Capítulo 3

Metrologia na Era do Aprendizado de Máquina

Sidney Loyola, Antônio A. de A. Rocha, Aline Paes e José F. de Rezende

Abstract

The growth in the number of mobile and cellular networks arouses interest in end-to-end performance measurements of these networks and their impact on mobile applications [15]. Among these measurements, we find those based on the quality of the user experience, which is an excellent source of information about network management. In this context, the implementation of Artificial Intelligence and Machine Learning increases the efficiency of the monitoring process [21]. In this short course, we will discuss how we can use objective parameters of quality of services such as delay, throughput, packet loss, and jitter to estimate and predict the quality of user experience using Machine Learning [21]. Furthermore, it is possible to use Artificial Intelligence to estimate the multiple indicators of the quality of user experience, even in a network that adopts end-to-end encryption [49, 27]. In addition, Machine Learning methods help acquire knowledge about the functioning of networks, understand how the structures of existing networks are, and allow the establishment of new networks with less effort. Among the several Network Metrology problems that can benefit from Machine Learning, we can mention network attack detection, user experience quality prediction, application anomaly detection, and network failures.

Resumo

O crescimento da quantidade de redes móveis e celulares desperta o interesse em medições do desempenho fim-a fim dessas redes e o seu impacto em aplicações móveis [15]. Dentre essas medições encontramos aquelas baseadas em qualidade de experiência do usuário que constitui uma grande fonte de informações sobre o gerenciamento da rede. Nesse contexto, a implementação de Inteligência Artificial e Aprendizado de Máquina aumenta a eficiência do processo de monitoramento [21]. Abordaremos nesse minicurso de que forma podemos utilizar os parâmetros objetivos de qualidade do serviço tais como latência, vazão, perda de pacotes e jitter para estimar e predizer a qualidade de experiência do usuário utilizando Aprendizado de Máquina [21]. Além disso, é possível utilizar a

Inteligência Artificial para estimar os múltiplos indicadores de qualidade de experiência do usuário, mesmo em uma rede que adote criptografia fim-a-fim [49, 27]. Além disso, métodos de Aprendizado de Máquina auxiliam na aquisição do conhecimento sobre o funcionamento das redes, o entendimento de como são as estruturas de redes existentes, permitindo estabelecer novas redes com menor esforço. Entre os diversos problemas de Metrologia de Redes que podem se beneficiar da utilização do Aprendizado de Máquina podemos citar detecção de ataques de redes, predição de qualidade de experiência do usuário, detecção de anomalias em aplicações e falhas de redes.

3.1. Introdução

A expansão das funcionalidades da Internet fez com que o número de usuários crescesse, acarretando em mais de 4.66 bilhões de pessoas conectadas na web segundo relatórios do ano de 2021 [19]. Esse crescimento acelerado vem acompanhado da popularização de *smartphones*, *tablets*, *smart TVs* e *smart watchs*, criando um ambiente interconectado de dispositivos de diversos tipos conhecido como *Internet of Things* (IoT) [3]. Surgem, ainda, novos tipos de redes com características próprias, como as redes móveis, redes sem fio e as redes das *smart cities*. Além disso, a virtualização permite interconectar nós que são dissociados dos equipamentos físicos, criando, assim, diferentes topologias entre o mundo real e o mundo virtual, possibilitando a implantação de redes definidas e gerenciadas por software.

Redes móveis, redes sem fio e o ambiente IoT oferecem desafios singulares. O primeiro deles está relacionado à mobilidade e dinamicidade, uma vez que os dispositivos que fazem parte de tais redes não são fixos e nem estáticos; ao contrário, estão em constante deslocamento. Por exemplo, um celular conectado pode se mover de uma antena para outra vizinha da primeira, bem como pode ir, viajando de avião, por exemplo, de um continente para outro. Dessa forma, eles se conectam e se desconectam das redes com grande velocidade, dificultando a criação de modelos para entendimento e gerenciamento de tais conexões. Outra dificuldade está na heterogeneidade dos dados, sendo os mesmos dispositivos utilizados para assistir vídeos, jogar videogames, acessar contas bancárias e enviar mensagens de texto ou voz. Essa grande variedade de aplicações por si só adiciona complexidade às redes. Entretanto, com o avanço dos ambientes IoT não podemos nem determinar com precisão a origem do tráfego: por exemplo, é perfeitamente possível que uma cafeteira envie um e-mail, ou, até mesmo, envie uma publicação para uma rede social avisando aos colaboradores de uma empresa que o café está pronto.

Esse avanço tecnológico e utilização massiva de recursos computacionais impõem desafios que demandam a utilização de novos métodos e técnicas de gerenciamento, além do desenvolvimento de novas arquiteturas de redes. Por exemplo, a quantidade de informação transportada e a quantidade de dados disponíveis dos diferentes tipos de redes, torna os métodos analíticos para caracterização de tráfego ineficientes e custosos. Assim, algumas técnicas para entendimento e monitoramento, como as tarefas de classificação e predição de tráfego tornam-se um problema de *big data*. Outro desafio imposto pela evolução tecnológica refere-se à dificuldade de realizar medições diretas em redes, devido, em grande parte, à popularização da criptografia. É evidente que isso contribuiu para a privacidade e segurança das redes, porém dificulta o gerenciamento e monitoramento, forçando o desenvolvimento e aplicação de novas técnicas de medição [1].

A área de Metrologia de Redes aborda técnicas, ferramentas e métodos que permitem compreender o comportamento, dinâmica e propriedades das redes. Porém, atualmente, já não basta entendermos as redes, detectarmos falhas e anomalias; é necessário antever os acontecimentos para que ações preventivas possam ser realizadas. Para tanto, modelos de Aprendizado de Máquina (AM) podem ser utilizados para extrair informações úteis e padrões dos dados [6]. Com isso, o processo de tomada de decisão tende a ser mais eficiente e rápido, com o mínimo possível de interferência humana.

Entre as áreas em que modelos de AM têm contribuído para um melhor monitoramento e gerenciamento de redes encontram-se a classificação de tráfego, predição de tráfego, estimativa de qualidade de experiência do usuário, segurança de redes e gerenciamento de falhas. Modelos de AM de naturezas diversas têm sido aplicados com sucesso nos problemas apresentados, com destaque especial às redes neurais, que, além de lidarem com grande quantidade de dados, apresentam alto desempenho preditivo e a habilidade de aprender automaticamente representações, evitando a etapa de engenharia de atributos que seria realizada manualmente [1].

Outros dois cursos sobre metrologia já ocorreram em edições passadas do SBRC, um em 2005, Ziviani e Duarte [57], e outro em 2016, Rocha et al. [42]. Os conceitos apresentados neste minicurso revisitam os minicursos anteriores, apresentando as técnicas adotadas atualmente, diante das novas tecnologias e tendências de pesquisa. Dentre essas novas técnicas e tecnologias, a que mais se destaca é o aprendizado de máquina, em especial as redes neurais, que são aplicadas na resolução dos mais variados problemas.

Veremos no decorrer desse minicurso que os problemas de rede beneficiam-se de outras áreas da Inteligência Artificial, além do Aprendizado de Máquina. Como exemplos de outras áreas cujos métodos podem ser aplicados para modelar problemas de rede, temos a visão computacional e o processamento de linguagem natural. Em ambos os casos, os conceitos dessas áreas são utilizados para modelar o fluxo de dados, tanto por meio de representações de imagens, como por modelos de linguagem. Esse minicurso aborda os conceitos teóricos e fundamentos da Metrologia de Redes e o uso de Aprendizado de Máquina para resolver problemas envolvendo metrologia, de forma eficiente e rápida. Além da apresentação teórica contida nesse texto, forneceremos uma aplicação prática implementada e disponível em `https://github.com/loyoladesa/srbc2022`. A aplicação é autocontida e apresenta um exemplo de como aplicar os conceitos apresentados em um conjunto de dados extraídos de redes.

O principal objetivo deste presente minicurso, portanto, é apresentar as novas tecnologias de Metrologia de Redes. Isso inclui as recentes aplicações com Inteligência Artificial, a utilização em áreas emergentes e a discussão de problemas em aberto que podem fomentar novos trabalhos de pesquisa e desenvolvimento na área. Essas ferramentas e métodos têm influência direta em outras áreas de redes, como o planejamento de redes, engenharia de tráfego, garantia de qualidade de serviço e gerenciamento de redes [57].

O restante do minicurso está estruturado como segue. Primeiro, o minicurso oferece uma breve fundamentação teórica com conceitos de Aprendizado de Máquina, qualidade de serviços e qualidade de experiência do usuário na Seção 3.2. Em seguida, o minicurso discute as principais aplicações de AM para realizar medições e estimativas na Seção 3.3. Na Seção 3.4 é apresentado a forma como irá funcionar a apresentação do

estudo de caso. Finalmente, apresentamos nossas considerações finais na Seção 3.5.

3.2. Fundamentação Teórica

Esta seção tem como objetivo apresentar o estado da arte tanto de Metrologia de Redes quanto dos métodos de Aprendizado de Máquina. Além disso, serão discutidos conceitos de Qualidade de Serviço em redes (QoS) e Qualidade de Experiência do Usuário (QoE). A seção aponta as diversas definições e métodos existentes, de forma a conceitualizar de maneira hierárquica as tarefas, técnicas e problemas envolvidos.

Citaremos os métodos de Aprendizado de Máquina comumente utilizados, os principais experimentos e seus resultados e de que forma eles são utilizados para realizar medições em redes, tal que sejam identificados os modelos e variáveis de acordo com os problemas a serem investigados. Conforme apresentado em Casas [9], não há um método de Aprendizado de Máquina que atenda de forma genérica os problemas de medições de redes, sendo necessário encontrar um modelo específico para cada problema a ser investigado.

A metrologia de redes refere-se às técnicas utilizadas para monitorar, gerenciar e mensurar o tráfego em redes com um nível de detalhamento compatível com a tarefa desejada [1]. Esses conhecimentos são utilizados para entender como as redes trabalham, para monitorar a performance delas, descobrir como os recursos estão sendo consumidos pelo usuários e identificar como pode ser realizado um controle efetivo de modo que a rede forneça os requisitos dos acordos de níveis de serviço [1].

Para cumprir esses objetivos existem duas abordagens para realizar as mensurações. As medições passivas e ativas, como vistas na Figura 3.1:

- Medição passiva - basicamente realiza as medições observando o tráfego sem modificar o mesmo. Essas medições podem ser aplicadas em diversos pontos da rede e como exemplos cita-se o registro de perda de pacotes [42].
- Medição ativa - nesse caso é injetado tráfego padronizado na rede, também chamado de sonda, e monitorado o resultado da sua travessia pela rede, tornando possível extrair informações sobre os caminhos existentes [42].

3.2.1. Aprendizado de Máquina

Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial que visa equipar as máquinas com a habilidade de resolver problemas que requerem aprendizagem, obtida a partir de experiência. A principal motivação para o seu desenvolvimento é que nem todo problema pode ser modelado e resolvido utilizando um algoritmo pré-definido, ou seja, algoritmos que seguem um passo-a-passo para serem implementados.

Por exemplo, reconhecer uma pessoa a partir do seu rosto pode ser uma tarefa simples para os humanos, mas não é trivial especificar uma sequência de passos para a máquina resolver essa tarefa. Diante de cenários como esses, as técnicas de Aprendizado de Máquina constroem conhecimento sem serem previamente "programadas" para tal, ao invés disso elas vão "aprendendo" (melhorando o desempenho de alguma tarefa) a partir de exemplos [29].

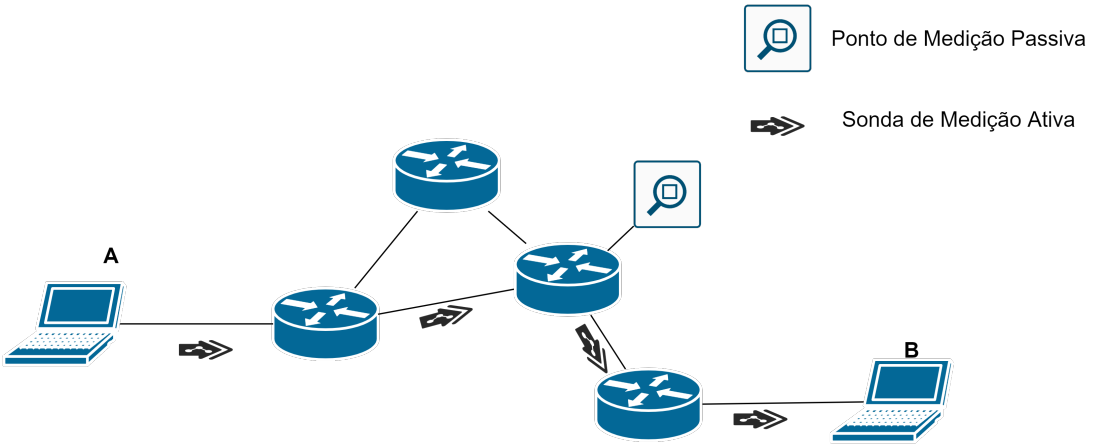

Figura 3.1. Exemplo de Medições. Figura adaptada de Ziviani e Duarte [57]

Este aprendizado é, na maioria das vezes, a busca por uma função alvo (desconhecida) capaz de resolver o problema proposto. Utilizando dados relacionados à tarefa (a experiência), os algoritmos induzem funções capazes de alcançar um determinado objetivo por si próprias. A experiência é comumente representada como o conjunto de dados compostos por exemplos – um exemplo é uma experiência individual e seus atributos – variáveis descrevendo a experiência [29].

Abaixo, apresentamos algumas definições de AM utilizadas neste minicurso:

- **Conjunto de Dados**. É a representação tabular dos atributos que representam os objetos estudados [29]. No caso desse minicurso seriam os dados medidos das redes como latência, vazão, perda de pacotes e jitter.
- **Atributo (*Feature*)**. Característica do tráfego da rede, obtida diretamente ou derivada (através de algum cálculo ou técnica). Cada atributo está associado a uma propriedade do objeto pesquisado, nesse caso o fluxo de rede [29].
- **Atributos preditivos**. São atributos utilizados como entradas para os modelos de AM. Normalmente a entrada é representada por um vetor de atributos [29].
- **Atributo alvo**. Também chamado de alvo ou saída, representa o fenômeno de interesse da previsão, em nosso caso pode ser a classe do tráfego, a estimativa de QoE ou apenas a identificação de um ataque malicioso [29].

Cada abordagem de AM pode escolher uma série de diferentes estratégias para aprender a função-alvo. Isso inclui a representação da experiência, incluindo matrizes de exemplos e atributos, pares de entrada e saída ou apenas entradas, interação com o meio ambiente; a representação da função aprendida, por exemplo, funções, regras, distribuições de probabilidade; e a forma como o método percorre o espaço de pesquisa para encontrar uma aproximação da função alvo [29].

Em relação ao tipo de experiência adquirida, os métodos de AM seguem três paradigmas principais, nas quais as tarefas de aprendizado são comumente classificadas:

aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço[1].

Aprendizado Supervisionado

Nesse paradigma, as tarefas são preditivas e o conjunto de dados de treinamento deve ter atributos (características) de entrada e de saída (também chamada de alvo). As saídas devem ser rotuladas simulando a atividade de um supervisor, ou seja, alguém que sabe a "resposta". A tarefa de aprendizado supervisionado pode ser definida da seguinte forma [44]:

> Dado um conjunto de treinamento de N pares de exemplos de entrada e saída
>
> $$(x_1,y_1),(x_2,y_2),...(x_n,y_n),$$
>
> onde cada y_i foi gerado por uma função desconhecida $y = f(x)$, o algoritmo deve encontrar uma função h que se aproxime da função verdadeira f.

A função de hipótese h deve ser válida para outros objetos do mesmo domínio que não pertençam ao conjunto de treinamento. A essa propriedade dá-se o nome de generalização. A baixa capacidade de generalização significa que o modelo e os dados estão super ajustados (*overfitting*), ou subajustado aos dados (*underfitting*) [29].

