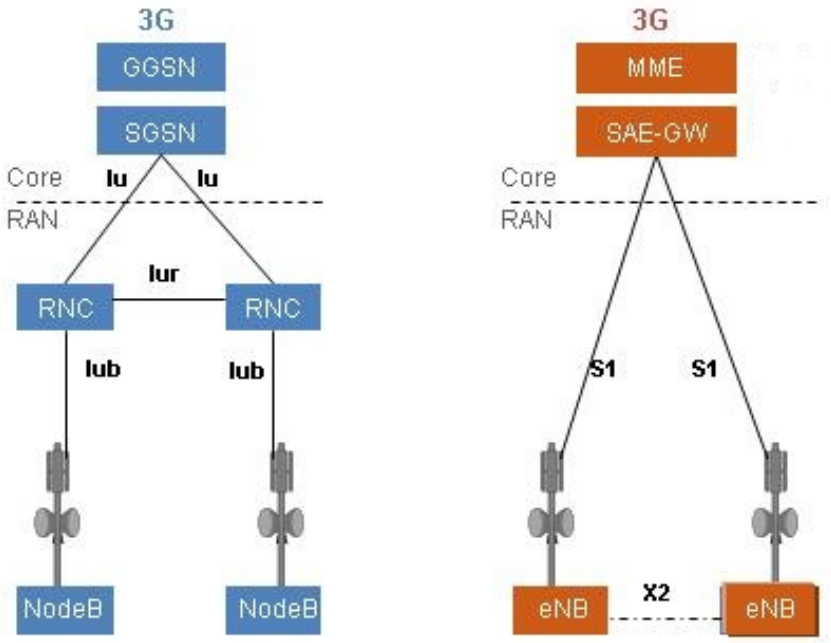



Figura 2.4: Topologia da rede 4G e sua relação com o 3G.

## 2.2 Operação simultânea das diferentes gerações das redes móveis

Mesmo com os grandes avanços realizados através da adoção do 4G, ainda existe a necessidade contínua de operação e manutenção das redes móveis de geração anteriores (2G/3G), uma vez que diversos clientes continuam operando com terminais móveis que dependem dessas tecnologias. De acordo com o relatório da GSMA [6], a tecnologia 2G ainda representa 29% das conexões mundiais de redes móveis, comparado com 28% de 3G e 43% 4G. O mesmo relatório aponta que esse conjunto de tecnologias deve coexistir por um longo período de tempo, como mostrado na Figura 2.5. Esses números incluem tanto uso tanto para equipamentos *Machine-to-Machine* (M2M) como celulares pessoais. No Brasil, empresas de telecomunicações preservam clientes M2M e mantém a rede 2G disponível com a mesma cobertura, mas provendo um serviço de maior qualidade, uma vez que os celulares deixaram de usar esta tecnologia migrando para 3G ou 4G. Logo, as redes 2G ficam quase que exclusivamente para clientes M2M. Não há, no momento, nenhum plano para desligar a rede 2G. De fato, existe grande esforço em migrar os usuários de voz para rede 3G, deixando a rede 2G para uso cada vez mais exclusivo dos equipamentos M2M.

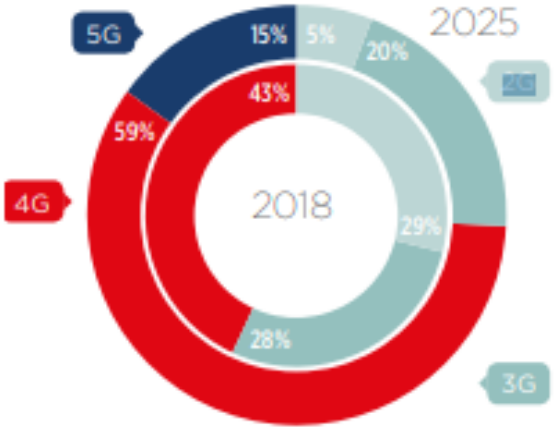



Figura 2.5: Medidas de tráfego de redes móveis em 2018 e expectativas para 2025. Fonte: [6]

Na contra mão dessas iniciativas, manter a rede 2G está se tornando mais difícil, porque os fornecedores dessa tecnologia começam a dar menos suporte. As empresas de telecomunicações brasileiras, recentemente, têm enfrentado casos de paralisação de BTSs por roubo das peças[1]. Ainda assim, não existem planos de curto prazo para desligamento da rede 2G. Dessa forma, tem-se um cenário das redes móveis como é descrito na Figura 1.1.

Com o uso das redes móveis crescendo de forma exponencial em diferentes mercados e a crescente demanda por conectividade solicitada por novos serviços, aplicações e aparelhos, fica clara a necessidade de maiores investimentos na infraestrutura das redes móveis. Tais investimentos devem considerar a manutenção e operação contínua das redes de gerações anteriores. É importante considerar que a necessidade de expansão constante, implementação de novos padrões e necessidade de manutenção das gerações anteriores trazem uma grande complexidade para manutenção das redes móveis na sua operação diária. Como mencionado no Capítulo 1, a necessidade de manter algumas gerações de redes moveis em operação sobre as redes atuais pode não ser uma tarefa simples.

Com a evolução dos elementos de comutação de pacotes dentro das redes móveis, rapidamente se percebeu que o elemento MME pertencente à quarta geração poderia realizar as funções de SGSN das redes 3G e 2,5G, assim como o elemento PGW também poderia realizar as funções de GGSN das gerações anteriores. Além disso, esses elementos possuem como requisitos de conectividade apenas redes IP. Dessa forma, a rede IP passou a ser compartilhada para os elementos de acesso, provendo conectividade entre

[1]fonte: https://www.mobiletime.com.br/noticias/25/08/2015/2g-3g-ou-4g-como-sera-o-futuro-para-m2m-iot/

os elementos da rede móvel 3G e 4G. Porém, continua ainda a necessidade de uma rede determinística para o tráfego 2G, visto que, nas BTSs legadas dessa geração de rede móvel, os equipamentos possuem apenas interfaces determinísticas, em sua grande maioria interfaces *Time-Division Multiplexing* (TDM) no padrão *Pulse-Code modulation*(PCM-30). Nesse padrão, 32 canais de 2.048 Mbps TDM são agrupados. Desses 32 canais, dois normalmente são utilizados para controle e sinalização, sobrando 30 canais disponíveis para transporte de informações úteis dos usuários.

O núcleo das redes de transporte usadas para telefonia móvel também evoluiu ao longo dos últimos anos, sendo adotadas tecnologias mais novas para comutação dos pacotes. Atualmente, as redes de transporte utilizam a tecnologia IP/MPLS (*MultiProtocol Label Switching* - MPLS) como rede de transporte para os serviços 3G e 4G. Como consequência dessa evolução, se viabilizou também a implantação de serviços de simulação de circuitos determinísticos sobre rede de pacotes IP/MPLS, fazendo com que uma rede IP/MPLS pudesse ser usada para transporte de todas as arquiteturas das redes móveis. Os serviços que permitem simular circuitos determinísticos, ou seja, circuitos TDM sobre uma rede IP/MPLS são denominados de *Circuit Emulation over Packet* e *Structure-agnostic TDM over Packet* (CEoP / SAToP), normalmente chamados de CEM. Tanto os serviços CEoP quanto o SAToP definem meios para fornecer o transporte por *Time-Division Multiplexing* (TDM)[2] através de redes de pacotes MPLS. O SAToP é o nome padronizado para o transporte não estruturado[3], enquanto o CEoP é usado frequentemente para transporte de circuitos cuja carga útil é estruturada[4]. Com o uso das tecnologias de emulação, é possível transportar os quadros TDM, normalmente transportados por uma rede determinística *Plesiochronous Digital Hierarchy*(PDH) ou *Synchronous Digital Hierarchy*(SDH), em uma rede de pacotes IP/MPLS, evitando, assim, a onerosa manutenção de numerosos circuitos físicos de uma rede determinística. De fato, o transporte tradicional TDM usando redes determinísticas implica em uma grande quantidade de circuitos dedicados providos fisicamente através meios de transmissão dedicados e muitas vezes não otimizados.

Na Figura 2.6, é mostrado um diagrama de uma topologia típica de uma rede móvel

[2]TDM é uma forma de multiplexação onde é realizada a transmissão simultânea de diversos sinais em um mesmo meio de transmissão, porém essa transmissão é realizada em tempos diferentes, de forma que o processo inverso (demultiplexação) possa ser realizado no receptor a fim de recuperar os sinais transmitidos simultaneamente. Esse processo é possível pois cada canal disponibilizado pelo processo de multiplexação possui um tempo bem definido para ser usado.

[3]O transporte não estruturado se dá quando o serviço de emulação não tem conhecimento da estrutura TDM que está transportando. Ou seja o serviço de emulação não tem nenhuma informação a respeito do *framing* do quadro TDM.

[4]O transporte estruturado se dá quando o serviço de emulação não só tem conhecimento da estrutura TDM que está transportando, como também participa ativamente da construção do quadro TDM, podendo realizar alterações nas posições dos *slots*.

com seu transporte sendo realizado apenas por meios de transmissão TDM. Esse tipo de acesso predominou nas redes móveis de primeira geração e suas tecnologias de transmissão envolvidas eram:

- *Plesiochronous Digital Hierarchy* (PDH) - As redes de transmissão PDH foram desenvolvidas objetivando apenas serviços de voz. Utilizam a tecnologia de hierarquia digital para canais de transmissão onde ocorre multiplexação de canais sucessivamente, utilizando o processo TDM. Na hierarquia PDH, os canais são agrupados, formando os níveis hierárquicos, onde o primeiro nível dessa hierarquia corresponde a 32 canais TDM de 64 kbit/s, formando, assim, um canal com 2,048 Mbit/s, via intercalação sequencial de bytes. Esse primeiro canal PDH é comumente chamado de E1. A segunda ordem dessa hierarquia se dá com a combinação de quatro canais da primeira hierarquia, onde é obtido um canal chamado de E2, com taxa de 8 Mbit/s. Quatro canais E2 formam a terceira ordem com taxa 34 Mbit/s (E3). A quarta ordem é um conjunto de E3s com 140 Mbit/s (E4) e a quinta ordem é um conjunto de E5s com 565 Mbit/s (E5).

  Na tecnologia PDH, falhas ou imperfeições nos canais são corrigidas com o uso de bits de justificação. Contudo, a inclusão desses bits deixam as taxas dos canais ligeiramente diferentes e por isso deu-se o nome de quase síncrono (*Plesiochronous*).

- *Synchronous Digital Hierarchy* (SDH) - Nos meios de transmissão SDH, agrupamentos de canais são realizados de forma semelhante ao agrupamento nas redes PDH, porém as taxas usadas pelas redes SDH são respectivamente: 155 Mbps, 622 Mbps, 2,5 Gbps, 10 Gbps e 40Gbps. Porém, nas redes SDH, todos os relógios dos elementos da rede estão sincronizados a fim de manter as taxas dos canais idênticas em todos os elementos. Esse sincronismo é obtido através da escolha de um elemento como mestre, onde todos os demais recuperam a informação de sincronismo do mesmo.

Nesse tipo de rede, as antenas da rede móvel usam circuitos físicos até seu controlador ou até o *Mobile Switching System* (MSC). Especificamente, os circuitos TDM alugados e mesmo os circuitos próprios da operadora são caros de manter. Dessa forma, o SAToP fornece uma alternativa para manter circuitos TDM físicos, antenas e seus controladores, quando há uma conectividade IP/MPLS disponível, como mostrado na Figura 2.7.

Nota-se, na arquitetura baseada em IP/MPLS, que tanto as antenas como seus concentradores ainda possuem circuitos TDM E1, mas os circuitos terminam fisicamente em cada roteador local. O roteador transporta, então, os quadros TDM através de uma rede

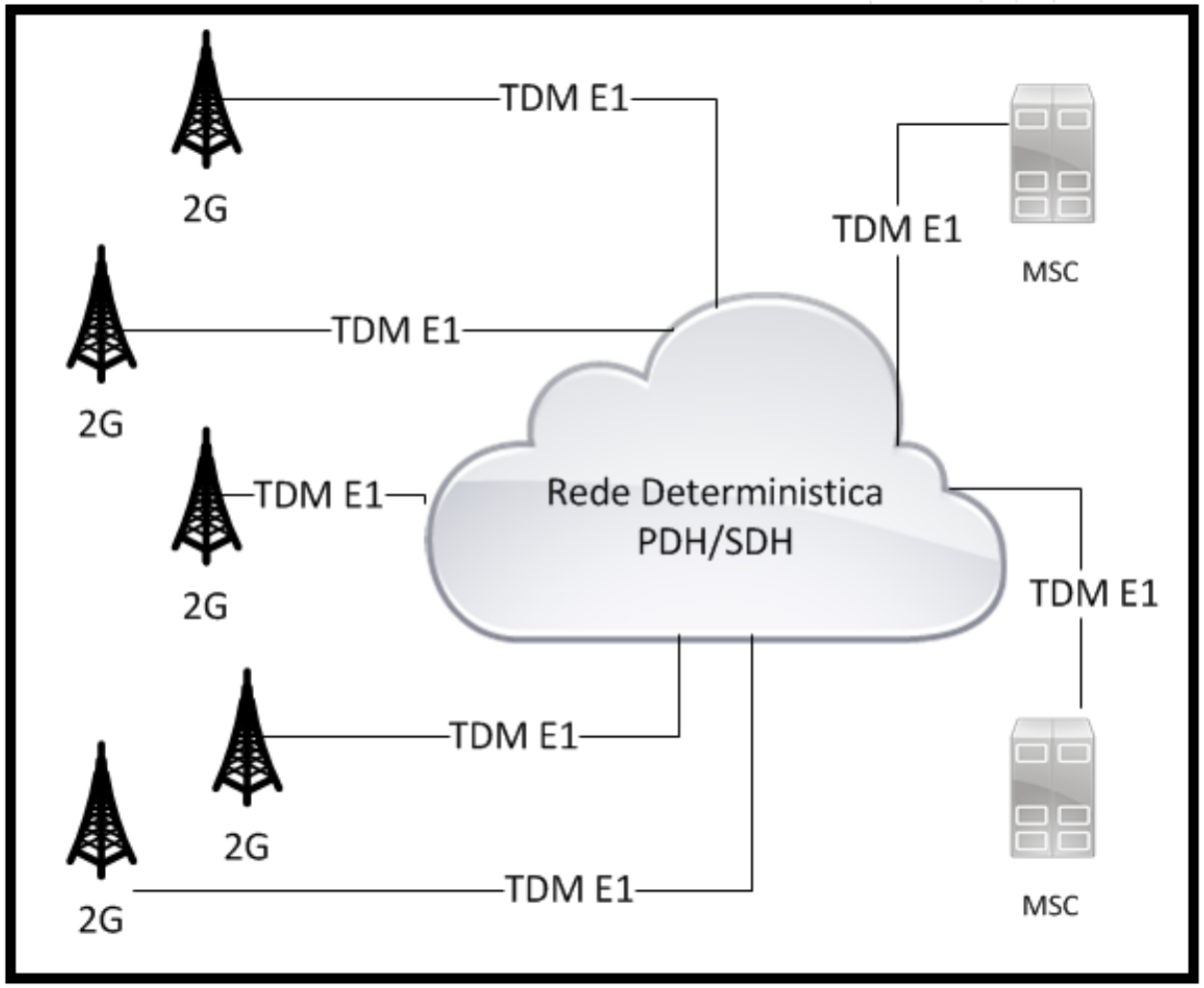



Figura 2.6: Topologia básica de interligação de antenas a seus concentradores via rede TDM.

MPLS por meio das tecnologias de emulação de circuitos chamada de *Circuit Emulation* (CEM), já mencionadas. Tanto no CEoP, quanto no SAToP, cada circuito E1 transportado pela rede IP/MPLS é chamado de *pseudowires* (PWs) e é transportado com o uso de informações da rede MPLS para o roteador remoto, de modo que as informações MPLS possam ser removidas e o circuito TDM recuperado no roteador de destino. Assim, é fornecido um meio de comunicação TDM emulado para comunicação dos elementos móveis, sem a necessidade de diferentes circuitos físicos ou de uma rede determinística separada para esse tráfego. A migração para essa solução comparada ao transporte clássico TDM faz ainda mais sentido quando uma rede IP/MPLS já está prontamente disponível (mesma rede que fornece suporte para rede 3G e 4G). Tal migração também leva em conta a inclusão ou migração de novos serviços que podem ser incorporados em conexões Ethernet/IP nativas.

A forma como os circuitos TDM são emulados via *pseudowires* pode ser resumida em

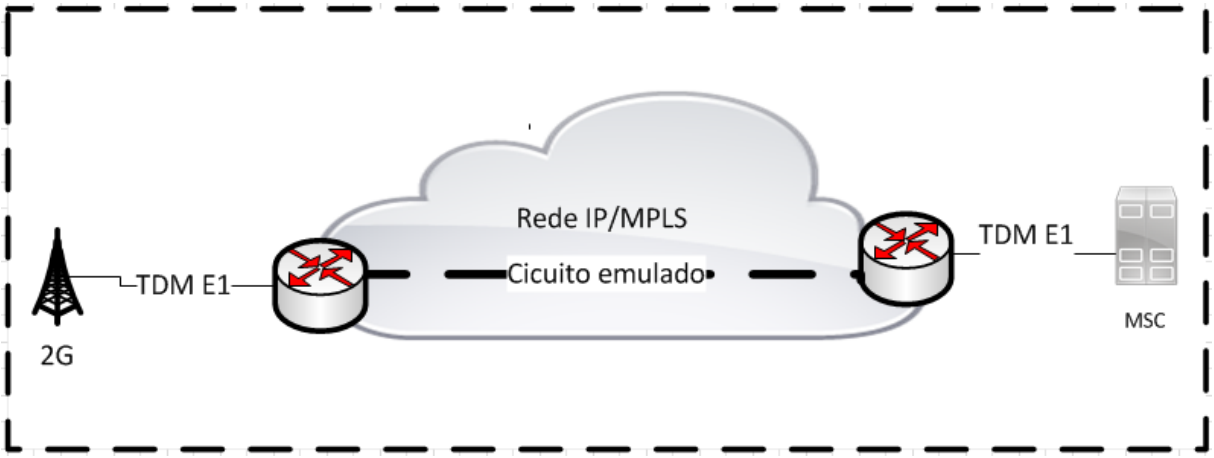



Figura 2.7: Topologia básica de interligação de antenas a seus concentradores via rede IP/MPLS.

cinco etapas, como mostrado na Figura 2.8:

1. Os quadros TDM são gerados pela antena e transmitidos para o roteador CEM, que possui um controlador TDM (E1/STM-1).

2. O roteador CEM recebe o quadro TDM, adiciona o encapsulamento do circuito e, em seguida, anexa o cabeçalho de MPLS, transmitindo o quadro através da rede IP/MPLS. No momento da criação do serviço de emulação, uma conexão *Label distribution protocol* (LDP)[5] é criada entre os roteadores CEM, a fim de determinar informações sobre status do serviço, tais como *Maximum Transmission Unit* (MTU), *label* MPLS usado para identificação do TDM, etc.

3. Os elementos da rede IP/MPLS comutam os quadros MPLS baseando-se apenas no *Label-Switched Path* (LSP)[6] existente entre os roteadores.

4. O roteador de destino do *pseudowire* recebe o quadro MPLS e o associa com o controlador de TDM apropriado baseado no *label* MPLS determinado pelo LDP existente entre os roteadores CEM. O quadro chega ao *buffer* do controlador CEM, que remove as informações da rede MPLS e do controlador CEM e envia para o controlador TDM.

[5]O LDP é o protocolo usado nas redes MPLS para distribuição de labels entre os roteadores, assim os elementos da rede sabem quais labels devem ser usados na construção do caminho MPLS entre os nós da rede.

[6]O LSP nada mais é do que o caminho de comutação realizado pelo tráfego quando o mesmo é encaminhado via *label* MPLS. Quando um tráfego IP entra em uma rede MPLS, ele é comutado via *label*. O pacote IP é ligeiramente alterado com a inclusão do cabeçalho MPLS, que contém pelo menos um *label MPLS*.

5. O controlador TDM recupera as informações de quadro TDM e envia para interface E1.

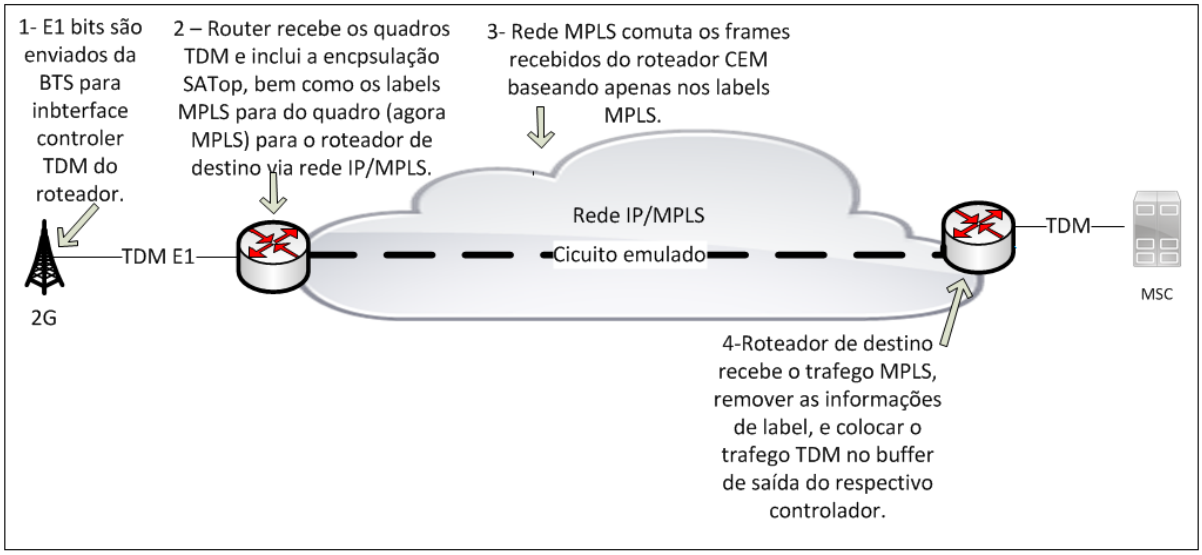



Figura 2.8: Topologia e funcionamento básico de circuitos determinísticos emulados via IP/MPLS.