Adota-se como boa prática, quando lidando com AM, utilizar três conjuntos de dados: treinamento, validação e teste. O conjunto de treinamento é utilizado para ajustar o modelo, ou seja, são fornecidos dados para que o algoritmo aprenda: encontre uma função que aproxime f, a partir de exemplos conhecidos. O conjunto de validação é importante para avaliar a capacidade de generalização do modelo, verificando se ele não está superajustado e nem subajustado, além de permitir a escolha do modelo por meio do ajuste dos hiperparâmetros. Por fim, com o conjunto de teste, avalia-se o desempenho preditivo do modelo, constatando se ele resolve ou não o problema proposto.

As tarefas preditivas podem ser divididas em problemas de **classificação**, quando a saída y for um conjunto de valores qualitativos, por exemplo, tipo de tráfego de rede (*email*, *streaming*, *gaming*). Já quando a saída é um valor numérico, ex: *estimativa de QoE*, a tarefa é chamada de **regressão**.

Russel e I.Norvig [44] apresentam as seguintes definições:

- Classificação: $y_i = f(x_i) \in \{c_1,...,c_m\}$, isto é, $f(x_i)$ assume valores em um conjunto discreto, não ordenado;
- Regressão: $y_i \in \mathbb{R}$, isto é, $f(x_i)$ assume valores em um conjunto infinito e ordenado de valores.

[1] Outros tipos de supervisão também existem, a saber, aprendizado semi-supervisionad, quando apenas um subconjunto dos exemplos tem uma saída; e auto-supervisionado, quando o rótulo é extraído da própria tarefa sem a supervisão humana

Aprendizado Não Supervisionado

O objetivo das tarefas neste paradigma é descrever o conjunto de dados, descobrindo alguma ordem ou padrão relevante que auxilie na representação dos dados. Assim, não é necessário que o conjunto de dados seja rotulado e os algoritmos utilizam agrupamentos ou regras de associação entre grupos de atributos. O agrupamento identifica a similaridade dos itens separando-os em grupos distintos e, além disso, busca-se maximizar a não similaridade entre os itens de grupos diferentes.

Já as regras de associação buscam padrões frequentes entre grupos de atributos. Esta técnica é bem explorada no contexto de cestas de produtos para descobrir os itens que são comprados em conjunto, e é bastante utilizada para guiar as ações de marketing de lojas online [29]. Essas regras de associação vêm sendo aplicadas em problemas de redes, auxiliando na classificação de tráfego e na identificação de dados que não obedecem a um padrão conhecido, permitindo identificar anomalias em redes. Outro problema em que esta técnica pode ser utilizada é a detecção de intrusos, tanto em redes cabeadas como em redes sem fio. As regras de associação aprendidas permitem observar dados anômalos realizando a detecção de agentes maliciosos [50].

Aprendizado por Reforço

Diferente dos outros dois paradigmas, não há a necessidade de um conjunto de dados para treinamento. As tarefas simulam aprendizado por tentativa e erro, premiando as ações positivas e punindo as ações negativas. Essa é a forma com que os robôs aprendem a andar e, até mesmo, a dançar. Muitas aplicações estão relacionadas com a robótica, com jogos de estratégia e atividades difíceis de programar, por exemplo, o controle de um helicóptero autônomo que realiza voos acrobáticos. Uma desvantagem significativa dessa abordagem é a necessidade de muitas repetições para a obtenção de bons resultados, o que gera um enorme custo computacional [44].

Alguns Algoritmos de Aprendizado de Máquinas

A seguir, apresentamos uma breve descrição dos principais algoritmos de Aprendizado de Máquina encontrados na literatura que também foram aplicados na metrologia de redes:

- **k-NN**. O algoritmo dos vizinhos mais próximos (k-NN) é um método baseado em distância. Assim, o exemplo de entrada será classificado de acordo com as instâncias que estão próximos a ele no conjunto de treinamento. O parâmetro k define a quantidade de instâncias a serem levados em conta para a previsão. É considerado um algoritmo preguiçoso (*lazy*) por apenas memorizar os dados do conjunto de treinamento, sem necessariamente aprender um modelo. Tem dificuldade em lidar com grande número de atributos (dimensionalidade), por causa do cálculo da similaridade em um espaço dimensional muito grande, e o processo de classificação é lento, já que no pior caso, tem de percorrer todo o conjunto de dados [29].

- ***Naive* Bayes**. O algoritmo *Naive* Bayes é um modelo probabilístico que utiliza o Teorema de Bayes, assumindo que os valores dos atributos são independentes entre si. Assim, a probabilidade de um item pertencer à determinada classe está representada na Equação 1 e o algoritmo associa o exemplo x à classe y_k para a qual

$P(y_k|x)$ é máxima [29].

$$P(y_i|x) \propto P(y_i) \prod_{j=1}^{d} P(x_j|y_i) \tag{1}$$

- **Árvore de Decisão**. Uma árvore de decisão apresentada é um grafo acíclico direcionado em que cada nó ou é um nó de divisão, com dois ou mais sucessores, ou um nó folha. Cada nó de divisão realiza um teste para decidir o caminho a ser percorrido. Os nós folhas são rotulados com valores, representando regras e conclusões [29]. Existem diversos algoritmos que utilizam a formalidade das árvores de decisão, sendo um dos mais conhecidos o C4.5 [39], que tem uma implementação em JAVA com a denominação "J48" [35].

- **Máquinas de Vetores de Suporte - SVM**. Foram desenvolvidas utilizando a teoria de aprendizado estatístico buscando-se hiperplanos que separem linearmente os conjuntos de dados [13]. A equação de um hiperplano é apresentada na Equação 2, em que $w \,.\, x$ é o produto escalar entre os vetores w *e* x, $w \in X$ é o vetor normal ao hiperplano descrito e $\frac{b}{||w||}$ corresponde à distância do hiperplano em relação à origem, com $b \in \Re$ [29].

$$h(x) = w \,.\, x + b \tag{2}$$

- **Redes Neurais Artificiais**. As redes neurais foram inspiradas no modelo de sinapses biológicas, tentando simular o cérebro humano. Elas são compostas por unidades de processamento simples, neurônios artificiais (foram apresentados pela primeira vez em McCulloch e Pitts [28]), dispostas em camadas e interligadas por um grande número de conexões. Quando todos os neurônios de uma camada estão conectados à camada seguinte, diz-se que é uma camada completamente conectada. Cada conexão tem um valor associado, chamado de peso.

  O neurônio artificial executa uma combinação linear de suas entradas com os pesos associados, seguido pelo cálculo de uma função de ativação não-linear sobre a combinação linear. Os pesos associados às conexões influenciam na importância de cada entrada. Dessa forma, a rede aprende por meio dos dados. Para aprender os pesos, a abordagem usual é o emprego do algoritmo de retropropagação juntamente com a técnica de otimização do gradiente descendente [29].

  As redes de uma única camada resolvem apenas problemas linearmente separáveis, mas adicionando-se camadas e utilizando funções de ativação não lineares nas camadas intermediárias, é possível resolver problemas não lineares, tornando-se um aproximador universal. As redes multicamadas também são chamadas de redes perceptron multicamadas (MLP, do inglês *multilayer perceptron*) [29]. Um exemplo de rede MLP pode ser observado na Figura 3.2.

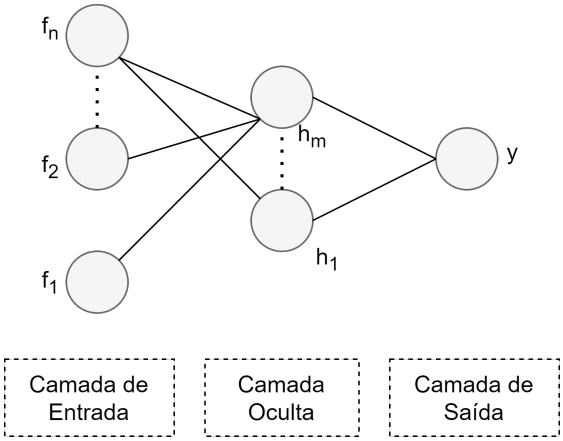

Figura 3.2. Rede MLP. Figura adaptada de Li et al. [23]

As redes neurais profundas são aquelas que possuem diversas camadas ocultas, podendo haver dezenas delas [2]. O aprendizado realizado por essas redes é chamado de *Deep Learning* possuindo algoritmos que apresentam excelentes resultados para uma variedade de problemas. Porém, tem como desvantagens o tempo de treinamento e a necessidade de uma grande quantidade de dados. Essas redes também são utilizadas para aprendizado de características especificas dos dados a serem analisados, eliminando a etapa de engenharia de atributos [1].

- **Redes Neurais Convolucionais (CNN)**. Similarmente às redes neurais, elas foram inspiradas na capacidade do cérebro de detectar e processar textos, imagens e vídeos. A etapa convolucional dessas redes realiza a detecção de atributos aplicando filtros que reduzem o tamanho da entrada, permitindo que o processamento seja mais rápido [6]. Elas utilizam pelo menos uma camada que realiza a operação de convolução [2]. Após isso, existe uma etapa conhecida como *pooling*, para reduzir a dimensionalidade e levar em consideração as características espaciais. As redes convolucionais podem ser utilizadas em uma gama diversificada de aplicações como detecção de objeto, classificação, processamento de texto e classificação de imagens. Dessa forma, as útlimas camadas da rede é que irão transformar a representação matricial em único vetor que determinará a saída da rede [34]. É comum essas redes serem utilizadas no domínio de duas dimensões, no caso de imagens, mas podem ser empregadas em problemas unidimensionais como os casos de séries temporais [6]. Um exemplo de rede convolucional pode ser visto na Figura 3.3.

- **Redes Neurais Recorrentes (RNN)**. Foram projetadas com o propósito de aprimorar as redes neurais fazendo com que fosse possível observar eventos anteriores. São compostas de unidades que conseguem observar a sequência dos elementos que passam pela rede [6]. São ideais para dados sequenciais como sentenças de textos, séries temporais e outras sequências como mapeamentos genéticos [2]. Elas possuem um estado oculto que guarda informações que irão interagir com o próximo elemento na sequência. A arquitetura básica pode ser vista na Figura 3.4.

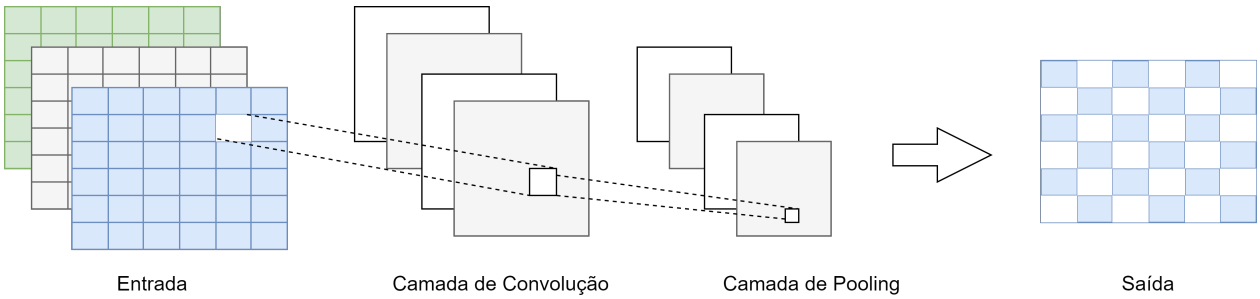

Figura 3.3. Rede Convolucional. Figura adaptada de Li et al. [23]

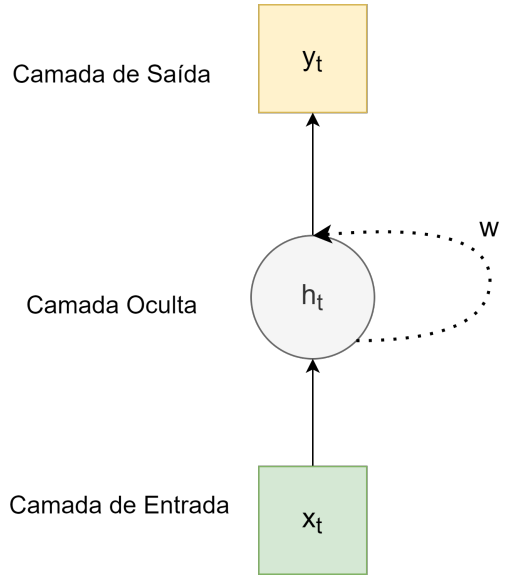

Figura 3.4. Rede RNN. Figura adaptada de Li et al. [23]

3.2.2. Qualidade de Serviços de Rede

Com a popularização de serviços oferecidos através da Internet e com a consequente convergência de infraestrutura de comunicação na web, além do desenvolvimento e utilização de diversos tipos de redes, por exemplo redes móveis, sem fio e dispositivos IoT, surgiu a preocupação com a qualidade de serviço da rede. Dessa forma, diversas métricas podem ser definidas para caracterizar o tráfego como *delay*, *jitter*, largura de banda e perda de pacotes [16]. Esses parâmetros são utilizados para medir a qualidade de serviço (QoS) em um fluxo de dados.

- **Jitter**. É a diferença entre o tempo estimado de entrada e o tempo real de entrada. Sinaliza uma variação no atraso da entrega dos dados, causando descontinuidade na sincronização do fluxo [32].
- ***Delay***. Tempo que o pacote leva para ir da origem até o destino. Essa métrica pode ser otimizada com a utilização de buffers [32].
- **Largura de banda**. Quantidade de dados que podem ser transmitidos em um período de tempo [8].

- **Taxa de perda de pacotes**. Quantidade de pacotes perdidos em relação à quantidade de pacotes enviados [8].

Cada aplicação demanda diferentes requisitos de qualidade de serviço, assim as métricas de QoS podem ser utilizadas para verificar a performance da rede. Portanto, para uma avaliação eficiente da rede é necessário que diferentes aplicações possuam distintos requisitos de QoS. Provedores de serviços elaboram acordos de níveis de serviço (SLA) comprometendo-se em garantir QoS para os usuários finais. Cada SLA acaba gerando especificações de níveis de serviço que irão guiar o monitoramento do QoS e SLA [43]. Uma das formas para se garantir o SLA é realizando classificação do tráfego de forma a prover os requisitos de QoS necessários em cada caso.

As métricas de qualidade foram projetadas para serem objetivas, focadas na avaliação do cumprimento dos requisitos de rede, não incluindo elementos qualitativos e nem subjetivos, por isso, em alguns casos, mesmo que os requisitos de QoS tenham sido cumpridos o usuário pode não ter ficado satisfeito [8, 16]. Por isso, foram desenvolvidas métricas de qualidade de experiência do usuário (QoE). Nessa busca pela satisfação do usuário, adota-se postura preventiva ao invés da reativa. Dessa forma, procura-se estimar os valores de QoS para que caso seja detectada uma degradação, medidas corretivas possam ser adotadas [41].

Ray [41] propôs uma rede formada por pacotes cognitivos. Esses pacotes teriam a capacidade de se rotearem e se comunicarem, buscando melhorar os valores de QoS, consumo de energia e até mesmo segurança. Esses pacotes cognitivos seriam ideais para utilizar em redes de IoT em que os dispositivos estão conectados entre si e o gerenciamento centralizado é dificultado. Esses pacotes utilizam algoritmos de aprendizado com redes neurais recorrentes para mapear atributos cognitivos e enriquecer a qualidade de serviço da rede.

Na busca pela performance das redes, além de medir as métricas de QoS pode-se medir a degradação da qualidade. Assim, em Nguyen e Thai [33] foi formulado o problema da degradação de QoS e os parâmetros pelos quais ele pode ser caracterizado. Além disso, foram propostos quatro algoritmos que tentam evitar esta degradação utilizando Aprendizado de Máquina. Os resultados foram otimistas encontrando limitações quanto ao tempo de execução, ou seja, nessa avaliação de QoS é interessante que as soluções possam ser aplicadas em tempo real [33].