O processo descrito é bidirecional e *buffers* são criados em cada roteador CEM a fim de que os quadros sejam transmitidos/recebidos nos controladores TDM dentro da velocidade de *clock* correta, de forma a emular um circuito físico TDM fim a fim. Quando um circuito é emulado com CEoP/SAToP, os quadros CEM são susceptíveis a atrasos variáveis, os quais são típicos em uma rede IP/MPLS. Dessa forma, os *buffers* são uma das formas como os serviços de *pseudowire* evitam as consequências do atraso variável, pois os quadros TDM são retirados do *buffer* de saída em taxa constante da ordem de milissegundos, para assegurar que os quadros estejam disponíveis para transmitir ao controlador TDM. O tamanho de *buffer* é um valor configurável e pode ser aumentado para acomodar o atraso da rede potencial. Nota-se que para o funcionamento correto do serviço de emulação do circuito TDM, é necessário que os geradores de *clock* de ambos os roteadores usados para criação da emulação do serviço estejam sincronizados, mesmo não tendo mais uma rede síncrona disponível entre esses elementos. Para que o sincronismo de relógio ocorra da forma correta entre os roteadores, é comum o uso do protocolo IEEE 1588v2 - *Precision Time Protocol* (PTP). O PTP é usado para distribuir a informação de pulso através de uma rede IP e não há nenhuma necessidade de conexão direta entre os geradores de *clock* e os roteadores que irão receber a mensagem e recuperar esse *clock*, que, nesse caso, é o roteador CEM. Somente a conectividade IP estável é necessária entre os dispositivos para distribuir a informação do pulso na carga útil dos pacotes IP. Dessa forma, obtém-se

uma rede IP/MPLS capaz de transportar todos os serviços móveis de todas as gerações entre as antenas e seus controladores. Além disso, por se tratar de uma rede puramente IP/MPLS, essa rede é capaz também de dar suporte a qualquer outro serviço que necessite de conectividade IP/MPLS e/ou Ethernet, bem como os circuitos TDM já mencionados.

Dada à necessidade de sincronismo e a geração de uma grande quantidade de pacotes de tamanho pequeno, os serviços de *pseudowire* implementam também a capacidade de fornecer informações precisas a respeito do estado do circuito emulado, bem como do desempenho do mesmo através da rede.

Essa dissertação tem como escopo o uso de tecnologias de aprendizado para uma melhor compreensão das relações entre as diversas variáveis de rede e seus impactos na qualidade do serviço de *pseudowire*, visto que nem sempre as falhas nos serviços estão relacionadas diretamente com falhas na rede ou com falhas que são facilmente identificadas pelas ferramentas de gestão da rede. São usadas para esse estudo as informações disponíveis pelos serviços de *pseudowire*, bem como informações importantes para correlação com as variáveis que possuem informações dos status dos elementos de rede.

# Capítulo 3

# Aprendizado de Máquina

A ideia de que computadores e máquinas pudessem aprender a realizar uma atividade sem serem previamente programados para tal atividade não é nova. Em 1997, Tom M. Mitchell publicou seu livro intitulado *"Machine Learning"*, onde forneceu uma definição para o aprendizado de máquinas:

"Diz-se que um programa de computador aprende pela experiência E, com respeito a algum tipo de tarefa T e desempenho P, se seu desempenho P nas tarefas em T, na forma medida por P, melhora com a experiência E."

O aprendizado de máquinas tem como objetivo a execução de uma ou mais tarefas por uma máquina ou programa, em qualquer circunstância que envolva a execução, sem que essa máquina ou computador tenha sido programado para realizar essa tarefa especificamente. Como a máquina e/ou o programa de computador não foi programado para realizar tal tarefa, entende-se que o mesmo aprendeu a realizar a atividade. O uso de aprendizado de máquinas tem crescido a cada dia e cresce o uso de aplicações com base em aprendizado de máquinas.

## 3.1 Tipos de tarefas

Normalmente, os sistemas ou algoritmos de aprendizado de máquina têm como entrada várias informações que são chamadas de *características* ou variáveis independentes e a tarefa que precisam aprender esta associada à modelagem ou relação entre essas variáveis de entrada e as informações de saída, que nesse caso são chamadas de *alvo* ou variável dependente. Em alguns casos específicos os algoritmos de aprendizado de maquina podem trabalhar tanto com as variáveis independentes quanto com as variáveis

dependentes durante o processo de aprendizado. A fim de que as variáveis de entrada produzam corretamente as variáveis de saída, os sistemas de aprendizado de máquinas executam as tarefas relacionadas com os algoritmos de forma que novos dados possam ser interpretados da forma correta, gerando as variáveis de saída esperadas. As tarefas executadas nos algoritmos de aprendizado de máquinas podem ser divididas em três categorias: classificação, clusterização e regressão.

### 3.1.1 Classificação

A tarefa de classificação tem como objetivo final um valor discreto. Ou seja, uma quantidade de *características* de entrada são passadas para o sistema de aprendizado e o mesmo informa como resultado final um valor dentro de um grupo já pré-definido. Por exemplo, o dono de uma floricultura possui diversas informações a respeito de três tipos de flores: A,B e C. Sempre que uma nova flor chega até sua loja, é realizada uma observação das mesmas informações na nova flor, a fim de constatar a qual grupo essa flor pertence. Dessa forma, um sistema de aprendizado de máquina com a tarefa de classificação pode ser usado para relacionar as diversas características de entrada com o grupo ao qual a flor pertence. Com a tarefa de classificação, os algoritmos de aprendizado são capazes de prever a qual tipo de flor pertence uma nova amostra com base no aprendizado das informações passadas. A Figura 3.1 demonstra de forma visual a classificação de dados em dois grupos que pode ser alcançada pelos algoritmos de aprendizado de máquinas na tarefa de classificação.

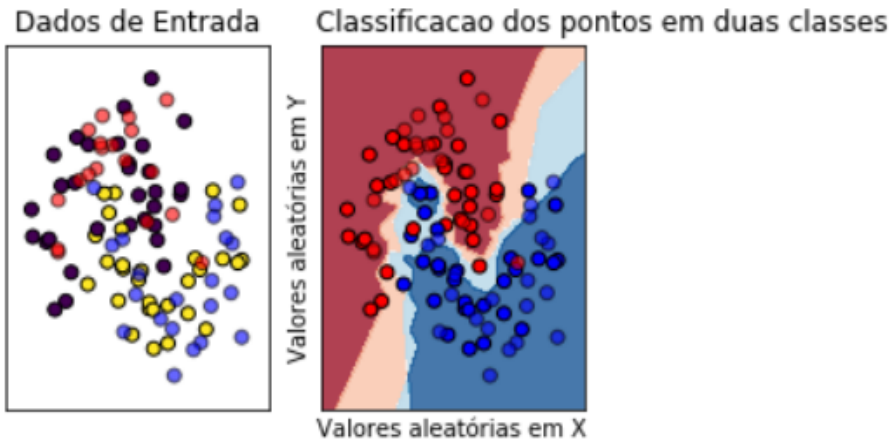



Figura 3.1: Tarefa de classificação realizada pelos algoritmos de aprendizado de máquina.

Casos comuns de uso da tarefa de classificação são:

- Verificar se um e-mail é span ou não;
- Verificar se uma transação bancaria é verdadeira ou falsa;
- Verificar se um tumor é maligno ou benigno.

### 3.1.2 Clusterização

A tarefa de clusterização está relacionada com o agrupamento de informações, de forma que o resultado final tem as amostras com informações semelhantes agrupadas, gerando conjuntos contendo apenas objetos com características similares. É importante perceber que para tal tarefa não teríamos uma resposta certa ou errada e sim os agrupamentos que melhor relacionam as informações de entrada para a formação dos respectivos grupos. É muito comum esses grupos serem chamados de *clusters*. Na Figura 3.2, é possível verificar a clusterização alcançada por algoritmos de aprendizado de máquinas em pontos aleatórios. Nesse exemplo, os pontos são divididos em três *clusteres*, onde os pontos vermelhos são os centros dos *clusters*.

Os sistemas de aprendizado normalmente usam tarefas de clusterização nas seguintes aplicações:

- Agrupamento de notícias de um mesmo segmento;
- Recomendação de filmes em função de tipos assistidos; e
- Agrupamento de cores parecidas no tratamento de imagens.

### 3.1.3 Regressão

Os algoritmos de aprendizado de máquina que realizam as tarefas de regressão têm como objetivo final chegar a um valor numérico que pode ser qualquer valor, consequentemente não discreto e não pertencente a um grupo de valores já conhecido. Esse valor numérico é previsto para um novo conjunto de informações de entrada baseando-se no aprendizado realizado previamente com as outras entradas. Embora tanto a regressão como a classificação tenham como resultado final um valor, existe uma diferença significativa entre essas tarefas, pois, na regressão, o valor de saída pode ser qualquer valor numérico. Por isso, é dito que a regressão é capaz de prever valores contínuos em sua saída ao passo que na classificação os valores de saída são discretos, ou seja, a saída só terá determinados valores já especificados anteriormente. Assim, todas as novas entradas

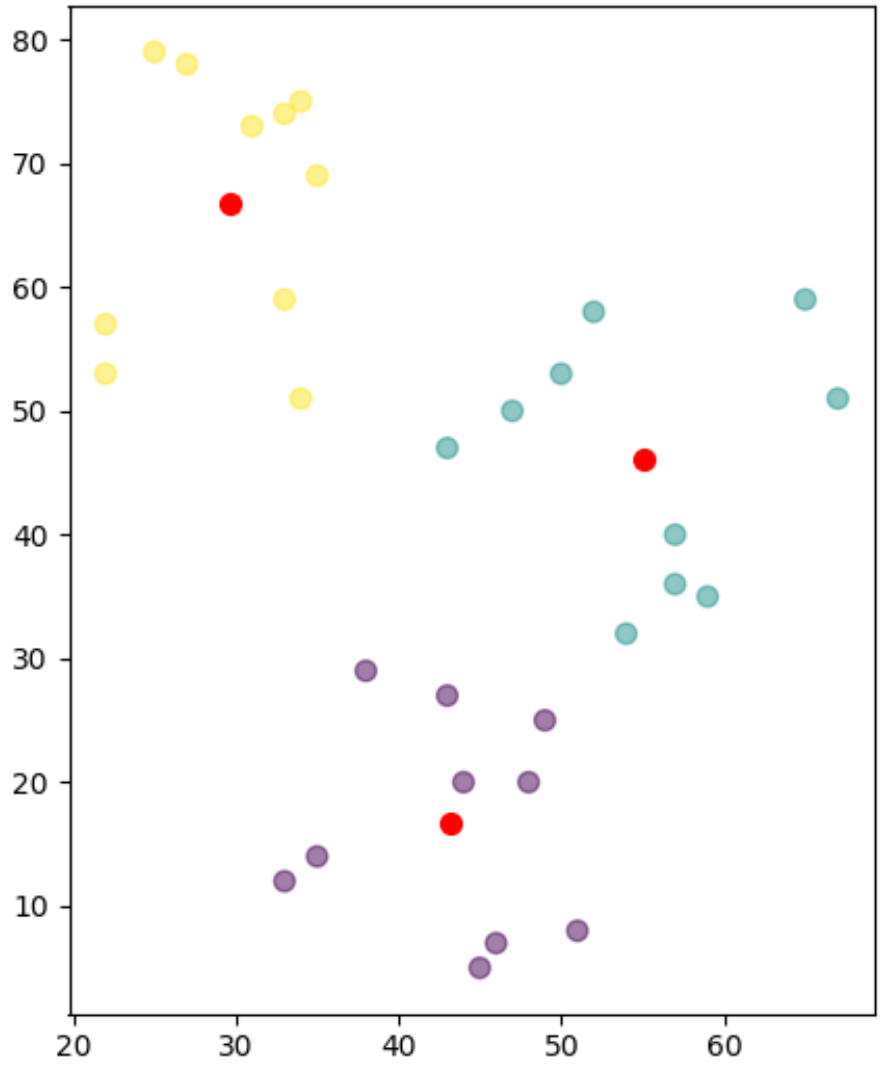



Figura 3.2: Tarefa de clusterização que pode ser realizada pelos algoritmos de aprendizado de máquinas.

terão como saída apenas um dos valores verificados na amostragem durante o processo de aprendizado, o que já não é uma regra para tarefas de regressão. As tarefas de regressão normalmente são usadas nas seguintes aplicações:

- Determinar a produção de um grão tendo como relação às informações de clima e solo;
- Determinar o preço de uma casa em função das informações de vendas anteriores; e
- Prever a avaliação de um filme dado o perfil de um usuário.

A Figura 3.3 exemplifica a regressão realizada pelos algoritmos de aprendizado de

máquinas. Na figura, é possível visualizar o uso de uma função de reta que permite a regressão a um valor para as amostras passadas aos algoritmos.

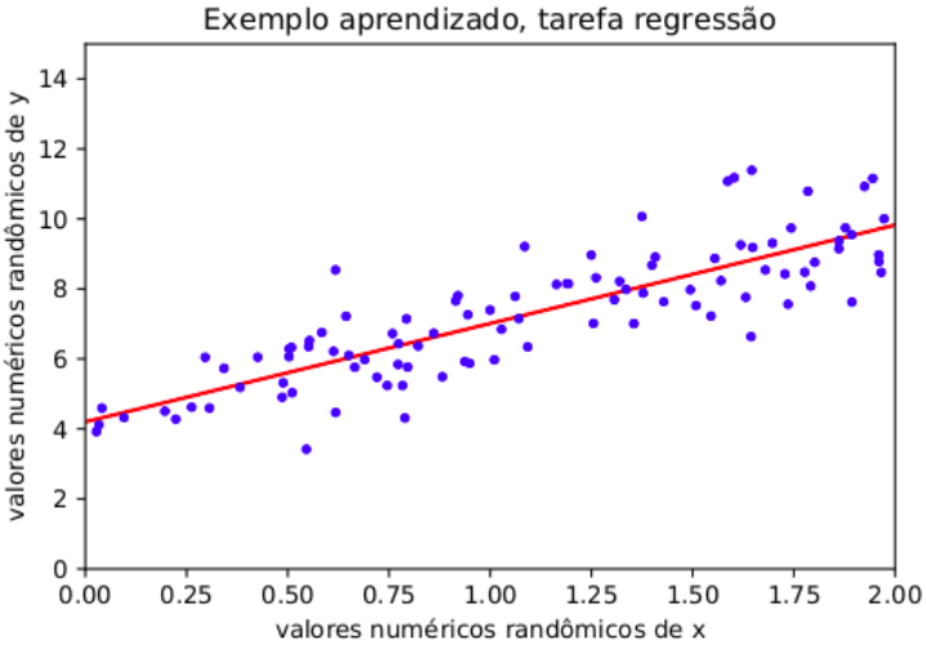



Figura 3.3: Tarefa de Regressão.

## 3.2 Tipos de aprendizado

Os algoritmos de aprendizado de máquina normalmente são divididos em três grandes grupos. Esses grupos possuem diversas características em comum e estão diretamente relacionadas com a forma pela qual o sistema de aprendizado compreende se o aprendizado foi realizado da forma correta. A informação que relaciona se o aprendizado foi realizado de forma correta e aceitável também pode ser usada para verificar a precisão do modelo de aprendizado obtido. As categorias de aprendizado são descritas a seguir.

### 3.2.1 Aprendizado Supervisionado

No aprendizado supervisionado, são apresentadas ao sistema de aprendizado várias amostras de características de entrada (variáveis independentes) e suas respectivas saídas (alvo/variável dependente). O objetivo é aprender a regra e/ou relação que melhor mapeia as entradas (características) para as suas respectivas saídas (alvos). Normalmente, esse aprendizado é alcançado através de uma primeira etapa, chamada treinamento, onde uma parte dos dados é apresentada ao sistema de aprendizado para que o mesmo possa,

pela modelagem matemática dessas informações, obter a melhor função que relaciona as entradas com as saídas informadas. A parte dos dados não usada no processo de treinamento é utilizada para se verificar o grau de precisão do aprendizado. Assim, com um bom grau de certeza, o algoritmo de aprendizado pode ser exposto a outro conjunto de entradas e prever com certo grau de certeza o resultado (saída) para esse novo conjunto de entradas.

Para ilustrar a forma de funcionamento do aprendizado supervisionado, imagine que um corretor de imóveis está com uma grande quantidade de trabalho e deseja criar um sistema que possa auxilia-lo a prever o valor de venda de uma casa, usando como base as informações específicas de diversos imóveis. Então, o corretor começa a cadastrar informações a cada vez que alguém vende uma casa, como por exemplo, número de quartos, tamanho em metros quadrados, vizinhança, além do preço final de venda, conforme a Tabela 3.1.

Tabela 3.1: Exemplo de dados para aplicação de aprendizado supervisionado.

| **Numero de quartos** | **Metros quadrados** | **Vizinhança** | **Preço de venda** |
|---|---|---|---|
| 3 | 2000 | A | 300.000 |
| 1 | 450 | A | 100.000 |
| 2 | 850 | B | 250.000 |
| 2 | 550 | B | 200.000 |
| 1 | 500 | C | 120.000 |
| 4 | 2000 | A | 350.000 |

Nesse conjunto de informações, são considerados características (dados de entrada) o número de quartos, metros quadrados e vizinhança. Esses dados serão usados para modelagem/treinamento do algoritmo para previsão da saída ou alvo, que nesse caso é o campo preço de venda. E novas informações de outras casas podem ser passadas para o algoritmo a fim de se obter o valor de venda. Essa forma de aprendizado é chamada de aprendizado supervisionado, porque durante o processo de aprendizado e/ou treinamento o algoritmo recebeu não śó as informações de entrada (características), como também teve conhecimento do valor de saída relacionado com cada conjunto de entrada. Assim, é possível realizar o aprendizado necessário para construção da lógica que relacionada cada conjunto de entrada e sua respectiva saída.

### 3.2.2 Aprendizado não supervisionado

Ao contrario do aprendizado supervisionado, no aprendizado não supervisionado, nenhuma saída é dada para o sistema de aprendizado. Nesse caso, o sistema de aprendizado tenta encontrar uma estrutura, similaridade ou correlação nas entradas fornecidas. Ou seja, no aprendizado não supervisionado, não existe saídas certas ou erradas. Ao invés disso, o objetivo é encontrar um padrão que relaciona as diversas amostras de entrada. Dessa forma, normalmente, o algoritmo de aprendizado de máquina não supervisionado tem como objetivo descobrir novos padrões nos dados ou realizar o agrupamento de entradas, sendo muito utilizados para tarefas de clusterização.

### 3.2.3 Aprendizado por reforço

No aprendizado por reforço, o sistema de aprendizado interage com um ambiente completamente dinâmico, onde a função do programa é interagir com o ambiente e alcançar determinado objetivo. Contudo, a forma de realizar esse objetivo pode sofrer modificações em função da dinâmica do ambiente. Para verificar a precisão de como a tarefa está sendo realizada no aprendizado por reforço, uma realimentação é dada ao sistema de aprendizado em forma de punição ou premiação, a fim de que o mesmo possa realizar a tarefa da melhor forma possível mesmo em um ambiente totalmente dinâmico.

Exemplos onde à tarefa de aprendizado por reforço incluem dirigir um veículo ou jogar um determinado jogo contra um oponente.

### 3.2.4 Aprendizado Semi-Supervisionado

O aprendizado semi-supervisionado está entre o aprendizado supervisionado e o não supervisionado. No sistema de aprendizado semi-supervisionado, os algoritmos normalmente não recebem todos os dados completos. Ocorre que, nesse caso, um conjunto de dados de treinamento com algumas (muitas vezes várias) das saídas desejadas não existe. Uma aplicação típica de aprendizado semi-supervisionado é a tradução, onde um conjunto inteiro das instâncias do problema é conhecido no momento do aprendizado, mas com parte dos objetivos ausente.

## 3.3 Overffiting e Underffiting

Durante o processo de aprendizado dos algoritmos de aprendizado de maquina, os algoritmos realizam alguma das tarefas já mencionadas (clusterização, regressão ou classificação) normalmente com parte dos das informações afim de a fim de obterem o aprendizado necessário sobre os dados informados de forma que realizando a mesma tarefa em dados desconhecidos o mesmo possa obter a saída desejada com boa margem de precisão, levando em consideração o aprendizado obtido. Para que isso seja possível, o algoritmo precisa obter o aprendizado necessário de forma generalista, ou seja, para que o mesmo possa aplicar esse aprendizado em novas informações. Um dos principais problemas durante o processo de aprendizado esta associado à ação de memorização, onde o algoritmo não obtém pode ocorrer onde o algoritmo não consegue obter as informações necessárias para realização da tarefa com boa precisão. Quando isso ocorre diz-se que o algoritmo apresentou underffiting. O ideal é que os algoritmos consigam todo aprendizado necessário para realizar a mesma tarefa em novas informações obtendo com bom grau de precisão o resultado da tarefa, quando isso ocorre diz-se que o algoritmo alcançou um bom aprendizado e grau de generalização. A figura 3.4 demonstra de forma gráfica um processo de aprendizado sofrendo overffiting, underffiting e uma boa generalização em um conjunto de informações nos eixos X,Y, onde a linha em preto representa a forma como o algoritmo de aprendizado de maquina moldou as informações no processo de aprendizado.

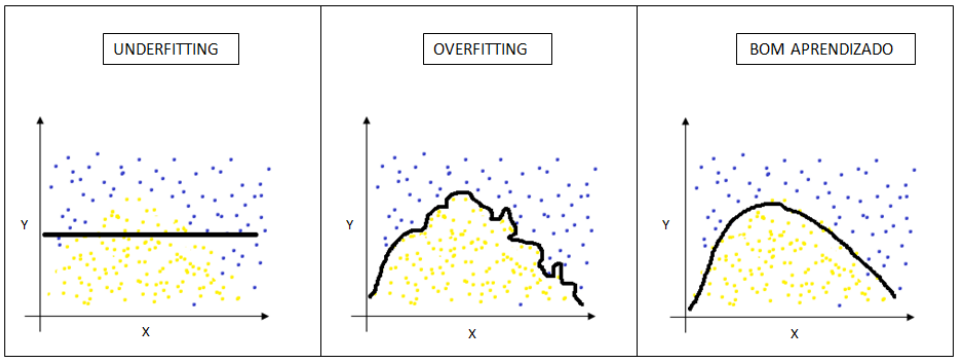



Figura 3.4: Overffiting vs. Underffiting.

## 3.4 Algoritmos de aprendizado de máquina usados nesse estudo

Essa dissertação utiliza a tarefa de classificação do estado dos serviços da rede de forma a constatar quais algoritmos alcançariam uma melhor precisão na classificação,

pois a avaliação da precisão dos algoritmos na tarefa de classificação implica em uma excelente correlação entre as variáveis de rede e os status do serviço no momento da realização da coleta. Como pode ser constatado no 6 os algoritmos que apresentaram excelentes níveis de precisão no processo de classificação, são algoritmos que implementam alguma função *Ensemble*. Ensemble são técnicas implementadas por alguns algoritmos de aprendizado de maquina que combinam vários modelos de aprendizado conjuntos a fim de reduzir a variância e os erros incrementando assim a precisão. A figura 3.5 demonstra de forma visual o que significa a variância e a precisão para os algoritmos de aprendizado de maquinas.

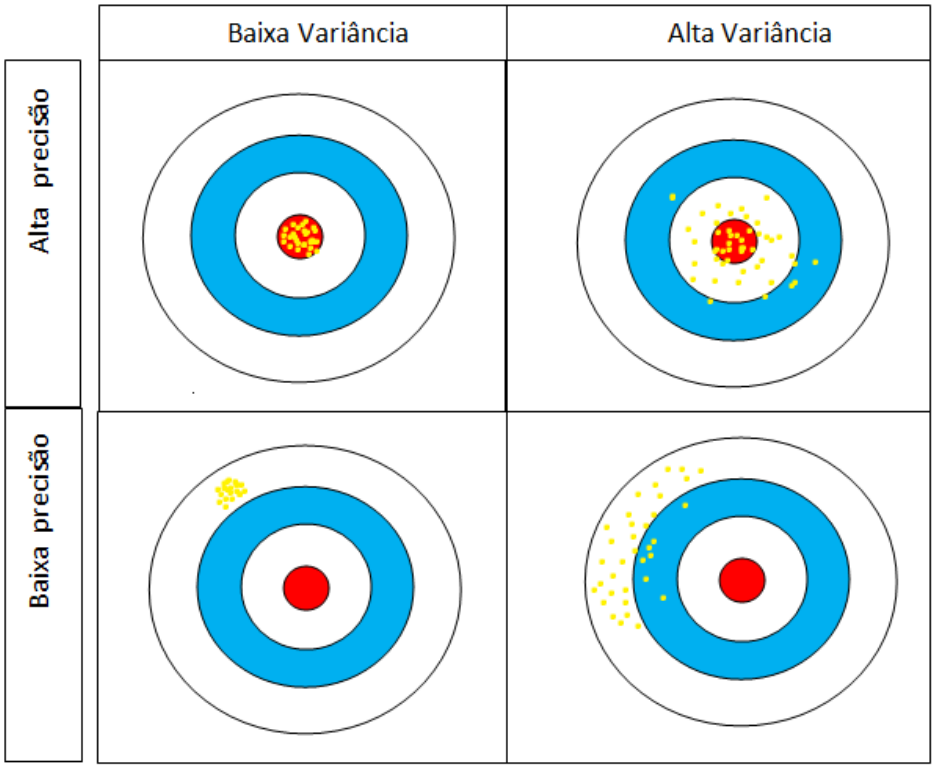



Figura 3.5: Variância Vs. Precisão

Como mostrado na figura 3.6 os algoritmos que implementa as técnicas de Ensembre utilizados nesse estudo podem ser divididos em dois grupos:

- Boosting: Os algoritmos que implementam a técnica de Ensembre via boosting, realizam as melhorias necessárias na precisão com o uso sequencial de algoritmos de aprendizado. O uso das técnicas de boosting esta associada a melhor exploração da dependência que ocorre entre cada algoritmo de aprendizado usado em sequencia, pois o desempenho pode ser melhorado em virtude de pesos atribuídos a cada erro ocorrido em cada algoritmo presente na forma sequencial.

- Bagging: Os algoritmos utilizam aprendizado em paralelo são chamados de Bagging. Ao contrario das técnicas de boosting o uso paralelo de algoritmos de aprendizado explora a independência dos algoritmos, pois o uso de aprendizado em paralelo permite melhorias da precisão com o uso da media dos algoritmos.

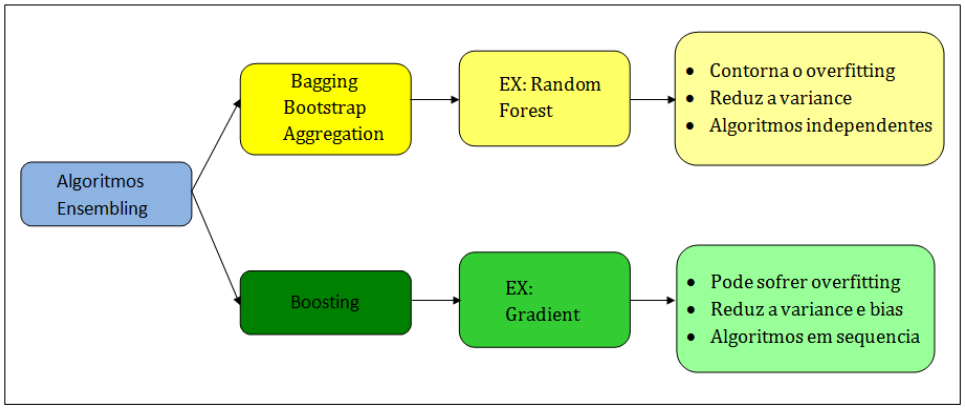



Figura 3.6: Algorítimos Ensemble

A figura 3.7 demonstra a forma de interligação dos métodos boosting e bagging usados para implementação das funções Ensemble.

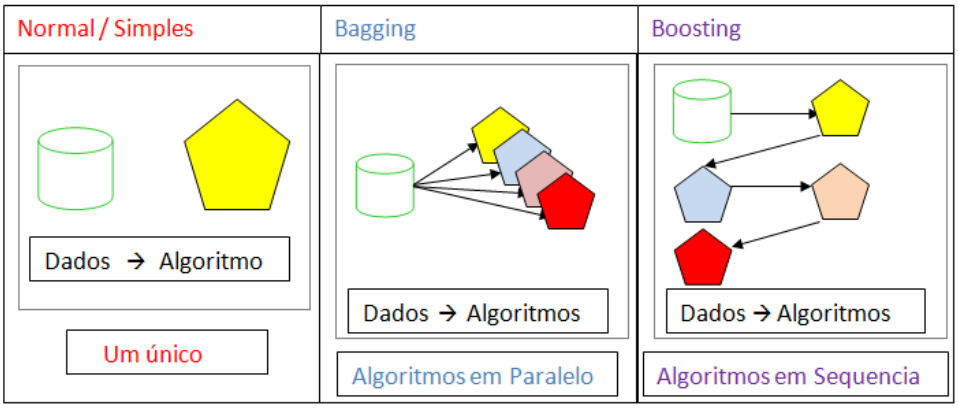



Figura 3.7: Algoritmos Ensemble

Assim, os principais algoritmos relacionados à classificação usados nessa dissertação são apresentados a seguir.

### 3.4.1 Árvores de Decisão

Árvores de decisão são métodos de aprendizado de máquinas supervisionado muito utilizados em tarefas de classificação e regressão. Árvores são estruturas de dados formadas por um conjunto de elementos que armazenam informações chamadas nós. Em cada

nó da árvore, uma decisão é tomada e um caminho é seguido. Toda árvore possui um nó chamado raiz, que possui o maior nível hierárquico (o ponto de partida) e ligações para outros elementos, denominados filhos. Esses filhos podem possuir seus próprios filhos, que, por sua vez, também possuem outros filhos e assim por diante. O nó que não possui filho é conhecido como nó folha ou terminal. Dessa forma, uma árvore de decisão nada mais é que uma árvore que armazena regras em seus nós, e os nós folhas representam a decisão a ser tomada a fim de se obter um determinado resultado. Logo, a partir do nó raiz da árvore, decisões são tomadas até que se alcance um nó folha.

### 3.4.2 ADABoost

O termo *AdaBoost* é usado como acrônimo para *Adaptative Boosting* e está relacionado com as técnicas de melhorias de precisão e desempenho chamadas de *Boosting*. As técnicas usadas pelos algoritmos de *AdaBoost* podem, na verdade, auxiliar em melhores precisões para qualquer outro algoritmo de aprendizado de máquina. Normalmente, o algoritmo de *AdaBoost*, quando iniciado sem nenhuma modificação dos seus parâmetros, é utilizado com algoritmos de árvore de decisão, gerando com as técnicas de *boosting* várias árvores de decisão. Os dados de entrada juntamente com suas respectivas saídas são aleatoriamente divididos e cada parte dos dados é passada para uma árvore. Assim, o algoritmo *AdaBoost* avalia a precisão de cada árvore e atribui um peso a mesma em função da precisão alcançada, permitindo ter, no final, um algoritmo resultante da combinação das árvores geradas, respeitando o peso atribuído às mesmas. Com isso o algoritmo de aprendizado de máquinas *AdaBoost* consegue ótimos valores de precisão e desempenho, evitando o *overfiting*. Na Figura 3.8, é possível visualizar o funcionamento básico do algoritmo *AdaBoost*.

### 3.4.3 *Gradient Boosting*

O algoritmo *Gradient Boosting*, da mesma forma que o *AdaBoost*, funciona intercalando algoritmos sequencialmente. Contudo, em vez de ajustar os pesos de cada algoritmo como faz o *AdaBoost*, o *Gradient Boosting* tenta ajustar o próximo algoritmo para minimizar os erros de precisão do algoritmo anterior, obtendo, dessa forma, melhores níveis de precisão.

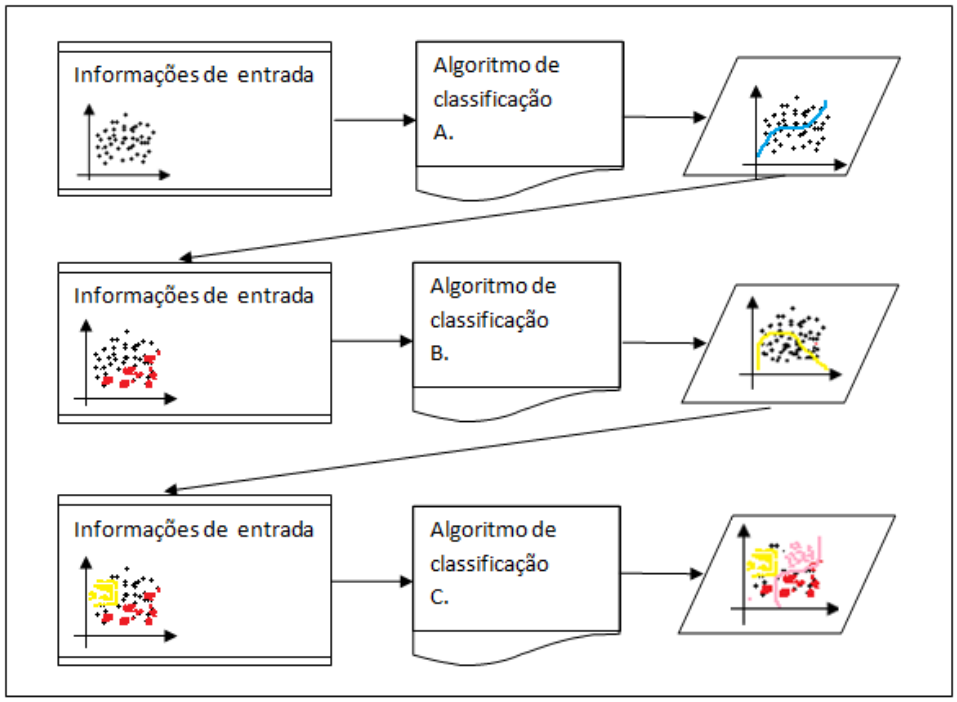



Figura 3.8: Algoritmo *AdaBoost*.

### 3.4.4 *Random Forest Classifier*

O algoritmo de aprendizado de máquina *Random Forest Classifier*, descrito na Figura 3.9, pertence à família de árvores de decisão. Árvores de decisão são muito comuns em diversas áreas da computação, especialmente em aprendizado de máquinas. As árvores de decisão possuem duas características que as fazem ser muito utilizadas: primeiramente são precisas no não uso de características irrelevantes e, em segundo, são facilmente inspecionáveis, se comparadas com outros algoritmos. Porém, os algoritmos de árvore de decisão dificilmente são precisos em seus resultados. Quando os algoritmos de árvore de decisão trabalham com árvores muito profundas (grande quantidade de nós), os mesmos tendem a sofrer *overfiting*. De forma análoga ao *ADAboost*, os algoritmos de *Random Forest* geram e treinam várias árvores de decisão a fim de obter uma melhor precisão. Porém, uma particularidade dos algoritmos de *Random Forest* está no fato dos mesmos usarem um método de aprendizado modificado, onde, em cada nó das árvores, um conjunto de características é testado de forma randômica. Assim, se um conjunto pequeno de características tiver uma maior relação com o alvo, essas características estarão selecionadas para o modelo final. O objetivo de treinar diversas árvores com partes diferentes e randômicas dos dados é reduzir a variância do modelo e alcançar uma boa precisão sem sofrer o *overfiting*. O custo disso é a perda de parte da facilidade de interpretação do modelo, visto que agora existem diversas árvores simultâneas.

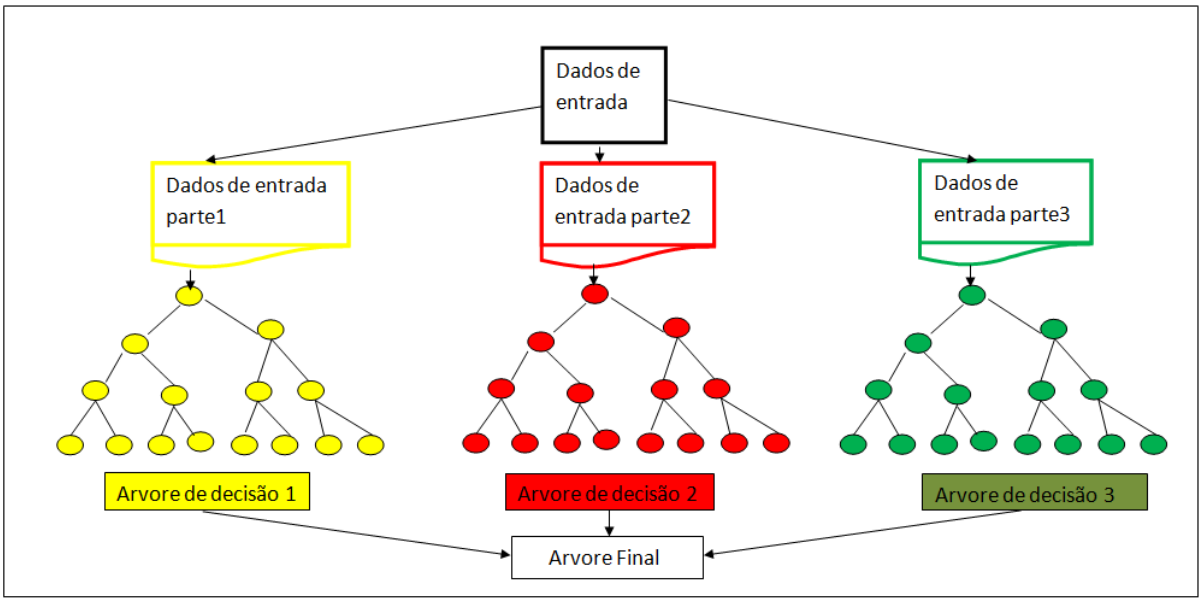



Figura 3.9: Algoritmo *Random Forest*.

### 3.4.5 ExtraTreesClassifier

As técnicas que dão origens aos algoritmos *ExtraTreesClassifier* são semelhantes às usadas nos algoritmos *RandomForest*, porém, no *ExtraTreesClassifier*, é realizada uma randomização no momento da escolha de características para aprendizado de cada nó.

# Capítulo 4

# Trabalhos Relacionados

Neste capítulo, são apresentados os principais trabalhos acadêmicos relacionados ao uso de algoritmos de aprendizado de máquina para melhores resultados na gerencia de redes. Tais trabalhos foram fundamentais para orientar o trabalho de análise do problema apresentado e proposta de solução para gerência de aplicações em tempo real.

Os algoritmos de aprendizado de máquina podem e devem compor soluções inovadoras nas plataformas de gerência de redes e aplicações, dado a grande quantidade de informações e tendências que esses algoritmos podem fornecer em tempo real. Em função disso, os algoritmos de aprendizado de máquina estão sendo vastamente estudados e utilizados. De fato, diversos artigos têm sido publicados apresentando meios de utilização dos algoritmos de aprendizado de máquinas na verificação de anomalias em redes IP [54], [51], [26], [12], [20].

Zubair et al [52] demonstram o uso de um grande *framework* para avaliação de dados de Netflow em tempo real, a fim de que sejam detectados anomalias no padrão de tráfego da rede. Embora o uso dessa abordagem seja eficaz para verificação de anomalias no tráfego, o uso das coletas NetFlow e não das variáveis da rede pode gerar imprecisões nos resultados.

Alguns trabalhos propõem sistemas baseados em aprendizado de máquina capazes de administrar e operar a rede sem qualquer intervenção humana [7, 49, 36, 16, 23]. Sabe-se que os ambientes de redes atuais são complexos, de forma que a gerência da rede demanda grande quantidade de conhecimento. Traduzir esse conhecimento em linhas de comandos para as configurações dos elementos de redes atuais pode levar a uma quantidade razoável de erros humanos no que tange a modificações das configurações dos elementos de rede a fim de acomodar novos serviços e configurações necessárias. Assim, sistemas de aprendi-

zado de máquina podem ser muito eficazes na padronização e verificação do *deployment* de serviços de rede, evitando assim boa parte dos erros. Tais sistemas também seriam capazes de predizer e auxiliar de forma direta as tarefas de diagnóstico de falhas nas redes [7, 2, 31, 29, 24]. Contudo, hoje, essa possibilidade esbarra na dificuldade de se obter informações padronizadas e de fácil acesso dos elementos de rede. Esse cenário pode receber grandes contribuições com o avanço da adoção de redes definidas por software [11, 39], [50] e do uso cada vez maior das tecnologias de *Network Functions Virtualization* (NFV) [53, 34, 38].

Além do uso dos algoritmos de aprendizado de máquina estar largamente em estudo e avaliação para detecção de anomalias no ambiente de redes, alguns artigos já citam testes com o uso desses algoritmos a fim de determinar com boa margem de precisão a previsão e probabilidade de falha em determinados elementos e/ou componentes da rede [46, 28, 42]. A vantagem dessa abordagem está no relacionamento dos fatores do ambiente com o uso da rede. Nesses casos, fatores ambientais e de uso podem ser levados em consideração durante o treinamento do algoritmo de aprendizado de máquina, de forma que se pode prever falha para partes especificas da rede.

Diversos trabalhos usando algoritmos de aprendizado de máquinas em aplicações conseguem classificar de forma automática aplicações na rede em função das informações de fluxos e não somente das portas de origem e destino [47, 48, 22]. Tais estudos demonstraram que algoritmos de aprendizado de máquinas tais como *Feature Selection* tiveram grande exatidão quando usados para a identificação automática de aplicações e padrões de fluxos. Em alguns casos, foram obtidas classificações automáticas com precisão de 90%, sem *overfitting* de dados. Tais algoritmos possuem grandes probabilidades de relacionar variáveis de rede com padrões de falhas, como é o objetivo de nosso trabalho.

Com a evolução do *hardware* de equipamentos de rede, o crescente aumento da quantidade de elementos da rede e o aumento no volume de tráfego de dados, os novos elementos de rede possuem uma boa quantidade de recursos computacionais para realizar a tarefa de calcular o melhor caminho e realizar a comutação dos pacotes. Usualmente, os equipamentos de rede apresentam *hardware* especializado apenas para comutação dos pacotes e *hardware* especializados para gerenciamento e controle dos recursos e tarefas do dispositivo. De fato, muitos equipamentos já possuem capacidade para realizar tarefas complexas ao ponto de poderem receber um conjunto de propriedades e um conjunto de parâmetros de desempenho necessários para uma determinada aplicação e realizar todas as tarefas necessárias a fim de determinar se a rede esta funcionando como esperado para alcance

das métricas necessárias [44]. Como o conjunto de propriedades e métricas não são a priori conhecidos pelos elementos de rede, o uso dos recursos dos mesmos para realização dessas aferições só podem ser alcançadas com o uso de algoritmos de aprendizado de máquinas.

O uso de funções baseadas em software *Network Functions Virtualization*(NFV) já é realidade em algumas áreas da tecnologia de redes e permite grandes incrementos de recursos de hardware para realização de diversas funções, pois recursos de hardware baseados em *Graphics Processing Unit*(GPUs) vêm sendo incluídos em novos elementos de rede, criando uma nova família de roteadores chamados de *Software Defined Routers* (SDR), onde funções virtualizadas podem sem implementadas. Os SDR poderão compor grandes sistemas de computação, visto que esses novos roteadores terão grandes aumentos na capacidade de processamento de pacotes [14]. Embora algoritmos de aprendizado de máquinas estejam sendo estudados e aprimorados constantemente, o uso dos mesmos nas áreas relacionadas com as redes de comutação de pacotes ainda são relativamente pequenos. Contudo, o uso de GPUs com novos recursos permitirão a composição dos novos sistemas de rede, com grandes avanços em áreas como classificação de pacotes ou até mesmo roteamento.

Além das diversas dificuldades operacionais trazidas pelo aumento da necessidade de conectividade que acarreta uma maior quantidade de elementos, links e novas topologias. Implementações de novas redes estão cada vez mais complexas, pois precisam atender todos os requisitos de rede demandados pela crescente necessidade de conectividade móvel. Logo, o uso de algoritmos de aprendizado de máquinas também pode e deve auxiliar na escolha de topologias e no projeto da rede. Esse auxilio é fornecido através dos diversos submódulos de aprendizado que permitem a escolha do melhor projeto e a verificação das métricas de rede fornecidas pelo mesmo. Para obtenção das métricas de rede é comum o uso de simulação dos projetos selecionados. Tais métricas servem também para atualizar o modelos de aprendizado do projeto escolhido pelos algoritmos de aprendizado de máquina [15].

O estudo e os avanços para aplicações de ferramentas baseadas em algoritmos de aprendizado de máquinas nas redes de telecomunicações estão ganhando cada vez mais espaço, principalmente em áreas de classificação de tráfego para definição e aplicação de QoS [37, 30, 18]. Atualmente, os modelos de classificação baseiam-se principalmente nas informações de porta disponíveis nos pacotes IP. Porém, o uso cada vez maior de portas não conhecidas, bem como o uso de criptografia nas diversas aplicações, tende a limitar os atuais modelos de classificação de tráfego. Ferramentas que possam classificar pacotes

e fluxos em tempo real são necessárias não só para melhorias no projeto da rede, como também para sua operação diária e para definição das políticas de qualidade de serviço. O uso de algoritmos de aprendizado de máquinas na criação de um sistema que possa fazer a classificação em tempo real do tráfego da rede tem sido estudado e diversas propostas tem sido avaliadas [27, 13, 45]. Tais propostas deverão utilizar recursos dos novos *SDRs* a fim de obter recursos de processamento necessários para auxiliar em uma melhora no desempenho das redes.

Para disponibilizar uma melhor cobertura, as redes de telecomunicações móveis atuais (4G) possuem uma grande diversidade de antenas de diferentes capacidades e alcances de cobertura. A disponibilidade dessas antenas está atrelada a diversos parâmetros como: a área de cobertura provida pela rede, faixas de frequências disponíveis, área de cobertura e quantidade esperada de usuários para cada antena, bem como parâmetros de mobilidade. A grande quantidade de antenas geram pequenas áreas sobrepostas de cobertura que auxiliam na mobilidade dos usuários. Essas áreas, por outro lado, trazem consigo uma grande complexidade para o *deployment*, para projeto de arquitetura e para o gerenciamento e uso das redes móveis. A complexidade tende a aumentar ainda mais com a chegada das redes 5G, em função da grande densidade de antenas necessária e da diversidade de terminais móveis que deverão usar essa rede.