3.2.3. Qualidade de Experiência do Usuário

A qualidade de experiência do usuário (QoE) é um indicador qualitativo da satisfação do usuário final com os serviços e aplicações utilizadas. Essa medida, por muitas vezes, é focada em avaliações subjetivas, interpretada em termos de percepção do usuário. Nessas avaliações, o usuário classifica o serviço ou aplicação com uma nota através de entrevistas ou questionários [21].

Esse conceito de QoE é muito importante, direcionando os investimentos e projetos de arquitetura na área de multimídias, principalmente nos serviços de vídeo para os usuários finais. Por exemplo, os grandes provedores de *streaming* armazenam os vídeos com diferentes qualidades de transmissão para que a experiência do usuário seja

otimizada, mesmo que os parâmetros de QoS da rede flutuem [8].

As métricas de QoE são afetadas por diversos fatores que podem ser divididos em três categorias principais: humanos, sistema e contexto. Dentre os fatores humanos que afetam a qualidade de experiência do usuário encontram-se o nível de escolaridade, idade, gênero, personalidade, humor, condições sócio-econômicas e até mesmo experiências de vida. Os fatores de sistema estão baseados em configurações técnicas e relacionadas ao conteúdo. Por exemplo, no caso de vídeos podemos citar a resolução, sincronização e as métricas de QoS. Os fatores de contexto relacionam-se ao ambiente em que o usuário recebe o conteúdo [8].

O conceito de QoE pode direcionar o entendimento e análise das operações de rede sobre a ótica do usuário final. Sendo, assim, um indicador importante para o controle da qualidade dos serviços, permitindo encontrar medidas que afetem a funcionalidade das redes complementando as métricas tradicionais de QoS [21]. Mesmo tendo naturezas distintas, QoS e QoE têm alto grau de correlação, por isso diversos estudos na literatura pesquisam a relação entre elas. Modelos de AM foram desenvolvidos para estimarem as métricas de QoE baseados nos valores de QoS como será visto em 3.3.4.

3.3. Uso de Aprendizado de Máquina em Metrologia de Redes

Nesta seção, trataremos os aspectos de Metrologia de Redes aplicados a áreas de redes sem fio e mobilidade, caracterização de topologia em diferentes níveis e redes definidas por software. Além disso, incluiremos discussão sobre medições voltadas aos usuários finais, principalmente as relacionadas com QoE.

As técnicas de metrologia buscam medir e monitorar o tráfego de redes nos seus diferentes níveis de granularidade, obtendo conhecimento da operação das redes, descobrindo como elas funcionam e analisando a performance delas em termos de QoS e QoE. Além disso, nessa utilização de técnicas para caracterização das redes faz sentido descobrir como os usuários consomem os serviços e como otimizar os recursos para melhor atendê-los. Unindo as informações a respeito da infraestrutura e do comportamento dos usuários é possível, propor, controlar e gerenciar acordos de níveis de serviço apropriados [1].

O crescimento da quantidade de redes móveis e celulares desperta o interesse em medições do desempenho fim-a fim dessas redes e o seu impacto em aplicações móveis [15]. Dentre essas medições encontramos aquelas baseadas em qualidade de experiência do usuário que constitui uma grande fonte de informações sobre o gerenciamento da rede. Nesse contexto, a implementação de Inteligência Artificial e Aprendizado de Máquina aumenta a eficiência do processo de monitoramento [21].

Abordaremos, ainda, de que forma podemos utilizar os parâmetros objetivos de qualidade do serviço tais como *delay*, *throughput*, perda de pacotes e *jitter* para estimar e predizer a qualidade de experiência do usuário utilizando Aprendizado de Máquina [21]. Além disso, é possível utilizar a Inteligência Artificial para estimar os múltiplos indicadores de qualidade de experiência do usuário, mesmo em uma rede que adote criptografia fim-a-fim [49, 27].

A caracterização de topologias sempre foi um dos objetivos das pesquisas de mo-

nitoração da Internet, que buscou ao longo dos anos a implementação e aperfeiçoamento de técnicas de descoberta de topologias nas camadas física, de rede e overlay. A descoberta de topologias em todos esses níveis é bastante útil no estudo de novos protocolos e serviços e análise das infraestruturas existentes. Então, métodos de Aprendizado de Máquina auxiliam na aquisição do conhecimento sobre o funcionamento das redes, o entendimento de como são as estruturas de redes existentes, permitindo estabelecer novas redes com menor esforço.

Dentre os diversos problemas de Metrologia de Redes que podem se beneficiar da utilização do Aprendizado de Máquina, podemos citar detecção de ataques de redes, predição de qualidade de experiência do usuário, detecção de anomalias em aplicações, estimativa de latência da rede [31] e falhas de redes. Nesse sentido, Zanotelli et al. [55] apresentam estudo para detecção de falhas na rede Ipê utilizando redes neurais, encontrando bons resultados quando regionalizaram os dados e as detecções. Sendo a globalização, ou seja, encontrar um modelo que conseguisse detectar falhas em qualquer ponto da rede nacional, um problema em aberto.

Em Streit et al. [48], os autores utilizam informações de tráfego residencial, fornecidas por um Provedor de Serviços de Internet, para analisar o tráfego de usuários domésticos. Neste trabalho, especificamente, eles comparam a QoE dos usuários de um período imediatamente antes com o período imediatamente depois do início da quarentena da COVID-19, em cidades do estado do Rio de Janeiro. Para isso, utilizam técnicas de decomposição tensorial, clusterização e classificação para identificar perfis de tráfego residencial dos clientes.

Dentre diversas aplicações, identificam-se trabalhos que utilizam medidas de telemetria de rede em plano de dados programável, em conjunto com modelos de Aprendizado de Máquina, para estimar métricas de qualidade de serviço [4]. Além dos desafios inerentes à Metrologia de Redes, discutiremos o estado da arte da medição em redes definidas por software (SDN), pontuando as principais características e dificuldades encontradas. Nota-se que a utilização de Aprendizado de Máquina para medições em rede é uma tendência atual. Diversos trabalhos têm lançado mão dessas técnicas para resolver problemas de medições de redes.

Entretanto, os dados capturados da rede para análise com os modelos de AM possuem características específicas que não são encontradas em aplicações mais comuns de Aprendizado de Máquina. Entre elas está a heterogeneidade, pois existe uma infinidade de dispositivos diferentes que podem estar conectados às redes estudadas, cada um deles consumindo e gerando tipos de tráfegos distintos [1]. Grande parte dos dados capturados estão criptografados, mantendo a privacidade dos usuários e a segurança das redes, porém dificultando a utilização desses dados para análises.

3.3.1. Definição de Fluxo de Tráfego

A maioria dos métodos de Aprendizado de Máquina aplicados em Metrologia de Redes utiliza o conceito de fluxo de tráfego. Esse conceito pode ser definido de diferentes formas. Neste minicurso utilizaremos a definição apresentada em Wei Wang et al. [54], por ser a mais utilizada na literatura.

Assim, um fluxo de tráfego é definido por meio dos pacotes que possuam os mesmos elementos, formando uma tupla : IP de origem, porta de origem, IP de destino, porta de destino e protocolo de transporte. Já uma sessão inclui fluxos em ambas direções, ou seja, temos o fluxo origem $\rightarrow$ destino e destino $\rightarrow$ origem [54]. Formalizando, temos:

- Tráfego bruto: o conjunto de pacotes é definido por $P = \{p^1,...,p^{|P|}\}$, onde cada pacote $p^i \in P$ é definido como $p^i = (x^i,b^i,t^i), i = 1,2....,|P|$. O primeiro elemento é a tupla x^i, o segundo elemento é o tamanho do pacote $b^i \in [0,\infty)$ em bytes e o último elemento $t^i \in [0,\infty)$ em segundos é o instante de início da transmissão.
- Fluxo de tráfego: é um subconjunto de um conjunto de tráfego bruto P, definido como $f = (x,b,d_t,t)$. O x representa a tupla, o b é a soma do tamanho de todos os pacotes em um fluxo, $d_t = t^n - t^1$ é a duração do fluxo e t é o início da transmissão do primeiro pacote.

3.3.2. Classificação de Tráfego

A tarefa de classificação de tráfego permite identificar os protocolos ou aplicações de determinado fluxo de tráfego, sendo uma importante ferramenta de gerenciamento que pode ser aplicada em tarefas tais como monitoramento de redes, provisionamento de qualidade de serviço, prioridade de tráfego, detecção de intrusão e aplicações de segurança [5, 3]. Existem três principais abordagens para essa tarefa, a saber, baseada em portas, baseada no conteúdo (payload) e baseada em informações estatísticas do fluxo [52].

- **Classificação baseada em portas.** A classificação baseada em portas identifica o tráfego de acordo com a padronização de portas de cada aplicação. Porém, este método é ineficiente ao lidar com a alocação de portas dinâmicas [5].
- **Classificação baseada em conteúdo.** A classificação baseada em conteúdo, também conhecida como *Deep Packet Inspection (DPI)*, identifica características do conteúdo dos pacotes (sequência de bytes) que diferem um protocolo de outro, sendo consideradas como a assinatura dos tipos de tráfego [42]. Essa abordagem enfrenta problemas de privacidade do conteúdo e não lida bem com fluxos criptografados, além de ser computacionalmente custosa [5].
- **Classificação baseada em fluxo.** A abordagem baseada em fluxo permite utilizar as características estatísticas para classificar o tráfego, permitindo a utilização de modelos de AM para identificar padrões existentes em cada fluxo. Essa abordagem utiliza parâmetros independentes, tais como comprimento dos pacotes, tempo de intervalo entre as chegadas e duração do fluxo. Dessa forma, ela consegue lidar com o problema da alocação dinâmica de portas, já que não depende das portas pelo qual o tráfego foi transmitido, também com o fluxo criptografado [52].

Um dos contextos em que estão sendo estudadas diversas aplicações de AM com classificação de tráfego refere-se às *smart cities*. As aplicações de Internet das Coisas (IoT do inglês *Internet of Things*) interconectam diversos objetos, aparelhos, sensores e dispositivos inteligentes criando uma rede complexa e que processa grande quantidade de

dados. Essas tecnologias são essenciais para a construção de *smart cities*, oferecendo serviços em diversas áreas de atuação, como educação, saúde, transporte e casas inteligentes [3]. Essas diferentes aplicações geram uma grande quantidade de dados e possuem diferentes requisitos de qualidade de serviço (QoS), como largura de banda, taxa de perda de pacotes, taxa de *delay* e *jitter* [5]. Assim, a classificação de tráfego permite implementar um mecanismo que diferencie o fluxo de tráfego de acordo com o tipo de aplicação (Ex: e-mail, jogos, *streaming*). Por fim, os recursos da rede podem ser alocados de forma a garantir os requisitos de QoS [5].

Nesse contexto, AlZoman e Alenazi [5] realizaram a comparação de quatro modelos de AM para classificação de tráfego, utilizando um conjunto de dados com características de tráfego semelhantes às *smart cities* como múltiplas fontes, vários tipos de dados e grande quantidade de tráfego. No conjunto de dados, existiam 11 tipos de tráfego e suas respectivas aplicações como navegação na web, acesso a banco de dados, jogos em rede, tráfego de ataques de segurança, multimídia entre outros. Os autores encontraram a melhor performance com algoritmos de árvore de decisão, com acurácia de 99.18%, superando os métodos tradicionais de classificação de tráfego. Ao avaliar a performance dos modelos, os autores compararam o tempo de execução dos diferentes algoritmos de AM, não utilizando apenas as métricas tradicionais, como acurácia, precisão e f1-Score, pois em aplicações de tempo real, o *delay* pode ser tão ou mais importante do que essas métricas [5].

A classificação de tráfego utilizando AM vêm se tornando um método de alta performance, embarcando inteligência nas funções de rede e melhorando o seu gerenciamento [5]. Apesar dessa evolução e da capacidade de lidar com uma grande quantidade de dados, ainda existem desafios na aplicação desses modelos e um deles é lidar com uma grande quantidade de atributos de entrada, ainda mais, quando eles são esparsos. Para solucionar este problema, uma das técnicas empregadas é a seleção de atributos que permite identificar os atributos essenciais para melhorar a performance dos modelos. Nesse processo de seleção, utiliza-se uma função de importância dos atributos que calcula o ganho de informação de todos os atributos e retorna os atributos com ganhos mais elevados, como visto em Alhumyani et al. [3].

Em Jonathan, Misra e Osamor [18], os autores compararam a aplicação de modelos de AM para classificação de tráfego sob duas perspectivas: executando diferentes métodos de seleção de atributos e sem realizar a seleção de atributos. Os resultados demonstram que a seleção de atributos possibilitou incrementar a performance dos modelos [18]. Dessa forma, a adoção de uma análise prévia do problema a ser tratado para a identificação da seleção de atributos ideal pode ser considerada como uma boa estratégia.

Além dos modelos tradicionais de AM, os pesquisadores têm proposto diversos modelos que se adaptem melhor aos tipos de dados utilizados. Pesquisas recentes utilizam redes neurais profundas (DNN) para classificação de tráfego, pois estas redes permitem a captura de características (que não são perceptíveis aos humanos) e conseguem lidar com grande quantidade de dados. Uma DNN possui uma camada de entrada, múltiplas camadas escondidas e uma camada de saída. Assim, em Alhumyani et al. [3] os autores desenvolveram uma DNN com sete camadas escondidas e um classificador que utiliza entropia máxima como função de custo na camada de saída para classificar o tráfego em

diferentes classes.

Resultados experimentais mostram que a rede proposta atinge acurácia máxima de 99.23%, superando alguns modelos tradicionais de AM, como SVM e KNN [3]. Para melhorar a performance, os autores realizaram uma seleção de atributos com o algoritmo *extra-trees* que cria diversas árvores de decisão para realizar a classificação. Assim, utilizando-se uma função de ganho de importância obtiveram-se os atributos mais importantes para melhorar o classificador, resultando em um total de 12 atributos, aplicando-os com a rede DNN proposta.

Outro modelo de rede neural utilizada em classificações de tráfego são as denominadas *autoencoders*. Essas redes são construídas com a mesma dimensionalidade nas camadas de entrada e de saída, mas as camadas intermediárias são implementadas com dimensionalidade reduzida. O objetivo desses modelos é reconstruir a informação da entrada na saída, a partir de transformações codificadoras e decodificadoras realizadas pelas camadas intemediárias. Espera-se que com esse processo, ocorra o aprendizado de outras características que permitam otimizar os classificadores [2].

Em Li et al. [24] é proposto um *stacked autoencoder* aprimorado para aprender as relações complexas sobre fontes múltiplas. Este modelo foi proposto para possibilitar que o modelo consiga lidar com a incerteza de grandes fluxos de redes. Especificamente, para modelar a incerteza, a rede é treinada utilizando uma estratégia não supervisionada que aplica a teoria Bayesiana de probabilidade para calcular as distribuições dos parâmetros do modelo. Os resultados demonstram que o modelo proposto supera os tradicionais em termos de acurácia.

O *stacked autoencoder* é treinado em duas fases. Na primeira, chamada de pré-treinamento, os parâmetros de cada camada oculta são aprendidos utilizando um aprendizado não supervisionado camada por camada. Na segunda fase, esses parâmetros são ajustados utilizando um método de aprendizado supervisionado para capturar as características finais. Uma importante propriedade do modelo proposto é a capacidade de lidar com a incerteza dos dados de uma maneira natural [24].

Redes Definidas por Software (em inglês, *Software Defined Networks* - SDN) é um paradigma de redes que permite a virtualização da infraestrutura de rede, desacoplando os planos de controle e dados e criando uma arquitetura dinâmica e flexível [52]. Além da camada de controle e da camada de dados, existe a camada de aplicação, completando a arquitetura SDN. O controlador central gerencia o fluxo de dados e conhece a topologia da rede, permitindo um controle de fluxo eficiente e facilitando as medições nessa camada [52]. Apesar disso, o gerenciamento de QoS fim-a-fim pelo controlador central pode ser dificultado devido aos diferentes requisitos de cada aplicação e usuários finais. Surge, então, a necessidade de se medir e monitorar as demais camadas de uma rede SDN [42].

Wang et al. [52] propuseram um arcabouço que potencializa os conceitos de SDN, permitindo monitoramento e gerenciamento em camadas separadas e garantindo controle e flexibilidade com otimização de recursos computacionais. Além disso, desenvolveram redes neurais denominadas Datanets baseadas em três arquiteturas de redes neurais: *multilayer perceptron*, *stacked autoencoder* e redes neurais convolucionais. As Datanets permitem classificar o tipo de tráfego, mesmo em redes criptografadas, sem ter acesso ao

conteúdo, mantendo a privacidade dos dados [52]. As Datanets são posicionadas no controlador SDN que realizam a classificação de tráfego. Este controlador central, ainda, cria e atualiza periodicamente essas redes conforme os dados são coletados dos diversos tipos de tráfego. Dessa forma, é possível realizar medições de QoS, monitoramento de redes, estimativa de QoE e detecção de malwares [52]. Assim, o controlador consegue entregar a QoS necessária para cada aplicação. Das redes criadas no estudo de Wang et al. [52], as redes convolucionais obtiveram os melhores resultados, com acurácia acima de 98%.

Para classificação de tráfego criptografado existem três abordagens principais. A primeira, diferencia tráfego criptografado do tráfego não criptografado. A segunda abordagem permite identificar o aplicativo que está gerando o fluxo de tráfego. A terceira abordagem visa identificar o tipo de tráfego, por exemplo, é possível determinar se o fluxo refere-se a um e-mail, a um serviço de streaming ou a uma rede social [53]. Quando redes neurais são utlilizadas para resolver problemas de classificação de tráfego, é possível aproveitar o conhecimento gerado em um campo distinto de aplicação para resolver problemas de rede. Nesse sentido, Wang et al. [53] propuseram um modelo fim-a-fim que aprende automaticamente atributos preditivos do fluxo de tráfego bruto aplicando uma modelagem feita em problemas de Processamento de Linguagem Natural (PLN), não necessitando da etapa de seleção de atributos. Na proposta dos autores, o tráfego de rede é sequencial e hierarquizado, estruturado em bytes, pacotes, sessões e fluxo de tráfego, que serve para comunicação de dois ou mais agentes, de maneira similar aos modelos de linguagem. Assim, há uma relação com a estrutura de caracteres, palavras, sentenças e artigos completos.

Então, utilizando-se de conhecimento para classificação de texto com redes convolucionais [20], Wang et al. [53] alimentam uma rede convolucional de uma dimensão com o tráfego bruto criptografado. A rede é alimentada com os primeiros 748 bytes de cada fluxo, extraindo os atributos preditivos e passando pela última camada que realiza a classificação de tipos de tráfego. Outra forma de modelo fim-a-fim em que não é necessário criar e selecionar atributos é proposta em Chen et al. [12]. Os autores propuseram uma rede convolucional de duas dimensões que é normalmente utilizada para classificar imagens. Dessa forma, o fluxo de tráfego é transformado em uma representação de imagens, adotando o *Reproducing Kernel Hilbert Space (RKHS)*. Após esse passo, a rede é alimentada com esses dados para realizar a classificação de tráfego. Esse modelo permite eliminar a etapa de engenharia de atributos não sendo necessário realizar atualizações periódicas da rede com novos exemplos de treinamento e lidar com fluxo de tráfego criptografado.

Outra abordagem para classificar tráfego está relacionada à aplicação de modelos de linguagens conhecidos como n-gramas. Zhao, Zhang e Sang [56] utilizam essa estratégia combinando *embeddings* de n-gramas, redes neurais e algoritmos de clusterização para construção de um esquema não supervisionado que resolve o problema de identificação de fluxo de tráfegos desconhecido. A etapa de aprendizado de atributos utiliza uma rede do tipo *autoencoder* que consegue aprender representações de dados não rotulados, como é o caso do fluxo de tráfego desconhecido nas redes. O tráfego dos pacotes é caracterizado via *embeddings* de n-gramas agrupando-os em cluster de tráfego desconhecido permitindo identificar dados não rotulados.

Mesmo com o avanço das aplicações de Aprendizado de Máquina na classificação de tráfego, em algumas situações a falta de uma base de dados rotuladas em tamanho suficiente impede que os modelos de AM alcancem resultados satisfatórios. Os resultados obtidos em Shahraki et al. [45] demonstram que o aprendizado ativo (AL, do inglês *Active Learning*) permite obter alto valor de acurácia, mesmo com uma quantidade pequena de dados. AL é um subcampo do Aprendizado de Máquina que tenta reduzir a quantidade de exemplos rotulados para o treinamento dos modelos. A ideia básica é consultar os exemplos rotulados de forma inteligente, buscando exemplos selecionados pelo algoritmo de AL para a construção do melhor modelo. Assim, um modelo aprendiz segue uma estratégia iterativa, selecionando exemplos de uma base não rotulada para que um oráculo, humano ou máquina, possa rotular.

Essa estratégia reduz o tempo e o custo de rotular todos os dados, pois utiliza uma quantidade significativa de menos exemplos [56]. Zhao, Zhang e Sang [56] demonstraram que os modelos implementados com AL superaram os modelos offline e alguns modelos que utilizavam redes neurais.

3.3.3. Predição de Tráfego

A predição de tráfego pode resolver diversos problemas de gerenciamento de redes, entre eles o provisionamento de recursos, a detecção de congestionamento e a tolerância a falhas. Apesar do sucesso obtido com a aplicação dos modelos de AM, ainda existem desafios a serem superados como alto custo computacional, dificuldade de treinamento e retreinamento, alta volatilidade dos dados e falta de dados rotulados [25]. As técnicas de predição de tráfego são aplicadas com diferentes escalas de tempo ou, até mesmo, em problemas independentes de tempo. Dessa forma, dividimos as aplicações de predição em curto prazo, quando se refere a escalas de tempo que vai de milissegundos a dias. Quando estamos falando de semanas, meses e anos chamamos de longo prazo [11].

O avanço da infraestrutura virtualizada possibilita a existência de inúmeras redes virtuais conectando diferentes dispositivos. Essas fatias de redes demandam serviços da infraestrutura de rede física subjacente, criando ambientes complexos e difíceis de controlar. Para resolver esse problemas foram propostas novas soluções que executam as tarefas de gerenciamento e alocação de recursos automaticamente, criando redes em que não há intervenção humana chamadas *zerotouch networks* [7]. Nessas redes autônomas é possível criar múltiplas instâncias lógicas da rede física, isolando o tráfego em cada uma delas. Assim, cada fatia é alocada para um tipo específico de tráfego de acordo com as aplicações. Dessa forma, é possível alocar diferentes serviços em uma mesma rede permitindo que eles coexistam [7].

A principal dificuldade imposta por essas redes autônomas é o gerenciamento de recursos. Essas fatias de redes exigem aumento dos requisitos de capacidade da rede, sendo necessário uma alocação de recursos eficiente, dinâmica e preemptiva para manter a infraestrutura operacional e controlar os custos da virtualização. Nesse contexto, em que as demandas de recursos mudam rapidamente, o tamanho e a complexidade das redes atingiram níveis em que não é possível ser gerenciada sob a ótica da percepção humana, uma orquestração e alocação de recursos orientada a dados vêm sendo implementada [7]. As redes virtualizadas são flexíveis e escaláveis, sendo muitas delas fornecidas como

software em serviços da Amazon AWS e Google Cloud. Apesar das vantagens e custos competitivos, nesses ambientes é comum ocorrer o erro de superdimensionamento, situação que aumenta a despesa sem necessidade pois os recursos alocados não são utilizados. Outro problema frequente é o subdimensionamento, em que poucos recursos são alocados, nesse caso o prejuízo financeiro é indireto, pois os serviços podem ficar fora de operação caso ocorram demandas que superem a capacidade da rede. Existe ainda o caso em que os recursos são alocados de forma correta, mas não se consegue mensurar o momento exato em que a infraestrutura deva ser reduzida ou aumentada, gastando mais do que o necessário [7].

O trabalho em Bega et al. [7] propõe um modelo que realiza a orquestração de redes virtualizadas e autônomas em duas escalas de tempo. Existe um orquestrador em escala de longo prazo que aloca uma capacidade dedicada para cada fatia e outro de longo prazo para capacidade compartilhada. Essas capacidades são constantes, porém cada fatia só tem acesso a sua capacidade dedicada e todas as fatias têm acesso às capacidades compartilhadas. Esse compartilhamento é gerenciado por um orquestrador de curto prazo, decidindo em que momento cada fatia terá esse recurso alocado. O framework apresentado chama-se AZTEC e é implementado em três blocos. O primeiro bloco é uma rede neural alimentada pelo fluxo de tráfego, transformado em imagens, que realiza a predição de longo prazo da capacidade dedicada. Esta rede consiste de três camadas 3D-CNN interligadas por camadas de *dropout*. O segundo bloco realiza a predição de longo prazo da capacidade compartilhada e possui uma rede neural com três camadas completamente conectadas com 128, 51 e 1 neurônio. O terceiro bloco possui uma rede semelhante à do primeiro bloco e realiza a predição da capacidade de curto prazo. Esse modelo proposto quando comparado com o estado da arte de orquestração possibilitou a redução de 47% do custo de alocação de recursos [7].

Outro problema preditivo que pode ser auxiliado com a utilização de AM é a predição do tamanho do fluxo em redes. Detectar fluxos muito grandes é importante para melhorar o roteamento, balanceamento e escalonamento nos diferentes tipos de redes. Essa detecção costuma acontecer após o congestionamento da rede devido ao fluxo. Aplicando modelos inteligentes é possível predizer o tamanho do fluxo de acordo com os dados do primeiro pacote, permitindo que o tráfego seja roteado e a rede não fique congestionada [37].

Esse problema foi atacado em Poupart et al. [37] que extrai os seguintes atributos dos primeiros pacotes: IP de origem, IP de destino, porta da origem, porta do destino, protocolo, identificação de servidor ou cliente e tamanho dos primeiros pacotes. Após essa extração, os autores analisaram regressores para determinar o tamanho do fluxo, encontrando os melhores resultados com um Processo Gaussiano de Regressão. Dessa forma, o modelo apresentado em Poupart et al. [37] permitiu estimar o tamanho do fluxo utilizando os primeiros pacotes e definir um limite para adoção de políticas de roteamento. Essa configuração permitiu que a rede funcionasse evitando congestionamentos devido ao tamanho dos fluxos. Outra vantagem desse modelo comparado às metodologias clássicas é que não existe a necessidade de alteração dos dispositivos de roteamento e conexão [37].

Outro modelo para calcular o tamanho do fluxo e tentar evitar congestionamentos é apresentado em Andreoletti et al. [6]. Eles empregam um modelo de AM baseado em

grafos chamado de Rede Neural Recorrente Convolucional de Difusão (DCRNN do inglês *Diffusion Convolutional Recurrent Neural Network*). A grande novidade desse modelo é capturar tanto as propriedades dos atributos, como as propriedades estruturais da rede. Esse modelo superou os demais modelos de redes neurais avaliados [6].

O objetivo é evitar a interferência humana no gerenciamento de redes, permitindo que sistemas inteligentes realizem tarefas de configuração, provisão, teste e detecção automaticamente. Dessa forma, Andreoletti et al. [6] tentam predizer o volume de tráfego em um enlace baseado na sequência histórica de tráfego. Então, a rede DCRNN executa uma tarefa de regressão minimizando o erro absoluto médio. O processo de difusão representa a relação entre dois nós do grafo, especificamente é a probabilidade que um caminho de K passos comece no primeiro nó e termine no segundo. A ideia das redes DCRNN é que a modelagem do processo de difusão permita descobrir a influência de cada nó para o resultado da predição. Assim, a representação do nó dentro do espaço de atributos é aprimorada com a aplicação dos filtros das camadas convolucionais [6].

Em Andreoletti et al. [6] foi utilizado o backbone da rede Abilene [47] que possui topologia pública, de 12 nós e 30 links, bem como dados estatísticos de tráfego real. Utilizando esses dados foi possível demonstrar que a rede DCRNN supera as demais na predição de volume de tráfego, sendo uma ferramenta eficiente para evitar congestionamentos. Os avanços na aplicação de Inteligência Artificial aos problemas de redes e o aumento da complexidade das mesmas criam a necessidade de criar gerenciadores adaptativos capazes de garantir boa qualidade de serviço (QoS) e oferecer qualidade de experiência do usuário (QoE) com segurança e baixo consumo de energia [14].

Dessa forma, Gelenbe et al. [14] propõem o desenvolvimento de redes auto-conscientes capazes de eliminar a complexidade de programação, gerenciar tarefas, de autoconfiguração, monitorar, gerenciar e corrigir falhas com mínima intervenção humana. Essas redes são possíveis desde que haja uma IA capaz de predizer o tráfego melhorando o roteamento, aumentando a confiabilidade, segurança e resiliência. A arquitetura proposta é uma rede SDN que utiliza pacotes cognitivos [41] capazes de realizar medições entre os nós de origem e destino. Esses pacotes compartilham informações com o controlador SDN que possui um mecanismo de decisão baseado em aprendizado por reforço com redes neurais recorrentes para modificar os caminhos do fluxo dinamicamente atingindo a melhor qualidade de serviço possível. Esse roteamento inteligente é executado predizendo o tráfego na rede SDN [14]. Além dos parâmetros de QoS essa rede auto-consciente busca melhorar a segurança e o consumo de energia. Eles implementaram um protótipo para análise e os resultados obtidos demonstraram performance superior às redes convencionais em termos de QoS, segurança e consumo de energia [14].

Foi visto que é possível utilizar Inteligência Artificial com Aprendizado de Máquina para criar redes *zerotouch* em que não há intervenção humana. Essas redes autônomas são mais eficientes, possuem melhor tempo de resposta, além de serem seguras e poderem oferecer até uma melhor controle do consumo de energia [14]. O gerenciador e tomador de decisão baseia-se na predição de fluxo de tráfego, evitando congestionamento de pacotes garantindo requisitos de QoS e buscando uma melhor experiência para o usuário. Essa análise em função do QoE foi realizada em Gelenbe et al. [14] e as redes autônomas superaram as redes tradicionais.

3.3.4. Estimativa de QoE

A estimativa de Qualidade de Experiência do Usuário (QoE) utilizando AM baseia-se no mapeamento de um conjunto de parâmetros de QoS objetivos e mensuráveis da rede, tais como latência, vazão, perda de pacotes e jitter, em uma medida da QoE do usuário. Dessa forma, é possível aplicar funções matemáticas nas medidas de QoS, encontrando valores para a QoE. O sucesso dos modelos para realizar essa estimativa deve-se ao grau de correlação entre essas medidas. Esse mapeamento ainda permite eliminar parte da subjetividade encontrada na estimativa de QoE [21]. O monitoramento de QoE é uma tarefa desafiadora em qualquer rede, nas redes móveis esse desafio é ainda maior. Devido à grande quantidade de usuários e dispositivos conectados, bem como à dificuldade de análise do tráfego de rede disponível. Além disso, a segurança e privacidade impõem outro desafio que é lidar com dados criptografados nas transmissões fim-a-fim.

O trabalho em Casas [9] investigou esse problema, realizando a inferência de QoE com a métrica *Speed Index* (SI) através de modelos de AM usando como entrada apenas dados de nível de pacote. Nele, os autores avaliam a QoE da navegação web em aplicativos de dispositivos móveis, como smartphones e tablets. O carregamento de uma simples página web pode envolver conteúdos que estejam localizados em diferentes servidores e diferentes fontes, por exemplo, é comum que uma página html exibida ao usuário possua dados provenientes de diferentes bancos de dados. Então, nessas situações a rede pode impactar a experiência do usuário. A abordagem para a solução do problema consiste na utilização de modelos supervisionados, por isso é necessário que o conjunto de dados contenha exemplos rotulados. Os atributos preditivos foram coletados e extraídos de tráfego criptografado no acesso das páginas web pelos dispositivos móveis [9].