Nesse contexto, as redes 4G já fornecem uma grande quantidade de informações sobre medições, controle e gerenciamento. Essa quantidade de informações tende a ser aumentada nas redes 5G. A disponibilidade de grande quantidade de informações irá proporcionar um bom ambiente para o uso de algoritmos de aprendizado de máquinas, visando melhorar não só uso da rede, como também a experiência do usuário. De fato, o controle de mobilidade de usuários em um ambiente denso de cobertura necessita de sistemas automáticos, inteligentes e responsivos para as condições da rede [4]. Embora a disponibilidade abundante de informações não seja um problema nas redes móveis de $4^a$ geração, a relação entre usuários das redes móveis e as redes de transporte, bem como informações granulares das próprias redes de transporte podem não estar disponíveis de forma fácil para sistemas que usam algoritmos de aprendizado de máquinas. Em função da grande quantidade de elementos e da diversidade dos mesmos nas redes de transporte atuais, a coleta e a correta interpretação dessas informações dificulta o uso correto das mesmas em algoritmos de aprendizado de máquinas. Além dessa grande quantidade e diversidade de elementos, a implementação de novos elementos e características acarreta em novas necessidades de desenvolvimento para interpretação das informações. Novas tecnologias, como as redes definidas por software, devem trazer novas formas de acesso

e validação das informações a serem usadas nos sistemas de aprendizado de máquinas. Se o acesso às informações puder ser simples, direto e documentado, isso não só facilita a implementação e validação do uso dos algoritmos de aprendizado de máquinas, como também auxilia no desenvolvimento e aprimoramento de novos padrões para extração dessas informações. Em redes *Software-Defined Networking*(SDN), já existem extensões para coleta de informações e aplicação de algoritmos de aprendizado de máquina [5].

A quantidade de informações usadas para aplicação de um classificador de tráfego IP pode ser exageradamente grande, se for levada em consideração a necessidade de realizar a tarefa de classificação de alguns milhares de fluxos. É percebido que, embora usem diferentes métodos para disponibilizar um modelo classificatório, os diferentes algoritmos de classificação possuem precisões semelhantes, embora não apresentem os mesmos tempos de processamento. É sabido também que técnicas para redução das informações passadas para os algoritmos de aprendizado de máquinas podem melhorar substancialmente o desempenho e a precisão dos algoritmos, uma vez que muitas informações podem ser redundâncias ou não tem relação pertinente aos eventos sendo analisados nos algoritmos de aprendizado de máquina. Dessa forma, técnicas de redução de características se mostraram de grande valia na melhora do desempenho de vários algoritmos de aprendizado de máquina sem prejudicar a precisão dos mesmos [47].

A possibilidade de prevenção de uma falha antes mesmo de sua ocorrência é vital para a as redes atuais, dado a criticidade e custo das redes de comunicações e seus empregos nos mais diversos contextos [17]. Os algoritmos de aprendizado de máquinas possuem papel fundamental e podem auxiliar na detecção de anomalias em variáveis de rede, mesmo que essas variáveis possuam padrões irregulares. Para tal, os algoritmos precisam possuir boas quantidades de amostras das variáveis coletadas. A correta correlação entre anomalias de rede e sua causa raiz esta diretamente relacionada com a coleta adequada das informações necessárias. Como as redes atuais são compostas por um grande numero de sistemas complexos interligados entre si, as próprias redes se tornam sistemas complexos, onde diversas informações de diversos sistemas precisam ser coletadas e analisadas a fim de se conhecer o desempenho da rede.

As informações, normalmente, podem ser obtidas através de *probes* em execução na rede, informações a respeito do status dos próprios elementos ou ate mesmo informações estatísticas dos fluxos que passam pela rede. A obtenção dos dados sobre os fluxos ativos é a tarefa mais complexa em função das características dos elementos de rede [43] e da necessidade de realização de amostragem dos fluxos, o que evita a exaustão dos recursos

nos elementos de rede.

Hoje, o uso de algoritmos de aprendizado de máquinas já alcançou com excelência partes importante das redes móveis, de forma que novas redes totalmente automatizadas já são realizadas em algumas áreas, como são as redes *Self-organized network*(SON) para áreas de acesso móvel das redes 4G. Tais níveis de excelência são possíveis em função das diversas informações já padronizadas e disponíveis nas redes 4G [25]. Por outro lado, esse fato não é verdade para redes IP, visto que diversas informações precisam ser coletadas e tratadas de formas diferentes até que possam ser efetivamente usadas. Embora as redes IP/MPLS não tenham todas as informações disponíveis, já é possível ter algumas funções importantes para as redes sendo automatizadas e realizadas com melhor eficácia por algoritmos de aprendizado de máquinas. É o caso, por exemplo, da classificação de tráfego. A realização de classificação pelas ferramentas convencionais, como identificação de portas ou avaliação do *payload*, normalmente não atende a todas as necessidades das novas aplicações, visto que estas aplicações podem não usar portas padrões e a inspeção do *payload* pode demandar muitos recursos ou ser inviabilizada em função do uso de criptografia ou de políticas de privacidade. Nesse sentido, o uso de algoritmos de aprendizado de máquina pode satisfazer a necessidade de novas formas de classificação rápida e eficaz sem a necessidade de altos volumes de processamento ou quebras de privacidade [41]. Embora o acesso a muitos pacotes de rede para realização de inspeção ainda seja uma realidade distante para as redes móveis em função de sua abrangência geográfica, tal tarefa pode ser alcançada em ambientes de *datacenter* [8], onde o uso de algoritmos de aprendizado de máquina pode ser feito de forma centralizada, obtendo as informações necessárias nos pacotes de dados e identificando falhas de rede em função das mesmas.

Por reunir diversas fontes de dados, as redes de telecomunicações móveis tornam difíceis as implantações de um sistema de monitoração baseados em aprendizado de máquinas supervisionado, dada a grande quantidade e diversidade de origens das informações. Uma abordagem que pode se tornar eficaz seria assumir que as diversas origens de informação são geradores constantes de dados em uma série temporal, de forma que algoritmos não supervisionados possam tratar cada série e seus padrões como também a relação entre diversas séries, como descrito em [3]. Como demonstrado em [9], verificações não automáticas em ambiente de redes tem se tornado uma tarefa árdua e complexa em virtude dos diversos sistemas interconectados e da natureza distribuída das informações que compõem esses sistemas. Nesse artigo, com o uso de aprendizado de máquina, foi possível determinar a causa raiz de falhas em diversos sistemas usando algoritmos de árvore de

decisão. Porém com informações que indicam uma situação de falha ou anomalias.

Como demonstrado Singh [40], para alcançar um bom desempenho na classificação automática do tráfego de rede, pode ser necessária uma captura parcial ou total do tráfego da rede. Em redes móveis, esse trabalho pode se tornar inviável, dada as dimensões da rede e dos caminhos usadas pelos fluxos da rede. Com a explosão no volume de dados das redes moveis, interfaces cada vez mais robustas têm sido usadas e realizar a captura de dados em ambientes com interfaces de alto tráfego sem interrupção do serviço pode se tornar inviável. Por essa limitação, o trabalho descrito nessa dissertação se limitou apenas a informações possíveis de serem coletadas sem qualquer risco de interrupção do serviço. Como indicado em [19], a capacidade e propriedade dos algoritmos de aprendizado de máquinas em relacionar as diversas variáveis com as suas respectivas variáveis de saída já auxiliam na detecção de padrões de falhas e ataques que possam estar ocorrendo em redes de comunicação, não só reduzindo a quantidade de variáveis necessárias para avaliação de um possível ataque, como também indicando quais são as variáveis mais importantes e necessárias para uma correta avaliação. Nesse estudo, as propriedades de redução de variáveis foram utilizadas a fim de constatar quais variáveis possivelmente estão relacionadas com falhas em algumas aplicações.

Nessa dissertação, foi possível o uso de algoritmos de aprendizado de máquina na tarefa de correlacionar variáveis de rede sua relação com degradações e indisponibilidades do serviço de emulação de circuitos de forma simples e não custosa para os elementos de rede. A proposta apresentada nessa dissertação está baseada na realização das coletas das variáveis disponíveis e o uso de algoritmos de aprendizado de máquina para obter a relação entre as variáveis e os estados dos serviços no momento da coleta de forma que seja possível identificar as variáveis de rede que possuem maior contribuição para as degradações ou interrupções dos serviços. Com essa proposta não é necessário à ocorrência ou detecção de anomalias na rede, como visto em outros trabalhos, visto a ocorrência de anomalias não detectáveis possa existir, como demonstrado no caso em estudo.

# Capítulo 5

# Proposta do Sistema MAREMOTO

Como observado no capitulo 2 o crescente numero de elementos de rede e a complexidade das redes móveis faz com que a verificação das interrupções ou degradações nos serviços móveis torna-se uma tarefa complexa. Nesse cenário observa-se também a ocorrência de falhas ou degradações nos serviços sem qualquer relação com falhas de rede, quando se monitora a mesma através das ferramentas de monitoração padrão *SNMP,ICMP,CLI*. O objetivo da ferramenta MAREMOTO é o uso de algoritmos de aprendizado de máquinas com a tarefa de obter informações relevantes das variáveis de rede que possam estar diretamente relacionadas com indisponibilidades ou degradação nos serviços móveis. O trabalho se aplica no cenário no qual a qualidade de experiência do usuário pode ser afetada, no entanto, as ferramentas de monitoramento de rede padrão, podem não conseguir detectar ou correlacionar o problema automaticamente. A ferramenta proposta pode ser utilizada para relacionar qualquer variável de rede e qualquer serviço móvel que utilize a rede desde que o mesmo possa ter seu *status* avaliado. A proposta da ferramenta MAREMOTO pode ser observada na figura 5.1.

## 5.1 Arquitetura da proposta

A proposta de sistema de Monitoramento Avançado de REdes MÓveis de múlTiplas geraçÕes (MAREMOTO) tem como objetivo criar uma forma de coletar informações da rede automaticamente, tais como topologia, atrasos entre os elementos e uso das filas de QoS, e coletar também informações dos serviços, como contadores de perdas de pacotes, para determinar se ocorreram falhas nos serviços. Assim, pode-se ser verificar quais variáveis de rede estão relacionadas com os status dos serviços. Isso é importante, pois identificando essas variáveis e as relações entre elas e o serviço, pode-se obter um melhor

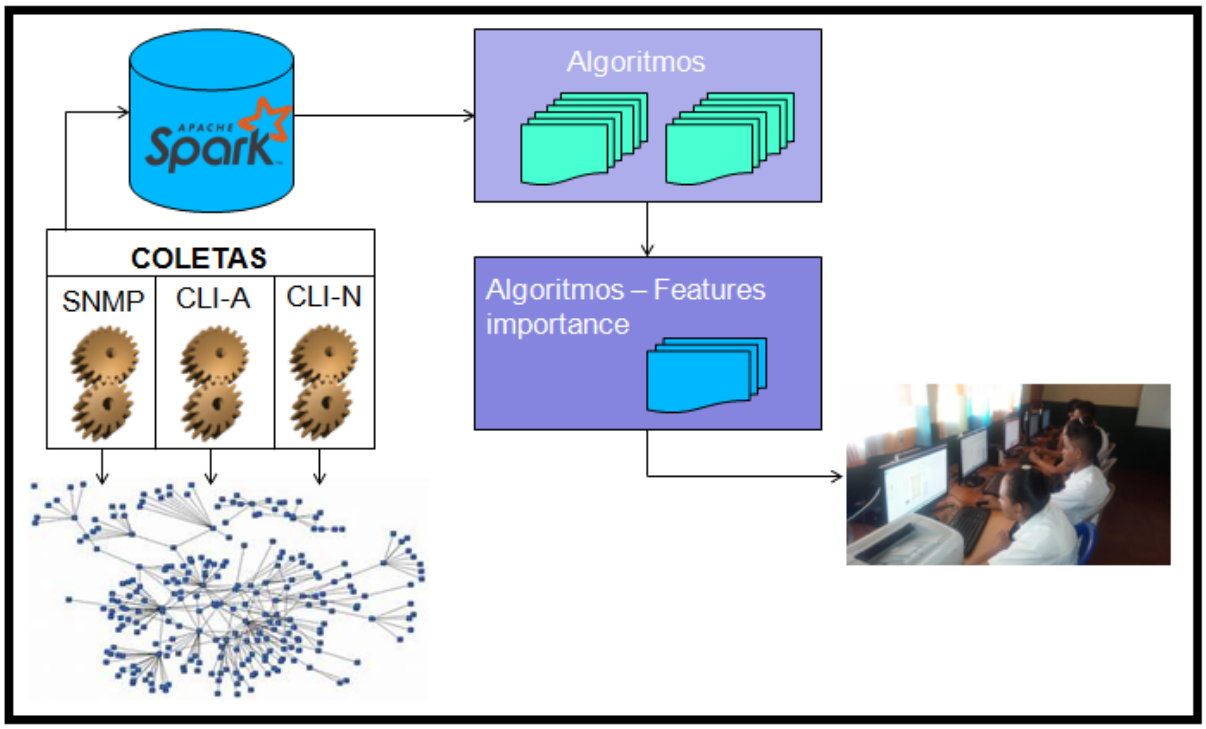



Figura 5.1: Arquitetura de funcionamento da proposta de monitoramento de redes móveis MAREMOTO.

uso da rede, além de melhorar o desempenho da emulação de circuitos determinísticos via rede *MPLS*, visto que o uso continuo do serviço 2G para aplicações *M2M* continuam crescente na rede móvel. Na Figura 5.2, se observa um diagrama básico para o sistema proposto.

## 5.2 Módulos

Para uma melhor compreensão e implementação/atualização mais simples, a proposta foi dividida em três módulos:

- Módulo de coleta: O módulo de coleta utiliza módulos a fim de realizar as tarefas relacionadas à criação dos inventários de dados sobre a rede e seus serviços. As informações coletadas abrangem informações dos elementos de rede e informações dos serviços que utilizam a rede.

  Inventários: Os inventários são criados a cada coleta realizada com base nos serviços ativos e monitorados. Tais inventários também são consultados pelos próprios sub módulos de coleta a fim de que possam saber não só quais são os serviços, mas também os roteadores disponíveis na rede.

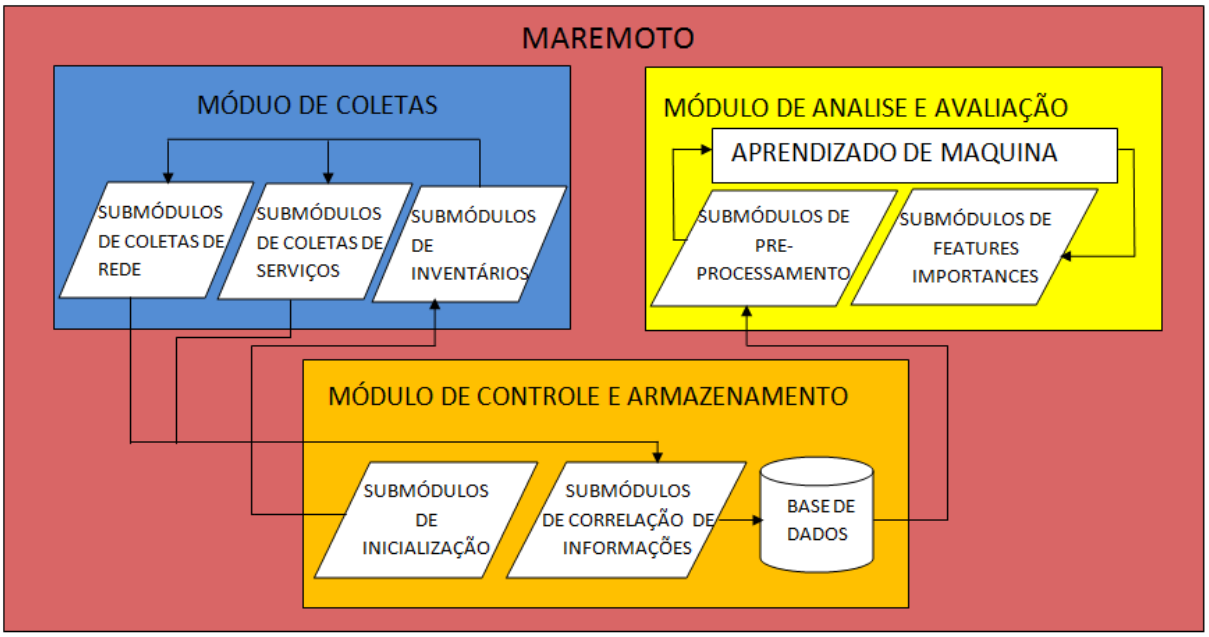



Figura 5.2: Arquitetura da proposta de sistema de monitoramento de redes móveis MAREMOTO.

Coletas de serviços: O sub módulo de coleta de serviços utiliza as informações do sub módulo de inventários para coletar as informações relacionadas aos status dos serviços ativos na rede.

Coletas de rede: O sub modulo de coleta de rede utiliza as informações do sub módulos de inventários e do sub modulo de coletas serviços para coletar todas as informações de rede necessárias para avaliação da mesma em relação a todos os serviços a serem verificados.

- Módulo de Controle e armazenamento: Esse módulo possui sub módulos que realizam o controle e armazenamento dos inventários do pertencente ao módulo de Coleta. O módulo de controle tem como funções principais inicializar todos os módulos na ordem correta e verificar a disponibilidade das informações requeridas por cada um dos módulos. O módulo de armazenamento disponibiliza uma função simples com o objetivo de armazenar todas as informações coletadas pelos módulos de coleta na base *Spark* de forma que tais informações possam estar relacionadas com as o intervalo de cada coleta.

- Módulo de análise e avaliação: Os sub módulos disponibilizados pelo módulo de análise e avaliação usam as informações armazenadas na base *Spark* e implementam algoritmos de aprendizado de máquina para obter as informações de peso das variáveis *Feature Importance* dos algoritmos de aprendizado de máquina, a fim de

obter qual ou quais variáveis possuem maior relação com o estado dos serviços.

Os detalhes de cada subsistema são descritos a seguir.

### 5.2.1 Módulo de Coleta

O módulo de coleta é composto por um conjunto de sub módulos que permitem acessar os equipamentos da rede e armazenar os dados coletados. O módulo de coleta tem como requisitos ser multiplataforma, já que deve acessar equipamentos de diferentes fabricantes, além de ser escalável, já que redes móveis usualmente apresentam redes de acesso de larga escala.

#### 5.2.1.1 Geração de Inventários

A fim de gerar um inventário contendo todos os elementos de redes a ser usado durante a coleta, o módulo Geração de Inventários realiza duas funções:

- Geração do inventário dos elementos de acesso - Com essa função, uma lista com todos os IPs dos roteadores que possuem apenas um processo OSPF, possuindo ou não serviços de emulação de circuitos, é gerada e fica a disposição dos diversos módulos a serem usados no decorrer das coletas. Essa lista é gerada através de uma requisição na base de dados dos sistemas de gerência, limitando a busca apenas a roteadores que possam ter os serviços que estão sendo monitorados ativos na região determinada.
- Geração de inventário de elementos de distribuição ou de núcleo da rede - Com a função de geração de inventário de distribuição, uma lista com todos os IPs dos roteadores que possuem as funções de agregação de rede, ou seja, que possuem mais de um processo de roteamento IGP ou mais de uma área interna, é gerada através de uma requisição na base de dados dos sistemas de gerência, limitando-se à área que está sendo monitorada na rede.

#### 5.2.1.2 Coleta de Serviços Ativos

O módulo Coleta de Serviços Ativos usa as informações geradas pelo módulo Geração de Inventário dos elementos de acesso para realizar a verificação de todos os serviços de emulação ativos na rede, os IPs dos roteadores usados para o estabelecimento desse serviço, bem como os contadores de desempenho de cada serviço.

#### 5.2.1.3 Geração de Informações de Equipamentos

O módulo de Geração de Informações de Equipamentos tem como função gerar um dicionário que possui como chave o IP do elemento e como valor uma tupla contendo as seguintes informações: versão de hardware, versão de software e nome do roteador. Essa informação é obtida através do arcabouço [35], que permite a obtenção de diversas informações dos elementos de rede via *Simple Network Management Protocol*(SNMP).

#### 5.2.1.4 Geração de Informações de IPs Ativos

O módulo Geração de Informações de IPs Ativos utiliza o módulo ios_cli do Ansible, que permite a obtenção de diversas informações dos elementos de rede via *Command-line interface*(CLI). A partir dessa coleta, é criado um dicionário contendo os IPs de identificação dos roteadores como chave e, como valores, uma lista com todos os IPs de interface desses roteadores. Esse dicionário permite relacionar todos os IP da rede e seus respectivos elementos.

#### 5.2.1.5 Geração de Topologia

O módulo Geração de Topologia tem como tarefa gerar a topologia da rede no momento da coleta. Essa informação é obtida através de consultas SNMP à *Management Information Base* (MIB) do *Open Shortest Path First* (OSPF) de todos os processos OSPF da rede. A geração da topologia é feita usando a biblioteca NetworkX do Python.

Do ponto de vista de rede, o mesmo arcabouço usado pelo protocolo OSPF para redes broadcast (eleição de *Designated Router*(DR) e associação do IP do DR na topologia com custo zero) foi usado para interligar todos os endereços IPs de um elemento ao seu IP de gerencia, pois é comum existir mais de um processo de roteamento IGP nas redes de acesso, isso ocorre para uma melhor distribuição das rotas pelos processos de roteamento criados. Porem com isso um roteador possui mais de uma identificação de roteamento *Router ID*(RID), logo a fim de que elemento fosse visto por apenas um IP, criou-se uma conexão com custo zero entre os IPs de um mesmo elemento. Para isso, o módulo Geração de topologia usa o dicionário gerado pelo módulo Geração de informações de IPs ativos. Assim é possível gerar uma topologia completa e coerente da rede.

Após a geração da topologia, a mesma é armazenada em forma de arquivo Pickle [33], que guarda a topologia em forma de base de dados para grafos.

#### 5.2.1.6 Geração de Caminhos

O módulo Geração de Caminhos relaciona os IPs de origem e destino dos serviços de emulação com os caminhos usados na topologia para cada serviço, usando o algoritmo *Shortest Path Firs* (SPF) na topologia gerada pelo módulo Geração de topologia. Para realização dessa tarefa, o módulo de geração de caminhos verifica os IPs usados por cada serviço e os associa às redes usadas para interconectar os roteadores usados ao longo do caminho usado pelo serviço. As interfaces de cada rede também são identificadas e disponibilizadas nessa associação.