A métrica SI indica o tempo necessário para que o usuário consiga visualizar a página web, mesmo que essa página não esteja completamente carregada. Por exemplo, se a imagem do rodapé demorar a carregar e o usuário for lendo e observando os outros elementos da página, ele pode não perceber essa demora. O que importa na experiência do usuário é que ele consiga visualizar os itens desejados, qualquer atraso no carregamento que não esteja visível será imperceptível. Os modelos foram treinados para inferir o SI como um problema de regressão, analisando os dados obtidos do fluxo de pacotes. Como existe diferença de QoE entre os diferentes tipos de dispositivos, os autores analisaram a inferência em três experimentos sendo que no primeiro eles utilizaram os smartphones, no segundo experimento foram utilizados apenas tablets. O terceiro experimento foi realizado com múltiplos dispositivos. Dessa forma, foi possível concluir que a abordagem baseada em AM para estimativa de QoE é confiável, produzindo uma baixa taxa de erro, além de conseguir utilizar atributos extraídos diretamente dos dados criptografados [9].

Essa estimativa de QoE a partir das métricas de QoS aplica-se a diversos casos, não estando restrita a um determinado tipo de conteúdo. Os autores em Ul Mustafa, Moura e Rothenberg [49] analisaram esse mapeamento utilizando AM no contexto de redes 5G para transmissão de vídeos. Nesse trabalho, eles verificaram o impacto na QoE dos usuários quando assistindo vídeos transmitidos com HTTP adaptive streaming (HAS), utilizando HTTPS para entregar criptografia fim-a-fim. Foi desenvolvido um algoritmo para extração de atributos de QoS que podem ser utilizados para mapear a QoE, com acurácia de 91%.

As transmissões de vídeos ao vivo (LIVE) cresceram enormemente nos últimos anos, principalmente durante a pandemia do COVID-19. Há uma variedade de tipos de vídeos como shows, transmissões de eventos esportivos, streaming de vídeos games, até aulas nas plataformas Twitch e YouTube. Essas transmissões ao vivo são mais suscetíveis a problemas de congestionamento de rede do que os vídeos sob demanda (VOD), necessitando de requisitos específicos para obter melhores medidas de QoE. Entretanto, existe um desafio para esse monitoramento devido a utilização da mesma infraestrutura para as transmissões ao vivo e VOD. Dessa forma, Madanapalli et al. [26] desenvolveram um método de aprendizado de máquina para detectar os vídeos que são transmitidos ao vivo e mensura a QoE baseado nas seguintes características de cada pedaço do fluxo: carimbo de tempo para o instante da requisição, início de transmissão do primeiro pacote, fim de transmissão do último pacote, número de pacotes, quantidade de bytes transmitidos.

O fato dos operadores de rede utilizarem a mesma infraestrutura para as transmissões de vídeos VOD e as transmissões ao vivo, dificulta a inspeção DPI que permitiria distingui-los. Porém, a extração de atributos do comportamento dos pacotes permite a construção de um modelo que realize essa separação. Madanapalli et al. [26] desenvolveram um método para estimar a resolução dos vídeos inferindo a medida de QoE, além disso eles apresentaram uma forma de detectar a presença de paradas de buffer, contribuindo para a estimativa de QoE. Tradicionalmente, a modelagem e medidas de QoE eram realizadas através da análise dos logs HTTP. Nessa abordagem, no entanto, os autores fornecem um preditor que atua em tempo real e com fluxo de tráfego criptografado. Assim, mesmo aqueles provedores que não têm acesso aos logs conseguem inferir a QoE das transmissões.

Poucos estudos são realizados para inferência de QoE em transmissões ao vivo, concentrando a maior parte dos trabalhos em vídeos VOD. Vídeos LIVE são gravados e transmitidos em tempo real, possuindo características distintas e requisitos específicos. A começar pelo buffer que é menor para os vídeos ao vivo, o usuário vai solicitando requisições HTTP e armazenando poucos segmentos do vídeo diminuindo a latência entre a transmissão e visualização pelo usuário final [26]. Já os vídeos VOD realizam requisições a um servidor que armazena os vídeos em diferentes resoluções pré-codificadas, permitindo sofisticados esquemas de compressão, bem como que o cliente mantenha um buffer maior evitando a deterioração das métricas de QoE. Outra diferença é que os clientes de transmissões ao vivo realizam downloads de segmentos do vídeo a cada dois segundos, enquanto que para vídeos VOD esse tempo sobe para dez segundos. Essa diferença de periodicidade no download de segmentos é uma das principais características que permite distinguir os dois tipos de vídeos [26].

Foi desenhada e implementada uma rede LSTM que aprende características comportamentais do fluxo de rede, evitando que seja necessário realizar a etapa de engenharia de atributos. A rede LSTM recebe um vetor de série temporal com a contagem das requisições de pacotes e passa por um classificador MLP que diz a probabilidade do fluxo ser uma transmissão ao vivo. Após isso, utiliza-se uma outra rede neural para predizer a parada de buffer, estimando o QoE entregue ao usuário [26]. Uma rede LSTM mantém um estado oculto e uma célula de estado. O estado da célula age como uma memória relembrando informações que serão utilizados na tarefa de classificação. O estado oculto é um canal de saída, que seleciona as informações da célula de estado que irão para o

classificador como visto na Figura 3.5.

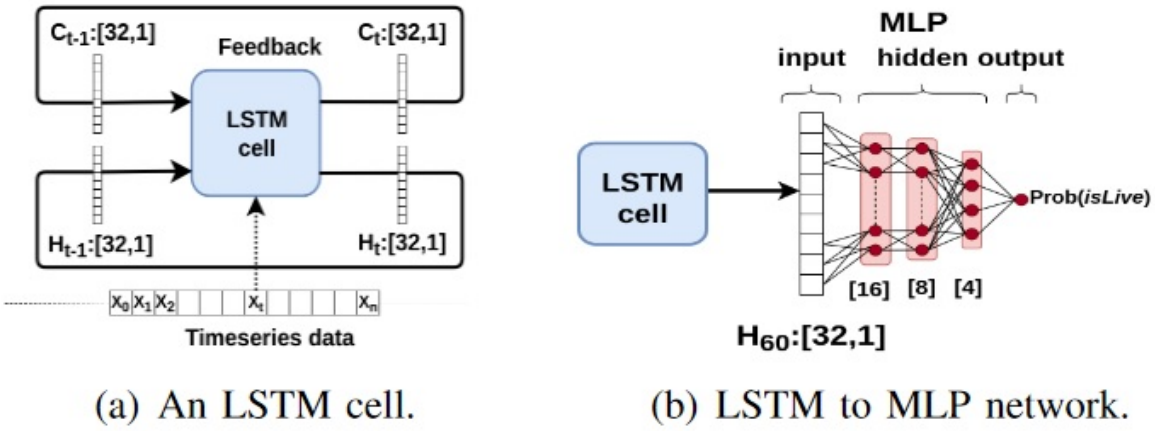

Figura 3.5. Rede LSTM Madanapalli et al. [26]

Com o crescimento e popularização da distribuição de vídeos na Internet, os usuários tornaram-se mais exigentes desejando assistir os vídeos online em alta resolução. Dessa forma, a QoE para os serviços de multimídia tornou-se uma métrica importante, já que os usuários esperam encontrar nos vídeos transmitidos a mesma qualidade encontrado no mundo offline. Sabe-se, no entanto, que diversos fatores influenciam a qualidade de uma transmissão através da rede [22].

Os provedores de serviços monitoram constantemente os parâmetros de rede para entregar a melhor experiencia para usuário. Entre as diversas técnicas encontra-se o mapeamento de qualidade de serviço para estimar a QoE, otimizando a qualidade dos vídeos. Porém, existem outros fatores que influenciam nessa experiência como o contexto social em que o usuário está inserido. Laiche et al. [22] estudam os fatores sociais e de engajamento de usuários que podem ser medidos e utilizados como atributos para estimar a QoE com modelos de AM, obtendo alta acurácia. A maior parte das métricas de QoE são subjetivas e refletem o grau de satisfação ou desapontamento do usuário. A mais utilizada é a MOS (do inglês mean opinion score), que é uma nota dada pelo usuário relacionada a experiência que ele experimentou. Esta medida é padronizada para ITU-T. O estimador proposto em Laiche et al. [22] tenta predizer o valor MOS de acordo com os atributos de contexto social.

A proposta dos autores em Laiche et al. [22] é que a contagem de visualizações, comentários e notas que ficam visíveis nas redes sociais refletem o grau de satisfação dos usuários com os vídeos relacionados. Pouco é conhecido sobre os efeitos da popularidade e do engajamento dos usuários sobre a medida final de QoE. Então as métricas de popularidade e engajamento nas redes sociais, foram utilizadas para predizer a qualidade de experiência do usuário. Assim, eles puderam identificar a influência que essas métricas possuem em relação à estimativa de QoE.

Laiche et al. [22] analisaram o modelo proposto com três algoritmos de AM: KNN, Random Forest e Árvore de Decisão. Foi encontrado o melhor resultado com a

árvore de decisão. Esse trabalho é relevante pois demonstrou que as informações do contexto social, popularidade e engajamento, contribuem para a estimativa de QoE. Os autores sugerem como trabalho futuro a utilização de redes neurais para incrementar a performance, além de demonstrar que o modelo proposto pode ser integrado em redes SDN.

3.3.5. Segurança de Redes

Um ataque de negação de serviço distribuído (DDoS do inglês *Distributed Denial of Service*) persiste como um problema de segurança das redes. Existem diversas abordagens para detectar tais ataques, incluindo a utilização de métodos baseados em AM. Os modelos propostos utilizam uma seleção manual de atributos baseada no entendimento de especialistas sobre o tema, o que gera um problema de generalização e os detectores acabam ficando muito especializados no contexto do ataque [51].

Ataques de DDoS enviam uma grande quantidade de tráfego para o sistema alvo, geralmente com a utilização de botnets. Esses bots frequentemente escravizam dispositivos que estão conectados à Internet para realizar esses ataques, inclusive beneficiando-se da enorme quantidade de dispositivos de IoT que acabam sendo fáceis de capturar. Essa grande quantidade de tráfego impede que os usuários reais tenham acesso ao serviço. Um dos grandes desafios para detectar esse tipo de ataque está associado ao fato dos agentes maliciosos utilizarem pacotes de dados normais, ou seja, não há indicação de tráfego malicioso, o que dificulta a utilização de classificadores de tráfego [51].

Wang, Lu e Qin [51] propuseram um classificador MLP para detectar ataques DDoS combinado com uma seleção automática de atributos com MLP que tenta encontrar um conjunto ótimo de atributos que incremente a performance do modelo detector do ataque. Além disso, no modelo proposto existe um mecanismo de feedback para detectar o momento em que o classificador deva ser atualizado. Os resultados encontrados demonstram que o modelo proposto é comparável aos demais encontrados na literatura com a vantagem de corrigir o detector quando a performance se deteriora. Esta é uma grande dificuldade dos modelos propostos com a utilização de AM, a acurácia obtida com os dados de treinamento não se reflete quando o modelo é posto em produção, em contato com dados do mundo real.

O modelo proposto utiliza uma classificação binária utilizando um classificador MLP. Na etapa de treinamento é um utilizado o algoritmo SBS (do inglês *Sequential Backward Selection*) para selecionar os atributos. Esse algoritmo funciona retirando cada atributo dos dados de treinamento e calculando o ganho ou a perda do modelo nessa situação. No final permanecem os atributos que incrementaram a performance do modelo. Dessa forma, o classificador é treinado com esses atributos finais. Na etapa de detecção existe um mecanismo de feedback que possui um limitador de erro, após esse limite ser atingido o classificador é atualizado com os novos exemplos que passaram pelo detector após a fase de treinamento. Essa estratégia tende a corrigir o detector quando a performance começa a deteriorar [51].

Os avanços tecnológicos recentes já permite o projeto de redes ad hoc veiculares (VANET do inglês *Vehicular Ad hoc Network*) que promete fornecer diversos serviços inteligentes de transporte em *smart cities*. Um modelo VANET pode ser visto na Figura

3.6, os veículos comunicam-se entre si, com os controladores SDN e com os RSU (Roadside Units) que facilita a transmissão de informações para outros veículos que passaraão no mesmo ponto. Porém, todo esse avanço vêm acompanhado de diversos perigos o que faz essas VANETs vulneráveis a vários tipos de ataques [46]. Os sistemas de detecção de intrusos (IDS do inglês *Intrusion Detection Systems*) tentam mitigar a possibilidade de ataques, mas eles ainda estão restritos a sub-redes, não abrangendo a VANET inteira. As SDN tentam mitigar esse problema oferecendo um gerenciamento centralizado de toda rede, permitindo que um IDS localizado no controlador possa verificar toda a VANET.

Para mitigar, ainda, o problema de ataque centralizado no controlador SDN é proposto um sistema colaborativo entre vários controladores SDN que mantém a comunicação e o funcionamento do IDS, mesmo que um controlador SDN fique inoperante. A arquitetura das VANETS interconecta a comunicação dos veículos aproveitando-se da utilização das RSUs, tudo sendo gerenciado por um controlador central SDN. Essas redes oferecem serviços como alertas de emergência, segurança das vias, tráfego eficiente, serviços para os motoristas como detecção de congestionamento, assistência para estacionamento e outros [46]. Shu et al. [46] utilizam redes neurais com *deep learning* e GAN (*Generative Adversarial Networks*) explorando SDN distribuídas para desenvolver um sistema de detecção de intrusão colaborativo (CIDS do inglês Collaborative Intrusion Detection System).

A mobilidade dos veículos nessas redes dificulta a localização de veículos maliciosos, porque no momento que um veículo com comportamento anormal é detectado ele pode se afastar do correspondente controlador SDN. Então, essa abordagem colaborativa entre múltiplos SDN permite que o aviso sobre o veículo malicioso seja transmitido para toda a rede VANET, identificando e gerenciando o elemento malicioso [46].

Uma botnet é uma rede composta de computadores que foram comprometidos e são controlados remotamente, executando tarefas comuns, espalhando vírus de computadores ou executando ataques de DDoS. Uma estratégia para se descobrir uma botnet é analisar o fluxo de comunicação entre dois pontos da rede para identificar a comunicação de um bot com o servidor de comando e controle (C&C), essa comunicação possui características estatísticas que podem ser detectadas utilizando AM. Entretanto, os modelos aplicados nesse problema precisam atuar em tempo real e serem confiáveis [36].

Pektaş e Acarman [36] apresentam uma rede neural para identificar botnets que combina camadas convolucionais, redes neurais recorrentes e extrai estatísticas baseadas em fluxo de redes entre dois hosts tais como duração, tamanho dos pacotes e tempo de chegada entre os pacotes. Essa abordagem pode ser aplicada com protocolos de comunicação que são criptografados, pois mesmo sem ter acesso ao conteúdo dos pacotes as informações estatísticas do fluxo de comunicação se mantém [36]. Na extração de atributos os autores utilizaram uma estrutura de grafos para representar o fluxo de comunicação entre dois hosts, assim cada fonte fonte e destino IP são representados por nós, e caso exista comunicação entre eles é acrescentada uma aresta no grafo. A rede neural é utilizada como um classificador binário e é composta de *embedding*, rede convolucional, LSTM e rede completamente conectada. O modelo proposto identifica botnets com acurácia de 99% com tempo de performance compatível com outros modelos, porém a fase de treinamento tem um custo de tempo elevado [36].