#### 5.2.1.7 Geração de Relação de *Probes*

O módulo Geração de Relação de *Probes* tem como objetivo gerar um dicionário contendo como chave o IP de destino do serviço sendo monitorado para cada roteador que possui serviço ativo e associa-lo a uma probe, como os roteadores que possuem CEM podem possuir mais de uma probe, porem normalmente apenas serviços de emulação com mais um outro roteador, a associação com entre IP de destino e probe é realizada com o objetivo de buscar a probe próxima ao destino do serviço de emulação de circuitos. Para isso o módulo Geração de Relação de *Probes* usa as informações do módulo Geração de Informações de Equipamentos Ativos e do Módulo Coleta de Serviços Ativos e realiza uma consulta na rede para saber quais são os IPs de destino de cada serviço de emulação existentes na rede. As probes irão fornecer ao sistema de monitoração proposto, as informações de perdas de pacotes e delay nos caminhos usados pelo serviço de emulação de circuitos. Como mencionado no 2

#### 5.2.1.8 Estado de Caminhos

O módulo Estado de caminho usa as informações de interfaces e caminhos disponibilizas pelo módulo Geração de Caminhos e realiza a coleta de uso das filas de QoS de tempo real de todas as interfaces usadas por todos os caminhos. Dessa forma as coletas ficam restritas apenas as interfaces que são utilizadas para comutação de pacotes dos serviços de emulação de circuitos.

Nesse momento apenas as filas de QOS realtime estão sendo monitoras, visto que o serviço de emulação de circuitos utilizam essa fila de QOS.

Para cada caminho, as seguintes informações são geradas: tamanho de mínimo da fila de tempo real, uso máximo da fila e uso mínimo da fila.

#### 5.2.1.9 Coleta de Atraso

O módulo de coleta seleção de probe usa as informações dos módulos Geração de Relação de *Probes* e Geração de Topologia para obter a *probe* mais próxima ao destino do serviço.

Após a obtenção da *probe*, uma requisição SNMP é realizada para a mesma a fim de obter informações de atraso entre os roteadores executam o serviço sendo monitorado.

#### 5.2.1.10 Coleta de Perdas de Pacotes

O módulo Coleta de Perdas de Pacotes usa as informações dos módulos Geração de Relação de *Probes* e Geração de Topologia a fim de obter a *probe* mais próxima ao destino do serviço.

Após a obtenção dessas *probes*, uma requisição SNMP é realizada para a mesma a fim de obter informações de perdas de pacotes entre os roteadores que possuem o serviço sendo monitorado.

#### 5.2.1.11 Coleta de PTP

O módulo Coleta de *Precision time protocol*(PTP) usa as informações geradas pelos módulos Geração de Inventário dos Elementos de Acesso e Geração de Informações de Equipamentos para realizar requisições SNMP ou o Ansible com módulo de ios_cli para obter o estado do sincronismo dos roteadores. Essa informação é importante quando é feita a monitoração do serviço de emulação de enlace. A coleta de informações de status do sincronismo dos roteadores que fornecem os serviços de emulação de circuitos é importante, pois para que a estrutura do quadro *Pulse-Code Modulation*(PCM) esteja correta entre os dois roteadores os mesmos precisar esta com o relógio de clock na mesma frequência de oscilação. Como nem todos os modelos de roteadores possuem recursos para gerar informações de *Precision time protocol*(PTP) via *Simple Managment network protocol*(SNMP), as informações do módulo Geração de Informações de Equipamentos são necessárias, pois os as primeiras famílias de *Application Specific Integrated Circuits*(ASIC) de *Precision time protocol*(PTP) não tinham suporte a coletas via *Simple Managment network protocol*(SNMP).

As querys SNMP serão preferidas em relação às querys via *Command-line interface*(CLI), visto que as mesmas normalmente demandam uma quantidade menor de re-

cursos nos roteadores. Ocorre que querys realizadas por CLI precisam passar pela interpretação do comando e somente após isso se executa o processamento necessário. Já as querys SNMP requerem apenas o processamento direto das mesmas.

### 5.2.2 Módulo de Controle e Armazenamento

Os sub módulos de Controle e Armazenamento são responsáveis por realizar duas tarefas:

- Iniciar na forma correta as coletas de informação, dado alguns módulos dependem de informações que são obtidas por outros módulos.
- Correlacionar as informações de status geradas pelos diversos módulos a fim de obter uma base de informações integra e relacional. Tais informações são armazenadas em uma base Spark **??**, com o uso do Spark é possível realizar uma grande quantidade de tratamento e pré-processamento nos dados antes de submetê-los aos algoritmos de aprendizado de máquina.

## 5.3 Módulo de Análise e Avaliação

As informações geradas pelos subsistemas de coletas devem ser avaliadas e processadas a fim de se tornarem uteis para os administradores da rede. As tarefas do subsistema de analise e avaliação é verificar a correta disponibilidade das informações e aplicar as mesmas aos algoritmos de aprendizado de maquina, a fim de que as correlações necessárias entre variáveis de rede e status dos serviços possam auxiliar os administradores da rede. A figura 5.3 demonstra a interligação dos subsistemas da proposta.

### 5.3.1 Processamento de dados

Os dados coletados, gerados e correlacionados pelos diversos módulos tem como produto final uma matriz n-dimensional de dados, gerada na implementação utilizando o *Pandas DataFrame*. Cada coleta gera uma matriz, onde as linhas são os diversos serviços monitorados e as colunas, as variáveis de redes associadas ao estado do serviço.

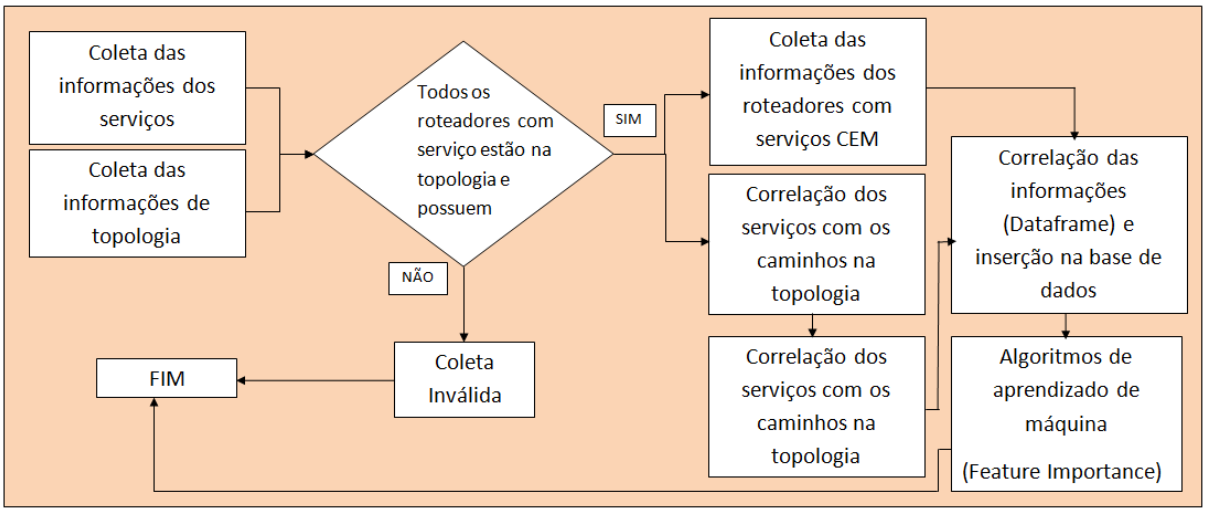



Figura 5.3: Fluxograma da proposta.

### 5.3.2 Análise das variáveis

Como resultado de cada ação de monitoração, o MAREMOTO realiza uma avaliação das variáveis de rede, correlacionando-as com o estado dos serviços monitorados. Nos algoritmos de aprendizado de máquina, cada variável apresentada é chamada de *Feature*, e o resultado ou saída é chamada de *target*. Assim, no MAREMOTO, as variáveis de rede são chamadas de *features* e os estados dos serviços são os *targets*.

#### 5.3.2.1 Princípios básicos da análise de variáveis - *Feature Importances*

Em diversos modelos de aprendizado máquina, não apenas a exatidão disponibilizada pelo modelo é importante, mas também a interpretação do mesmo, de forma que as relações e pesos das variáveis sejam facilmente interpretados. Por exemplo, ao usar um algoritmo de aprendizado de máquina para prever os preços de venda de uma casa, além da exatidão do modelo, também é importante entender quais variáveis estão fortemente ligadas com o resultado obtido. De forma semelhante, ao criar um sistema de aprendizado de máquinas para prever falhas em rede, não só a exatidão da previsão é importante, mas também quais variáveis levaram o modelo a obter esse resultado. Portanto, é importante conhecer as variáveis de rede que estão fortemente ligadas com as ocorrências de falhas.

Durante o processo de aplicação de algoritmos de aprendizado de máquinas, é recomendado que fosse minimizado a quantidade de *features* a fim de que seja mais simples o processo de identificação das *features* mais importantes. Isso se dá em função da complexidade dos modelos, pois quanto mais *features* usadas, mais complexo é o modelo gerado.

O processo de selecionar e restringir as *features* usadas é chamado de *feature engineering*. Sem esse processo, os modelos se tornam mais complexos e passíveis de erro, não apenas devido à complexidade, mas também devido à variabilidade das diversas variáveis.

Em função da necessidade de redução de variáveis e da grande importância da informação que relaciona cada variável com o resultado, diversos modelos oferecem informações que descrevem o impacto de cada *feature* com sua saída, bem como com outras features. A disponibilidade de informações referentes a importância de cada *feature* é muito comum nos algoritmos baseados em árvores de decisão (*Decision Tree models*), como *Random Forest*, *Gradient Boosting* e *Ada Boost*. Com esses algoritmos, é possível estabelecer as informações a respeito do impacto de cada *feature* em relação ao resultado, que, nesse caso, são os estados dos serviços.

A aplicação de *features importances* nas redes de telecomunicações auxilia provedores de serviço a alcançar melhores qualidades de experiência para o usuário. As redes móveis possuem uma grande quantidade de elementos e são dotadas de grande complexidade. Hoje, as diversas variáveis de rede são coletadas e analisadas por analistas em busca de soluções para falhas correntes nas redes. Contudo, esse trabalho é, na maioria das vezes, reativo, no que tange a identificar uma falha já ativa na rede. Embora existam ferramentas preditivas e automáticas em diversas redes, a correlação em busca da "causa raiz", normalmente, é realizada de forma pontual e manualmente.

O sistema proposto visa reduzir a necessidade de análises pontuais e manuais. Com o uso de *features importances* no MAREMOTO, as variáveis coletadas servem de dados de entrada para algoritmos de aprendizado de máquina, bem como os estados dos serviços servem como variáveis de saída. Assim, as ferramentas de *features importances* dos algoritmos de aprendizado de máquinas podem servir como ponto de partida para avaliações mais precisas com melhor e menor tempo de resposta para falhas e degradações nos ambientes de rede. Utilizando o MAREMOTO, analistas e engenheiros poderão identificar com facilidade a raiz do problema e trabalhar na variável de rede que está diretamente relacionada com a falha em análise.

## 5.4 Implementação da proposta

O sistema proposto pode ser implementado de diversas formas, visto que ferramentas para coleta, armazenamento e aplicação de algoritmos de aprendizado de máquinas estão disponíveis em diversos ambientes e linguagens. Para realizar uma validação da proposta,

o sistema foi implementado utilizando a linguagem Python e testado em uma rede de acesso real.

Foram criados módulos em Python de forma que as coletas pudessem ser realizadas de forma completa e coerente. Para a criação das matrizes de dados, utilizou-se o módulo *Pandas*[32] do Python. Assim, diretórios com as coletas foram criados e alimentados com as saídas dos diversos módulos, bem como com os *Pandas dataframes*. Após essas coletas, os diversos *dataframes* foram apresentados aos algoritmos de aprendizado de máquina. Foram utilizados algoritmos pertencentes ao grupo Árvore de Decisão, a fim de obter as informações de *features importances*. Com isso, é obtida a relação direta de cada *feature* com o estado do serviço.

Para a visualização das informações, foi usado o módulo [1]. Com essa ferramenta, foi possível a visualização gráfica da relação de cada *feature* com o estado dos serviços monitorados.

Nessa implementação do MAREMOTO, todas as informações são relacionadas e inseridas na base de dados além de serem armazenadas em arquivos [33]. Arquivos [33] são gerados pelo módulo [33], que permite serializar e de-serializar qualquer estrutura de dados provida pela linguagem Python. Assim, todos os módulos geram seus arquivos Pickle como saída e esses arquivos podem ser usados como entradas pelos demais módulos do MAREMOTO.

Apenas o módulo de correlação gera um *Pandas DataFrame* que é armazenado no formato *Comma-separated values* (CSV). Assim, todas as saídas de qualquer coleta estão disponíveis para análise dos algoritmos de aprendizado de máquinas, visto que o formato *Dataframe* é amplamente usado como arquivo de entrada dos algoritmos de aprendizado de máquinas.

### 5.4.1 Coleta de dados

As informações nas redes móveis reais estão disponíveis de diversas formas. Usualmente, os engenheiros conseguem obter boa parte das informações necessárias ao MAREMOTO, porém coletando-as em diversos sistemas proprietários que não disponibilizam uma interface para interoperabilidade. Assim, se fosse possível obter acesso direto de forma automática a tais informações, não seriam necessárias algumas das coletas do MAREMOTO e, consequentemente, reduziria-se o consumo de recursos da rede e dos elementos de rede. Por essa razão, nesse estudo, a coleta dos dados foi realizada de forma

autônoma e sem uso das informações já disponíveis nos diversos sistemas proprietários. Assim, as coletas foram realizadas de duas formas:

- SNMP - Todas as informações necessárias e disponíveis em MIBs SNMP foram coletadas via SNMP. As coletas via SNMP foram realizadas utilizando o módulo *netsnmp* do Python, que possui diversas facilidades para a realização de consultas SNMP. A preferência pelo SNMP se deu em função da economia de recursos tanto da rede quando da ferramenta proposta, já que no SNMP as informações já estão disponíveis nos elementos, de forma que os elementos não precisam gastar processamento ou memória para gerá-las.

- textitCommand-line interface (CLI) A outra forma utilizada pelos módulos de coleta para obtenção das informações necessárias foi a CLI. O acesso CLI foi realizado via [35], que permite acesso simultâneo e formação de diversos comandos em estruturas de dados próprias para Python. Com o uso do [35] é possível obter em forma de dicionario diversos comandos já tratados não em texto puro. Esse tratamento automático dado pelo [35] auxilia o tratamento das informações obtidas dos roteadores.

O estudo de caso utilizando essa implementação do MAREMOTO se deu em uma rede em operação, logo as restrições de acesso e disponibilidade de recursos de servidores e rede só permitiu o a realização das coletas a cada 30 minutos em um número restrito de equipamentos. Esse intervalo foi definido em função dos recursos de máquinas disponíveis para execução do sistema proposto, bem como do consumo de recursos de rede. Como se trata basicamente de consulta a comandos simples e querys SNMP, não foi observado incrementos expressivos de consumo dos recursos nos elementos de rede.

#### 5.4.1.1 Criação dos diretórios e coleta de dados de cada serviço

A função de controle presente no módulo de controle e armazenamento inicializa via cron realizando a criação de um diretório para cada coleta. Esses diretórios são criados usando a data atual no formato A-M-D-H:M, que significa Ano, Mês, Dia, Hora e Minuto, respectivamente. A criação de diretórios com esse formato pode auxiliar em verificações com base em tempo, permitindo relacionar falhas ou a ausência das mesmas com determinadas horas do dia.

Após a criação dos diretórios, a função de controle inicializa os processos de criação do inventário de elementos e coleta das informações de serviços. Essas ações são realizadas

pelos módulos Geração de Inventários e Coleta de Serviços Ativos, respectivamente.

#### 5.4.1.2 Coleta e validação da topologia

Ao término da execução do módulo Geração de Inventário, são relacionados todos os IPs dos os elementos de distribuição com seus IPs de gerência. Isso é necessário, pois, em alguns elementos, diferentes IPs podem ser usados para diferentes processos OSPF. Após essa coleta, o módulo Geração de Topologia é iniciado a fim de gerar a topologia da rede. Essa topologia é armazenada em um arquivo do no mesmo diretório.

As topologias de rede de acesso possuem diversos processos OSPF sendo executados ao mesmo tempo. Isso se dá pelo fato de ser uma rede muito grande e dada à necessidade de conectividade entre elementos em anéis diferentes. Logo, para garantir que todos os elementos relacionados aos serviços monitorados estão representados, a topologia é avaliada. Essa avaliação observa se todos os IPs usados pelos roteadores que proveem os serviços estão presentes na topologia e se existe um caminho válido entre esses elementos. Caso esse caminho não exista, a coleta é reiniciada.

Com a topologia validada, o módulo de controle inicia os módulos: Geração de Caminhos e Geração de relação de probes. Com esses módulos é possível obter o caminho usado para cada serviço de emulação, bem como as probes a serem usadas para coleta de desempenho do caminho.

Com as informações geradas pelo módulo Geração de Caminhos, o módulo de controle do subsistema de Controle e Armazenamento inicia o módulo Estado de Caminhos a fim de coletar as informações de quantidade de elementos no caminho, bem como os tamanhos e usos da fila de *Quality of Service* (QoS). Assim, todos os serviços ficam associados às informações de rede dos seus respectivos caminhos. Para cada caminho, o sistema registra o tamanho máximo da fila de QOS, tamanho mínimo da fila de QOS, o uso máximo e uso mínimo da fila de QoS.

Após a execução dos módulos acima, o módulo de Controle também inicia, nessa sequência, os seguintes módulos: Geração de Informações de Equipamentos, Coleta de PTP, Coleta de atraso e Coleta de Perdas de Pacote. Essa ordem é necessária, pois as informações sobre versão de software e hardware, bem como o nome do elemento são necessárias para determinar à forma de acesso as informações de PTP e quais as probes em uso, visto na rede em teste os nomes das probes estão relacionadas com os nomes dos elementos.

Com isso, passa a ser possível identificar atrasos em serviços, perdas de pacotes por serviço e também o estado do sincronismo de cada roteador da rede de acesso.

Após a coleta de todas as informações de rede, uma nova coleta dos contadores de desempenho dos serviços é realizada. Dessa forma, é possível identificar se houve degradação do serviço durante o processo de coleta.

Após a finalização de todos os processos de coleta, o módulo de controle realiza a correlação das informações coletadas e gera uma tabela com base no *Pandas DataFrame*. Todas as informações são armazenadas no formato pickle nos respectivos diretórios pelo módulo de armazenamento. As informações disponibilizadas para o Subsistema de Processamento de Dados estão mostradas na Tabela 5.1.

### 5.4.2 Implementação da análise de variáveis - *Feature Importances*

Nos algoritmos de árvore de decisão, o aprendizado de máquinas ocorre em função da criação automática da árvore, onde os algoritmos realizam diversas operações com intuído de obter um melhor desempenho e precisão na construção da árvore. Para a implementação do MAREMOTO, foram utilizados diversos algoritmos de aprendizado de máquinas disponibilizados pelo módulo *scikit-learn*.

Cada modelo de árvore de decisão possui particularidades especificas na forma de calcular a importância de cada variável. Porém, de forma genérica, a importância com que cada variável se relaciona com as demais e principalmente com o resultado está atrelada a fração de vezes que a variável é usada na árvore para tomada de decisão. Dessa forma, as variáveis que estão nos nós mais próximos ao topo da árvore são mais usadas para tomada de decisão em relação as variáveis que estão nos nós que se localizam na parte inferior da árvore. Dessa forma, essas variáveis tendem a possuir um peso maior no modelo gerado. Além da posição das variáveis na árvore, o grau de impureza da decisão tomada em função dessa variável também é levado em consideração ao se calcular a importância da variável.

No MAREMOTO, os *Pandas DataFrames* resultantes de cada coleta são apresentados aos algoritmos de aprendizado de máquina, a fim de que os mesmos possam identificar as variáveis que possuem maiores pesos no estado do serviço. Assim, torna-se possível identificar as variáveis que mais se relacionam com as falhas nos serviços de CEM. A

Figura 5.4 demonstra como esta proposta descrita é implementada e como os diversos módulos interagem entre si.

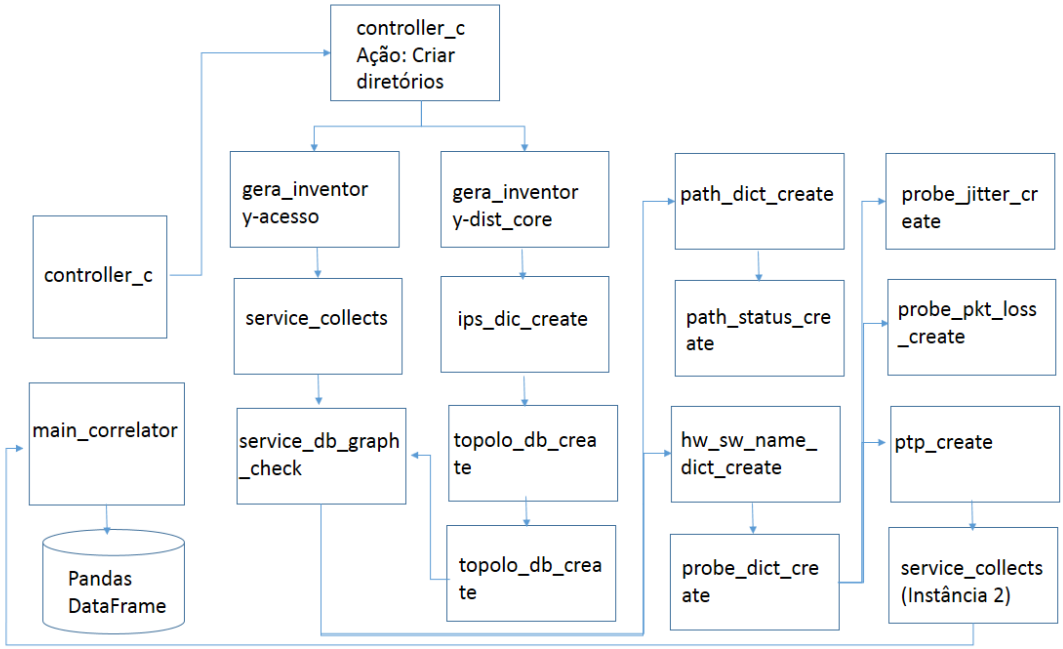



Figura 5.4: Fluxograma da proposta

A fim de avaliar o uso da proposta um estudo de caso para análise de um problema não detectável em nível de gerência e monitoração de rede, mas que afeta o serviço que deveria ser provido. Esse cenário se refere a redes móveis de segunda geração que utilizam as tecnologias de emulação de circuitos (SATOP / CESOP), bem como protocolos de sincronismo (PTP) como meio de transporte sob uma rede de pacotes IP/MPLS, coexistindo com outras gerações. O uso do serviço de emulação de circuitos para avaliação da proposta esta relacionado com o fato de tais serviços possuírem informações de fácil a acesso a respeito do funcionamento e desempenho do circuito emulado TDM através de uma rede *IP/MPLS* consiste de coletar e analisar o tráfego da rede, permitindo identificar quais variáveis que podem ser monitoradas para indicar possíveis falhas no serviço.

Nesse contexto, o uso de algoritmos para detectar os fatores predominantes nos cenários com erros de aplicação se tornou necessário, devido a:

- complexidade das redes de acesso - Como descrito no Capítulo 2, as redes de transporte apresentam gerenciamento complexo devido ao suporte para o acesso das redes móveis. De fato, as redes atuais precisam disponibilizar tecnologias suficientes para acomodar e transportar todas as gerações de redes móveis, bem como serem escaláveis, apresentando uma arquitetura como a mostrada na Figura 1.1.

- natureza dos serviços de 2G emulados - Os serviços de emulação de circuitos sobre redes IP/MPLS possuem requisitos agressivos no que tange a perdas de pacotes e atraso. Com isso, se torna difícil o uso de ferramentas tradicionais de verificação, ex: SNMP, ICMP etc.

Tabela 5.1: Informações disponibilizadas para os algoritmos de aprendizado de máquina.

| Informação | Significado | Forma de acesso |
|---|---|---|
| SERVIÇO | Identificação do circuito emulado usado pela BTS | Informação obtida através da coleta de todos os circuitos emulado na rede via Ansible com módulo de cli. |
| CAPAC LNK | Verificação da existência de links não ópticos no path do CEM | Essa informação obtida através a verificação dos labels das interfaces e da presença de sub-bandas nas interfaces. |
| HW MODEL | Modelo de hardware do roteador CEM mais próximo a BTS | Informação coletada via Ansible, usando sub módulo para elementos de rede. |
| SW VER | Versão de software ativo do roteador CEM mais próximo a BTS | Informação coletada via Ansible, usando sub módulo para elementos de rede. |
| DELAY MAX | Valor de atraso máximo entre os roteadores CEM responsáveis por prover o circuito emulado | Valor obtido através, das probes de IPSLA disponíveis na rede. |
| DELAY MIN | Valor de atraso mínimo entre os roteadores CEM responsáveis por prover o circuito emulado | Valor obtido através, das probes de IPSLA disponíveis na rede. |
| PATH LEN | Quantidade de elementos de camada 3 existentes entre os roteadores CEM responsáveis por prover o circuito emulado | informação de caminho obtida diretamente da topologia coletada. |
| PTP STATUS | Status do sincronismo do roteador CEM mais próximo a BTS | Essa informação pode estar disponível via SNMP ou via CLI, dependendo do modelo de HW. Para os modelos com SNMP disponível, é feita uma consulta SNMP usando o módulo netsnmp. Para os demais, foi usada a CLI via módulo [21]. |
| PTP CHANGE | Verificação de possíveis mudanças de status no sincronismo do roteador CEM mais próximo a BTS | Essa informação foi obtida através da comparação do status coletado e o status do PTP da coleta anterior. |
| Q BW | Valor mínimo de banda da fila de tempo real em todo o caminho usado pelos roteadores CEM | Essa informação é obtida através da verificação de todas as interfaces do path e, para esse caso, utilizou-se o Ansible com módulo de CLI. |
| Q MAX UTIL | Valor de banda máximo da fila de tempo real em todo o caminho usado pelos roteadores CEM | Essa informação é obtida através da verificação de todas as interfaces do caminho e, para esse caso, utilizou-se o Ansible com módulo de CLI. |
| Q MIN UTIL | Valor de banda mínimo da fila de real time mínimo de todo path usado pelos roteadores CEM | Essa informação obtida através da verificação de todas as interfaces do path e para esse caso usamos o Ansible com módulo de CLI. |
| STATUS | Estado do serviço | Essa informação é obtida através da comparação dos contadores de qualidade da coleta atual com a coleta anterior. |
| DATE | Hora e Minuto da coleta | Essa informação faz referencia não só ao tempo da coleta atual, como o diretório da mesma. |

# Capítulo 6

# Estudo de caso com serviços baseados pseudowire

O sistema proposto pode ser utilizado para monitoração de diversos serviços disponibilizados em redes de operadoras de telecomunicação. Para avaliar a eficiência da proposta, foi realizado um estudo de caso considerando um problema real em uma rede de acesso. Assim, a proposta de uso de algoritmos de aprendizado de máquinas para determinação da relação das diversas variáveis de rede na qualidade de um serviço foi implementada para verificação dos serviços de emulação de circuitos (*pseudowires*). A escolha dos serviços de *pseudowires* está relacionada com o fato de os mesmos poderem indicar situações de falhas e degradações, bem como a percepção de ocorrências de falhas, sem a correlação direta de falhas de redes ou detecção dessas falhas com ferramentas cotidianas de testes em redes, como testes de conectividade, coletas SNMP, etc. Assim, escolheu-se um serviço com um problema desafiador para os administradores e engenheiros responsáveis pela rede de acesso sendo analisada.

Para caracterizar o problema, inicialmente, foi feita uma medida das variáveis de rede utilizando os sistemas de monitoramento tradicionais e associou-se a evolução dessas variáveis ao estado do serviço ao longo do tempo. Durante cada coleta, informações normalmente usadas para refletir o estado da rede foram coletadas a fim de se obter alguma relação entre as mesmas e o estado do serviço. Essas informações incluem dados como perdas de pacotes, atrasos fim-a-fim, uso da fila de QoS de tempo real, a qual é usada pelo serviço de *psewdowire* e características dos roteadores envolvidos no serviço de emulação de circuitos. Essas informações foram coletadas a cada 30 minutos durante uma semana.

Como pode ser observado na Figura **??**, as características da rede como: Uso da fila

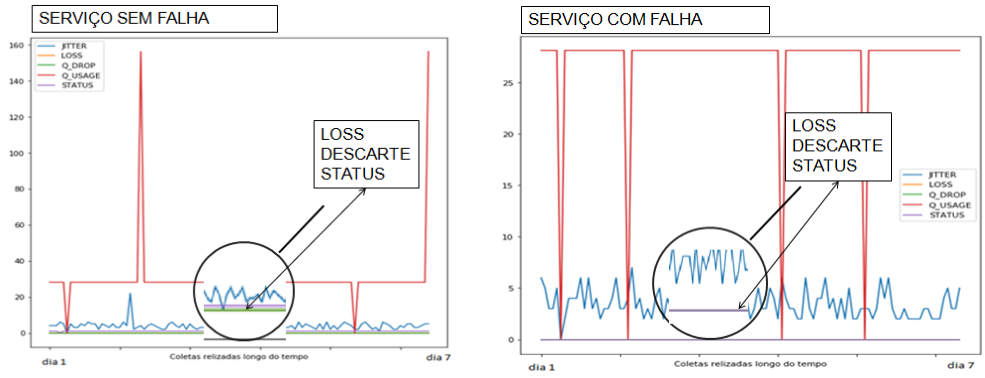



Figura 6.1: Parâmetros de rede Vs Status dos serviços

de QOS *Realtime*, variação do delay no caminho do serviço *Delay* etc. não apresentam grandes mudanças quando os serviços estão sem degradação, ou seja as características da rede estão normais quando não identificado falhas no serviço. Porém, as mesmas características também não apresentam variações fora do normal quando se observa momentos de degradação do serviço, como mostrado na Figura **??**. Percebe-se que o jitter esta com variações próximas tanto para serviços com falhas como para serviços sem falhas. O mesmo ocorre com o uso da fila de realtime, nessa ainda se percebe grande variação, tal variação esta relacionada com falhas nos serviços de Voz sobre IP. É possível perceber que o status do serviço permanece em zero, ou seja degradação constatada nos contadores do serviço CEM, mesmo com os parâmetros: de jitter, perdas de pacotes *loss*, bem como os parâmetros de descartes de pacotes e uso da fila de realtime $Q\_DROP$ *e* $Q\_USE$ respectivamente. Essas observações gráficas são comumente usadas na operação diária da rede, bem como refletem resultados de testes providos por ferramentas usadas para verificações de falhas na rede. Com isso, fica claro que os operadores da rede não tem como correlacionar o problema observado no serviço com o comportamento da rede, o que impede que possam identificar a raiz do problema e solucioná-lo com rapidez. Comparando as variações das variáveis de rede, percebe-se que não ocorrem mudanças bruscas independente do status do serviço. Os status das variáveis de rede tanto para situações de falhas dos serviços como para situações de serviços normais, pode ser observada na figura 6.1

O MAREMOTO, após realizar essas coletas, remove automaticamente coletas incompletas, realizando os processos de pré-processamento (*normalização*) dos dados. Na sequência, o sistema apresenta os dados aos algoritmos de aprendizado de máquina a fim de que tais algoritmos possam auxiliar na correlação entre essas informações e o estado do serviço.

## 6.1 Cenário de estudo

Afim de que a proposta pudesse ser testada, as coletas foram realizadas em uma região especifica que possui em torno de 1150 roteadores ativos com aproximadamente 600 circuitos TDM sendo emulados. O estudo de caso foi realizado nessa área, pois se observou que diversos sistemas de 2G estavam apresentando degradação sem nenhuma falha detectada na rede e/ou relacionadas com o caminho usado pelos *pseudowires*.

## 6.2 Escolha dos parâmetros

Nesse cenário de testes, coletaram-se parâmetros de rede que se julgava possivelmente relacionados com os status dos serviços, dessa forma os seguintes parâmetros de rede foram levados em consideração:

- Perdas de pacotes - A perda de pacotes pode ocorrer em qualquer momento em uma rede e por diversos motivos. Boa parte das ferramentas disponíveis para avaliação do estado da rede possui alguma informação a respeito do percentual de perda de pacotes corrente. Dessa forma, é necessário entender se as perdas estão fortemente ligadas à degradação dos serviços de emulação de circuitos. Tais perdas foram mensuradas durante os períodos de coleta com a obtenção dos valores de perdas mensurados pelas probes de IPSLA. A obtenção dessa informação se dá através de querys SNMP nos *Object identifiers*(OIDs) de cada probes relacionada com os serviços de emulação. Essas probes foram configuradas para usar como IPs de origem e destino os IPs dos roteadores responsáveis pela emulação de circuitos e para utilizar o mesmo valor de QoS configurado para as filas do serviço.

- Atraso e variação do atraso - O atraso sofrido por uma aplicação está fortemente ligado a distancia entre os elementos finais dessa aplicação, bem como com as condições de uso da rede. Como as condições de uso podem mudar ao longo do tempo, o atraso, por sua vez, também pode mudar. As variações no atraso, também chamadas de *jitter*, geram grandes degradações nas aplicações de rede como a emulação de circuitos, visto que tal aplicação trabalha com preenchimento e esvaziamento de *buffers* nos seus *end points* para recuperação de um sinal síncrono.

- Uso da fila de *QoS* - *RealTime* - Como o posicionamento dos elementos tende a não ser alterado em função de instalações físicas e suas respectivas áreas de cobertura,

o atraso gerado pela distância tende a ser sempre constante. Porém, as condições das filas de QoS podem gerar flutuações no atraso. Dessa forma, os usos máximo e mínimo das filas de QoS foram levados em consideração pelo MAREMOTO.

- Perdas na fila de QoS - Perdas de pacotes podem ocorrer por falhas de rede bem como por uso da mesma. No caso dos circuitos emulados, a fila de QoS usada está atrelada ao parâmetro de alta prioridade da rede (*realtime*), de forma que tais aplicações não sofram perdas e variações de atraso. Porém, as degradações dos circuitos emulados ocorrem em diversos momentos da rede, o que tornou necessário considerar possíveis faltas de recursos na fila de alta prioridade.

Nessa dissertação, vários dados coletados foram apresentados a diversos algoritmos de aprendizado de máquina, onde se observou, primeiramente, a precisão dos algoritmos em relação ao estado do serviço, visto que o estado do serviço já é conhecido. Os três primeiros algoritmos que apresentaram melhor precisão foram usados para se obter os pesos de cada variável e assim identificar as relações das variáveis com o estado do serviço. Como a proposta do estudo é encontrar a relação entre as variáveis e status dos serviços, buscou-se trabalhar com os algoritmos que conseguiram maiores precisão, ou seja, os que estão trabalhando próximo ou em *overffiting*, porem não em overffing. Nesse caso o uso de algoritmos de aprendizado de maquina que implementam funções Enssembre / Bagging que permitem usos próximos ao overffiting garantindo bons níveis de generalização. Para avaliações momentâneas se a necessidade de tarefas de classificação pode-se entender que a generalização não é necessária e a maior quantidade de informações foram absorvidas pelo algoritmo utilizando indicando assim que a variável de rede indicada pelo algoritmo como sendo de maior importância, terá maior relação com o status do serviço.

Ao apresentar as informações para os algoritmos de aprendizado de máquina, o esperado é que os mesmos possam apresentar quais variáveis de rede possuem maior correlação com o estado atual dos serviços de emulação. É importante ressaltar que o objetivo da proposta não é a obter a modelagem genérica que possa observar uma nova amostra e definir se o serviço de emulação de circuitos está com ou sem degradação. Isso não é necessário visto que durante a coleta já é possível determinar essa informação, pois a cada período de coleta duas observações a respeito dos contadores dos serviços são executadas: uma no inicio e uma próxima ao final. A determinação de possível falha ou degradação no serviço se dá ao verificar se o contador de pacotes não encontrados no *buffer* de saída. O objetivo do MAREMOTO é permitir que os operadores pudessem identificar com maior velocidade e precisão problemas da rede que estão afetando os serviços ofertados e que

Tabela 6.1: Precisão alcançada pelos algoritmos de aprendizado de máquina durante a tarefa de classificação

| Algortimos | Familia | Precisão | Cross-Validation |
|---|---|---|---|
| LinearSVC | Suport Vector Machine | 84.65% | 88.0% (+/- 0.001) |
| DecisionTreeClassifier | Decision Tree | 88.72% | 89.0% (+/- 0.011) |
| ExtraTreesClassifier | Randomic Decision Tree | 89.48% | 90.0% (+/- 0.010) |
| RandomForestClassifier | Randomic Decision Tree | 89.58% | 90.0% (+/- 0.010) |
| AdaBoostClassifier | Ensemble | 87.46% | 88.0% (+/- 0.006) |
| GradientBoostingClassifier | Ensemble | 90.96% | 91.0% (+/- 0.008) |
| GaussianNB | Bayesian | 85.79% | 69.0% (+/- 0.531) |
| LinearDiscriminantAnalysis | Dimensionality Reduction | 65.53% | 87.0% (+/- 0.010) |
| QuadraticDiscriminantAnalysis | Dimensionality Reduction | 65.533% | 27.0% (+/- 0.337) |
| LogisticRegression | Regression | 85.79% | 88.0% (+/- 0.007) |

não são evidentes utilizando ferramentas de monitoramento padrão.

A classe de dados chamada *feature_ importances* do algoritmos de árvore de decisão informam o percentual de importância de cada *feature* (x1,x2,x3...xn) em relação ao resultado alcançado (Y). O cálculo do peso de cada *feature* em relação ao resultado final é baseado na redução total (normalizada) que cada *feature* consegue obter na árvore de decisão. Se uma *feature* consegue chegar ao valor final de Y, ela possui maior peso em relação às *features* que necessitam de outras em seus ramos. Esse método de cálculo é conhecido como *Gini imputity*.

## 6.3 Resultados obtidos

Com as informações de precisão dos algoritmos de aprendizado de máquina, percebe-se que os algoritmos de aprendizado de máquina da família Árvore de Decisão conseguiram um melhor desempenho na tarefa de classificar o estado do serviço, como é mostrado na Tabela 6.1. Observa-se que os algoritmos *GradientBoostingClassifier*, *ExtraTreesClassifier* e *RandomForestClassifier* alcançaram os melhores resultados de precisão quanto a tarefa de classificação do estado do serviço.

Os algoritmos de aprendizado de máquina da família Árvore de Decisão possuem diversas características, como:

- Simplicidade de acesso e visualização - Além de implementar o acesso simplificado às varias informações definidas pelo algoritmo, é de implementação simples e permite

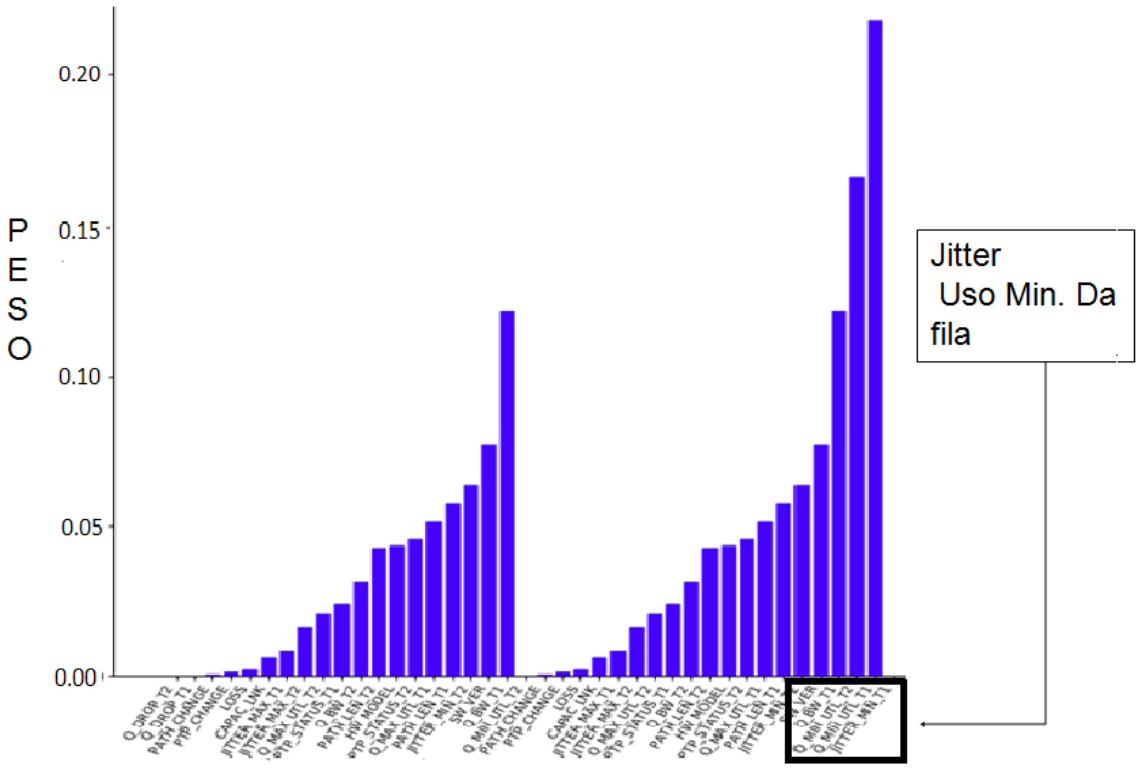



Figura 6.2: *Features* mais importantes associadas pelo algoritmo *GradientBoosting*.

a visualização da árvore definida para classificação das informações disponibilizas para o algoritmo, o que pode ser usado para validação do modelo, dependendo da dimensão da árvore.

- Implementação de acesso à informação de *feature importance* - Embora todas as *features* passadas para um algoritmo de aprendizado de máquina possam ser usadas para se obter o valor correto das variáveis de saídas, nem todas as informações são usadas, dado que algumas variáveis passadas para os algoritmos podem possuir informações irrelevantes ou até mesmo duplicadas. Dessa forma, os algoritmos de aprendizado de máquina precisam definir quais *features* devem ser usadas e seus pesos no resultado final da variável de saída. Cada algoritmo possui uma forma particular de adicionar pesos para as *features* durante a criação do modelo que melhor representa as informações contidas nos dados, mas nem todos apresentam essa informação de forma simplificada. Os algoritmos de aprendizado de máquina da família *decision Tree* obtiveram melhores valores de precisão na tarefa de classificação do status do serviço e permitem o acesso ao *rank* de importância de cada feature usada pelo algoritmo em relação à tarefa de classificação do status do serviço.

Como os algoritmos *GradientBoostingClassifier*, *RandomForestClassifier* e *ExtraTre-*

*esClassifier* são da família das Árvores de Decisão, os mesmos disponibilizam as informações a respeito da importância de cada *feature* de forma simples. A Figura 6.2 mostra a importância de cada *feature* durante a execução do algoritmo de *GradientBoostingClassifier* para tarefa de classificação das variáveis relacionadas ao estado do serviço de emulação de circuitos.

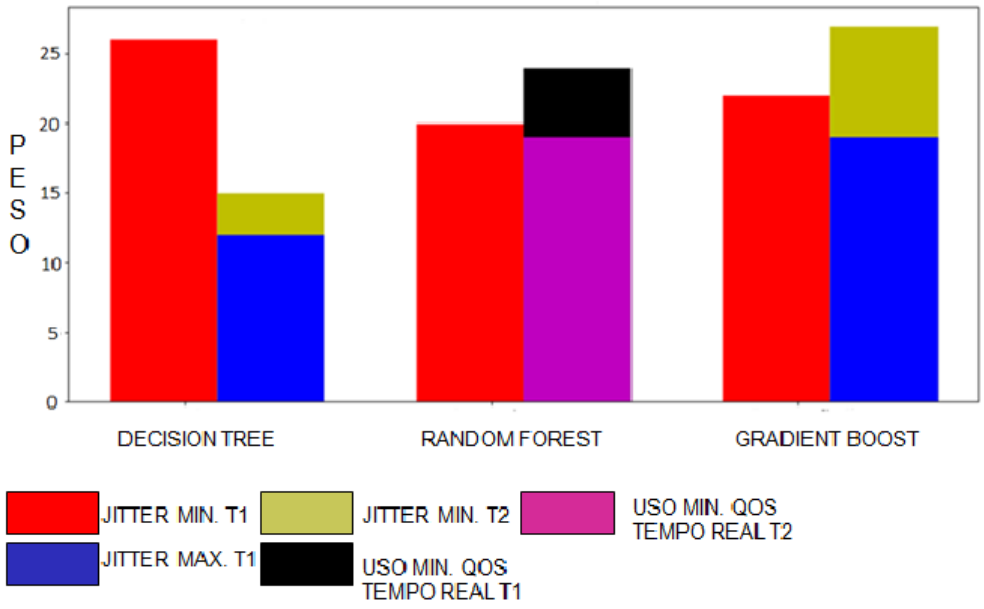
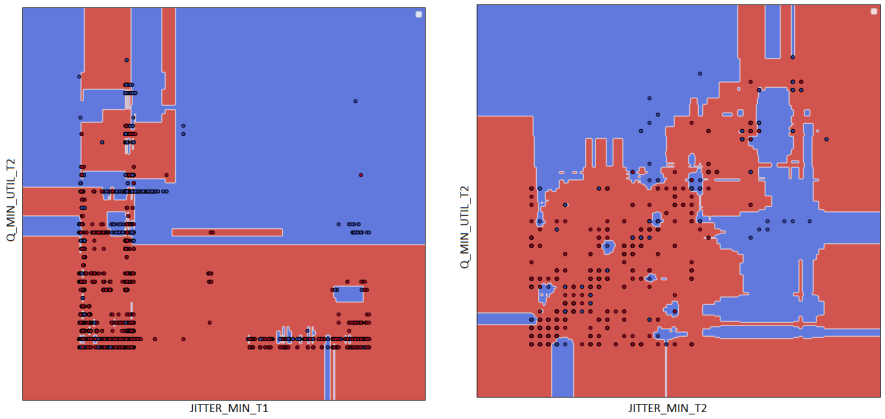



Figura 6.3: *Features* indicadas como mais importantes.