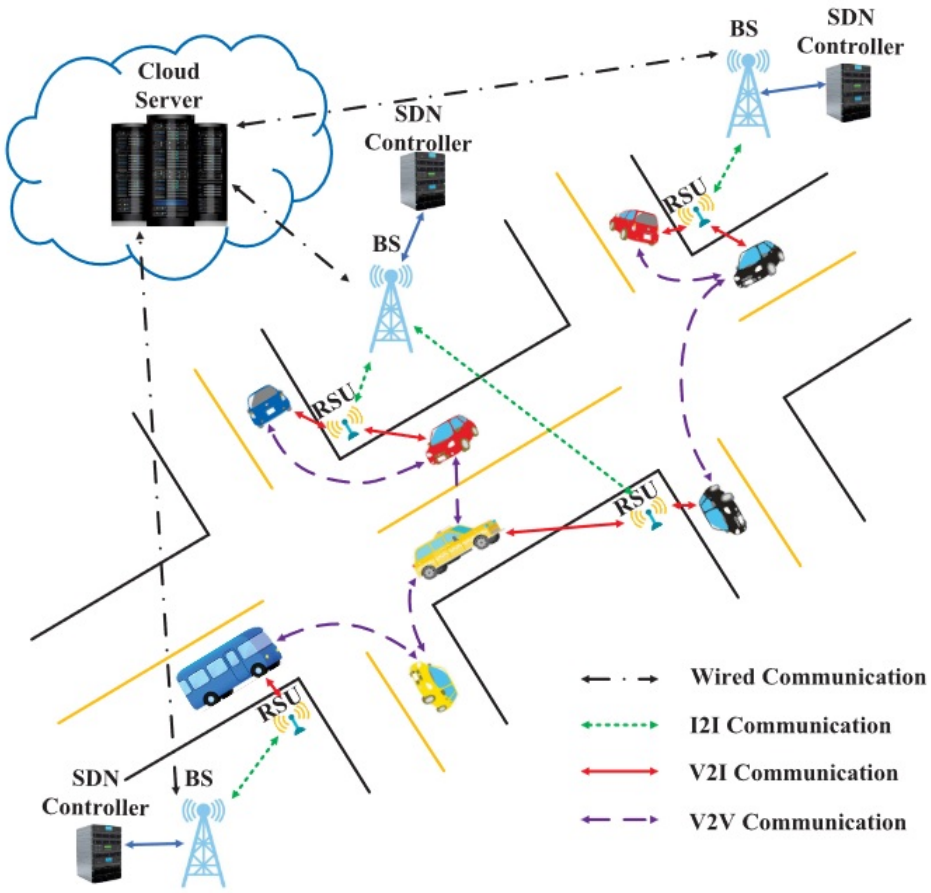

Figura 3.6. Modelo VANET Shu et al. [46]

A classificação de tráfego é utilizada para a detecção de anomalias na rede e exerce um papel importante no domínio de segurança de redes. Entretanto na detecção de malwares existe a dificuldade de se determinar os atributos relevantes para que o classificador seja bem sucedido. Tentando resolver este problema, Wei Wang et al. [54] propôs um classificador de malware que utiliza redes convolucionais para aprender uma representação dos dados do fluxo, que neste trabalho é representado como imagens. Os autores não extraem atributos dos dados, eles transformam o fluxo de tráfego bruto em imagens que alimentam uma rede CNN com diversas camadas. As camadas iniciais aprendem uma representação dos dados, enquanto que as camadas finais classificam as imagens em classes pré-determinadas como visto na Figura 3.7. Essa abordagem permite que o detector alcance acurácia de 99.41% mesmo com dados criptografados [54].

Devido a diferença de continuidade entre o fluxo de tráfego e sua representação em imagens, foram testadas diversas configurações sendo a que apresentou os melhores resultados as que utilizavam a sessão para alimentar as redes. [54]. Os tamanhos dos fluxos e sessões são variáveis de acordo com a transmissão realizada, mas a entrada da rede CNN deve ser uniforme. Então a solução encontrada foi a utilização dos primeiros 784 bytes da sessão. Caso a sessão excedesse esse número o restante da sessão era descartada, caso o tamanho fosse menor os bytes eram completados com 0 no final.

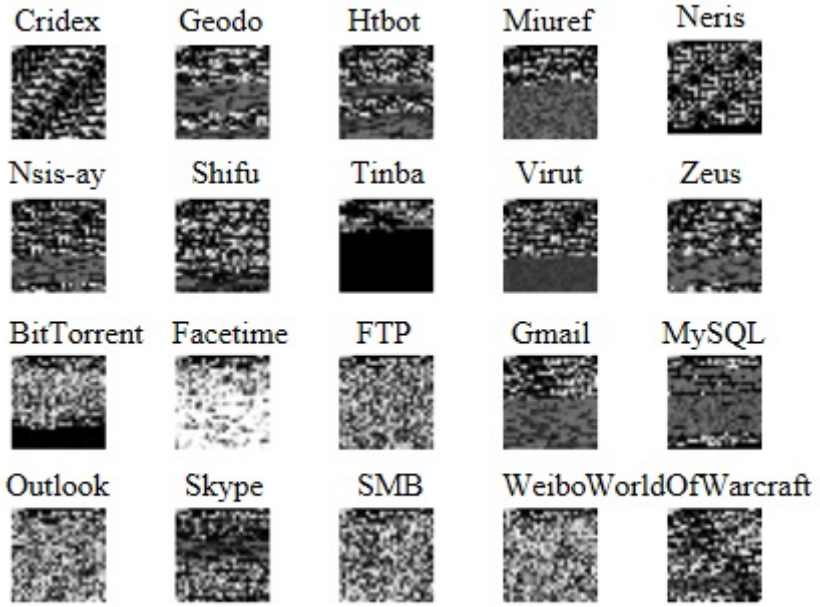

Figura 3.7. Visualização das Classes de Tráfego Wei Wang et al. [54]

As detecções de anomalias vêm se tornando cada vez mais importantes, conforme cresce o tráfego nas redes, principalmente devido à explosão de dispositivos de IoT. As aplicações de AM conseguem obter excelentes resultados nessa detecção, mas esse crescimento vertiginoso impõem dificuldades como velocidade e tempo de processamento devido à quantidade de dados excessiva. Dessa forma, além dos modelos de AM é necessário utilizar tecnologias Big Data desenvolvidas para oferecerem performance e capacidade adequada a tarefas com uma quantidade enorme de dados [38].

Pwint e Shwe [38] investigam a performance do Apache Spark aplicado aos problemas de detecção de anomalias valendo-se de diversos modelos de AM. Além disso, os autores testaram variedade de configuração de paralelismo de processamento e utilização de memória. A conclusão é de que o uso do Spark permite que o processamento dos modelos de AM ocorra em tempo hábil. Além disso, a distribuição e execução paralela das tarefas nos núcleos do Spark se adequam bem aos diversos modelos. Dessa forma, a utilização das tecnologias Big Data no domínio de segurança de redes permite superar as dificuldade de tempo de processamento.

Existem métodos não supervisionados que tentam detectar anomalias nas redes, prevenindo ataques cibernéticos. Entre essas propostas encontra-se o trabalho de Radford et al. [40], que se aproveita dos conceitos e modelagens de NLP para propor um detector de anomalias. Os autores demonstram que rede neural recorrente é capaz de aprender um modelo para representar sequências de comunicação entre computadores e detectar tráfegos anômalos. Espera-se que o fluxo de dados na rede compartilhe características com os modelos de linguagem, seguindo um conjunto de regras similares a uma gramática [40]. O fluxo de rede é tokenizado (separado em sequências menores) em sequência de bytes que foram considerados como "palavras"e que podem ser combinadas originando "sentenças"representativas da comunicação dos computadores. Essas "sentenças"foram

utilizadas para gerar um modelo que aprende semântica e gramática sintática dessa linguagem gerada. Foi utilizada uma rede recorrente com LSTM para capturar essas relações e nuances da linguagem [40]. Esse modelo é utilizado para predizer a comunicação entre dois IPs e o erro de predição é utilizado para identificar transmissões atípicas, caracterizando atividades maliciosas.

A rede implementada possuía duas camadas LSTM, uma camada densa de ativação, uma camada softmax completamente conectada como saída. Cada camada LSTM era composta de 50 células ocultas com ativação linear na primeira camada e ReLU na segunda camada. O treinamento realizado era não supervisionado, ou seja, não possuía exemplos rotulados. Os exemplos rotulados foram utilizados apenas para teste, verificando se as anomalias detectadas referiam-se a ataques de segurança. Para uma melhor avaliação do modelo proposto, realizou-se o treinamento da rede com duas bases distintas, uma com exemplos de ataques maliciosos e uma que não possuía tais ataques.

A ideia era que a base que não possuía ataques entre os exemplos permitisse que a rede detectasse melhor os tráfegos anômalos, já que esses tráfegos teriam características muito diferentes. Enquanto que o fato de possuir diversos exemplos de ataques poderia levar o modelo a "entender"que este tipo de tráfego é normal. Porém, isto não ocorreu. A rede treinada com exemplos maliciosos obteve performance superior ao detectar anomalias, os autores acreditam que isso possa ter acontecido devido ao desbalanceamento das bases de dados disponíveis [40]. Por fim, Radford et al. [40] demonstraram que o fluxo de tráfego pode ser associado a um modelo de linguagem que permite a detecção de tráfego anômalo, caraterizando assim fluxo malicioso. Entretanto, este modelo pode não detectar todos os tipos de ataques, sendo necessário combinar esse monitoramento com outros detectores.

3.3.6. Gerenciamento de Falhas

O crescimento das redes criou arquiteturas complexas e com uma grande quantidade de informações, tornando inviável que eles sejam monitoradas e gerenciadas manualmente. A evolução dos modelos de AM é possível criar ferramentas pra classificação de redes e detecção de falhas de forma autônoma. Os problemas de redes atuais, acabam sendo problemas de big data que são muito difíceis de serem resolvidos analiticamente [30]. A aplicação de algoritmos de AM permite não só resolver os problemas eficientemente, como mantêm o projeto de redes mais simples.

Em redes programáveis é possível criar regras para gerenciamento de falhas a ser entregue em forma de política para um orquestrador. Esse sistema centralizado então monitora a rede e quando detecta um serviço ou dispositivo com problema, ele desvia o tráfego ou aciona o dispositivo backup. Desabilita o aparelho, ou caminho, problemático e emite um aviso de problema informando a ação adotada [30]. Na operação clássica dos sistemas de rede, quando detectada uma falha ia ser emitido um alarme para uma equipe de prontidão que deveria localizar e corrigir a falha.

Zanotelli et al. [55] apresentam um método de predição de falhas com AM aplicados a Rede Ipê que interconecta universidade e centros de pesquisa de todo o país. A abordagem adotada utilizada uma rede LSTM para predizer se numa janela de tempo à frente a rede irá falhar ou não. Dessa forma, em redes programáveis caso a predição de

falha ocorra é possível adotar medidas para manter a disponibilidade da rede, sendo o preditor proposto uma excelente ferramenta de monitoramento.

Os modelos clássicos de interconectividade têm mudado de conexões de dispositivos físicos para ambiente altamente virtualizados em que planos programáveis permitem novas aboradagens para monitoramento e gerenciamento de redes. A aplicação de modelos de AM para predição de falhas é um deles, ao contrário dos modelos clássicos, eles podem ser aplicados em tempo real e tomarem decisões automáticas em tempos inimagináveis para a ação humana [55].

Para a realização dos testes e avaliação do modelo, foram conectados dados de diversos pontos de presença da Rede Ipê utilizando a ferramenta ViaIpê. Essa ferramenta reúne característica de qualidade da rede operada pela RNP, dando transparência para a qualidade dos enlaces da rede acadêmica brasileira. Os atributos retirados dos dados foram: perda de pacotes, RTT (milissegundos), Download e Upload (bits por segundo) [55]. A caracterização de falha utilizada em Zanotelli et al. [55]foi a taxa de perda de pacotes superior a 3%. O modelo tenta predizer se ocorrerá pelo menos uma falha no perído de 15 minutos a frente.

O modelo foi analisado em função das métricas de acurácia, precisão e revocação. Como a Rede Ipê apresenta poucas falhas, o conjunto de dados de treinamento era desbalanceado e o modelo obteve alta acurácia, mas com valores baixos de precisão e revocação. Ou seja, o modelo tinha problemas em detectar os casos positivos de falha. Os testes inciais foram realizados com um modelo global para todos os pontos de presença. Depois da constatação dessa baixa precisão, os autores regionalizaram o modelo, tendo um modelo treinado e utilizado para cada região do país. Com essa redução de heterogeneidade nos dados, os modelos obtiveram bons resultados em algumas regiões continuando insatisfatórios para outras regiões [55].

Os resultados são promissores para a utilização de modelos de AM para caraterização de falhas na Rede Ipê. Entretanto, ainda existe várias possibilidades de estudo para melhorar os resultados. Baseando-se em outros trabalhos já apresentados nesse minicurso, este problema poderia ser abordado com um modelo de aprendizado de representação que eliminasse a etapa de engenharia de atributos, bem como a transformação do fluxo em imagens para classificação entre falha de rede ou não.

Além da detecção de falhas, modelos de AM podem determinar classes para os estados da rede. Em Mohammed et al. [30], os autores determinam três estados que seria o normal, quando não há falhas, estado de falha, quando a comunicação é interrompida e congestionado, quando os parâmetros de QoS deterioraram comprometendo o QoE. Além disso, os autores pretendem identificar o enlace que ocorreu a falha para que ações corretivas possam ser tomadas.

Esse processo autônomo de identificação e localização de falhas pretende evitar custos operacionais e financeiros, automatizando uma importante tarefa para manter a disponibilidade das redes. Para a utilização do sistema foram utilizadas métricas de QoS e QoE como entrada do modelo. As métricas de QoS são. jitter, taxa de perda de pacotes, casos de pacotes fora de sequência, pacotes descartados. Para o QoE foi utilizada a velocidade de transferência de download [30].

Para realizar a classificação foram testados 4 modelos: árvore de decisão, os modelos de agrupamento GB (*Gradient Boosting*) e XGB (*eXtreme Gradiente Boosting*) e uma rede neural completamente conectada. O melhor resultado foi obtido com XGB obtendo acurácia superior a 99%. O conjunto de dados era desbalanceado com as classes de congestionamento e falha tendo menos exemplos do que o estado normal, porém essa é a realidade que o modelo irá se deparar no mundo real. Para tentar minimizar esse problema foi ajustado pesos diferentes para as classes na rede neural, mas os resultados não foram promissores. Ao analisar as métricas de precisão e revocação, constata-se que o modelo proposto com XGB é viável para a detecção e localização de falhas.

Redes de sensores sem fio (WSN - *Wireless Sensor Networks*) estão sujeitas a falhas, principalmente por causa dos ambientes em que são implantadas. Essa falahas, pode ocorrer em nível de software, hardware ou comunicação. Essa redes são compostas por sensores independentes conectados por canais sem fio. Sensores são equipamento projetados para realizar tarefas especificas, por muitas vezes eles são projetados para não serem intrusivos, por isso não dispõem de elevada capacidade de processamento. Exemplos de sensores são termômetros e medidores de pressão [34].

Redes WSN geralmente possuem restrições de energia e armazenamento, podendo ser utilizadas para monitorar saúde, em ambientes de vigilância, em aplicações militares e aplicações industriais. Muitas vezes os ambientes em que elas operam são de difícil acesso e não vão permitir intervenção humana em caso de falhas. Por essa razão, as WSNs necessitam de monitoramento de falhas eficiente e autônomo. Entretanto, o fato desses sensores não possuírem recursos computacionais faz com que o detector precise ser preciso e rápido e possa operam em ambientes com riscos elevados [34].