Observa-se que as *features* mais importantes para o algoritmo *GradientBoosting* foram o *jitter* mínimo e o menor uso da fila de QoS. Ao selecionar essas duas variáveis, é possível observar as fronteiras de decisão criadas pela árvore de decisão do algoritmo.

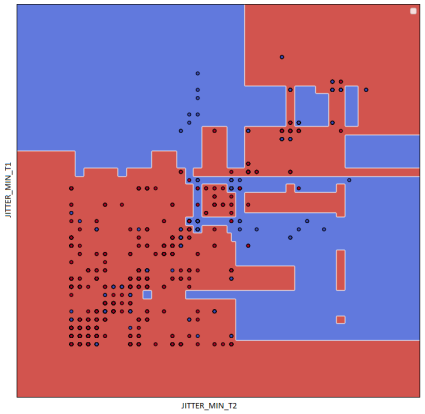
A fim de se obter a melhor relação das características da rede com o estado do serviço de emulação de circuitos, na implementação do MAREMOTO realizou-se o cálculo das *features importances* para cada intervalo de coleta para os três algoritmos que apresentaram os melhores desempenhos nos cálculos de precisão. A Figura 6.3 demonstra que as *features* que mas obtiveram um numero de ocorrências como feature mais importantes, onde é possível observar maiores ocorrências relacionas ao *jitter* possuem um peso expressivo em relação ao status final do serviço.

Com as informações de *features importance* de cada algoritmo, é possível visualizar de forma clara as superfícies de decisão dos algoritmos usados baseados nas *features* mais importantes de cada um dos algoritmos. Essas superfícies de decisão demonstram como os algoritmos conseguem capturar as relações entre as *features* e o estado da variável de




(a) Fronteira de decisão do algoritmo RandomForest

(b) Fronteira de decisão do algoritmo ExtraTrees




(c) Fronteira de decisão do algoritmo GradientBoosting

Figura 6.4: Fronteiras de decisão dos algoritmos que tiveram os melhores resultados de precisão.

saída. Normalmente, as *decision boundaries* demonstram as fronteiras onde a classificação da variável de saída é ambígua. A ambiguidade é removida quando se tem uma superfície plana, resultante de uma relação linear entre as *features* e a variável de saída, o que não é observado nas fronteiras de decisão abaixo, o que implica em uma não existência de uma relação linear entre as features e o status do serviço. A visualização das fronteiras de decisão *Decision boundaries* é importante à verificação visual da não presença de overffiting. A Figura 6.4 demonstra as *decision boundaries* dos algoritmos usados, usando apenas as duas informações que tiveram maiores valores no r*rank* de feature importance durante a execução do algoritmo. Já na Figura 6.5, observa-se a árvore de decisão criada pelo algoritmo *GradientBoosting*.

Nessa primeira implementação do MAREMOTO, observa-se que as *features* relacionadas com o *jitter* 6.3 foram usadas uma quantidade maior de vezes em relação às demais,

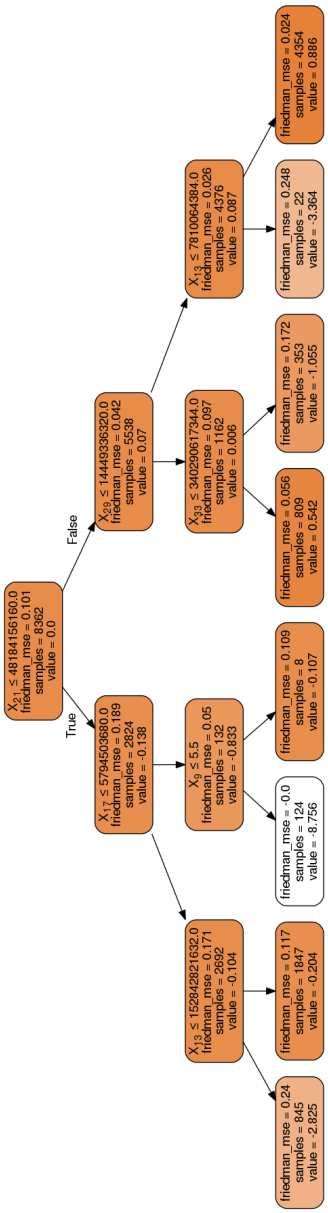



Figura 6.5: Árvore de decisão do algoritmo *GradientBoosting*.

bem como obtiveram maiores pesos em alguns algoritmos. Esse resultado é de difícil compreensão prática, pois os pacotes de emulação de serviço usam a fila de alta prioridade (*realtime*) e são de tamanho pequeno (256bytes). De fato, espera-se que tráfegos de pacotes pequenos na fila de alta prioridade não apresentem grandes variações de atraso. Uma provável causa para grandes variações do atraso seriam grande variações no tráfego em meios cujo QoS pode não estar sendo cumprido. Assim, para avanço do estudo de caso, foram realizadas novas coletas contendo novas variáveis, em uma segunda versão da implementação do MAREMOTO, a fim de obter novas correlações que sejam mais úteis aos operadores da rede.

Assim, novas coletas das variáveis de rede, incluindo também o uso das filas de QOS de cada interface, foram feitas com o MAREMOTO. Com essa coleta, esperava-se que se conseguisse a relação entre os perfis de tráfego (em função do uso da fila de QoS) com a condição de falhas no serviço. As filas de QoS que foram coletadas estão definidas na Tabela 6.2.

Tabela 6.2: Filas de QOS e seus tráfegos específicos.

| Filas | Tráfego típico |
|---|---|
| *Real Time* | Essa fila é destinada aos tráfego de altíssima prioridade, como trafego de voz e também o tráfego de *psewdowire*. |
| *Signaling* | Essa fila é destinada ao tráfego de sinalização. Esse possui alta prioridade. |
| *DataUser* | Essa fila é destinada ao tráfego de usuário. Normalmente, esse tráfego é caracterizado pelo tráfego de dados móvel. |
| CD/MGM | Essa fila é destinada ao tráfego crítico de gerenciamento e bilhetagem, bem como para as conexões que não são de alta prioridade. |
| *Default* | Essa fila é destinada para tráfego não classificado em nenhuma das filas anteriores e não recebe tratamento específico na rede. |

Ao realizar as novas coletas incluindo as informações de todas as filas de QoS, se fez necessário realizar, novamente, os testes de precisão dos algoritmos de aprendizado de máquina, incluindo os algoritmos que não obtiveram bom desempenho com as informações anteriores. Como as aplicações de QoS dependem muito do *hardware* e *software* usados, nesse estudo essas informações também são coletadas em todos os elementos de redes utilizados. A Tabela 6.3 apresenta as informações de QoS que foram incluídas nas novas coletas. Essas informações foram coletadas para todas as filas já descritas na Tabela 6.2. Dada a natureza da fila *RealTime*, algumas informações são inexistentes ou não utilizadas para essa fila específica.

As novas coletas passaram pelo mesmo processo de estandardização, a fim de serem processadas pelos algoritmos de aprendizado de máquina.

Tabela 6.3: Variáveis coletadas para cada fila de QoS.

| Variável | Informação |
|---|---|
| queue_30_drop_rate_bps | Essa variável possui a quantidade de drops em bps ocorrida nos últimos 30s. |
| queue_30_off_rate_bps | Essa variável possui a quantidade de tráfego em bps recebida na fila nos últimos 30s. |
| queue_NoBuffer | Essa variável informa a quantidade de exaustão da fila, ou seja, quantas vezes a fila estorou o *buffer*. |
| queue_bytes | Essa variável indica o tamanho da fila em bytes. |
| queue_deep | Essa variável indica quantas vezes a fila foi expandida automaticamente a fim de reduzir os descartes da mesma. |
| queue_drop | Essa informação demonstra a quantidade de descartes ocorridos na fila. |
| queue_limit | Essa variável indica o tamanho da fila. |
| queue_pkts' | Essa informação indica a quantidade de pacotes trafegados na fila. |

Na nova versão do MAREMOTO, foi observada uma ligeira diferença no desempenho dos algoritmos, como pode ser observado na Figura 6.6. Observa-se que, com a inclusão das informações das filas de QoS, os algoritmos GradientBoostingClassifier, AdaBoostClassifier e RandomForestClassifier obtiveram melhores resultados. Ao observar os valores de *feature importance* para os três algoritmos que apresentaram melhores desempenho, percebe-se que mesmo mantendo as *features* de rede usadas anteriormente (*jitter,loss, os version*, etc), as *features* de QoS receberam uma importância maior, como pode ser observado na Figura 6.7.

Verificando todas as informações de *features importance* para o algoritmo *GradientBoostingClassifier*, algoritmo que obteve melhor acurácia, percebe-se que as *features* atreladas às informações de *jitter* não só tiveram baixo índice de importância, como outras informações coletadas anteriormente receberam valores de *features importance* maiores. Ao verificar as features desse algoritmo percebe-se que as *features* de capacidade de enlace (*Link Capacit*) e perda (*LOSS*) receberam maiores valores de importância que as *features* de *jitter*.

Com a seleção das *features* mais importantes para cada algoritmo, é possível visualizar as novas fronteiras de decisão, como mostrado na Figura 6.8.

A fim de identificar qual fila possui maior relação com o status final do serviço, realizou-se o somatório da variável de *feature importance* para cada *feature* de cada fila,

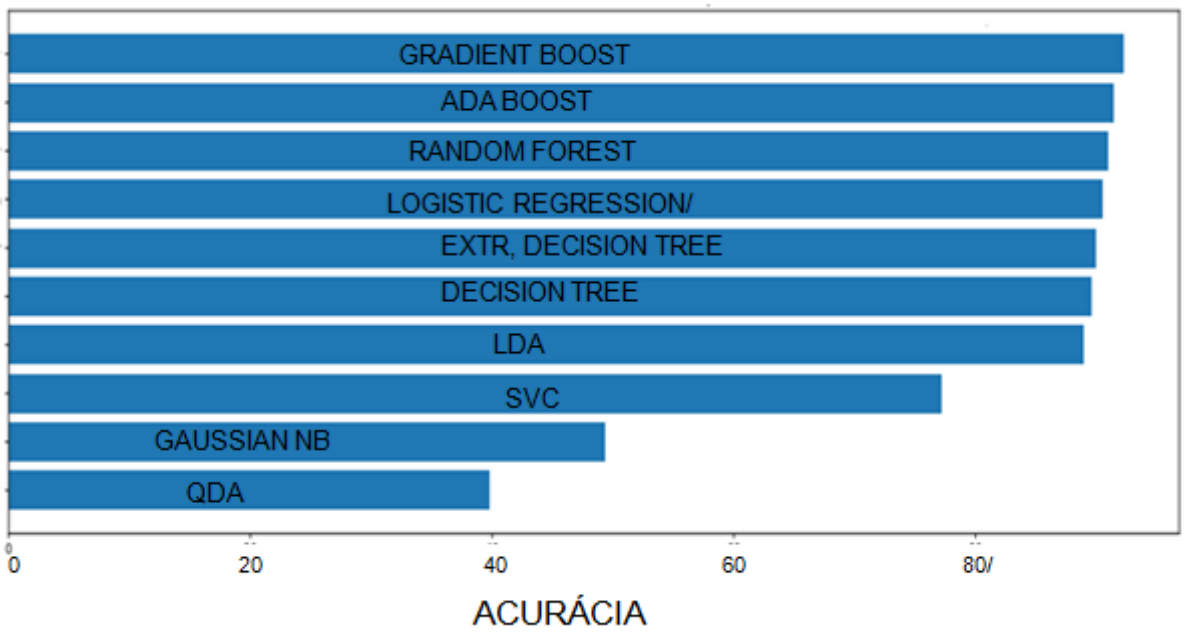



Figura 6.6: Acurácia dos algoritmos com adição das informações de QoS.

de forma que cada fila possa refletir o valor total de importância de suas *features*, visto que entre os três algoritmos que obtiveram os maiores índices de precisão não ocorreram repetição de features de maior importância em uma mesma classe de QoS. Ou seja, cada algoritmo teve como *feature* de maior importância uma *feature* dentro de uma classe de QOS diferente dos demais.

Realizando esse somatório percebe-se que os dois algoritmos que obtiveram os dois melhores índices de precisão tiveram a fila de QoS "Default"como a fila com maior importância, como pode ser observado na Figura 6.9. A Tabela 6.4 resume os dados observados, mostrando que a fila *default* concentrou os maiores índices de *features* com maiores importância.

A obtenção dos maiores índices e quantidade de *features* da fila *default* indica que o uso da mesma está fortemente relacionado com o estado do serviço, e que, de alguma forma, algum tráfego nessa fila pode estar gerando as falhas dos serviços de emulação, mesmo o serviço de emulação estando na fila *RealTime*.

Com base nesse resultado obtido com a ferramenta desenvolvida, buscou-se analisar a rede observando a característica indicada. Assim, a fim de constatar os efeitos dos tráfegos que trafegam nas filas não *realtime* sobre o serviço de emulação que trafegam na fila

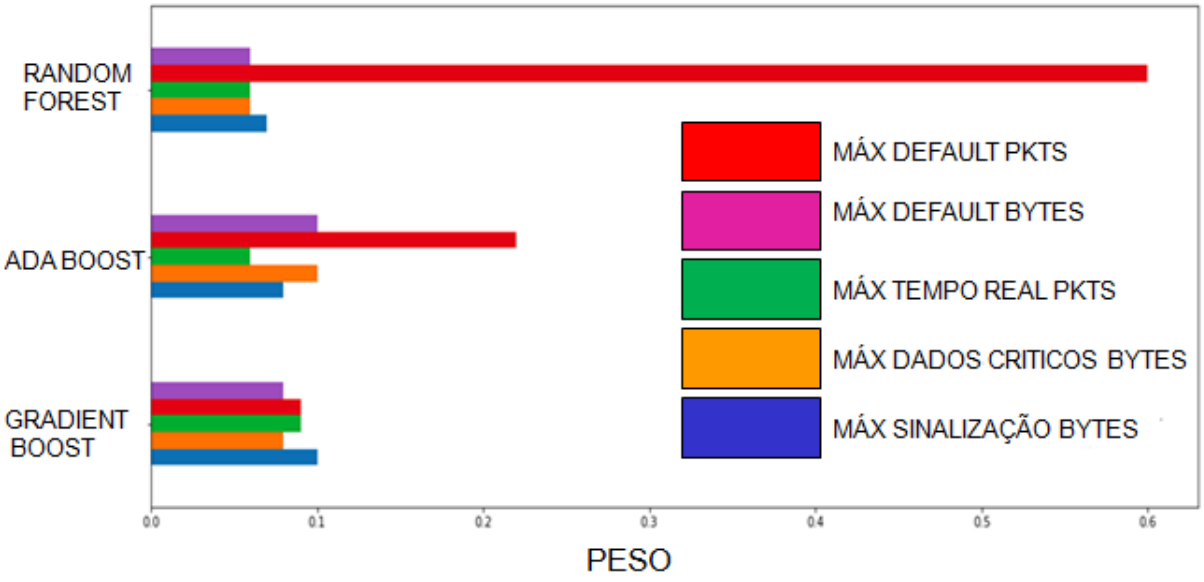
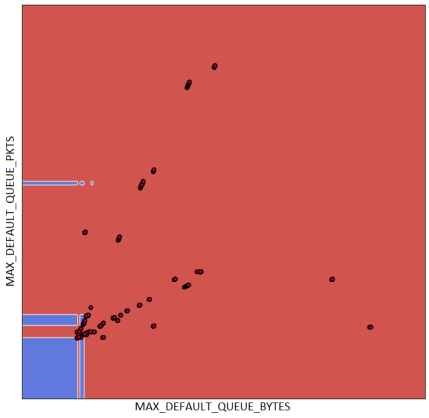



Figura 6.7: *Features* mais importantes quando se tem as informações de QoS.

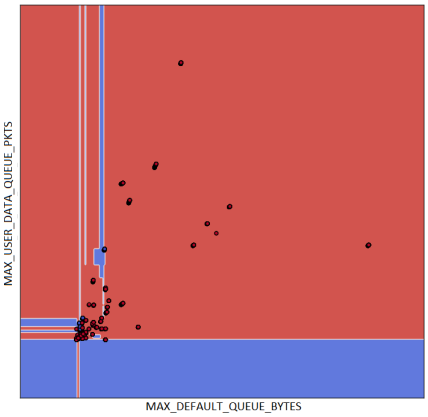
*realtime*, realizou-se uma redução do limite da quantidade de pacotes da fila *default*. Com isso, esperava-se observar melhorias no serviço de emulação de circuitos, como consequência direta dos resultados estabelecidos pelos algoritmos de aprendizado de máquina. As filas *default* normalmente apresentam um limite de 8k pacotes, e, como pode ser observado na Figura 6.10, com esse limite de pacotes, pode-se observar o serviço *pseudowire* com falhas.

De forma empírica, o limite da fila *default* foi reduzido em 50%, ou seja, o mesmo passou para 4000 pacotes apenas no primeiro link não 100% óptico no sentido do núcleo de rede para o acesso usado pelo serviço de emulação de circuitos. Após essa redução, é percebido um melhor desempenho nos contadores dos serviços de emulação. Embora não ocorra perdas em nenhuma das filas, percebe-se que, após a limitação da fila *default*, atinge-se o desempenho desejado para os serviços de emulação, como pode ser verificado na Figura 6.11.

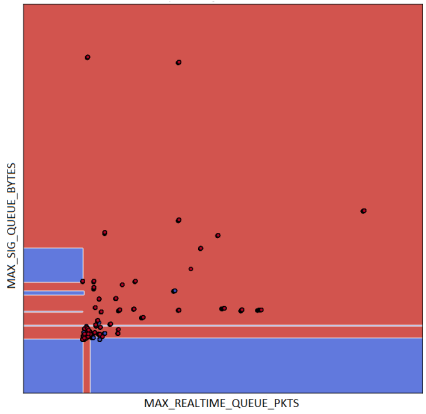
Além das variáveis de tráfego relacionadas com o estado do serviço, a próxima variável que possui maior importância é a variável LINK CAPAC, que está associada a caminhos que não possuem apenas enlaces ópticos. Isso indica que possíveis atrasos de magnitude muito pequena podem estar sendo inserido por esses meios face aos tráfegos da fila *default*. Tais atrasos são bem pequenos e não geram perdas de pacotes como percebidos pelos resultados das coletas de *probes*, mas podem ser correlacionados com as falhas dos serviços de emulação.




(a) Fronteira de decisão para o algoritmo AdaBoosting




(b) Fronteira de decisão para o algoritmo RandomForest




(c) Fronteira de decisão para o algoritmo GradientBoosting

Figura 6.8: *Decision Boundaries* com informações de QoS.

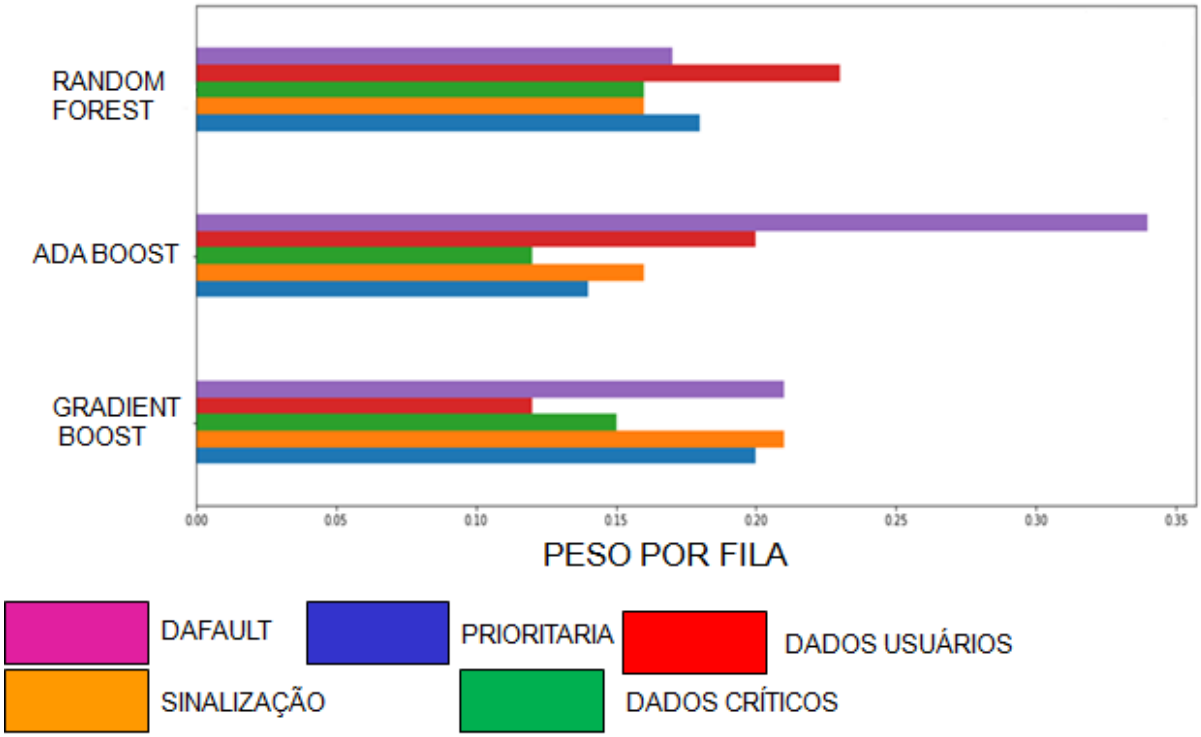



Figura 6.9: Filas mais importantes usando informações de QoS.

Com isso, esse estudo de caso permitiu mostrar que o MAREMOTO pode ser utilizado para a detecção de falhas de redes que não são observadas pelas ferramentas de monitoração tradicionais, ajudando na operação das redes de acesso usadas na telefonia móvel.