Então em Noshad et al. [34], os autores avaliam a utilização de 6 classificadores para detecção de falha em redes WSNs. Os experimentos foram realizados com medidores de umidade e termômetros, construindo uma rede artificial para que os dados pudessem ser coletados. As falhas foram introduzidas propositalmente simulando ações do mundo real. Avaliando diferentes classificadores como SVM, CNN, MLP, RF, redes neurais e gradiente descendente estocástico. Esses classificadores são posicionados nos nós centrais dos clusters de sensores, que são responsáveis pelo monitoramento da rede [34]. O melhor resultado encontrado foi com o Random Forest que superou os demais modelos em todas as métricas utilizadas.

Na busca por redes autônomas, que possam ser auto-gerenciadas, diversos modelos foram propostos. Em Huang et al. [17], os autores propõem uma arquitetura para gerenciamento de falhas que utilizam um algoritmo chamado GBRM baseado em redes neurais com auto-encoder. Esse modelo superou outros modelos populares de AM. Redes auto-gerenciadas pretendem funcionar sem intervenção humana, detectando falhas, analisando-as e corrigindo as mesmas. Esse processo é complexo e envolve diversas tomadas de decisão que utilizando AM podem ser realizadas em um intervalo de tempo ínfimo [17]. Esses modelos são orientados a dados combinado com análise estatísticas, demandando também a necessidade por equipamentos especificos que irão realizar medições e monitoramento.

As redes auto-encoders (DAE) possuem duas fases chamadas de encoder e decoder. Elas são similares e funcionam da seguinte forma: na fase de encoder os dados de

entrada passam por camadas que vão reduzindo a sua dimensionalidade, preservando as informações mais relevantes daquele exemplo, o decoder tenta reconstruir os dados originais. Como isso não é possível, o que a rede DAE realiza é minimizar o erro entre o exemplo original e a reconstrução [17]. Essas redes realizam treinamento não supervisionado por não necessitar de rótulos para realizar essa reconstrução. Um dos principais problemas dessas redes é a inicialização dos parâmetros, interferindo diretamente no desempenho da rede. Dessa forma, o algoritmo GBRBM proposto em Huang et al. [17] utiliza uma máquina restrita de Boltzmann(RBM) para pré-treinar a rede otimizando a performance do modelo. Os resultados contribuem para o desenvolvimento da arquitetura auto-gerenciado e alcança acurácia de 89.2%.

Nesta seção foram apresentados diversos modelos para serem aplicados em uma diversidade de problemas. Apesar do avanço realizado na abordagem orientada a dados com a utilização de Aprendizado de Máquina, ainda é difícil selecionar o melhor modelo para resolução de um problema específico. Realmente, não existe um modelo único que resolva todos os problemas [9]. Entretanto, existem iniciativas de agrupamento de modelos criando um super modelo de aprendizado que possa ser generalizado para mais de um tipo de problema [10].

Casas, Vanerio e Fukuda [10] introduzem o GML learning, um modelo genérico para analisar problemas de medições de redes, empregando técnicas de agrupamento seguindo os conceitos de modelo de super aprendizado. O GML learning é um agrupamento de modelos que busca encontrar a melhor combinação entre eles fornecendo uma predição mais precisa. Ele tem performance assintoticamente melhor que qualquer modelo individual do agrupamento [10]. Além disso, como a maioria dos métodos de agrupamento, ele exibe robustez quanto a incerteza apresentada nos dados [10].

Enquanto a aplicação de técnicas de aprendizado em problemas de medições de rede vêm sendo utilizadas intensamente, existem poucas abordagens a respeito dos métodos de agrupamento. Esse fenômeno acontece, mesmo sendo observado na prática que esses métodos atingem melhores resultados do que modelos simples. Os métodos de agrupamento, além de apresentarem melhor performance, combinam diferentes abordagens para resolver o mesmo problema, buscando complementaridade entre os diferentes modelos. Dessa forma, cada modelo potencializa o outro e compensa as limitações dos demais [10]. Entre os métodos tradicionais de agrupamento encontramos o *Bagging* e o *Boosting*.

- **Bagging**. Abreviação de *Bootstrap Aggregation* busca reduzir a variância dos modelos de previsão, gerando subconjuntos de dados de treinamento retirados do conjunto de dados original. Cada modelo individual é treinado com um subconjunto sorteado aleatoriamente, após isso ele é combinado com os demais utilizando um esquema de votação por maioria com mesmo peso para encontrar a predição final [10]. O exemplo mais conhecido dessa abordagem é o algorimto Random Forest.

- **Boosting**. Constrói incrementalmente um agrupamento, treinando cada nova instância de modelo com base no desempenho do modelo anterior. *Boosting* é uma abordagem de duas etapas, onde primeiro usa subconjuntos dos dados originais para produzir vários modelos e, em seguida, aumenta seu desempenho combinando-os,

também usando votação por maioria. A criação de novos conjuntos não é aleatória, depende do desempenho dos modelos anteriores, e cada novo subconjunto contém as instâncias mal classificadas pelos modelos anteriores. Ou seja, o modelo continua aumentando a quantidade de subconjunto enquanto estiver melhorando a sua performance [10]. Um exemplo conhecido é o algoritmo AdaBoost.

O GML learning foi testado com problemas diferentes incluindo detecção de ataques, detecção de anomalias e predição de QoE. Os resultados demonstraram que o GML learning superou os melhores modelos individuais aplicados em cada caso, bem como superou os métodos tradicionais de *bagging* , com a aplicação do Random Forest e *boosting*, com a utilização do AdaBoost [10]. Dessa forma, percebe-se que uma das principais vantagens do GML refere-se a capacidade de lidar com problemas diferentes utilizando o mesmo conjunto de dados e a mesma etapa de treinamento tornando-se uma iniciativa para uma generalização de melhores práticas em medições de redes.

Com base na análise de desempenho de diversas pesquisas e modelos, Casas, Vanerio e Fukuda [10] observaram que tanto as redes neurais quanto os modelos baseados em árvore de decisão fornecem, em geral, melhores resultados em termos de precisão e previsão do que outros modelos únicos. Além da vantagem de sobrecarga computacional muito menor para árvores de decisão em comparação com modelos baseados em redes neurais. Modelos baseados em árvore de decisão representam, portanto, um modelo de aprendizado de máquina muito atraente para análise de redes, não apenas por sua alta precisão e baixo custo computacional, mas também devido a uma série de propriedades incorporadas, como visibilidade do modelo, robustez ao ruído de entrada, entre outras [10].

3.4. Estudo de Caso

Para facilitar a compreensão do assunto, será proposto um estudo de caso com a utilização modelos de Aprendizado de Máquina para resolução de problemas de medições. Dentre os modelos apresentados, utilizaremos métodos interpretáveis, como Árvores de Decisão, e métodos com alto poder preditivo, como Random Forest e AdaBoost, Redes Neurais e Redes Neurais profundas. Além disso, serão utilizados alguns modelos clássicos e mais simples, como o método probabilístico Naive Bayes e o algoritmo baseado em distância KNN, para comparação. Incluiremos medições de tempo de execução dos modelos, para verificar quais poderiam ser utilizados em tempo real, o que sem dúvida é um desafio interessante.

Os resultados serão apresentados com uma discussão detalhada, incluindo suas vantagens e desvantagens, refinamentos adotados, possíveis melhorias e algumas explicações de resultados extremos. Explicaremos as métricas selecionadas para comparação, apresentando a motivação e justificativa para cada uma delas, enriquecendo o entendimento dos experimentos realizados. Durante a apresentação desse estudo de caso, pretende-se demonstrar o processo completo de construção de um modelo de Aprendizado de Máquina, desde a coleta de dados, perpassando as etapas de pré-processamento, construção do conjunto de dados, seleção de atributos, bem como as escolhas dos modelos a serem utilizados. O conjunto de dados a ser utilizado estará e será obtido de dados reais de monitoramento. Uma dessas fontes será a base de dados pública da RNP (Rede

Nacional de Ensino e Pesquisa). Os resultados, explicações, apresentação e implementações realizadas serão disponibilizadas em `https://github.com/loyoladesa/srbc2022`.

3.5. Conclusões

A quantidade de dispositivos móveis conectados às redes cresce exponencialmente impondo novos desafios para os administradores de redes. Esses profissionais buscam garantir os requisitos de qualidade de serviço (QoS) para proporcionar uma melhor qualidade de experiência (QoE) para o usuário final. Além dos dispositivos móveis, o crescimento de dispositivos IoT conectados contribuem para o aumento de tráfego, o crescimento da infraestrutura e da complexidade das redes. Dessa forma, novas maneiras de mensurar e gerenciar as redes foram propostas, aproveitando-se da evolução e capacidade dos modelos de Aprendizado de Máquina. Adotando uma metodologia orientada a dados, as ferramentas e técnicas de metrologia de redes contribuem para um melhor monitoramento e gerenciamento eficiente e preciso das redes.

Existe uma ampla gama de estudos demonstrando as oportunidades que se apresentam na utilização de AM no monitoramento de redes. Este trabalho focou em aplicações de metrologia apresentando modelos capazes de realizar a classificação e predição de tráfego, estimativa de QoE, gerenciamento de falhas e segurança de redes. Observa-se que uma das aplicações mais importantes é a classificação, tendo, consequentemente uma variedade de trabalhos a respeito. Ela é útil para que seja alocado os recursos necessários, de forma que se alcance os requisitos de qualidade de serviço para cada tipo de tráfego. Em alguns casos, pode bloquear serviços em ambientes corporativos (ex: bloquear o uso Netflix em empresas). A utilização de AM permite lidar e classificar tráfego criptografado, sem que seja preciso acessar o conteúdo dos dados, mantendo a privacidade dos usuários e segurança da rede.

A predição exerce um papel central no gerenciamento e alocação de recursos, principalmente em redes definidas por software. A identificação precoce de fluxos muito grandes permite que sejam adotadas medidas para evitar o congestionamento. Medidas essas, que com o auxílio de IA, são disparadas autonomamente sem intervenção humana em uma velocidade, antes inimaginável. Dessa forma, podemos utilizar as técnicas de predição para estimar QoE, aproveitando o poder dos modelos de AM. O mapeamento de QoE a partir das métricas de QoS é amplamente empregado para obter atributos preditivos do fluxo de tráfego, mesmo que ele esteja criptografado.

A segurança de redes é outro campo em que o Aprendizado de Máquina contribui significativamente para a performance das medições, pois grande parte da detecção de anomalias passa por algum sistema inteligente. Esses sistemas analisam os dados e descobrem padrões que permite classificar o tráfego em diferentes classes, detectando comportamentos maliciosos. A vantagem dessas aplicações fica visível no tratamento de ataques DDoS. A maneira tradicional de lidar com esse tipo de ataque é, uma vez que o ataque tenha sido detectado, disparar alarme e bloquear o tráfego ao servidor. Porém, nessa abordagem mesmo evitando um estrago maior o ataque já foi realizado. Com o auxilio dos modelos de AM é possível identificar o comportamento de DDoS, antes que o ataque se manifeste, adotando-se medidas preventivas.

As redes tornaram-se complexas e difíceis de gerenciar, mas utilizando-se ferramentas e técnicas de Metrologia é possível compreender o seu funcionamento, caracterizando-as e retornando mensurações para avaliação de performance. Os modelos de AM permitem atingir outro nível, melhorando o monitoramento e gerenciamento. Algumas propostas foram realizadas para que no futuro, as redes sejam auto-gerenciadas, diminuindo a intervenção humana, sendo capazes de detectar anomalias, analisar e disparar ações necessárias para a correção das falhas.

As técnicas, ferramentas e modelos apresentados abrangeram uma variedade de tipos de redes como redes móveis, redes sem fio, redes IoT, *smart cities*, backbone, redes domésticas e redes definidas por software. Essa ampla gama de contextos indica que a Inteligência Artificial pode ser utilizada para resolver problemas de redes, bem como caracteriza-las e ampliar o nosso entendimentos sobre as mesmas.

Os principais desafios para o futuro residem no gerenciamento de dados provenientes de fontes heterogêneas presentes em redes IoT e em *smart cities*. Além dessas, as redes de comunicação que irão auxiliar veículos autônomos tanto terrestres quanto aéreos precisa de um gerenciador de falhas preciso, autônomo e que funcione em tempo real em velocidade quase que instantânea. Esse é um problema desafiador para futuras pesquisas.

Agradecimentos

Este trabalho é parcialmente financiado pela RNP e CNPq (projeto 300123/2021-3). Agradecemos também à RNP pela sessão dos dados utilizados na parte prática deste minicurso.

Referências

[1] Mahmoud Abbasi, Amin Shahraki e Amir Taherkordi. "Deep Learning for Network Traffic Monitoring and Analysis (NTMA): A Survey". en. Em: *Computer Communications* 170 (mar. de 2021), pp. 19–41. ISSN: 01403664. DOI: 10.1016/j.comcom.2021.01.021. URL: https://linkinghub.elsevier.com/retrieve/pii/S0140366421000426 (acesso em 14/04/2022).

[2] Charu C. Aggarwal. *Neural Networks and Deep Learning: A Textbook*. en. Cham: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-94463-0. URL: http://link.springer.com/10.1007/978-3-319-94463-0 (acesso em 29/04/2022).

[3] Hesham Alhumyani et al. "An Efficient Internet Traffic Classification System Using Deep Learning for IoT". Em: *Computers, Materials & Continua* 71.1 (2022), pp. 407–422. ISSN: 1546-2226. DOI: 10.32604/cmc.2022.020727. URL: https://www.techscience.com/cmc/v71n1/45387 (acesso em 22/04/2022).

[4] Leandro Almeida, Fábio Verdi e Rafael Pasquini. "Estimando métricas de serviço através de In-band Network Telemetry". Em: *Anais do XXXIX SBRC*. Disponível em https://sol.sbc.org.br/index.php/sbrc/article/view/16725 Acessado em 07 de maio de 2022. Uberlândia: SBC, 2021, pp. 252–265.

[5] Razan M. AlZoman e Mohammed J. F. Alenazi. "A Comparative Study of Traffic Classification Techniques for Smart City Networks". Em: *Sensors* 21.14 (8 de jul. de 2021), p. 4677. ISSN: 1424-8220. DOI: 10.3390/s21144677. URL:

https://www.mdpi.com/1424-8220/21/14/4677 (acesso em 22/04/2022).

[6] Davide Andreoletti et al. “Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks”. en. Em: *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. Paris, France: IEEE, abr. de 2019, pp. 246–251. ISBN: 978-1-72811-878-9. DOI: 10.1109/INFCOMW.2019.8845132. URL: https://ieeexplore.ieee.org/document/8845132/ (acesso em 03/05/2022).

[7] Dario Bega et al. “AZTEC: Anticipatory Capacity Allocation for Zero-Touch Network Slicing”. en. Em: *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*. Toronto, ON, Canada: IEEE, jul. de 2020, pp. 794–803. ISBN: 978-1-72816-412-0. DOI: 10.1109/INFOCOM41043.2020.9155299. URL: https://ieeexplore.ieee.org/document/9155299/ (acesso em 03/05/2022).

[8] Abdelkader Benmir et al. “Survey on QoE/QoS Correlation Models for Video Streaming over Vehicular Ad-hoc Networks”. en. Em: *Journal of Computing and Information Technology* 26.4 (mar. de 2019), pp. 267–287. ISSN: 13301136, 18463908. DOI: 10.20532/cit.2018.1004278. URL: http://cit.fer.hr/index.php/CIT/article/view/4278 (acesso em 05/05/2022).

[9] Pedro Casas. “On the Analysis of Network Measurements Through Machine Learning: The Power of the Crowd”. Em: (2018), pp. 1–8. DOI: 10.23919/TMA.2018.8506486.

[10] Pedro Casas, Juan Vanerio e Kensuke Fukuda. “GML learning, a generic machine learning model for network measurements analysis”. en. Em: *2017 13th International Conference on Network and Service Management (CNSM)*. Disponível em http://ieeexplore.ieee.org/document/8255998/ Acessado em 07 de maio de 2022. Tokyo: IEEE, nov. de 2017, pp. 1–9. ISBN: 978-3-901882-98-2. DOI: 10.23919/CNSM.2017.8255998. (Acesso em 22/04/2022).

[11] Aaron Chen, Jeffrey Law e Michal Aibin. “A Survey on Traffic Prediction Techniques Using Artificial Intelligence for Communication Networks”. en. Em: *Telecom* 2.4 (dez. de 2021), pp. 518–535. ISSN: 2673-4001. DOI: 10.3390/telecom2040029. URL: https://www.mdpi.com/2673-4001/2/4/29 (acesso em 03/05/2022).

[12] Zhitang Chen et al. “Seq2Img: A sequence-to-image based approach towards IP traffic classification using convolutional neural networks”. Em: *2017 IEEE International Conference on Big Data (Big Data)*. 2017 IEEE International Conference on Big Data (Big Data). Disponível em http://ieeexplore.ieee.org/document/8258054/ Acessado em 07 de maio de 2022. Boston, MA: IEEE, dez. de 2017, pp. 1271–1276. ISBN: 978-1-5386-2715-0. DOI: 10.1109/BigData.2017.8258054.

[13] Harris Drucker et al. “Support Vector Regression Machines”. Em: *Advances in neural information processing systems* 9 (2003), pp. 155–161.

[14] Erol Gelenbe et al. “Self-Aware Networks That Optimize Security, QoS, and Energy”. en. Em: *Proceedings of the IEEE* 108.7 (jul. de 2020), pp. 1150–1167. ISSN: 0018-9219, 1558-2256. DOI: 10.1109/JPROC.2020.2992559. URL: https://ieeexplore.ieee.org/document/9103525/ (acesso em 06/05/2022).

[15] Utkarsh Goel et al. “Survey of End-to-End Mobile Network Measurement Testbeds, Tools, and Services”. Em: *IEEE Communications Surveys Tutorials* 18.1 (2016), pp. 105–123. DOI: 10.1109/COMST.2015.2485979.

[16] A HajaAlaudeen, E Kirubakaran e D Jeya Mala. “Approaches for Utility-Based QOE/QOS Architecture for Streaming Server in a Heterogeneous Wireless Device Based on SVM”. en. Em: (2015), p. 6.

[17] Huakun Huang et al. “Machine Fault Detection for Intelligent Self-Driving Networks”. en. Em: *IEEE Communications Magazine* 58.1 (jan. de 2020), pp. 40–46. ISSN: 0163-6804, 1558-1896. DOI: 10.1109/MCOM.001.1900283. URL: https://ieeexplore.ieee.org/document/8970164/ (acesso em 24/04/2022).

[18] Oluranti Jonathan, Sanjay Misra e Victor Osamor. “Comparative Analysis of Machine Learning techniques for Network Traffic Classification”. Em: *IOP Conference Series: Earth and Environmental Science* 655.1 (1 de fev. de 2021), p. 012025. ISSN: 1755-1307, 1755-1315. DOI: 10.1088/1755-1315/655/1/012025. URL: https://iopscience.iop.org/article/10.1088/1755-1315/655/1/012025 (acesso em 24/04/2022).

[19] Kemp, Simon. *Digital 2021:Global Overview Report*. Disponível em https://datareportal.com/reports/digital-2021-global-overview-report Acessado em 07 de maio de 2022. 2021.

[20] Yoon Kim. “Convolutional Neural Networks for Sentence Classification”. en. Em: *arXiv:1408.5882 [cs]* (set. de 2014). arXiv: 1408.5882. URL: http://arxiv.org/abs/1408.5882 (acesso em 30/04/2022).

[21] Georgios Kougioumtzidis et al. “Machine Learning for QoE Management in Future Wireless Networks”. Em: *2021 XXXIVth General Assembly and Scientific Symposium of the International Union of Radio Science*. 2021, pp. 1–4. DOI: 10.23919/URSIGASS51995.2021.9560226.

[22] Fatima Laiche et al. “When Machine Learning Algorithms Meet User Engagement Parameters to Predict Video QoE”. en. Em: *Wireless Personal Communications* 116.3 (fev. de 2021). Disponível em https://link.springer.com/10.1007/s11277-020-07818-w Acessado em 07 de maio de 2022., pp. 2723–2741. ISSN: 0929-6212, 1572-834X. DOI: 10.1007/s11277-020-07818-w.

[23] Ming Li et al. “A deep learning method based on an attention mechanism for wireless network traffic prediction”. en. Em: *Ad Hoc Networks* 107 (out. de 2020), p. 102258. ISSN: 15708705. DOI: 10.1016/j.adhoc.2020.102258. URL: https://linkinghub.elsevier.com/retrieve/pii/S1570870519310923 (acesso em 03/05/2022).

[24] Peng Li et al. “An Improved Stacked Auto-Encoder for Network Traffic Flow Classification”. Em: *IEEE Network* 32.6 (nov. de 2018), pp. 22–27. ISSN: 0890-8044, 1558-156X. DOI: 10.1109/MNET.2018.1800078. URL: https://ieeexplore.ieee.org/document/8553650/ (acesso em 16/04/2022).

[25] Iraj Lohrasbinasab et al. “Fromstatistical-to machine learning-based network traffic prediction”. en. Em: *Transactions on Emerging Telecommunications Technologies* (2021), p. 102258. DOI: 10.1002/ett.4394. (Acesso em 03/05/2022).

[26] Sharat Chandra Madanapalli et al. “ReCLive: Real-Time Classification and QoE Inference of Live Video Streaming Services”. en. Em: *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. Tokyo, Japan: IEEE, jun. de 2021, pp. 1–7. ISBN: 978-1-66541-494-4. DOI: 10.1109/IWQOS52092.2021.9521288. URL: https://ieeexplore.ieee.org/document/9521288/ (acesso em 04/05/2022).

[27] Mateus Marim et al. “Caracterização e Classificação do Tráfego da Darknet com Modelos Baseados em Árvores de Decisão”. Em: *Anais do XXXIX SBRC*. Disponível em https://sol.sbc.org.br/index.php/sbrc/article/view/16716 Acessado em 07 de maio de 2022. Uberlândia: SBC, 2021, pp. 127–140.

[28] Warren S. McCulloch e Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. Em: *The bulletin of mathematical biophysics* 4 (1943), p. 19.

[29] Tom M Mitchell. *Machine Learning*. McGraw-Hill Science, 1997.

[30] Ayse Rumeysa Mohammed et al. “Machine Learning-Based Network Status Detection and Fault Localization”. en. Em: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–10. ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2021.3094223. URL: https://ieeexplore.ieee.org/document/9474482/ (acesso em 22/04/2022).

[31] Shady A. Mohammed, Shervin Shirmohammadi e Sa’di Altamimi. “A Multimodal Deep Learning-Based Distributed Network Latency Measurement System”. en. Em: *IEEE Transactions on Instrumentation and Measurement* 69.5 (mai. de 2020), pp. 2487–2494. ISSN: 0018-9456, 1557-9662. DOI: 10.1109/TIM.2020.2967877. URL: https://ieeexplore.ieee.org/document/8963623/ (acesso em 24/04/2022).

[32] K. Mor et al. “Evaluation of QoS Metrics in Ad-Hoc Wireless Sensor Networks using Zigbee”. en. Em: *International Journal of Computer Sciences and Engineering* 6.3 (mar. de 2018), pp. 90–94. ISSN: 23472693. DOI: 10.26438/ijcse/v6i3.9094. URL: http://www.ijcseonline.org/full_paper_view.php?paper_id=1766 (acesso em 05/05/2022).

[33] Lan N. Nguyen e My T. Thai. “Network Resilience Assessment via QoS Degradation Metrics: An Algorithmic Approach”. en. Em: *arXiv:1902.01701 [cs]* (fev. de 2019). arXiv: 1902.01701. URL: http://arxiv.org/abs/1902.01701 (acesso em 05/05/2022).

[34] Zainib Noshad et al. “Fault Detection in Wireless Sensor Networks through the Random Forest Classifier”. en. Em: *Sensors* 19.7 (abr. de 2019), p. 1568. ISSN: 1424-8220. DOI: 10.3390/s19071568. URL: https://www.mdpi.com/1424-8220/19/7/1568 (acesso em 24/04/2022).

[35] Poonam Pandey e Radhika Prabhakar. “An analysis of machine learning techniques (J48 AdaBoost)-for classification”. Em: *2016 1st India International Conference on Information Processing (IICIP)*. 2016, pp. 1–6.

[36] Abdurrahman Pektaş e Tankut Acarman. “Deep learning to detect botnet via network flow summaries”. en. Em: *Neural Computing and Applications* 31.11 (nov. de 2019), pp. 8021–8033. ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-018-3595-x. URL: http://link.springer.com/10.1007/s00521-018-3595-x (acesso em 01/05/2022).

[37] Pascal Poupart et al. “Online flow size prediction for improved network routing”. en. Em: *2016 IEEE 24th International Conference on Network Protocols (ICNP)*. Singapore: IEEE, nov. de 2016, pp. 1–6. ISBN: 978-1-5090-3281-5. DOI: 10.1109/ICNP.2016.7785324. URL: http://ieeexplore.ieee.org/document/7785324/ (acesso em 22/04/2022).

[38] Phyo Htet Pwint e Thanda Shwe. “Network Traffic Anomaly Detection based on Apache Spark”. Em: *2019 International Conference on Advanced Information Technologies (ICAIT)*. 2019 International Conference on Advanced Information Technologies (ICAIT). Yangon, Myanmar: IEEE, nov. de 2019, pp. 222–226. ISBN: 978-1-72815-173-1. DOI: 10.1109/AITC.2019.8920897. URL: https://ieeexplore.ieee.org/document/8920897/ (acesso em 20/04/2022).

[39] J. R. Quinlan. *C4.5: Programs for Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann Publishers Inc., 1993.

[40] Benjamin J. Radford et al. “Network Traffic Anomaly Detection Using Recurrent Neural Networks”. Em: *arXiv:1803.10769 [cs]* (28 de mar. de 2018). Disponível em http://arxiv.org/abs/1803.10769 Acessado em 07 de maio de 2022. arXiv: 1803.10769.

[41] Partha Pratim Ray. “A survey on cognitive packet networks: Taxonomy, state-of-the-art, recurrent neural networks, and QoS metrics”. en. Em: *Journal of King Saud University - Computer and Information Sciences* (jun. de 2021), S1319157821001324. ISSN: 13191578. DOI: 10.1016/j.jksuci.2021.05.017. URL: https://linkinghub.elsevier.com/retrieve/pii/S1319157821001324 (acesso em 05/05/2022).

[42] Antonio Rocha et al. “Revisitando Metrologia de Redes: Do Passado às Novas Tendências”. Em: *Minicursos do SBRC*. 2016.

[43] Carlos Rodrigues et al. “An ontology for managing network services quality”. en. Em: *Expert Systems with Applications* 39.9 (jul. de 2012), pp. 7938–7946. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.01.106. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417412001248 (acesso em 05/05/2022).

[44] Stuart J. Russel e Peter I.Norvig. *Artificial Intelligence: A Modern Approach*. Pearson, 2020.

[45] Amin Shahraki et al. "Active Learning for Network Traffic Classification: A Technical Study". Em: *IEEE Transactions on Cognitive Communications and Networking* 8.1 (mar. de 2022), pp. 422–439. ISSN: 2332-7731, 2372-2045. DOI: 10.1109/TCCN.2021.3119062. URL: https://ieeexplore.ieee.org/document/9566310/ (acesso em 22/04/2022).

[46] Jiangang Shu et al. "Collaborative Intrusion Detection for VANETs: A Deep Learning-Based Distributed SDN Approach". en. Em: *IEEE Transactions on Intelligent Transportation Systems* 22.7 (jul. de 2021), pp. 4519–4530. ISSN: 1524-9050, 1558-0016. DOI: 10.1109/TITS.2020.3027390. URL: https://ieeexplore.ieee.org/document/9216536/ (acesso em 01/05/2022).

[47] *SNDlib*. Acessado em 05/05/2022. URL: http://sndlib.zib.de/home.action.

[48] Ananda Streit et al. "Efeito do confinamento causado pela pandemia Covid-19 nos perfis de tráfego residencial". Em: *Anais do XXXIX SBRC*. Disponível em https://sol.sbc.org.br/index.php/sbrc/article/view/16724 Acessado em 07 de maio de 2022. Uberlândia: SBC, 2021, pp. 238–251. DOI: 10.5753/sbrc.2021.16724.

[49] Raza Ul Mustafa, David Moura e Christian Esteve Rothenberg. "Machine Learning Approach to Estimate Video QoE of Encrypted DASH Traffic in 5G Networks". Em: *2021 IEEE Statistical Signal Processing Workshop (SSP)*. 2021, pp. 586–589. DOI: 10.1109/SSP49050.2021.9513804.

[50] Hui Wang et al. "Mining Association Rules for Intrusion Detection". en. Em: *2009 Fourth International Conference on Frontier of Computer Science and Technology*. Disponível em http://ieeexplore.ieee.org/document/5392848/. Acessado em 12 de maio de 2022. Shanghai, TBD, China: IEEE, dez. de 2009, pp. 644–648. ISBN: 978-1-4244-5466-2 978-0-7695-3932-4. DOI: 10.1109/FCST.2009.22.

[51] Meng Wang, Yiqin Lu e Jiancheng Qin. "A dynamic MLP-based DDoS attack detection method using feature selection and feedback". en. Em: *Computers & Security* 88 (jan. de 2020). Disponível em https://linkinghub.elsevier.com/retrieve/pii/S0167404819301890 Acessado em 07 de maio de 2022., p. 101645. ISSN: 01674048. DOI: 10.1016/j.cose.2019.101645.

[52] Pan Wang et al. "Datanet: Deep Learning Based Encrypted Network Traffic Classification in SDN Home Gateway". Em: *IEEE Access* 6 (2018). Disponível em https://ieeexplore.ieee.org/document/8473682/ Acessado em 07 de maio de 2022., pp. 55380–55391. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2872430.

[53] Wei Wang et al. “End-to-end encrypted traffic classification with one-dimensional convolution neural networks”. Em: *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*. 2017 IEEE International Conference on Intelligence and Security Informatics (ISI). Disponível em http://ieeexplore.ieee.org/document/8004872/ Acessado em 07 de maio de 2022. Beijing, China: IEEE, jul. de 2017, pp. 43–48. ISBN: 978-1-5090-6727-5. DOI: 10.1109/ISI.2017.8004872.

[54] Wei Wang et al. “Malware traffic classification using convolutional neural network for representation learning”. en. Em: *2017 International Conference on Information Networking (ICOIN)*. Da Nang, Vietnam: IEEE, 2017, pp. 712–717. ISBN: 978-1-5090-5124-3. DOI: 10.1109/ICOIN.2017.7899588. URL: http://ieeexplore.ieee.org/document/7899588/ (acesso em 30/04/2022).

[55] Vitor Zanotelli et al. “Caracterização e Previsão de Falhas em Serviços de Conectividade: uma Aplicação à Rede Ipê”. Em: *Anais do XXXIX SBRC*. Disponível em https://sol.sbc.org.br/index.php/sbrc/article/view/16717 Acessado em 07 de maio de 2022. Uberlândia: SBC, 2021, pp. 141–154.

[56] Shuyuan Zhao, Yongzheng Zhang e Yafei Sang. “Towards Unknown Traffic Identification via Embeddings and Deep Autoencoders”. Em: *2019 26th International Conference on Telecommunications (ICT)*. 2019 26th International Conference on Telecommunications (ICT). Disponível em https://ieeexplore.ieee.org/document/8798803/ Acessado em 07 de maio de 2022. Hanoi, Vietnam: IEEE, abr. de 2019, pp. 85–89. ISBN: 978-1-72810-273-3. DOI: 10.1109/ICT.2019.8798803.

[57] Artur Ziviani e Otto Carlos M. B. Duarte. “Metrologia na Internet”. Em: *Minicursos do SBRC*. Fortaleza, CE, 2005.