Tabela 6.4: Peso das filas durante a tarefa de classificação.

| Algoritm | Queue | Peso | Numero de Features |
|---|---|---|---|
| GradientBoostingClassifier | Real Time | 0.20 | 3 |
| | Signaling | 0.21 | 3 |
| | MGM | 0.15 | 4 |
| | User Data | 0.12 | 4 |
| | Default | 0.21 | 6 |
| AdaBoostClassifier | Real Time | 0.14 | 3 |
| | Signaling | 0.16 | 3 |
| | MGM | 0.12 | 1 |
| | User Data | 0.20 | 4 |
| | Default | 0.34 | 3 |
| RandomForest | Real Time | 0.18 | 3 |
| | Signaling | 0.16 | 7 |
| | MGM | 0.16 | 4 |
| | User Data | 0.23 | 7 |
| | Default | 0.17 | 7 |

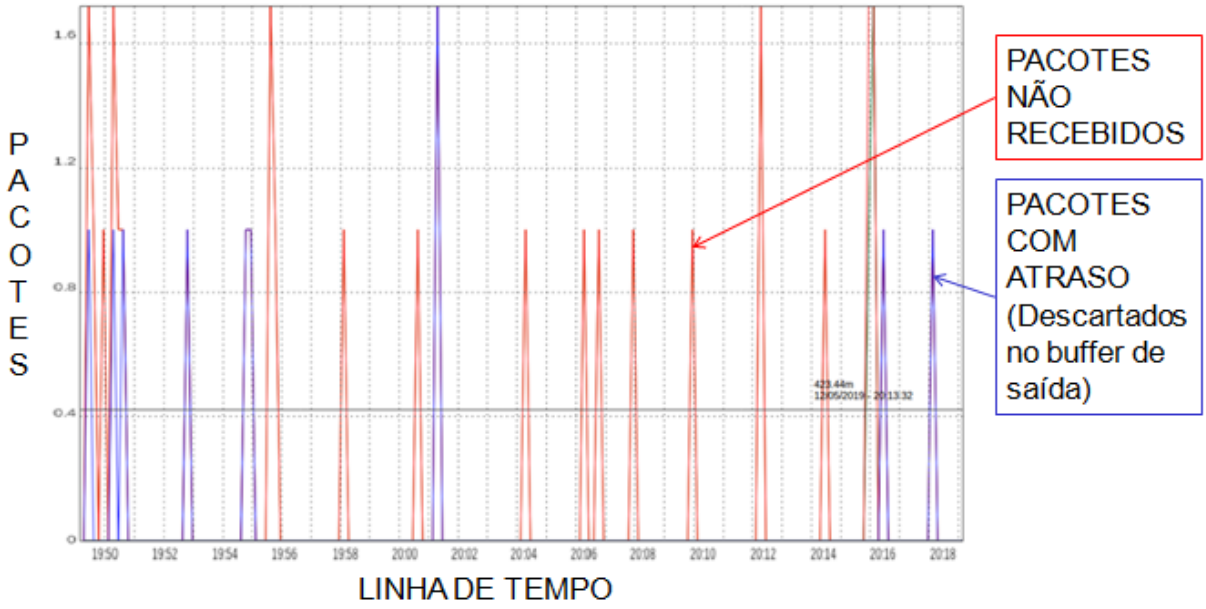



Figura 6.10: Serviço *pseudowire* com degradação ao utilizar a fila *default* com tamanho máximo de 8000 pacotes.

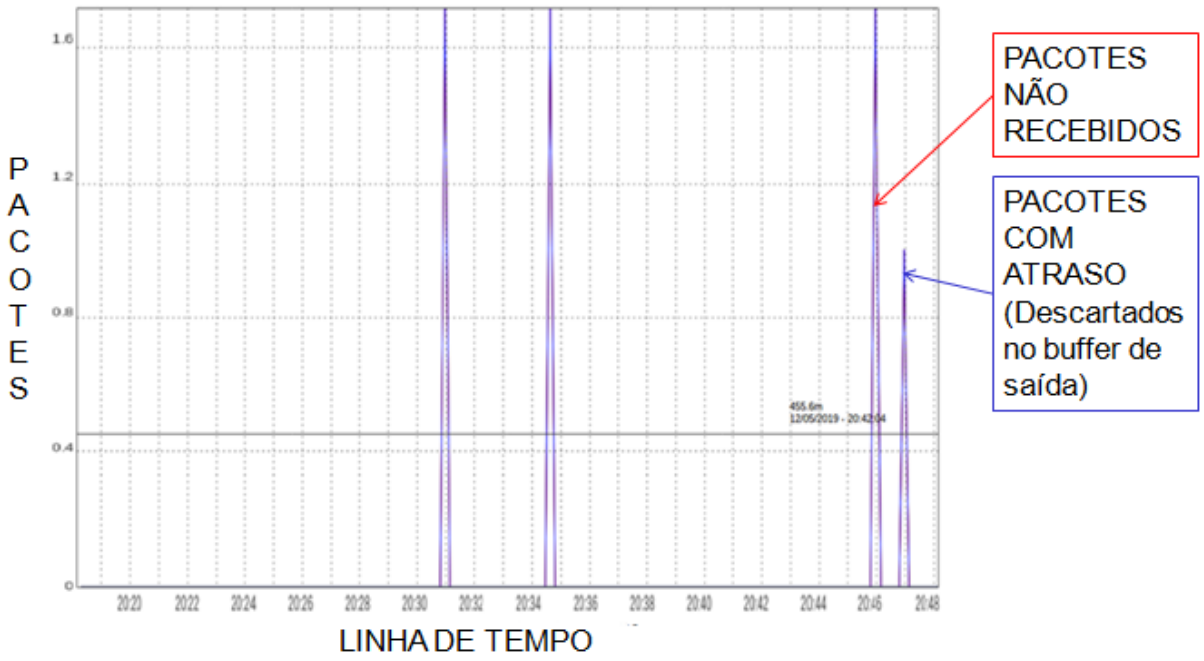



Figura 6.11: Serviço *pseudowire* sem degradação ao utilizar a fila *default* com tamanho máximo de 4000 pacotes.

# Capítulo 7

# Conclusão

A fim de validar não só o uso dos algoritmos de aprendizado de maquinas na melhoria da gestão de redes, mas também na relação das variáveis de rede com o estado dos serviços CEM, foi desenvolvido e testado em uma rede em operação o sistema MAREMOTO. O MAREMOTO realiza tarefas de coleta, armazenamento e processamento de medidas utilizando ferramentas de aprendizado de máquinas. Com o uso da ferramenta, foi possível detectar uma causa não trivial de problema em um serviço crítico da rede. Assim, fica clara a importância desse tipo de sistema para a gerência e operação de redes de telecomunicação de grande porte e complexidade.

O sistema desenvolvido armazena as informações coletadas em um *Pandas DataFrame*, correlacionando as informações relevantes. Tais *DataFrames* são, então, apresentados a algoritmos de aprendizado de máquina, a fim de se obter dados que permitam compreender e solucionar a causa dos problemas observados nos serviços. A assertividade dos algoritmos pode ser constatada visto que o estado dos serviços testados são previamente sabidos.

Ao estabelecer quais algoritmos possuem maior precisão na tarefa de classificação, a informação de pesos de cada *feature* é obtida dos algoritmos, a fim de se estabelecer uma relação direta entre as diversas variáveis de rede e o estado dos serviços.

Como demonstrado 6 os algoritmos de aprendizado de máquina foram capazes de encontrar diversas relações entre as diversas variáveis de rede e o estado dos serviços. Embora tenha sido necessário aumentar a abrangência da coleta de informações, o estudo de caso utilizando uma rede real em operação demonstrou que o uso de algoritmos de aprendizado de máquina para estabelecer relações diretas entre as diversas variáveis de rede e o estado dos serviços que utilizam a rede não só é possível, como também pode identificar relações não triviais com boa margem de precisão.

Ao indicar uma possível relação entre as falhas nos serviços e o volume de tráfego e, posteriormente, estabelecer uma relação entre o tráfego das filas específicas de QoS com falhas nos serviços de emulação, percebe-se que o uso dos algoritmos de aprendizado de máquina na determinação da causa raiz para falhas de serviços depende diretamente das *features* coletadas e da real existência de uma relação entre as *features* coletadas e o estado dos serviços.

De fato, os algoritmos de aprendizado de máquina não terão como indicar de forma assertiva uma relação entre as variáveis de entrada e o *target* se essa relação não existir. Da mesma forma, a apresentação do maior número de *features* possível aos algoritmos de aprendizado de máquina também não traria bons resultados. Mesmo que isso fosse possível sem a exaustão dos recursos de hardware alocados para os algoritmos de aprendizado de máquina, se faz necessário perceber que a necessidade de coletas extensas e exageradas pode levar a exaustão os recursos dos elementos de rede, visto que os mesmos são limitados.

Nesse estudo, foi possível a verificação da precisão das relações encontradas pelos algoritmos de aprendizado de máquina para as diversas *features* e as variáveis de *output* usando dados de uma rede em operação. Da mesma forma, também se verificou que alterações na rede a fim de limitar tráfego nas filas de QoS apontadas com principais ofensoras pelos algoritmos de aprendizado de máquina no estado dos serviços fizeram com que tais serviços alcançassem bons indicadores de desempenho, validando, assim, as relações geradas pelos algoritmos de aprendizado de máquina em uma rede em operação. Os resultados desses testes indicam que novos sistemas de monitoração podem se tornar mais eficazes se usarem de forma correta os diversos algoritmos de aprendizado de máquina para correlação direta entre as diversas falhas de serviços e as *features* de rede.

# Capítulo 8

# Trabalhos futuros

Como demonstrado no 6, com o uso dos algoritmos de aprendizado de maquina foi possível não só estabelecer uma relação entre as variáveis de rede e os status do serviço, como também realizar alterações manuais nas variáveis de rede associadas pelos algoritmos de aprendizado de maquinas e obter melhorias nos status do serviço. Embora alterações em redes sejam realizadas diariamente e comumente são realizadas em busca de: melhorias, correções, ativações ou desativações de novos serviços. Tais alterações podem não ser uma tarefa simples e podem gerar novas falhas, dessa forma as mesmas alterações manuais realizadas nas variáveis de rede que beneficiaram os serviços de emulação de circuitos podem também podem gerar falhas nesse caso especifico no serviço móvel de quarta geração. Embora o serviço de quarta geração utilize normalmente protocolos *TCP* e consequentemente sejam mais robustos a perdas de pacotes e *delay*, alterações de maior escaldas na fila desse serviço podem gerar falhas no mesmo. Assim se faz necessário que a ferramenta MAREMOTO possa também prever quais o melhores valores a serem alterados nas variáveis de rede, de forma que seja alcançada as melhores performance em todos os serviços prestados. Para realizar tal tarefa a ferramenta MAREMOTO deve obter informações a respeito de todos os serviços que utilizam a rede e seus status e qualidade. Além das informações dos demais serviços é essencial que futuramente a ferramenta receba um modulo capaz de relacionar as diversas alterações os status nos serviços que utilizam a rede, para que tais avaliações possam ser avaliadas. Dessa forma futuras implementações da ferramenta deve prever para um modulo de aprendizado de maquina semi-supervisionado. Tal módulo deve ser capaz de receber as informações de feature importance do modulo de Analise e Avaliação, prover as alterações automáticas nas configurações de rede e avaliar as consequências de tais alterações. A avaliação da consequência das alterações não se limita apenas aos serviços em degradação ou falhas que devem ser beneficiados com as

alterações, mas deve se avaliar também a consequência das alterações nos demais serviços da rede. Para realizar uma melhor avaliação de uma quantidade maior de serviços provavelmente será necessário a coleta de uma quantidade maior de informações, de forma que quanto maior a quantidade de informações únicas coletadas, maior será a assertividade fornecida pelo modulo de Analise e Avaliação. Embora os algoritmos de aprendizado de máquina possam receber um grande volume de informações, a acurácia dos mesmos se reduz dado que os mesmos passam a receber um volume exagerado de informações. Pois com tal volume os modelos gerados pelos algoritmos de aprendizado de maquina se tonam muito complexos. Para obtenção de um bom nível de acurácia frente a um grande volume de informações se faz necessário o uso de *Deep learn*(algoritmos de aprendizado profundo), como mostrado na figura 8.1 tais algoritmos conseguem obter um bom nível de acurácia mesmo com um grande volume de dados.

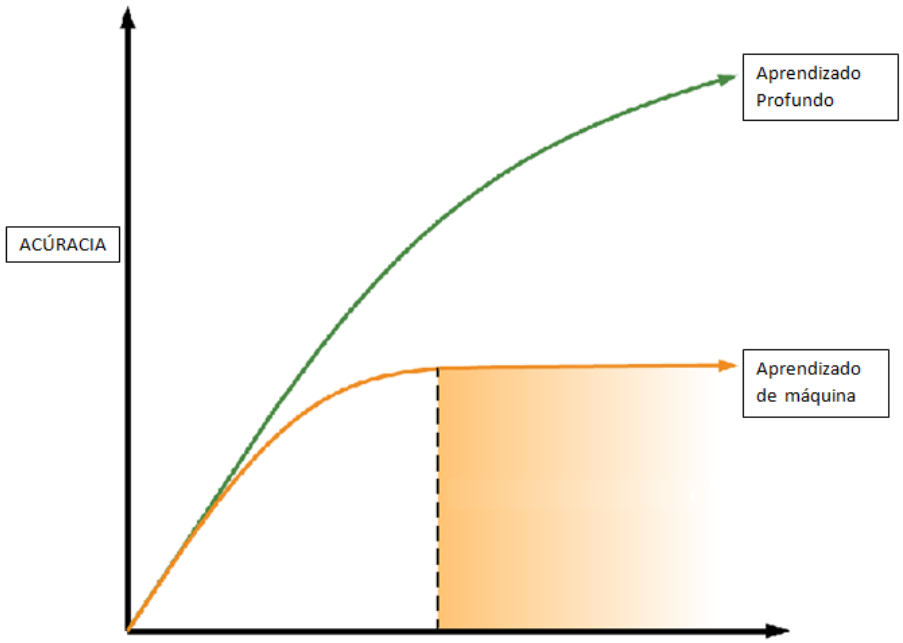



Figura 8.1: Acurácia dos algoritmos em função do volume de dados.

Dessa forma as futuras implementação da ferramenta proposta deve avaliar a possibilidade de algoritmos de aprendizado profundo afim de que um bom nível de acurácia seja obtido mesmo com um grande volume de dados.

# Referências


[1] John D. Hunter. Matplotlib.

[2] Adams, J. N. . N. Real-time dynamic network anomaly detection. *IEEE Intelligent Systems* (2018).

[3] Ahmad, S., Lavin, A., Purdy, S., Agha, Z. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* (2017).

[4] Ali, Z., Baldo, N., Mangues-Bafalluy, J., Giupponi, L. Machine learning based handover management for improved qoe in lte. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium* (abril de 2016), p. 794–798.

[5] Amaral, P., Dinis, J., Pinto, P., Bernardo, L., Tavares, J., Mamede, H. S. Machine learning in software defined networks: Data collection and traffic classification. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)* (novembro de 2016), p. 1–5.

[6] Association, G. The mobile economy. *GSM Intelligence Journal* (2019).

[7] Ayoubi, S., Limam, N., Salahuddin, M. A., Shahriar, N., Boutaba, R., Estrada-Solano, F., Caicedo, O. M. Machine learning for cognitive network management. *IEEE Communications Magazine* (2018), 158–165.

[8] Chandrasekaran, B., Srinivas, A., Zafer, M. System and method for using real-time packet data to detect and manage network issues. *United States Patent Vasseur et al.*, US10230609B2 (2016).

[9] Chen, M., Zheng, A. X., Lloyd, J., Jordan, M. I., Brewer, E. Failure diagnosis using decision trees. In *International Conference on Autonomic Computing, 2004. Proceedings.* (maio de 2004), p. 36–43.

[10] Cisco, C. V. N. I. Global mobile data traffic forecast update. 2017–2022 (white paper), 2017.

[11] Cusack, G., Michel, O., Keller, E. Machine learning-based detection of ransomware using sdn. In *Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization* (New York, NY, USA, 2018), SDN-NFV Sec'18, ACM, p. 1–6.

[12] Dai, X., Wang, N., Wang, W. Application of machine learning in BGP anomaly detection. *Journal of Physics: Conference Series Robotics and Artificial Intelligence* (2019).

[13] Dias, K. L., Pongelupe, M. A., de Errico, W. M. C. L. An innovative approach for real-time network traffic classification. *Computer Networks* (2019).

[14] Fadlullah, Z. M., Tang, F., Mao, B., Kato, N., Akashi, O., Inoue, T., Mizutani, K. State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS 19*, 4 (2017), 2432–2455.

[15] Fahmy, H. I., Develekos, G., Douligeris, C. Application of neural networks and machine learning in network design. *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 15*, 2 (1997).

[16] Gacanin, H. Knowledge-driven wireless networks with artificial intelligence: Design, challenges and opportunities. *Computer Science* (2018).

[17] Hood, C. S., Ji, C. Proactive network-fault detection. *IEEE TRANSACTIONS ON RELIABILITY 46* (97).

[18] Hughes, D. A. Multi-level learning for classifying traffic flows. *United States Patent Vasseur et al.*, US10257082B2 (2017).

[19] Iglesias, F., Zseby, T. Analysis of network traffic features for anomaly detection. *Mach Learn* (2015).

[20] Jr, G. F., Rodrigues, J. J. P. C., Carvalho, L. F., Al-Muhtadi, J. F., Jr., M. L. P. A comprehensive survey on network anomaly detection. *Telecommunication Systems* (2019).

[21] Ktbyers. Netmiko.

[22] Lim, H., Kim, J., Heo, J., Kim, K., Hong, Y., Han, Y. Packet-based network traffic classification using deep learning. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)* (Feb 2019), p. 046–051.

[23] Lynch, D., Fenton, M., Fagan, D., Kucera, S., Claussen, H., O'Neill, M. Automated self-optimization in heterogeneous wireless communications networks. *IEEE/ACM Transactions on Networking* (2019).

[24] Miao, X., Liu, Y., Zhao, H., Li, C. Distributed online one-class support vector machine for anomaly detection over networks. *IEEE Transactions on Cybernetics* (2019).

[25] Moysen, J., Giupponi, L. From 4g to 5g: Self-organized network management meets machine learning. *Computer Communications*, 129 (2018), 248–268.

[26] Nawir, M., Amir, A., Yaakob, N., Lynn, O. B. Effective and efficient network anomaly detection system using machine learning algorithm. *Bulletin of Eletrical Engineering and Informatics (BEEI)* (2019).

[27] Nguyen, T. T., Armitage, G. A survey of techniques for internet traffic classification using machine learning. *IEEE COMMUNICATIONS SURVEYS & TUTORIALS 10*, 4 (2008).

[28] Pacheco, F., Exposito, E., Gineste, M., Baudoin, C., Aguilar, J. Towards the deployment of machine learning solutions in network traffic classification: A systematic survey. *IEEE Communications Surveys & Tutorials* (2018).

[29] Park, J., Choi, D. H., Jeon, Y.-B., NamMin, Y., Park, H.-S. Network anomaly detection based on probabilistic analysis. *Soft Computing* (2018).

[30] Possebon, I., da Silva, Anderson, Granville, L., Schaeffer-Filho, A., Marnerides, A. Improved network traffic classification using ensemble learning. In *IEEE Symposium on Computers and Communications (ISCC) 2019* (4 2019), IEEE.

[31] Putina, A., Barth, S., Bifet, A., Pletcher, D., Precup, C., Nivaggioli, P., Rossi, D. Unsupervised real-time detection of bgp anomalies leveraging high-rate and fine-grained telemetry data. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (abril de 2018), p. 1–2.

[32] Python / Pandas community. Pandas.

[33] Python community. Pickle.

[34] Qiu, J., Du, Q., He, Y., Lin, Y., Zhu, J., Yin, K. Performance anomaly detection models of virtual machines for network function virtualization infrastructure with machine learning. In *Artificial Neural Networks and Machine Learning – ICANN 2018* (Cham, 2018), V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, Eds., Springer International Publishing, p. 479–488.

[35] Red Hat. Ansible.

[36] Saadon, G., Haddad, Y., Simoni, N. A survey of application orchestration and oss in next-generation network management. *Computer Standards & Interfaces* (2018).

[37] Saeed, A., Kolberg, M. Towards optimizing WLANs power saving: Novel context-aware network traffic classification based on a machine learning approach. *IEEE Access* (2018).

[38] Sampaio, L. S. R., Faustini, P. H. A., Silva, A. S., Granville, L. Z., Schaeffer-Filho, A. Using NFV and Reinforcement Learning for Anomalies Detection and Mitigation in SDN. In *2018 IEEE Symposium on Computers and Communications (ISCC)* (jun 2018), p. 00432–00437.

[39] Shi, Y., Zhang, Y., Jacobsen, H.-A., Han, B., Wei, M., Li, R., Chen, J. Using machine learning to provide differentiated services in SDN-like publish/subscribe systems for IoT. In *Service-Oriented Computing* (Cham, 2018), C. Pahl, M. Vukovic, J. Yin, and Q. Yu, Eds., Springer International Publishing, p. 532–540.

[40] Singh, H. Performance analysis of unsupervised machine learning techniques for network traffic classification. In *2015 Fifth International Conference on Advanced Computing Communication Technologies* (fevereiro de 2015), p. 401–404.

[41] Soysal, M., Schmidt, E. G. Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison. *Performance Evaluation* (2010).

[42] Sun, G., Liang, L., Chen, T., Xiao, F., Lang, F. Network traffic classification based on transfer learning. *Computers & Electrical Engineering* (2018).

[43] Thottan, M., Ji, C. Anomaly detection in ip networks. *IEEE TRANSACTIONS ON SIGNAL PROCESSING 51* (2003).

[44] Vasseur, J.-P., Mermoud, G., Dasgupta, S. Learning machine based detection of abnormal network performance. *United States Patent Vasseur et al.*, US 9,628,362 B2 (2017).

[45] Wang, P., Chen, X., Ye, F., Sun, Z. A survey of techniques for mobile service encrypted traffic classification using deep learning. *IEEE Access* (2019).

[46] WANG, Z., ZHANG, M., WANG, D., SONG, C., LIU, M., LI, J., LOU, L., , LIU, Z. Failure prediction using machine learning and time series in optical network. *Optics Express 25*, 16 (2017), 18553–18565.

[47] Williams, N., Zander, S., Armitage, G. Evaluating machine learning algorithms for automated network application identification. *CAIA Technical Reports* (2006), 1–14.

[48] Wu, Z., Dong, Y., Yang, L., Tang, P. A new structure for internet video traffic classification using machine learning. In *Proceedings of 2018 Sixth International Conference on Advanced Cloud and Big Data (CBD)* (08 2018), p. 322–327.

[49] Xiao, Y., Nazarian, S., Bogdan, P. Self-optimizing and self-programming computing systems: A combined compiler, complex networks, and machine learning approach. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* (2019).

[50] Yu, C., Lan, J., Xie, J., Hu, Y. Qos-aware traffic classification architecture using machine learning and deep packet inspection in sdns. *Procedia Computer Science* (2018).

[51] Zhang, J., Gardner, R., Vukotic, I. Anomaly detection in wide area network meshes using two machine learning algorithms. *Future Generation Computer Systems* (2019).

[52] Zhao, S., Chandrashekar, M., Lee, Y., Medhi, D. Real-time network anomaly detection system using machine learning. In *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)* (marï¿½ de 2015), p. 267–270.

[53] Zhou, Z., Zhang, T. Applying machine learning to service assurance in network function virtualization environment. In *Proceedings of 2018 First International Conference on Artificial Intelligence for Industries (AI4I)* (09 2018), p. 112–115.

[54] Zhu, M., Ye, K., Xu, C.-Z. Network anomaly detection and identification based on deep learning methods. *Lecture Notes in Computer Science 10967* (2018).